

Lab. Creating Linux Server Instance on AWS EC2

1. 목적

Amazon EC2(Elastic Compute Cloud)를 사용하여 Linux 인스턴스를 생성하고 접속하는 방법을 학습한다. 또한 생성된 Linux 서버의 시작, 중지 및 EC2 인스턴스에 대한 삭제 방법을 다룬다. 이 학습은 AWS Free-Tier를 활용하여 진행한다.

2. 사전 준비물

- AWS Free-Tier 계정
- Google Chrome or Mozilla Firefox

3. Tasks

- Task1. AWS Login
- Task2. VPC Network 구성하기
- Task3. 보안 그룹 생성하기
- Task4. EC2 Instance 생성하기
- Task5. Ubuntu Linux 인스턴스 접속하기

Task1. AWS Login

1. 웹 브라우저를 열고 <https://aws.amazon.com/ko/> 에 접속한다. 우상단에 [콘솔에 로그인] 버튼이 보이면 클릭하여 로그인한다. 가급적이면 Root 계정이 아닌 IAM 계정으로 로그인한다.

aws

Sign in as IAM user

Account ID (12 digits) or account alias
[REDACTED]

IAM user name
[REDACTED]

Password
.....

Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

English ▾

[Terms of Use](#) [Privacy Policy](#) © 1996-2023, Amazon Web Services, Inc. or its affiliates.

SERVERLESS

AWS Application Composer

Visually design and build serverless applications quickly

[Learn more ›](#)



Task2. VPC Network 구성하기

1. AWS Console에서 [서비스] > [네트워킹 및 콘텐츠 전송] > [VPC]를 클릭하여 들어간다.

The screenshot shows the AWS Cloud Map service page. On the left, there's a sidebar with various service categories like AR 및 VR, AWS 비용 관리, Customer Enablement, Machine Learning, Quantum Technologies, and many more. A red box highlights the '네트워킹 및 콘텐츠 전송' (Networking & Content Delivery) category. Below it, another red box highlights the 'VPC' service, which is described as a '격리형 클라우드 리소스' (isolated cloud resource). Other services listed include CloudFront, Direct Connect, Global Accelerator, AWS Private 5G, and Route 53.

2. VPC 페이지로 들어왔다. 먼저 확인할 것은 언어는 [한국어]로 설정하고, 또한 화면 우측 상단의 Region이 "서울"인지 확인한다.

The screenshot shows the AWS VPC service page. At the top right, there's a dropdown menu showing '오래전' (Old) and '지금' (Now), with '지금' highlighted by a red box. In the bottom left corner of the main content area, there's a red box around the '한국어' (Korean) language selection button. The rest of the page displays various VPC-related resources and metrics, such as VPCs, NAT Gateways, and Network ACLs, all in the '미국 서부' (US West) region.

3. 만일 Region이 "서울"이 아니라면 다음 그림과 같이 설정하여 [아시아 태평양(서울)]로 맞춘다.

The screenshot shows the AWS VPC Service Dashboard. On the right side, there is a sidebar titled 'Additional Information' which lists various AWS regions. The region 'Asia Pacific (Seoul) ap-northeast-2' is highlighted with a red box. Other regions listed include us-east-1, us-east-2, us-west-1, us-west-2, af-south-1, ap-east-1, ap-south-1, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ca-central-1, eu-central-1, eu-west-1, and eu-west-2.

4. 페이지 위쪽의 [VPC 생성]을 클릭하여 VPC 생성을 시작하도록 한다.

The screenshot shows the VPC Dashboard. On the left, there is a sidebar titled 'Virtual Private Cloud' with a 'Create VPC' button highlighted with a red box. The main area displays a summary of Amazon VPC resources: 2 VPCs in Asia Pacific (Seoul), 6 subnets, 3 routing tables, 2 internet gateways, 2 network ACLs, and 6 security groups. The 'Create VPC' button is located at the top center of the dashboard.

5. [VPC 생성] 페이지에서, 다음과 같이 설정한다. 나머지는 기본값을 사용한다.

- [생성할 리소스] : [VPC 등]
- [이름 태그 자동 생성] : [자동 생성] Check, “dockerlab”
- [IPv4 CIDR 블록] : “10.0.0.0/16”

VPC > VPC > VPC 생성

VPC 생성 정보

VPC는 AWS 클라우드의 격리된 부분으로서, Amazon EC2 인스턴스와 같은 AWS 객체를 관리합니다.

VPC 설정

생성할 리소스 정보
VPC 리소스 또는 VPC 및 기타 네트워킹 리소스만 생성합니다.

VPC만 VPC 등

이름 태그 자동 생성 정보
이름 태그의 값을 입력합니다. 이 값은 VPC의 모든 리소스에 대한 이름 태그를 자동으로 생성하는데 사용됩니다.

자동 생성
dockerlab

IPv4 CIDR 블록 정보
CIDR 표기법을 사용하여 VPC의 시작 IP와 크기를 결정합니다.

10.0.0.0/16 65,536 IPs

IPv6 CIDR 블록 정보
 IPv6 CIDR 블록 없음
 Amazon 제공 IPv6 CIDR 블록

테넌시 정보
기본값 ▾

6. 계속해서 다음과 같이 설정하고 나머지 값은 기본값을 그대로 사용한다.

- [가용 영역(AZ) 수] : 1
- [첫 번째 가용 영역] : ap-northeast-2a
- [퍼블릭 서브넷 수] : 1
- [프라이빗 서브넷 수] : 1
- [ap-northeast-2a 퍼블릭 서브넷 CIDR 블록] : "10.0.10.0/24"
- [ap-northeast-2a 프라이빗 서브넷 CIDR 블록] : "10.0.20.0/24"

가용 영역(AZ) 수 정보
서브넷을 프로비저닝할 AZ 수를 선택합니다. 고가용성을 위해서는 최소 2개 이상의 AZ를 사용하는 것이 좋습니다.

1	2	3
---	---	---

▼ AZ 사용자 지정

첫 번째 가용 영역

ap-northeast-2a

퍼블릭 서브넷 수 정보
VPC에 추가할 퍼블릭 서브넷 수입니다. 인터넷을 통해 공개적으로 액세스할 수 있어야 하는 웹 애플리케이션에는 퍼블릭 서브넷을 사용합니다.

0	1
---	---

프라이빗 서브넷 수 정보
VPC에 추가할 프라이빗 서브넷 수입니다. 프라이빗 서브넷을 사용하여 퍼블릭 액세스가 필요 없는 백엔드 리소스를 보호합니다.

0	1	2
---	---	---

▼ 서브넷 CIDR 블록 사용자 지정

ap-northeast-2a 퍼블릭 서브넷 CIDR 블록

10.0.10.0/24 256 IPs

ap-northeast-2a 프라이빗 서브넷 CIDR 블록

10.0.20.0/24 256 IPs

7. 계속해서 다음과 같이 설정하고, [VPC 생성] 버튼을 클릭한다.

- [NAT 게이트웨이] : [1개의 AZ에서]
- [VPC 엔드포인트] : [S3 게이트웨이]
- [DNS 옵션] : 2개 모두 체크



8. [VPC 워크플로 생성] 페이지를 통해 성공적으로 각 리소스가 생성되었음을 확인할 수 있다. [VPC 보기]를 클릭한다.

VPC > VPC > VPC 생성 > VPC 리소스 생성

VPC 워크플로 생성

성공

세부 정보

- VPC 생성: vpc-05f27e14a8345ce8a
- DNS 호스트 이름 활성화
- DNS 확인 활성화
- VPC 생성 확인: vpc-05f27e14a8345ce8a
- S3 엔드포인트 생성: vpce-0d611cb69f51ce749
- 서브넷 생성: subnet-03f25fa8a7558fb8d
- 서브넷 생성: subnet-0d3de635d81bb3997
- 인터넷 게이트웨이 생성: igw-0ae5c8794428fc567
- VPC에 인터넷 게이트웨이 연결
- 라우팅 테이블 생성: rtb-05faf87a18cf886a7
- 경로 생성
- 라우팅 테이블 연결
- 탄력적 IP 할당: eipalloc-0dc751e35f251338d
- NAT 게이트웨이 생성: nat-05376f7241bec7ae1
- NAT 게이트웨이가 활성화될 때까지 대기
- 라우팅 테이블 생성: rtb-0caa737835ccfb75a
- 경로 생성
- 라우팅 테이블 연결
- 라우팅 테이블 생성 확인
- S3 엔드포인트를 프라이빗 서브넷 라우팅 테이블과 연결: vpce-0d611cb69f51ce749

VPC 보기

9. 방금 생성한 VPC의 정보를 확인할 수 있다.

