

처음 사용하는 분을 위한 안내

cloudflared 설치부터 터널 연결까지

프로그램 설치 → Cloudflare 웹 로그인 → 터널 생성·토큰 명령 복사 → 서버에서 실행 → 연결·웹 접속 확인까지 따라합니다. 커넥터를 실행할 서버의 운영체제를 고르세요.

Windows · macOS · Linux · 전체 34단계

시작 전에 준비하세요 설치할 서버 또는 VM과 설치 권한 터널을 관리할 수 있는 Cloudflare 계정 서버의 인터넷 연결

웹 접속 시험: 본인 도메인과 실행 중인 웹 앱

Windows 설치 안내 11단계

Windows · 1 / 11단계

공식 Windows 설치 파일을 받습니다

일반적인 64비트 Windows PC·서버라면 아래 **.msi 설치 파일**을 받습니다. 사내에서 지정한 버전이 있으면 그 버전을 사용하세요.

32비트 등 다른 환경은 [공식 다운로드의 Windows 표](#)에서 맞는 파일을 고릅니다.

[Windows 64비트 MSI 받기 ↗](#)

Windows · 2 / 11단계

받은 .msi 파일을 실행합니다

1. 다운로드한 cloudflared 설치 파일을 엽니다.
2. Windows가 설치 허용이나 관리자 권한을 물으면 회사에서 허용한 계정으로 진행합니다.
3. 설치가 끝나면 열려 있던 PowerShell을 닫고 새로 엽니다.

cloudflared는 명령으로 실행하는 프로그램입니다. 설치 후 로그인 창이 자동으로 뜨지 않아도 다음 단계에서 확인할 수 있습니다.

PowerShell에서 버전이 나오는지 확인합니다

시작 메뉴에서 **PowerShell**을 열고 아래 명령을 실행합니다. `cloudflared version` 뒤에 버전 번호가 나오면 설치를 확인한 것입니다.

명령을 찾을 수 없다고 나오면 새 PowerShell에서 다시 실행하세요. 계속 같으면 `Get-Command cloudflared` 로 실행 파일 경로를 확인하고 설치 파일이 정상적으로 완료됐는지 확인합니다. Windows 자동 업데이트 여부도 [공식 안내](#)를 확인하세요.

```
cloudflared --version
```

Cloudflare 웹에 로그인하고 터널 목록을 엽니다

- 아래 버튼을 열어 Cloudflare 계정에 로그인합니다. 사용하는 SSO나 2단계 인증이 있으면 완료합니다.
- 여러 계정이 보이면 **이 서버의 터널을 관리할 계정을 선택합니다.**
- Networking > Tunnels**로 이동합니다. 계정에 따라 기존 Cloudflare One의 **Networks > Connectors**에서 같은 터널 목록으로 들어갈 수도 있습니다.

이 단계는 Cloudflare **관리 웹사이트 로그인**입니다. cloudflared 프로그램에는 WARP처럼 회사 계정을 넣는 로그인 창이 없습니다.

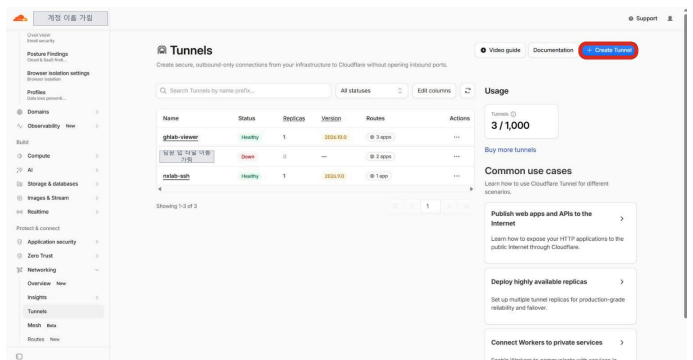
[Tunnels 열기 · 내 계정 선택](#)

Create Tunnel을 누르고 터널 이름을 정합니다

- 목록 오른쪽 위 **Create Tunnel**을 누릅니다. 유형 선택이 나오면 **cloudflared**를 선택합니다.
- 터널 이름에는 연결할 서버의 용도를 적습니다. 예: `office-web-connector`.
- Create Tunnel** 또는 **Save tunnel**을 눌러 생성합니다. 다음으로 설치 환경과 실행 명령 화면이 열립니다.

이미 만들어 둔 터널에 서버를 연결하는 경우에는 새로 만들지 말고, 해당 터널 → **Add a replica**를 엽니다. 이름만 생긴 단계에서는 서버 연결이 끝난 것이 아닙니다.

