

AWS Client VPN で作るリモート接続環境②

2020年9月7日 2020年9月28日

こんにちは。米須です。

今回からリモート環境を構築するための設定について説明していきますが、まずは認証に関する設定から始めていきたいと思います。

目次

1. AWS Client VPN における認証
2. 相互認証の構築
 - 2.1. 1. 証明書の作成
 - 2.2. 2. ACM へのインポート
3. さいごに

AWS Client VPN における認証

認証の方法には下記の3つがあります。

- Active Directory 認証（ユーザベース）
- 相互認証（証明書ベース）
- シングルサインオン（SAML ベースのフェデレーション認証）（ユーザベース）

また、相互認証とその他の認証（Active Directory 認証 or フェデレーション認証）を組み合わせることもできます。相互認証のみだと、証明書の漏洩などがあった場合に容易に入られてしまうので、相互認証 + Active Directory 認証の構成を選択しました。

[AWS ドキュメント]

認証

https://docs.aws.amazon.com/ja_jp/vpn/latest/clientvpn-admin/client-authentication.html

相互認証の構築

1. 証明書の作成

証明書の作成手順は AWS ドキュメントに記載されています（さすが、ドキュメントが充実しています^^）。github から クローンする手順があるので、AWS 上のパブリックサブネットに EC2（Linux）を構築して実施しました。

※ Amazon Linux2 には git が入っていないので、必要に応じて git もインストールしましょう

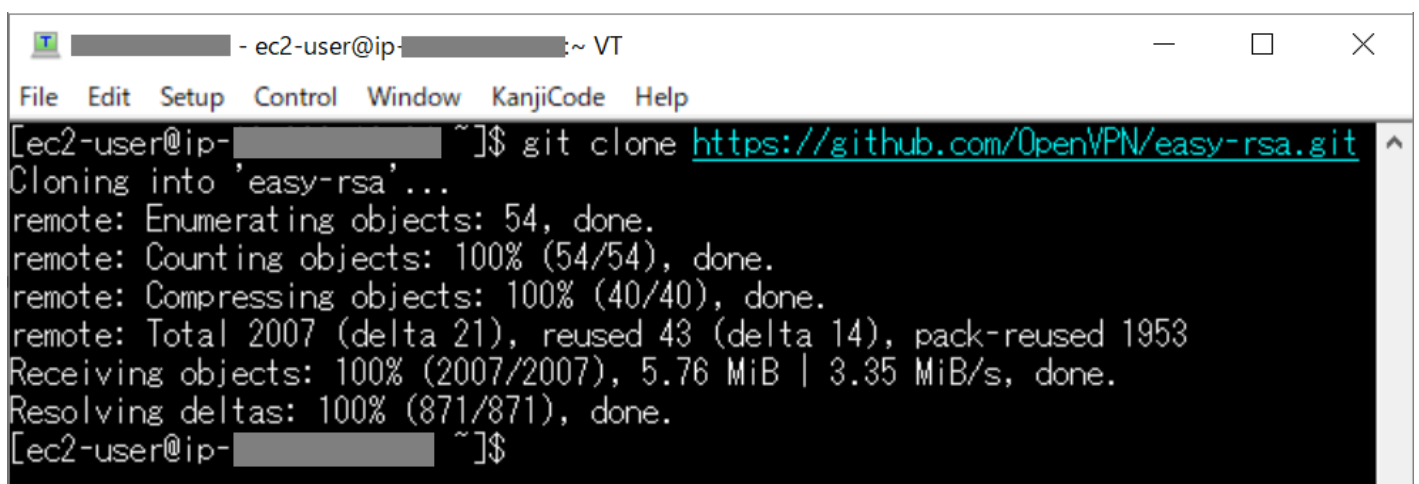
[AWS ドキュメント]

相互認証

https://docs.aws.amazon.com/ja_jp/vpn/latest/clientvpn-admin/client-authentication.html#mutual

