
記事

[Tomohiro Iwamoto](#) · 2021年2月16日 19m read

InterSystems IRIS デプロイガイド - AWS CloudFormation テンプレート - Windows編

目的

CloudFormationの記事は、Linux系のものが多いですが、Windowsでも自動化したいという需要がありそうですので、[オリジナル記事](#)を元に、CloudFormationを使用してミラークラスターをWindowsサーバにデプロイする例を実装してみました。また、簡単な実行例も追加しました。ソースコード一式はこちらの[Gitレポジトリ](#)にあります。

更新: 2021年3月1日 ワンライナーで踏み台ホスト経由でWindowsに公開鍵認証する方法を追記しました

更新: 2022年11月29日 [QuickStart](#)の形式に合わせて大幅に変更しました。以前の内容は[こちら](#)に保存してあります。

更新: 2022年12月21日 踏み台ホストの使用を止め、代わりにAWS System Manager(SSM)を有効化しました。プライベートサブネット上のインスタンスへのアクセスが簡素化されます。以前の内容は[こちら](#)に保存してあります。

事前準備

[QuickStart](#)のデプロイ方法に従います。

非公開のS3バケットにIRISのキット及びライセンスキーをアップロードします。

- IRISをWindowsにデプロイする場合

```
S3BucketName=<my bucket>
aws s3 mb s3://$S3BucketName
aws s3 cp iris.key s3://$S3BucketName
aws s3 cp ISCAgent-2022.1.0.209.0-lnxrh7x64.tar.gz s3://$S3BucketName
aws s3 cp IRIS-2022.1.0.209.0-win_x64.exe s3://$S3BucketName
```

- IRISをLinux(AmazonLinuxあるいはRedHat7/x64)にデプロイする場合

```
S3BucketName=<my bucket>
aws s3 mb s3://$S3BucketName
aws s3 cp iris.key s3://$S3BucketName
aws s3 cp ISCAgent-2022.1.0.209.0-lnxrh7x64.tar.gz s3://$S3BucketName
aws s3 cp IRIS-2022.1.0.209.0-lnxrh7x64.tar.gz s3://$S3BucketName
```

- IRISをLinux(Ubuntu/x64)にデプロイする場合

```
S3BucketName=<my bucket>
aws s3 mb s3://$S3BucketName
```

```
aws s3 cp iris.key s3://$S3BucketName
aws s3 cp ISCAgent-2022.1.0.209.0-lnxrh7x64.tar.gz s3://$S3BucketName
aws s3 cp IRIS-2022.1.0.209.0-lnxubuntu2004x64.tar.gz s3://$S3BucketName
```

このテンプレートは、AmazonLinuxを使用しています。RedHat7もしくはUbuntuを利用する場合は、別途「[カスタマイズ候補](#)」のチャプターの作業が必要です

また、事前にAWS上に用意されているテンプレートではなく、カスタマイズを行ったテンプレートを使用するので、下記を実行します。

QSS3BucketNameはS3BucketNameと同じでも、異なっても構いません。

```
$ git clone https://github.com/IRISmeister/AWSIRISDeployment.git --recursive
$ cd AWSIRISDeployment
```

Windowsにデプロイする場合

```
QSS3BucketName=<my bucket>
aws s3 cp quickstart-intersystems-iris-windows s3://$QSS3BucketName/quickstart-
intersystems-iris-windows --recursive
aws s3 cp submodules/quickstart-aws-vpc/templates s3://$QSS3BucketName/quickstart-
intersystems-iris-windows/submodules/quickstart-aws-vpc --recursive
aws s3 cp submodules/quickstart-linux-bastion/templates s3://$QSS3BucketName/quicksta
rt-intersystems-iris-windows/submodules/quickstart-linux-bastion --recursive
aws s3 cp submodules/quickstart-linux-bastion/scripts s3://$QSS3BucketName/quickstart-
intersystems-iris-windows/submodules/quickstart-linux-bastion --recursive
```

