ACME CERTIFICATE MANAGEMENT

Paul van Brouwershaven

8 June 2022

Server Certificate Working Group
CA/Browser Forum F2F #59 - Redmond, WA, USA

ENTRUST

SECURING A WORLD IN MOTION

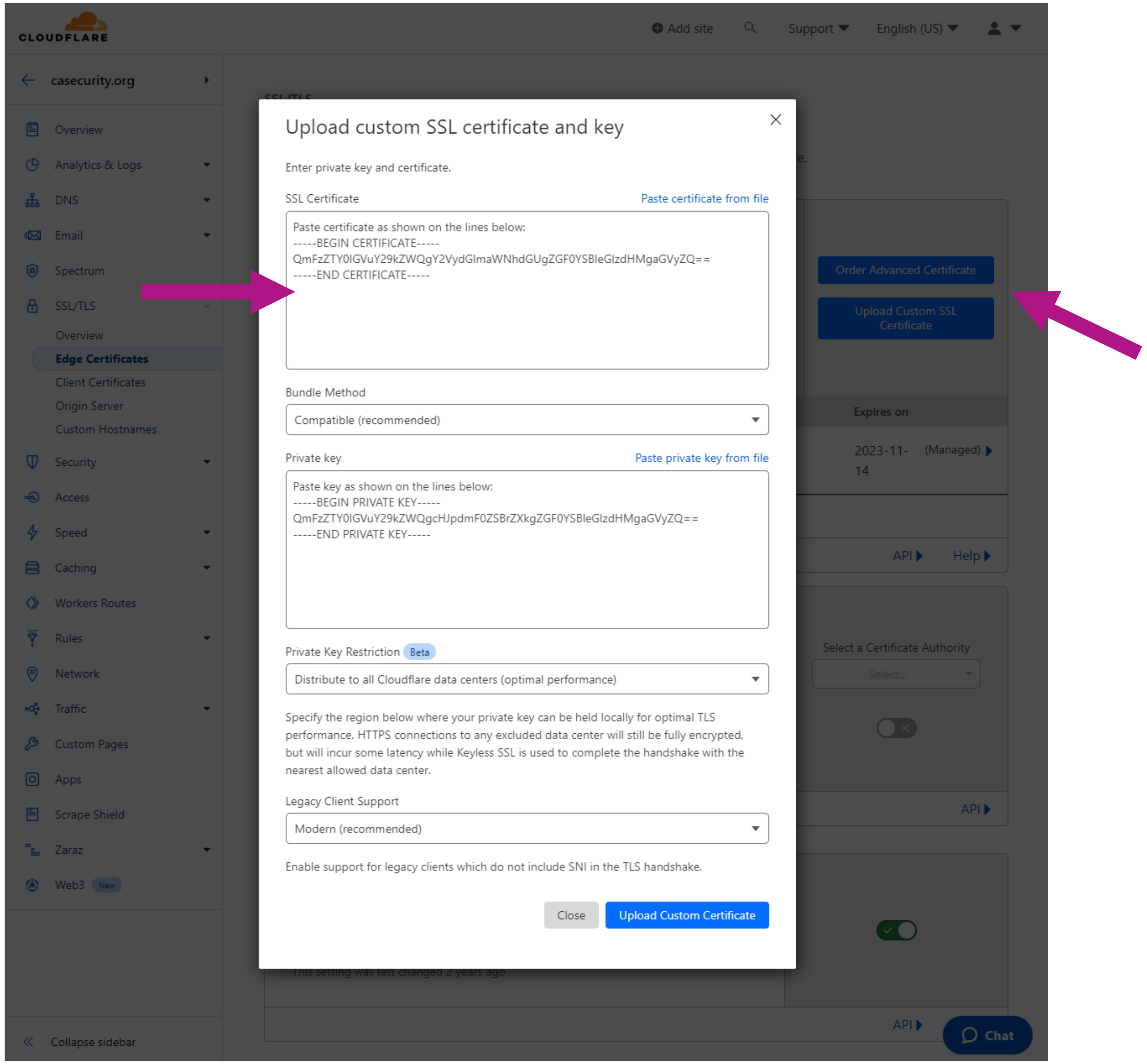
ACME CERTIFICATE MANAGEMENT

How do we deploy custom certificates?

We reviewed the certificate management processes of some well-known online service providers and concluded that the provisioning of custom certificates is a manual process… unless the requestor has the knowledge and budget to develop and maintain a custom integration with a proprietary API for each of the platforms they use.

An alternative to a custom integration is the usage of a Certificate Lifecycle Management (CLM) provider or using a plugin for Ansible, Terraform, etc. (if such integrations are available).

CLOUDFLARE (CDN)



Upload custom SSL certificate and key

Enter private key and certificate.

SSL Certificate

```
Paste certificate as shown on the lines below:
-----BEGIN CERTIFICATE-----
QmFzZTY0IGVuY29kZWQgY2VydGlmaWNhdGUgZGF0YSBleGlzdHMgaGVyZQ==
-----END CERTIFICATE-----
```

Bundle Method: Compatible (recommended)

Private key

```
Paste key as shown on the lines below:
-----BEGIN PRIVATE KEY-----
QmFzZTY0IGVuY29kZWQgcHJpdmF0ZSBrZXkgZGF0YSBleGlzdHMgaGVyZQ==
-----END PRIVATE KEY-----
```

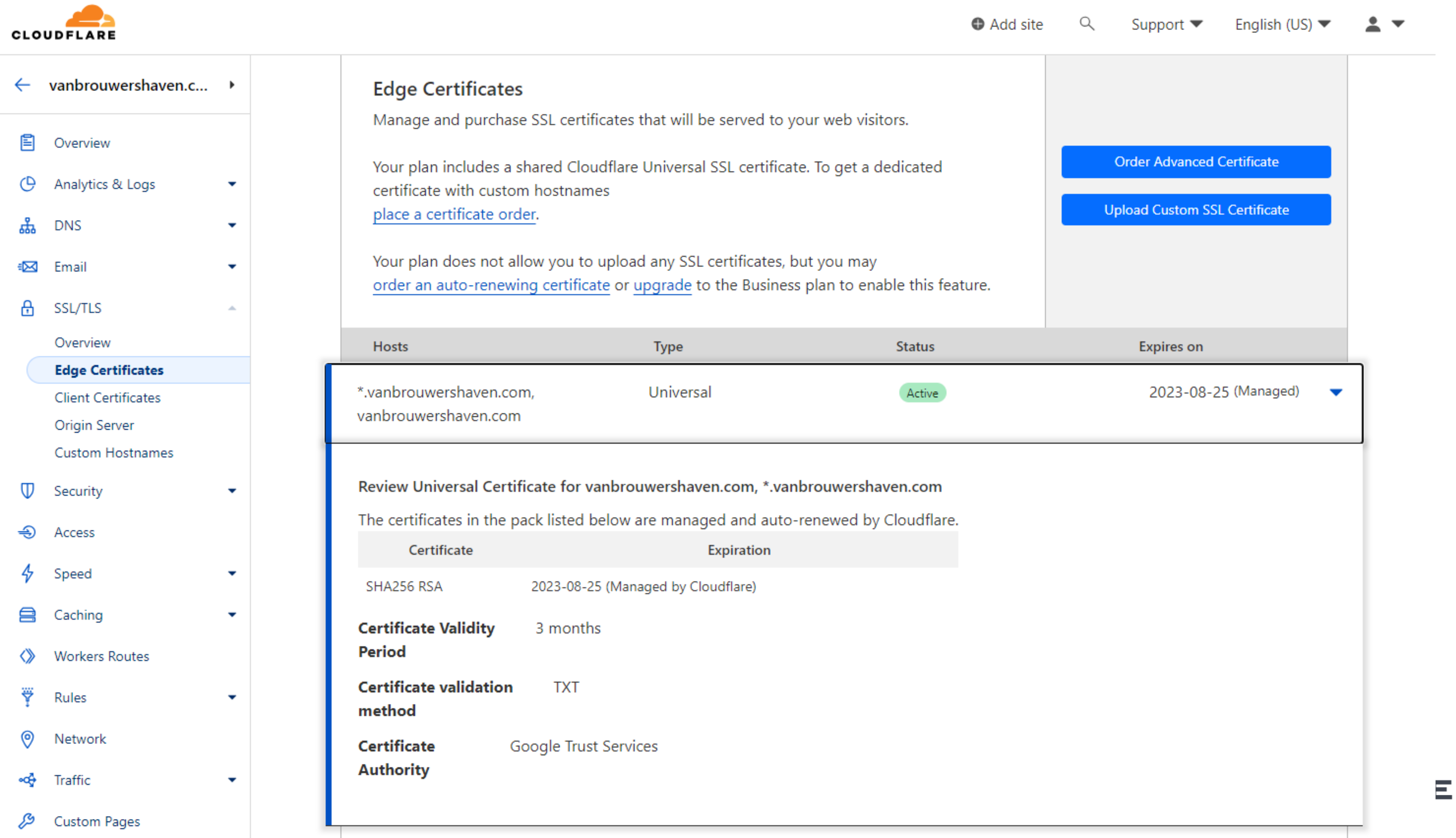
Private Key Restriction Beta: Distribute to all Cloudflare data centers (optimal performance)

