Setting up Cloudflare Tunnel for Clips

I have a cloudflare tunnel on my win11 machine, and i am running comfyui with a cname to it ... i just setup a new machine .... what do i have to do to set up a new tunnel (I can have multiple tunnels on multiiple machines, right?)

Yes, you can absolutely have multiple Cloudflare Tunnels running on different machines! Each tunnel can point to different services.

Here's what you need to do to set up a new tunnel on your new machine:

Quick Setup Steps

1. **Install cloudflared on the new machine**
   - Download from: https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/downloads/
   - Or use winget: `winget install --id Cloudflare.cloudflared`
2. **Authenticate (if you haven't already on this machine)**

bash

```
cloudflared tunnel login
```

This opens a browser to authenticate with your Cloudflare account.

3. **Create a new tunnel**

bash

```
cloudflared tunnel create <tunnel-name>
```

(Use a different name than your first tunnel, like "comfyui-machine2")

4. **Create a config file** Create `config.yml` in `C:\Users\<YourUsername>\.cloudflared\` or wherever you prefer:

yaml

```
tunnel: <tunnel-id-from-step-3>
credentials-file: C:\Users\<YourUsername>\.cloudflared\<tunnel-id>.json

ingress:
  - hostname: your-subdomain.yourdomain.com
    service: http://localhost:8188
```

```
  - service: http_status:404
```

5. **Create a DNS CNAME record**

bash

```
cloudflared tunnel route dns <tunnel-name> your-subdomain.yourdomain.com
```

6. **Run the tunnel**

bash

```
cloudflared tunnel run <tunnel-name>
```

7. **Set up as a service** (so it runs automatically)

bash

```
cloudflared service install
```

Key Points

- Each machine needs its own tunnel with a unique name
- Each tunnel gets its own credentials file
- You can point different CNAMEs to different tunnels
- They all show up in your Cloudflare Zero Trust dashboard

Does your new machine also run ComfyUI, or are you exposing a different service?

installed cloudflare, did the login , it opened the website, i logged in , the cli is still waiting for the login ... stuck

This is a common issue with the Cloudflare Tunnel login flow on Windows. Here are some solutions:

Quick Fixes to Try:

1. **Check if a browser process is holding the connection**
   - Close the browser window that opened
   - Go back to your terminal and press Enter or wait a moment
   - Sometimes it completes but doesn't show immediately
2. **Try the login with a specific browser**

bash

```
cloudflared tunnel login --browser firefox
```

Or specify your default browser explicitly

3. **Manual certificate method** (most reliable):
    - Go to: https://dash.cloudflare.com/
    - Navigate to your domain
    - Go to **SSL/TLS → Origin Server**
    - Actually, scratch that - try this instead:

    Just manually get the cert file:

    - The browser should have downloaded a `cert.pem` file when you authenticated
    - Check your Downloads folder for `cert.pem`
    - Move it to: `C:\Users\<YourUsername>\.cloudflared\cert.pem`
    - Then cancel the CLI (Ctrl+C) and try the next step
4. **If still stuck, try running as administrator**
    - Open PowerShell or CMD as Administrator
    - Run `cloudflared tunnel login` again
5. **Check firewall/antivirus**
    - Sometimes security software blocks the callback
    - Temporarily disable and try again

Which of these you want to try first? The manual cert.pem method usually works when the callback gets stuck.

do i need to both run the tunnel and start the service?

No, you only need to do **one or the other**:

Option 1: Run as a Service (Recommended)

This makes the tunnel start automatically on boot and run in the background.

bash

```
cloudflared service install
```

Then it runs automatically. You can manage it with:

- `cloudflared service start`
- `cloudflared service stop`
- `cloudflared service uninstall`

Option 2: Run Manually

Just for testing or temporary use:

bash

```
cloudflared tunnel run <tunnel-name>
```

This runs in the foreground and stops when you close the terminal.

Best practice:

1. First test with `cloudflared tunnel run` to make sure everything works
2. Once confirmed, stop it (Ctrl+C) and install as a service
3. The service will use the same config file you created

So the flow is:

- Create tunnel → Create config → Test with `run` → If it works, install as service

Are you at the testing stage or ready to install as a service?

it works, sorta when run manually, but not at all as a service, is it because the tunnel run - tunnelname is not correct?

for this, i put the long tunnel id for the tunnel name:

cloudflared tunnel route dns <tunnel-name> your-subdomain.yourdomain.com

Good catch! Yes, that could be part of the issue. Let me clarify:

Tunnel Name vs Tunnel ID

- **Tunnel name**: The friendly name you gave it when creating (e.g., "comfyui-machine2")
- **Tunnel ID**: The long UUID (e.g., "a1b2c3d4-e5f6-...")

