Springfield Amazon Web Services User Group

February 2021

#SGFAWS

Let's Encrypt 101

Secure Certificates for Web Services

Photo by JJ Ying on Unsplash

About Jason Klein

- **15+ years experience deploying and managing (many) 100's of paid and free certificates.**
- **Experience with numerous certificate authorities** (e.g. Let's Encrypt, Amazon, Comodo, RapidSSL, Verisign, Symantec, Geotrust) directly or through certificate vendors (e.g. OpenSRS, Namecheap).
- **Experience with numerous certificate types** (e.g. Single Domain, Multi Domain, Wildcard, Code Signing, Self Signed Individual, Self Signed CA)
- Follow me! @JasnK @jason-klein

AGENDA

- Secure Certificates Background
- Issuing Certificates
- How Does Let's Encrypt Work?
- Comparing Certificate Types
- Generating and Inspecting Private Keys
- Best Practices
- Common Issues, Configuration Tips, Resources

Photo by Matthew Henry on Unsplash

Why Use Secure Certificates?

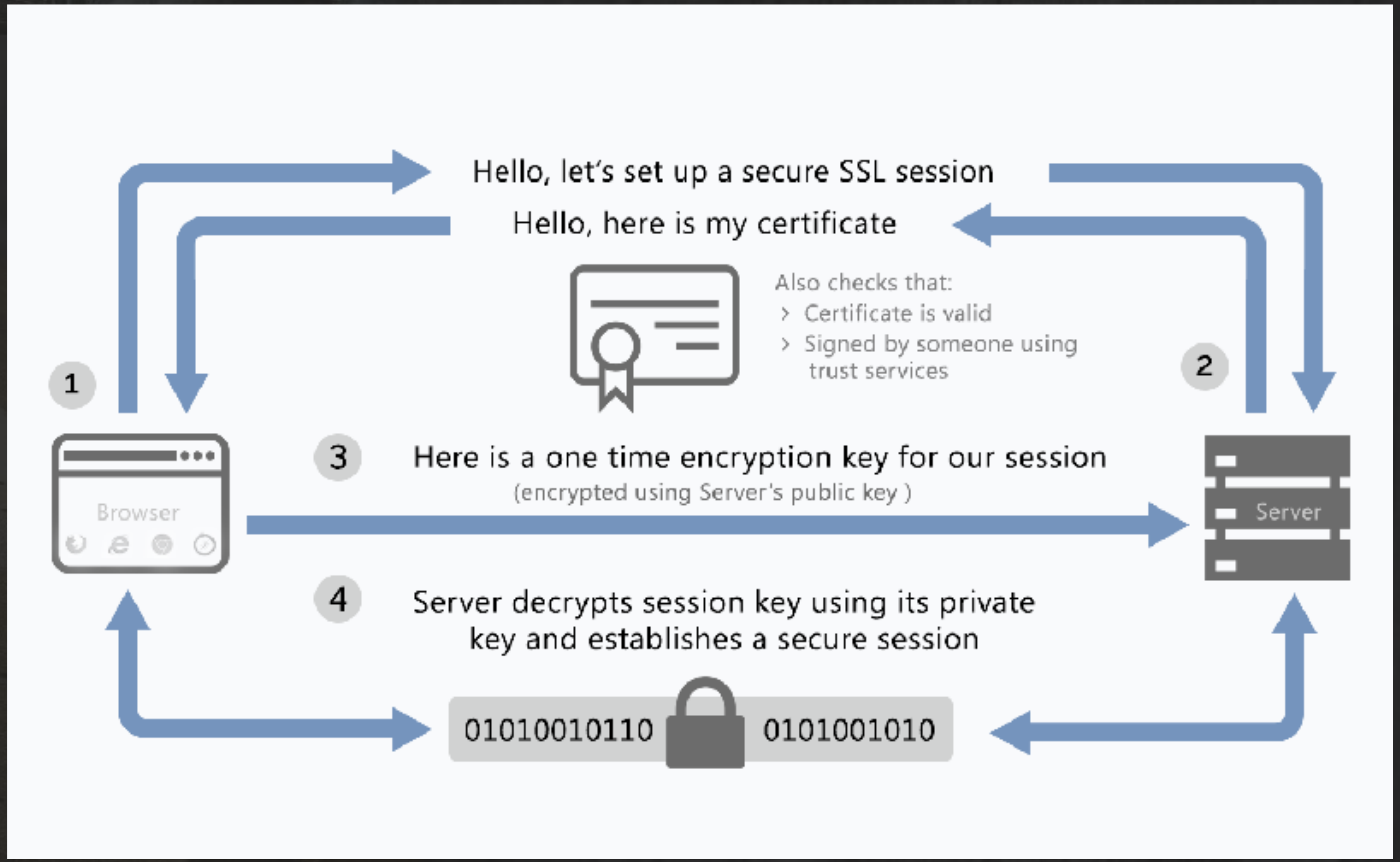
- Security - Encrypt Data between Browser and Server
- Trust - Browsers warn users not to trust insecure sites
- SEO - Google confirms HTTPS is a ranking factor
- Speed - HTTP/2 requires HTTPS

Learn More https://www.quora.com/Why-do-we-need-SSL

Photo by Matthew Henry on Unsplash

How are Secure Certificates Used to Encrypt Data?

- User's Browser requests Secure Session
- Web Server responds with its Secure Certificate, **User's Browser validates Secure Certificate.**
- User's Browser responds with Session Key, encrypted with Server public key.
- Web Server decrypts Session Key and establishes Secure Session with User.



1 Hello, let's set up a secure SSL session

2 Hello, here is my certificate

Also checks that:
- Certificate is valid
- Signed by someone using trust services

3 Here is a one time encryption key for our session (encrypted using Server's public key)

4 Server decrypts session key using its private key and establishes a secure session

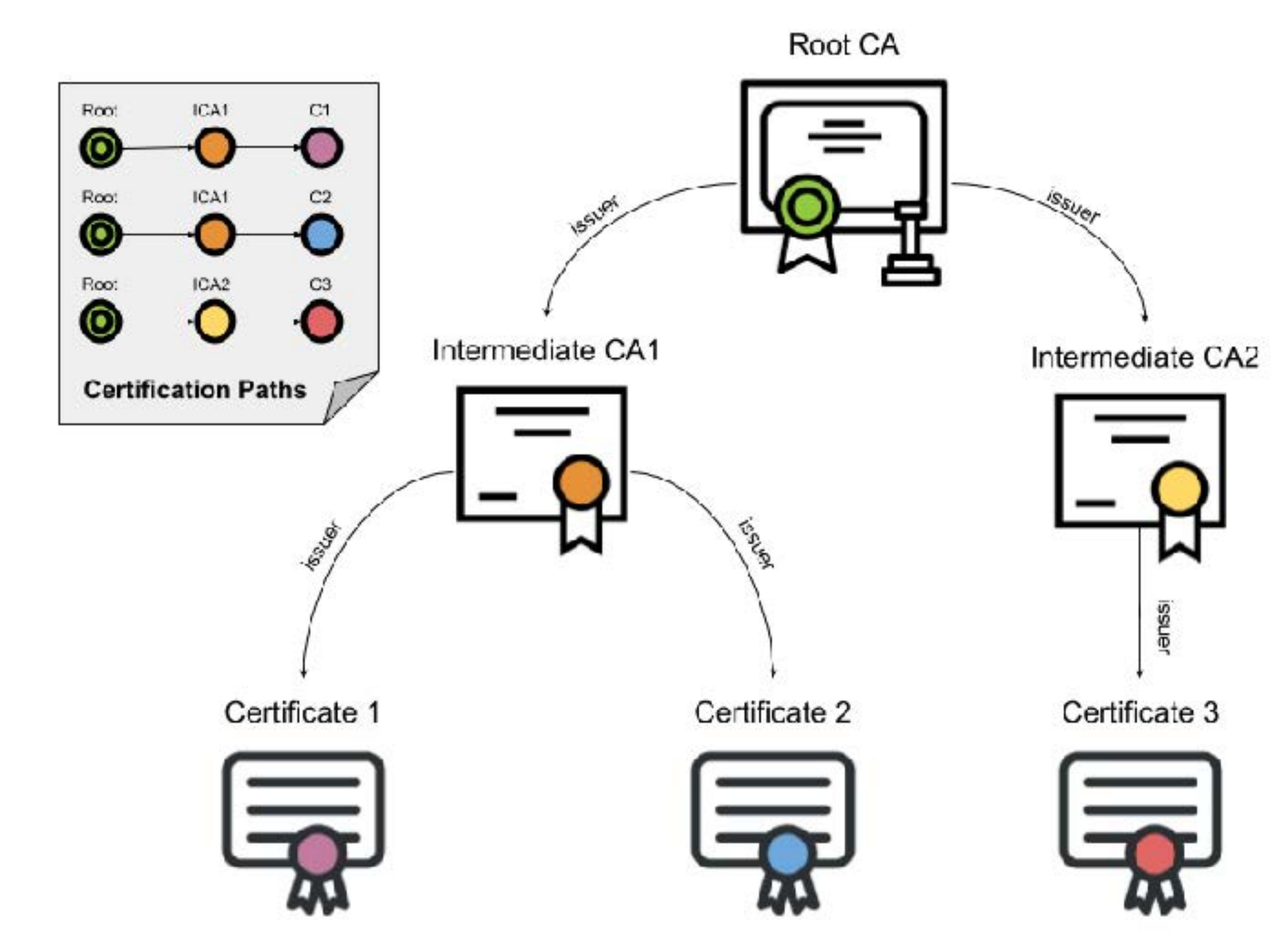
Client/Server SSL Handshake

Image https://www.cryptomathic.com/news-events/blog/encryption-https-attack-on-authentication-in-remote-banking-services-a-russian-perspective

Photo by Georg Bommeli on Unsplash

How are Secure Certificates Validated by the Web Browser?

- Browser ships with trusted Root Certificates
- User's Browser requests an HTTPS website
- Server sends Secure Certificate and optional Intermediate Certificates (aka Certificate Chain)
- Browser verifies Secure Certificate name matches the website FQDN (fully qualified domain name)
- **Browser verifies the Secure Certificate (or at least one Intermediate) is signed by a trusted Root CA**
- Browser checks for revoked certificate (CRL/OCSP)



Browser Certificate Paths

Learn More https://www.ssl.com/article/browsers-and-certificate-validation/

Photo by Agence Olloweb on Unsplash

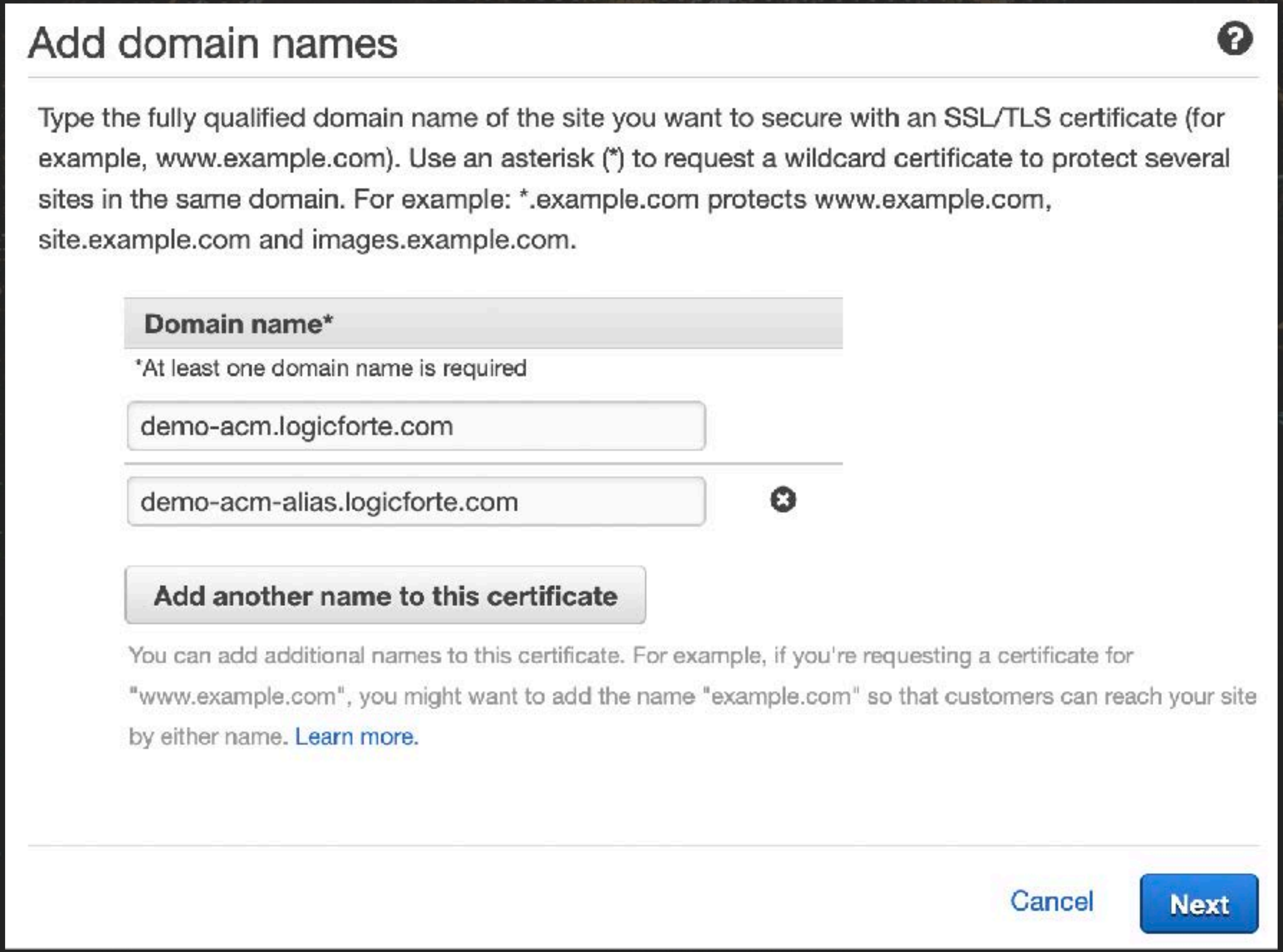
Issuing a Traditional "Domain Validated" Certificate