VPC > VPC > vpc-05f27e14a8345ce8a

vpc-05f27e14a8345ce8a / dockerlab-vpc

세부 정보 정보

VPC ID vpc-05f27e14a8345ce8a	상태 Available	DNS 호스트 이름 활성화됨	DNS 확인 활성화됨
데넌시 Default	DHCP 옵션 세트 dopt-0a5ca099f137a6ff6	기본 라우팅 테이블 rtb-0742e49bd2a9be2fc	기본 네트워크 ACL acl-02e10cc1388868f6c
기본 VPC 아니요	IPv4 CIDR 10.0.0.0/16	IPv6 플 -	IPv6 CIDR(네트워크 경계 그룹) -
네트워크 주소 사용 지표 비활성화됨	Route 53 Resolver DNS 방화벽 규칙 그룹 -	소유자 ID 789534828835	

Resource map New CIDR 플로우 로그 태그

Resource map 정보

VPC 세부 정보 표시 AWS 가상 네트워크

Introducing the VPC resource map

서브넷(2개)
이 VPC 내의 서브넷

ap-northeast-2a
dockerlab-subnet-public1-ap-northeast-2a
dockerlab-subnet-private1-ap-northeast-2a

라우팅 테이블(3개)
네트워크 트래픽을 리소스로 라우팅

rtb-0742e49bd2a9be2fc
dockerlab-rtb-public
dockerlab-rtb-private1-ap-northeast-2a

네트워크 연결(3개)
다른 네트워크에 연결

dockerlab-igw
dockerlab-nat-public1-ap-northeast-2a
dockerlab-vpce-s3

Task3. 보안 그룹 생성하기

- 네트워크 ACL이 서브넷 단위의 방화벽 역할을 한다면, **보안 그룹**은 인스턴스에 대한 Inbound 및 Outbound 트래픽을 제어하는 가상 방화벽 역할을 한다.
- 페이지 좌측 메뉴 중 [보안] > [보안 그룹]을 클릭한다.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the '보안' (Security) section, the '보안 그룹' (Security Groups) item is highlighted with a red box. The main content area displays a grid of security group resources. A prominent orange button at the top center says '보안 그룹 생성' (Create Security Group). Below it, a note says '참고: 인스턴스는 아시아 태평양 리전에서 시작됩니다.' (Note: Instances start in the Asia Pacific Region). The grid includes categories like VPC, NAT 게이트웨이, 서브넷, VPC 피어링 연결, 라우팅 테이블, 네트워크 ACL, 인터넷 게이트웨이, 외부 전용 인터넷 게이트웨이, DHCP 옵션 세트, 탄력적 IP, 사이트 간 VPN 연결, 엔드포인트, 고객 게이트웨이, 가상 프라이빗 게이트웨이, 규칙 그룹, 도메인 목록, and 실행 중인 인스턴스. Each category has a '모든 리전 보기' (View all regions) link.

- [보안 그룹] 페이지로 들어왔다. 새 보안 그룹을 생성하기 위해 [보안 그룹 생성] 버튼을 클릭한다.

The screenshot shows the '보안 그룹' (Security Groups) list page. At the top right, there is a large orange button labeled '보안 그룹 생성' (Create Security Group), which is also highlighted with a red box. The table below lists existing security groups with columns for Name, 보안 그룹 ID, 보안 그룹 이름 (Name), VPC ID, and 설명 (Description). The first few rows show security groups like 'henry-sg', 'default', and 'henry-devops-sg' associated with various VPCs and descriptions like 'Security Group' and 'default VPC security group'.

Name	보안 그룹 ID	보안 그룹 이름	VPC ID	설명
-	sg-064e2d93ecdb2ec81	henry-sg	vpc-06cc1e03aaa8fd14e	Security Group
-	sg-05a8b5b957bec0e2b	default	vpc-06cc1e03aaa8fd14e	default VPC security group
-	sg-0023a8a89de63e7e8	henry-devops-sg	vpc-06cc1e03aaa8fd14e	Security Group
-	sg-04c3556dec9c26d4f	default	vpc-09a2f4d65437bc5cf	default VPC security group
-	sg-0f1ec4a9f2aed96f0	default	vpc-05f27e14a8345ce8a	default VPC security group
-	sg-0c11a597747c76180	henry-demo-ec2-sg	vpc-06cc1e03aaa8fd14e	launch-wizard
-	sg-0765721f46d0d778e	henry-datalake-sg	vpc-09a2f4d65437bc5cf	Security Group

4. [보안 그룹 생성] 페이지에서 다음과 같이 설정한다.

- A. [보안 그룹 이름] : “**dockerlab-ubuntu-ec2-sg**”
- B. [설명] : “**Security group for dockerlab ubuntu ec2 instance**”
- C. [VPC] : **dockerlab-vpc**

보안 그룹 생성 정보

보안 그룹은 인바운드 및 아웃바운드 트래픽을 관리하는 인스턴스의 가상 방화벽 역할을 합니다. 새 보안 그룹 만들기

기본 세부 정보

보안 그룹 이름 정보
dockerlab-ubuntu-ec2-sg

생성 후에는 이름을 편집할 수 없습니다.

설명 정보
Security group for dockerlab ubuntu ec2 instance

VPC 정보
vpc-05f27e14a8345ce8a

X

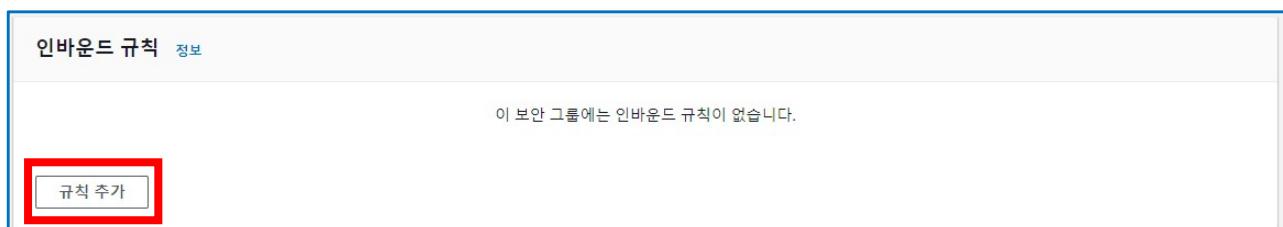


5. [인바운드 규칙] > [규칙 추가] 버튼을 클릭한다.

인바운드 규칙 정보

이 보안 그룹에는 인바운드 규칙이 없습니다.

규칙 추가



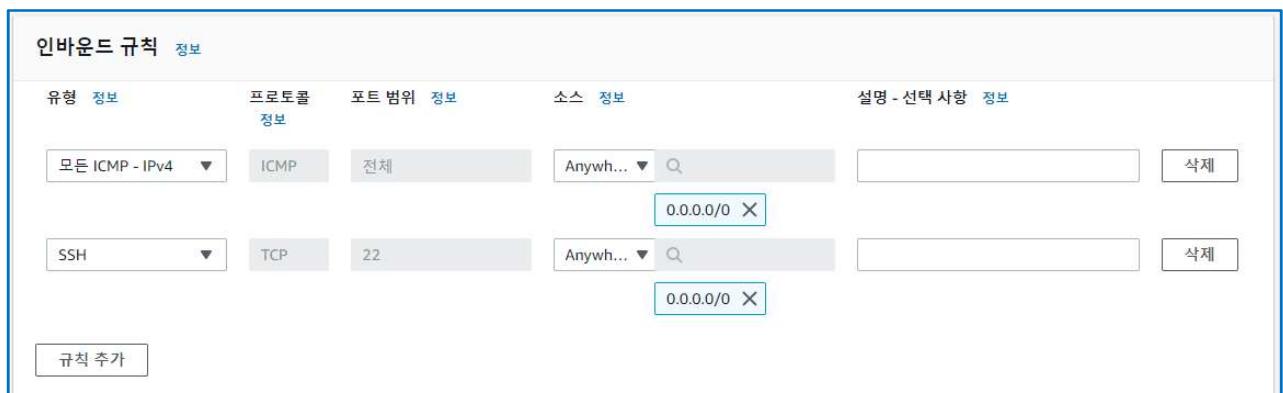
6. 다음 그림과 같이 2개의 규칙을 추가한다.

- A. [유형] : 모든 ICMP – IPv4, [프로토콜] : ICMP, [포트 범위] : 전체, [소스] : Anywhere-IPv4
- B. [유형] : SSH, [프로토콜] : TCP, [포트 범위] : 22, [소스] : Anywhere-IPv4

인바운드 규칙 정보

유형 정보	프로토콜	포트 범위 정보	소스 정보	설명 - 선택 사항 정보
모든 ICMP - IPv4 ▾	ICMP	전체	Anywh... ▾ 0.0.0.0/0 X	
SSH ▾	TCP	22	Anywh... ▾ 0.0.0.0/0 X	

규칙 추가



7. [아웃바운드 규칙]은 기본값 그대로 사용한다. 모든 설정을 마치면 페이지 하단의 [보안 그룹 생성] 버튼을 클릭한다.

The screenshot shows the 'Outbound Rules' section of the AWS Security Group configuration. It includes a search bar, a table header with columns for 'Name', 'Security Group ID', 'IP Version', 'Type', 'Protocol', and 'Port Range'. Two rules are listed:

Name	Security Group ID	IP Version	Type	Protocol	Port Range
-	sgr-0d9a05e710074c5...	IPv4	SSH	TCP	22
-	sgr-0cd8e3561275b9d...	IPv4	모든 ICMP - IPv4	ICMP	전체

At the bottom right, the 'Create Security Group' button is highlighted with a red box.

