記事
Tomohiro Iwamoto · 2020年8月25日 6m read

CloudFormationを使ってAWSにICMをデプロイする

本稿について

ICM(InterSystems Cloud Manager)のセットアップは難しいものではありませんが、様々な理由でそもそもDockerが使いづらいという状況があり得ます。
また、セキュリティ的に堅固な環境を得るために、既存VPC内のプライベートサブネット上にIRISクラスタをデプロイする方法のひとつに、同VPC内でICM実行する方法があります。
本稿では、ICMをAWSにデプロイする作業を、CloudFormationで自動化する方法をご紹介します。ICMに関しては、こちらの記事をご覧ください。

更新: 2020年11月24日 デフォルトVPC以外でも動作するよう変更しました。

> ICMは、マルチクラウドに対してIRISクラスタをプロビジョン、アンプロビジョンするためのユーティリティですので、Clou
> dFormationとは機能的に重なる部分があ
> ります。必要であれば、CloudFormationでIRISのミラー構成をプロビジョン
> することも可能です。ここでは割り切って、CloudFormationはICMをデプロイすることだけに利用しています。

また、ICMキット入手方法が今までのWRCサイトからの手動ダウンロードに加えて、InterSystems Container Registryから直接Pullする方法が利用可能となりました。本稿はこれを利用しています。

前提条件

EC2インスタンスをプロビジョンする先としてVPC、サブネット、セキュリティーグループ(SSHがインバウンド可)が存在していること。

事前準備

もしお持ちでなければSSH接続のためのEC2キーペアを作成します。
ICMで使用するAWSリソースの操作に必要なポリシーを持つロールを用意します。本稿ではAmazonEC2FullAccessポリシーを持つロール(iwamoto-ICM-role)を使用しています。

> このロールはICMをデプロイするEC2インスタンスに付与されますので、使用するEC2キーペアの管理にはご注意ください(わざわざ言うまでもありませんね)。

事前準備はこれだけです。
本稿では扱いませんが、同ロールに、AmazonS3ReadOnlyAccessを加えておくと、コンテナレスのデプロイを行う際に使用するIRISのインストーラキット(tar形式)やライセンスキーを自前のS3バケットからダウンロード出来るので、ICMのスクリプト機能を利用して、アプリケーション配布まで含めた完全な自動化を実現できます。

