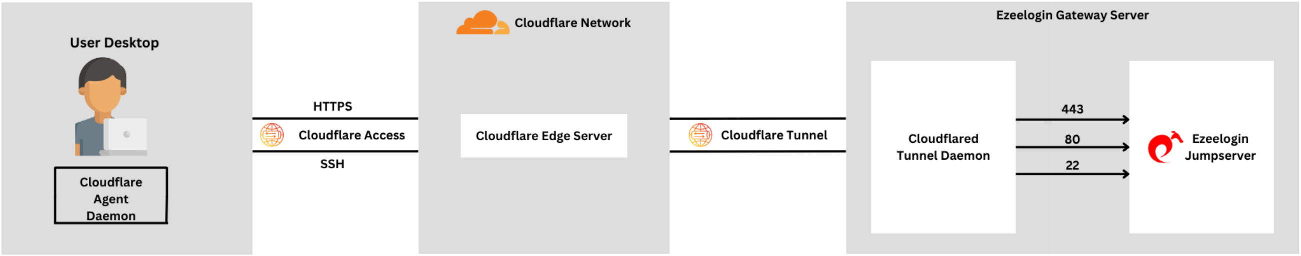
Cloudflare ZTNA for securing SSH access to endpoint devices

720 Manu Chacko July 2, 2024 Security Compliances, Tweaks & Configuration 721

Setup Cloudflare ZTNA (Zero Trust Network Access) with Ezeelogin for securing SSH access

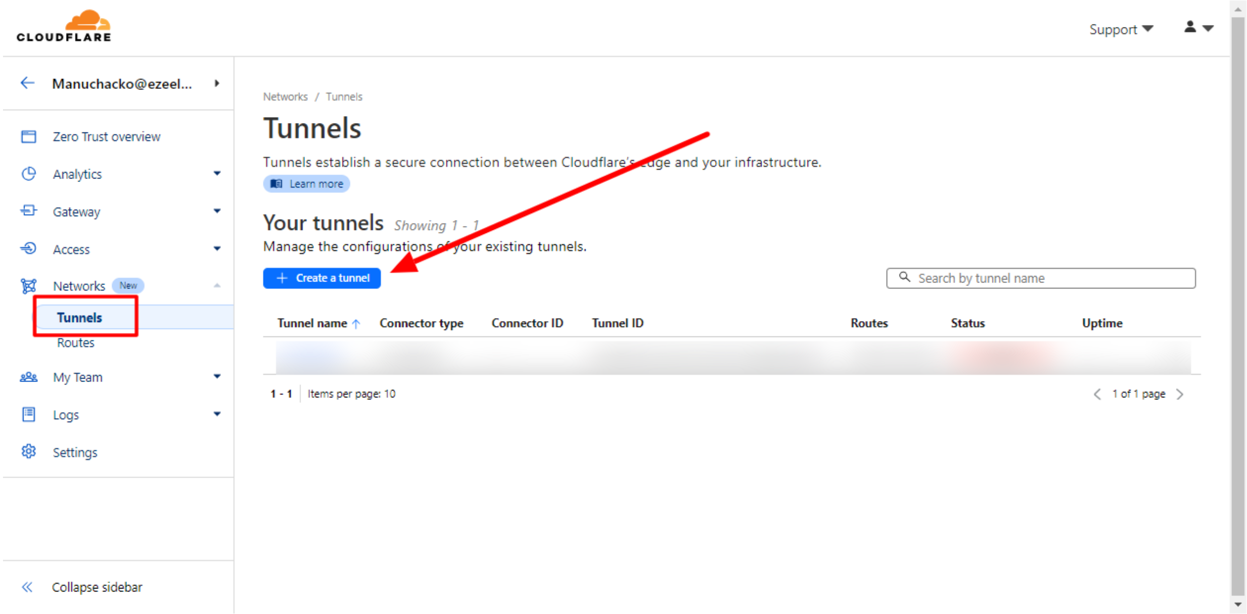
Cloudflare Tunnel can also route applications through a public hostname, allowing users to connect to an SSH server with Cloudflared access. This method requires Cloudflared to be installed on both the server and client machines, as well as an active zone on Cloudflare. The traffic is proxied over this connection.

The benefit of this method is that it provides a secure and reliable way to access Ezeelogin. By routing applications through a public hostname and using Cloudflare **Zero Trust Network Access**, you can enhance security by hiding your server's true IP address and exposing it through a secure Cloudflare Tunnel. This approach simplifies access, allowing users to connect to Ezeelogin from anywhere without the need for a VPN. Additionally, it improves reliability by leveraging Cloudflare's robust network, ensuring consistent and dependable access to your server. Centralized management of access and configurations is made easy through Cloudflare's dashboard, and the attack surface is reduced by minimizing the risk of direct attacks on your server using Cloudflare's security features.



1. Connect the server to Cloudflare

1. Login to Cloudflare dashboard. click on Zero Trust > Network > Tunnels



Tunnels

Tunnels establish a secure connection between Cloudflare's edge and your infrastructure.