Linuxにデプロイする場合

```
QSS3BucketName=<my bucket>
aws s3 cp quickstart-intersystems-iris s3://$QSS3BucketName/quickstart-intersystems-
iris --recursive
aws s3 cp submodules/quickstart-aws-vpc/templates s3://$QSS3BucketName/quickstart-
intersystems-iris/submodules/quickstart-aws-vpc --recursive
aws s3 cp submodules/quickstart-linux-bastion/templates s3://$QSS3BucketName/quicksta
rt-intersystems-iris/submodules/quickstart-linux-bastion --recursive
aws s3 cp submodules/quickstart-linux-bastion/scripts s3://$QSS3BucketName/quickstart-
intersystems-iris/submodules/quickstart-linux-bastion --recursive
```

実行例

以下、既存VPCにWindows版IRISのミラー構成をデプロイする際の実行例です。

VPCを新規作成する場合は、下記の[VPC関連の事前準備](#)は不要です。

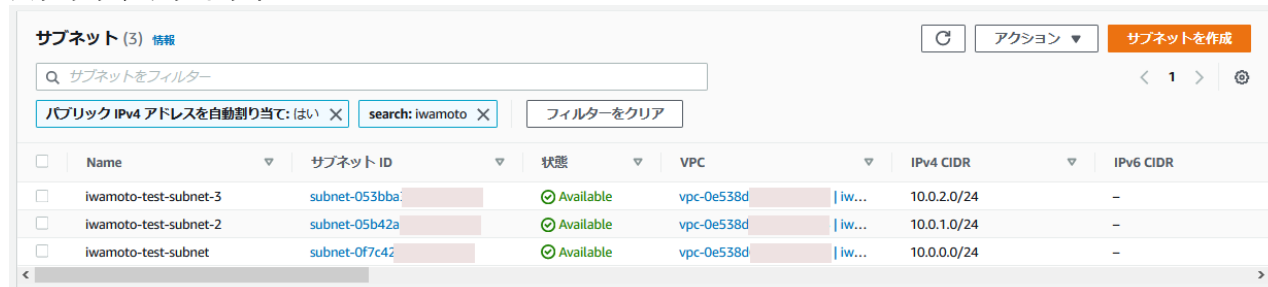
事前準備

VPC関連

既存VPC上に、以下のリソースを事前に用意しました。

デプロイ実行時にNATゲートウェイが無いと、スタックの作成に失敗します。必ず用意してください。

- パブリックサブネット



☐	Name	サブネット ID	状態	VPC	IPv4 CIDR	IPv6 CIDR
☐	iwamoto-test-subnet-3	subnet-053bba	Available	vpc-0e538d	10.0.2.0/24	-
☐	iwamoto-test-subnet-2	subnet-05b42a	Available	vpc-0e538d	10.0.1.0/24	-
☐	iwamoto-test-subnet	subnet-0f7c42	Available	vpc-0e538d	10.0.0.0/24	-

- プライベートサブネット
- ルートテーブル/ルート
- ルートテーブル/サブネットの関連付け
- NATゲートウェイ

S3への各種ファイルのアップロード

```
S3BucketName=iwamoto-cf-templates
aws s3 mb s3://$S3BucketName
aws s3 cp ISCAgent-2022.1.0.209.0-lnxrh7x64.tar.gz s3://$S3BucketName
aws s3 cp iris.key s3://$S3BucketName
aws s3 cp IRIS-2022.1.0.209.0-win_x64.exe s3://$S3BucketName

git clone https://github.com/IRISmeister/AWSIRISDeployment.git
cd AWSIRISDeployment
# ??????????????bucket?????????
QSS3BucketName=iwamoto-cf-templates
aws s3 cp quickstart-intersystems-iris-windows s3://$QSS3BucketName/quickstart-
intersystems-iris-windows --recursive
aws s3 cp submodules/quickstart-aws-vpc/templates s3://$QSS3BucketName/quickstart-
intersystems-iris-windows/submodules/quickstart-aws-vpc --recursive
aws s3 cp submodules/quickstart-linux-bastion/templates s3://$QSS3BucketName/quicksta
rt-intersystems-iris-windows/submodules/quickstart-linux-bastion --recursive
aws s3 cp submodules/quickstart-linux-bastion/scripts s3://$QSS3BucketName/quickstart-
intersystems-iris-windows/submodules/quickstart-linux-bastion --recursive
```

