

Testablish on AWS チャレンジ

浦郷 央昭

DX 技術本部 DX 技術開発室

Testablish とは

Testablish¹は弊社で開発している Web アプリケーションのテスト自動化支援ツールです。テスト対象となる Web システムに対して Web ブラウザ操作をキャプチャし、その情報をもとにテストケースを GUI 上で作成して、自動テスト用の Selenium テストコードを生成します。

Testablish の本体 (Testablish サーバ) は Docker Compose 上で動作するツールです。Testablish の機能やシステム構成・アーキテクチャなどの詳細は、前号「Web アプリケーション自動テスト支援ツール Testablish の紹介」(Vol.160)²を参照下さい。

利用時の課題

とはいえ、これまでの自分の受諾開発案件の経験からも、Testablish を実際の開発プロジェクトに導入するには、環境面でハードルがあるケースがあると思います。

- Testablish サーバ用のサーバ (筐体) を用意することが難しい(お守りも大変)
- Docker Desktop の有償化に伴い、Windows ベースの開発用 PC を Testablish サーバに流用し辛くなった

¹ <https://www.sra.co.jp/testablish/>

² <https://www.sra.co.jp/Portals/0/files/gletter/pdf/GSletterNeoVol160.pdf>

- 開発中の Web システムや開発者の PC が、閉鎖 LAN やインターネットへのアクセスが極端に制限された環境にある

一方で、昨今ではオンプレ+専用線でのクラウドアクセスによるハイブリッドな閉鎖環境も多いかと思います。クラウドサービス側のプライベートサブネット内であれば、開発チームでも仮想サーバをたてるなど、比較的容易に環境が用意できるケースがあるのではないのでしょうか？