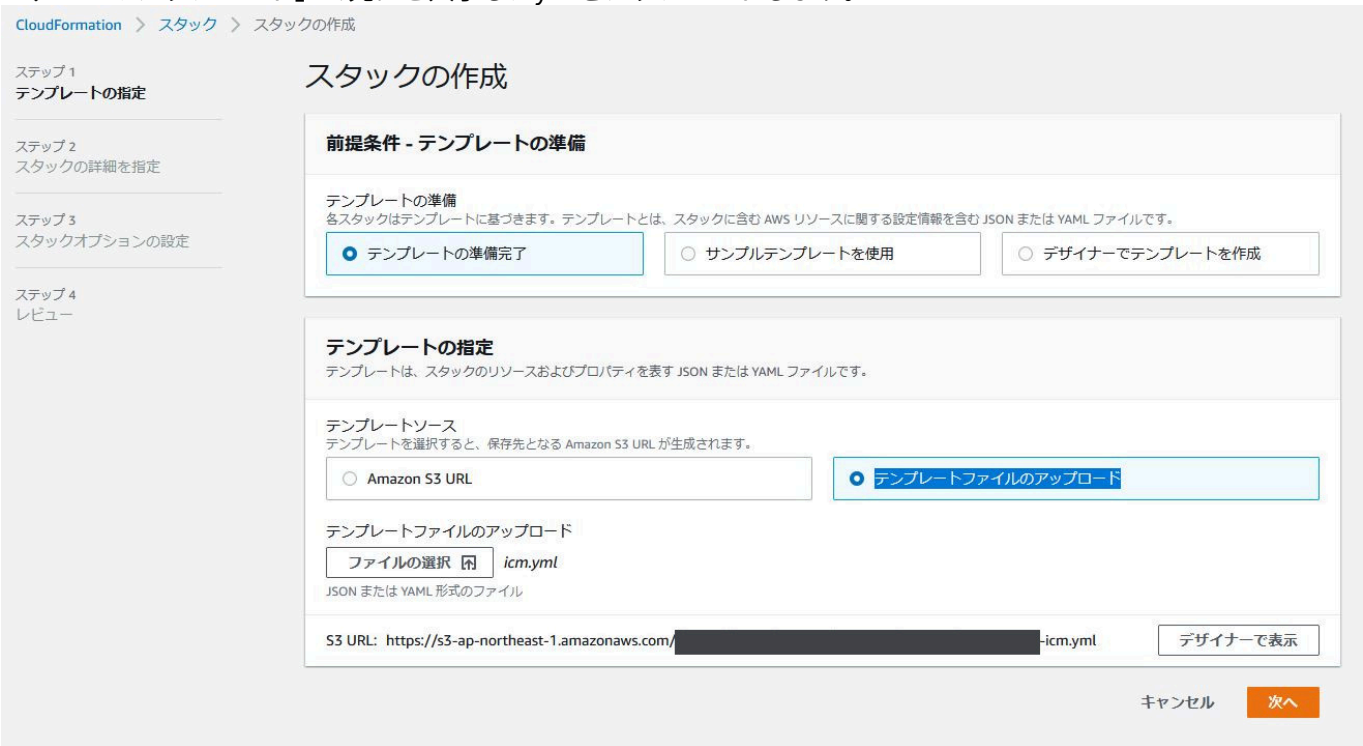
手順

1. CloudFormationテンプレートの入手

こちらから入手できます。

2. スタックの作成

AWS CloudFormationコンソールで「スタックの作成」「新しいリソースを使用」を選択します。「テンプレートファイルのアップロード」で先ほど入手したymlをアップロードします。