このような内容になります。

```
aws s3 ls s3://$QSS3BucketName --recursive
2022-06-15 07:48:57 675737136 IRIS-2022.1.0.209.0-win_x64.exe
2022-10-12 01:09:57 16139988 ISCAgent-2022.1.0.209.0-lnxrh7x64.tar.gz
2022-03-29 07:11:24 546 iris.key
2022-11-25 09:15:50 0 quickstart-intersystems-iris-windows/
2022-11-25 09:16:05 0 quickstart-intersystems-iris-windows/scripts/
2022-11-28 05:34:49 6008 quickstart-intersystems-iris-
windows/scripts/MirrorInstaller.xml
2022-11-28 13:34:52 0 quickstart-intersystems-iris-windows/submodules/
2022-11-28 13:34:52 0 quickstart-intersystems-iris-
windows/submodules/quickstart-aws-vpc/
2022-11-28 13:34:52 0 quickstart-intersystems-iris-
windows/submodules/quickstart-aws-vpc/templates/
2022-11-28 13:34:52 59966 quickstart-intersystems-iris-
windows/submodules/quickstart-aws-vpc/templates/aws-vpc.template.yaml
2022-11-28 13:34:52 0 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/
2022-11-28 13:34:52 0 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/scripts/
2022-11-28 13:34:52 531 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/scripts/auditing_configure.sh
2022-11-28 13:34:52 881 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/scripts/banner_message.txt
2022-11-28 13:34:52 11763 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/scripts/bastion_bootstrap.sh
2022-11-28 13:34:52 0 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/templates/
2022-11-28 13:34:52 24068 quickstart-intersystems-iris-
windows/submodules/quickstart-linux-bastion/templates/linux-bastion.template
2022-11-25 09:16:16 0 quickstart-intersystems-iris-windows/templates/
2022-11-28 09:35:42 6059 quickstart-intersystems-iris-windows/templates/iris-
cluster-arbiter-node.template.yaml
2022-11-28 09:35:42 21529 quickstart-intersystems-iris-windows/templates/iris-
cluster-iris-node.template.yaml
2022-11-28 09:35:42 13736 quickstart-intersystems-iris-windows/templates/iris-
cluster-main.template.yaml
2022-11-28 09:35:43 10403 quickstart-intersystems-iris-windows/templates/iris-
entrypoint-new-vpc.template.yaml
```

(オプション)全パラメータ設定済みyamlの編集

繰り返し実行することを考慮して、パラメータを全て指定した状態のテンプレートを作成しておくと便利です。

```
vi test-iris-cluster-main-windows.yaml
```

編集後のファイルをうっかり公開してしまわないよう注意してください

編集時の注意

- BastionSubnetIdParameterには、異なるAZに属する2つのpublic subnetを指定してください。
- InstanceSubnetIdParameterには、異なるAZに属する3つのsubnetを指定してください。はじめの2つがIRISホスト、最後に指定したサブネットがArbiterホストが稼働するサブネットになります。
- IRISをインストールするホストは、インストール作業の際にインターネットへのアクセスを行います。具体的にはs3アクセスのためにaws cliが必要(amazon linuxと違ってUbuntuにはプリインストールされていま

せん)なのですが、その他にもchocolateyを使用してaws cliをインストールしています。NATゲートウェイ等を構成済みのプライベートサブネットを構築済みで無い場合は、[こちら](#)を参考にして一時的に追加してください。

