
記事

[Hiroshi Sato](#) · 2020年6月30日 17m read

icmを利用してIRISクラスターを構成する方法

1. 初めに

IRISでは、複数ノードでクラスターを構成し、ワークロードのスケールアウト、データボリュームのスケールアウトやトランザクション処理と分析処理を異なるノードで処理するマルチワークロードを実現しています。

しかし、クラスターを構成するための設定は、ノード数が増えるにつれ煩雑になり、それらを人手の作業に全て委ねると設定ミス等を招きやすいといえます。

また、クラスタの構成を処理負荷の増加に基づいて拡張する、または逆に縮小する、あるいは、データ冗長性を追加するためにミラーリングの構成を追加するなど構成変更は、想定するより多いかもしれません。

しかもクラスタ毎に同様の設定を毎回行うとなると、人手による作業では、煩雑性だけでなく俊敏性に欠けると言わざるを得ません。

そこで、IRISには、クラスター構成作業を自動化する新しいツールICM (InterSystems Cloud Manager) が用意されました。

ここでは、ICMを使用したクラウド上でのIRIS構成の自動化の手順について説明します。

2. 事前に準備するもの

以下の作業を行うためには、InterSystemsが用意している 2 つのDocker Imageを事前に取得する必要があります。

- ICMイメージ
- IRISイメージ

これらのDockerイメージを取得するには、InterSystemsが運営しているWRC Directというシステムにログインする必要があります。

WRC Directシステムにログインするアカウントは、InterSystems製品のパートナー様やInterSystemsの製品を直接ご購入いただいたエンドユーザー様は取得可能です。

そのアカウントをお持ちでない方は、お手数ですがインターフェースのカスタマーサポートセンターまでお問い合わせください。

- WRC Directにログイン後、InterSystemsタブからSoftware Distributionを選択します。
- 表示される複数のメニューからContainersを選択します。
- InterSystems IRISとInterSystems Cloud Managerを選び、ダウンロードボタンを押します。
- IRISとICMのバージョンは合わせてください。
- 特に条件等なければ、その時点の最新バージョンを選んでください。
- アーキテクチャーの所でx64とARMがあるので、間違えないようにしましょう。

3. Dockerイメージの登録

クラウドからicmで指定するDockerイメージを取得できる様にインターネットからアクセスできるレポジトリが必要です。

一番簡単な方法は、Docker Hubに登録することです。

3.1 Docker Hubへの登録

以下はDocker Hubに登録するサンプルです。

- イメージの名前は、違う名前でタグ付してください。（hsato0719/icm:2020.1.0.215.0）
- docker hubにログインするためには、Dockerのアカウント（ID）が必要です。
- アカウントは無償で作成可能です。
- WRC Directからダウンロードしたアーカイブファイルを使用して以下のコマンド発行

```
$ docker load -i icm-2020.1.0.215.0-docker.tar.gz  
$ docker load -i iris-2020.1.0.215.0-docker.tar.gz
```

```
$ docker tag intersystems/icm:2020.1.0.215.0 hsato0719/icm:2020.1.0.215.0  
$ docker tag intersystems/iris:2020.1.0.215.0 hsato0719/iris:2020.1.0.215.0
```

```
$ docker login
```

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

Username: satohiroshi0719

Password:

Login Succeeded

```
$ docker push hsato0719/icm:2020.1.0.215.0  
$ docker push hsato0719/iris:2020.1.0.215.0
```

4. ライセンスキー、クレデンシャル情報のセットアップ

4.1 icm dockerの起動

以下を実行するまえにシステムにDockerがインストールされているか確認してください。(Docker for Windows, Docker for Mac)

```
$ docker run --name icm -v /Users/hsato:/myhome -it --cap-add SYSTIME hsato0719/icm:2020.1.0.215.0
```

-v で設定しているディレクトリーは、ホストシステム（WindowsやMac）とDockerの間でファイルを交換したい場合に設定します。

4.2 クレデンシャル情報の生成

以下のコマンドを実行します。

```
$ ICM/bin/keygenSSH.sh  
$ ICM/bin/keygenTLS.sh
```

4.3 ライセンスキーのコピー

- ライセンスキーはUNIXフォーマット（Windowsフォーマットだとicm runの時にコピーが永遠に終わらない状態になる）で作成してください。
- 評価用ライセンスは、サポート契約のあるエンドユーザー、パートナー様であれば、製品評価ポータル（<https://evaluation.InterSystems.com>）から取得できます。（WRC Directアカウントでログイン）