Specify the region below where your private key can be held locally for optimal TLS performance. HTTPS connections to any excluded data center will still be fully encrypted, but will incur some latency while Keyless SSL is used to complete the handshake with the nearest allowed data center.

Legacy Client Support: Modern (recommended)

Enable support for legacy clients which do not include SNI in the TLS handshake.

CLOUDFLARE (CDN) – CERTIFICATE OPTIONS



Edge Certificates

Manage and purchase SSL certificates that will be served to your web visitors.

Your plan includes a shared Cloudflare Universal SSL certificate. To get a dedicated certificate with custom hostnames place a certificate order.

Your plan does not allow you to upload any SSL certificates, but you may order an auto-renewing certificate or upgrade to the Business plan to enable this feature.

Hosts	Type	Status	Expires on
*.vanbrouwershaven.com, vanbrouwershaven.com	Universal	Active	2023-08-25 (Managed)

Review Universal Certificate for vanbrouwershaven.com, *.vanbrouwershaven.com

The certificates in the pack listed below are managed and auto-renewed by Cloudflare.

Certificate	Expiration
SHA256 RSA	2023-08-25 (Managed by Cloudflare)

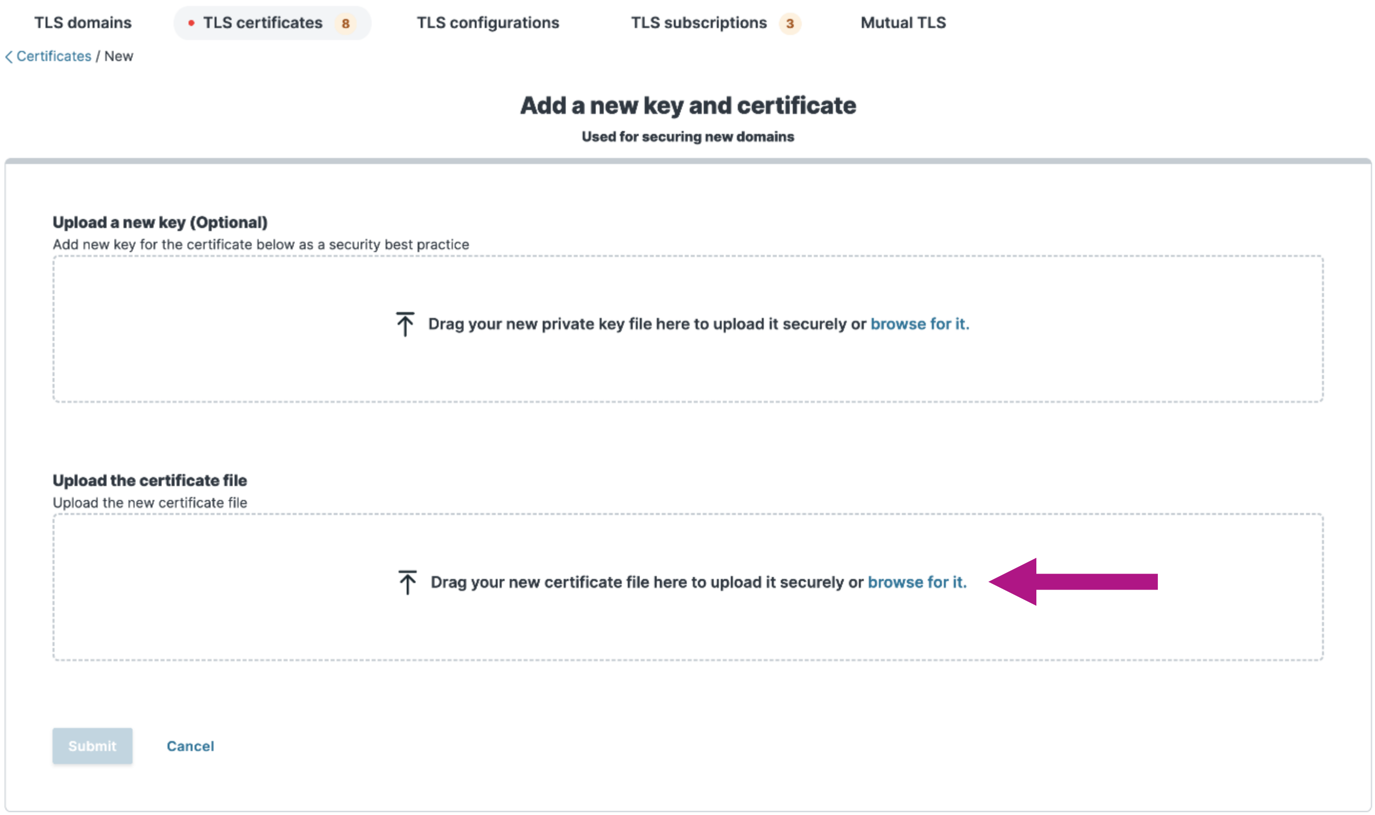
Certificate Validity Period 3 months

Certificate validation method TXT

Certificate Authority Google Trust Services

FASTLY (CDN)

While "*Fastly-managed certificates use the ACME protocol to procure and renew TLS certificates to procure and renew TLS certificates from Let's Encrypt, a non-profit certification authority, and GlobalSign, a commercial certification authority*", they do not allow you to configure your own ACME server and key binding.



Add a new key and certificate

Used for securing new domains

Upload a new key (Optional)

Add new key for the certificate below as a security best practice

Drag your new private key file here to upload it securely or browse for it.