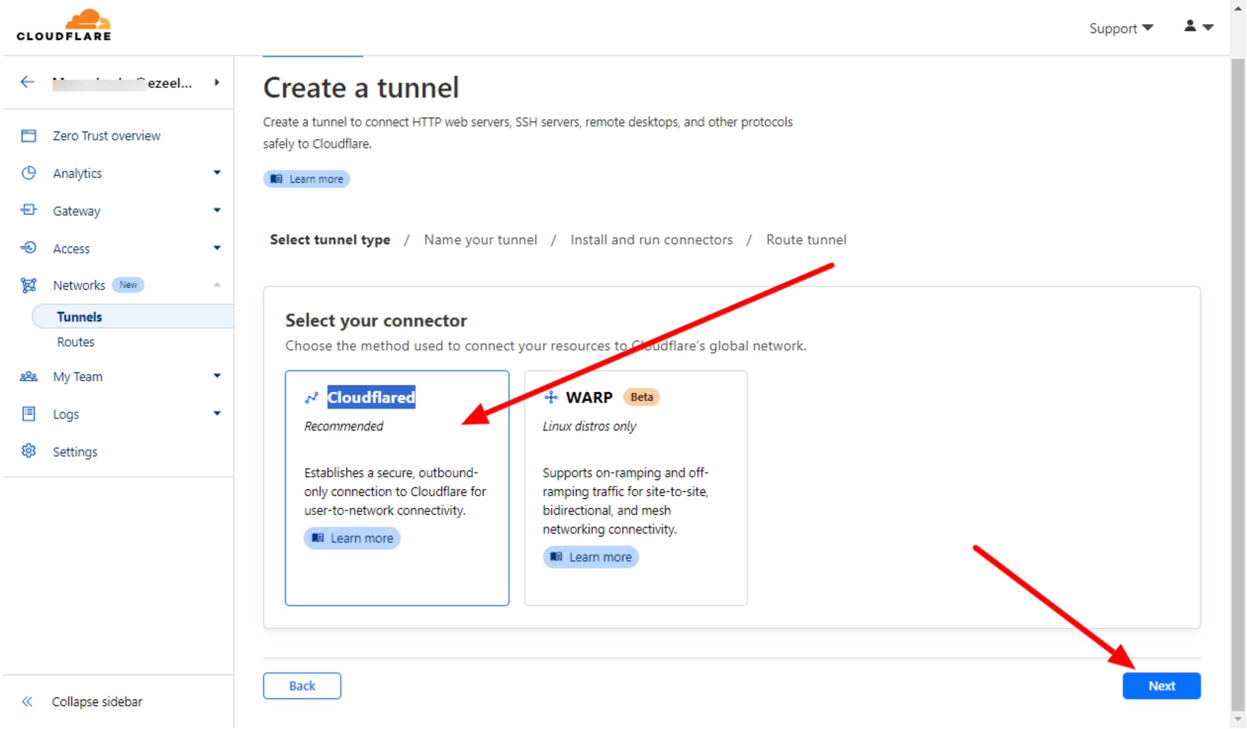
Your tunnels *Showing 1 - 1*

Manage the configurations of your existing tunnels.

Tunnel name	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

1 - 1 Items per page: 10 1 of 1 page

2. Click and **Create tunnel** and select **Cloudflared** connector, then click **next**



Create a tunnel

Create a tunnel to connect HTTP web servers, SSH servers, remote desktops, and other protocols safely to Cloudflare.

Select tunnel type / Name your tunnel / Install and run connectors / Route tunnel

Select your connector

Choose the method used to connect your resources to Cloudflare's global network.

Cloudflared *Recommended*

Establishes a secure, outbound-only connection to Cloudflare for user-to-network connectivity.

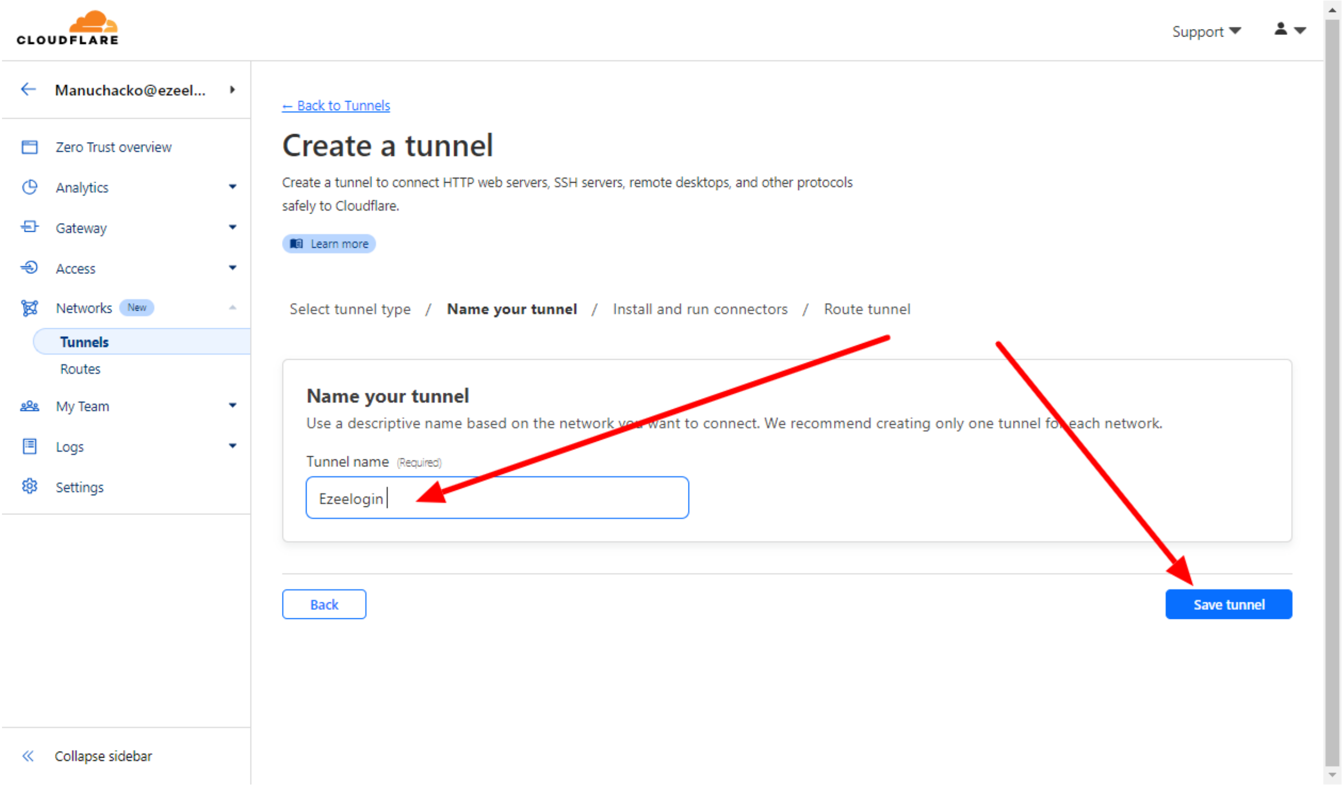
Learn more

WARP Beta *Linux distros only*

Supports on-ramping and off-ramping traffic for site-to-site, bidirectional, and mesh networking connectivity.

Learn more

3. Name your tunnel and **save tunnel**



← Back to Tunnels

Create a tunnel

Create a tunnel to connect HTTP web servers, SSH servers, remote desktops, and other protocols safely to Cloudflare.