[Tunnels 목록 열기 · 내 계정](#)



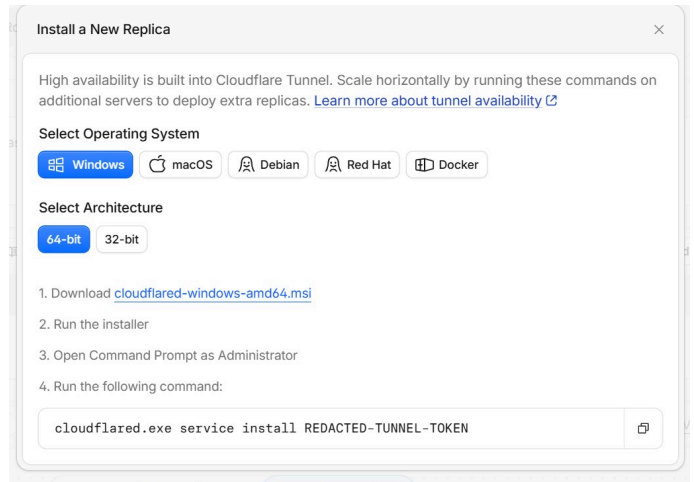
실제 화면 예시 · 누르면 크게 볼 수 있습니다

서버 OS를 고르고 토큰이 포함된 명령을 복사합니다

1. 새 터널은 생성 직후의 설치 화면에서, 기존 터널은 **터널 이름** → **Add a replica**에서 명령을 확인합니다.
2. **Setup Environment** 또는 **Select Operating System** · **Select Architecture**에서 서버 OS와 CPU 종류를 고릅니다. 실제 대시보드에서도 본인 환경을 맞춰 선택합니다.
3. **Install and Run** 또는 **Run the following command**를 찾습니다. 이미 설치했다면 **cloudflared**를 서비스로 등록·실행하는 명령을 사용합니다.
4. 명령 오른쪽의 **복사 아이콘**으로 토큰이 포함된 **명령 전체**를 복사합니다. 명령 뒤의 긴 문자열이 이 터널의 연결 토큰입니다. 토큰이 화면에서 가려져 있어도 별표를 직접 입력하지 않습니다.

API Tokens 메뉴에서 API 키를 발급하는 단계가 아닙니다. 이 토큰으로 어느 터널에 붙을지가 정해집니다. 토큰과 토큰이 든 명령은 문서·채팅·화면 캡처에 공유하지 않습니다.

[Tunnels → 내 터널 열기 ↗](#)



실제 Add a replica 화면 · Windows 예시 · 토큰은 가렸습니다. 눌러서 확대하세요.

복사한 실제 명령을 서버에서 실행합니다

1. 시작 메뉴에서 **PowerShell**을 찾고 **관리자 권한으로 실행**합니다.
2. 대시보드에서 복사한 명령을 붙여 실행합니다. 프로그램 경로가 포함돼 있으면 그 경로까지 그대로 사용합니다.
3. 서비스 설치 성공 메시지를 확인합니다. 상태 확인은 `Get-Service cloudflared` 로 하며 **Running**인지 봅니다.

이 서비스가 서버를 지정한 터널에 연결합니다. 설치 파일만 실행한 것과는 다른 단계입니다.

아래는 토큰을 가린 형식 예시입니다. 여기에 적힌 가림 문자열을 실행하지 말고, 본인 웹 대시보드에서 복사한 실제 명령을 사용합니다.

형식 예시 · 실제 토큰은 가림

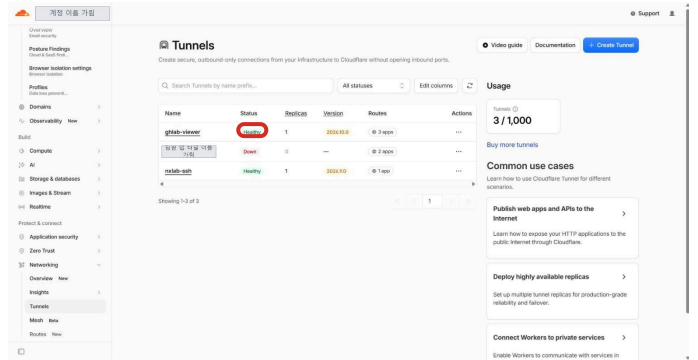
```
cloudflared.exe service install REDACTED-TUNNEL-TOKEN
```

```
Get-Service cloudflared
```

웹으로 돌아가 서버의 연결 상태를 확인합니다

1. 명령 실행 뒤 웹 대시보드의 설치 화면으로 돌아옵니다.
2. 커넥터 연결이 잡히면 **Continue**를 누릅니다. 이전 화면에서는 커넥터 목록에 연결된 서버가 표시될 수 있습니다.
3. **Networking > Tunnels** 목록에서 방금 선택한 터널이 **Healthy**인지 확인합니다.

터널 이름만 있거나 상태가 **Inactive·Down**이면 연결이 끝나지 않았습니다. **Healthy**가 된 다음 실제 앱이 열리는지도 뒤 단계에서 따로 확인합니다.



실제 화면 예시 · 누르면 크게 볼 수 있습니다

Tunnels 연결 상태 확인 ↗

커넥터 서버에서 앱 주소가 열리는지 확인합니다

이후 절차는 **HTTP 웹 앱을 본인 도메인으로 여는 예시**입니다. 연결할 앱이 이미 실행 중이어야 합니다.

1. cloudflared를 설치한 서버에서 실제 앱 주소를 엽니다. 같은 서버의 8080 포트라면 예를 들어 `http://localhost:8080` 입니다.
2. 다른 서버의 앱이면 그 서버의 실제 내부 주소·포트를 사용합니다.
3. 응답이 없으면 터널 설정 전에 앱 실행 상태와 서버 사이 연결부터 확인합니다.