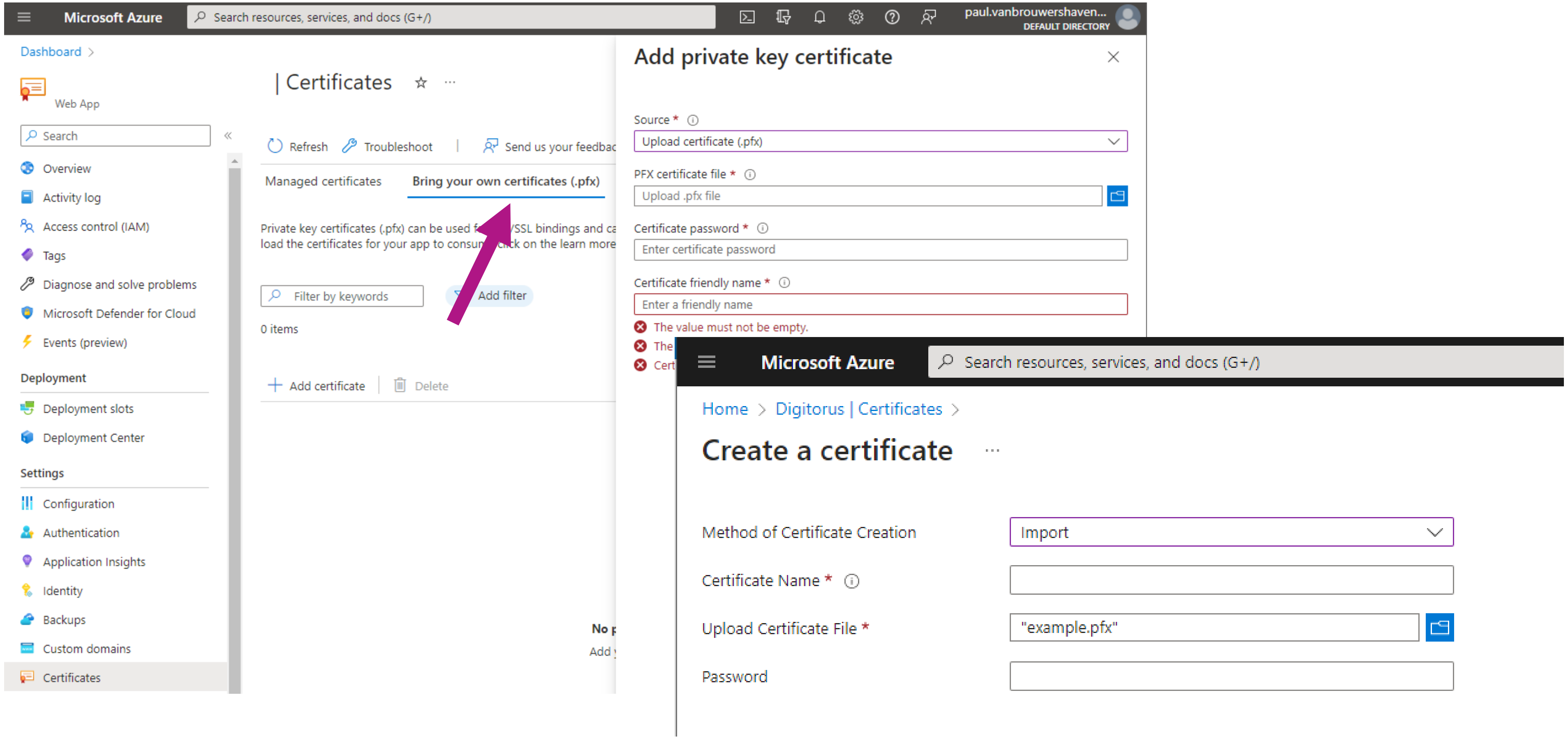
Upload the certificate file

Upload the new certificate file

Drag your new certificate file here to upload it securely or browse for it.

ENTRUST

AZURE (CSP) - KEY VAULT / APP SERVICE



Private key certificates (.pfx) can be used f[...] /SSL bindings and ca load the certificates for your app to consum[...] ck on the learn more

Add private key certificate

Source * Upload certificate (.pfx)

PFX certificate file * Upload .pfx file

Certificate password * Enter certificate password

Certificate friendly name * Enter a friendly name

The value must not be empty.

The

Cert

Create a certificate

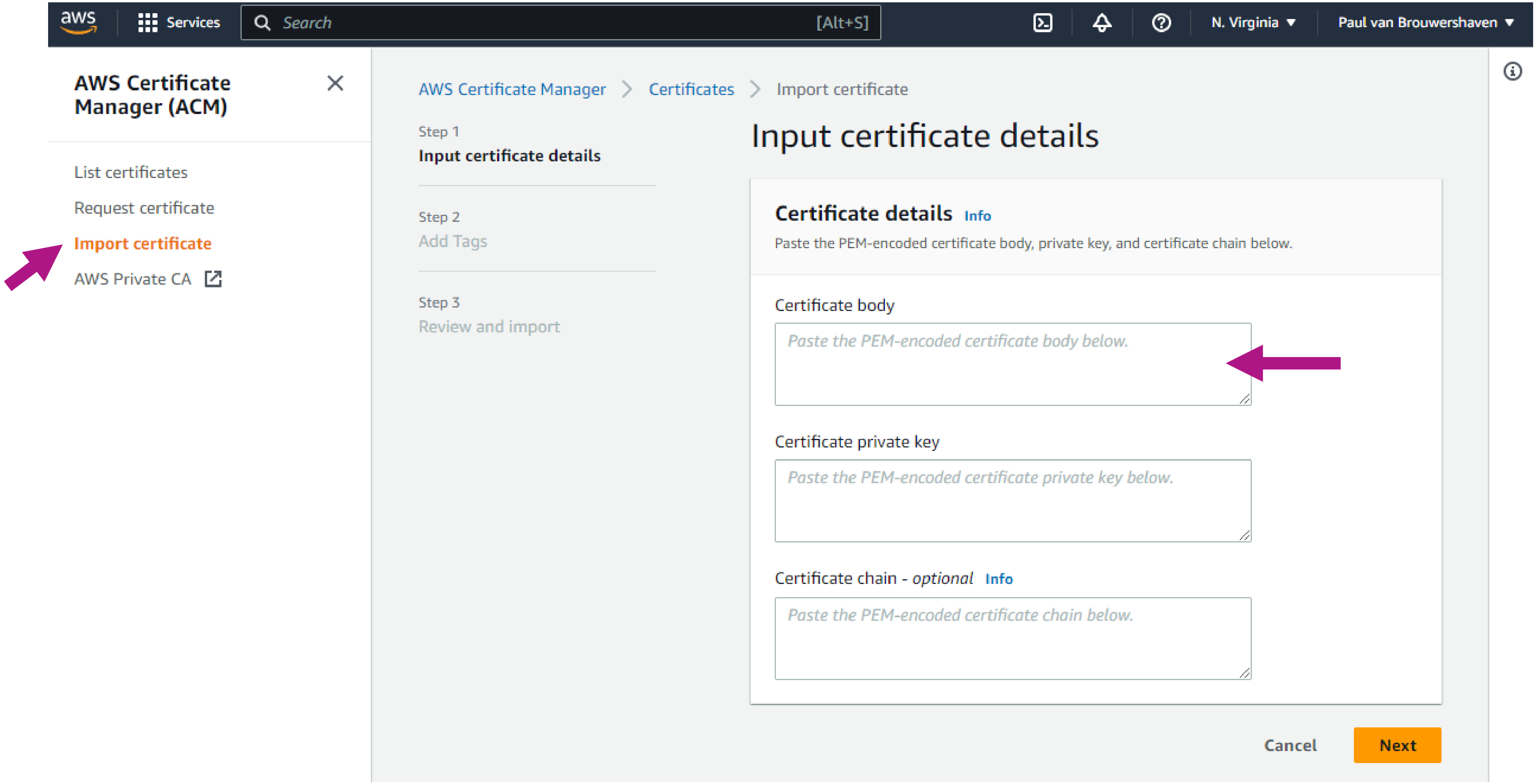
Method of Certificate Creation: Import

Certificate Name *

Upload Certificate File * "example.pfx"

Password

AWS (CSP) - CERTIFICATE MANAGER



AWS Certificate Manager (ACM)

- List certificates
- Request certificate
- Import certificate
- AWS Private CA

AWS Certificate Manager > Certificates > Import certificate

Step 1
Input certificate details

Step 2
Add Tags

Step 3
Review and import

Input certificate details

Certificate details Info

Paste the PEM-encoded certificate body, private key, and certificate chain below.

Certificate body

Paste the PEM-encoded certificate body below.

Certificate private key

Paste the PEM-encoded certificate private key below.