Learn more

Select tunnel type / **Name your tunnel** / Install and run connectors / Route tunnel

Name your tunnel

Use a descriptive name based on the network you want to connect. We recommend creating only one tunnel for each network.

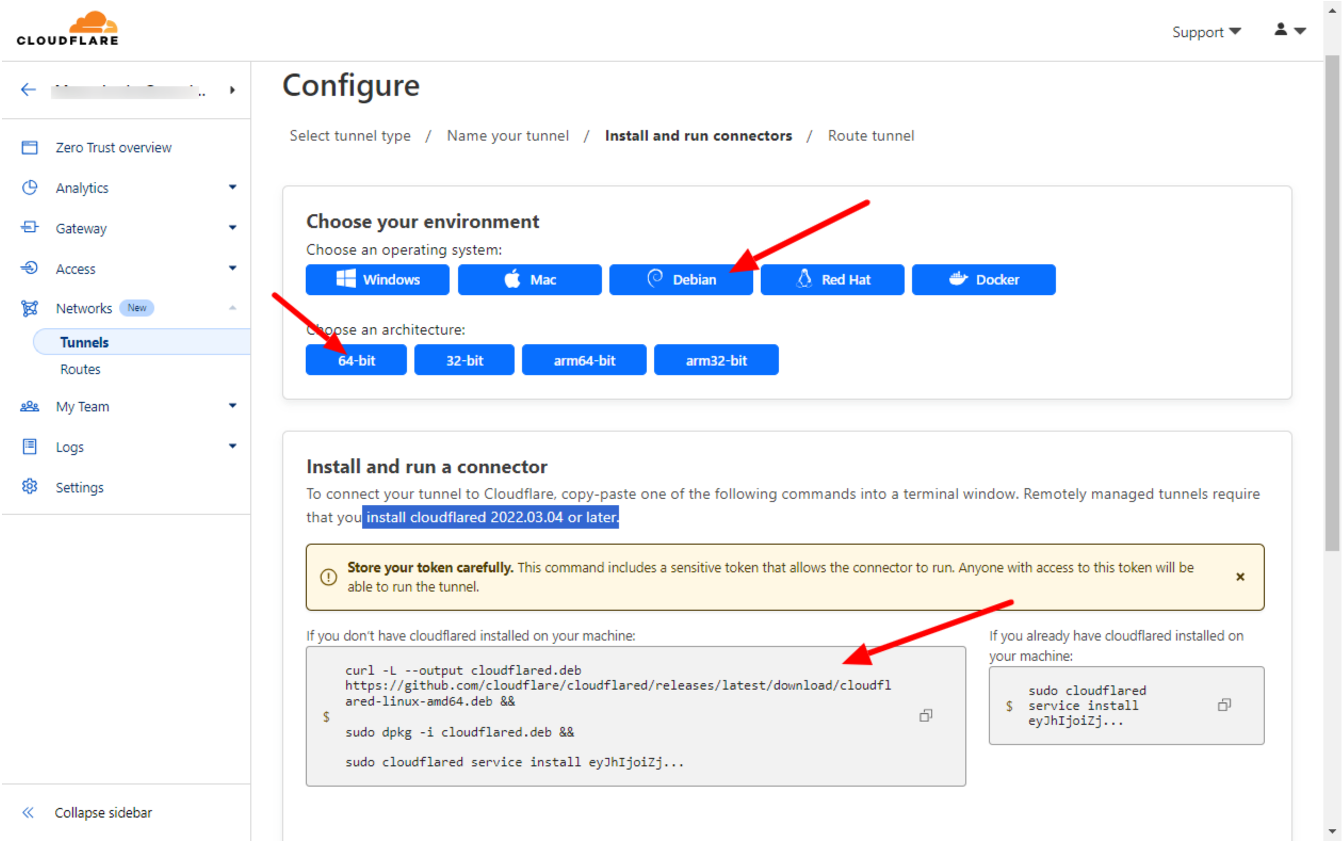
Tunnel name (Required)

Ezeelogin

Back

Save tunnel

4. Now you can install Cloudflared on your Ezeelogin server. Select the operating system of your server and copy the corresponding installation command as follows:



Configure

Select tunnel type / Name your tunnel / **Install and run connectors** / Route tunnel

Choose your environment

Choose an operating system:

Windows | Mac | Debian | Red Hat | Docker

Choose an architecture:

64-bit | 32-bit | arm64-bit | arm32-bit

Install and run a connector

To connect your tunnel to Cloudflare, copy-paste one of the following commands into a terminal window. Remotely managed tunnels require that you install cloudflared 2022.03.04 or later.

Store your token carefully. This command includes a sensitive token that allows the connector to run. Anyone with access to this token will be able to run the tunnel.