以下のコマンドを実行します。

```
$ cd /Samples  
$ mkdir license  
$ cd license  
$ cp /myhome/license/ubuntu-sharding-iris.key .
```

（/myhome/licenseにubuntu-sharding-iris.keyを事前に置きます。）

5. defaults.jsonとdefinitions.jsonの編集

ICMの設定の中心となるdefaults.jsonとdefinitions.jsonの編集を行います。

defaults.jsonの内容は、サポートしているすべてのクラウドで共通な部分とクラウド固有な部分があります。

5.1 defaults.json

以下のサンプルは、AWS用の設定です。

```
/Samples # cat AWS/defaults.json
{
  "Provider": "AWS",
  "Label": "Sato",
  "Tag": "TEST",
  "DataVolumeSize": "10",
  "SSHUser": "ubuntu",
  "SSHPublicKey": "/Samples/ssh/insecure-ssh2.pub",
  "SSHPrivateKey": "/Samples/ssh/insecure",
  "DockerImage": "hsato0719/iris:2020.1.0.215.0",
  "DockerUsername": "",
  "DockerPassword": "",
  "TLSKeyDir": "/Samples/tls/",
  "LicenseDir": "/Samples/license/",
  "Region": "ap-northeast-1",
  "Zone": "ap-northeast-1a",
  "DockerVersion": "5:19.03.53-0ubuntu-bionic",
  "AMI": "ami-0278fe6949f6b1a06",
  "InstanceType": "m4.large",
  "Credentials": "/Samples/AWS/sample.credentials",
  "Mirror": "false",
  "UserCPF": "/Samples/cpf/iris.cpf"
}
```

- Label, Tagは適当にご自分専用の名前を使用してください。
- DataVolumeSizeも用途に応じてサイズを調整してください。
- SSHUserはAWSのubuntuの場合、ubuntuで固定です。
- SSHPublicKeyとSSHPrivateKeyは4.2で生成したキーファイルをそのまま使用します。
- DockerImageは3.1でDocker Hubに登録したタグ名を指定します。
- InstanceTypeは用途に応じてタイプを選択してください。

それ以外は、そのままの値でOKです。

以下のサンプルは、Azure用の設定です。

```
/Samples # cat Azure/defaults.json
{
  "Provider": "Azure",
  "Label": "Hiroshi",
  "Tag": "TEST",
  "DataVolumeSize": "50",
  "SSHUser": "ubuntu",
  "SSHPublicKey": "/Samples/ssh/insecure-ssh2.pub",
  "SSHPrivateKey": "/Samples/ssh/insecure",
  "DockerImage": "hsato0719/iris:2020.1.0.215.0",
  "DockerUsername": "",
  "DockerPassword": "",
  "TLSKeyDir": "/Samples/tls/",
  "LicenseDir": "/Samples/license/",
  "Location": "Japaneast",
  "Zone": "1",
  "PublisherName": "Shardtest",
  "Offer": "UbuntuServer",
  "Sku": "18.04-LTS",
  "Version": "18.04.201909030",
  "DockerVersion": "18.06.1ee3-0ubuntu",
  "Size": "Standard_DS3_v2",
  "AccountTier": "Premium",
  "AccountReplicationType": "LRS",
  "SubscriptionId": "xxxxxxxxxx",
  "ClientId": "xxxxxxxxxxxx",
  "ClientSecret": "xxxxxxxxxxxx",
  "TenantId": "xxxxxxxxxxxx",
  "Mirror": "false",
  "UserCPF": "/Samples/cpf/iris.cpf"
}
```

変更する部分は、AWS用とほぼ同様です。
LabelとTagは適当に変更してください。

以下の設定は、6.2のクラウド個別設定の所を参照してください。

- SubscriptionId
- ClientId
- ClientSecret
- TenantId

Definitions.jsonは基本的にクラウド固有の情報はありません。

以下の設定では、2ノードのシャードデータクラスターを構成しています。

5.2 definitions.json

```
/Samples # cat Azure/definitions.json
[
  {
    "Role": "DATA",
    "Count": "2",
    "LicenseKey": "ubuntu-sharding-iris.key"
  }
]
```

6. クラウド個別設定

6.1 sample.credentials (for AWS)