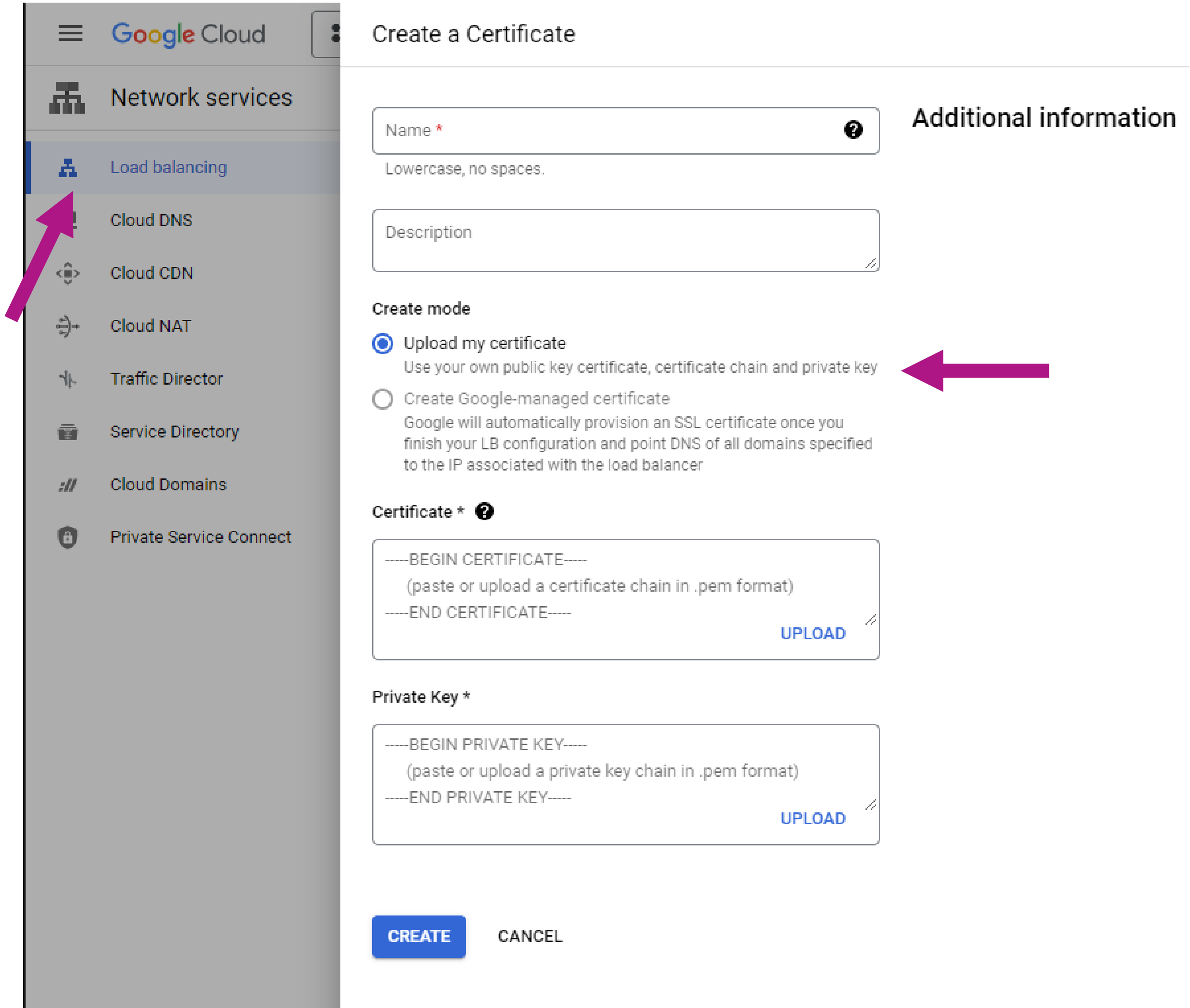
Certificate chain - *optional* Info

Paste the PEM-encoded certificate chain below.

Cancel | Next

ENTRUST

GOOGLE CLOUD (CSP) - LOAD BALANCER



Create a Certificate

Name *

Lowercase, no spaces.

Description

Create mode

● Upload my certificate
Use your own public key certificate, certificate chain and private key

○ Create Google-managed certificate
Google will automatically provision an SSL certificate once you finish your LB configuration and point DNS of all domains specified to the IP associated with the load balancer

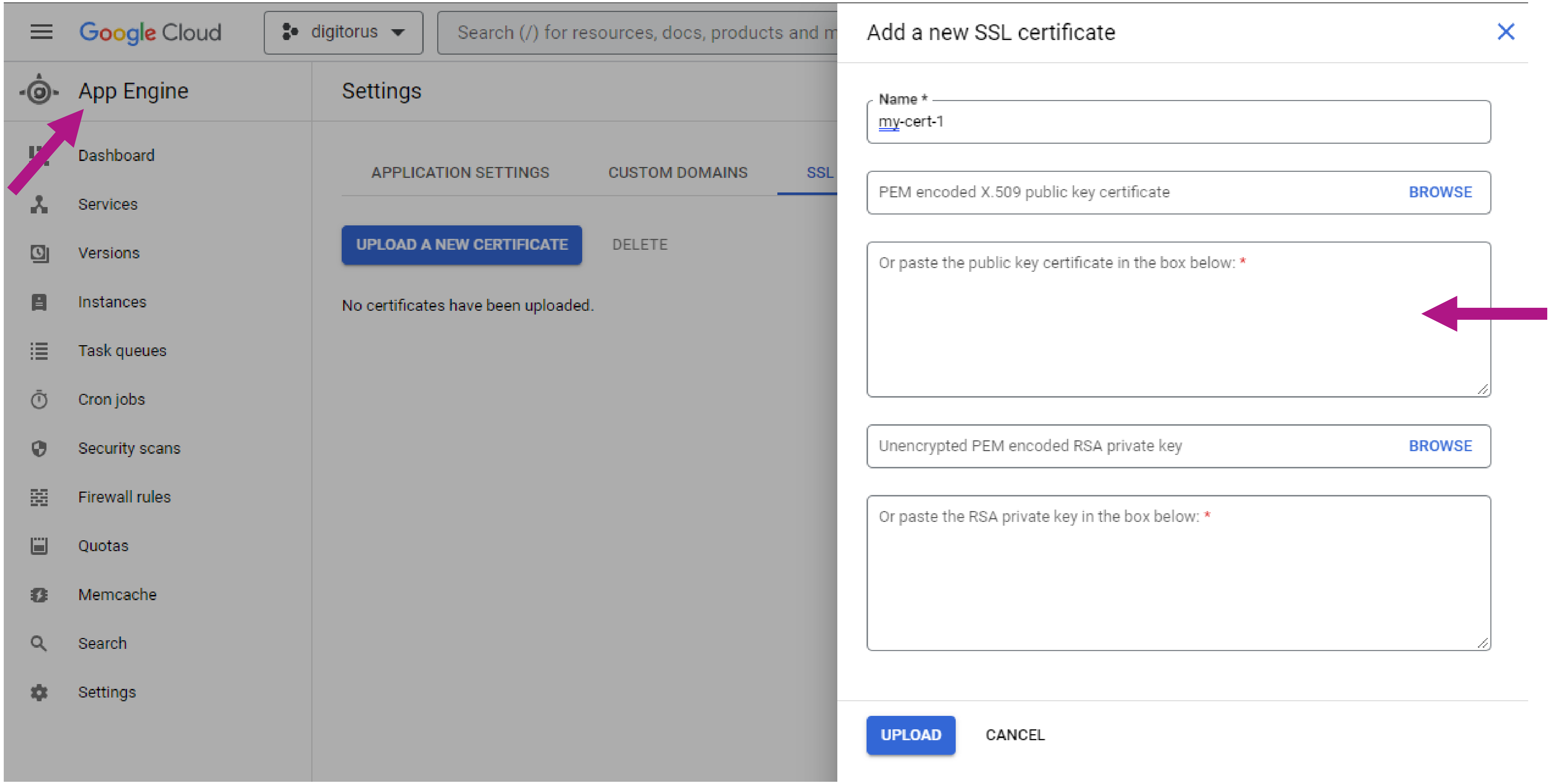
Certificate *

```
-----BEGIN CERTIFICATE-----
    (paste or upload a certificate chain in .pem format)
-----END CERTIFICATE-----
```

Private Key *

```
-----BEGIN PRIVATE KEY-----
    (paste or upload a private key chain in .pem format)
-----END PRIVATE KEY-----
```

GOOGLE CLOUD (CSP) - APP ENGINE



No certificates have been uploaded.