- **Customer** generates Private Key (KEY) and Certificate Request (CSR) for the website FQDNs (e.g. "example.com" and "www.example.com")
- **Customer** begins a Secure Certificate order, uploads CSR, selects domain validation method (e.g. email, HTTP, DNS), pays fee.
- **Customer** manually creates HTTP file or DNS record for HTTP or DNS validation (or responds to domain validation email when received from Vendor)
- **Vendor** performs domain validation using method chosen above
- **Vendor** issues certificate to Customer

Photo by Guillaume Bolduc on Unsplash

Issuing a Certificate in Amazon Certificate Manager

- AWS Console, Certificate Manager, Request Certificate, Add Domain Names



Add domain names

Type the fully qualified domain name of the site you want to secure with an SSL/TLS certificate (for example, www.example.com). Use an asterisk (*) to request a wildcard certificate to protect several sites in the same domain. For example: *.example.com protects www.example.com, site.example.com and images.example.com.

Domain name*

*At least one domain name is required

demo-acm.logicforte.com

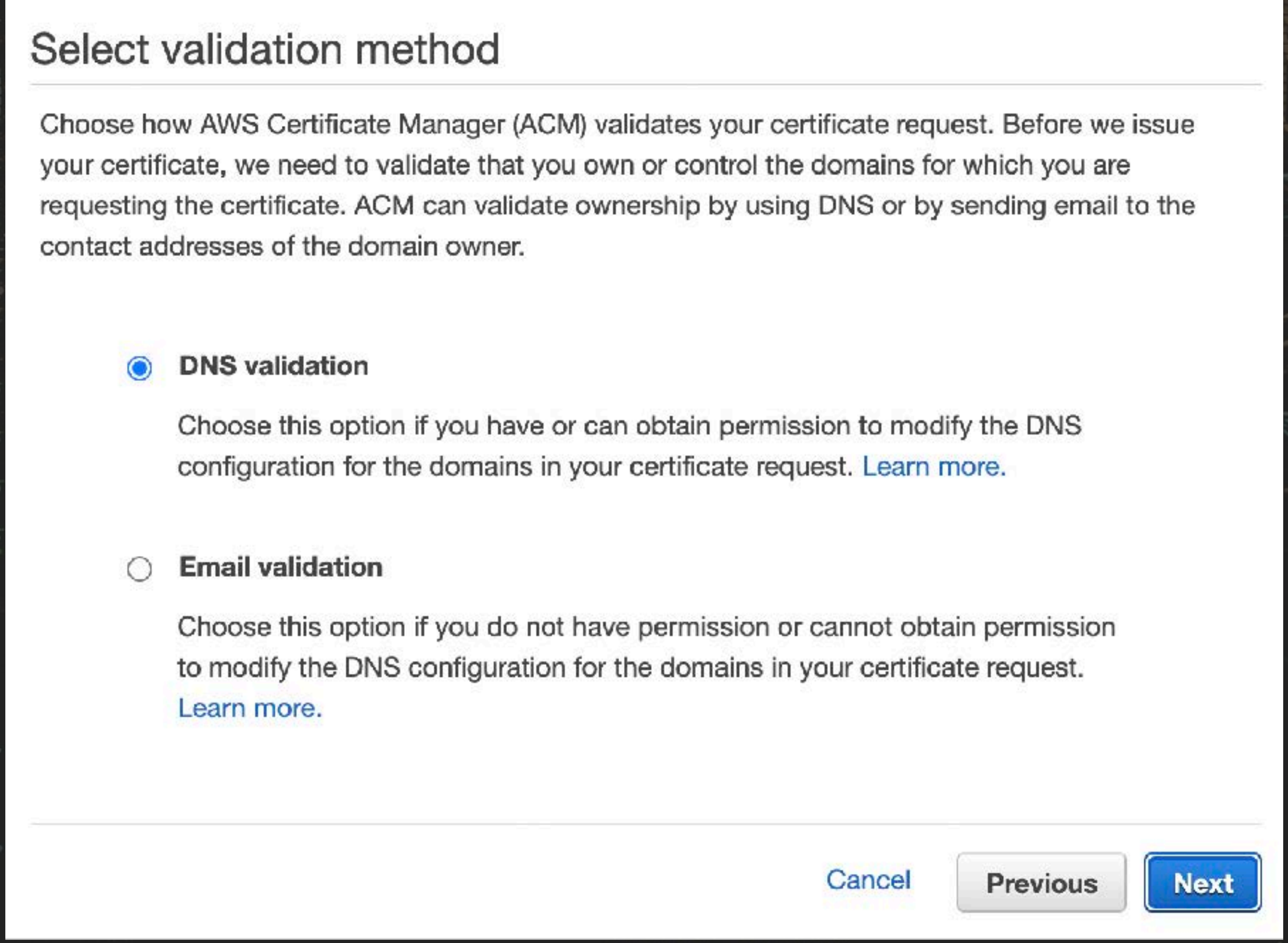
demo-acm-alias.logicforte.com

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name. Learn more.

Photo by Taylor Vick on Unsplash

Issuing a Certificate in Amazon Certificate Manager

- Select Validation Method. DNS Validation can configure DNS zones hosted in Route 53.



Select validation method

Choose how AWS Certificate Manager (ACM) validates your certificate request. Before we issue your certificate, we need to validate that you own or control the domains for which you are requesting the certificate. ACM can validate ownership by using DNS or by sending email to the contact addresses of the domain owner.

● **DNS validation**

Choose this option if you have or can obtain permission to modify the DNS configuration for the domains in your certificate request. Learn more.

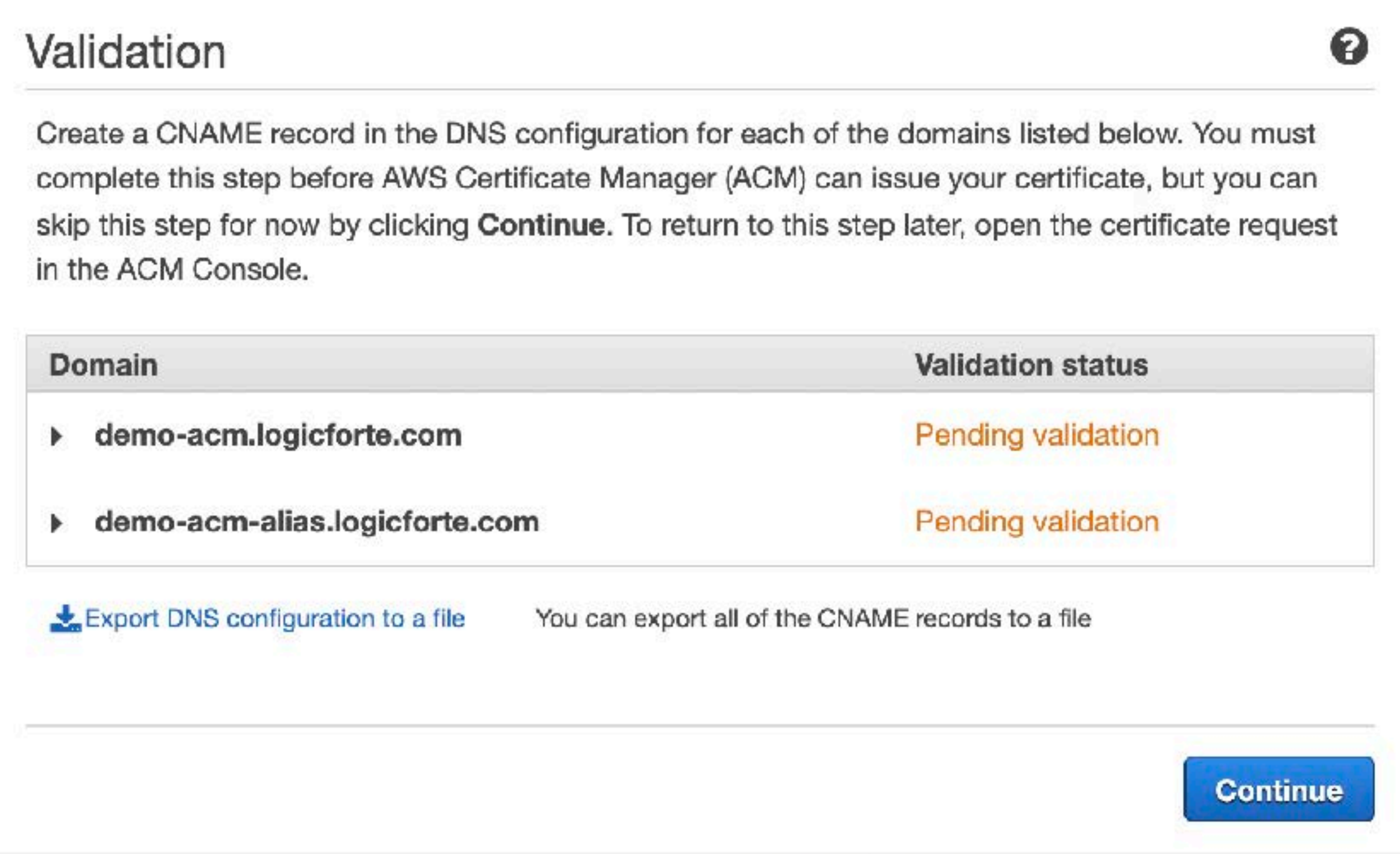
○ **Email validation**

Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request. Learn more.

Photo by Taylor Vick on Unsplash

Issuing a Certificate in Amazon Certificate Manager

- Perform Validation
- Expand each Name, Create Record in Route 53

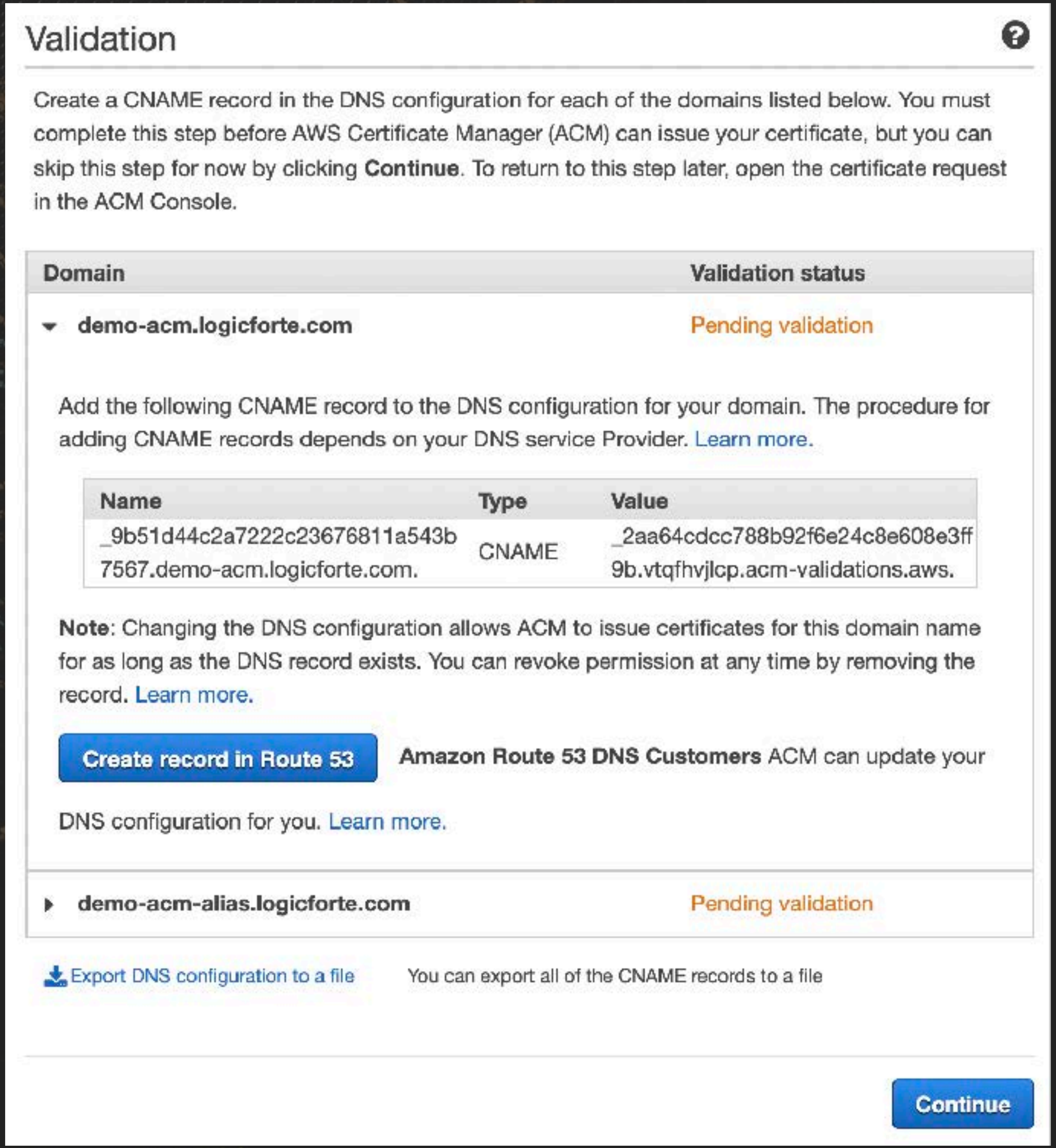


Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking **Continue**. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
▸ **demo-acm.logicforte.com**	Pending validation
▸ **demo-acm-alias.logicforte.com**	Pending validation

Export DNS configuration to a file   You can export all of the CNAME records to a file



Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking **Continue**. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
▾ **demo-acm.logicforte.com**	Pending validation

Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. Learn more.