뒤에서 넣는 **localhost**는 **cloudflared가 실행되는 서버 자신**을 뜻합니다. 지금 브라우저를 보는 개인 PC를 뜻하지 않습니다. 사설 IP/CIDR을 연결할 목적이라면 공개 웹 경로 대신 사설 네트워크 연결 안내를 이어서 봅니다.

Routes에서 웹 호스트와 앱 주소를 연결합니다

1. 터널을 열고 **Routes > Add route > Published application**을 고릅니다. 기존 화면의 이름은 **Public hostnames** 또는 **Published application routes**일 수 있습니다.
2. **Subdomain**에 사용할 이름을 넣고 **Domain**에서 본인 도메인을 선택합니다. 예시: `app` + `example.com` → `app.example.com`.
3. 앱 전체를 연결한다면 **Path**는 비웁니다. **Service URL**에는 앞 단계에서 확인한 실제 앱 주소를 넣습니다. 예시: `http://localhost:8080`.
4. 실제 값과 보호 정책을 확인하고 **Add route** 또는 **Save**로 저장합니다. 이때 DNS도 설정됩니다.

그림은 예시값을 채운 양식입니다. 본인 도메인과 실제 앱 주소로 바꿉니다. Type·URL이 나뉜 화면은 **Type: HTTP, URL: localhost:8080**입니다.

공개 웹 경로는 인터넷에서 접근할 수 있습니다. 회사 로그인 뒤에만 열려야 하는 업무 앱은 먼저 [Access 애플리케이션과 허용 정책](#)을 준비하세요. 터널 토큰은 직원의 웹 로그인 정책을 대신하지 않습니다.

[Tunnels](#) → [내 터널](#) → [Routes](#) ↗

The screenshot shows a 'Add published application' dialog box. It contains the following fields and options:

- Hostname:** Multi-level subdomains require Advanced Certificate Manager. [Learn more](#)
- Subdomain (optional):** `app` (with a note: e.g., www, blog, api)
- Domain:** `example.com` (with a note: Select a zone, or enter a [custom hostname](#) validated with Cloudflare for SaaS)
- Full hostname:** `app.example.com`
- Path (optional):** `^/blog` (with a note: Optional regex, e.g., `^/api` or `\.(jpg|png)$`. [Learn more](#))
- Service URL:** `http://localhost:8080` (with a note: e.g., `https://localhost:8080`, `tcp://localhost:3306`)
- Additional application settings:** (collapsed)
- Buttons:** Cancel, Add route

실제 추가 양식 · 도메인은 `example.com`으로 가렸습니다. 예시값만 입력했고 저장하지 않았습니다.

새 브라우저에서 웹 주소까지 열어 확인합니다

1. 설정한 실제 주소를 새 브라우저에서 엽니다. 위 예시라면 `https://app.example.com` 형식입니다.
2. Access를 적용했다면 **허용된 회사 계정으로 로그인**한 뒤 앱이 열리는지 봅니다. 허용되지 않은 계정은 차단되는지도 확인합니다.
3. **대시보드 Healthy**와 **실제 앱 응답** 두 가지가 맞으면 연결 확인이 끝났습니다.

Healthy인데 **502가 나오면** 앞 단계의 Service URL·포트·HTTP/HTTPS와 앱 응답을 다시 확인합니다. **Inactive·Down**이면 서버에서 실행한 연결 명령과 서비스 상태를 먼저 확인합니다.

macOS 설치 안내 11단계

macOS · 1 / 11단계

터미널에서 Homebrew를 확인합니다

 + **Space** 를 누르고 **터미널**을 검색해 엽니다. 아래 명령으로 Homebrew가 있는지 확인하세요.

명령을 찾을 수 없다면 [Homebrew 공식 안내](#)에 따라 준비하거나, [공식 다운로드의 macOS 항목](#)에서 CPU에 맞는 파일을 선택합니다.

```
brew --version
```

macOS · 2 / 11단계

Homebrew로 cloudflared를 설치합니다

Homebrew가 준비된 터미널에서 아래 명령을 실행합니다. 패키지를 받고 설치하는 동안 기다립니다.

Homebrew 설치 명령 앞에 **sudo** 를 붙이지 않습니다.

```
brew install cloudflared
```

macOS · 3 / 11단계

버전 번호와 실행 파일 경로를 확인합니다

cloudflared version 뒤에 버전 번호가 나오면 실행 파일을 사용할 수 있습니다. **command -v** 는 실제 실행되는 파일의 경로를 보여줍니다.

설치 뒤에도 명령을 못 찾으면 터미널을 새로 열고, Homebrew가 안내한 PATH 설정을 확인하세요.

```
cloudflared --version  
command -v cloudflared
```

Cloudflare 웹에 로그인하고 터널 목록을 엽니다

1. 아래 버튼을 열어 Cloudflare 계정에 로그인합니다. 사용하는 SSO나 2단계 인증이 있으면 완료합니다.
2. 여러 계정이 보이면 **이 서버의 터널을 관리할 계정**을 선택합니다.
3. **Networking > Tunnels**로 이동합니다. 계정에 따라 기존 Cloudflare One의 **Networks > Connectors**에서 같은 터널 목록으로 들어갈 수도 있습니다.

