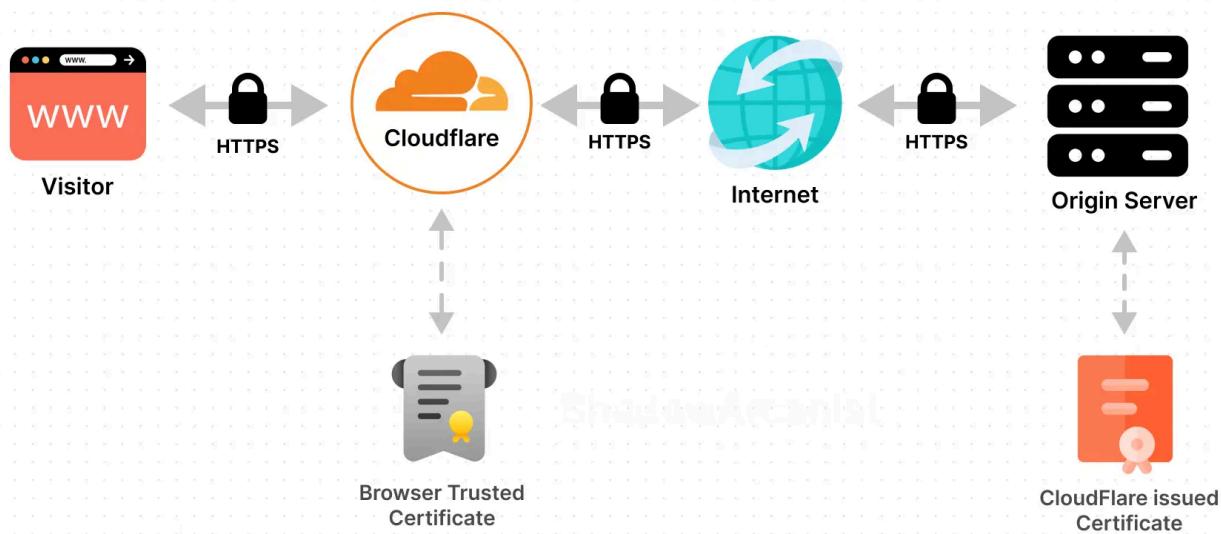


[≡ On this page](#) > Cloudflare Origin Certificate

Cloudflare Origin Certificate

CloudFlare Encryption all the way to the origin



The Cloudflare Origin Certificate ensures secure communication between your server and Cloudflare when using Cloudflare's Proxy, CDN, and security features.

It encrypts the data exchanged between your server and Cloudflare, keeping it safe.

Why Use Cloudflare Origin Certificate with Coolify?

1. No need for HTTP or DNS challenges to create certificates.
2. Keep port 80 closed — everything happens securely over TLS.
3. Longer certificate validity (15 years) — once set up, you don't need to worry about renewing every few months.

When to Avoid Using Cloudflare Origin Certificate



access your server or apps.

Example Data

The following data is used as an example in this guide. Please replace it with your actual data when following the steps:

- IPv4 Address of Origin Server: 203.0.113.1
- Domain Name: shadowarcanist.com
- Username: shadowarcanist

1. Create the Origin Certificate

To create your Cloudflare Origin Certificate, follow these steps:

The screenshot shows the Cloudflare dashboard for the domain `shadowarcanist.com`. The left sidebar has a tree view with sections like Overview, AI Audit, Analytics & Logs, DNS, Email, and SSL/TLS. The SSL/TLS section is expanded, and the 'Origin Server' option under it is selected. A large arrow labeled '1' points to this 'Origin Server' link. Another arrow labeled '2' points to the 'SSL/TLS' link in the main navigation bar at the top. A third arrow labeled '3' points to the blue 'Create Certificate' button in the main content area.