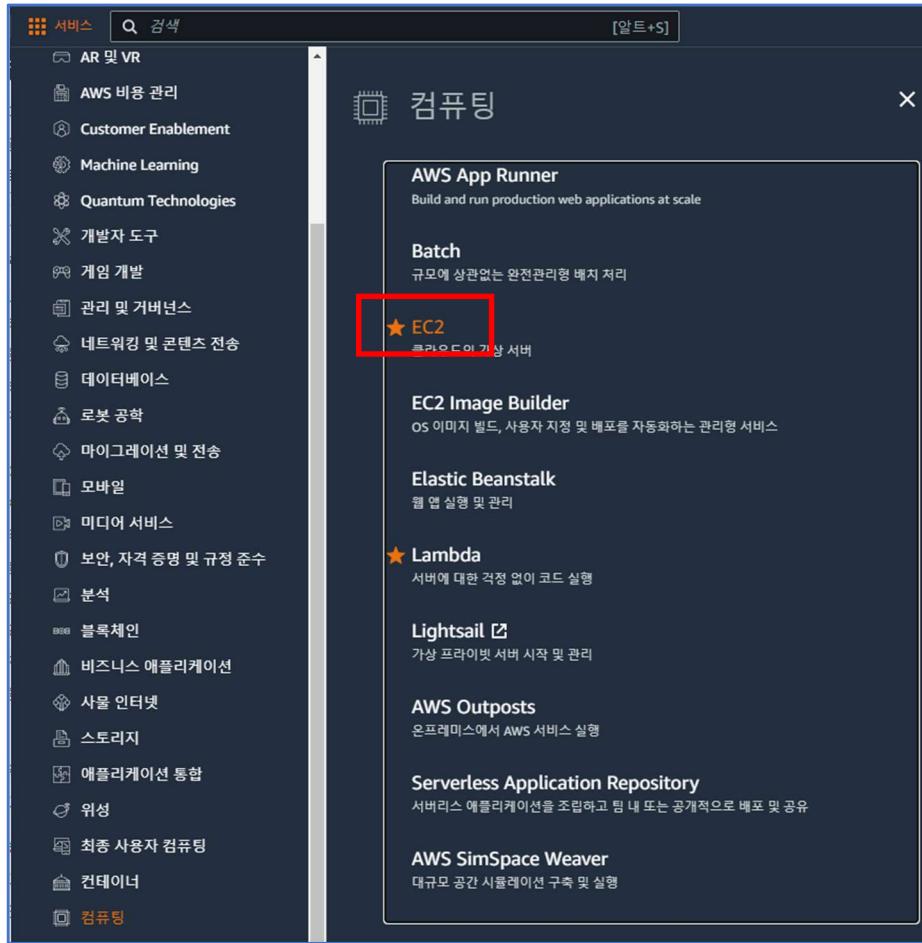
8. 방금 생성한 보안 그룹을 확인할 수 있다.

The screenshot shows the details of the security group 'sg-0568ce92d5fe96bbe - dockerlab-ubuntu-ec2-sg'. It displays information such as the security group ID, name, description, VPC ID, owner, and number of rules. Below this, there are tabs for 'Inbound Rules', 'Outbound Rules', and 'Tags'. A note about Reachability Analyzer is shown, along with a 'Run Reachability Analyzer' button. The 'Inbound Rules' table is expanded, showing two specific rules.

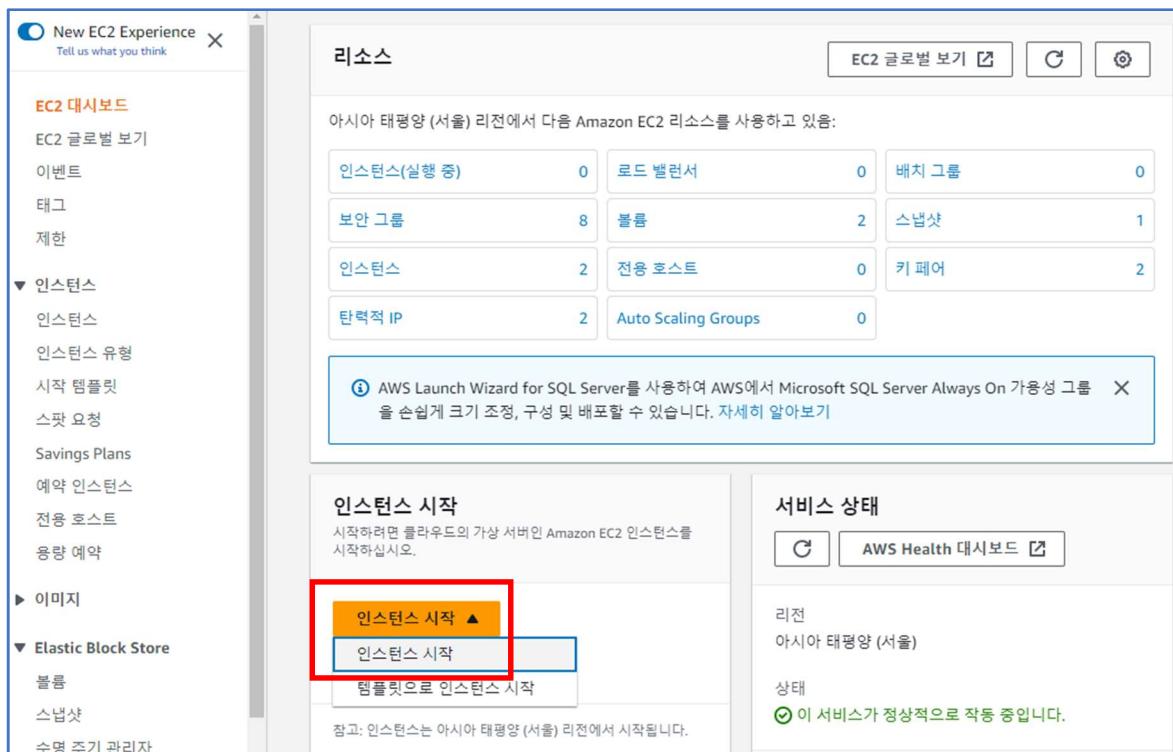
Name	Security Group ID	IP Version	Type	Protocol	Port Range
-	sgr-0d9a05e710074c5...	IPv4	SSH	TCP	22
-	sgr-0cd8e3561275b9d...	IPv4	모든 ICMP - IPv4	ICMP	전체

Task4. EC2 Instance 생성하기

- 좌측 상단의 [서비스] > [컴퓨팅] > [EC2]를 클릭하여 해당 페이지로 이동한다.



- [EC2 대시보드] 페이지에서 [인스턴스 시작] > [인스턴스 시작]을 클릭한다.



3. [인스턴스 시작] 페이지에서 [Name and tags] > [이름]은 “docker-ubuntu-ec2”로 입력한다.

인스턴스 시작 정보

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

Name and tags 정보

이름 dockerlab-ubuntu-ec2 Add additional tags

4. [애플리케이션 및 OS 이미지] 섹션에서 다음과 같이 선택한다.

- [Quick Start] : [Ubuntu Ubuntu]
- [Amazon Machine Image(AMI)] : [Ubuntu Server 22.04 LTS(HVM), SSD Volume Type]
- [아키텍처] : [64비트(x86)]

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는데 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

내 AMI Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat >

더 많은 AMI 찾아보기 AWS, Marketplace 및 커뮤니티의 AMI 포함

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type ami-04cebc8d6c4f297a3 (64비트(x86)) / ami-0084ff4520f7fd92a (64비트(Arm)) 가상화: hvm ENA 활성화됨: true 루트 디바이스 유형: ebs

설명 Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

아키텍처 AMI ID

64비트(x86) ami-04cebc8d6c4f297a3 확인된 공급 업체

5. [인스턴스 유형] : [t2.micro]

▼ 인스턴스 유형 정보

인스턴스 유형

t2.micro 프리 티어 사용 가능

All generations

인스턴스 유형 비교

패밀리: t2 1 vCPU 1 GiB 메모리 Current generation: true
온디맨드 RHEL 요금: 0.0744 USD 시간당 온디맨드 Linux 요금: 0.0144 USD 시간당
온디맨드 SUSE 요금: 0.0144 USD 시간당 온디맨드 Windows 요금: 0.019 USD 시간당

6. [키 페어(로그인)] > [새 키 페어 생성] 클릭

▼ 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

7. [키 페어 생성] 창에서 [키 페어 이름]을 “dockerlab-ubuntu-ec2-key”로 입력하고 나머지 값은 그대로 사용한다. [키 페어 생성] 버튼을 클릭한다.

키 페어 생성

키 페어를 사용하면 인스턴스에 안전하게 연결할 수 있습니다.

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

키 페어 이름

dockerlab-ubuntu-ec2-key

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(Windows 인스턴스에는 지원되지 않음)

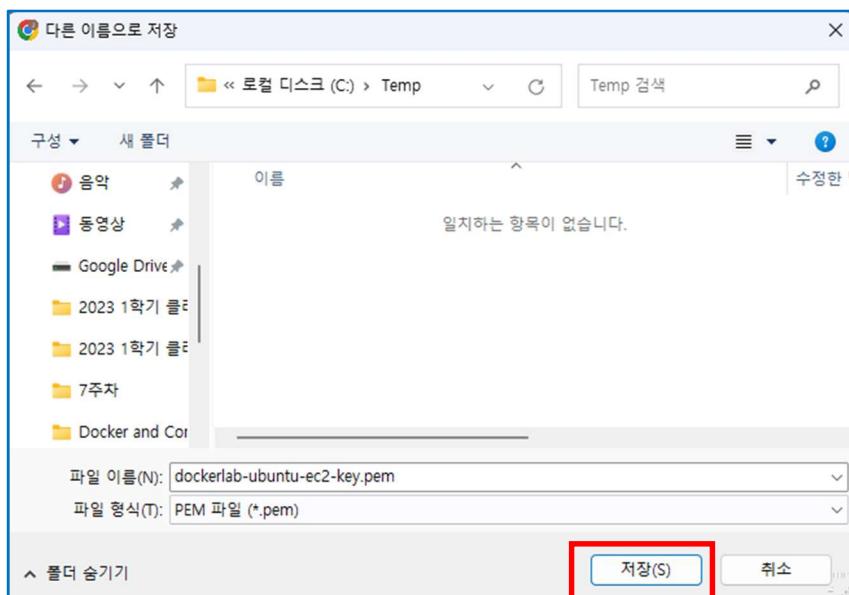
프라이빗 키 파일 형식

.pem
OpenSSH와 함께 사용

.ppk
PuTTY와 함께 사용

취소 키 페어 생성

8. 방금 생성한 키 페어 파일을 찾기 쉬운 디렉토리에 저장한다.