이 단계는 Cloudflare **관리 웹사이트 로그인**입니다. cloudflared 프로그램에는 WARP처럼 회사 계정을 넣는 로그인 창이 없습니다.

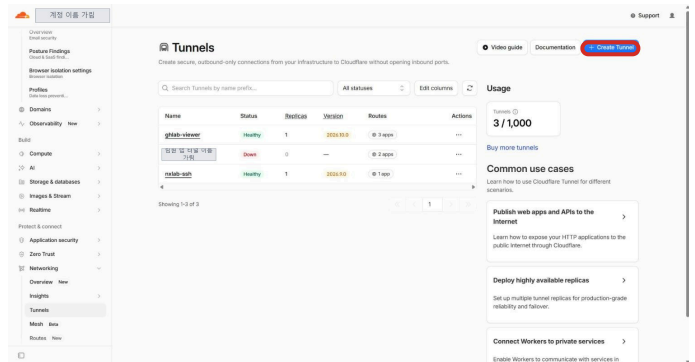
[Tunnels 열기 · 내 계정 선택](#)

Create Tunnel을 누르고 터널 이름을 정합니다

1. 목록 오른쪽 위 **Create Tunnel**을 누릅니다. 유형 선택이 나오면 **cloudflared**를 선택합니다.
2. 터널 이름에는 연결할 서버의 용도를 적습니다. 예: `office-web-connector`.
3. **Create Tunnel** 또는 **Save tunnel**을 눌러 생성합니다. 다음으로 설치 환경과 실행 명령 화면이 열립니다.

이미 만들어 둔 터널에 서버를 연결하는 경우에는 새로 만들지 말고, **해당 터널 → Add a replica**를 엽니다. 이름만 생긴 단계에서는 서버 연결이 끝난 것이 아닙니다.

[Tunnels 목록 열기 · 내 계정](#)



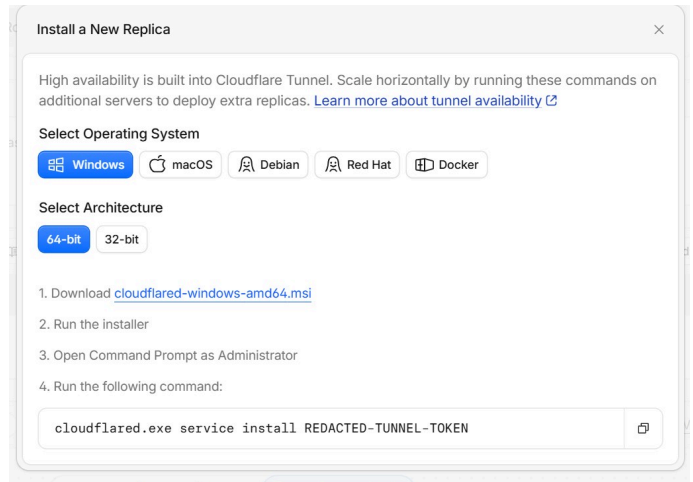
실제 화면 예시 · 누르면 크게 볼 수 있습니다

서버 OS를 고르고 토큰이 포함된 명령을 복사합니다

1. 새 터널은 생성 직후의 설치 화면에서, 기존 터널은 **터널 이름** → **Add a replica**에서 명령을 확인합니다.
2. **Setup Environment** 또는 **Select Operating System** · **Select Architecture**에서 서버 OS와 CPU 종류를 고릅니다. 실제 대시보드에서도 본인 환경을 맞춰 선택합니다.
3. **Install and Run** 또는 **Run the following command**를 찾습니다. 이미 설치했다면 **cloudflared**를 서비스로 등록·실행하는 명령을 사용합니다.
4. 명령 오른쪽의 **복사 아이콘**으로 토큰이 포함된 **명령 전체**를 복사합니다. 명령 뒤의 긴 문자열이 이 터널의 연결 토큰입니다. 토큰이 화면에서 가려져 있어도 별표를 직접 입력하지 않습니다.

API Tokens 메뉴에서 API 키를 발급하는 단계가 아닙니다. 이 토큰으로 어느 터널에 붙을지가 정해집니다. 토큰과 토큰이 든 명령은 문서·채팅·화면 캡처에 공유하지 않습니다.

[Tunnels → 내 터널 열기 ↗](#)



실제 Add a replica 화면 · Windows 예시 · 토큰은 가렸습니다. 눌러서 확대하세요.

복사한 실제 명령을 서버에서 실행합니다

1. 커넥터를 설치한 Mac에서 **터미널**을 엽니다.
2. 대시보드에서 복사한 **sudo cloudflared service install** 명령을 토큰까지 통째로 실행합니다.
3. 암호를 물으면 이 Mac의 관리자 암호를 입력합니다. 입력 중 글자가 표시되지 않아도 정상입니다.
4. 설치 성공 뒤 다음 단계에서 대시보드의 연결 상태를 확인합니다.