NATゲートウェイは存在するだけでコストが発生します。(VM起動直後のS/Wインストール時など)必要な時に作成し、不要になったらすぐ削除したかったので、別のCFテンプレートにしています。

コンソールからCloudFormationの実行

- 1. スタックを「新しいリソースを使用」して作成します。

テンプレートソースには「テンプレートファイルのアップロード」を選択し、先ほど編集したファイル(test-iris-cluster-main-windows.yaml)を指定します。

「スタックの詳細を指定」画面でパラメータを設定します。

パラメータ	設定値例
BastionSubnetIdParameter	subnet-0f7c4xxxxxxxxxxxxx,subnet-05b42xxxxxxxxxxxxx
IRISPasswordParameter	SYS1
InstanceSubnetIdParameter	subnet-0180bxxxxxxxxxxxxx,subnet-03272xxxxxxxxxxxxx,subnet-08e8fxxxxxxxxxxxxx
QSS3BucketName	iwamoto-cf-templates
QSS3BucketRegion	ap-northeast-1
QSS3KeyPrefix	quickstart-intersystems-iris-windows/
S3BucketNameParameter	iwamoto-cf-templates
SshKeyParameter	aws
VpcIdParameter	vpc-0e538xxxxxxxxxxxxx

「スタックオプションの設定」画面に特に設定はありません。実行がうまくいかない場合は、スタックの作成オプションの「失敗時のロールバック」を無効にしておく、作成された環境がロールバックされずに残りますので、問題の解析がしやすくなります(不要になったら、忘れずに削除すること)。

「レビュー」画面で、「The following resource(s) require capabilities:」で、下記にチェックをいれる必要があります。

- AWS CloudFormation によって IAM リソースがカスタム名で作成される場合があることを承認します。
- AWS CloudFormation によって、次の機能が要求される場合があることを承認します:

CAPABILITYAUTOEXPAND

「スタックの作成」ボタンを押します。複数のネストされたスタックの作成が開始されます。

- 2. 出力内容を確認します。
スタックのステータスがCREATECOMPLETEになるまで待機します(15分ほどかかりました)。
スタック(アップロードしたスタックではなく、ネストされたxxx-IRISStack-xxxx)の出力を表示します。ギアアイコンで行の折り返し指定が出来ます。

Stack	キー	値	説明
IRISStack	JDBCEndpoint	jdbc:IRIS://xxxx-NLB-xxxxx.elb.ap-northeast-1.amazonaws.com:51773/DATA	JDBC Connection String
IRISStack	Node01InstanceId	i-xxxxxxxxxx	Node 01 Instance ID
IRISStack	Node01PrivateIP	10.0.12.85	Node 01 Private IP
IRISStack	Node02InstanceId	i-xxxxxxxxxx	Node 02 Instance ID
IRISStack	Node02PrivateIP	10.0.10.159	Node 02 Private IP

これらの出力値を記録しておいてください。

IRIS管理ポータルへのアクセス

Node01ホスト(ミラープライマリサーバ)の管理ポータルに接続します。下記のコマンドを実行します。

- 踏み台ホストを使用しない場合

セッションマネージャのポート転送を利用します。事前に[AWS CLIのインストール・構成](#)と[SSMプラグインのインストール](#)が必要です。私はWindows10+WSL2(Ubuntu)を使用していますので、下記を実行しました。

```
$ curl "https://s3.amazonaws.com/session-manager-downloads/plugin/latest/ubuntu_64bit/session-manager-plugin.deb" -o "session-manager-plugin.deb"
$ sudo dpkg -i session-manager-plugin.deb
```

```
$ export instanceid=i-xxxxxxxxxx
$ aws ssm start-session --target $instanceid \
                        --document-name AWS-StartPortForwardingSession \
                        --parameters '{"portNumber":["52773"],"localPortNumber":["52773"]}'
```