SSL/TLS
Origin Server

Customize encryption of traffic between your origin server and Cloudflare.
[Origin server SSL/TLS documentation](#)

Origin Certificates
 Generate a free TLS certificate signed by Cloudflare to install on your origin server.
 Origin Certificates are only valid for encryption between Cloudflare and your origin server.

Hosts	Expires On
*.shadowarcanist.com, shadowarcanist.com (2 hosts)	Nov 13, 2039

[Download](#) [Revoke](#)

Authenticated Origin Pulls
 TLS client certificate presented for authentication on origin pull.
 Configure expiration notification for your certificates [here](#).

This setting was last changed a month ago



2. Select Origin Server.
3. Click the Create Certificate button.

You'll be asked to choose a private key type, hostnames, and certificate validity.

SSL/TLS

Origin Server

Customize encryption of traffic between your origin server and Cloudflare.

[Origin server SSL/TLS documentation](#)

[← Back](#)

Origin Certificate Installation

Follow the steps below to install a certificate on your origin server.

The first step in generating a certificate for your origin is creating a private key and a Certificate Signing Request (CSR). You can provide your own CSR or we can generate a key and CSR using your web browser.

Generate private key and CSR with Cloudflare

Private key type

RSA (2048) ◀ 1

Use my private key and CSR

List the hostnames (including wildcards) on your origin that the certificate should protect. By default your origin certificate covers the apex of your domain (`example.com`) and a wildcard (`*.example.com`). If there are others you wish to add, e.g., those not covered by the wildcard such as `one.two.example.com`, you can add them below.

Hostnames

*.shadowarcanist.com ◀ 2 shadowarcanist.com

Choose how long before your certificate expires. By default your certificate will be valid for fifteen (15) years. If you'd like to decrease how long your certificate will be valid make a selection below.

Certificate Validity

15 years ◀ 3

1. Choose RSA (2048) for the key type.
2. Add the hostnames you want the certificate to cover.

Note

- `shadowarcanist.com` will cover only the main domain.
- `*.shadowarcanist.com` will cover all subdomains.

On Cloudflare's free plan, wildcard certificates cover just one level of subdomains

For example, it works for `coolify.shadowarcanist.com` but not `www.coolify.shadowarcanist.com`.

To cover multiple levels, you'll need to purchase the [Advanced Certificate Manager](#)

3. Set the certificate validity to 15 years.



CLOUDFLARE

ShadowArcanist shadowarcanist.com Active Starred Free plan

SSL/TLS

Origin Server

Customize encryption of traffic between your origin server and Cloudflare.

[Origin server SSL/TLS documentation](#)

Origin Certificate Installation

Save the certificate and private key below to your client. To save, **Click to copy** and paste the contents into different files on your client, e.g. example.com.pem and example.com.key

Key Format ①
PEM

-----BEGIN CERTIFICATE-----
MIIEsdCCAS5igAwIBAgIU02EbCpLrUd0pIEzUscYK2jA9+7swDQYJKoZIhvNAQEL
BQAwgYsxCzAJBgNVBAYTA1VTMRkwFwYDVQQKExbDbg91ZEZsYXJ1LCBjbhMuMTQw
MgYDVQQLEytDbg91ZEZsYXJ1IEyawdpibTU0bg02VydGImadWNhdGUgQXV0aG9y
aRSRMRYwFAYDVQHWE1YW4gRnJhbhNpc2NvMRMwEQYDVQQIEwpDYWxpZm9ybmlh

② Click to copy

Private Key ③
Copy the contents of your private key below to your web server and set file permissions such that only your http server can access it. Additionally, you can optionally encrypt this file and provide a password to decrypt it during your origin web server startup. The private key data will not be stored at Cloudflare and will no longer be accessible once the creation is complete. Please make sure you have a local copy of this key.