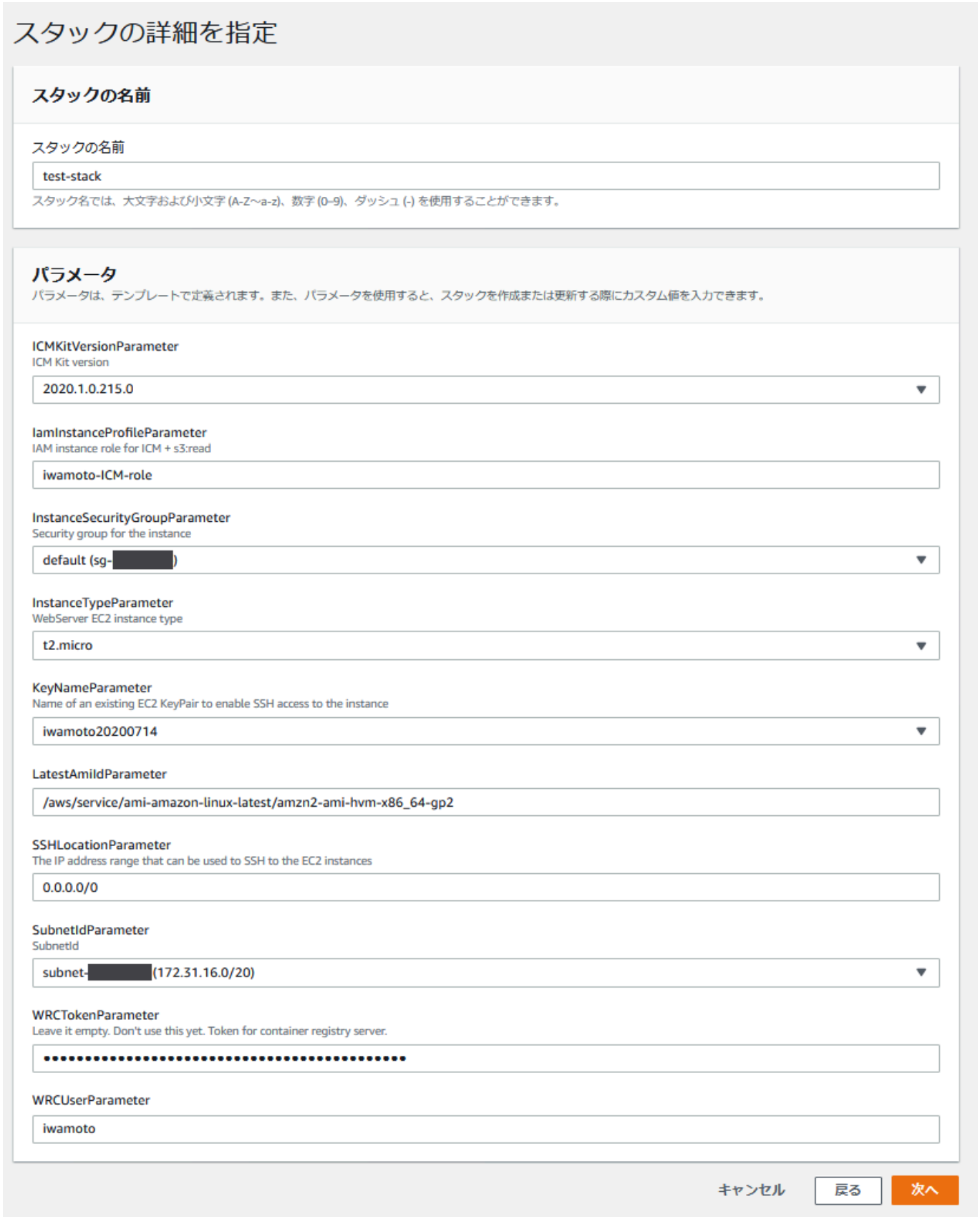


ウィザードに沿って「次へ」進みます。

[スタックの詳細を指定]画面では、各種パラメータを指定します。

スタックの名前	任意の文字列
ICMKitVersionParameter	ICMのキットのバージョン
IamInstanceProfileParameter	事前準備したロール名
InstanceSecurityGroupParameter	EC2インスタンスに適用するセキ サブネットと同じVPCに属してい SSHをインバウント可能にしてく

InstanceTypeParameter	EC2インスタンスタイプ
KeyNameParameter	事前準備したEC2キーペアの名前
LatestAmiIdParameter	使用するAMI(本稿ではAmazon Li
SSHLocationParameter	SSH接続を許可するホストのCID (0.0.0.0/0は全て許可。自分のグロ
SubnetIdParameter	EC2インスタンスを起動するサブ セキュリティーグループと同じVI
WRCTokenParameter	https://containers.intersystems.co
WRCUserParameter	同サイトへのログインに使用した