9. [키 페어 이름]에 방금 생성한 키 페어 이름을 확인할 수 있다.

▼ 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
dockerlab-ubuntu-ec2-key

새 키 페어 생성

10. [네트워크 설정] 섹션에서 [편집] 버튼을 클릭하여 다음과 같이 설정한다.

- [VPC] : "dockerlab-vpc"
- [서브넷] : dockerlab-subnet-public1-ap-northeast-2a
- [퍼블릭 IP 자동 할당] : [활성화]

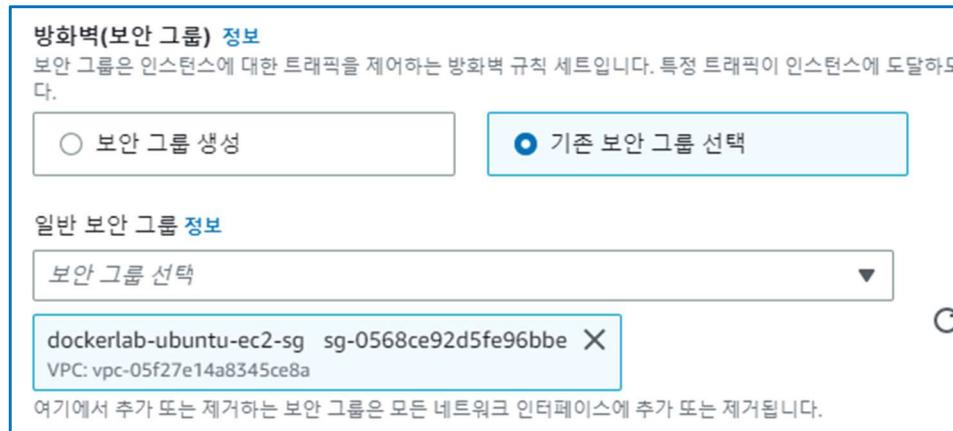
▼ 네트워크 설정 정보

VPC - 필수 정보
vpc-05f27e14a8345ce8a (dockerlab-vpc)
10.0.0.0/16

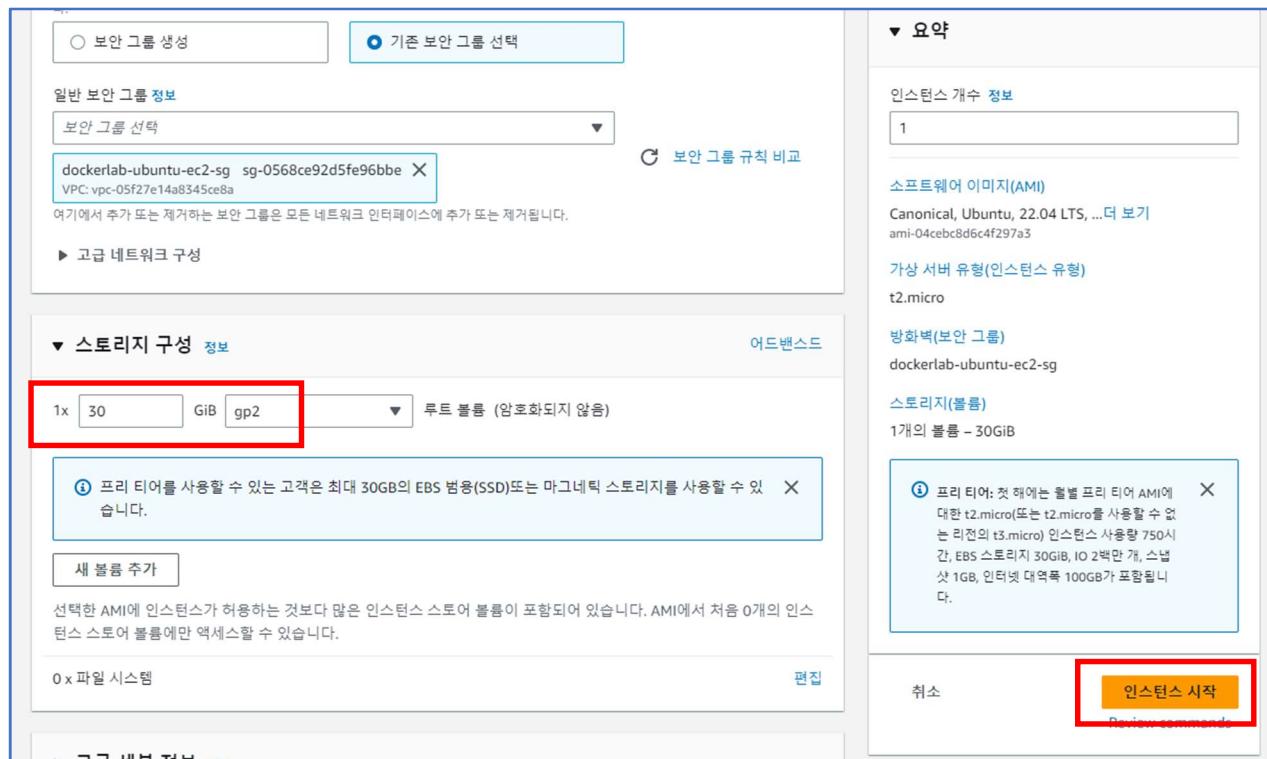
서브넷 정보
subnet-03f25fa8a7558fb8d dockerlab-subnet-public1-ap-northeast-2a
VPC: vpc-05f27e14a8345ce8a 소유자: 789534828835 사용 영역: ap-northeast-2a
IP 주소 사용 가능: 250 CIDR: 10.0.10.0/24

퍼블릭 IP 자동 할당 정보
활성화

11. [방화벽(보안 그룹)] : [기준 보안 그룹 선택], [일반 보안 그룹]은 이미 앞에서 생성했던 “**dockerlab-ubuntu-ec2-sg**”를 선택한다.



12. [스토리지 구성] 섹션에서 **30 GiB**를 지정한다. 그리고 페이지 오른쪽에 있는 [인스턴스 시작] 버튼을 클릭하여 지금까지 설정한 값을 기준으로 새 인스턴스를 생성한다.



13. 성공적으로 인스턴스가 생성되었다. [모든 인스턴스 보기] 또는 생성하는 인스턴스의 링크를 클릭한다.

The screenshot shows the AWS EC2 Instances page. At the top, there is a success message: "성공 인스턴스를 시작했습니다 (i-03a6c1a54f99b05b8)". Below this, there are three cards: "결제 및 프리 티어 사용 알림 생성", "인스턴스에 연결", and "RDS 데이터베이스 연결". At the bottom right of the page, there is a yellow button labeled "모든 인스턴스 보기" which is highlighted with a red box.

14. 인스턴스가 생성되면 시스템 상태 검사와 인스턴스 상태 검사 2가지를 수행한다. [상태 검사]가 [2/2개 검사 통과]라고 상태 검사가 모두 마칠 때까지 기다린다. 상태 검사가 모두 마치면 이제 인스턴스와 연결할 수 있다.

The screenshot shows the AWS EC2 Instances details page for instance i-03a6c1a54f99b05b8. The status bar at the top indicates "2/2개 검사 통과...". The main table shows the instance details, including Name: dockerlab-ubuntu..., Instance ID: i-03a6c1a54f99b05b8, Status: 실행 중 (Running), Instance Type: t2.micro, and State Check: 2/2개 검사 통과... 경보 없음.

15. 해당 인스턴스를 선택하고 페이지 상단의 [연결]을 클릭한다.

The screenshot shows the AWS EC2 Instances details page for instance i-03a6c1a54f99b05b8. The "연결" (Connect) button in the top right corner is highlighted with a red box. The page displays various instance details such as IP addresses, instance type, and VPC information.

16. [인스턴스에 연결] 페이지에서 [EC2 인스턴스 연결] 탭을 선택한다. 그리고 [연결] 버튼을 클릭한다.

EC2 > 인스턴스 > i-03a6c1a54f99b05b8 > 인스턴스에 연결

인스턴스에 연결 정보

다음 옵션 중 하나를 사용하여 인스턴스 i-03a6c1a54f99b05b8 (dockerlab-ubuntu-ec2)에 연결

EC2 인스턴스 연결 Session Manager SSH 클라이언트 EC2 직렬 콘솔

인스턴스 ID
i-03a6c1a54f99b05b8 (dockerlab-ubuntu-ec2)

퍼블릭 IP 주소
3.39.228.97

사용자 이름
인스턴스를 시작하는 데 사용되는 AMI에 정의된 사용자 이름을 입력합니다. 사용자 지정 사용자 이름을 정의하지 않은 경우 기본 사용자 이름인 ubuntu(들) 사용합니다.
ubuntu

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

취소 **연결**

17. 생성한 인스턴스에 잘 연결되는 것을 확인할 수 있다.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

 System information as of Wed Apr 19 01:59:08 UTC 2023

 System load: 0.080078125 Processes: 95
 Usage of /: 5.3% of 28.89GB Users logged in: 0
 Memory usage: 21% IPv4 address for eth0: 10.0.10.23
 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.