-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAscBkgwgSkAgEAAoIBAQDq5ktF0U2DZCrk
F3AOYnUgeTwZd35ov/w4xvK3pDPSGMltvhkG6ESuz7f977Wa+6M3CgHmHvET0
ZQ/93618+2m9kbPZ1FuFk+Ex5nV3vc13pB1195WgnE84+EdFNQaKzAAIf0hFDc
vd1kfFF0zt+00rnunggsREhWoZdEU6lFsv1pZ3YKb1ITGzbHVhDuZRbQ1NXHdw

④ Click to copy

1. Choose PEM as the key format.
2. Copy your Certificate.
3. Copy your Private Key.

Next, you'll add these to your server running Coolify and configure Coolify to use this certificate.

2. Add Certificate to Your Server

SSH into your server or use Coolify's terminal feature. For this guide, I'm using SSH:

```
ssh shadowarcanist@203.0.113.1
```

Once logged in, navigate to the Coolify proxy directory:

```
$ cd /data/coolify/proxy
```



Envix

```
$ ls  
acme.json  docker-compose.yml  dynamic
```

If there's no `certs` folder, create it:

```
$ mkdir certs
```

Verify it was created:

```
$ ls  
acme.json  certs  docker-compose.yml  dynamic
```

Now, navigate into the `certs` directory:

```
$ cd certs
```

Create two new files for the certificate and private key:

```
$ touch shadowarcanist.cert shadowarcanist.key
```

Verify the files were created:

```
$ ls  
shadowarcanist.cert  shadowarcanist.key
```

Open the `shadowarcanist.cert` file and paste the certificate from the Cloudflare dashboard:

```
$ nano shadowarcanist.cert
```

Save and exit after pasting the certificate.

Do the same for the `shadowarcanist.key` file and paste the private key:



Save and exit.

Now the origin certificate is installed on your server.

> Commands Used on Server (click to view)

3. Set Up DNS Records and TLS Encryption

To make the origin certificate work, configure your DNS records, enable TLS, and set up HTTP to HTTPS redirects in Cloudflare:

Type	Name	Content	Proxy status	TTL
A	*	203.0.113.1	Proxied	Auto
A	shadowarcanist.com	203.0.113.1	Proxied	Auto
CNAME	email	email.example.com	DNS only	1 hr
CNAME	status	reports.example.com	DNS only	Auto
MX	shadowarcanist.com	mailstore1.example.com	DNS only	1 hr
MX	shadowarcanist.com	smtp.example.com	DNS only	1 hr

1. In Cloudflare, go to DNS.

2. Select Records.

3. Add 2 A records:



- Set the proxy status to **Proxied** for better security.

Next, set up TLS encryption:

The screenshot shows the Cloudflare dashboard for the domain "shadowarcanist.com". The left sidebar has a "SSL/TLS" section with a dropdown menu. Step 1 points to the "SSL/TLS" option in the dropdown. Step 2 points to the "Overview" link under the "SSL/TLS" section. Step 3 points to the "Configure" button in the main content area. The main content area displays the "SSL/TLS Overview" page, which includes a diagram showing traffic flow from a "Browser" through "Cloudflare" to an "Origin Server", and a section titled "Traffic Served Over TLS" showing statistics for different TLS versions.

TLS Version	Count
None (not secure)	617
TLS v1.2	533
TLS v1.3	25.1k

- Go to **SSL/TLS** in Cloudflare.
- Select **Overview**.
- Click **Configure** button

Choose **Full (Strict)** as the encryption mode.



Overview

AI Audit Beta

Analytics & Logs

DNS

Email

SSL/TLS

- Overview
- Edge Certificates
- Client Certificates
- Origin Server
- Custom Hostnames

Security

Access

Speed

Caching

Workers Routes

Rules

Network

Traffic

Custom SSL/TLS

Select the encryption mode that Cloudflare uses to connect to your origin server

Selected

Strict (SSL-Only Origin Pull)

Enforce encryption between Cloudflare and your origin. Use this mode to guarantee connections to your origin will always be encrypted, regardless of your visitor's request.