Add a new SSL certificate

Name *
my-cert-1

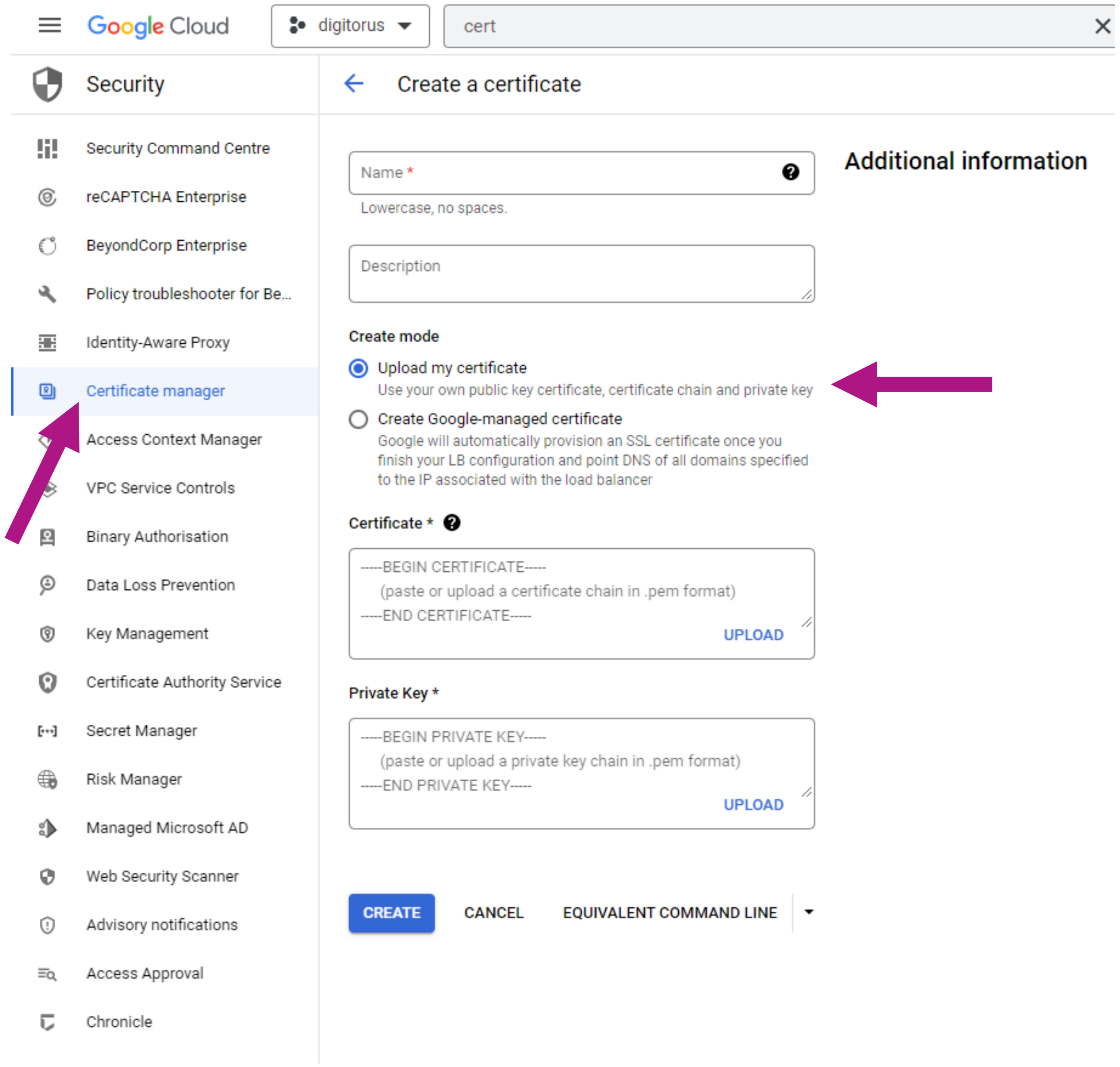
PEM encoded X.509 public key certificate BROWSE

Or paste the public key certificate in the box below: *

Unencrypted PEM encoded RSA private key BROWSE

Or paste the RSA private key in the box below: *

GOOGLE CLOUD (CSP) - CERTIFICATE MANAGER



Create a certificate

Additional information

Name *

Lowercase, no spaces.

Description

Create mode

● Upload my certificate
Use your own public key certificate, certificate chain and private key

○ Create Google-managed certificate
Google will automatically provision an SSL certificate once you finish your LB configuration and point DNS of all domains specified to the IP associated with the load balancer

Certificate *

```
-----BEGIN CERTIFICATE-----
  (paste or upload a certificate chain in .pem format)
-----END CERTIFICATE-----
```

UPLOAD

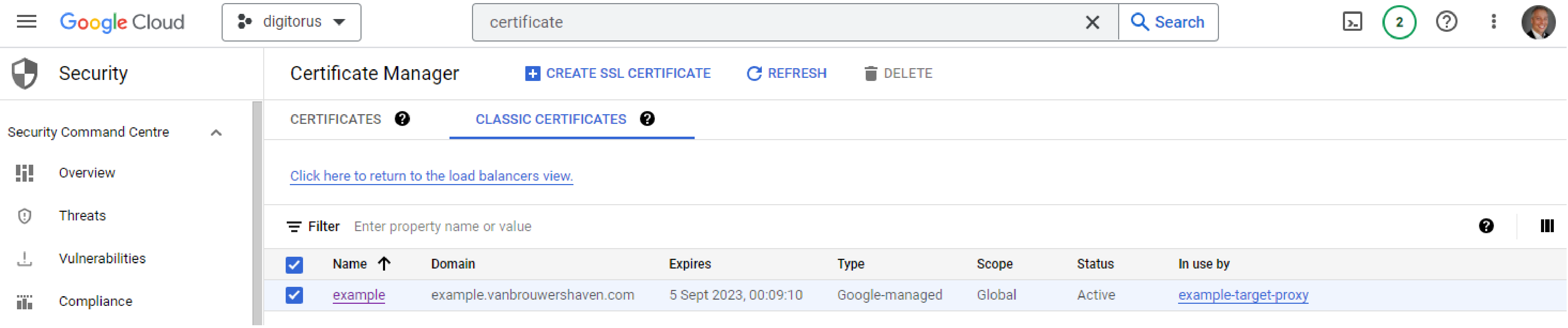
Private Key *

```
-----BEGIN PRIVATE KEY-----
  (paste or upload a private key chain in .pem format)
-----END PRIVATE KEY-----
```