- AWSコンソールから右上のマイアカウント（自分の登録したメールアドレス）の所をクリック
- 表示されるメニューからマイセキュリティ資格情報をクリック
- アクセスキー（アクセスキー ID とシークレットアクセスキー）の左側の三角をクリック
- 新しいアクセスキーの生成ボタンを押す
- ダウンロードcsvファイルを選んでクリック
- 生成されたファイルを以下の形式に変換(XXXXの部分はそのまま)し、sample.credentialsという名前にする

```
[default]
awsaccesskeyid = XXXX
awssecretaccesskey = XXXX
```

- そのsamle.credentialsをicm docker内の/Samples/AWSにコピーする

```
$ cd /Samples/AWS
$ cp /myhome/sample.credentials .
```

- そのdefaults.jsonとdefinitions.jsonをicm docker内の/Samples/AWSにコピーする

```
$ cd /Samples/AWS
$ cp /myhome/defaults.json .
$ cp /myhome/definitions.json .
```

6.2 Azure

- subscriptionid Azureホームから移動の所のサブスクリプションをクリック
- サブスクリプションID
- tenant-id AzureホームからAzureサービスのAzure Active Directoryをクリック
- 既定のディレクトリーの所のテナントID
- 左のメニューからアプリの登録を選び、 + 新規登録を押す
- アプリケーションの名前 入力
- サポートされているアカウントの種類
- この組織ディレクトリのみに含まれるアカウント
- リダイレクトURLWeb <https://intersystems.com>
- 登録ボタンを押す
- 表示されるアプリケーションIDがクライアントID左のメニューから証明書とシークレットを選択
- 新しいクライアントシークレットを押す
- 左のメニューからAPIのアクセス許可をクリック
- + 右のペインからアクセス許可の追加ボタンを押す右側からAzure Service Managementをクリック
- Access Azure Service Management as organization userを選択
- アクセス許可の追加ボタンを押す
- Azure トップからサブスクリプションを選択
- 左側のメニューからアクセス制御IAMをクリック
- ロールの割り当てを追加するの追加ボタンを押す
- 役割 所有者
- アクセスの割り当て先 Azure ADのユーザー、 グループ、 サービスプリンシパル
- 選択 先ほど作ったアプリケーションの名前
- 保存ボタンを押す
- そのdefaults.jsonとdefinitions.jsonをicm docker内の/Samples/AZureにコピーする

```
$ cd /Samples/AZure  
$ cp /myhome/defaults.json .  
$ cp /myhome/definitions.json .
```

7. プロビジョニング

6.で設定した構成情報に基づいて、クラウド上で必要なリソースの割り当てや基本的な設定等を実施します。

icm provisionコマンドでそれを実行します。

以下、Azureでの例を示します。

```
$ cd /Samples/AWS  
/Samples/Azure # icm provision  
Starting init of Sato-ICM...  
...completed init of Sato-ICM  
Starting plan of Sato-ICM...  
...completed plan of Sato-ICM  
Starting refresh of Sato-ICM...  
...completed refresh of Sato-ICM  
Starting apply of Sato-ICM...  
Elapsed time for apply on Sato-ICM: 15s...  
Elapsed time for apply on Sato-ICM: 30s...  
...completed apply of Sato-ICM
```