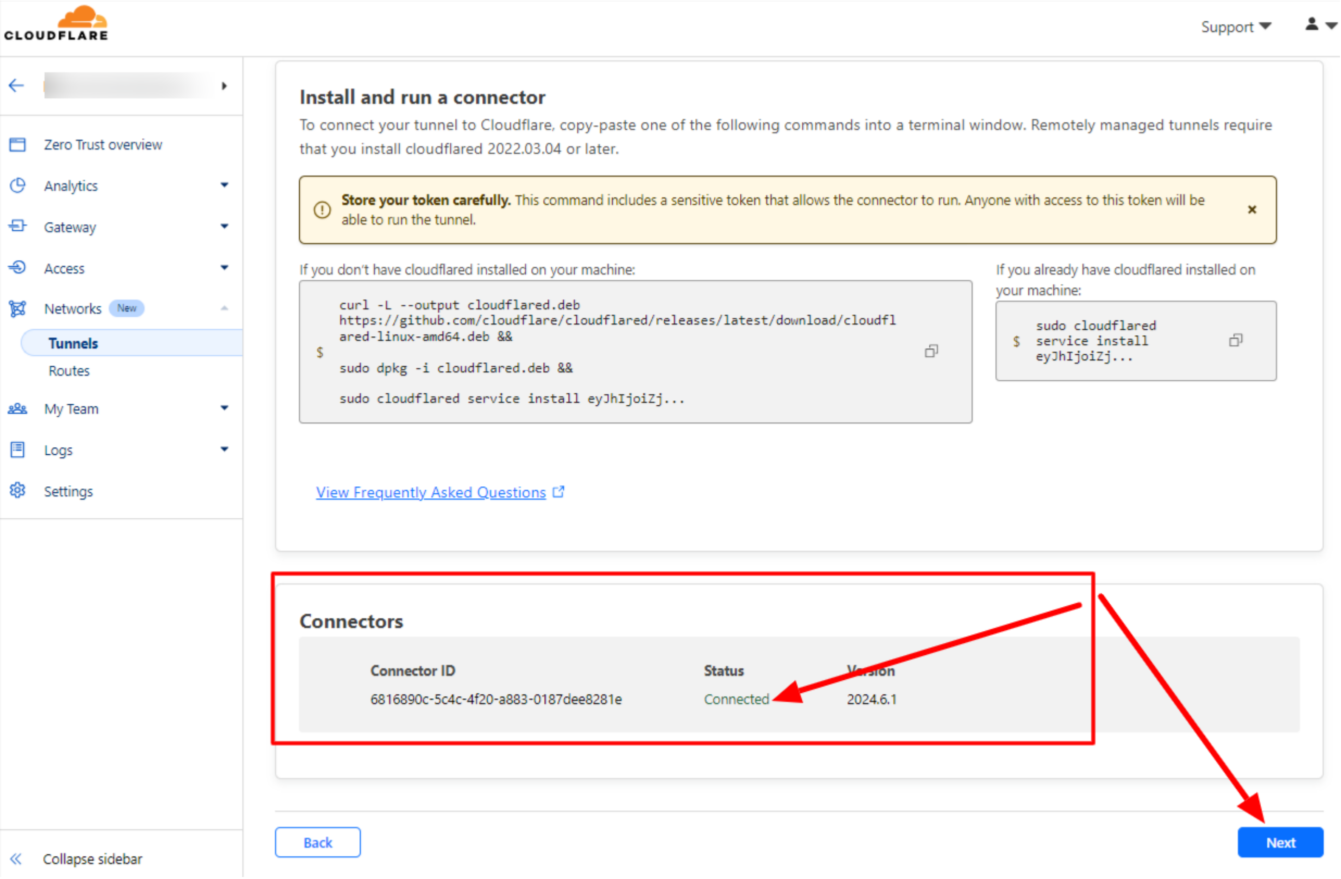
If you don't have cloudflared installed on your machine:

```
$  curl -L --output cloudflared.deb
   https://github.com/cloudflare/cloudflared/releases/latest/download/cloudfl
   ared-linux-amd64.deb &&

   sudo dpkg -i cloudflared.deb &&

   sudo cloudflared service install eyJhIjoiZj...
```

If you already have cloudflared installed on your machine:

```
$  sudo cloudflared
   service install
   eyJhIjoiZj...
```

5. After installation you can view the status as **connected** in the dashboard and click **next**



Install and run a connector

To connect your tunnel to Cloudflare, copy-paste one of the following commands into a terminal window. Remotely managed tunnels require that you install cloudflared 2022.03.04 or later.

Store your token carefully. This command includes a sensitive token that allows the connector to run. Anyone with access to this token will be able to run the tunnel.

If you don't have cloudflared installed on your machine:

```
  curl -L --output cloudflared.deb
  https://github.com/cloudflare/cloudflared/releases/latest/download/cloudfl
  ared-linux-amd64.deb &&
$
  sudo dpkg -i cloudflared.deb &&

  sudo cloudflared service install eyJhIjoiZj...
```

If you already have cloudflared installed on your machine:

```
  sudo cloudflared
$ service install
  eyJhIjoiZj...
```