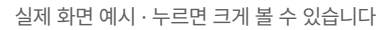
서비스는 **launchctl**로 관리됩니다. 일반 사용자의 WARP 로그인 창을 열어 인증하는 절차가 아닙니다.

아래는 토큰을 가린 형식 예시입니다. 여기에 적힌 가림 문자열을 실행하지 말고, 본인 웹 대시보드에서 복사한 실제 명령을 사용합니다.

형식 예시 · 실제 토큰은 가림

```
sudo cloudflared service install REDACTED-TUNNEL-TOKEN
```

- ## Tunnels 연결 상태 확인 ↗



커넥터 서버에서 앱 주소가 열리는지 확인합니다

뒤에서 넣는 **localhost**는 **cloudflared**가 실행되는 서버 자신을 뜻합니다. 지금 브라우저를 보는 개인 PC를 뜻하지 않습니다. 사실 IP/CIDR을 연결할 목적이라면 공개 웹 경로 대신 **사설 네트워크 연결 안내**를 이어서 봅니다.

Routes에서 웹 호스트와 앱 주소를 연결합니다

- 터널을 열고 **Routes > Add route > Published application**을 고릅니다. 기존 화면의 이름은 **Public hostnames** 또는 **Published application routes**일 수 있습니다.
- Subdomain**에 사용할 이름을 넣고 **Domain**에서 본인 도메인을 선택합니다. 예시: `app` + `example.com` → `app.example.com`.
- 앱 전체를 연결한다면 **Path**는 비웁니다. **Service URL**에는 앞 단계에서 확인한 실제 앱 주소를 넣습니다. 예시: `http://localhost:8080`.
- 실제 값과 보호 정책을 확인하고 **Add route** 또는 **Save**로 저장합니다. 이때 DNS도 설정됩니다.

그림은 예시값을 채운 양식입니다. 본인 도메인과 실제 앱 주소로 바꿉니다. Type·URL이 나뉜 화면은 **Type: HTTP, URL: localhost:8080**입니다.

공개 웹 경로는 인터넷에서 접근할 수 있습니다. 회사 로그인 뒤에만 열려야 하는 업무 앱은 먼저 [Access 애플리케이션과 허용 정책](#)을 준비하세요. 터널 토큰은 직원의 웹 로그인 정책을 대신하지 않습니다.

[Tunnels → 내 터널 → Routes ↗](#)

The screenshot shows a dialog box titled "Add published application". It contains the following fields and options:

- Hostname:** A note states "Multi-level subdomains require Advanced Certificate Manager. [Learn more](#)".
- Subdomain (optional):** A text input field containing "app". Below it, a note says "e.g., www, blog, api".
- Domain:** A dropdown menu showing "example.com". Below it, a note says "Select a zone, or enter a [custom hostname](#) validated with Cloudflare for SaaS".
- Full hostname:** A read-only field displaying "app.example.com".
- Path (optional):** A text input field containing "^/blog". Below it, a note says "Optional regex, e.g., */api or \.(jpg|png)\$" and a link "[Learn more](#)".
- Service URL:** A text input field containing "http://localhost:8080". Below it, a note says "e.g., https://localhost:8080, tcp://localhost:3306".
- Additional application settings:** A dropdown menu.
- Buttons:** "Cancel" and "Add route".

실제 추가 양식 · 도메인은 example.com으로 가렸습니다. 예시값만 입력했고 저장하지 않았습니다.

새 브라우저에서 웹 주소까지 열어 확인합니다

- 설정된 실제 주소를 새 브라우저에서 엽니다. 위 예시라면 `https://app.example.com` 형식입니다.
- Access를 적용했다면 **허용된 회사 계정으로 로그인**한 뒤 앱이 열리는지 봅니다. 허용되지 않은 계정은 차단되는지도 확인합니다.
- 대시보드 Healthy**와 **실제 앱 응답** 두 가지가 맞으면 연결 확인이 끝났습니다.

Healthy인데 **502가 나오면** 앞 단계의 Service URL·포트·HTTP/HTTPS와 앱 응답을 다시 확인합니다. **Inactive·Down**이면 서버에서 실행한 연결 명령과 서비스 상태를 먼저 확인합니다.

Linux 설치 안내 12단계

Linux · 1 / 12단계

Ubuntu·Debian 터미널에서 준비합니다

이 Linux 절차는 **Ubuntu·Debian 계열**의 apt 패키지 관리자를 사용합니다. 먼저 배포판과 `curl` 을 확인합니다.

`curl`이 없다면 `sudo apt-get update` 후 `sudo apt-get install curl ca-certificates` 로 준비합니다. RHEL·Amazon Linux 등은 [공식 저장소의 해당 배포판 절차](#)를 사용하세요.

```
cat /etc/os-release
curl --version
```

Linux · 2 / 12단계

Cloudflare 서명 키와 안정 버전 저장소를 등록합니다

아래 블록을 터미널에 붙여 실행합니다. 패키지 서명 키와 cloudflared 전용 apt 저장소를 추가합니다.