Name	Type	Value
_9b51d44c2a7222c23676811a543b7567.demo-acm.logicforte.com.	CNAME	_2aa64cdcc788b92f6e24c8e608e3ff9b.vtqfhvjlcp.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. Learn more.

Create record in Route 53   **Amazon Route 53 DNS Customers** ACM can update your DNS configuration for you. Learn more.

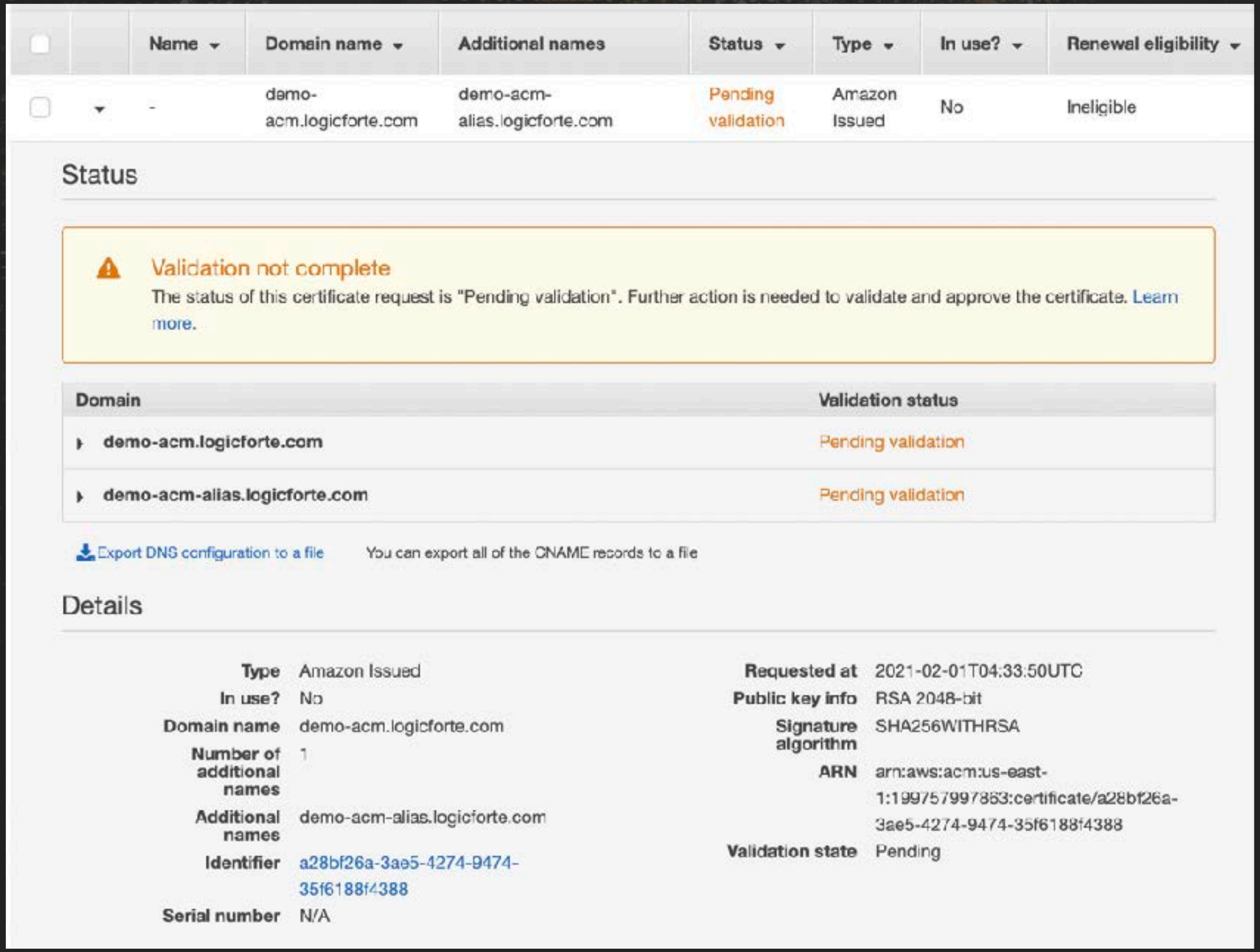
Domain	Validation status
▸ **demo-acm-alias.logicforte.com**	Pending validation

Export DNS configuration to a file   You can export all of the CNAME records to a file

Photo by Taylor Vick on Unsplash

Issuing a Certificate in Amazon Certificate Manager

- Wait for Validation…



□		Name	Domain name	Additional names	Status	Type	In use?	Renewal eligibility
□	▾	-	demo-acm.logicforte.com	demo-acm-alias.logicforte.com	Pending validation	Amazon Issued	No	Ineligible

Status

Validation not complete

The status of this certificate request is "Pending validation". Further action is needed to validate and approve the certificate. Learn more.

Domain	Validation status
▸ demo-acm.logicforte.com	Pending validation
▸ demo-acm-alias.logicforte.com	Pending validation

Export DNS configuration to a file   You can export all of the CNAME records to a file

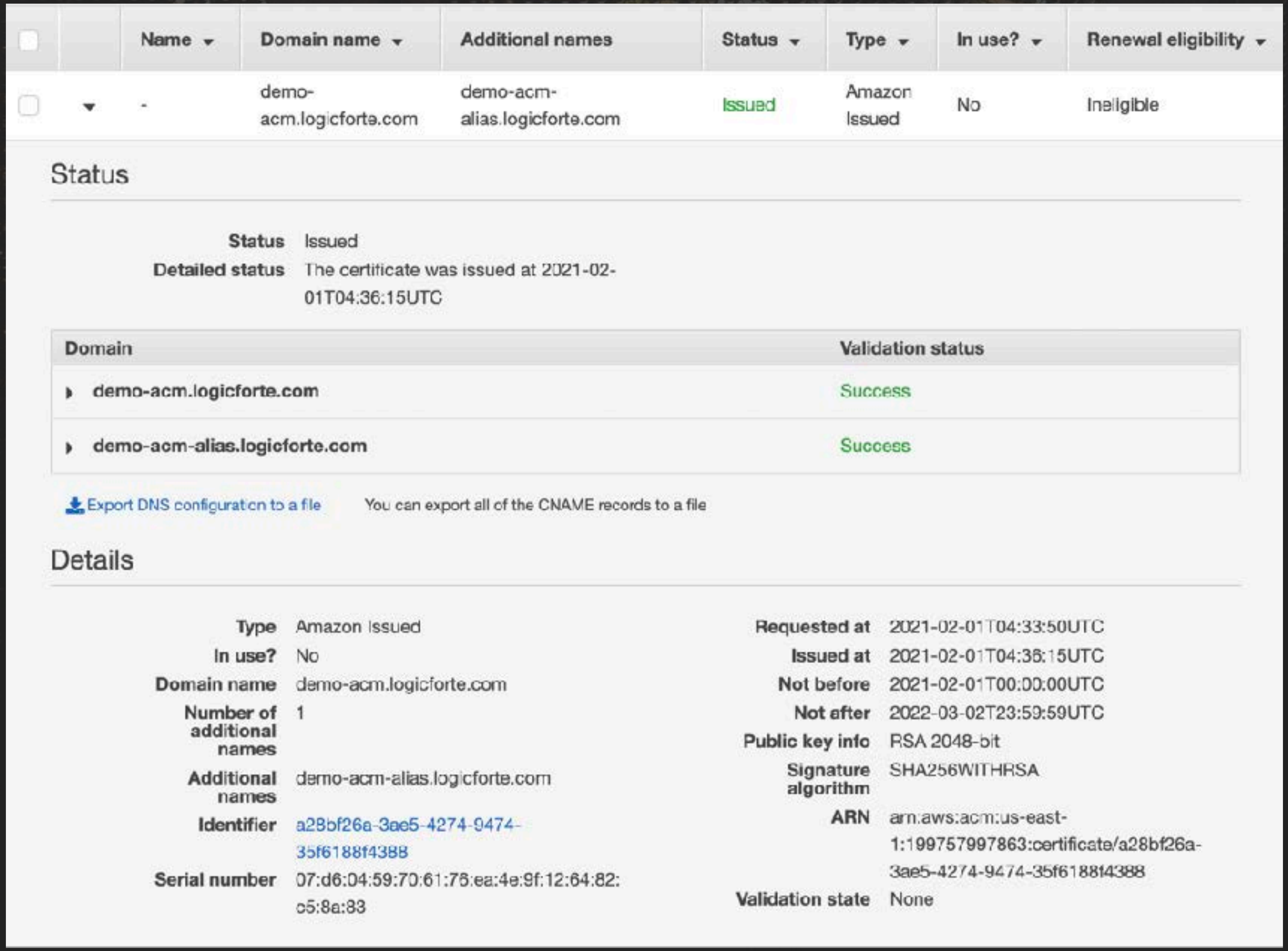
Details

Type	Amazon Issued	**Requested at**	2021-02-01T04:33:50UTC
In use?	No	**Public key info**	RSA 2048-bit
Domain name	demo-acm.logicforte.com	**Signature algorithm**	SHA256WITHRSA
Number of additional names	1	**ARN**	arn:aws:acm:us-east-1:199757997863:certificate/a28bf26a-3ae5-4274-9474-35f6188f4388
Additional names	demo-acm-alias.logicforte.com	**Validation state**	Pending
Identifier	a28bf26a-3ae5-4274-9474-35f6188f4388		
Serial number	N/A		

Photo by Taylor Vick on Unsplash

ISSUING A CERTIFICATE IN AMAZON CERTIFICATE MANAGER

- **Validation Complete!** Certificate is now ready for use in AWS services. [1]



□		Name ▾	Domain name ▾	Additional names	Status ▾	Type ▾	In use? ▾	Renewal eligibility ▾
□	▾	-	demo-acm.logicforte.com	demo-acm-alias.logicforte.com	Issued	Amazon Issued	No	Ineligible

Status

Status: Issued
Detailed status: The certificate was issued at 2021-02-01T04:36:15UTC

Domain	Validation status
▸ demo-acm.logicforte.com	Success
▸ demo-acm-alias.logicforte.com	Success

Export DNS configuration to a file — You can export all of the CNAME records to a file

Details

Type: Amazon Issued
In use?: No
Domain name: demo-acm.logicforte.com
Number of additional names: 1
Additional names: demo-acm-alias.logicforte.com
Identifier: a28bf26a-3ae5-4274-9474-35f6188f4388
Serial number: 07:d6:04:59:70:61:76:ea:4e:9f:12:64:82:c5:8a:83

Requested at: 2021-02-01T04:33:50UTC
Issued at: 2021-02-01T04:36:15UTC
Not before: 2021-02-01T00:00:00UTC
Not after: 2022-03-02T23:59:59UTC
Public key info: RSA 2048-bit
Signature algorithm: SHA256WITHRSA
ARN: arn:aws:acm:us-east-1:199757997863:certificate/a28bf26a-3ae5-4274-9474-35f6188f4388
Validation state: None

[1] https://docs.aws.amazon.com/acm/latest/userguide/acm-services.html

Photo by Taylor Vick on Unsplash

Issuing a Let's Encrypt Certificate

Two most common methods of requesting a Let's Encrypt certificate:

- **Hosting Control Panel** - Customer enables HTTPS. Hosting Provider manages request and renewals.
- **Certbot ACME Client**
  1) Run Certbot ONCE to request a free certificate for website with FQDN (app.example.com) hosted in a local directory (/var/www/html)

  ```
  certbot certonly --non-interactive --webroot -w /var/www/html -d app.example.com --agree-tos --email hostmaster@example.com
  ```