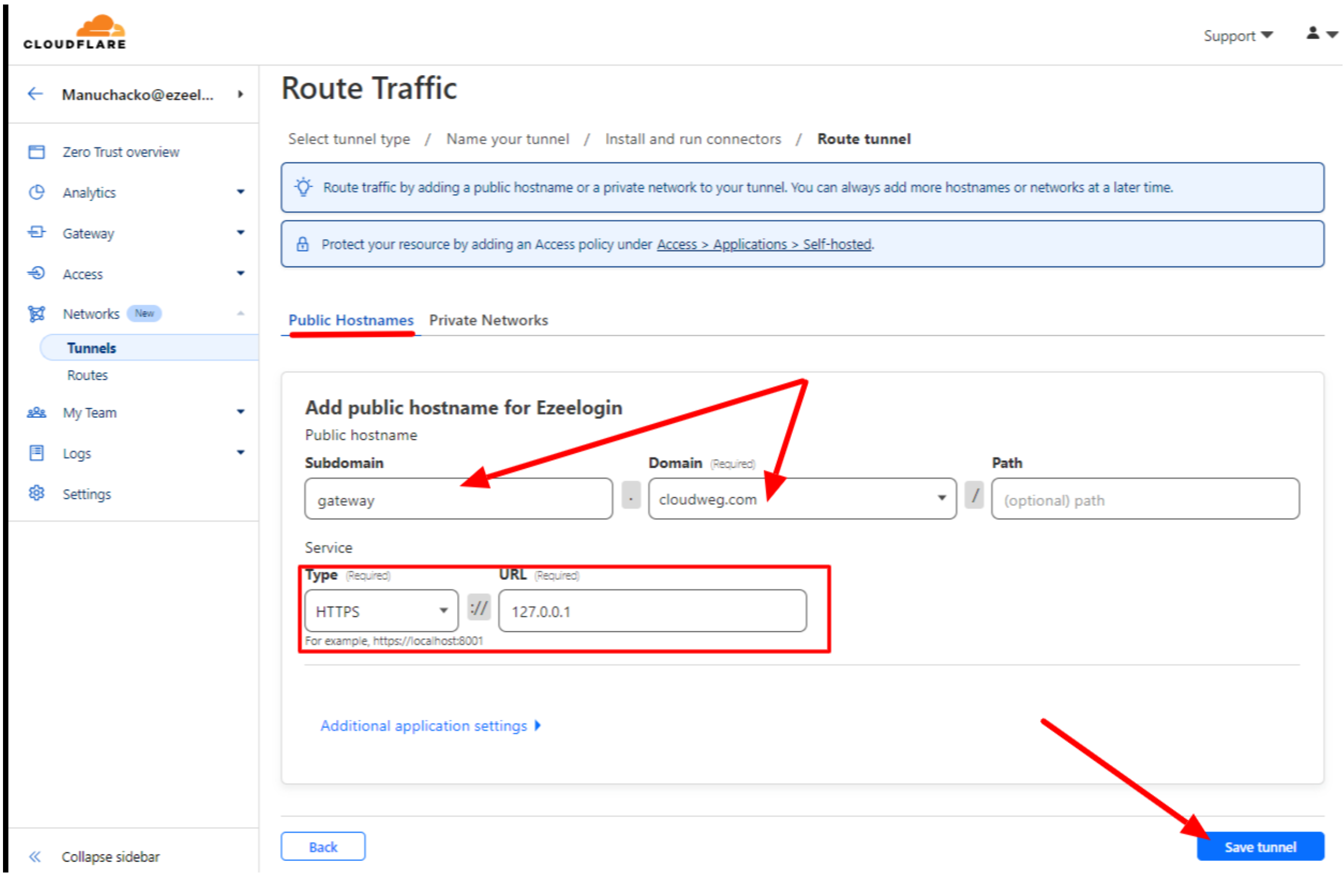
View Frequently Asked Questions

Connectors

Connector ID	Status	Version
6816890c-5c4c-4f20-a883-0187dee8281e	Connected	2024.6.1

6. Choose a **Domain** and specify any **subdomain**(eg: gateway). Select **HTTPS/HTTP** and enter the URL **127.0.0.1** or **localhost**

You can select any domain that is already added in Cloudflare.



Route Traffic

Select tunnel type / Name your tunnel / Install and run connectors / **Route tunnel**

Route traffic by adding a public hostname or a private network to your tunnel. You can always add more hostnames or networks at a later time.

Protect your resource by adding an Access policy under Access > Applications > Self-hosted.

Add public hostname for Ezeelogin

Public hostname

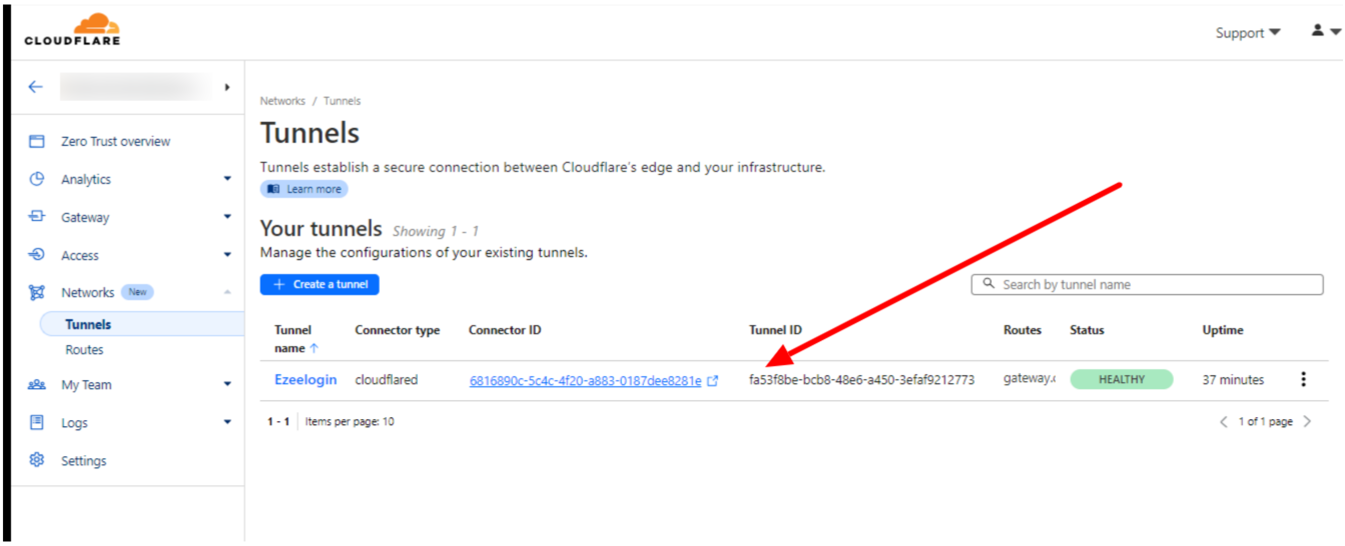
Subdomain	Domain (Required)	Path
gateway	cloudweg.com	(optional) path

Service

Type (Required)	URL (Required)
HTTPS	127.0.0.1

For example, https://localhost:8001

You can view the tunnel in the Tunnels dashboard after saving it.



Tunnels

Tunnels establish a secure connection between Cloudflare's edge and your infrastructure.

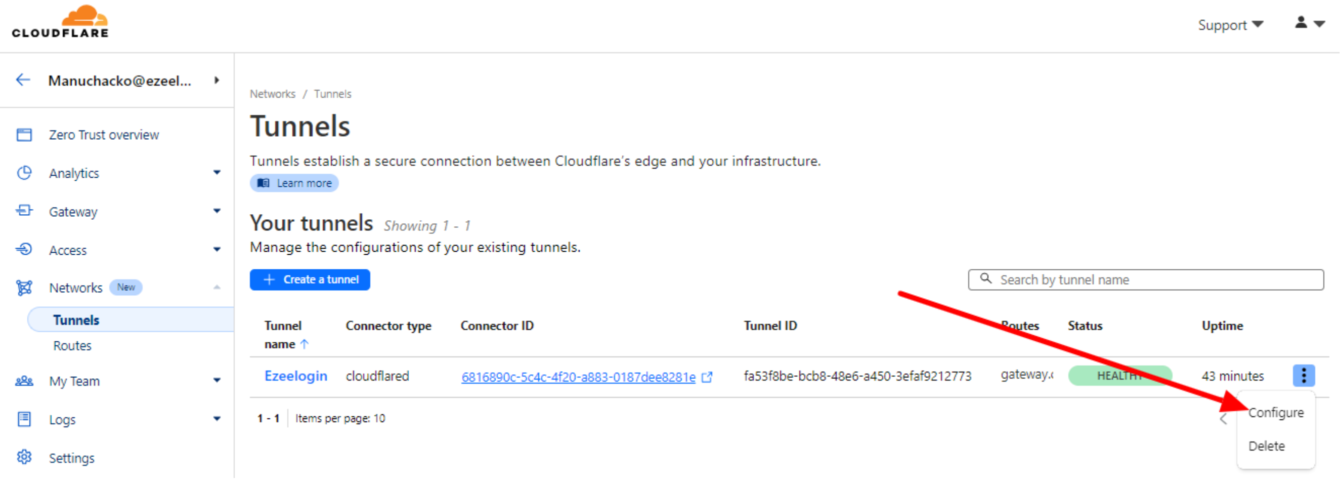
Your tunnels *Showing 1 - 1*

Manage the configurations of your existing tunnels.