For the DNS route command, you can use **either** the name or ID, but it's clearer to use the name.

Why the service might not be working:

1. **Check your config.yml file** - it should have the tunnel ID (not name):

yaml

```yaml
tunnel: a1b2c3d4-e5f6-... # <-- The long UUID
credentials-file: C:\Users\<YourUsername>\.cloudflared\a1b2c3d4-e5f6-....json

ingress:
  - hostname: your-subdomain.yourdomain.com
    service: http://localhost:8188
```

```
- service: http_status:404
```

2. **The service needs to find the config file** When you run manually: `cloudflared tunnel run <tunnel-name>` - it finds the config When running as a service, it looks for config in the default location:
    - `C:\Users\<YourUsername>\.cloudflared\config.yml`
    - OR you need to specify the path when installing

To fix:

Option A: Make sure config.yml is in the right place

bash

```
# Check if config exists here:
C:\Users\<YourUsername>\.cloudflared\config.yml
```

Option B: Reinstall service with explicit config path

bash

```
cloudflared service uninstall
cloudflared --config C:\path\to\your\config.yml service install
```

What does your manual run command look like exactly? And where is your config.yml file located?

yes, but the: cloudflared tunnel route dns <tunnel-name> your-subdomain.yourdomain.com .... route has the wrong name on it

Ah, got it! So you created the DNS route with the wrong tunnel name/ID, and now traffic is being routed to the wrong tunnel.

To fix the DNS route:

1. **Delete the incorrect route:**

bash

```
cloudflared tunnel route dns delete your-subdomain.yourdomain.com
```

2. **Create the correct route with the right tunnel name:**

bash

```
cloudflared tunnel route dns <correct-tunnel-name> your-subdomain.yourdomain.com
```

Or check/fix via Cloudflare Dashboard:

You can also manage this in the Cloudflare dashboard:

- Go to https://one.dash.cloudflare.com/
- **Access → Tunnels**
- Find your tunnel and check its public hostnames
- Delete/add the correct hostname mapping

To verify which tunnel a hostname points to:

bash

```
cloudflared tunnel route ip show
```

This shows all your routes and which tunnels they point to.

Once you fix the DNS route to point to the correct tunnel, the service should work properly (assuming your config.yml is correct).

Does that solve it?

no such dlete cammand: cloudflared tunnel route dns delete your-subdomain.yourdomain.com

You're right, my bad! There's no direct delete command. Here's how to actually fix it:

Option 1: Use the Cloudflare Dashboard (Easiest)

1. Go to your Cloudflare DNS settings:
   - https://dash.cloudflare.com/
   - Select your domain
   - Go to **DNS → Records**
2. Find the CNAME record for `your-subdomain`
   - It will point to something like `<tunnel-id>.cfargotunnel.com`
   - Delete that CNAME record
3. Create the new route with the correct tunnel:

bash

```
cloudflared tunnel route dns <correct-tunnel-name> your-subdomain.yourdomain.com
```

Option 2: Manually edit the DNS record

In the same DNS Records page, you can just **edit** the existing CNAME to point to the correct tunnel ID instead of deleting and recreating.

The CNAME target should be: `<correct-tunnel-id>.cfargotunnel.com`

Option 3: Check via Zero Trust Dashboard

- Go to https://one.dash.cloudflare.com/
- **Access → Tunnels**
- Click on each tunnel to see/manage its public hostnames
- Remove hostname from wrong tunnel, add to correct one

Which method works best for you?

the dns record is fine

Ah okay, so the DNS record is correct. Let's troubleshoot why the service isn't working:

Check these:

1. **Is the service actually running?**

bash

```
sc query cloudflared
```

Should show "RUNNING"

2. **Check the service logs** (this will tell us what's wrong):

bash

```
cloudflared service uninstall
cloudflared --config C:\Users\<YourUsername>\.cloudflared\config.yml service install
```

Then check Windows Event Viewer:

- Open Event Viewer → Windows Logs → Application
- Look for cloudflared errors

3. **What's in your config.yml?** Can you share the contents? It should look like:

yaml

```
tunnel: <long-tunnel-id>
credentials-file: C:\Users\<YourUsername>\.cloudflared\<tunnel-id>.json

ingress:
  - hostname: your-subdomain.yourdomain.com
```

```
  service: http://localhost:8188
- service: http_status:404
```

4. **When you run manually, what exact command works?**

bash

```
cloudflared tunnel run <what-exactly-do-you-type-here>
```

The difference between manual working and service not working is usually:

- Config file location/path issue
- Permissions issue
- Service can't find ComfyUI (localhost:8188)

What does `sc query cloudflared` show?

it will not let me stop the service ... .. I notice on my other win11 machine i have the cloudflare running as a task in task scheduler, not a service ... i wonder if that has any thing related to this ... BUT ... let's get it working from the

cloudflared

Good point! Let's just get it working from the command line first and figure out the service/task scheduler thing later.