Starting init of Sato-DATA-ICM...
...completed init of Sato-DATA-ICM
Starting plan of Sato-DATA-ICM...
...completed plan of Sato-DATA-ICM
Starting refresh of Sato-DATA-ICM...
...completed refresh of Sato-DATA-ICM
Starting apply of Sato-DATA-ICM...
Elapsed time for apply on Sato-DATA-ICM: 15s...
Elapsed time for apply on Sato-DATA-ICM: 30s...
Elapsed time for apply on Sato-DATA-ICM: 45s...
Elapsed time for apply on Sato-DATA-ICM: 60s...
Elapsed time for apply on Sato-DATA-ICM: 75s...
Elapsed time for apply on Sato-DATA-ICM: 90s...
Elapsed time for apply on Sato-DATA-ICM: 105s...
Elapsed time for apply on Sato-DATA-ICM: 120s...
...completed apply of Sato-DATA-ICM
Copying files to Sato-DATA-ICM-0002...
Copying files to Sato-DATA-ICM-0001...
...copied files to Sato-DATA-ICM-0001
Configuring Sato-DATA-ICM-0001...
...copied files to Sato-DATA-ICM-0002
Configuring Sato-DATA-ICM-0002...
...configured Sato-DATA-ICM-0001
Copying more files to Sato-DATA-ICM-0001...
...configured Sato-DATA-ICM-0002
Copying more files to Sato-DATA-ICM-0002...
...copied more files to Sato-DATA-ICM-0001
Mounting volumes on Sato-DATA-ICM-0001...
...copied more files to Sato-DATA-ICM-0002
Mounting volumes on Sato-DATA-ICM-0002...
...mounted volumes on Sato-DATA-ICM-0001
Installing Docker on Sato-DATA-ICM-0001...
...mounted volumes on Sato-DATA-ICM-0002
Installing Docker on Sato-DATA-ICM-0002...
Elapsed time for provision on Sato-DATA-ICM-0001: 15s...
Elapsed time for provision on Sato-DATA-ICM-0002: 15s...
Elapsed time for provision on Sato-DATA-ICM-0001: 30s...
Elapsed time for provision on Sato-DATA-ICM-0002: 30s...
Elapsed time for provision on Sato-DATA-ICM-0001: 45s...
Elapsed time for provision on Sato-DATA-ICM-0002: 45s...
Elapsed time for provision on Sato-DATA-ICM-0001: 60s...
Elapsed time for provision on Sato-DATA-ICM-0002: 60s...
...installed Docker on Sato-DATA-ICM-0001
Determining host OS of Sato-DATA-ICM-0001...
...installed Docker on Sato-DATA-ICM-0002
Determining host OS of Sato-DATA-ICM-0002...
...determined host OS of Sato-DATA-ICM-0001
Determining local IP address of Sato-DATA-ICM-0001...
...determined host OS of Sato-DATA-ICM-0002
Determining local IP address of Sato-DATA-ICM-0002...
...determined local IP address of Sato-DATA-ICM-0001
...determined local IP address of Sato-DATA-ICM-0002
Installing Weave Net on Sato-DATA-ICM-0001...
Installing Weave Net on Sato-DATA-ICM-0002...
Elapsed time for weave on Sato-DATA-ICM-0001: 15s...
Elapsed time for weave on Sato-DATA-ICM-0002: 15s...
...installed Weave Net on Sato-DATA-ICM-0002
Collecting Weave info for Sato-DATA-ICM-0002...
...installed Weave Net on Sato-DATA-ICM-0001

```
Collecting Weave info for Sato-DATA-ICM-0001...
...collected Weave info for Sato-DATA-ICM-0002
...collected Weave info for Sato-DATA-ICM-0001
Machine IP Address DNS Name Provider Region Zone
-----
Sato-DATA-ICM-0001 52.185.190.176 satodataicm1.japaneast.cloudapp.azure.co Azure Japaneast 1
Sato-DATA-ICM-0002 52.185.190.169 satodataicm2.japaneast.cloudapp.azure.co Azure Japaneast 1
To destroy: icm unprovision -stateDir state [-cleanUp] [-force]
```