  2) Schedule Certbot Renew to renew all Let's Encrypt certificates expiring in less than 30 days, since Certificates expire every 90 days. *The Debian certbot package automatically schedules this command to run twice per day.*

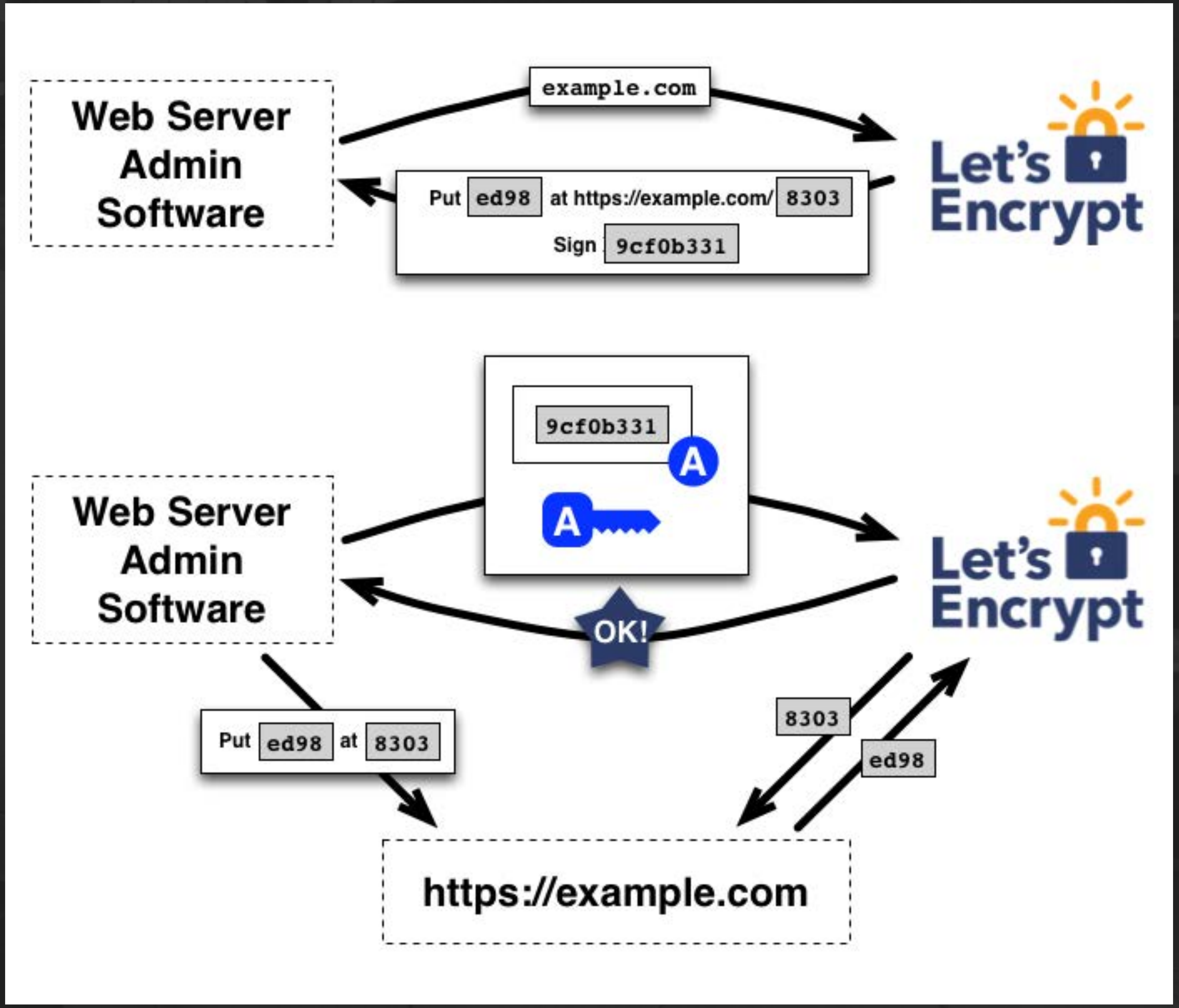
  ```
  certbot -q renew
  ```

  Learn more about the Certbot ACME client https://certbot.eff.org/

Photo by Gary Doughty on Unsplash

How Does Let's Encrypt Work?

- **Client** sends challenge request for FQDN
- **Server** lists available challenges (e.g. HTTP or DNS) and requests a verification signature
- **Client** sends the verification signature
- **Client** performs challenge (e.g. upload specific filename and contents to HTTP website)
- **Server** verifies client signature and client challenge, then issues certificate to Client
- **This entire process takes only a few seconds!**



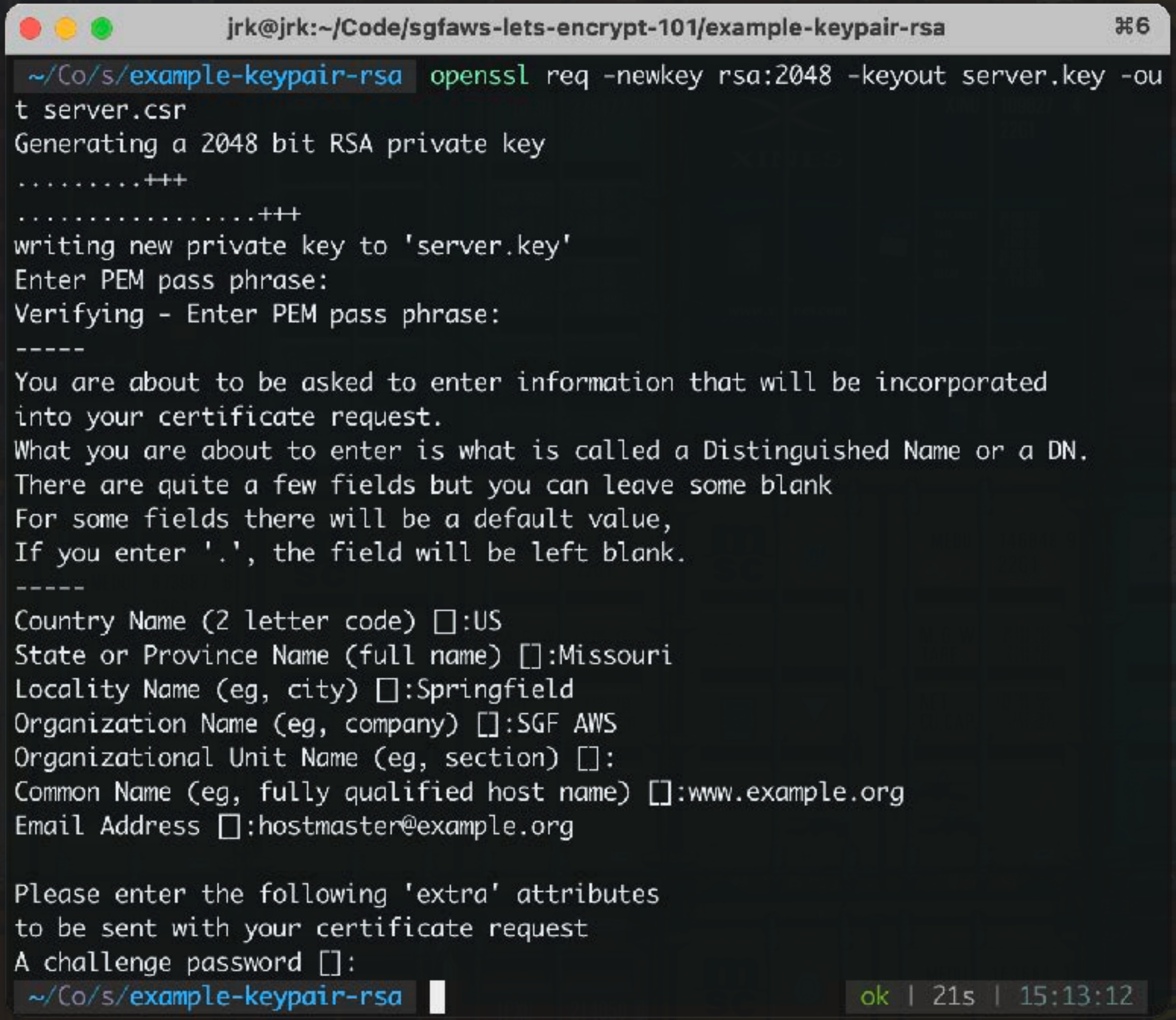
Challenge Request + Verification Exchange