기존에 같은 저장소를 관리하고 있다면 회사의 패키지 관리 절차를 따르세요. 직원용 WARP의 패키지 저장소와 주소가 다릅니다.

```
sudo mkdir -p --mode=0755 /usr/share/keyrings
curl -fsSL https://pkg.cloudflare.com/cloudflare-main.gpg | sudo tee /usr/share/keyrings/cloudflare-main.gpg >/dev/null
echo 'deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg.cloudflare.com/cloudflared any main' | sudo tee /etc/apt/sources.list.d/cloudflared.list
```

Linux · 3 / 12단계

목록을 갱신하고 cloudflared를 설치합니다

두 명령을 순서대로 실행합니다. 첫 명령에서 저장소나 서명 오류가 나오면 그 오류를 먼저 해결한 뒤 설치합니다.

설치할 패키지와 용량을 확인하고, 계속 진행할지 물으면 `y` 를 입력합니다.

```
sudo apt-get update
sudo apt-get install cloudflared
```

설치된 버전과 실행 파일 경로를 확인합니다

버전 번호와 실행 파일 경로가 나오면 설치 확인이 끝났습니다. 이 결과만으로 터널이 연결된 것은 아닙니다. 다음 작업은 실제 터널의 커넥터 실행입니다.

인터넷이 제한된 서버에서는 설치 파일을 받을 수 있는지와, 연결할 때 필요한 네트워크 조건을 [공식 시작 안내](#)에서 함께 확인하세요.

```
cloudflared --version
command -v cloudflared
```

Cloudflare 웹에 로그인하고 터널 목록을 엽니다

- 아래 버튼을 열어 Cloudflare 계정에 로그인합니다. 사용하는 SSO나 2단계 인증이 있으면 완료합니다.
- 여러 계정이 보이면 **이 서버의 터널을 관리할 계정**을 선택합니다.
- Networking › Tunnels**로 이동합니다. 계정에 따라 기존 Cloudflare One의 **Networks › Connectors**에서 같은 터널 목록으로 들어갈 수도 있습니다.

이 단계는 Cloudflare [관리 웹사이트 로그인](#)입니다. cloudflared 프로그램에는 WARP처럼 회사 계정을 넣는 로그인 창이 없습니다.

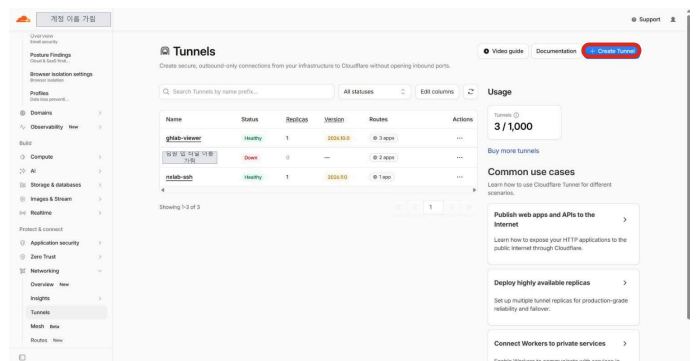
[Tunnels 열기 · 내 계정 선택](#)

Create Tunnel을 누르고 터널 이름을 정합니다

- 목록 오른쪽 위 **Create Tunnel**을 누릅니다. 유형 선택이 나오면 **cloudflared**를 선택합니다.
- 터널 이름에는 연결할 서버의 용도를 적습니다. 예: `office-web-connector`.
- Create Tunnel** 또는 **Save tunnel**을 눌러 생성합니다. 다음으로 설치 환경과 실행 명령 화면이 열립니다.

이미 만들어 둔 터널에 서버를 연결하는 경우에는 새로 만들지 말고, 해당 터널 → **Add a replica**를 엽니다. 이름만 생긴 단계에서는 서버 연결이 끝난 것이 아닙니다.

[Tunnels 목록 열기 · 내 계정](#)