今回は、Testablish サーバ環境の構築をクラウドサービス内（今回は AWS のプライベートサブネット内）で極力完結できる方法が無いか調査してみました。

環境構築

今回構築した環境は以下の通りです。

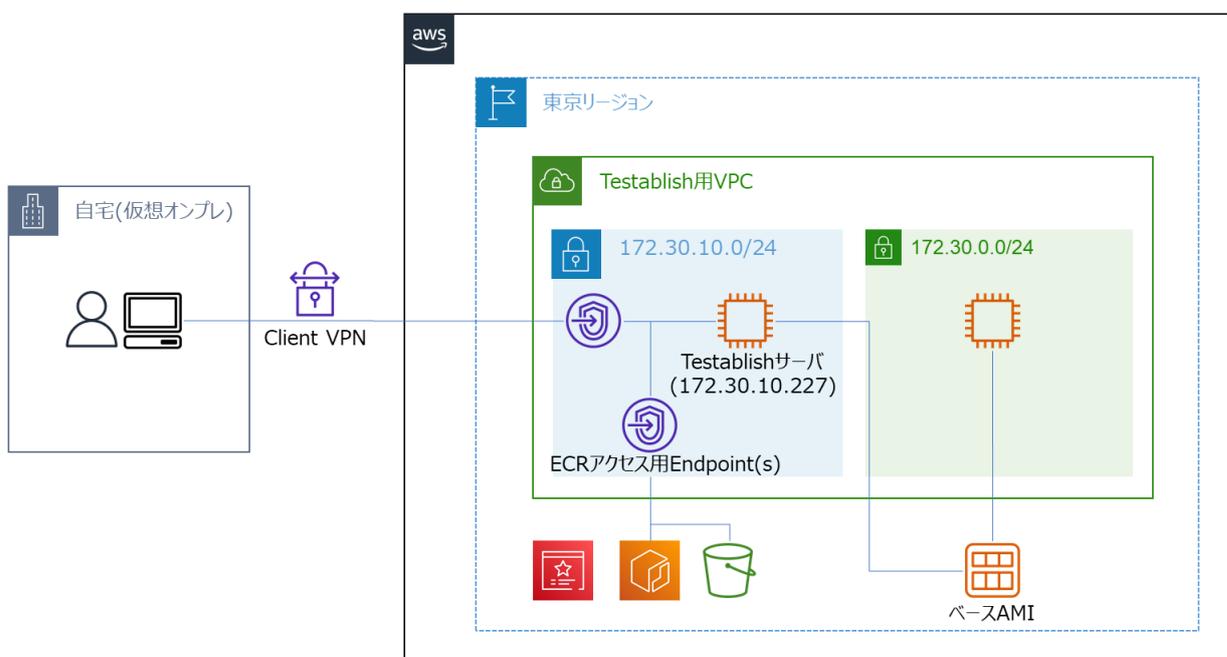


図 1. Testablish サーバ環境

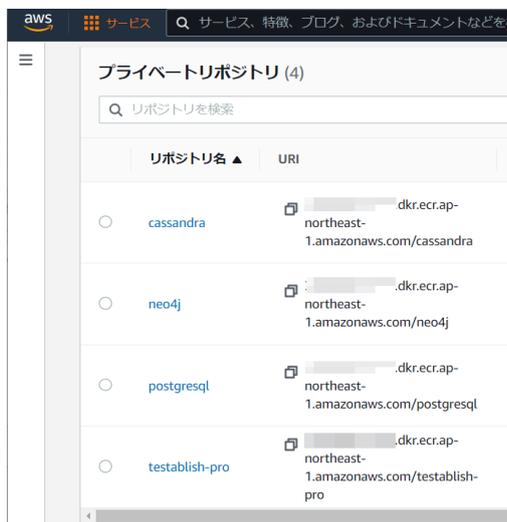
仮想オンプレ環境の用意

自宅 PC を開発 PC とみだてます。オンプレ環境からのアクセスを再現するためとはいえ、さすがに AWS Direct Connect を自宅に引くわけにもいかず、自宅 PC から AWS Client VPN 経由でプライベートサブネットへアクセスできるようにしました。

TESTABLISH 用 ECR リポジトリの用意

AWS ECR に以下のプライベートリポジトリを用意し、それぞれのリポジトリに、必要なイメージを Push しておきます。

- Testablish 用
- その他 Docker 公式イメージ用 (Neo4J、Cassandra、PostgreSQL)



※ AWS の今年 11 月の発表³で、Docker 公式イメージも ECR から取得できるようになりましたが、本記事の執筆時点ではエンドポイント経由で ECR Public Gallery からイメージを取得することができません。従って、一旦 ECR のプライベートリポジトリにアップする必要があります。

またプライベートサブネットから ECR へアクセスできるように、必要なエンドポイントを設定します。



³ <https://aws.amazon.com/blogs/containers/docker-official-images-now-available-on-amazon-elastic-container-registry-public/>

TESTABLISH サーバ (EC2) の用意

Testablish サーバとなる EC2 を用意します。今回は Docker、Docker Compose のインストールを端折るため、パブリックサブネット側で Amazon Linux 2 に Docker、Docker Compose をインストールした EC2 をベースイメージとして、プライベートサブネット側に EC2 インスタンスをたてました。実際の開発現場では、rpm の持ち込みが必要となるなど様々な対応が発生する部分かと思います。

TESTABLISH サーバ構築

プライベートサブネット内で EC2 を起動したら、AWS VPN Client を起動して Client VPN 経由で EC2 にアクセスし操作します。ここからは基本的に Testablish インストールマニュアルに従うのですが、そのままの docker-compose.yml では Docker イメージを ECR から取得できないので、イメージパスを書き換えます。

```
(変更前)
neo4j:
  image: neo4j:3.1

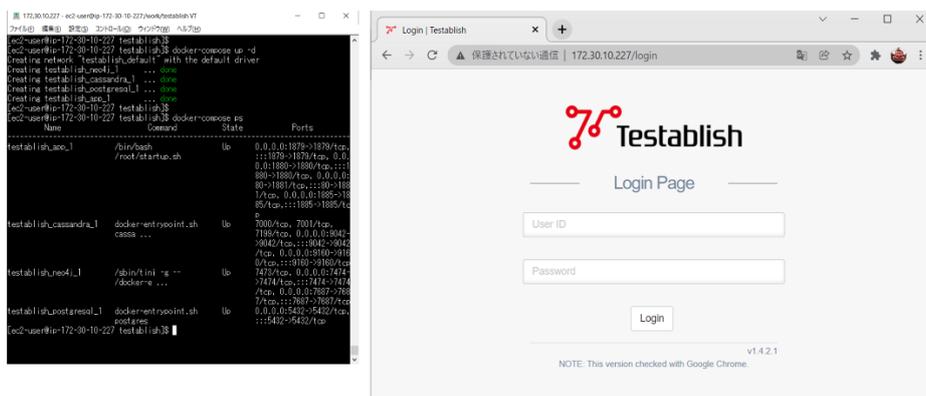
(変更後)
neo4j:
  image: 99999999999.dkr.ecr.ap-northeast-1.amazonaws.com/neo4j:3.1
```