ここからは、AWS ドキュメントに沿って実施してみます。

① github から esay-rsa をクローンし、easy-rsa/easyrsa3 フォルダに移動します

A terminal window titled "ec2-user@ip-...:~ VT" with a menu bar (File, Edit, Setup, Control, Window, KanjiCode, Help). The terminal shows the command "git clone https://github.com/OpenVPN/easy-rsa.git" being executed. The output shows the cloning process: "Cloning into 'easy-rsa'...", "remote: Enumerating objects: 54, done.", "remote: Counting objects: 100% (54/54), done.", "remote: Compressing objects: 100% (40/40), done.", "remote: Total 2007 (delta 21), reused 43 (delta 14), pack-reused 1953", "Receiving objects: 100% (2007/2007), 5.76 MiB | 3.35 MiB/s, done.", "Resolving deltas: 100% (871/871), done.", and the prompt returns to "ec2-user@ip-... ~]\$".

```
ec2-user@ip-... ~]$ git clone https://github.com/OpenVPN/easy-rsa.git
Cloning into 'easy-rsa'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 2007 (delta 21), reused 43 (delta 14), pack-reused 1953
Receiving objects: 100% (2007/2007), 5.76 MiB | 3.35 MiB/s, done.
Resolving deltas: 100% (871/871), done.
ec2-user@ip-... ~]$
```

```
ec2-user@ip-: ~/easy-rsa/easyrsa3 VT
File Edit Setup Control Window KanjiCode Help
[ec2-user@ip- ~]$ cd easy-rsa/easyrsa3
[ec2-user@ip- easyrsa3]$ pwd
/home/ec2-user/easy-rsa/easyrsa3
[ec2-user@ip- easyrsa3]$
```

② PKI 環境を初期化します

```
ec2-user@ip-: ~/easy-rsa/easyrsa3 VT
File Edit Setup Control Window KanjiCode Help
[ec2-user@ip- easyrsa3]$ ./easyrsa init-pki

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/ec2-user/easy-rsa/easyrsa3/pki

[ec2-user@ip- easyrsa3]$
```

③ 新しい認証機関 (CA) を構築します

※ Common Name はデフォルト値にしました。

```
ec2-user@ip-: ~/easy-rsa/easyrsa3 VT
File Edit Setup Control Window KanjiCode Help
[ec2-user@ip- easyrsa3]$ ./easyrsa build-ca nopass
Using SSL: openssl OpenSSL 1.0.2k-fips 26 Jan 2017
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
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.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/ec2-user/easy-rsa/easyrsa3/pki/ca.crt

[ec2-user@ip- easyrsa3]$
```

④ サーバー証明書とキーを生成します

```
ec2-user@ip-: ~/easy-rsa/easyrsa3 VT
File Edit Setup Control Window KanjiCode Help
[ec2-user@ip- easyrsa3]$ ./easyrsa build-server-full server nopass
Using SSL: openssl OpenSSL 1.0.2k-fips 26 Jan 2017
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/home/ec2-user/easy-rsa/easyrsa3/pki/easy-rsa-9031.D63wd0/tmp.6vHyJm'
-----
Using configuration from /home/ec2-user/easy-rsa/easyrsa3/pki/easy-rsa-9031.D63wd0/tmp.SY1Gqh
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'server'
Certificate is to be certified until Dec  4 05:22:31 2022 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

[ec2-user@ip- easyrsa3]$
```

⑤ クライアント証明書とキーを生成します

```
ec2-user@ip-: ~/easy-rsa/easyrsa3 VT
File Edit Setup Control Window KanjiCode Help
[ec2-user@ip- easyrsa3]$ ./easyrsa build-client-full client1.domain.tld nopass
Using SSL: openssl OpenSSL 1.0.2k-fips 26 Jan 2017
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/home/ec2-user/easy-rsa/easyrsa3/pki/easy-rsa-9119.oMhCqK/tmp.wG014n'
-----
Using configuration from /home/ec2-user/easy-rsa/easyrsa3/pki/easy-rsa-9119.oMhCqK/tmp.I91Qux
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'client1.domain.tld'
Certificate is to be certified until Dec  4 05:23:25 2022 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

[ec2-user@ip- easyrsa3]$
```