Tunnel name	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
Ezeelogin	cloudflared	6816890c-5c4c-4f20-a883-0187dee8281e	fa53f8be-bcb8-48e6-a450-3efaf9212773	gateway.c	HEALTHY	37 minutes

1 - 1 Items per page: 10 1 of 1 page

You can view the **subdomain** created from the **dashboard** > **configure** >**public hostname**



Tunnels

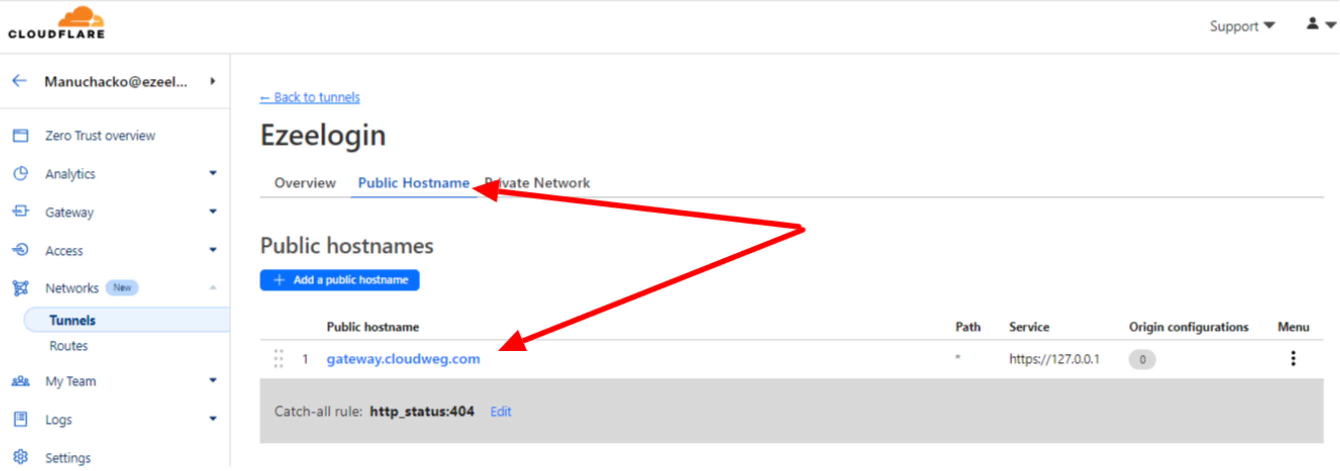
Tunnels establish a secure connection between Cloudflare's edge and your infrastructure.

Your tunnels *Showing 1 - 1*

Manage the configurations of your existing tunnels.

Tunnel name	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
Ezeelogin	cloudflared	6816890c-5c4c-4f20-a883-0187dee8281e	fa53f8be-bcb8-48e6-a450-3efaf9212773	gateway.c	HEALTHY	43 minutes

1 - 1 Items per page: 10



Ezeelogin

Public hostnames

	Public hostname	Path	Service	Origin configurations	Menu
1	gateway.cloudweg.com	*	https://127.0.0.1	0	⋮

Catch-all rule: **http_status:404** Edit

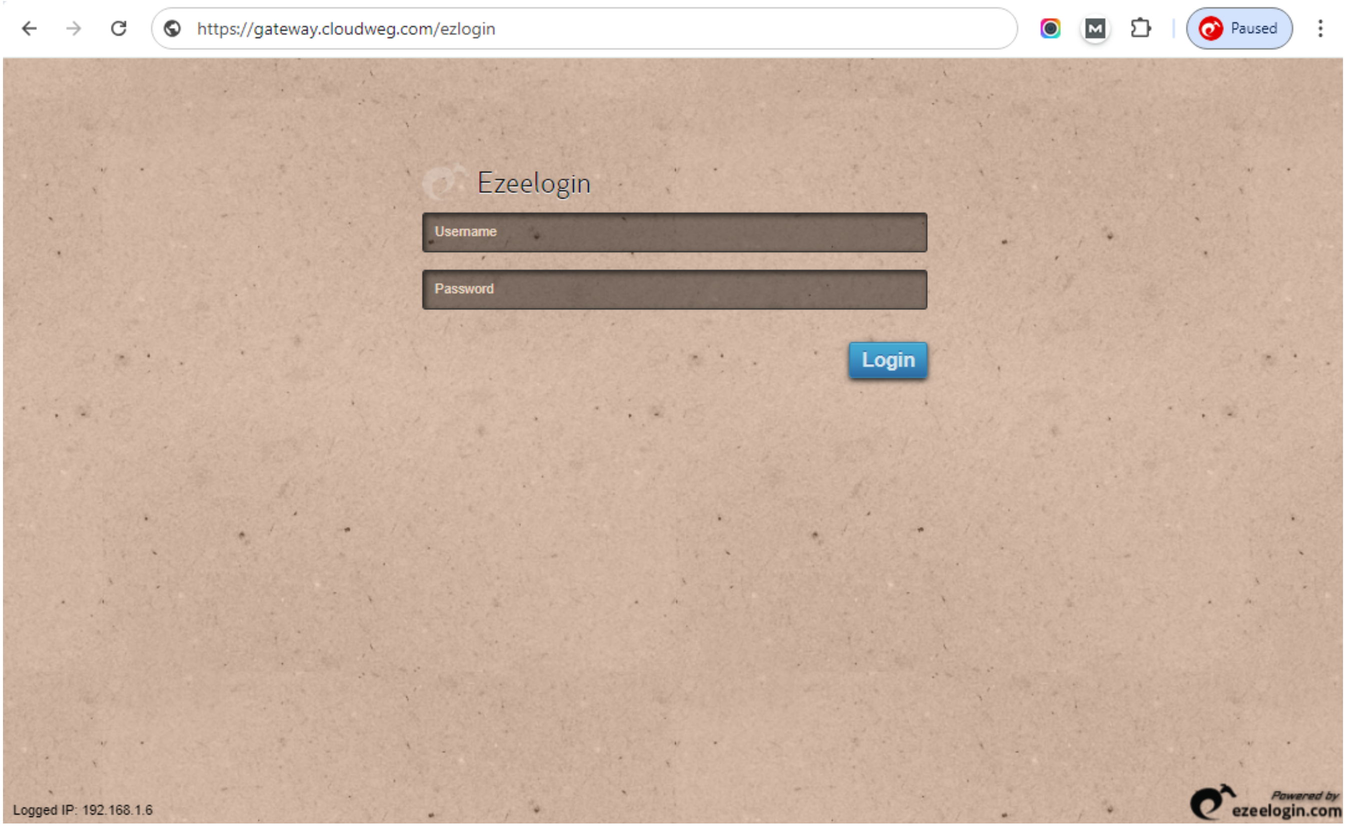
7. Append below lines in **/usr/local/etc/ezlogin/ez.conf**

```
root@gateway ~]# vim /usr/local/etc/ezlogin/ez.conf


host_url  http://gateway.cloudweg.com
cookie_match_ip 0
```