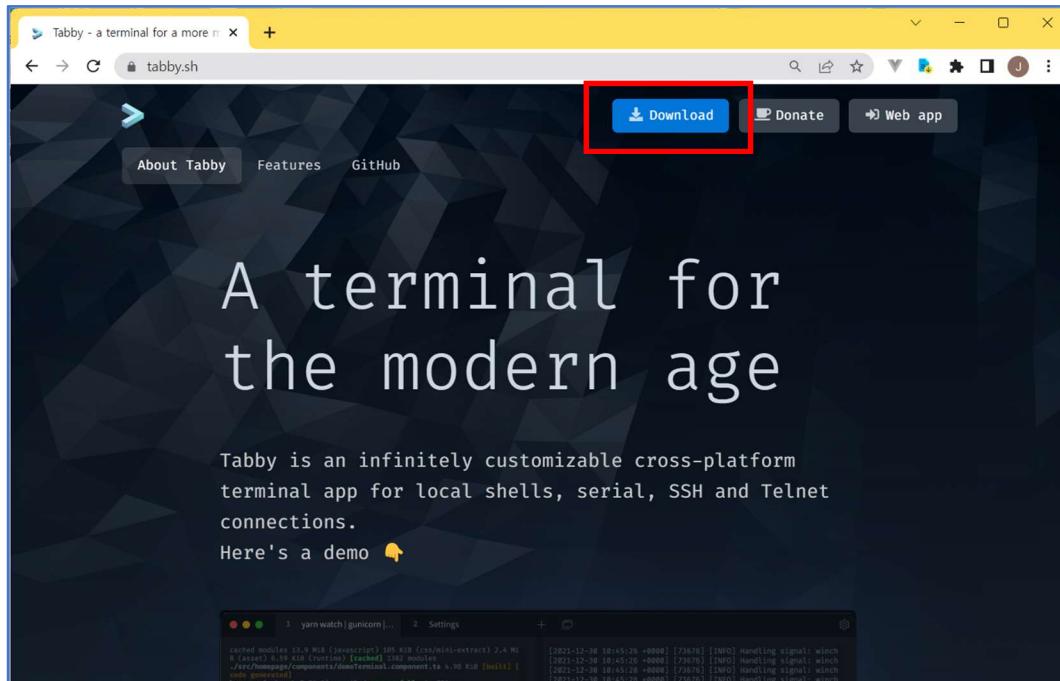
ubuntu@ip-10-0-10-23:~$
```

i-03a6c1a54f99b05b8 (dockerlab-ubuntu-ec2)
PublicIPs: 3.39.228.97 PrivateIPs: 10.0.10.23

Task5. Ubuntu Linux 인스턴스 접속하기

- Linux 인스턴스 접속을 위해서는 일반적으로 SSH 접속용 프로그램이 필요하다. 가장 일반적으로 사용하는 SSH 툴은 **Putty**이다. <https://www.putty.org/>에 접속한 후, 다운로드 받아서 사용할 수 있다. 하지만 우리가 AWS Console에 접근하기 위한 Key pair 파일이 pem 파일이기 때문에 이 파일을 이용해서 AWS Console에 접근하기 위한 좀 더 편리한 SSH Tool인 **Tabby**를 사용하기로 한다. Tabby의 홈페이지는 다음과 같다.

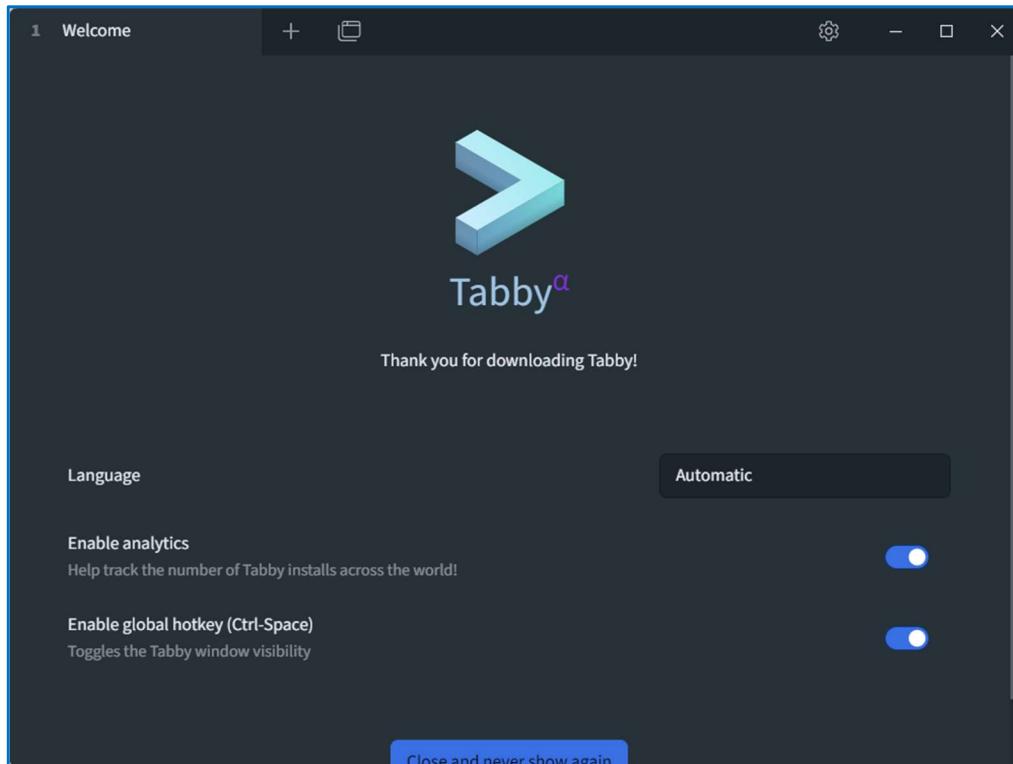
<https://tabby.sh/>



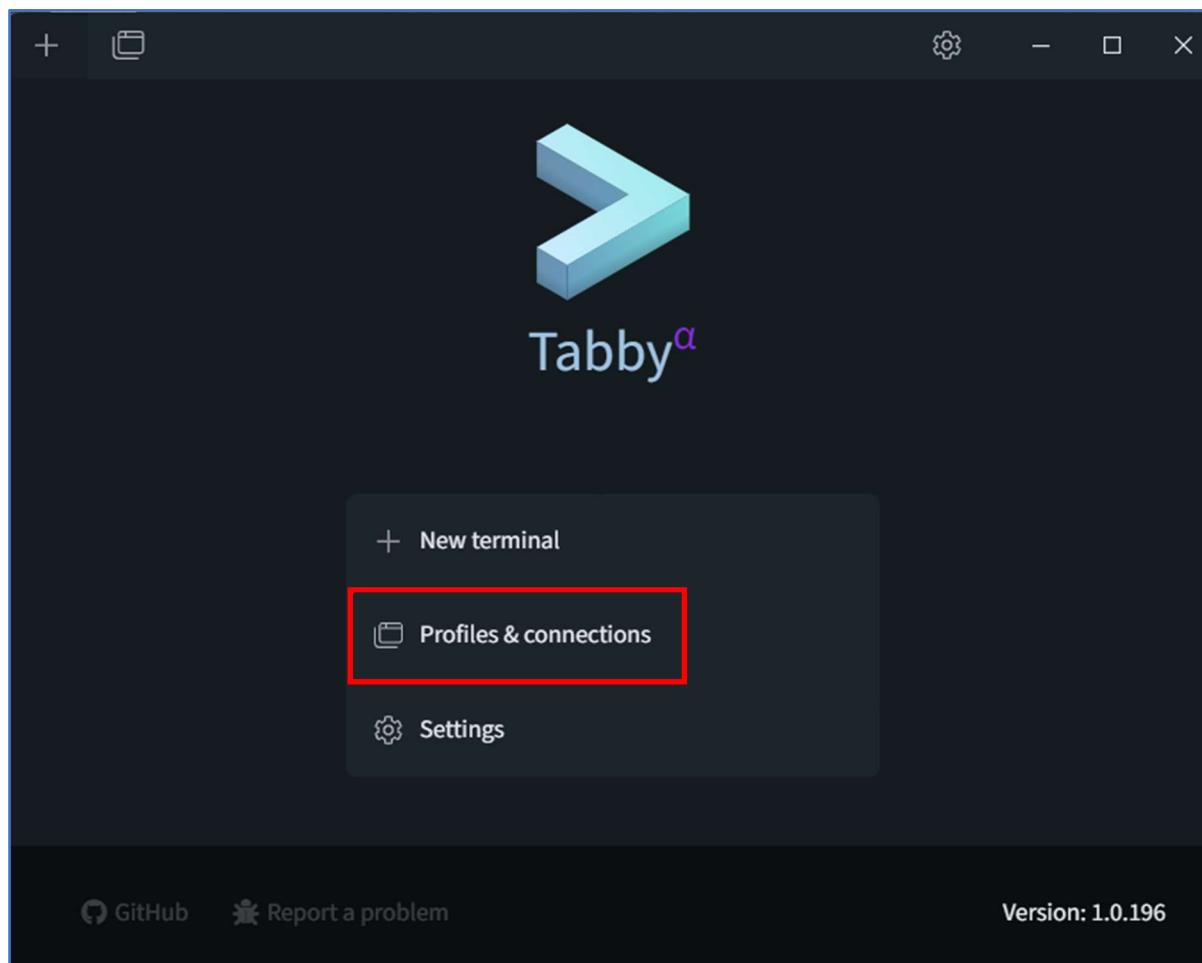
- 페이지 상단의 [Download] 버튼을 클릭하면 다음과 같이 Github으로 이동한다. 이 페이지에서 최신 버전의 Tabby를 플랫폼별로 다운로드할 수 있다. 각 플랫폼별로 다운로드하여 설치한다.

A screenshot of a GitHub repository page for 'Tabby'. The page title is 'Tabby / Releases'. On the left, there's a sidebar with options like 'Code', 'Issues', 'Pull requests', 'Releases', and 'Assets'. The 'Assets' tab is