Full (Strict)

Enable encryption end-to-end and enforce validation on origin certificates. Use Cloudflare's Origin CA to generate certificates for your origin.

Full

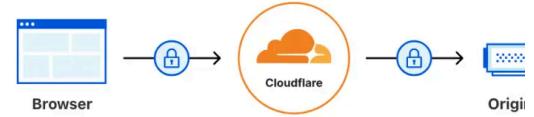
Enable encryption end-to-end. Use this mode when your origin server supports SSL certification but does not use a valid, publicly trusted certificate.

Flexible

Enable encryption only between your visitors and Cloudflare. This will avoid browser security warnings, but all connections between Cloudflare and your origin are made through HTTP.

Off (not secure)

No encryption applied. Turning off SSL disables HTTPS and causes browsers to show a warning that your website is not secure.



Finally, enable HTTP to HTTPS redirects:

ShadowArcanist shadowarcanist.com Active Starred Free plan

SSL/TLS

1. In Cloudflare, go to SSL/TLS
2. Select Edge Certificates

Total TLS

Cloudflare will issue individual certificates for every proxied hostname. Once enabled, all hostnames in your domain will be covered by a TLS certificate.

To enable Total TLS, purchase Advanced Certificate Manager.

Always Use HTTPS

Redirect all requests with scheme "http" to "https". This applies to all http requests to the zone.

This setting was last changed 2 months ago

HTTP Strict Transport Security (HSTS)

Enforce web security policy for your website.

This setting was last changed 5 days ago

3

Enable HSTS

API ▶ Help ▶ Ch

1. In Cloudflare, go to SSL/TLS

2. Select Edge Certificates.



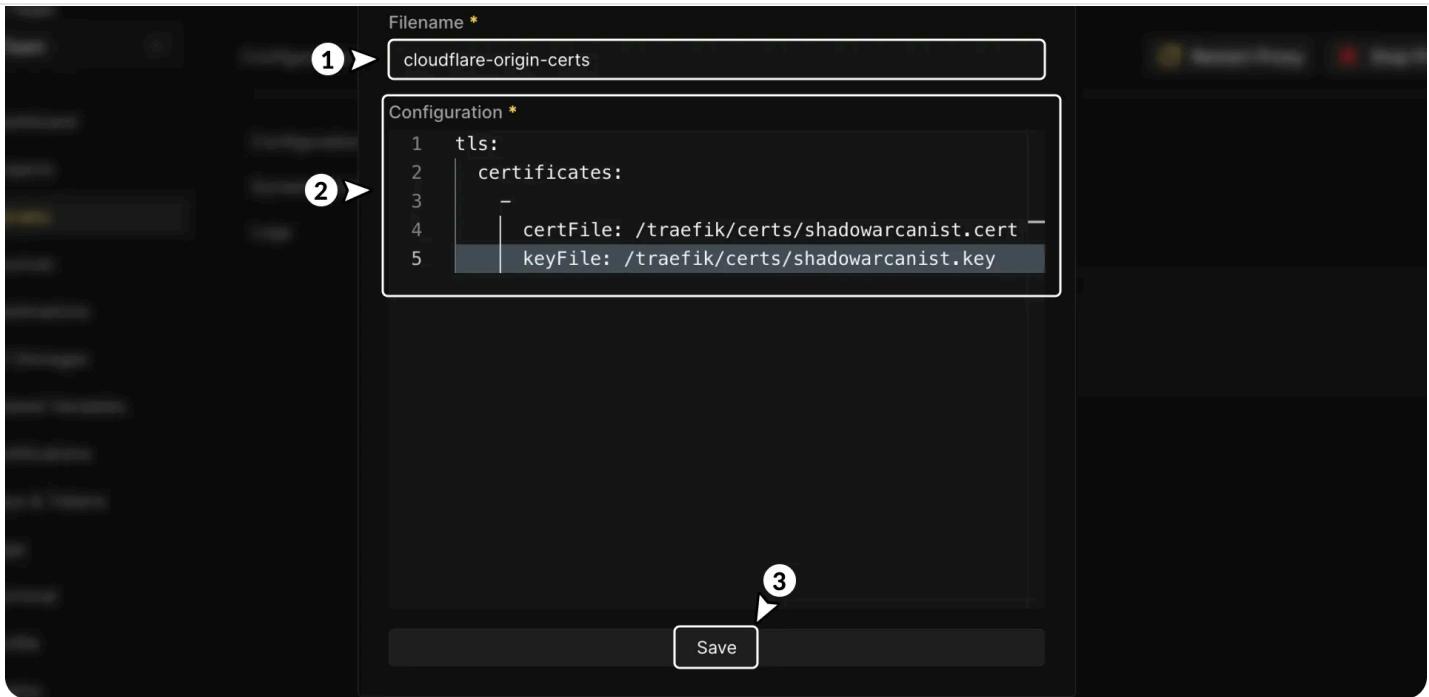
4. Configure Coolify to Use the Origin Certificate