[SSHトンネルを作成する方法](#)もあります。

- 踏み台ホストを使用する場合

```
BastionPublicIP=54.168.xxx.xxx
Node01PrivateIP=10.0.12.85
ssh -i aws.pem -L 52773:$Node01PrivateIP:52773 ec2-user@$BastionPublicIP
```

この時点で、<http://localhost:52773/csp/sys/UtilHome.csp> で管理ポータルにアクセスできます。アカウントはSuperUser、パスワードはスタックのパラメータで指定したものを使用します。ミラーの状態を確認するために、管理ポータルのホーム画面の右端に「ミラー・モニターを表示」というリンクがありますので、クリックします。以下の画面のように表示されていれば正常です。

WindowsへのRDP接続

RDP接続する場合は、(推奨に従って)Node01をプライベートサブネットに作成している場合、ssh同様、RDPも直接接続できません。フリーマネージャーをRDPの代替えとして利用可能です。

フリーマネージャーを使わない場合は、下記のようなコマンドをローカルで実行し、localhostから転送する必要があります。

- 踏み台ホストを使用しない場合

```
$ export instanceid=i-xxxxxxxxxx
$ aws ssm start-session --target $instanceid \
                        --document-name AWS-StartPortForwardingSession \
                        --parameters '{"portNumber":["3389"],"localPortNumber":["3389"]}'
```

- 踏み台ホストを使用する場合

```
ssh -i aws.pem -L 3389:$Node01PrivateIP:3389 ec2-user@$BastionPublicIP
```

Windowsのパスワードは、AWSコンソールからのRDP接続方法で取得します。
その上で、RDPでlocalhost:33389に接続し、Administrator/取得したパスワード、でログインします。

Linuxへの接続

- 踏み台ホストを使用しない場合

SSMのセッションマネージャを使用します。

あるいは、`ssh/config`を [設定する](#)ことで、SSM経由でSSHアクセスすることも可能です。

```
$ export instanceid=i-xxxxxxxxxx
$ ssh -i aws.pem ec2-user@$instanceid
```

- 踏み台ホストを使用する場合

```
BastionPublicIP=54.168.xxx.xx
Node01PrivateIP=10.0.12.85
ssh -i aws.pem -L 52773:$Node01PrivateIP:52773 ec2-user@$BastionPublicIP
ssh -i aws.pem -L 33389:$Node01PrivateIP:3389 ec2-user@$BastionPublicIP
ssh -i aws.pem -oProxyCommand="ssh -i aws.pem -W %h:%p ec2-user@$BastionPublicIP" ec2-user@$Node01PrivateIP
or
ssh -i aws.pem -oProxyCommand="ssh -i aws.pem -W %h:%p ec2-user@$BastionPublicIP" ubuntu@$Node01PrivateIP
```

BastionPublicIPは、事前に用意した踏み台ホストの公開IPです。

動作確認

- 外部ロードバランサ経由のアクセス
この時点で、JDBC経由でアクセス可能になっているはずです。JDBCEndpointを使用してアクセスします。
> 手順は省略します。実行にはJDBCドライバが必要です。

カスタマイズ候補

Windows

iris-cluster-iris-node.template.yamlでPSファイル等を作成しています。下記の箇所は、環境・目的に応じて変更してください。

- Windows環境のローカライズ(timezone, firewall設定)
 - `c:\cfn\scripts\Setup-config.ps1`
注意 firewall **を無効に設定しています**
- ドライブの作成、割り当て
 - Resourcesセクション

Resources:

```
NodeInstance:
  Properties:
    BlockDeviceMappings:
```

- `c:\cfn\scripts\drives.diskpart.txt`

- IRISインストール先など

- `c:\cfn\scripts\Install-IRIS.ps1`

```
$irisdir="h:\InterSystems\IRIS"
$irissvcname="IRIS_h-_intersystems_iris"
$irisdbdir="I:\iris\db\"
$irisjrndir="J:\iris\jrn1\pri"
$irisjrnlaltdir="K:\iris\jrn1\alt"
```