Learn More https://letsencrypt.org/how-it-works/

Photo by Gary Doughty on Unsplash

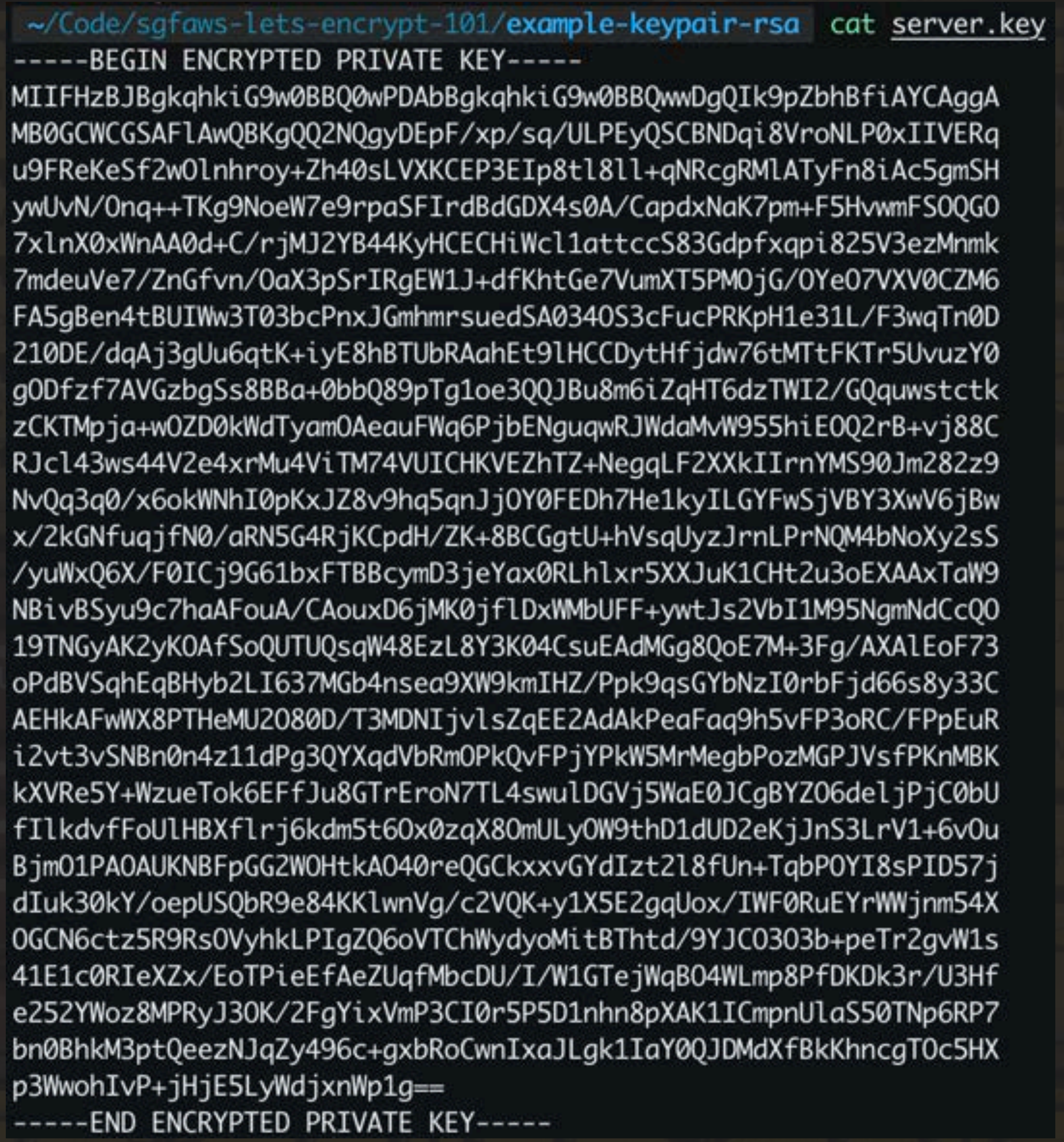
GENERATING AN RSA PRIVATE KEY AND CSR

- **Generate KEY and CSR**

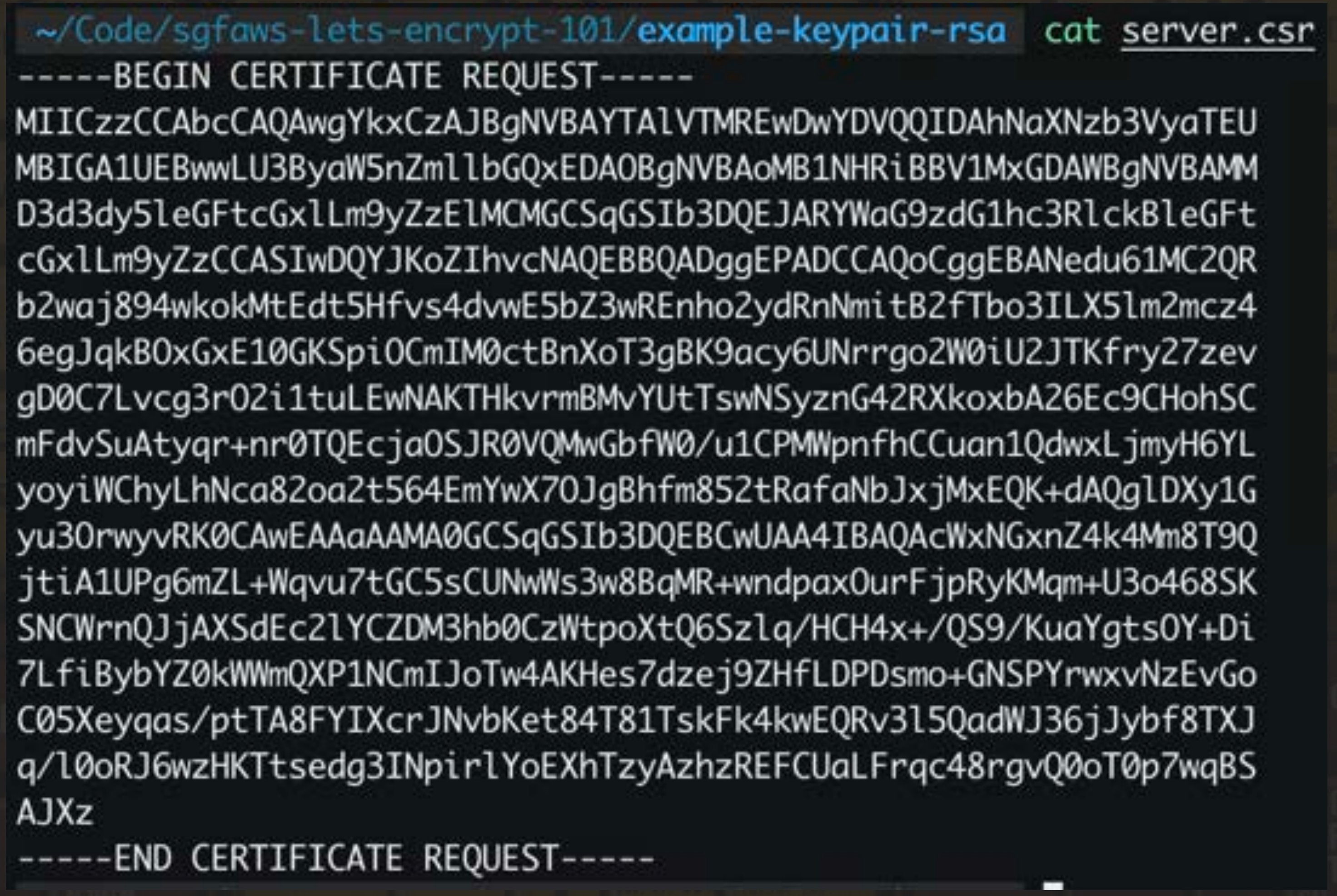


```
 ~/Co/s/example-keypair-rsa  openssl req -newkey rsa:2048 -keyout server.key -ou
t server.csr
Generating a 2048 bit RSA private key
.........+++
.................+++
writing new private key to 'server.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:Missouri
Locality Name (eg, city) []:Springfield
Organization Name (eg, company) []:SGF AWS
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:www.example.org
Email Address []:hostmaster@example.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
 ~/Co/s/example-keypair-rsa                                ok | 21s | 15:13:12
```



```
 ~/Code/sgfaws-lets-encrypt-101/example-keypair-rsa  cat server.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFHzBJBgkqhkiG9w0BBQ0wPDAbBgkqhkiG9w0BBQwwDgQIk9pZbhBfiAYCAggA
MB0GCWCGSAFlAwQBKgQQ2NQgyDEpF/xp/sq/ULPEyQSCBNDqi8VroNLP0xIIVERq
u9FReKeSf2wOlnhroy+Zh40sLVXKCEP3EIp8tl8ll+qNRcgRMlATyFn8iAc5gmSH
ywUvN/Onq++TKg9NoeW7e9rpaSFIrdBdGDX4s0A/CapdxNaK7pm+F5HvwmFSOQGO
7xlnX0xWnAA0d+C/rjMJ2YB44KyHCECHiWcl1attccS83Gdpfxqpi825V3ezMnmk
7mdeuVe7/ZnGfvn/OaX3pSrIRgEW1J+dfKhtGe7VumXT5PMOjG/OYeO7VXV0CZM6
FA5gBen4tBUIWw3T03bcPnxJGmhmrsuedSA034OS3cFucPRKpH1e31L/F3wqTn0D
210DE/dqAj3gUu6qtK+iyE8hBTUbRAahEt9lHCCDytHfjdw76tMTtFKTr5UvuzY0
gODfzf7AVGzbgSs8BBa+0bbQ89pTg1oe3QQJBu8m6iZqHT6dzTWI2/GQquwstctk
zCKTMpja+wOZD0kWdTyamOAeauFWq6PjbENguqwRJWdaMvW955hiEOQ2rB+vj88C
RJcl43ws44V2e4xrMu4ViTM74VUICHKVEZhTZ+NegqLF2XXkIIrnYMS90Jm282z9
NvQq3q0/x6okWNhI0pKxJZ8v9hq5qnJjOY0FEDh7He1kyILGYFwSjVBY3XwV6jBw
x/2kGNfuqjfN0/aRN5G4RjKCpdH/ZK+8BCGgtU+hVsqUyzJrnLPrNQM4bNoXy2sS
/yuWxQ6X/F0ICj9G61bxFTBBcymD3jeYax0RLhlxr5XXJuK1CHt2u3oEXAAxTaW9
NBivBSyu9c7haAFouA/CAouxD6jMK0jflDxWMbUFF+ywtJs2VbI1M95NgmNdCcQO
19TNGyAK2yKOAfSoQUTUQsqW48EzL8Y3K04CsuEAdMGg8QoE7M+3Fg/AXAlEoF73
oPdBVSqhEqBHyb2LI637MGb4nsea9XW9kmIHZ/Ppk9qsGYbNzI0rbFjd66s8y33C
AEHkAFwWX8PTHeMU2O80D/T3MDNIjvlsZqEE2AdAkPeaFaq9h5vFP3oRC/FPpEuR
i2vt3vSNBn0n4z11dPg3QYXqdVbRmOPkQvFPjYPkW5MrMegbPozMGPJVsfPKnMBK
kXVRe5Y+WzueTok6EFfJu8GTrEroN7TL4swulDGVj5WaE0JCgBYZO6deljPjC0bU
fIlkdvfFoUlHBXflrj6kdm5t6Ox0zqX8OmULyOW9thD1dUD2eKjJnS3LrV1+6vOu
BjmO1PAOAUKNBFpGG2WOHtkAO40reQGCkxxvGYdIzt2l8fUn+TqbPOYI8sPID57j
dIuk30kY/oepUSQbR9e84KKlwnVg/c2VQK+y1X5E2gqUox/IWF0RuEYrWWjnm54X
OGCN6ctz5R9RsOVyhkLPIgZQ6oVTChWydyoMitBThtd/9YJCO3O3b+peTr2gvW1s
41E1c0RIeXZx/EoTPieEfAeZUqfMbcDU/I/W1GTejWqBO4WLmp8PfDKDk3r/U3Hf
e252YWoz8MPRyJ3OK/2FgYixVmP3CI0r5P5D1nhn8pXAK1ICmpnUlaS50TNp6RP7
bn0BhkM3ptQeezNJqZy496c+gxbRoCwnIxaJLgk1IaY0QJDMdXfBkKhncgTOc5HX
p3WwohIvP+jHjE5LyWdjxnWp1g==
-----END ENCRYPTED PRIVATE KEY-----
```

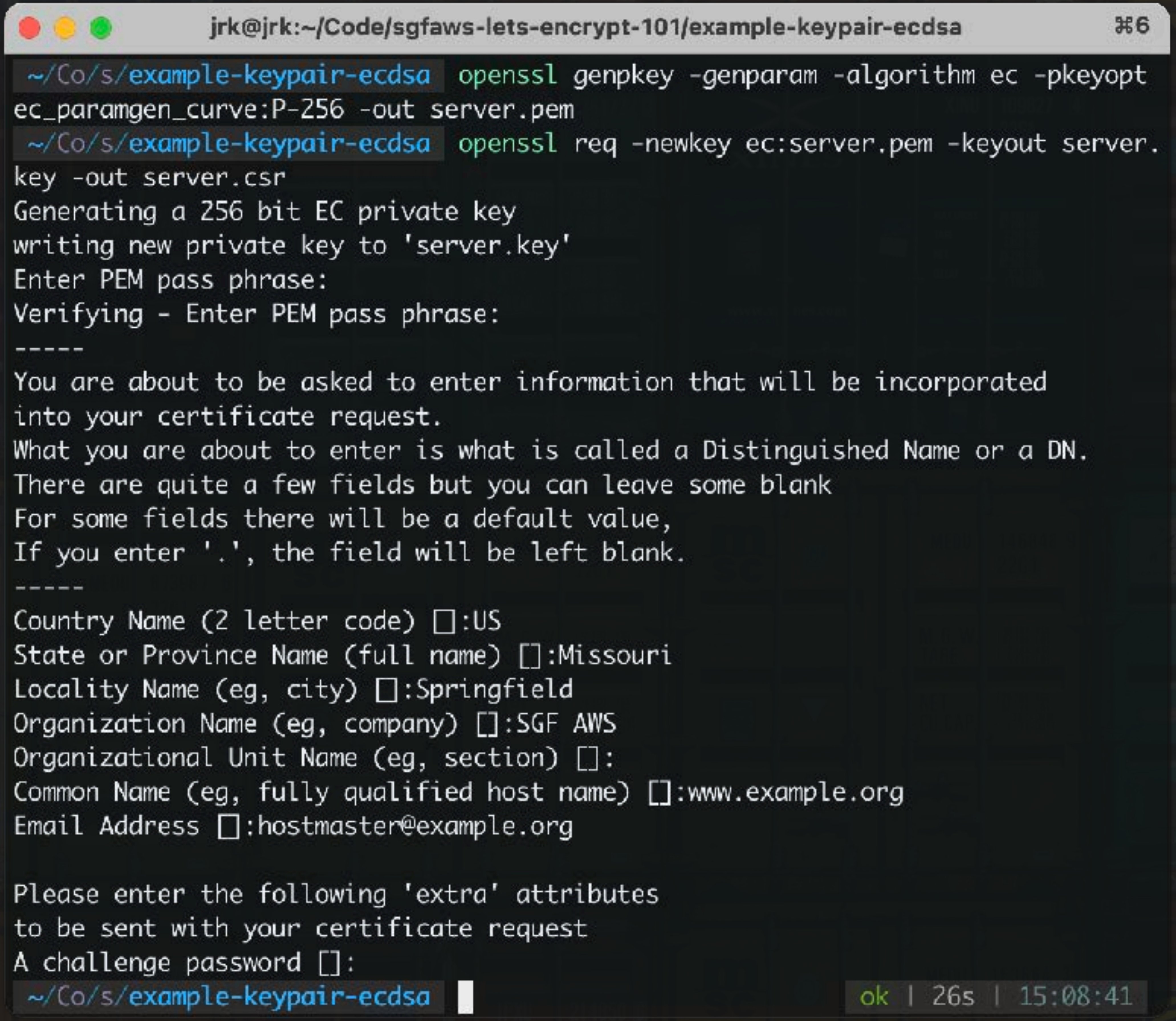


```
 ~/Code/sgfaws-lets-encrypt-101/example-keypair-rsa  cat server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICzzCCAbcCAQAwgYkxCzAJBgNVBAYTAlVTMREwDwYDVQQIDAhNaXNzb3VyaTEU
MBIGA1UEBwwLU3ByaW5nZmllbGQxEDAOBgNVBAoMB1NHRiBBV1MxGDAWBgNVBAMM
D3d3dy5leGFtcGxlLm9yZzElMCMGCSqGSIb3DQEJARYWaG9zdG1hc3RlckBleGFt
cGxlLm9yZzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANedu61MC2QR
b2waj894wkokMtEdt5Hfvs4dvwE5bZ3wREnho2ydRnNmitB2fTbo3ILX5lm2mcz4
6egJqkBOxGxE10GKSpiOCmIM0ctBnXoT3gBK9acy6UNrrgo2W0iU2JTKfry27zev
gD0C7Lvcg3rO2i1tuLEwNAKTHkvrmBMvYUtTswNSyznG42RXkoxbA26Ec9CHohSC
mFdvSuAtyqr+nr0TQEcjaOSJR0VQMwGbfW0/u1CPMWpnfhCCuan1QdwxLjmyH6YL
yoyiWChyLhNca82oa2t564EmYwX7OJgBhfm852tRafaNbJxjMxEQK+dAQglDXy1G
yu3OrwyvRK0CAwEAAaAAMA0GCSqGSIb3DQEBCwUAA4IBAQAcWxNGxnZ4k4Mm8T9Q
jtiA1UPg6mZL+Wqvu7tGC5sCUNwWs3w8BqMR+wndpaxOurFjpRyKMqm+U3o468SK
SNCWrnQJjAXSdEc2lYCZDM3hb0CzWtpoXtQ6Szlq/HCH4x+/QS9/KuaYgtsOY+Di
7LfiBybYZ0kWWmQXP1NCmIJoTw4AKHes7dzej9ZHfLDPDsmo+GNSPYrwxvNzEvGo
C05Xeyqas/ptTA8FYIXcrJNvbKet84T81TskFk4kwEQRv3l5QadWJ36jJybf8TXJ
q/l0oRJ6wzHKTtsedg3INpirlYoEXhTzyAzhzREFCUaLFrqc48rgvQ0oT0p7wqBS
AJXz
-----END CERTIFICATE REQUEST-----
```