8. Deploy

7.でプロビジョニングされた構成で、実際のシステムをデプロイしてみます。

```
/Samples/Azure # icm run
Enter desired InterSystems IRIS password: ---
構成するIRISのシステムパスワードを聞かれるので、ここで入力し、設定します。
Enter password again: --- 同じパスワードを入力します。
Pulling image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0001 ...
Pulling image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0002 ...
Elapsed time for pull on Sato-DATA-ICM-0001: 15s...
Elapsed time for pull on Sato-DATA-ICM-0002: 15s...
...pulled image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0002
...pulled image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0001
Stopping container iris on Sato-DATA-ICM-0001...
Stopping container iris on Sato-DATA-ICM-0002...
...completed stop of container iris on Sato-DATA-ICM-0001
...completed stop of container iris on Sato-DATA-ICM-0002
Removing container iris on Sato-DATA-ICM-0001...
Removing container iris on Sato-DATA-ICM-0002...
...removed container iris on Sato-DATA-ICM-0001
...removed container iris on Sato-DATA-ICM-0002
Inspecting docker image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0001...
Inspecting docker image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0002...
...inspected docker image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0001
...inspected docker image hsato0719/iris:2020.1.0.215.0 on Sato-DATA-ICM-0002
Creating container iris on Sato-DATA-ICM-0001...
Creating container iris on Sato-DATA-ICM-0002...
...created container iris on Sato-DATA-ICM-0001
...created container iris on Sato-DATA-ICM-0002
Copying license directory /Samples/license/ to Sato-DATA-ICM-0001...
Copying CPF deltas to Sato-DATA-ICM-0002...
...copied license directory /Samples/license/ to Sato-DATA-ICM-0001
Copying CPF deltas to Sato-DATA-ICM-0001...
...copied CPF deltas to Sato-DATA-ICM-0002
...copied CPF deltas to Sato-DATA-ICM-0001
Starting container iris on Sato-DATA-ICM-0001...
Starting container iris on Sato-DATA-ICM-0002...
...completed start of container iris on Sato-DATA-ICM-0001
...completed start of container iris on Sato-DATA-ICM-0002
Waiting for InterSystems IRIS to start on Sato-DATA-ICM-0001...
Waiting for InterSystems IRIS to start on Sato-DATA-ICM-0002...
Elapsed time for waitstarted on Sato-DATA-ICM-0001: 15s...
Elapsed time for waitstarted on Sato-DATA-ICM-0002: 15s...
```

```
...InterSystems IRIS started on Sato-DATA-ICM-0001
Elapsed time for waitstarted on Sato-DATA-ICM-0002: 30s...
...InterSystems IRIS started on Sato-DATA-ICM-0002
Getting ICM state from Sato-DATA-ICM-0001...
Getting ICM state from Sato-DATA-ICM-0002...
...got ICM state from Sato-DATA-ICM-0001
...got ICM state from Sato-DATA-ICM-0002
Configuring Web Gateway on Sato-DATA-ICM-0001...
Configuring Web Gateway on Sato-DATA-ICM-0002...
...configured Web Gateway on Sato-DATA-ICM-0002
...configured Web Gateway on Sato-DATA-ICM-0001
Configuring ODBC on Sato-DATA-ICM-0001...
Configuring ODBC on Sato-DATA-ICM-0002...
...configured ODBC on Sato-DATA-ICM-0002
...configured ODBC on Sato-DATA-ICM-0001
Configuring JDBC on Sato-DATA-ICM-0001...
Configuring JDBC on Sato-DATA-ICM-0002...
...configured JDBC on Sato-DATA-ICM-0001
...configured JDBC on Sato-DATA-ICM-0002
Configuring ECP on Sato-DATA-ICM-0002...
Configuring ECP on Sato-DATA-ICM-0001...
...configured ECP on Sato-DATA-ICM-0002
...configured ECP on Sato-DATA-ICM-0001
Changing WebGateway password on Sato-DATA-ICM-0001...
Changing WebGateway password on Sato-DATA-ICM-0002...
...changed WebGateway password on Sato-DATA-ICM-0002
...changed WebGateway password on Sato-DATA-ICM-0001
Validating license on Sato-DATA-ICM-0001...
Acquiring license on Sato-DATA-ICM-0002...
...acquired license on Sato-DATA-ICM-0002
...validated license on Sato-DATA-ICM-0001
Acquiring license on Sato-DATA-ICM-0001...
...acquired license on Sato-DATA-ICM-0001
Enable sharding on Sato-DATA-ICM-0001...
Enable sharding on Sato-DATA-ICM-0002...
...enabled sharding on Sato-DATA-ICM-0002
...enabled sharding on Sato-DATA-ICM-0001
Initializing shard Sato-DATA-ICM-0001...
...initialized shard Sato-DATA-ICM-0001
Attaching shard Sato-DATA-ICM-0002 as data node....
...attached shard Sato-DATA-ICM-0002 as data node
Setting shard options on Sato-DATA-ICM-0002...
...set shard options on Sato-DATA-ICM-0002
Setting shard options on Sato-DATA-ICM-0001...
...set shard options on Sato-DATA-ICM-0001
Verifying shards on Sato-DATA-ICM-0001...
...verified shards on Sato-DATA-ICM-0001
Setting ICM state on Sato-DATA-ICM-0001...
Setting ICM state on Sato-DATA-ICM-0002...
...set ICM state on Sato-DATA-ICM-0001
...set ICM state on Sato-DATA-ICM-0002
-> Management Portal available at:
```