※他のイメージも同様

認証トークンを取得し、レジストリに対して Docker クライアントを認証します。

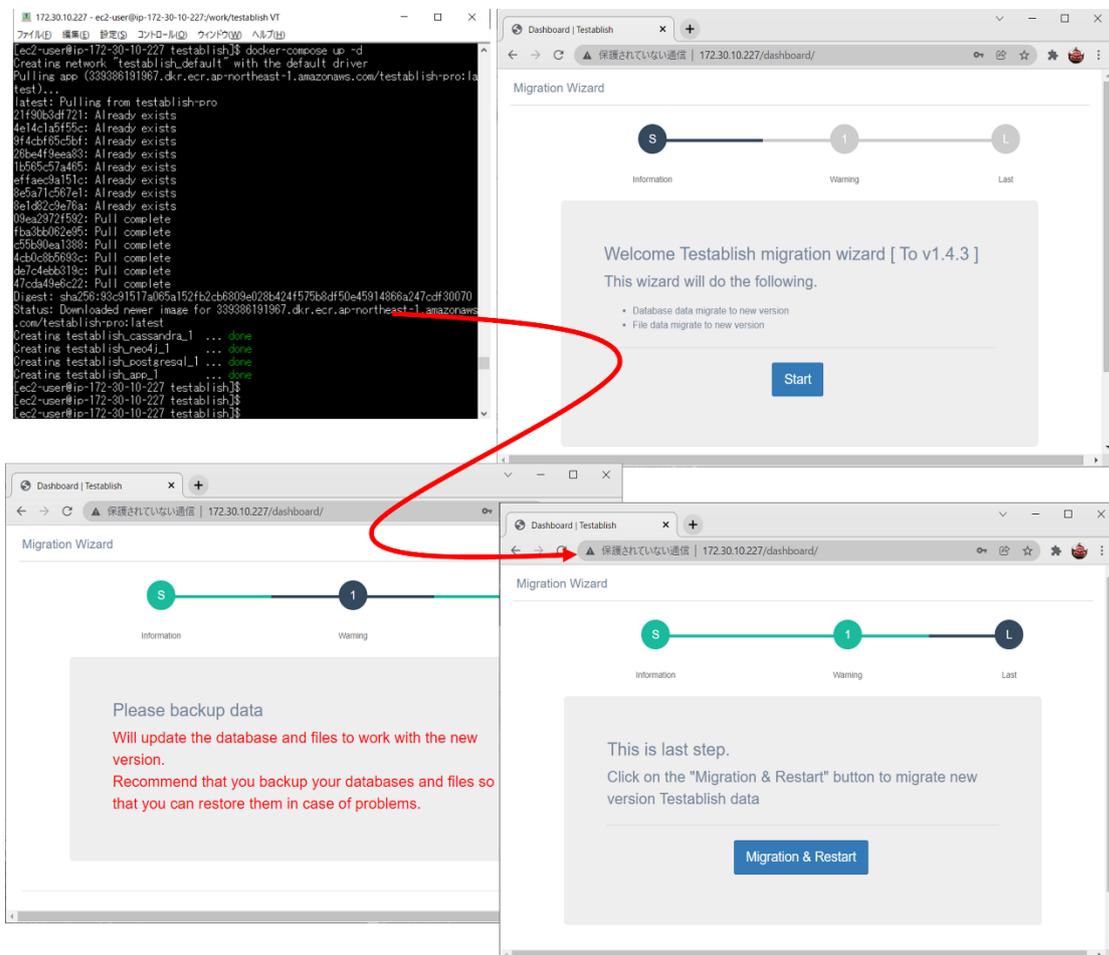
```
> aws ecr get-login-password --region ap-northeast-1 | docker login --username
AWS --password-stdin 99999999999.dkr.ecr.ap-northeast-1.amazonaws.com
... (略) ...
Login Succeeded
```

docker-compose up で ECR から必要イメージを取得し環境が立ち上がります。Client VPN 経由で、プライベートの IP アドレスを指定して Testablish のログインページへアクセスできます。



TESTABLISH のバージョンアップ

こちらも基本的には Testabliish インストールマニュアルに従うのですが、ECR に Testabliish の最新イメージを Push した後、docker-compose.yml のイメージパス（タグ）に注意しながら docker-compose down > docker-compose up とします。Testabliish が最新版に置き換わります。



おわりに

作業の都合で端折った部分もありましたが、今回は完全にクローズドな環境での構築までには至りませんでした。特に Docker 公式イメージは現状ではインターネットから取得するしか方法がなく、AWS 側で対応されることを待ちたいと思います⁴。

また、そもそも開発チームとして EC2 インスタンスを立上げなくとも、AWS であればマネージドサービスである ECS/Fargate を利用した環境とできるかと思えます。

次の機会があれば、より積極的にこれらを利用した環境構築にチャレンジしたいと思います。

⁴ <https://github.com/aws/containers-roadmap/issues/1160>

GSLetterNeo Vol.161

2021年12月20日発行

発行者 株式会社 SRA 先端技術研究所

編集者 熊澤努 方学芬

バックナンバー <https://www.sra.co.jp/public/sra/gsletter/>

お問い合わせ gsneo@sra.co.jp



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation
やわらかいのべーしょん