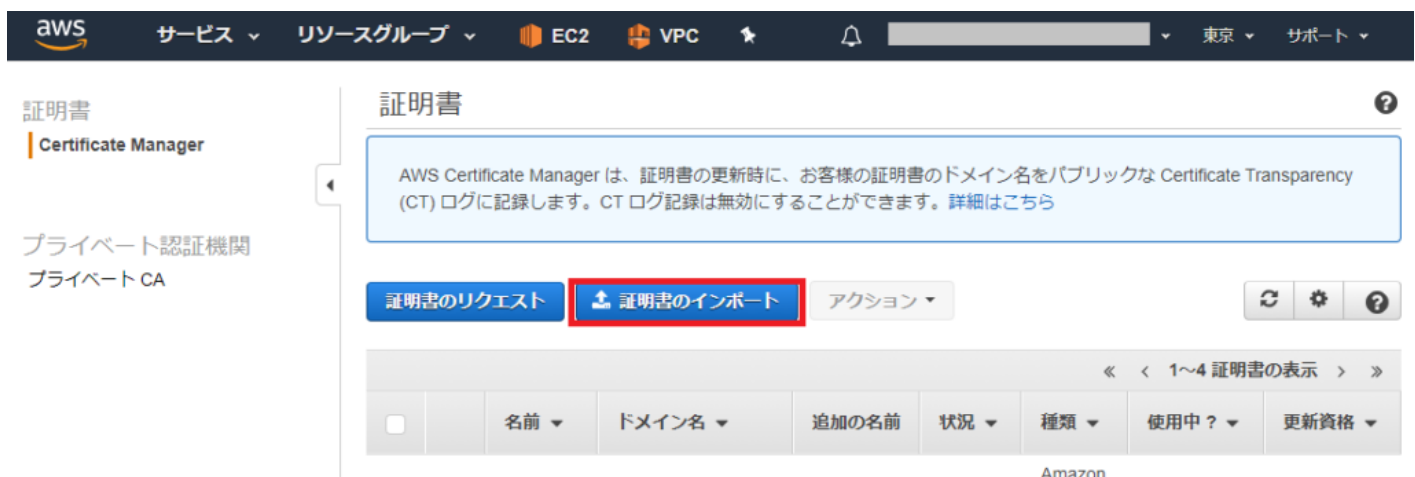
ここまでの手順で下記のファイルが作成されます。

- pki/ca.crt
- pki/issued/server.crt
- pki/private/server.key
- pki/issued/client1.domain.tld.crt
- pki/private/client1.domain.tld.key

2. ACM へのインポート

サーバー証明書とキー、およびクライアント証明書とキーを ACM にインポートします。AWS ドキュメントのように CLI で実行することもできますが、マネコンからインポートすることもできます。

まず、AWS Certificate Manager を開き、「証明書のインポート」ボタンを押下します。



証明書の作成のところで作成した server.crt、server.key、ca.crt のファイルの中身を、各項目に貼り付け、「次へ」ボタンを押下します。ファイルの内容を貼り付ける際は、「--BEGIN PRIVATE KEY--」や「--END PRIVATE KEY--」もそのまま貼り付けます。

aws

サービス

リソースグループ

EC2

VPC

★

🔔

東京

サポート

証明書のインポート

ステップ 1: 証明書のインポート

ステップ 2: タグを追加

ステップ 3: レビューとインポート

AWS Certificate Manager の証明書は、他の AWS サービスで使用できます。

証明書の選択

PEM エンコードされた証明書本文、プライベートキー、および証明書チェーンを下に貼り付けます。[詳細はこちら](#)

証明書本文*

server.crt を貼り付ける

証明書のプライベートキー

server.key を貼り付ける

証明書チェーン

ca.crt を貼り付ける

* 必須

キャンセル

次へ

必要に応じてタグをつけて、「レビューとインポート」ボタンを押下します。

証明書のインポート

ステップ 1: 証明書のインポート

ステップ 2: タグを追加

ステップ 3: レビューとインポート

タグを追加

証明書の管理に役立つように、オプションで各リソースには独自のメタデータをタグの形式で割り当てることができます。[詳細はこちら](#)

タグ名	値
<input type="text" value="Tag Name"/>	<input type="text" value="Value"/>

タグを追加

キャンセル

戻る

レビューとインポート

最後に「インポート」ボタンを押下します。

<https://www.ryucom.co.jp/blog/aws/278>

6/8

証明書のインポート

ステップ 1: 証明書のインポート

ステップ 2: タグを追加

ステップ 3: レビューとインポート

レビューとインポート



ドメイン	server
有効期限	675 日
パブリックキー情報	RSA-2048
署名アルゴリズム	SHA256WITHRSA
証明書本文*	<div></div>
証明書のプライベートキー*	<div>-----BEGIN PRIVATE KEY----- <div></div></div>
証明書チェーン	<div>-----BEGIN CERTIFICATE----- <div></div></div>

キャンセル

戻る

インポート

インポートできたことが確認できます。

証明書

Certificate Manager

プライベート認証機関

プライベート CA

証明書

AWS Certificate Manager は、証明書の更新時に、お客様の証明書のドメイン名をパブリックな Certificate Transparency (CT) ログに記録します。CT ログ記録は無効にすることができます。詳細はこちら

証明書のリクエスト

証明書のインポート

アクション



1~2 証明書の表示							
<input type="checkbox"/>	名前	ドメイン名	追加の名前	状況	種類	使用中?	更新資格
<input type="checkbox"/>	-	server	-	発行済み	インポート済み	いいえ	使用不可

状況

同様に、クライアント証明書やキーなどをインポートします。ステップ 1 : 証明書のインポートでは、上から順にclient1.domain.tld.crt、client1.domain.tld.key、ca.crt のファイルの内容を貼り付けます。

さいごに

サーバとクライアントの証明書がインポート出来たら、証明書関連の作業は終わりです。この辺りはあまり触った経験がなかったので手順の確認などに時間がかかりましたが、手順自体はそんなに難しくないなので、慣れたらすんなりできそうです。

次回は、Active Directory 認証の設定について説明したいと思います。