このpsファイルは、実行時に作成される/temp/envs.ps1と組み合わせれば、IRISの無人インストール用のスクリプトとして機能します。

- プリインストールするソフトウェア

- `c:\cfn\scripts\Install-choco-packages.ps1`
s3からファイル入手する場合、awscliは必須です。利便性のためnotepadplusplus, google chromeを追加インストールしています。

Linux

- O/Sの指定

現在、O/Sは利便性によりAmazon Linuxを指定してあります。

O/SをRedHatに変更するには下記ファイルの内容を変更してください。

`quickstart-intersystems-iris /templates /iris-cluster-iris-node.template.yaml`

```
LatestAmiIdParameter:
  Type: AWS::EC2::Image::Id
  Default: ami-0be4c0b05bbeb2afd
```

O/SをUbuntuに変更するには下記ファイルの内容を変更してください。

`quickstart-intersystems-iris /templates /iris-cluster-iris-node.template.yaml`

```
LatestAmiIdParameter:
  Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
  Default: /aws/service/canonical/ubuntu/server/focal/stable/current/amd64/hvm/ebs-gp2/ami-id
```

```
IRISKitNameParameter:
  Type: String
```

Default: IRIS-2022.1.0.209.0-linuxubuntu2004x64

オリジナル(quickstart)との差異

デプロイ先をWindowsに変更する際に、Linux版との互換性を維持したままで、いくつかの修正を加えました。

- Windowsの起動に時間がかかるため、デフォルトのInstance typeをm5.largeからm5.xlargeに変更。gp2をgp3に変更。IOPSを指定。
- LatestAmildForIRISParameterパラメータを新設。Windowsのキット(日本語版、英語版等)の指定。
- IRISKitNameParameterパラメータを新設。キット名の指定。
- RDPアクセス用にSecurityGroupIngress(ポート:3389)を追加
- SE.ShardInstallerクラス修正
 - CreateMirrorSet(), JoinAsFailover()を変更
##class(SYS.Mirror).CreateMirrorSet(),JoinAsFailover()実行時のECPAddressのデフォルト(\$system .INetInfo.LocalHostName())が、Windowsでは"EC2AMAZ-F1UF3QM"のようなWindowsホスト名になる。このホスト名では他のホストからDNSで名前解決できないので、JoinMirrorAsFailoverMember()がエラーになる。そのため、下記を追加。

```
CreateMirrorSet()
```

```
set mirror("ECPAddress") = hostName // Windows on AWS need this
```

```
JoinAsFailover()
```

```
set MirrorInfo("ECPAddress") = hostName // Windows on AWS need this
```

- Roleを追加

```
????S3????????s3:GetObject  
ec2:Describe*  
ec2:DeleteRoute  
ec2:CreateRoute  
ec2:ReplaceRoute
```

```
????????????????  
AmazonSSMManagedInstanceCore
```

- Linux用のcloud-initのUserData(初回起動時に実行されるshell)にO/SがUbuntu,RedHat7の場合分けを追加
- 踏み台ホストをデプロイしない代わりにAWS System Managerを有効化

その他

1. LBのヘルスチェック値

LBのヘルスチェックのデフォルト値を使用しています。quickstart-intersystems-iris-windows/templates/iris-cluster-main.template.yamlのコメントを解除して適切な値に調整してください。

```
#HealthCheckTimeoutSeconds: 10  
#HealthCheckIntervalSeconds: 10
```

#UnhealthyThresholdCount: 3

2. テンプレートとAWSリソースの関係

以下、テンプレートと作成されるAWSリソースの関係です。

テンプレート名	作成されるリソース
iris-cluster-iris-node.template.yaml	スタンダローン
iris-cluster-iris-main.template.yaml	ミラー環境
iris-entrypoint-new-vpc.template.yaml	VPC+ミラー環境
AWS System Managerを有効化し	