If you have enabled **HTTPS** you should add **host_url https://your_domain_name**

Now you can access Ezeelogin GUI via **Cloudflare Zero Trust Network Access**

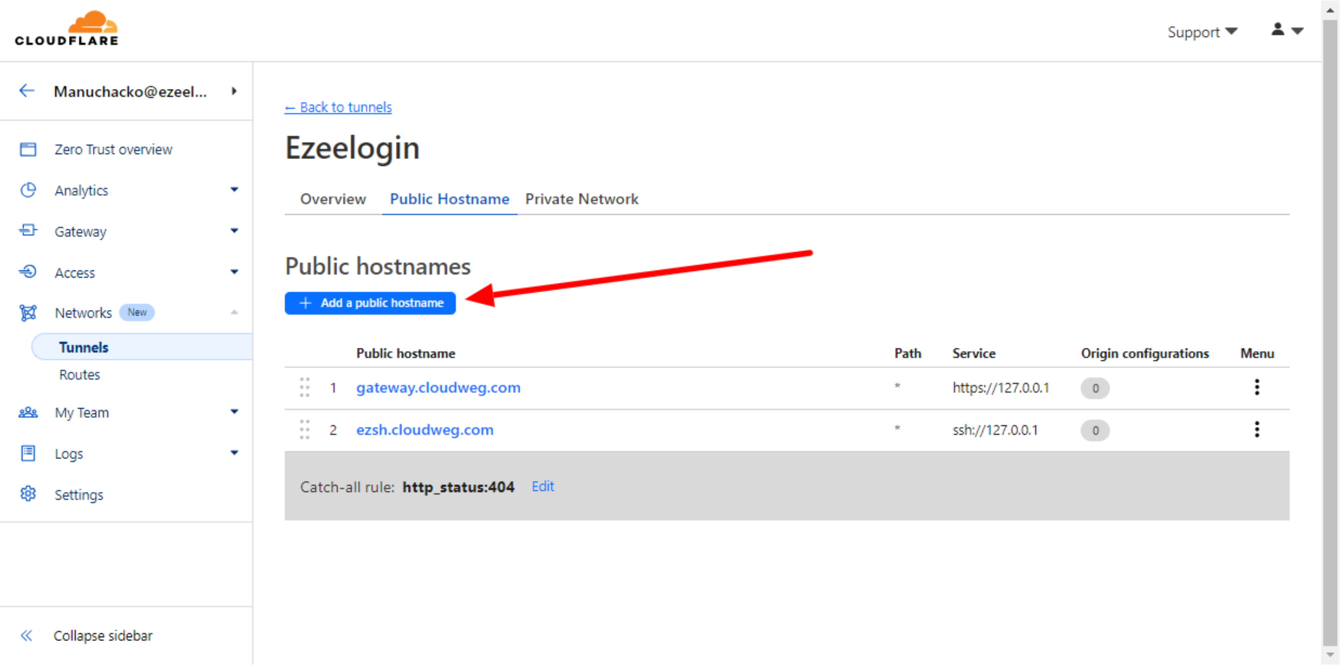
Access Ezeelogin GUI with the **subdomain** created https://gateway.cloudflare.com