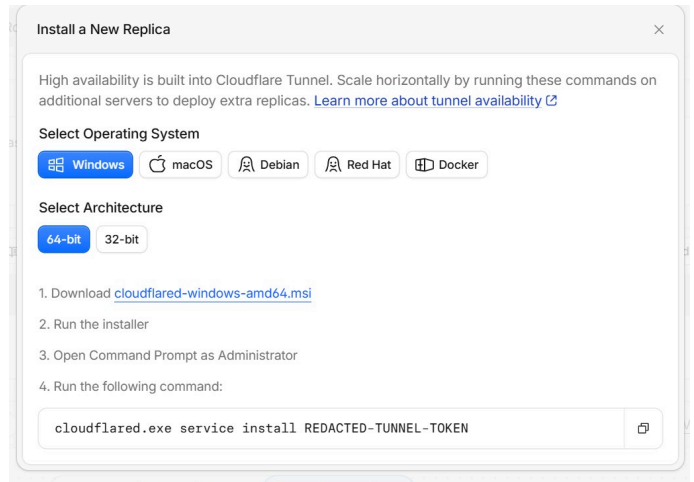
실제 화면 예시 · 누르면 크게 볼 수 있습니다

서버 OS를 고르고 토큰이 포함된 명령을 복사합니다

1. 새 터널은 생성 직후의 설치 화면에서, 기존 터널은 **터널 이름** → **Add a replica**에서 명령을 확인합니다.
2. **Setup Environment** 또는 **Select Operating System** · **Select Architecture**에서 서버 OS와 CPU 종류를 고릅니다. 실제 대시보드에서도 본인 환경을 맞춰 선택합니다.
3. **Install and Run** 또는 **Run the following command**를 찾습니다. 이미 설치했다면 **cloudflared**를 서비스로 등록·실행하는 명령을 사용합니다.
4. 명령 오른쪽의 **복사 아이콘**으로 토큰이 포함된 **명령 전체**를 복사합니다. 명령 뒤의 긴 문자열이 이 터널의 연결 토큰입니다. 토큰이 화면에서 가려져 있어도 별표를 직접 입력하지 않습니다.

API Tokens 메뉴에서 API 키를 발급하는 단계가 아닙니다. 이 토큰으로 어느 터널에 붙을지가 정해집니다. 토큰과 토큰이 든 명령은 문서·채팅·화면 캡처에 공유하지 않습니다.

[Tunnels → 내 터널 열기 ↗](#)



실제 Add a replica 화면 · Windows 예시 · 토큰은 가렸습니다. 눌러서 확대하세요.

복사한 실제 명령을 서버에서 실행합니다

1. cloudflared를 설치한 Ubuntu·Debian 서버의 터미널을 엽니다.
2. 대시보드에서 복사한 **sudo cloudflared service install** 명령을 토큰까지 통째로 실행합니다.
3. 서비스 설치 성공 메시지를 확인하고, 아래 상태 명령에서 **active**인지 봅니다.
4. 그다음 웹 대시보드로 돌아가 커넥터 연결과 **Healthy**를 확인합니다.

아래는 토큰을 가린 형식 예시입니다. 여기에 적힌 가림 문자열을 실행하지 말고, 본인 웹 대시보드에서 복사한 실제 명령을 사용합니다.

형식 예시 · 실제 토큰은 가림

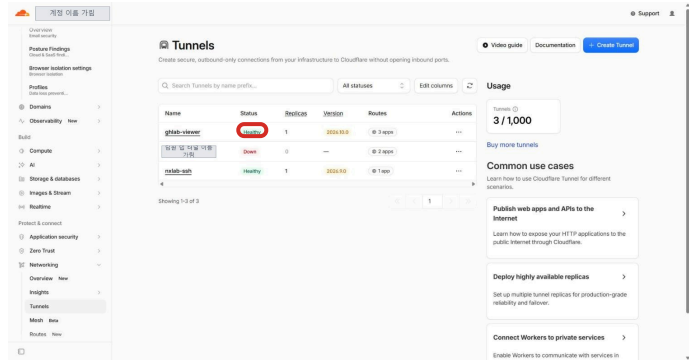
```
sudo cloudflared service install REDACTED-TUNNEL-TOKEN
```

```
sudo systemctl is-active cloudflared
```

웹으로 돌아가 서버의 연결 상태를 확인합니다

1. 명령 실행 뒤 웹 대시보드의 설치 화면으로 돌아옵니다.
2. 커넥터 연결이 잡히면 **Continue**를 누릅니다. 이전 화면에서는 커넥터 목록에 연결된 서버가 표시될 수 있습니다.
3. **Networking > Tunnels** 목록에서 방금 선택한 터널이 **Healthy**인지 확인합니다.

터널 이름만 있거나 상태가 **Inactive·Down**이면 연결이 끝나지 않았습니다. **Healthy**가 된 다음 실제 앱이 열리는지도 뒤 단계에서 따로 확인합니다.



실제 화면 예시 · 누르면 크게 볼 수 있습니다

Tunnels 연결 상태 확인 ↗

커넥터 서버에서 앱 주소가 열리는지 확인합니다

이후 절차는 **HTTP 웹 앱을 본인 도메인으로 여는 예시**입니다. 연결할 앱이 이미 실행 중이어야 합니다.

1. cloudflared를 설치한 서버에서 실제 앱 주소를 엽니다. 같은 서버의 8080 포트라면 예를 들어 `http://localhost:8080` 입니다.
2. 다른 서버의 앱이면 그 서버의 실제 내부 주소·포트를 사용합니다.
3. 응답이 없으면 터널 설정 전에 앱 실행 상태와 서버 사이 연결부터 확인합니다.

뒤에서 넣는 **localhost**는 **cloudflared**가 실행되는 서버 자신을 뜻합니다. 지금 브라우저를 보는 개인 PC를 뜻하지 않습니다. 사설 IP/CIDR을 연결할 목적이라면 공개 웹 경로 대신 사설 네트워크 연결 안내를 이어서 봅니다.