Photo by Guillaume Bolduc on Unsplash

Generating an ECDSA Private Key and CSR

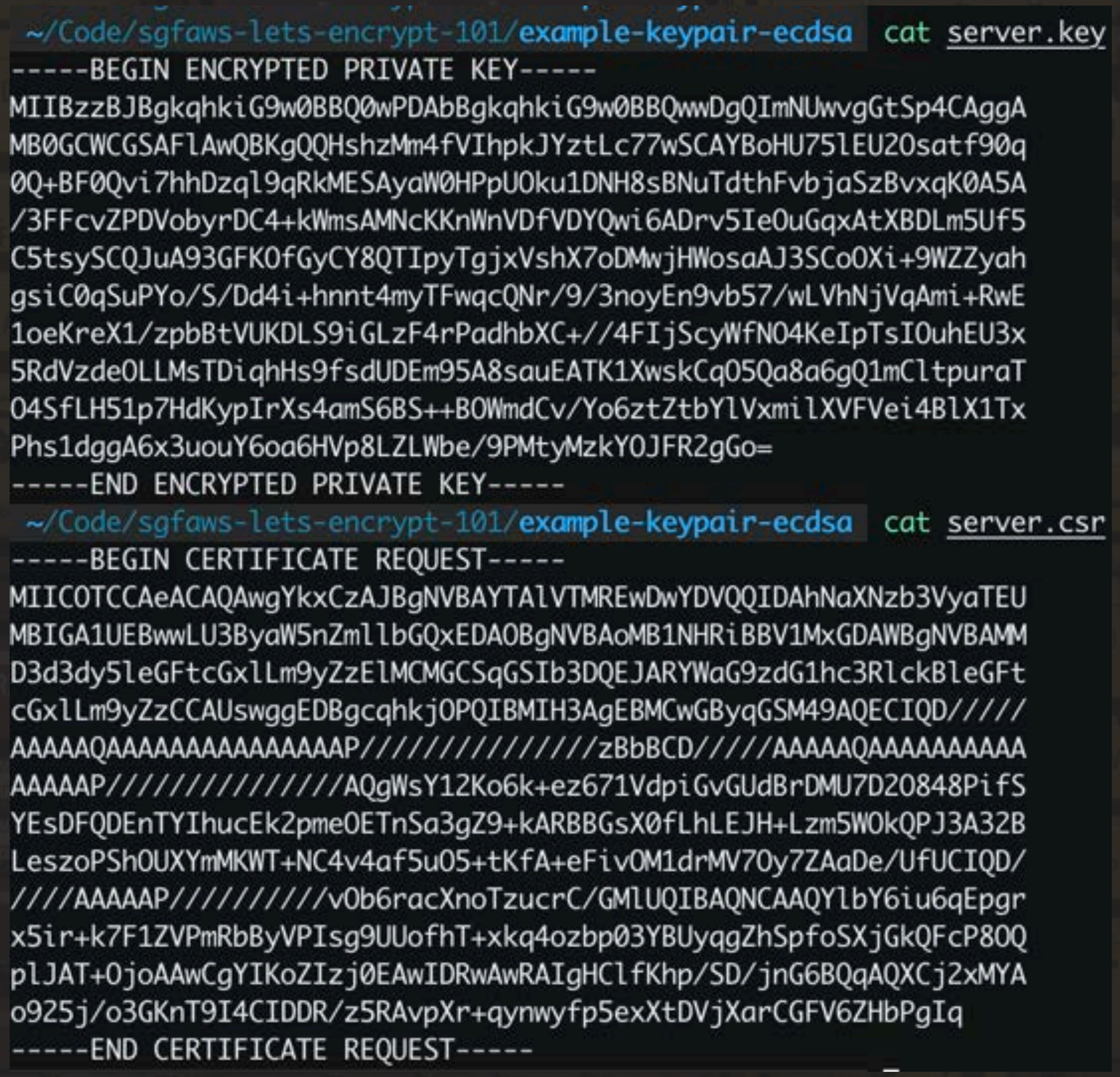
- **Generate Parameters, Generate Key/CSR**



jrk@jrk:~/Code/sgfaws-lets-encrypt-101/example-keypair-ecdsa ⌘6

```
~/Co/s/example-keypair-ecdsa  openssl genpkey -genparam -algorithm ec -pkeyopt
ec_paramgen_curve:P-256 -out server.pem
~/Co/s/example-keypair-ecdsa  openssl req -newkey ec:server.pem -keyout server.
key -out server.csr
Generating a 256 bit EC private key
writing new private key to 'server.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:Missouri
Locality Name (eg, city) []:Springfield
Organization Name (eg, company) []:SGF AWS
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:www.example.org
Email Address []:hostmaster@example.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
~/Co/s/example-keypair-ecdsa                              ok | 26s | 15:08:41
```



```
~/Code/sgfaws-lets-encrypt-101/example-keypair-ecdsa  cat server.key
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIBzzBJBgkqhkiG9w0BBQ0wPDAbBgkqhkiG9w0BBQwwDgQImNUwvgGtSp4CAggA
MB0GCWCGSAFlAwQBKgQQHshzMm4fVIhpkJYztLc77wSCAYBoHU75lEU2Osatf90q
0Q+BF0Qvi7hhDzql9qRkMESAyaW0HPpUOku1DNH8sBNuTdthFvbjaSzBvxqK0A5A
/3FFcvZPDVobyrDC4+kWmsAMNcKKnWnVDfVDYQwi6ADrv5IeOuGqxAtXBDLm5Uf5
C5tsySCQJuA93GFKOfGyCY8QTIpyTgjxVshX7oDMwjHWosaAJ3SCoOXi+9WZZyah
gsiC0qSuPYo/S/Dd4i+hnnt4myTFwqcQNr/9/3noyEn9vb57/wLVhNjVqAmi+RwE
1oeKreX1/zpbBtVUKDLS9iGLzF4rPadhbXC+//4FIjScyWfNO4KeIpTsIOuhEU3x
5RdVzdeOLLMsTDiqhHs9fsdUDEm95A8sauEATK1XwskCq05Qa8a6gQ1mCltpuraT
O4SfLH51p7HdKypIrXs4amS6BS++BOWmdCv/Yo6ztZtbYlVxmilXVFVei4BlX1Tx
Phs1dggA6x3uouY6oa6HVp8LZLWbe/9PMtyMzkYOJFR2gGo=
-----END ENCRYPTED PRIVATE KEY-----
~/Code/sgfaws-lets-encrypt-101/example-keypair-ecdsa  cat server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIICOTCCAeACAQAwgYkxCzAJBgNVBAYTAlVTMREwDwYDVQQIDAhNaXNzb3VyaTEU
MBIGA1UEBwwLU3ByaW5nZmllbGQxEDAOBgNVBAoMB1NHRiBBV1MxGDAWBgNVBAMM
D3d3dy5leGFtcGxlLm9yZzElMCMGCSqGSIb3DQEJARYWaG9zdG1hc3RlckBleGFt
cGxlLm9yZzCCAUswggEDBgcqhkjOPQIBMIH3AgEBMCwGByqGSM49AQECIQD/////
AAAAAQAAAAAAAAAAAAAAAAP///////////////zBbBCD/////AAAAAQAAAAAAAAAAAA
AAAAAP///////////////AQgWsY12Ko6k+ez671VdpiGvGUdBrDMU7D2O848PifS
YEsDFQDEnTYIhucEk2pmeOETnSa3gZ9+kARBBGsX0fLhLEJH+Lzm5WOkQPJ3A32B
LeszoPShOUXYmMKWT+NC4v4af5uO5+tKfA+eFivOM1drMV7Oy7ZAaDe/UfUCIQD/
////AAAAAP//////////vOb6racXnoTzucrC/GMlUQIBAQNCAAQYlbY6iu6qEpgr
x5ir+k7F1ZVPmRbByVPIsg9UUofhT+xkq4ozbp03YBUyqgZhSpfoSXjGkQFcP8OQ
plJAT+OjoAAwCgYIKoZIzj0EAwIDRwAwRAIgHClfKhp/SD/jnG6BQqAQXCj2xMYA
o925j/o3GKnT9I4CIDDR/z5RAvpXr+qynwyfp5exXtDVjXarCGFV6ZHbPgIq
-----END CERTIFICATE REQUEST-----
```