selected and shows a list of downloadable files. The list includes various binary files for different platforms: latest-arm64-mac.yml, latest-arm64.yml, latest-x64.yml, latest-x86_64-mac.yml, tabby-1.0.196-linux-x64.deb, tabby-1.0.196-linux-x64.pacman, tabby-1.0.196-linux-x64.rpm, tabby-1.0.196-linux-x64.tar.gz, tabby-1.0.196-macos-arm64.pkg, tabby-1.0.196-macos-arm64.zip, tabby-1.0.196-macos-x86_64.pkg, tabby-1.0.196-macos-x86_64.zip, tabby-1.0.196-portable-arm64.zip, tabby-1.0.196-portable-x64.zip, tabby-1.0.196-setup-arm64.exe, tabby-1.0.196-setup-arm64.exe.blockmap, tabby-1.0.196-setup-x64.exe, and tabby-1.0.196-setup-x64.exe.blockmap. Each item shows its file size and the date it was last updated. At the bottom of the assets list, there are two additional links: 'Source code (zip)' and 'Source code (tar.gz)'. Below the assets list, there are standard GitHub reaction icons (smiley face, thumbs up, heart, etc.) and a note that says '59 people reacted'.

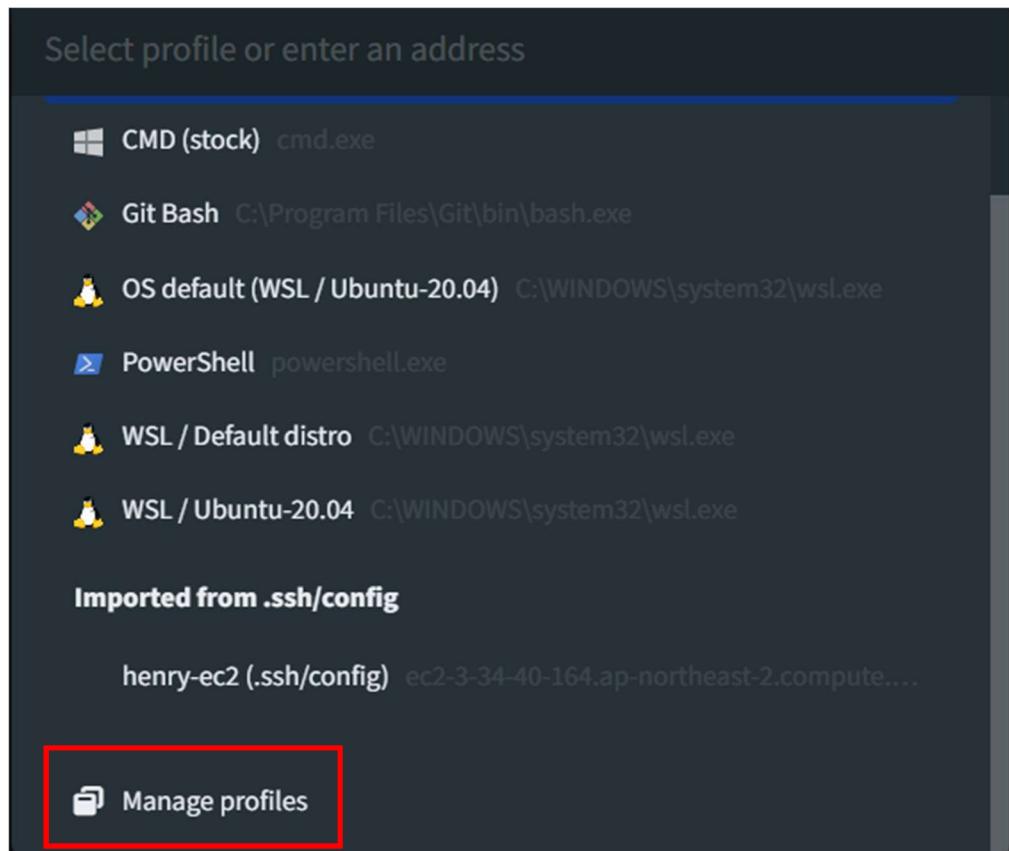
3. Windows 10 혹은 11에서는 [tabby-x.x.xxx-setup-x64.exe]를 다운로드하여 설치하면 아래의 그림과 같다.



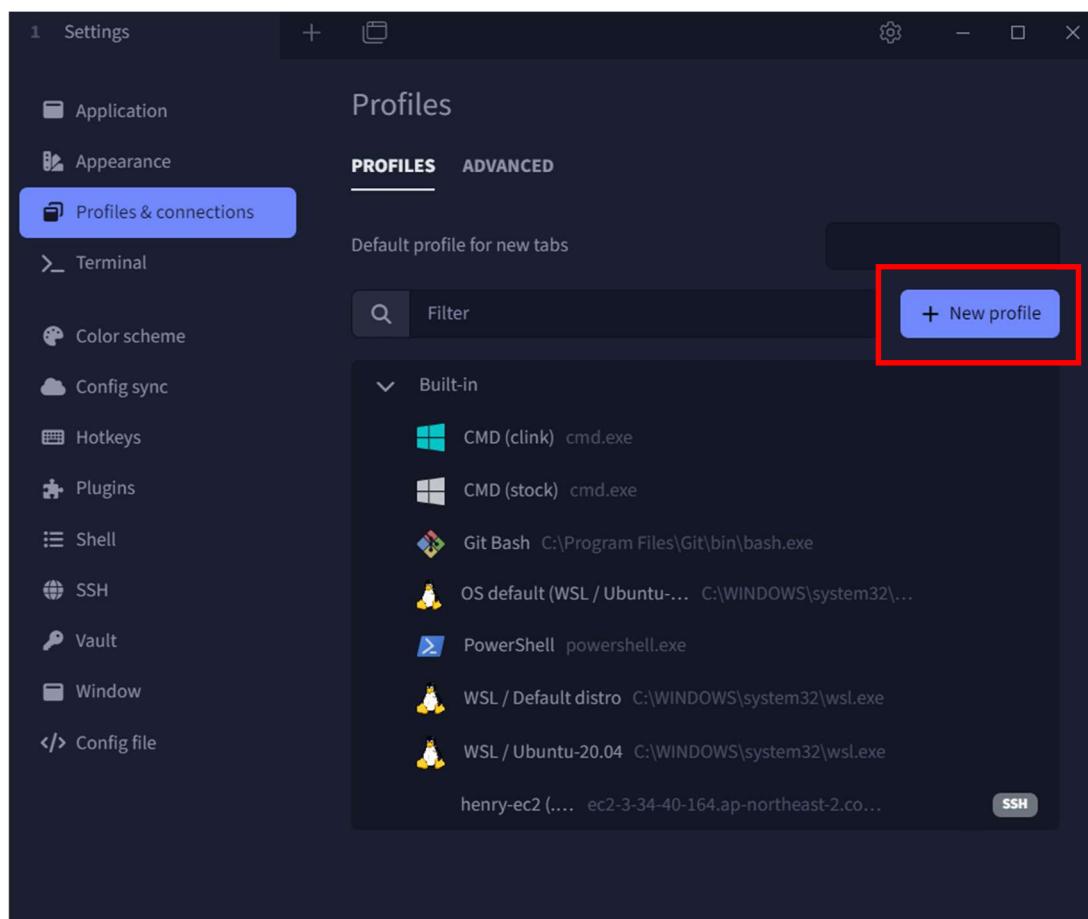
4. “Welcome” 창을 닫고 [Profiles & connections]를 클릭한다.



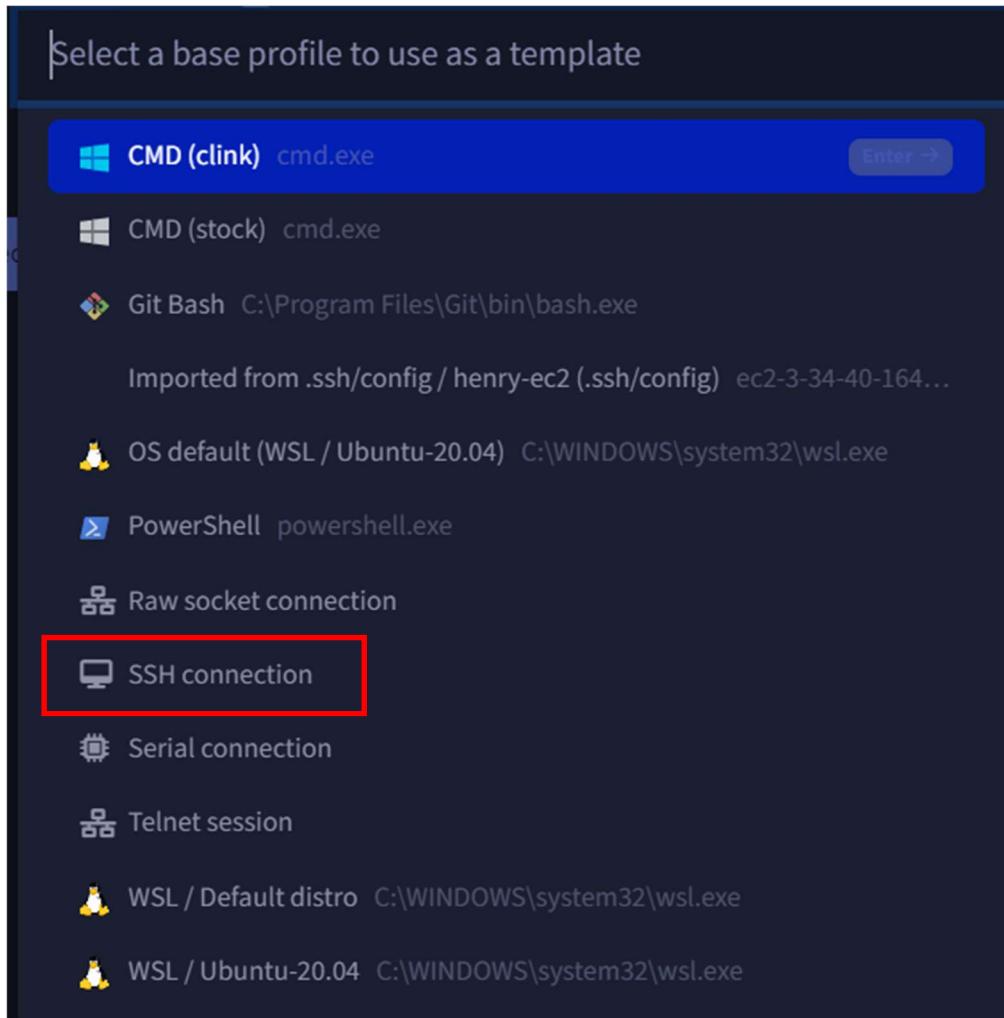
5. 처음 등록할 때는 설정을 해야 한다. [Manage profiles]를 클릭한다.



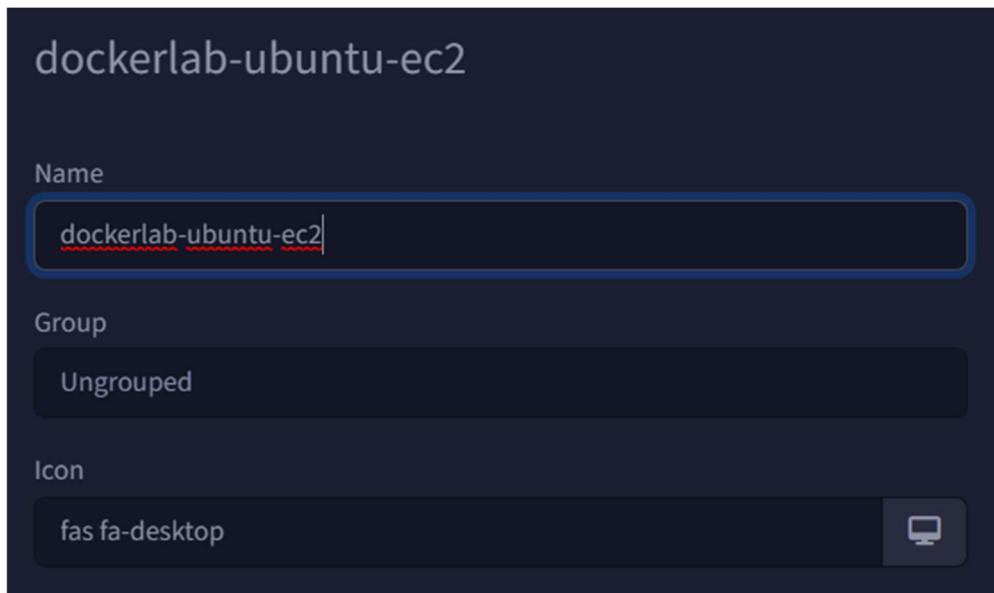
6. [Profiles] > [PROFILES]에서 [New profile]를 클릭한다.



7. [Select a base profile to use as a template] 창에서 [SSH connection]를 선택한다.

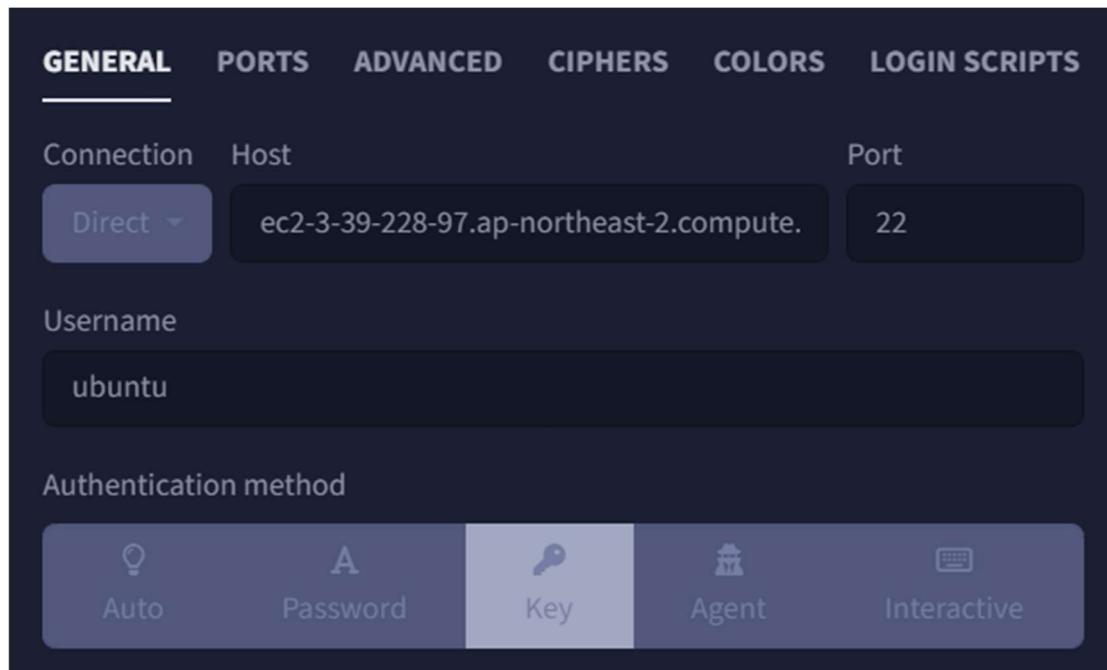


8. [Name]을 “dockerlab-ubuntu-ec2”라고 입력한다.

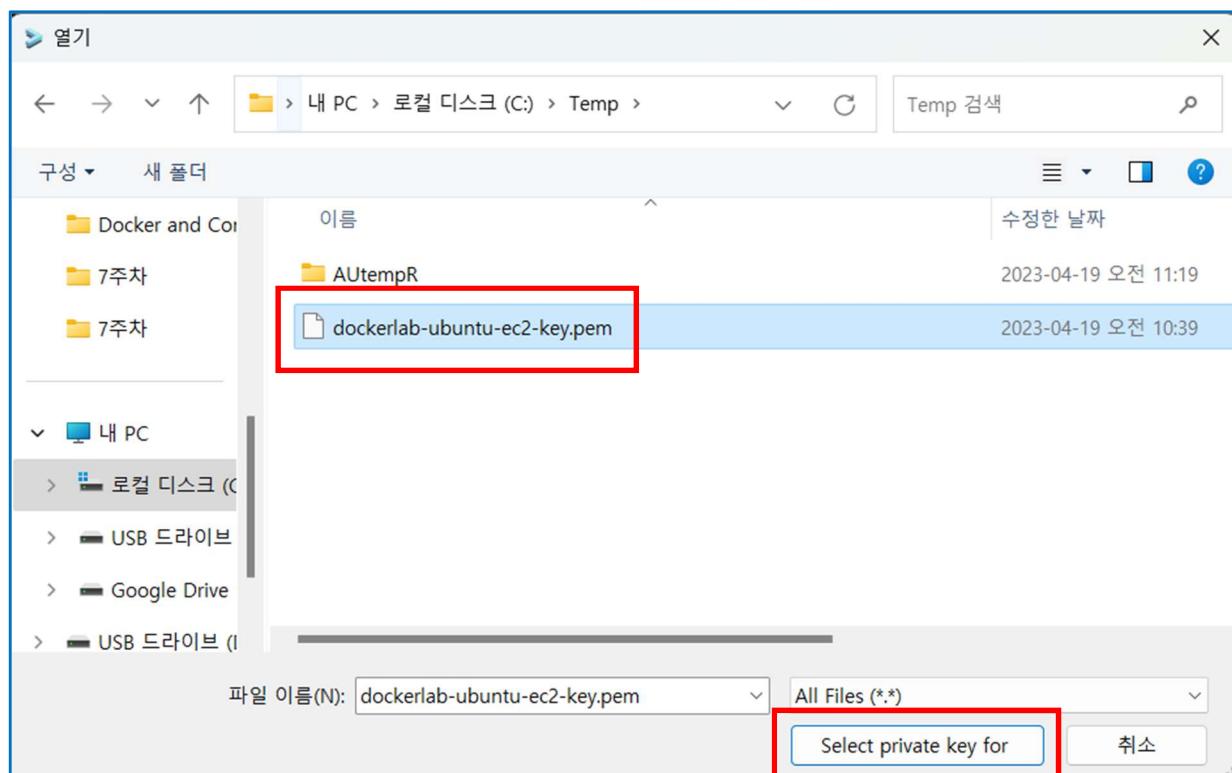


9. 다음과 같이 설정한다.

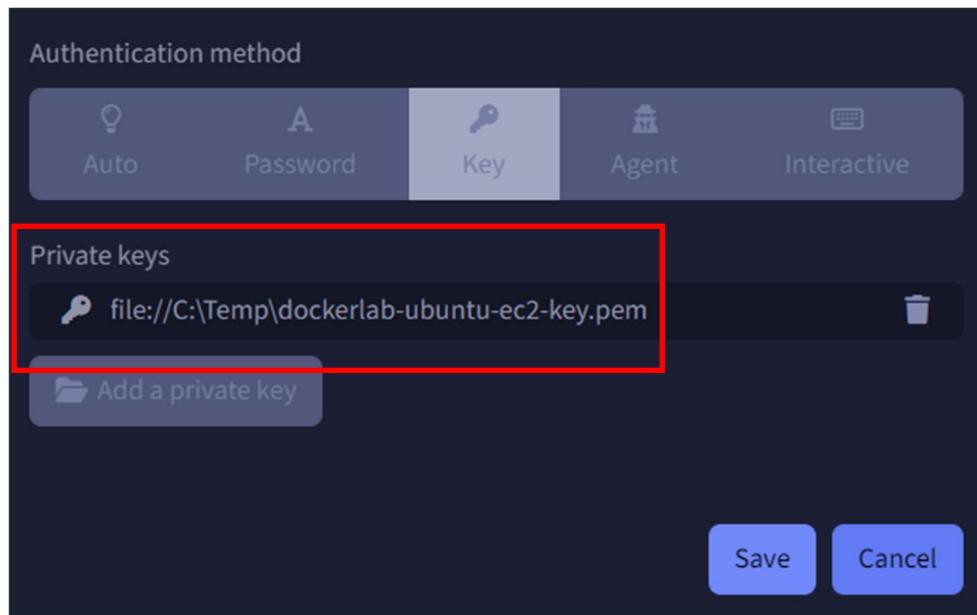
- [Host] : EC2의 [퍼블릭 IPv4 DNS] 값
- [Port] : 22
- [Username] : "ubuntu"
- [Authentication method] : [Key]



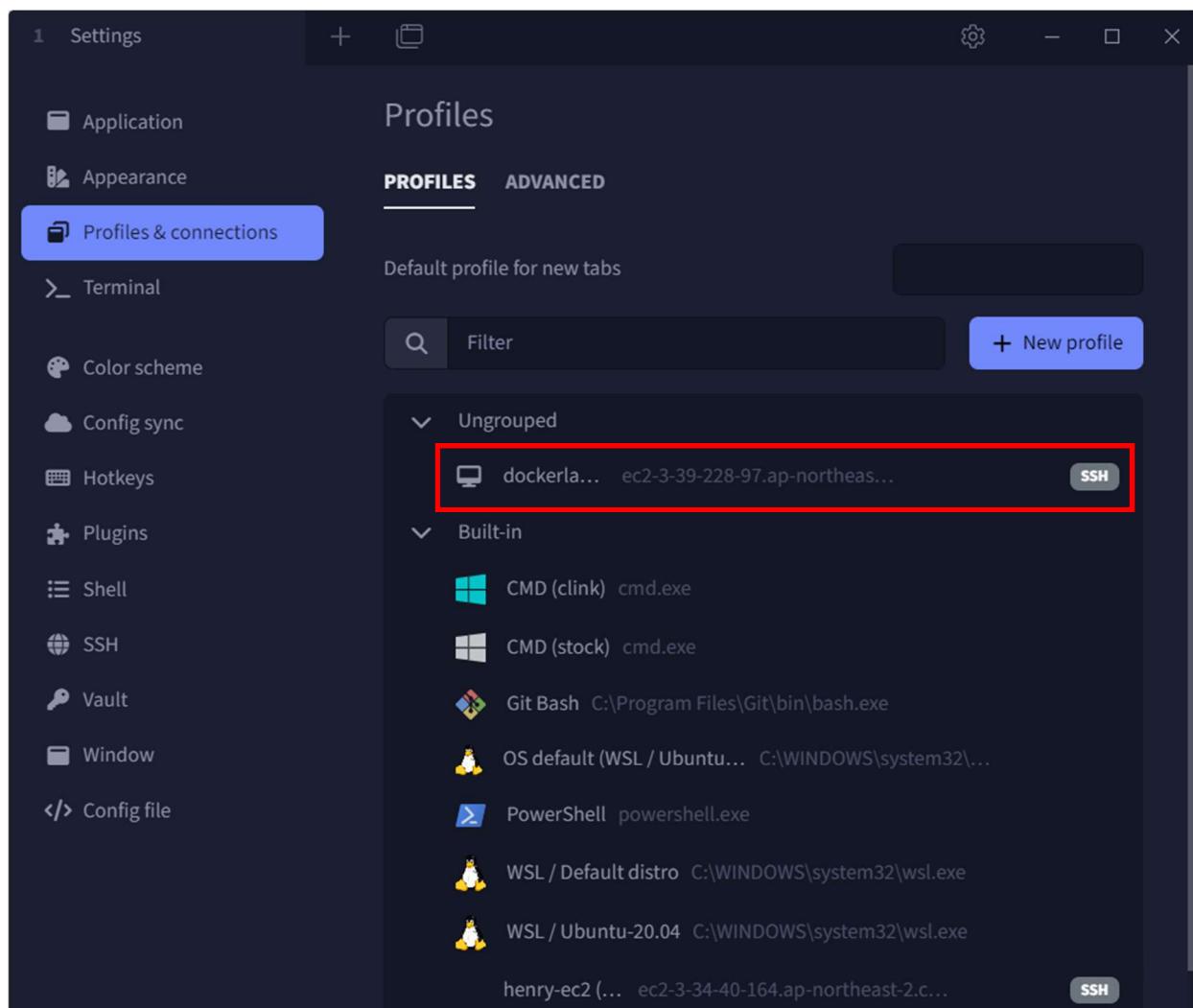