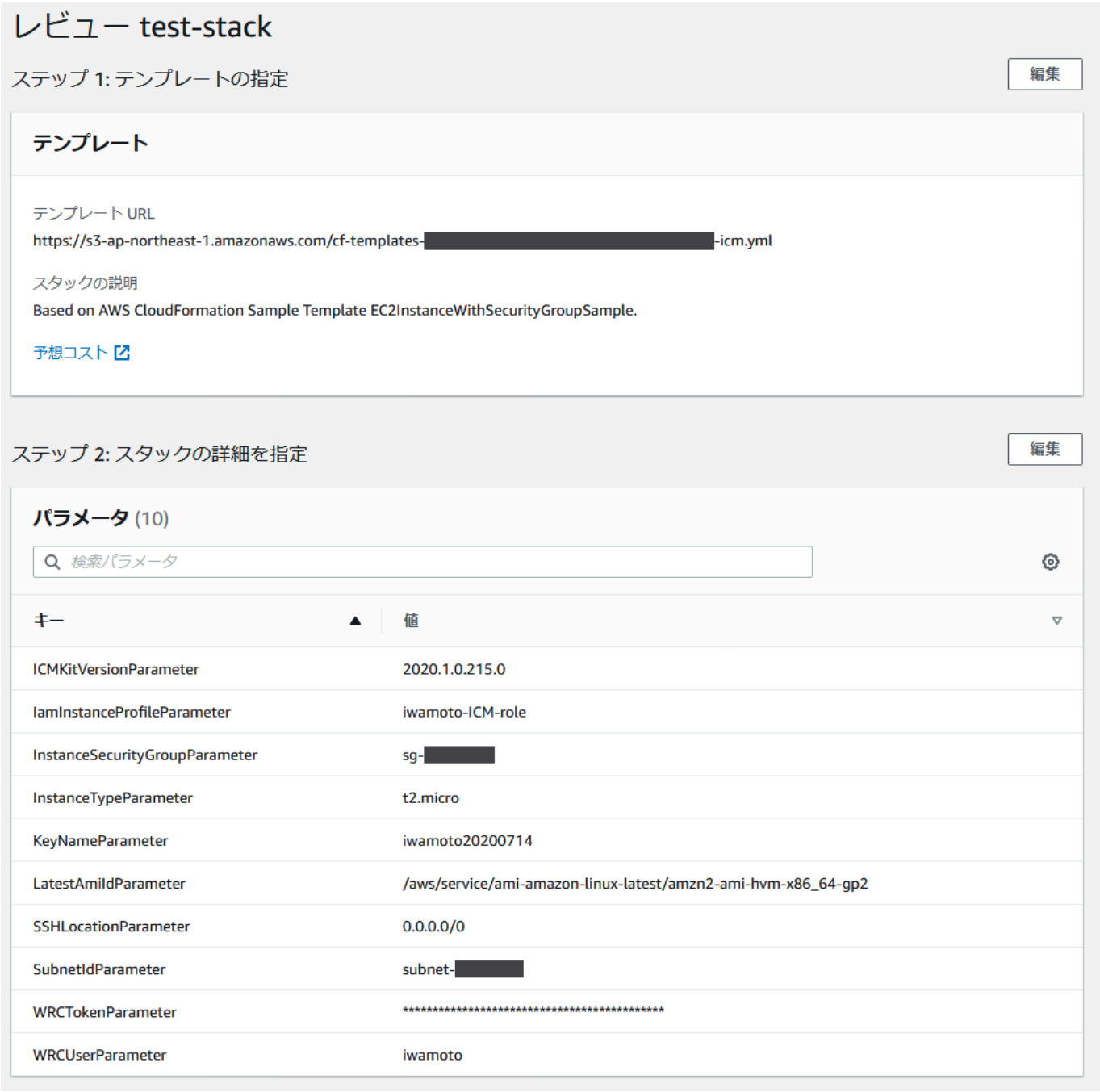


アップロードしたテンプレートファイルは、自動でS3に保存されます。わずかとは言え課金対象ですので、デプロイ後、不要であれば削除します。

ウィザードに沿って「次へ」進みます。

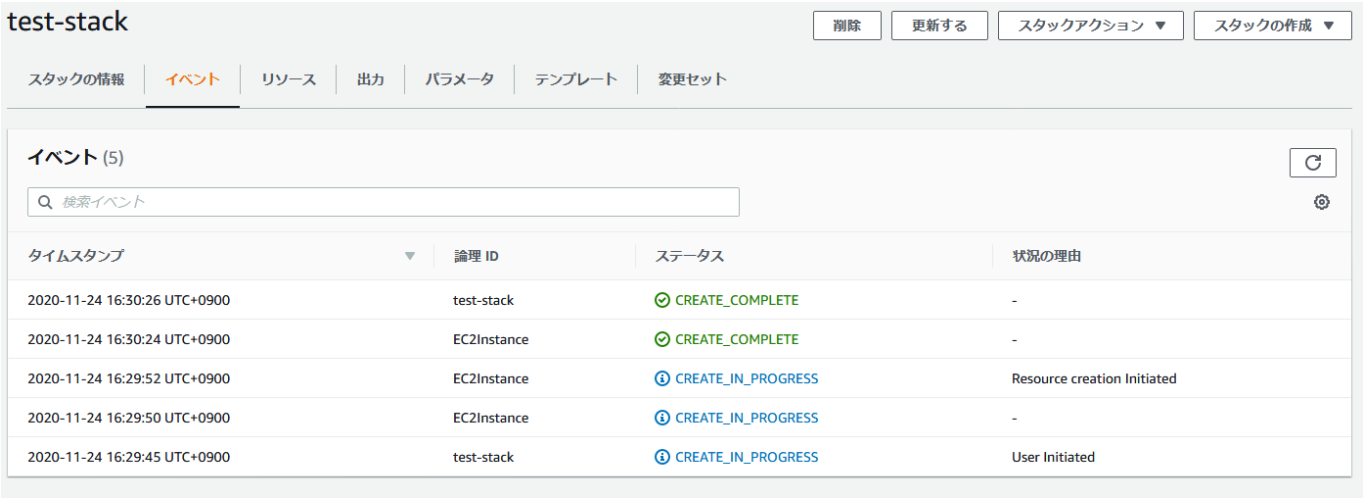
[スタックオプションの設定]はデフォルトのままで構いません。
ウィザードに沿って「次へ」進みます。