Photo by Guillaume Bolduc on Unsplash

Inspecting RSA CSR and Private Key

- **Certificate Request**

```
openssl req -noout \
-text -in server.csr
```

- **Private Key**

```
openssl rsa -noout \
-text -in server.key
```

```
$ openssl req -noout -text -in server.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=US, ST=Missouri, L=Springfield, O=SGF AWS,
CN=www.example.org/emailAddress=hostmaster@example.org
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d7:9d:bb:ad:4c:0b:64:11:6f:6c:1a:8f:cf:78:
                    c2:4a:24:32:d1:1d:b7:91:df:be:ce:1d:bf:01:39:
                    6d:9d:f0:44:49:e1:a3:6c:9d:46:73:66:8a:d0:76:
                    7d:36:e8:dc:82:d7:e6:59:b6:99:cc:f8:e9:e8:09:
                    aa:40:4e:c4:6c:44:d7:41:8a:4a:98:8e:0a:62:0c:
                    d1:cb:41:9d:7a:13:de:00:4a:f5:a7:32:e9:43:6b:
                    ae:0a:36:5b:48:94:d8:94:ca:7e:bc:b6:ef:37:af:
                    80:3d:02:ec:bb:dc:83:7a:ce:da:2d:6d:b8:b1:30:
                    34:02:93:1e:4b:eb:98:13:2f:61:4b:53:b3:03:52:
                    cb:39:c6:e3:64:57:92:8c:5b:03:6e:84:73:d0:87:
                    a2:14:82:98:57:6f:4a:e0:2d:ca:aa:fe:9e:bd:13:
                    40:47:23:68:e4:89:47:45:50:33:01:9b:7d:6d:3f:
                    bb:50:8f:31:6a:67:7e:10:82:b9:a9:f5:41:dc:31:
                    2e:39:b2:1f:a6:0b:ca:8c:a2:58:28:72:2e:13:5c:
                    6b:cd:a8:6b:6b:79:eb:81:26:63:05:fb:38:98:01:
                    85:f9:bc:e7:6b:51:69:f6:8d:6c:9c:63:33:11:10:
                    2b:e7:40:42:09:43:5f:2d:46:ca:ed:ce:af:0c:af:
                    44:ad
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
   Signature Algorithm: sha256WithRSAEncryption
        1c:5b:13:46:c6:76:78:93:83:26:f1:3f:50:8e:d8:80:d5:43:
        e0:ea:66:4b:f9:6a:af:bb:bb:46:0b:9b:02:50:dc:16:b3:7c:
        3c:06:a3:11:fb:09:dd:a5:ac:4e:ba:b1:63:a5:1c:8a:32:a9:
        be:53:7a:38:eb:c4:8a:48:d0:96:ae:74:09:8c:05:d2:74:47:
        36:95:80:99:0c:cd:e1:6f:40:b3:5a:da:68:5e:d4:3a:4b:39:
        6a:fc:70:87:e3:1f:bf:41:2f:7f:2a:e6:98:82:db:0e:63:e0:
        e2:ec:b7:e2:07:26:d8:67:49:16:5a:64:17:3f:53:42:98:82:
        68:4f:0e:00:28:77:ac:ed:dc:de:8f:d6:47:7c:b0:cf:0e:c9:
        a8:f8:63:52:3d:8a:f0:c6:f3:73:12:f1:a8:0b:4e:57:7b:2a:
        9a:b3:fa:6d:4c:0f:05:60:85:dc:ac:93:6f:6c:a7:ad:f3:84:
        fc:d5:3b:24:16:4e:24:c0:44:11:bf:79:79:41:a7:56:27:7e:
        a3:27:26:df:f1:35:c9:ab:f9:74:a1:12:7a:c3:31:ca:4e:db:
        1e:76:0d:c8:36:98:ab:95:8a:04:5e:14:f3:c8:0c:e1:cd:11:
        05:09:46:8b:16:ba:9c:e3:ca:e0:bd:0d:28:4f:4a:7b:c2:a0:
        52:00:95:f3
```

```
$ openssl rsa -noout -text -in
server.key
Enter pass phrase for server.key:
Private-Key: (2048 bit)
modulus:
    00:d7:9d:bb:ad:4c:0b:64:11:6f:6c:1a:8f:cf:78:
    c2:4a:24:32:d1:1d:b7:91:df:be:ce:1d:bf:01:39:
    6d:9d:f0:44:49:e1:a3:6c:9d:46:73:66:8a:d0:76:
    7d:36:e8:dc:82:d7:e6:59:b6:99:cc:f8:e9:e8:09:
    aa:40:4e:c4:6c:44:d7:41:8a:4a:98:8e:0a:62:0c:
    d1:cb:41:9d:7a:13:de:00:4a:f5:a7:32:e9:43:6b:
    ae:0a:36:5b:48:94:d8:94:ca:7e:bc:b6:ef:37:af:
    80:3d:02:ec:bb:dc:83:7a:ce:da:2d:6d:b8:b1:30:
    34:02:93:1e:4b:eb:98:13:2f:61:4b:53:b3:03:52:
    cb:39:c6:e3:64:57:92:8c:5b:03:6e:84:73:d0:87:
    a2:14:82:98:57:6f:4a:e0:2d:ca:aa:fe:9e:bd:13:
    40:47:23:68:e4:89:47:45:50:33:01:9b:7d:6d:3f:
    bb:50:8f:31:6a:67:7e:10:82:b9:a9:f5:41:dc:31:
    2e:39:b2:1f:a6:0b:ca:8c:a2:58:28:72:2e:13:5c:
    6b:cd:a8:6b:6b:79:eb:81:26:63:05:fb:38:98:01:
    85:f9:bc:e7:6b:51:69:f6:8d:6c:9c:63:33:11:10:
    2b:e7:40:42:09:43:5f:2d:46:ca:ed:ce:af:0c:af:
    44:ad
publicExponent: 65537 (0x10001)
privateExponent:
    41:26:d5:5e:01:1b:74:0a:5c:ab:c2:be:ef:c7:22:
    96:3a:a7:ec:4e:59:78:c7:ae:25:24:11:e2:31:d3:
    30:a5:38:4d:46:d4:15:ee:d4:29:ec:b3:47:58:76:
    6f:90:1f:89:9d:e9:69:f4:66:36:ec:83:e9:6c:7a:
    38:62:54:b2:0e:7f:28:bb:bb:dc:ab:16:f2:17:c3:
    90:f2:6b:be:46:a0:8f:60:17:28:85:96:bc:9b:9e:
    04:51:f6:75:51:1c:bc:a1:0f:78:c0:a2:3a:26:5a:
    ce:94:c2:a9:e5:71:09:3b:d5:eb:62:3b:2b:b7:50:
    0f:f3:1a:75:80:63:fc:6f:87:7e:3a:ef:33:4b:bd:
    ba:b3:9a:34:92:94:d7:bf:83:05:4d:4c:4d:5a:7a:
    03:eb:bb:96:28:40:9b:45:c4:b4:5c:68:20:fd:98:
    c3:8f:16:c0:3c:11:01:86:ff:ae:0b:41:ba:45:76:
    2e:f6:49:85:32:a1:cb:b6:41:f4:16:56:cb:ad:92:
    b4:84:a5:e9:e1:03:05:28:3d:d5:55:28:07:02:28:
    35:a7:bb:57:55:01:45:a7:39:f2:cf:78:21:50:18:
    09:44:bf:e2:29:e1:1c:85:e3:80:26:eb:26:d4:d6:
    e8:de:00:ec:24:57:15:90:31:5a:69:80:1f:06:64:
    81
prime1:
    00:f7:83:4e:89:a9:ca:ad:fe:50:02:ca:c8:f0:70:
    d9:02:24:d9:7e:1c:7c:7c:b1:b8:85:ed:b1:41:80:
    e8:30:9e:4b:5a:15:ab:ea:e9:76:fd:30:9c:4d:96:
    d3:46:c8:6a:f3:b4:9a:3d:b8:09:1e:e0:31:b8:c8:
    3d:94:15:66:60:bf:6b:b7:a6:22:d5:ad:6f:a5:c3:
    ed:ad:41:a2:24:1e:a3:b3:7d:df:a3:70:ea:82:b3:
    64:2d:64:b5:8b:cb:61:87:9d:66:ed:e9:b8:40:11:
    ba:c0:0d:06:f8:70:55:b8:26:63:24:af:0c:b6:7e:
    60:1a:46:82:da:d5:9f:a7:21
prime2:
    00:df:02:6e:ea:1b:32:64:88:a2:e2:18:de:49:3d:
    9e:54:f1:65:56:99:21:b6:5f:ad:57:40:a5:fe:ac:
    a9:b9:68:de:e7:8f:1d:e2:2e:6d:0f:51:e6:6b:05:
    52:21:6e:87:52:76:88:9a:e5:86:98:a0:fc:6c:d0:
    ec:9c:9f:a8:af:0e:ba:3c:c0:52:90:4d:4c:41:0a:
    7d:53:66:e8:e4:bc:6b:ae:e3:da:d0:99:b7:91:1a:
    48:31:8b:61:7b:82:9c:f3:7f:3e:0e:9e:21:ca:01:
    b0:8e:fb:52:ee:2a:78:1c:c8:35:23:bc:92:fe:f6:
    23:dc:ff:d6:60:92:c2:c8:0d
exponent1:
    74:a4:9f:cd:86:83:ea:ad:6f:bd:71:1d:73:1a:6b:
    5a:74:4d:3e:fe:63:b8:4b:f4:be:c0:fe:88:f5:1b:
    f7:55:92:03:39:35:54:b6:83:89:dc:6f:bc:ac:f8:
    92:61:fe:ed:2c:ce:87:89:84:5a:d5:a2:f9:06:fc:
    e6:1d:93:aa:c2:6b:1c:18:22:50:7a:b4:a3:f5:0a:
    bc:5c:b2:f1:bc:b2:be:f2:f2:02:cf:42:e5:27:6f:
    6d:69:09:99:80:d6:4c:97:e4:1a:f8:cb:08:fe:91:
    f8:d1:ea:d8:07:f0:8d:a1:21:95:f7:1c:d1:a0:0a:
    e0:37:1c:91:ce:9c:b7:a1
exponent2:
    0d:0f:59:6b:80:58:3d:26:fb:52:fe:5d:d6:30:33:
    9b:89:df:83:68:c8:5e:a3:cf:c4:f2:56:46:49:da:
    4e:af:63:8e:70:05:31:ff:c2:07:49:a6:92:d1:e6:
    f3:6a:43:a6:82:a6:91:5b:ab:bc:38:81:4f:e6:14:
    55:3f:cc:63:24:1b:a7:ff:23:56:ac:10:31:26:ea:
    1b:fe:44:d9:50:67:86:00:76:0a:0d:56:80:ba:e7:
    4b:6f:7c:2f:fd:80:2f:8a:5e:1a:01:0c:bd:85:c7:
    cf:37:cc:ad:81:f3:32:cc:4e:c0:5f:04:c4:c5:a9:
    68:01:db:8e:20:4a:23:e9
coefficient:
    00:f2:c2:a7:25:1c:63:ed:c7:29:c5:9e:ee:a9:dc:
    95:b3:db:e7:e8:48:9f:61:e5:10:e8:b5:ff:dc:73:
    19:eb:c2:49:1b:c2:d2:7b:05:fe:c0:76:98:20:94:
    a1:7d:0f:78:2e:5c:20:e3:06:62:ae:61:87:e1:c5:
    35:a4:36:6a:a2:a8:48:93:fb:c0:07:17:22:c1:05:
    46:ad:c1:b1:b6:24:90:cb:fc:b2:87:91:6a:8e:99:
    b3:39:59:f0:58:6a:13:35:66:38:62:8e:55:0a:4b:
    a7:21:96:8e:8f:88:8b:fd:05:af:73:ab:30:2b:e0:
    5a:3c:39:89:b8:71:3b:2e:65
```

Photo by Guillaume Bolduc on Unsplash

Inspecting ECDSA CSR, EC Params, Private Key

- **Certificate Request**

```
openssl req -noout \
-text -in server.csr
```

- **Elliptic Curve Parameters**

```
openssl ecparam -noout \
-text —in server.pem \
-param_enc explicit
```

- **Private Key**

```
openssl ec -noout \
-text -in server.key
```

```
$ openssl req -noout -text -in server.csr
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=US, ST=Missouri, L=Springfield, O=SGF AWS,
CN=www.example.org/emailAddress=hostmaster@example.org
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:18:95:b6:3a:8a:ee:aa:12:98:2b:c7:98:ab:fa:
                    4e:c5:d5:95:4f:99:16:c1:c9:53:c8:b2:0f:54:52:
                    87:e1:4f:ec:64:ab:8a:33:6e:9d:37:60:15:32:aa:
                    06:61:4a:97:e8:49:78:c6:91:01:5c:3f:c3:90:a6:
                    52:40:4f:e3:a3
                Field Type: prime-field
                Prime:
                    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
                    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
                    ff:ff:ff
                A:
                    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
                    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
                    ff:ff:fc
                B:
                    5a:c6:35:d8:aa:3a:93:e7:b3:eb:bd:55:76:98:86:
                    bc:65:1d:06:b0:cc:53:b0:f6:3b:ce:3c:3e:27:d2:
                    60:4b
                Generator (uncompressed):
                    04:6b:17:d1:f2:e1:2c:42:47:f8:bc:e6:e5:63:a4:
                    40:f2:77:03:7d:81:2d:eb:33:a0:f4:a1:39:45:d8:
                    98:c2:96:4f:e3:42:e2:fe:1a:7f:9b:8e:e7:eb:4a:
                    7c:0f:9e:16:2b:ce:33:57:6b:31:5e:ce:cb:b6:40:
                    68:37:bf:51:f5
                Order:
                    00:ff:ff:ff:ff:00:00:00:00:ff:ff:ff:ff:ff:ff:
                    ff:ff:bc:e6:fa:ad:a7:17:9e:84:f3:b9:ca:c2:fc:
                    63:25:51
                Cofactor:  1 (0x1)
                Seed:
                    c4:9d:36:08:86:e7:04:93:6a:66:78:e1:13:9d:26:
                    b7:81:9f:7e:90
        Attributes:
            a0:00
    Signature Algorithm: ecdsa-with-SHA256
         30:44:02:20:1c:29:5f:2a:1a:7f:48:3f:e3:9c:6e:81:42:a0:
         10:5c:28:f6:c4:c6:00:a3:dd:b9:8f:fa:37:18:a9:d3:f4:8e:
         02:20:30:d1:ff:3e:51:02:fa:57:af:ea:b2:9f:0c:9f:a7:97:
         b1:5e:d0:d5:8d:76:ab:08:61:55:e9:91:db:3e:02:2a
```

```
$ openssl ecparam -noout -text \
-in server.pem -param_enc explicit
Field Type: prime-field
Prime:
    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:ff:ff
A:
    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:ff:fc
B:
    5a:c6:35:d8:aa:3a:93:e7:b3:eb:bd:55:76:98:86:
    bc:65:1d:06:b0:cc:53:b0:f6:3b:ce:3c:3e:27:d2:
    60:4b
Generator (uncompressed):
    04:6b:17:d1:f2:e1:2c:42:47:f8:bc:e6:e5:63:a4:
    40:f2:77:03:7d:81:2d:eb:33:a0:f4:a1:39:45:d8:
    98:c2:96:4f:e3:42:e2:fe:1a:7f:9b:8e:e7:eb:4a:
    7c:0f:9e:16:2b:ce:33:57:6b:31:5e:ce:cb:b6:40:
    68:37:bf:51:f5
Order:
    00:ff:ff:ff:ff:00:00:00:00:ff:ff:ff:ff:ff:ff:
    ff:ff:bc:e6:fa:ad:a7:17:9e:84:f3:b9:ca:c2:fc:
    63:25:51
Cofactor:  1 (0x1)
Seed:
    c4:9d:36:08:86:e7:04:93:6a:66:78:e1:13:9d:26:
    b7:81:9f:7e:90
```

```
$ openssl ec -noout -text \
-in server.key
read EC key
Enter PEM pass phrase:
Private-Key: (256 bit)
priv:
    2e:91:84:39:03:fb:55:92:6f:94:21:09:9c:f4:f8:
    b8:40:02:63:1b:83:f7:76:58:76:c0:42:26:9a:c5:
    a4:55
pub:
    04:18:95:b6:3a:8a:ee:aa:12:98:2b:c7:98:ab:fa:
    4e:c5:d5:95:4f:99:16:c1:c9:53:c8:b2:0f:54:52:
    87:e1:4f:ec:64:ab:8a:33:6e:9d:37:60:15:32:aa:
    06:61:4a:97:e8:49:78:c6:91:01:5c:3f:c3:90:a6:
    52:40:4f:e3:a3
Field Type: prime-field
Prime:
    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:ff:ff
A:
    00:ff:ff:ff:ff:00:00:00:01:00:00:00:00:00:00:
    00:00:00:00:00:00:ff:ff:ff:ff:ff:ff:ff:ff:ff:
    ff:ff:fc
B:
    5a:c6:35:d8:aa:3a:93:e7:b3:eb:bd:55:76:98:86:
    bc:65:1d:06:b0:cc:53:b0:f6:3b:ce:3c:3e:27:d2:
    60:4b
Generator (uncompressed):
    04:6b:17:d1:f2:e1:2c:42:47:f8:bc:e6:e5:63:a4:
    40:f2:77:03:7d:81:2d:eb:33:a0:f4:a1:39:45:d8:
    98:c2:96:4f:e3:42:e2:fe:1a:7f:9b:8e:e7:eb:4a:
    7c:0f:9e:16:2b:ce:33:57:6b:31:5e:ce:cb:b6:40:
    68:37:bf:51:f5
Order:
    00:ff:ff:ff:ff:00:00:00:00:ff:ff:ff:ff:ff:ff:
    ff:ff:bc:e6:fa:ad:a7:17:9e:84:f3:b9:ca:c2:fc:
    63:25:51
Cofactor:  1 (0x1)
Seed:
    c4:9d:36:08:86:e7:04:93:6a:66:78:e1:13:9d:26:
    b7:81:9f:7e:90
```

Photo by Guillaume Bolduc on Unsplash

Comparing Let's Encrypt Certificates to Commercial Certificates

- **Certificate Types** - Let's Encrypt only offers Single Domain, Multi Domain, and Wildcard Certificates [1]. Other vendors also offer Certs for IP addresses [2] [3] and Code Signing [4].
- **Certificate Term** - Let's Encrypt Certificates are **valid for 90 days** and are **automatically renewed** every 60 days. Commercial Certificates are **valid for 1 year** and must be **manually renewed** each year, per Apple and Google. [5]
- **Validation Levels** - Let's Encrypt offers Domain Validation (DV). Commercial Certificates also offer Organization Validation (OV) and Extended Validation (EV).