<http://satodataicm1.japaneast.cloudapp.azure.com:52773/csp/sys/UtilHome.csp>

最後のurlをブラウザで指定して管理ポータルにログインできます。

ユーザー名: system

パスワード: icm deployで指定したパスワード

9. test

icmを使った動作確認のためのサンプルを以下に示します。

```
$ icm ps -container iris
$ icm exec -command "df -k" -machine Sato-DATA-ICM-0001
$ icm session -interactive -machine Sato-DATA-ICM-0001 -namespace USER
$ icm scp -localPath local-path [-remotePath remote-path]

$ icm sql -command "create table order (orderid varchar(20), primary key (orderid), shard)" -role DATA
$ icm sql -command "create table lineitem (product varchar(20), orderid varchar(20), quantity integer, shard)" -role DATA
$ icm sql -command "create table product (name varchar(20), price number(7,2))" -role DATA
$ icm sql -command "create table product (name varchar(20), price number(7,2))" -role DATA

$ icm sql -command "insert into product values ('hammer',9.95)" -role DATA
$ icm sql -command "insert into product values ('nail',0.02)" -role DATA
$ icm sql -command "insert into product values ('pliers',8.95)" -role DATA
$ icm sql -command "insert into product values ('screw',0.05)" -role DATA
$ icm sql -command "insert into product values ('screwdriver',7.95)" -role DATA
$ icm sql -command "insert into product values ('saw',12.95)" -role DATA
$ icm sql -command "insert into order values ('A-001')" -role DATA
$ icm sql -command "insert into order values ('A-002')" -role DATA
$ icm sql -command "insert into order values ('B-001')" -role DATA
$ icm sql -command "insert into order values ('B-002')" -role DATA
$ icm sql -command "insert into order values ('B-003')" -role DATA
$ icm sql -command "insert into order values ('B-004')" -role DATA
$ icm sql -command "insert into order values ('B-005')" -role DATA
$ icm sql -command "insert into lineitem values ('hammer','A-001',1)" -role DATA
$ icm sql -command "insert into lineitem values ('nail','A-001',1000)" -role DATA
$ icm sql -command "insert into lineitem values ('pliers','A-002',1)" -role DATA
$ icm sql -command "insert into lineitem values ('screwdriver','B-001',1)" -role DATA
$ icm sql -command "insert into lineitem values ('screw','B-001',500)" -role DATA
$ icm sql -command "insert into lineitem values ('screwdriver','B-002',2)" -role DATA
$ icm sql -command "insert into lineitem values ('pliers','B-002',1)" -role DATA
$ icm sql -command "insert into lineitem values ('screw','B-002',1500)" -role DATA
$ icm sql -command "insert into lineitem values ('saw','B-003',1)" -role DATA
$ icm sql -command "insert into lineitem values ('saw','B-004',1)" -role DATA
$ icm sql -command "insert into lineitem values ('pliers','B-004',1)" -role DATA
$ icm sql -command "insert into lineitem values ('pliers','B-005',2)" -role DATA

$ icm sql -command "select avg(quantity) from lineitem" -role DATA
$ icm sql -command "select %id, * from order" -role DATA
$ icm sql -command "select %id, * from lineitem" -role DATA
$ icm sql -command "select order.orderid, product, quantity from order, lineitem where order.orderid=lineitem.orderid order by 1" -role DATA
$ icm sql -command "select orderid, sum(price*quantity) from lineitem, product where lineitem.product=product.name group by orderid" -role DATA

$ icm sql -command "select orderid, sum(price*quantity) from lineitem, product where lineitem.product=product.name group by orderid" -machine Sato-DATA-TEST-0001
```

```
$ icm sql -command "select orderid, sum(price*quantity) from lineitem, product where lineitem.product=product.name group by orderid" -machine Sato-DATA-TEST-0002
```

10. cleanUp

- これを忘れるとなクラウド課金し続けるので注意

```
icm unprovision -cleanUp -force
```

[#Artificial Intelligence \(AI\)](#) [#DevOps](#) [#Docker](#) [#クラウド](#) [#シャーディング](#) [#継続的インテグレーション](#)
[#継続的デリバリー](#) [#InterSystems IRIS](#)

ソースURL:

<https://jp.community.intersystems.com/post/icm%E3%82%92%E5%88%A9%E7%94%A8%E3%81%97%E3%81%A6iris%E3%82%AF%E3%83%A9%E3%82%B9%E3%82%BF%E3%83%BC%E3%82%92%E6%A7%8B%E6%88%90%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95>