[レビュー]画面で値の最終確認を行います。



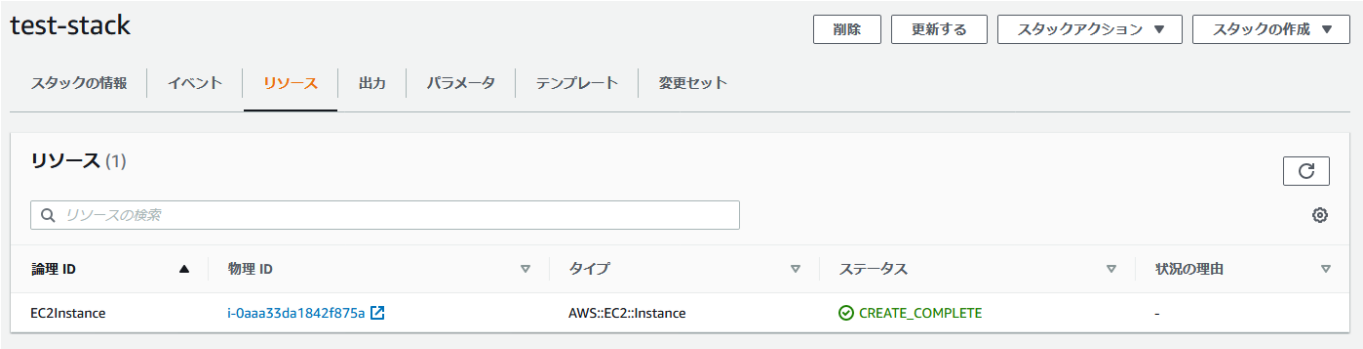
先ほど指定したパラメータに誤りがなければ「スタックの作成」を実行します。

スタック名と実行日時に下にステータスがCREATE_IN_PROGRESSと表示されていると思います。リフレッシュ矢印ボタンを押してステータスがCREATE_COMPLETEになれば完了です。私の場合、1分ほどで完了しました。



> もし、EC2Instanceのステータスが以下のような「状況の理由」でCREATE_FAILEDした場合は、セキュリティーグループとサブネットが同じVPCに属しているか確認してください。
>
> Security group sg-XXXXXXX and subnet subnet-YYYYYY belong to different networks.