To stop the stuck service:

Try these in order:

1. **Force stop via PowerShell (as Administrator):**

powershell

```
Stop-Service cloudflared -Force
```

2. **If that doesn't work, kill the process:**

powershell

```
Get-Process cloudflared | Stop-Process -Force
```

3. **If still stuck, uninstall the service:**

powershell

```
cloudflared service uninstall
```

(This should work even if the service is stuck)

Then let's just run it manually:

Once you've stopped/killed everything:

bash

```
cloudflared tunnel run <your-tunnel-name-or-id>
```

What exact command works when you run it manually? And does ComfyUI need to be running first, or does the tunnel work either way?

Once we confirm it's working manually, we can decide whether to:

- Set it up as a service (properly this time)
- Or use Task Scheduler like your other machine

What happens when you try to stop it?

i got rid of the service, it is working now .. the config had a .txt at the end of it .... i will set it up as a task schedular task ... that works on my other machine

Ah, the classic `.txt` extension issue on Windows! Glad you found it.

Setting up as a Task Scheduler task:

Here's what you probably have on your other machine:

1. **Open Task Scheduler**
   - Press Win+R, type `taskschd.msc`
2. **Create Basic Task**
   - Name it something like "Cloudflare Tunnel - ComfyUI"
   - Trigger: **At startup** or **At log on**
   - Action: **Start a program**
3. **Program settings:**
   - Program: `C:\Program Files (x86)\cloudflared\cloudflared.exe` (or wherever it's installed)
   - Arguments: `tunnel run <your-tunnel-name>`
   - Start in: `C:\Users\<YourUsername>\.cloudflared\` (where your config is)
4. **Important settings** (right-click task → Properties):
   - General tab: Check "Run whether user is logged on or not"
   - General tab: Check "Run with highest privileges"
   - Conditions tab: Uncheck "Start only if on AC power" (if laptop)
   - Settings tab: If task fails, restart every 1 minute

That should give you the same setup as your other machine. The task scheduler approach actually gives you more control than the service anyway.

Does your other machine have similar settings?

all working, the tunnel is to get to my comfyui server ... which i run manually from the c prompt .... should i try to task schedule it ... it seems like it need some manualy intervention/love ..... restarts, etc

Good question! ComfyUI can be finicky, so here's my take:

I'd recommend manual for ComfyUI because:

1. **Crashes/hangs are common** - You'll want to see the console output to troubleshoot
2. **Model loading** - Sometimes takes forever or fails, you want to see it
3. **VRAM issues** - Need to restart when things get stuck
4. **Updates/custom nodes** - Frequently need manual intervention
5. **Different workflows** - Might want to pass different flags/arguments

If you DO want to automate it:

Compromise approach - Create a Task Scheduler task but **don't** set it to run at startup. Instead:

- Set trigger to "On demand" only
- Then you can right-click → Run whenever you want
- Or create a desktop shortcut to trigger it
- Still gives you the console window to monitor

Better approach - Batch file:

Create a `start-comfyui.bat` on your desktop:

batch

```
@echo off
cd /d C:\path\to\ComfyUI
call venv\Scripts\activate
python main.py --listen 0.0.0.0 --port 8188
pause
```

This way you double-click when you want it, see all output, and can easily restart.

I'd stick with **manual** unless you have a specific need for it to always be running. What do you think?

so, other than crashes and reboots it will run ... lol

okay, i'll just sent up the machine for remote login //// rdp .. jump desktop etc

Ha! Yeah, "other than crashes and reboots" is doing a lot of heavy lifting there! 😄

RDP/Remote Desktop is the right call for this:

Good options:

- **Jump Desktop** - Great choice, works well
- **Windows RDP** - Built-in, reliable
- **Parsec** - If you need GPU access/monitoring (can see VRAM usage, etc.)
- **AnyDesk/TeamViewer** - Also solid

Pro tip for ComfyUI + Remote access:

Since you already have the Cloudflare tunnel for ComfyUI's web interface, you could also:

- Set up **another** Cloudflare tunnel on this same machine for RDP (port 3389)
- Then you can RDP in from anywhere without exposing port 3389 to the internet
- More secure than traditional RDP

But if Jump Desktop is already working for you, stick with that!

At least now when ComfyUI inevitably crashes at 3am, you can restart it from your phone while still in bed. Living the dream! 🎨

Good luck with the new setup!