[1] https://community.letsencrypt.org/t/wildcard-domain-step-by-step/58250/6
[2] https://www.ssl.com/faqs/order-ssl-tls-certificate-for-ip-address/
[3] https://community.letsencrypt.org/t/certificate-for-public-ip-without-domain-name/6082/88
[4] https://community.letsencrypt.org/t/do-you-support-code-signing/370
[5] https://www.godaddy.com/garage/ssl-term-change-2020/

Photo by Elena Mozhvilo on Unsplash

Comparing Let's Encrypt Certificates to Commercial Certificates

(CONTINUED)

- **Warranty** - Some Commercial Certificates offer a warranty (e.g. $10K, $100K, etc) to be paid if you experience a loss related caused by certain Certificate security issues. **Let's Encrypt does NOT offer a warranty. [6]**
- **Trust Site Seal** - Some Commercial Certificates offer a "Site Seal" to be displayed on your website. Let's Encrypt "will never offer a "site seal" that indicates some sort of security guarantee, because they are easy to spoof and confusing to users" [7]

[6] https://community.letsencrypt.org/tos
[7] https://community.letsencrypt.org/t/lets-encrypt-badge-for-websites/6863/12

Photo by Elena Mozhvilo on Unsplash

Comparing Secure Certificate Validation Levels

- **Domain Validation (DV)** - Automated validation that confirms Customer/Client controls the domain name. This could include an email to a WHOIS contact for the domain name, requesting/verifying a specific file be created on the HTTP server for the Certificate FQDN, or requesting/verifying a specific DNS record be created for the Certificate FQDN. **Validation can take as little as a few seconds.**

- **Organization Validation (OV)** - Authenticate organization (e.g. DNB lookup). Verify applicant’s right to request certificates for the organization. Enables **BLUE** address bar on some browsers. **Validation takes 1-2 days.**

- **Extended Validation (EV)** - Same as OV, except more rigorous org validation. Enables **GREEN** address bar on some browsers. **Validation takes 7-10 days.**

Photo by Elena Mozhvilo on Unsplash

Secure Certificate Best Practices

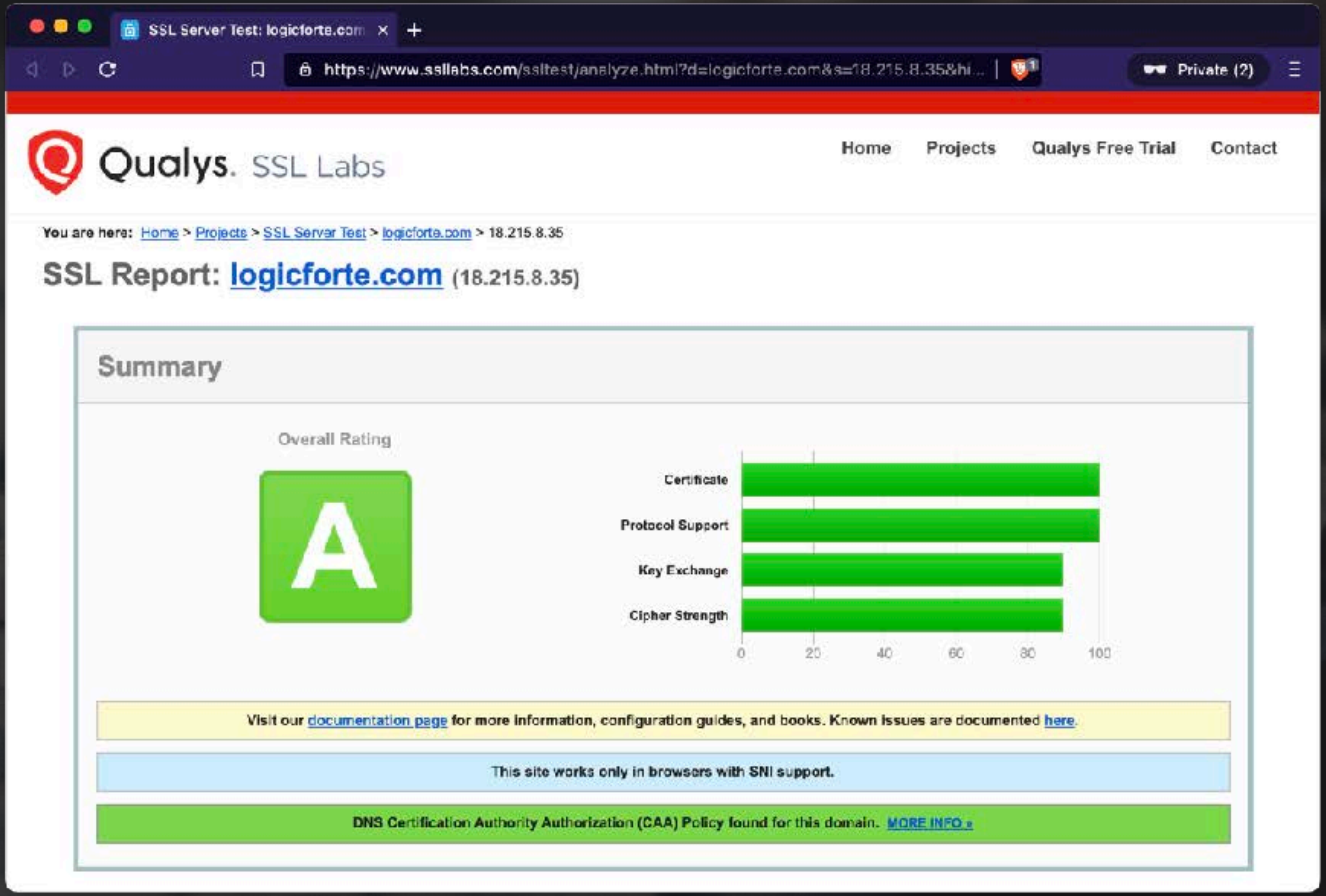
- **Private Key and Certificate** - Key Strength, Protecting Private Keys, Choosing a CA, Signature Algorithms, DNS CAA
- **Configuration** - Certificate Chains, Protocols, Cipher Suites, Forward Secrecy, Key Exchange, Mitigate Known Problems
- **Performance** - Excess Security, Session Resumption, Optimization, HTTP/2, Content Caching, OCSP Stapling, Cryptographic Primitives
- **HTTP and Application Security** - Encrypt Everything, Third-Party Trust, Secure Cookies, Secure Compression, Strict Transport Security, Content Security Policy, Do Not Cache Sensitive Content

Learn More
https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices
https://www.ssl.com/guide/ssl-best-practices/

Photo by Cristina Gottardi on Unsplash

Testing Your Secure Certificate

- Overwhelmed by Best Practices? Not sure where to begin?
- Run an SSL Server Test for your application and resolve any issues that prevent you from scoring an "A" https://www.ssllabs.com/ssltest/
- SSL Server Test Result from a web application hosted by Application Load Balancer (ALB) and protected by an Amazon Certificate



You are here: Home > Projects > SSL Server Test > logicforte.com > 18.215.8.35

SSL Report: logicforte.com (18.215.8.35)

Summary

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This site works only in browsers with SNI support.

DNS Certification Authority Authorization (CAA) Policy found for this domain. MORE INFO »

SSL Server Test Result

Photo by Louis Reed on Unsplash

Secure Certificate Common Issues

- **Using Self-Signed Certificates** - Visitors will receive security errors because your certificate is not trusted by their browser. *Exceptions: Local Development; Internal Corporate Networks*

- **Using an untrusted Certificate Authority** - Same result as above when a browser revokes trust for your chosen Certificate Authority [1] [2] [3]. Let's Encrypt and Amazon should both be safe.

- **Incomplete Certificate Chain** - Does your certificate only work in certain browsers? Some browsers cannot validate your Certificate without the full chain! Ensure your web server or load balancer is sending the entire Certificate Chain. Run an SSL Server Test (www.ssllabs.com/ssltest/) to check for Certificate Chain issues.

[1] https://www.zdnet.com/article/google-bans-another-misbehaving-ca-from-chrome/
[2] https://blog.mozilla.org/security/2018/03/12/distrust-symantec-tls-certificates/
[3] https://thehackernews.com/2017/07/chrome-certificate-authority.html

Photo by Sam Farallon on Unsplash

Secure Certificate Common Issues (continued)

- **Typo in Certificate Signing Request (CSR)** - If you are manually processing your Certificate Request, inspect your CSR and MAKE SURE the subject (FQDN) is spelled correctly BEFORE you upload the CSR to your Secure Certificate vendor.

- **Not Protecting your Private Key File** - When you generate a KEY and CSR, you must be very careful to protect your KEY file! The file should never be accessible to end users. Anyone who has the KEY file can decrypt all of the data transmitted between your server(s) and your users!

- **Forgetting Renewal Dates** - If your Secure Certificate expires, browsers will display a security errors and apps will break [1] [2]. Find a Certificate Management solution that notifies you of upcoming expirations.

Learn More https://programminginsider.com/common-mistakes-to-avoid-while-installing-an-ssl-certificate/

[1] https://www.engadget.com/spotify-us-outage-august-2020-130456478.html

[2] https://techcrunch.com/2020/03/16/microsoft-teams-down/

Photo by Sam Farallon on Unsplash

Secure Certificate Configuration Tips

- **CAA** - Publish DNS CAA records for your domain to restrict which Certificate Authorities can issue certificates for your domain [1].
- **OCSP Stapling** - Configure your web server to serve a signed OCSP response each time it is negotiating a new HTTPS connection, otherwise each visitor must perform an OCSP lookup [2].
- **HTTP Strict Transport Security (HSTS)** - Configure your web server to advertise that your website FQDN (or entire domain name) only accepts HTTPS connections. Browsers will refuse to connect to your website via HTTP. This prevents downgrade attacks and cookie hijacking.
- **Certificate Transparency (CT) Log Monitoring** - Do NOT opt out of CT logs!
  Monitor CT logs for any unauthorized certificates issued for your domains [3].

[1] https://sslmate.com/caa/
[2] https://www.ssl.com/faqs/faq-digital-certificate-revocation/
[3] https://github.com/SSLMate/certspotter

Photo by Janko Ferlič on Unsplash

Secure Certificate Configuration and Testing Resources

- **SSL Configuration Generator** (by Mozilla)
  https://ssl-config.mozilla.org/

- **SSL Website Test** (by Mozilla)
  https://observatory.mozilla.org/

- **SSL Website Test** (by Qualys)
  https://www.ssllabs.com/ssltest/

- **SSL Web Browser Test**
  https://www.howsmyssl.com/

Photo by Janko Ferlič on Unsplash

Questions?

Discuss in our Springfield Devs Slack channel .. #aws

- Follow me! @JasnK @jason-klein

SPRINGFIELD AMAZON WEB SERVICES USER GROUP
FEBRUARY 2021 #SGFAWS

LET'S ENCRYPT 101
SECURE CERTIFICATES FOR WEB SERVICES

Photo by JJ Ying on Unsplash

Thank you!

- Follow me! @JasnK @jason-klein

Springfield Amazon Web Services User Group
February 2021

#SGFAWS

Let's Encrypt 101
Secure Certificates for Web Services

Photo by JJ Ying on Unsplash