UPLOAD

GOOGLE CLOUD (CSP) - CERTIFICATE MANAGER

No option to revoke the certificate in the overview

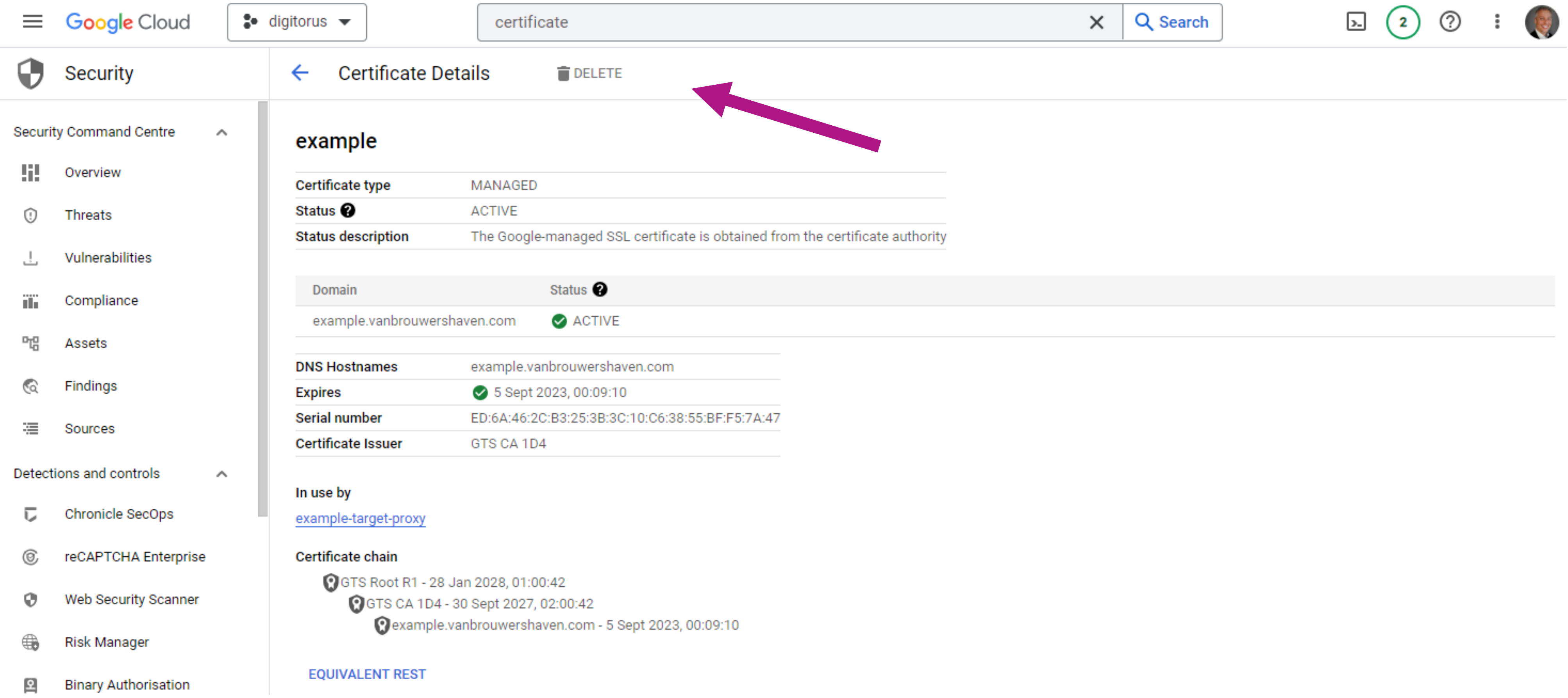


Click here to return to the load balancers view.

☑	Name ↑	Domain	Expires	Type	Scope	Status	In use by
☑	example	example.vanbrouwershaven.com	5 Sept 2023, 00:09:10	Google-managed	Global	Active	example-target-proxy

GOOGLE CLOUD (CSP) - CERTIFICATE MANAGER

No option to revoke the certificate in the certificate details



example

Certificate type	MANAGED
Status	ACTIVE
Status description	The Google-managed SSL certificate is obtained from the certificate authority

Domain	Status
example.vanbrouwershaven.com	✓ ACTIVE

DNS Hostnames	example.vanbrouwershaven.com
Expires	✓ 5 Sept 2023, 00:09:10
Serial number	ED:6A:46:2C:B3:25:3B:3C:10:C6:38:55:BF:F5:7A:47
Certificate Issuer	GTS CA 1D4

In use by

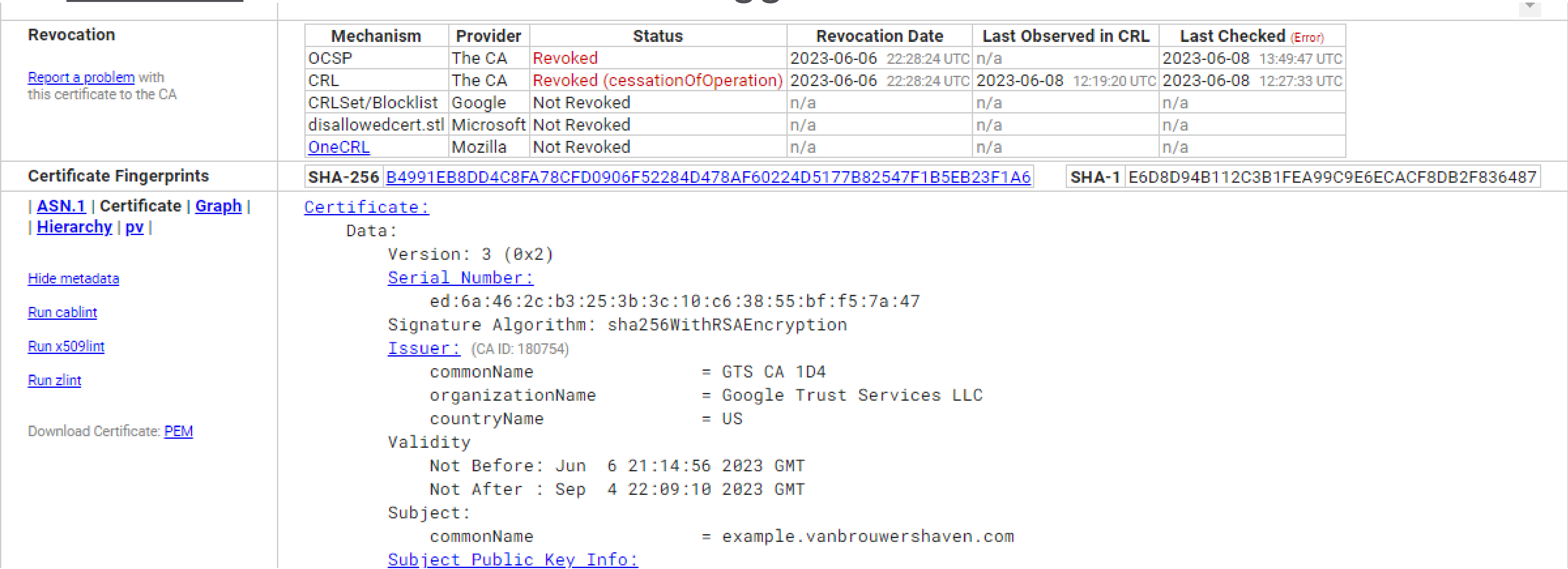
example-target-proxy

Certificate chain

- GTS Root R1 - 28 Jan 2028, 01:00:42
  - GTS CA 1D4 - 30 Sept 2027, 02:00:42
    - example.vanbrouwershaven.com - 5 Sept 2023, 00:09:10

GOOGLE CLOUD (CSP) - CERTIFICATE MANAGER

Deletion of the certificate will trigger automatic revocation



Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Revoked	2023-06-06 22:28:24 UTC	n/a	2023-06-08 13:49:47 UTC
CRL	The CA	Revoked (cessationOfOperation)	2023-06-06 22:28:24 UTC	2023-06-08 12:19:20 UTC	2023-06-08 12:27:33 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints

SHA-256 B4991EB8DD4C8FA78CFD0906F52284D478AF60224D5177B82547F1B5EB23F1A6 **SHA-1** E6D8D94B112C3B1FEA99C9E6ECACF8DB2F836487