ているため、いずれも踏み台ホストは作成しません。必要であれば、この[テンプレート](#)を参考にして作成してください。

3. テンプレートのデフォルト値

環境の明滅を繰り返し実行する可能性がある場合、パラメータを全て指定した状態のテンプレートを別途作成しておくとう便利です。

テンプレート名	O/S	用途
test-iris-cluster-iris-node-linux.yaml	Linux	スタンダローン
test-iris-cluster-iris-node-windows.yaml	Windows	スタンダローン
test-iris-cluster-iris-main-linux.yaml	Linux	ミラー環境
test-iris-cluster-iris-main-windows.yaml	Windows	ミラー環境
test-iris-entrypoint-new-vpc-linux.yaml	Linux	VPC+ミラー環境
test-iris-entrypoint-new-vpc-windows.yaml	Windows	VPC+ミラー環境

例えば、スタンダローン構成でIRISを起動したい場合は、下記のファイルを編集します。

```
$ #????????????(InstanceSubnetIdParameter??)????????????????????????
$ vi test-iris-cluster-iris-node-linux.yaml
```

スタック作成時に、test-iris-cluster-iris-node-windows.yamlを指定すれば、スタンダローン構成でIRISを起動することが出来ます。

3. WindowsへのSSH

SSMのセッションマネージャを使用します。

踏み台ホストの使用を止めたため、下記内容は無効になりました。
下記の方法を試す場合は、別途、[こちらのよう](#)なテンプレートを利用して踏み台ホストを作成してください。

IRIS稼働ホストにOpenSSHをインストールすれば、踏み台ホスト経由で、IRISホストにSSHする事が可能です。ただし、Linux版に比べて、Windows版のIRISではCLIで実行できることが限られているため、効果は限定的です。IRIS稼働ホストで実行。

PS C:\Users\Administrator> Add-

```
PS C:\Users\Administrator> Start-Service sshd
```

Windowクライアントからアクセスする際には、Windows版OpenSSHクライアント固有の問題「`posixspawn: No such file or directory`」を回避するために、Git同梱のBashを使用しました。

```
Administrator@10.0.0.62's password: RDP??????????????
```

(load pubkey "aws.pem": invalid format が出ますが無視)

また、ひと手間いりますが、踏み台ホストの`ssh/authorized_keys`(パブリックキー)を、デプロイ先のWindowsにコピーすれば、ワンライナーで公開鍵認証できます。

```
user@DESKTOP-XXXX MINGW64 ~ ssh -i aws.pem -oProxyCommand="ssh -i aws.pem -W %h:%p ec2-user@54.95.xxx.xxx" Administrator@10.0.0.62
```

Adminグループには特別の設定が必要でした。[こちら](#)のConfiguring the Serverを参考にしました。

4. cfn-init.logにPythonのエラーが出る場合

cfn-init.logに下記のようなエラーが出ることがあるようです。

```
2021-02-12 02:50:32,957 [ERROR] Unhandled exception during build: 'utf8' codec can't
decode byte 0x83 in position 8: invalid start byte
```

回避されることを期待して、Install-IRIS.ps1に、以下の命令を追加しています。

```
[Console]::OutputEncoding = [System.Text.Encoding]::UTF8
```

参考にしたサイト

下記のサイトを参考にしています。

- <https://aws.amazon.com/jp/quickstart/architecture/intersystems-iris>
- <https://dev.classmethod.jp/articles/about-windows-cfn-init-non-ascii-enc...>
- <https://dev.classmethod.jp/articles/aws-cloudformation-setup-windows-ser...>
- <https://aws.amazon.com/jp/premiumsupport/knowledge-center/cloudformation...>
- <https://www.concurrency.com/blog/may-2019/key-based-authentication-for-o...>

#AWS #Mirroring #クラウド #デプロイ #InterSystems IRIS #InterSystems IRIS for Health

[illegible]

[83%97%E3%83%AC%E3%83%BC%E3%83%88-windows%E7%B7%A8](#)