[リソース]で、どのようなリソースを作成したのかを確認できます。



[出力]を表示して、PublicDNS値を記録してください。

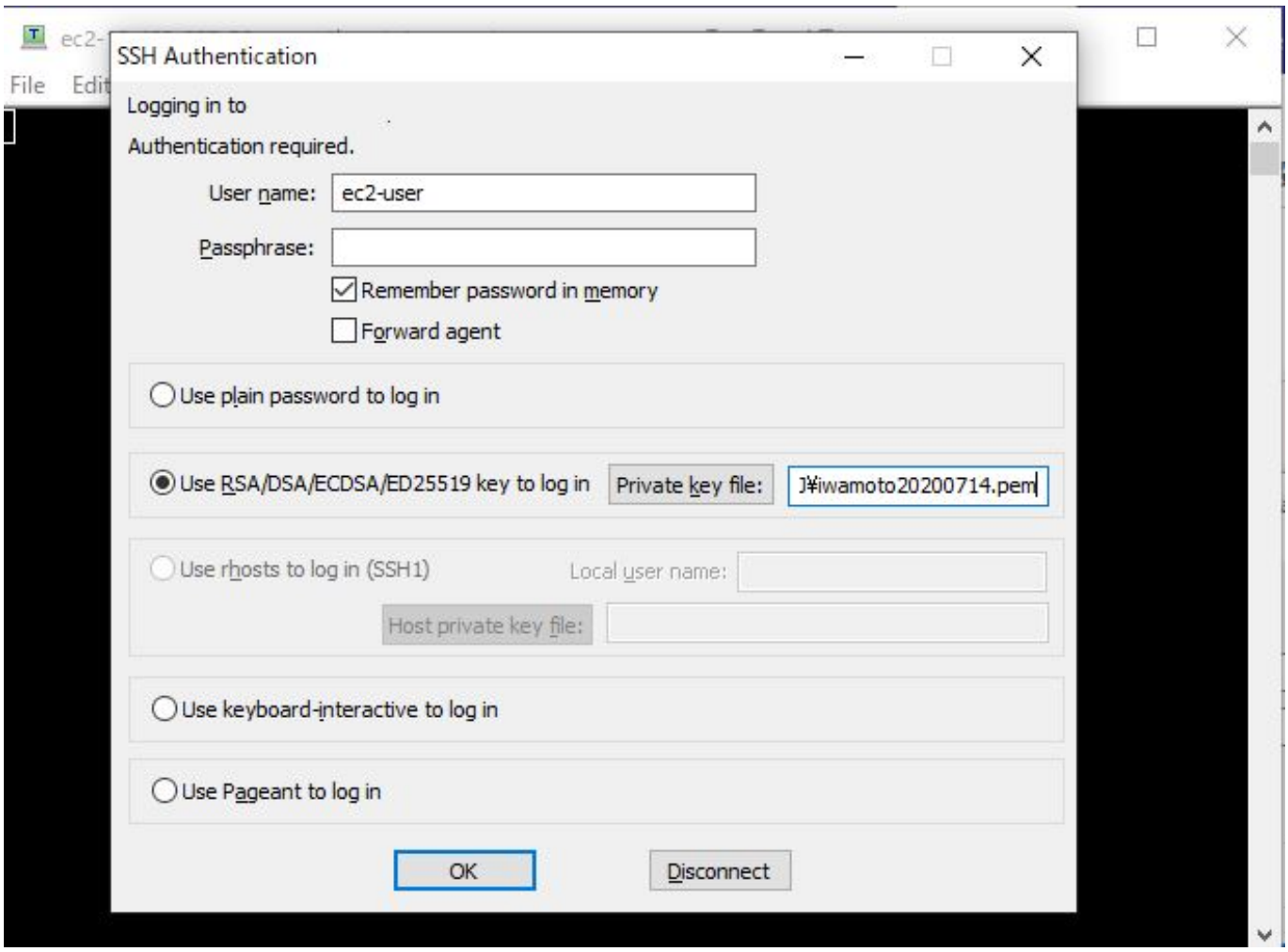
3. 作成されたEC2インスタンスへのログイン

先ほど指定したEC2キーペアを使用して、PublicDNSのホストにsshします。

```
C:\Users\iwamoto>ssh -i iwamoto20200714.pem ec2-user@ec2-18-179-52-235.ap-
northeast-1.compute.amazonaws.com
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-22-166 ~]$
```

あるいは、定番のターミナルエミュレータからですと



ICMがインストールされている事を確認します。イメージ名のレポジトリ名がcontainers.intersystems.com/intersystems/icmになっていることにご注意ください。

```
[ec2-user@ip-172-31-22-166 ~]$ docker images
REPOSITORY                                      TAG                 IMAGE ID
  CREATED             SIZE
containers.intersystems.com/intersystems/icm   2020.1.0.215.0      00d66fae9030
  4 months ago        889MB
[ec2-user@ip-172-31-22-166 ~]$ docker run --rm containers.intersystems.com/intersyste
ms/icm:2020.1.0.215.0 icm version
Version: 2020.1.0.215.0
[ec2-user@ip-172-31-22-166 ~]$
```

AWSのCLIを実行して、指定したロールが備わっていることを確認します。

```
[ec2-user@ip-172-31-22-166 ~]$ aws ec2 describe-
instances --query "Reservations[].Instances[].InstanceId"
[
    "i-0aaa33da1842f875a"
]
```

先ほど確認した[出力]のInstanceIdと同じですね。

ロールにS3へのアクセス権を付与していれば、S3からキットやライセンスキーをダウンロード出来るので便利(なにより別途クレデンシャル情報を取り扱わない分、セキュア)です。

```
[ec2-user@ip-172-31-22-166 ~]$ aws s3 cp s3://mybucket/IRIS-2020.1.0.215.0-lnxrhx64.t
ar.gz .
download: s3://mybucket/IRIS-2020.1.0.215.0-lnxrhx64.tar.gz to ./IRIS-2020.1.0.215.0-
lnxrhx64.tar.gz
```

もし、上記のような結果が得られなければ何か問題が発生しています。下記にログや実行されたシェルが保存されていますので、内容を確認してください。

```
[ec2-user@ip-172-31-22-166 ~]$ cat /var/log/cloud-init-output.log
[ec2-user@ip-172-31-22-166 ~]$ sudo cat /var/lib/cloud/instances/[EC2??????ID]/user-
data.txt
```

以後、ICMの操作を行います。

4.スタックの削除

不要になったら忘れずにリソース(特にEC2インスタンス)を削除しましょう。その際、EC2インスタンス等を個別に削除するのでなく、AWS CloudFormationのコンソールの[削除]で全てのリソースを一括削除します。
[スタックの情報]で、ステータスがDELETE_IN_PROGRESSからDELETE_COMPLETEに変われば削除終了です。

> ICMのEC2インスタンスを削除すると、ICMでプロビジョンしたIRISクラスタへの制御情報も完全に失われますのでご注意ください。

#AWS #クラウド #InterSystems IRIS #InterSystems IRIS for Health

ソースURL:
https://jp.community.intersystems.com/post/cloudformation%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%A6aws%E3%81%ABicm%E3%82%92%E3%83%87%E3%83%97%E3%83%AD%E3%82%A4%E3%81%99%E3%82%8B