Routes에서 웹 호스트와 앱 주소를 연결합니다

- 터널을 열고 **Routes > Add route > Published application**을 고릅니다. 기존 화면의 이름은 **Public hostnames** 또는 **Published application routes**일 수 있습니다.
- Subdomain**에 사용할 이름을 넣고 **Domain**에서 본인 도메인을 선택합니다. 예시: `app` + `example.com` → `app.example.com`.
- 앱 전체를 연결한다면 **Path**는 비웁니다. **Service URL**에는 앞 단계에서 확인한 실제 앱 주소를 넣습니다. 예시: `http://localhost:8080`.
- 실제 값과 보호 정책을 확인하고 **Add route** 또는 **Save**로 저장합니다. 이때 DNS도 설정됩니다.

그림은 예시값을 채운 양식입니다. 본인 도메인과 실제 앱 주소로 바꿉니다. Type·URL이 나뉜 화면은 **Type: HTTP, URL: localhost:8080**입니다.

공개 웹 경로는 인터넷에서 접근할 수 있습니다. 회사 로그인 뒤에만 열려야 하는 업무 앱은 먼저 [Access 애플리케이션과 허용 정책](#)을 준비하세요. 터널 토큰은 직원의 웹 로그인 정책을 대신하지 않습니다.

[Tunnels](#) → [내 터널](#) → [Routes](#) ↗

The screenshot shows a dialog box titled "Add published application". It contains the following fields and options:

- Hostname:** A note states "Multi-level subdomains require Advanced Certificate Manager. [Learn more](#)".
- Subdomain (optional):** A text input field containing "app". Below it, a note says "e.g., www, blog, api".
- Domain:** A dropdown menu showing "example.com". Below it, a note says "Select a zone, or enter a [custom hostname](#) validated with Cloudflare for SaaS".
- Full hostname:** A read-only field displaying "app.example.com".
- Path (optional):** A text input field containing "^/blog". Below it, a note says "Optional regex, e.g., ^/api or \.(jpg|png)\$" and a link "[Learn more](#)".
- Service URL:** A text input field containing "http://localhost:8080". Below it, a note says "e.g., https://localhost:8080, tcp://localhost:3306".
- Additional application settings:** A dropdown menu.
- Buttons:** "Cancel" and "Add route".

실제 추가 양식 · 도메인은 example.com으로 가렸습니다. 예시값만 입력했고 저장하지 않았습니다.

새 브라우저에서 웹 주소까지 열어 확인합니다

- 설정된 실제 주소를 새 브라우저에서 엽니다. 위 예시라면 `https://app.example.com` 형식입니다.
- Access를 적용했다면 **허용된 회사 계정으로 로그인**한 뒤 앱이 열리는지 봅니다. 허용되지 않은 계정은 차단되는지도 확인합니다.
- 대시보드 **Healthy**와 **실제 앱 응답** 두 가지가 맞으면 연결 확인이 끝났습니다.

Healthy인데 502가 나오면 앞 단계의 Service URL·포트·HTTP/HTTPS와 앱 응답을 다시 확인합니다. **Inactive·Down이면** 서버에서 실행한 연결 명령과 서비스 상태를 먼저 확인합니다.

대시보드 캡처는 우리 실습장 계정의 실제 화면입니다. 계정·팀원 정보와 터널 토큰은 가렸고, 웹 주소 연결 화면의 도메인은 example.com으로 바꿨습니다. 명령의 REDACTED-TUNNEL-TOKEN은 실제 키가 아닌 가림 표시입니다. 실행할 명령은 본인 대시보드에서 복사합니다. 대시보드에서 관리하는 터널 방식으로 진행합니다.

어디서 막혔는지 확인하세요

토큰 오류: 연결할 터널을 다시 열어 그 터널의 명령을 통째로 복사합니다. 예시의 가림 문자열이나 Cloudflare API 키를 넣지 않습니다.

이미 서비스가 있다고 나옴: 기존 cloudflared가 어느 터널을 연결하는지 먼저 확인합니다. 기존 서비스를 무작정 지우거나 덮어쓰지 않습니다.

Inactive·Down: 서버에서 프로세스·서비스가 실행 중인지, Cloudflare로 나가는 7844번 TCP·UDP 연결이 가능한지 확인합니다. **Degraded**는 연결 일부에 문제가 있는 상태입니다.

Healthy인데 웹이 502: 커넥터에서 앱에 닿는지와 Service URL의 주소·포트·HTTP/HTTPS를 확인합니다. Healthy는 터널 연결 상태이며 앱 응답까지 보장하지는 않습니다.

직원 PC의 회사 로그인: 서버 커넥터와 별개로 [Cloudflare One Client\(WARP\) 안내](#)를 따릅니다.

▼ cloudflared tunnel login은 언제 쓰나요?

브라우저 인증으로 cert.pem을 받는 로컬 관리 방식에서 쓰는 명령입니다. 이 안내의 대시보드 관리 방식은 터널 토큰으로 연결하므로 그 로그인을 먼저 실행하지 않습니다. [로컬 관리 방식 공식 안내](#)

공식 안내

[공식 cloudflared 다운로드 ↗](#)

[공식 Linux 패키지 저장소 ↗](#)

[대시보드에서 터널 만들고 연결하기 ↗](#)

[서버 서비스·실행 매개변수 ↗](#)

[터널 상태·연결 확인 ↗](#)

내용 확인 2026-10-19