Now, in your Coolify dashboard:

The screenshot shows the Coolify dashboard interface. On the left, there's a sidebar with various options like Dashboard, Projects, Servers (which is highlighted with a red box and a circled '1'), Sources, Destinations, S3 Storages, Shared Variables, Notifications, Keys & Tokens, Tags, and Terminal. The main area is titled 'Server' with a status indicator 'Proxy Running'. It has tabs for Configuration, Proxy (which is highlighted with a red box and a circled '2'), Resources, and Terminal. Under the Proxy tab, there's a 'Dynamic Configurations' section with a '+ Add' button (circled '3'). Below it is a 'File: Caddyfile' section containing the code: import /dynamic/*.caddy. Arrows numbered 1 through 4 point to the highlighted areas: 1 points to the 'Servers' sidebar item, 2 points to the 'Proxy' tab, 3 points to the '+ Add' button in the 'Dynamic Configurations' section, and 4 points to the 'File: Caddyfile' code area.

1. Go to the **Server** section in the sidebar.
2. Select **Proxy**.
3. Open the **Dynamic Configuration** page
4. Click **Add** button

You will now be prompted to enter the Dynamic Configuration.



1. Choose a name for your configuration.
2. Enter the following details in the configuration field:

```
tls:  
  certificates:  
    -  
      certFile: /traefik/certs/shadowarcanist.cert  
      keyFile: /traefik/certs/shadowarcanist.key
```

› [Adding Multiple Certificates \(click to view\)](#)

3. Save the configuration

From now on, Coolify will use the origin certificate for requests matching the hostname.

i All the steps below are completely optional

5. Optional: Configure Traefik



Since you're using an Origin Certificate, you no longer need HTTP challenges or port 80 open.

The screenshot shows the Coolify dashboard interface. On the left, there's a sidebar with various options like Dashboard, Projects, Servers (which is highlighted with a yellow box and has a circled '1' above it), Sources, Destinations, S3 Storages, Shared Variables, Notifications, Keys & Tokens, Tags, Terminal, Profile, Teams, and Settings. At the top, it says "Server" with a green "Proxy Running" status indicator, and there are tabs for Configuration, Proxy (which is highlighted with a yellow box and has a circled '2' above it), Resources, and Terminal. In the main area, there's a "Configuration" tab with a "Save" button and a warning message about switching proxies. Below that is an "Advanced" section with a "Generate labels only for Traefik" toggle. Under "Traefik", there's a "Default Redirect 404" field containing "https://app.coolify.io". A large callout box labeled '4' contains the following configuration file:

```

Configuration file
1 networks:
2   coolify:
3     | external: true
4
5 services:
6   traefik:
7     | container_name: coolify-proxy
8     | image: 'traefik:v3.1'
9     | restart: unless-stopped
10    | extra_hosts:
11      | - 'host.docker.internal:host-gateway'
12    networks:

```

1. Go Server in the Coolify dashboard.
2. Select Proxy.
3. Open Configuration.
4. Replace the configuration with this:

```

# Define external networks
networks:
  coolify:
    external: true # External network.

services:
  # Traefik reverse proxy
  traefik:
    container_name: coolify-proxy # Container name.
    image: 'traefik:v3.1' # Traefik image version.
    restart: unless-stopped # Auto-restart policy.
    extra_hosts:
      - 'host.docker.internal:host-gateway' # Host communication.
  networks:
    - coolify # Network connection.