| ASN.1 | Certificate | Graph |
| Hierarchy | pv |

Hide metadata

Run cablint

Run x509lint

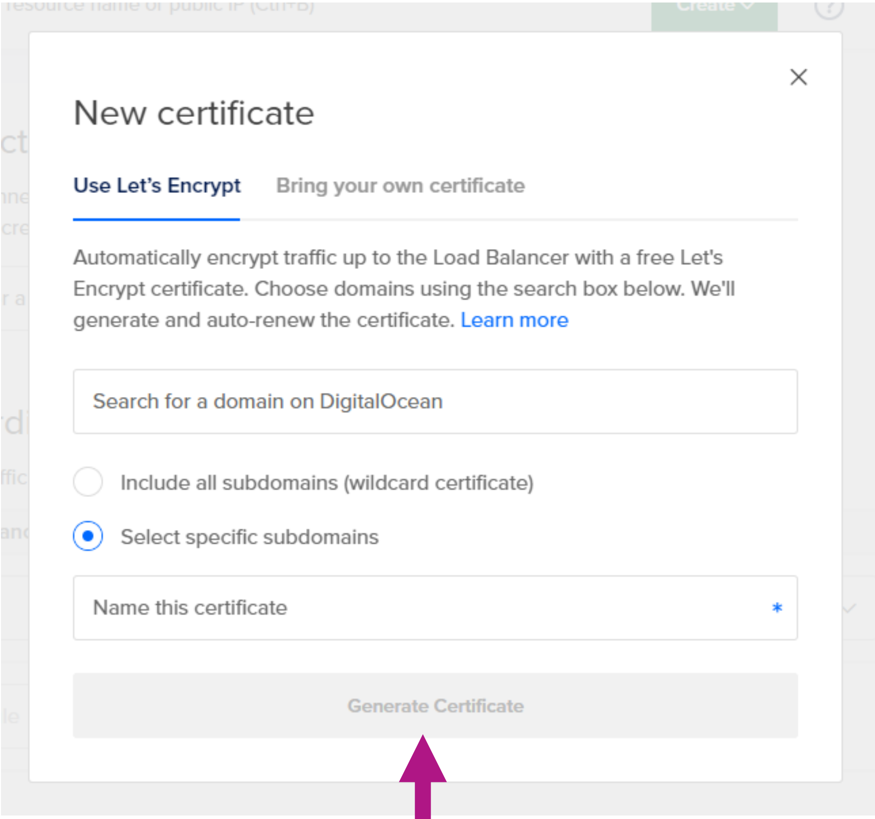
Run zlint

Download Certificate: PEM

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            ed:6a:46:2c:b3:25:3b:3c:10:c6:38:55:bf:f5:7a:47
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 180754)
            commonName                = GTS CA 1D4
            organizationName          = Google Trust Services LLC
            countryName               = US
        Validity
            Not Before: Jun  6 21:14:56 2023 GMT
            Not After : Sep  4 22:09:10 2023 GMT
        Subject:
            commonName                = example.vanbrouwershaven.com
        Subject Public Key Info:
```

ENTRUST

DIGITALOCEAN (CSP) - LOAD BALANCER



New certificate

Use Let's Encrypt | Bring your own certificate

Automatically encrypt traffic up to the Load Balancer with a free Let's Encrypt certificate. Choose domains using the search box below. We'll generate and auto-renew the certificate. Learn more

Search for a domain on DigitalOcean

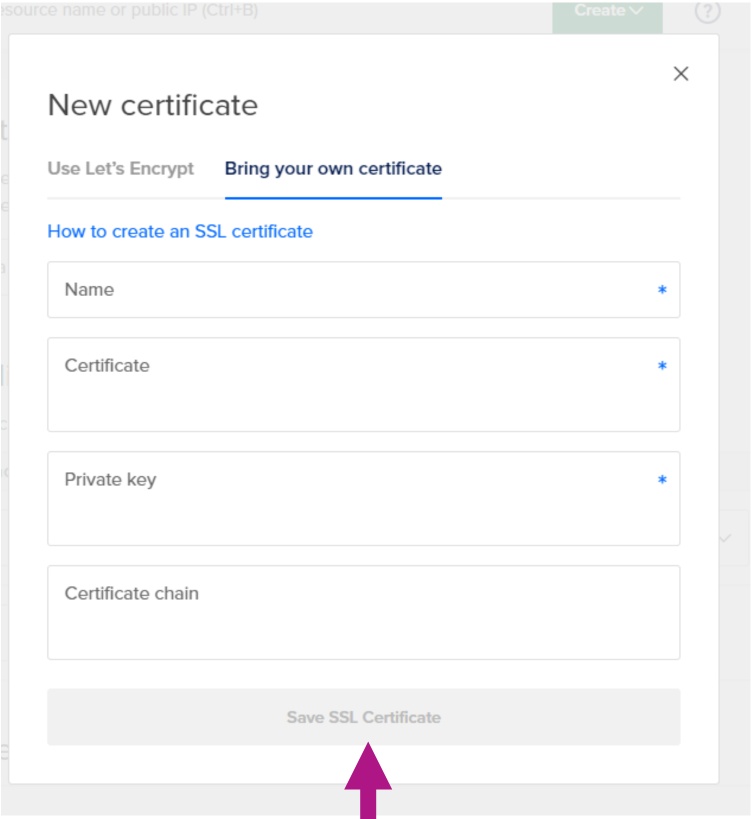
○ Include all subdomains (wildcard certificate)

● Select specific subdomains

Name this certificate *

Generate Certificate

You can use Let's Encrypt (ACME), provide some configuration, but you **can not** specify your own ACME server or account binding.



New certificate

Use Let's Encrypt | Bring your own certificate

How to create an SSL certificate

Name *

Certificate *

Private key *

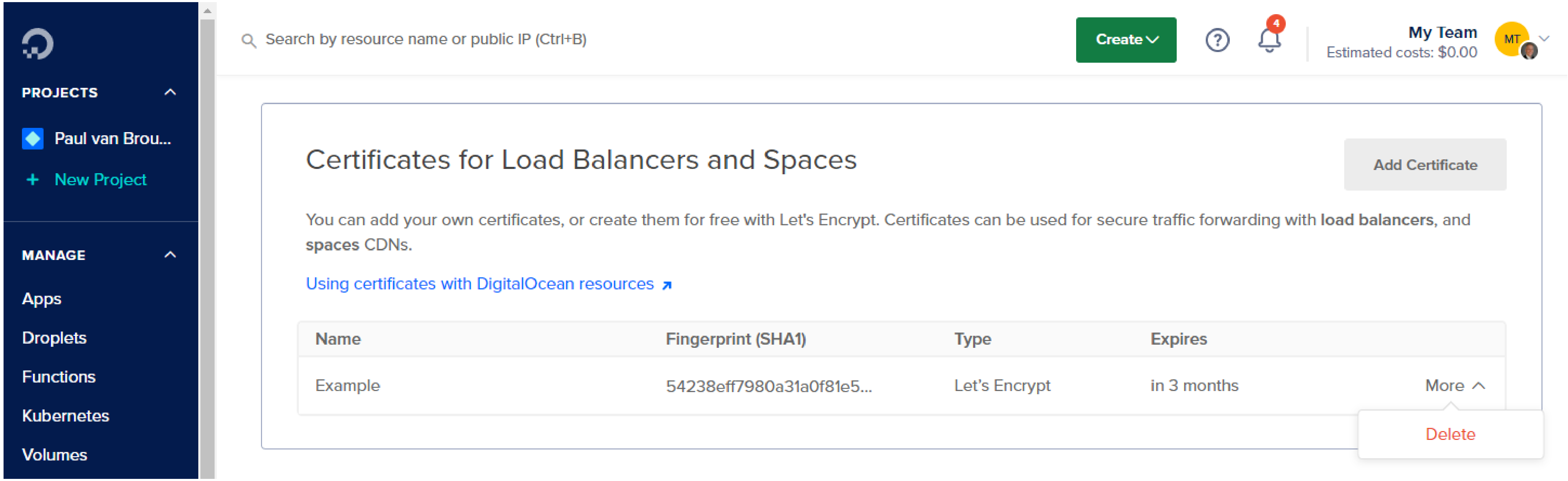
Certificate chain

Save SSL Certificate

Or you can upload a custom certificate.

DIGITALOCEAN (CSP) - LOAD BALANCER

No option to revoke, not revoked at deletion



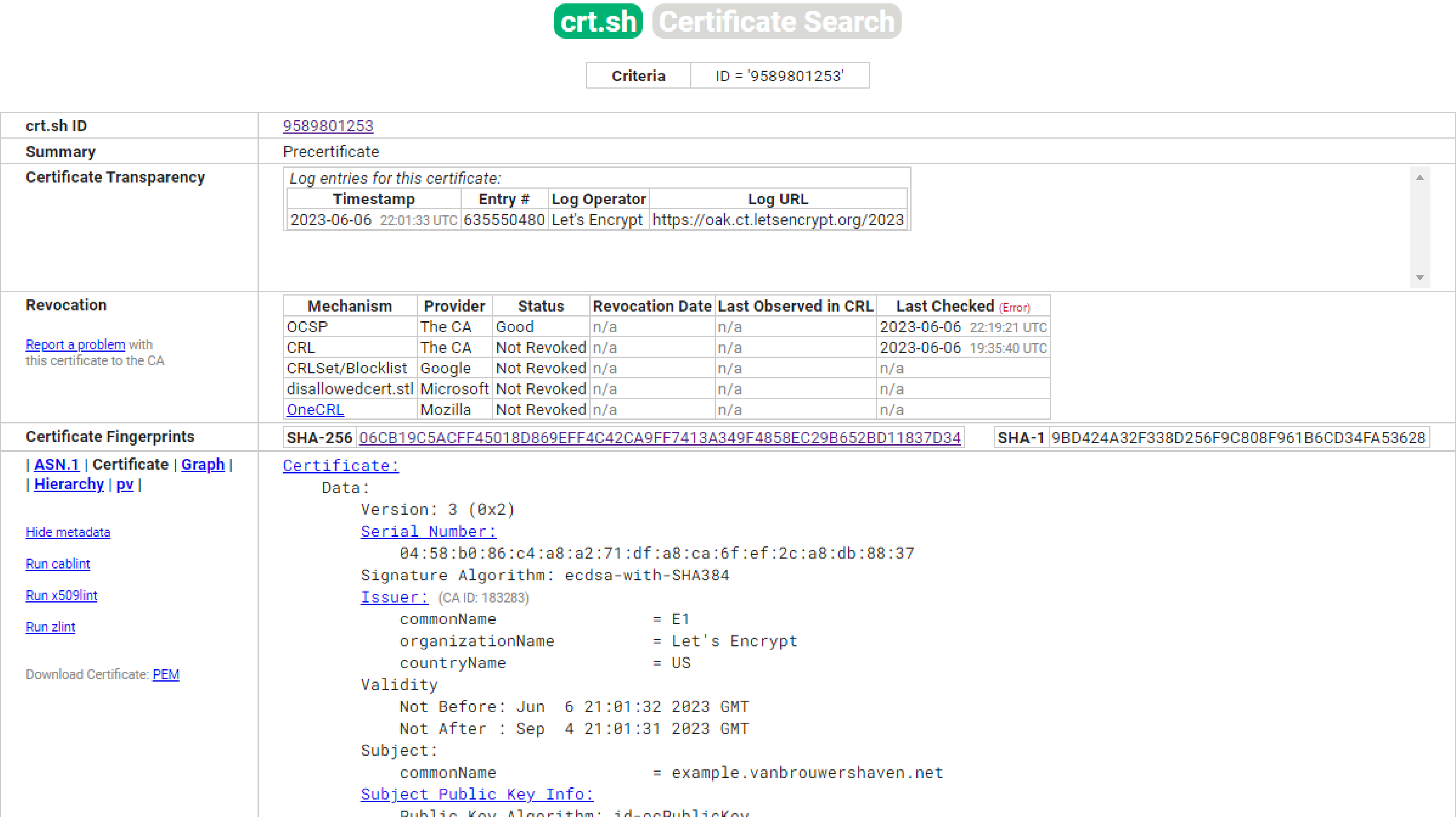
Certificates for Load Balancers and Spaces

You can add your own certificates, or create them for free with Let's Encrypt. Certificates can be used for secure traffic forwarding with **load balancers**, and **spaces** CDNs.

Using certificates with DigitalOcean resources ↗

Name	Fingerprint (SHA1)	Type	Expires
Example	54238eff7980a31a0f81e5...	Let's Encrypt	in 3 months

DIGITALOCEAN (CSP) - LOAD BALANCER



crt.sh Certificate Search

Criteria	ID = '9589801253'

crt.sh ID	9589801253
Summary	Precertificate

Certificate Transparency

Log entries for this certificate:

Timestamp	Entry #	Log Operator	Log URL
2023-06-06 22:01:33 UTC	635550480	Let's Encrypt	https://oak.ct.letsencrypt.org/2023

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Good	n/a	n/a	2023-06-06 22:19:21 UTC
CRL	The CA	Not Revoked	n/a	n/a	2023-06-06 19:35:40 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints

SHA-256	06CB19C5ACFF45018D869EFF4C42CA9FF7413A349F4858EC29B652BD11837D34	SHA-1	9BD424A32F338D256F9C808F961B6CD34FA53628

| **ASN.1** | **Certificate** | **Graph** |
| **Hierarchy** | **pv** |

Hide metadata

Run cablint

Run x509lint

Run zlint

Download Certificate: PEM

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            04:58:b0:86:c4:a8:a2:71:df:a8:ca:6f:ef:2c:a8:db:88:37
        Signature Algorithm: ecdsa-with-SHA384
        Issuer: (CA ID: 183283)
            commonName                = E1
            organizationName          = Let's Encrypt
            countryName               = US
        Validity
            Not Before: Jun  6 21:01:32 2023 GMT
            Not After : Sep  4 21:01:31 2023 GMT
        Subject:
            commonName                = example.vanbrouwershaven.net
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
```