10. [Authentication method]를 [Key]로 선택하면 [Private keys] > [Add a private key] 버튼을 클릭한다. 앞에서 이미 다운로드 받아 저장한 pem 파일을 선택하고 [Select private key for] 버튼을 클릭한다.



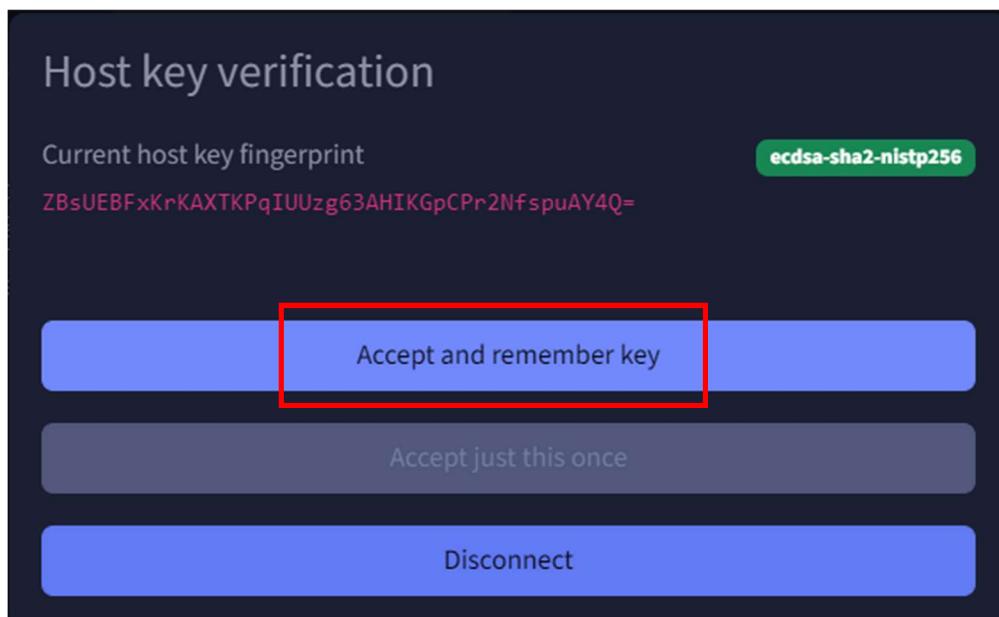
11. [Save] 버튼을 클릭한다.



12. 다시 [Profiles] 창으로 돌아왔다. 방금 설정한 Profile인 "dockerlab-ubuntu-ec2"의 ► 버튼을 클릭한다.



13. [Host key verification] 창에서 [Accept and remember key] 버튼을 클릭한다.



14. 성공적으로 EC2 인스턴스에 연결되었다.

```
1 dockerlab-ubuntu-ec2 + 🗃
● ubuntu@ec2-3-39-228-97.ap-northeast-2.compute.amazonaws.com:22
    Q Reconnect SFTP Ports Unpin

System information as of Thu Apr 20 01:31:58 UTC 2023

System load: 0.0          Processes:      96
Usage of /:  5.5% of 28.89GB  Users logged in:   0
Memory usage: 22%          IPv4 address for eth0: 10.0.10.23
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/aws/pro

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

Last login: Wed Apr 19 01:59:08 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-10-0-10-23:~$
```