https://gateway.cloudweg.com/ezlogin

Accessing Ezeelogin Shell via Cloudflare Zero Trust Network Access

1. Add a **Public Hostnames from dashboard > configure > public hostname > Add a Public Hostnames**



Ezeelogin

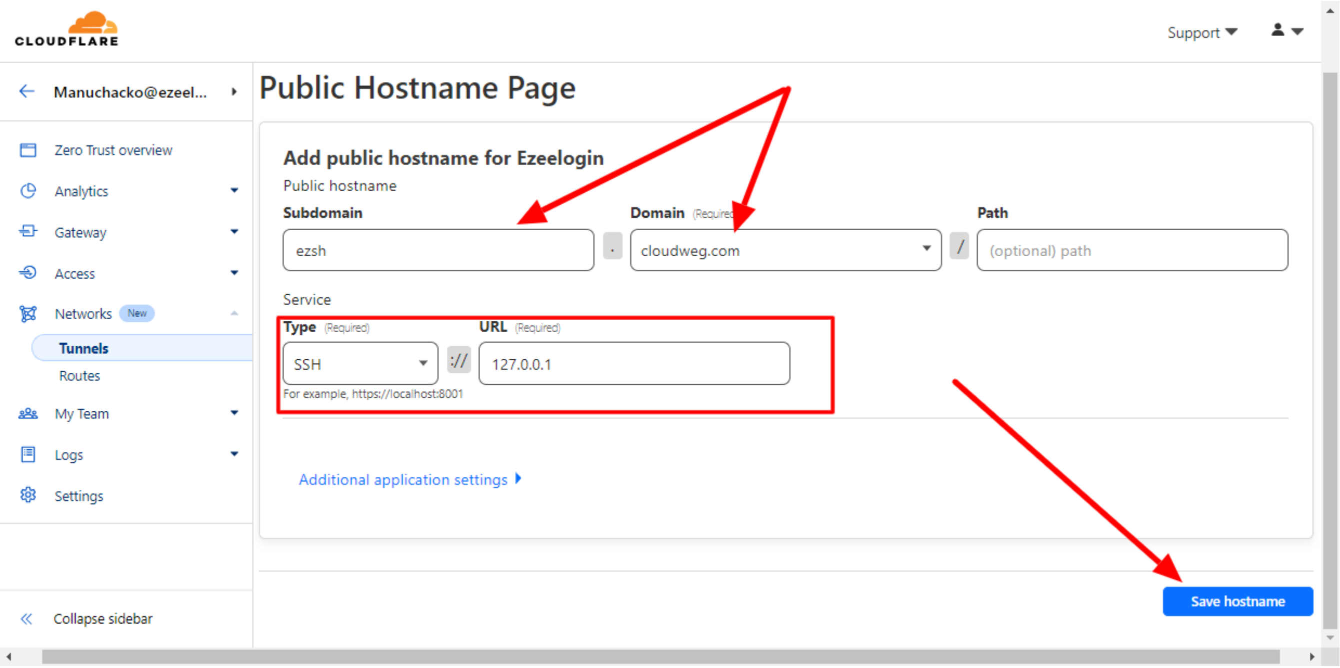
Public hostnames

	Public hostname	Path	Service	Origin configurations	Menu
1	gateway.cloudweg.com	*	https://127.0.0.1	0	⋮
2	ezsh.cloudweg.com	*	ssh://127.0.0.1	0	⋮

Catch-all rule: **http_status:404**

Choose a domain from the drop-down menu and specify any subdomain (for example,ezsh.cloudweg.com).

For Service, select **SSH,** enter **127.0.0.1** and click **save** hostname.



Public Hostname Page

Add public hostname for Ezeelogin

Public hostname

Subdomain: ezsh

Domain (Required): cloudweg.com

Path: (optional) path

Service

Type (Required): SSH

URL (Required): 127.0.0.1

For example, https://localhost:8001

??2. Connect as a Ezeelogin user from your Terminal via cloudflare Zero Trust Network Access

1. [redacted] Cloudflared on the client machine.

2. Add the following to your SSH configuration file on your machine **~/.ssh/config**:

Input the following values; replacing **ezsh.cloudweg.com** with the hostname you created. Edit **vim ~/.ssh/config**

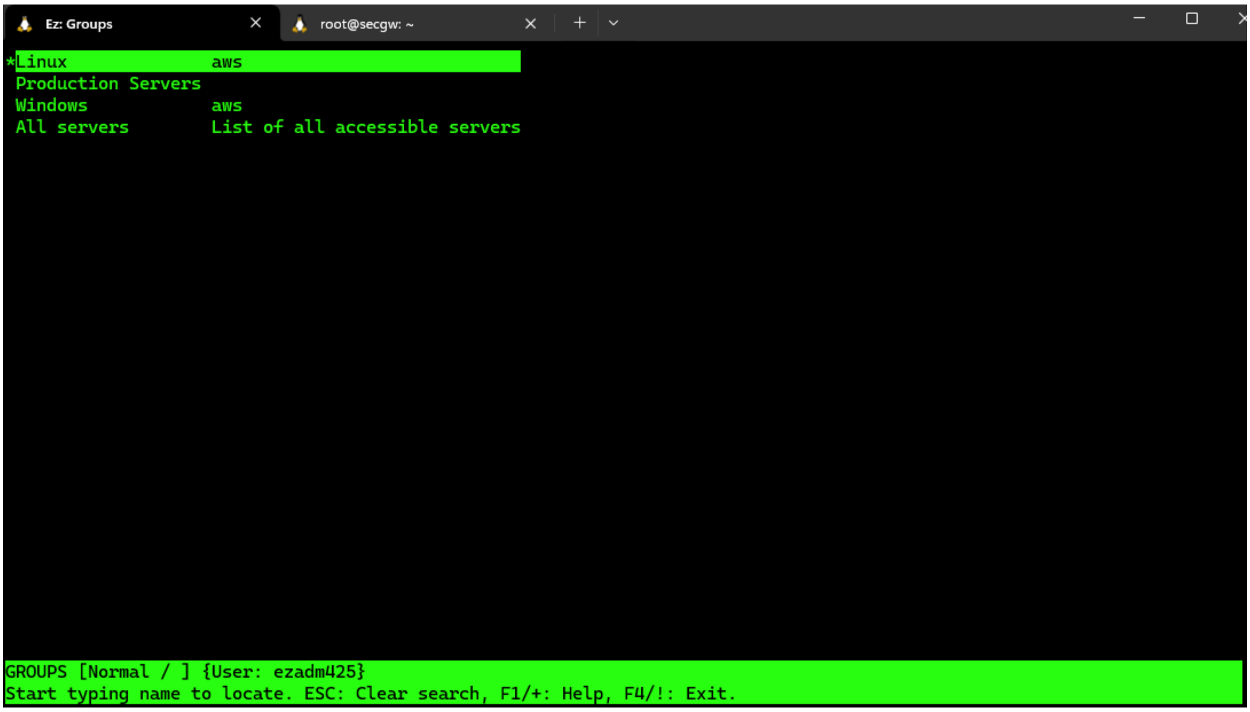
```
Host ezsh.cloudweg.com
```

```
ProxyCommand /usr/local/bin/cloudflared access ssh
--hostname %h
```

The cloudflared path may be different depending on your OS and package manager. For example, if you installed cloudflared on macOS with Homebrew, the path is /opt/homebrew/bin/cloudflared

3. You can now test the connection by running a command to acess Ezeelogin shell via **zero trust network access(ZTNA)**.
Replace **ezsh.cloudweg.com** with the hostname you created.

```
ssh <ezeelogin_username>@ezsh.cloudweg.com
```



```
*Linux                  aws
 Production Servers
 Windows                aws
 All servers            List of all accessible servers

GROUPS [Normal / ] {User: ezadm425}
Start typing name to locate. ESC: Clear search, F1/+: Help, F4/!: Exit.
```

Online URL:
https://www.ezeelogin.com/kb/article/cloudflare-ztna-for-securing-ssh-access-to-endpoint-devices-720.html