ENTRUST

AND SOME OTHERS WE CHECKED…

- **Content Delivery Network (CDN)**
  - Cloudflare
  - Fastly
  - Akamai
- **Cloud Service Provider (CSP)**
  - Azure
  - Google Cloud
  - AWS
  - IBM Cloud
  - DigitalOcean
  - OVH
  - Hertzner
  - Vultr
- **PaaS**
  - WordPress
  - Salesforce
  - HubSpot
- **Control panels**
  - CPANEL / WHM
  - Plesk
- **Appliances / other devices**
  - HP Officejet
  - Reolink
  - Ubiquity / Unifi
  - Synology

ACME IN SAAS/PAAS MODELS

While most SaaS/PaaS provide the ability to request/provision certificates, these are limited to one or more of the following mechanisms:

1. Request and provision a DV certificate through ACME from Let’s Encrypt
2. Request and provision a certificate from us or provider X (i.e., Google, Amazon, etc.)
3. Order an OV/EV certificate from a CA under contract with the platform
4. Upload a custom certificate (using PEM/PFX files or strings)
   a) Some providers create the key for you, others require you to generate one yourself.
5. Develop a custom integration via a proprietary API

CONCLUSION

- It’s **not possible** to choose a custom ACME server unless you manage the virtual/physical server yourself so that you can change default configuration or command line options
- What if there is a security need to move to 90-day validity periods?
  - A certificate with a validity of 90-days ‘requires’ automation
    - Renewing a certificate manually 4-6 times will not be ‘appreciated’
  - When subscribers can’t specify their preferred ACME server, the **default** will become the norm!
  - If the default is the norm, we **lack issuer diversity** which will become a major point of failure.

HOW CAN WE FIX THIS?

Most providers obtain their DV certificates from Let’s Encrypt through the ACME protocol but **do not allow** users to configure their own ACME endpoint and provide no option to configure an account binding.

Maybe an update to the ACME specification that **allows clients to lookup their preferred ACME server** before contacting any ACME servers could help.

For example:

1. ACME client checks the CAA record for the (first) DNS name (example.com)
2. CAA record points to “ca.example”
3. Client downloads and checks “https://ca.example/.well-known/acme.json”
4. The “acme.json” config file contains something like:

   ```
   {
       “server”: “https://acme.ca.example/v2”,
       “account-binding”: “optional”
   }
   ```

5. The ACME client requests the certificate from https://acme.ca.examplev/v2 as indicated in the configuration

USING CAA AS DOMAIN CA ‘PREFERENCE’

- Using CAA would also **allow users** to specify a backup CA through multiple CAA records (we might need to think about the priority and certificate type, which could be CAA attributes).
- The config file would **allow CAs** to update their configuration and gives them ability to switch to a backup server in the case of an incident.
- This method would **encourage the option of CAA** and a client preference might also reduce the reliance on Let’s Encrypt and remove the potential ‘single point of failure’ it could be in the ecosystem currently.
- Of course, we could also specify the ACME server directly in the CAA or other type of DNS record but for know we thought it would be easier if clients do not have to deal with the complexity of ACME server addresses etc.

ACME IN SAAS/PAAS MODELS

- What if there is a security need to move to 90-day validity periods?
  - A 90-day certificate ‘requires’ automation
  - The **default** will become the norm when subscribers can’t specify an ACME server
  - If the default is the norm, we **lack issuer diversity** which will become a major point of failure

- Can we collaborate on an ACME RFC and client change to require and implement domain specific ACME preferences based on CAA records?

- Should we advocate that platforms need to allow users to configure **their preferred** ACME server and account binding?

ENTRUST

SECURING A WORLD IN MOTION