```



Envix

```

test: 'wget -qO- http://localhost:80/ping || exit 1' # Ping endpoint for health check.
interval: 4s # Health check interval.
timeout: 2s # Health check timeout.
retries: 5 # Retry attempts.
volumes:
- '/var/run/docker.sock:/var/run/docker.sock:ro' # Docker socket access.
- '/data/coolify/proxy:/traefik' # Traefik config volume.
command:
# Traefik configuration options
- '--ping=true' # Enable ping for health check.
- '--ping.entrypoint=http' # Use HTTP entrypoint for ping.
- '--entrypoints.http.address=:80' # HTTP entry point for health checks.
- '--entrypoints.https.address=:443' # HTTPS entry point.

```

Note

The comments in this configuration explain each line. You can remove the comments when copying it into your configuration.

Next, you'll need to remove a few labels from your Dockerfile-based deployments. Below is an example of how I set this up for my website.

The screenshot shows the Coolify web interface with the following steps highlighted:

- Step 1: Click on the **Projects** button in the sidebar.
- Step 2: Click on the **Configuration** tab in the top navigation bar.
- Step 3: Click on the **General** tab in the left sidebar.
- Step 4: Click on the **Network** tab in the right sidebar.
- Step 5: Click on the **Container Labels** section in the Network tab.

In the Container Labels section, the following labels are listed:

```

1 # Enable Traefik
2 traefik.enable=true
3
4 # HTTPS Router Configuration
5 traefik.http.routers.shadowarcanist.entryPoints=https
6 traefik.http.routers.shadowarcanist.rule=Host(`shadowarcanist.
com`) && PathPrefix(``)
7 traefik.http.routers.shadowarcanist.service=shadowarcanist
8 traefik.http.routers.shadowarcanist.tls=true
9 traefik.http.services.shadowarcanist.loadbalancer.server.port=80
10

```

1. Go to **Projects** and select your project.



4. Check **Readonly labels** option
5. Replace the labels with the following:

```
# Enable Traefik for this configuration
traefik.enable=true

# Define the entry point for the router (HTTPS)
traefik.http.routers.shadowarcanist.entryPoints=https

# Set the routing rule for this router to match the domain "shadowarcanist.com" and any path
traefik.http.routers.shadowarcanist.rule=Host(`shadowarcanist.com`) && PathPrefix(`/`)

# Assign the service 'shadowarcanist' to this router
traefik.http.routers.shadowarcanist.service=shadowarcanist

# Enable TLS (HTTPS) for this router
traefik.http.routers.shadowarcanist.tls=true

# Specify the backend service and its port (port 80)
traefik.http.services.shadowarcanist.loadbalancer.server.port=80
```

Now you're done! Your server is set up to use the Cloudflare Origin Certificate, and all traffic is secured via TLS.

Note

If this guide doesn't resolve your issue, please join the Coolify Discord server and seek assistance there.

Links



[Coolify Support Server](#)



[Author's Website](#)



The header image is designed using icons from [Flaticon](#). Links to each icon can be found below:

- [Medal icon](#) by [Vlad Szirka](#)
- [Award icon](#) by [explanacion](#)
- [Worldwide icon](#) by [Freepik](#)
- [Lock icon](#) by [Those Icons](#)
- [Browser icon](#) by [Alfredo Hernandez](#)
- [Database icon](#) by [Tanah Basah](#)