



テクニカルホワイトペーパー

Veritas InfoScale Enterprise 7.4.1 for RHEL on AWS

構築手順書

マルチ AZ 環境での FSS + OverlayIP 切替 編

2019 年 11 月

ベリタステクノロジーズ合同会社

テクノロジーセールス本部

VERITAS™

The truth in information.

免責事項

ベリタステクノロジーズ合同会社は、この文書の著作権を留保します。また、記載された内容の無謬性を保証しません。Veritas InfoScale は将来に渡って仕様を変更する可能性を常に含み、これらは予告なく行われることもあります。なお、当ドキュメントの内容は参考資料として、読者の責任において管理/配布されるようお願いいたします。

目次

| | |
|--------------------------------------|----|
| 免責事項 | 2 |
| 1. はじめに | 4 |
| 本書の目的 | 4 |
| 2. システム構成 | 5 |
| 本書の前提となるシステム構成 | 5 |
| AWS 上に RHEL を構築する場合の注意点 | 6 |
| 3. インストール | 8 |
| インストールとクラスタの基本セットアップ | 8 |
| 4. VXVM と VXFS の設定 | 9 |
| ディスクの確認とイニシャライズ | 9 |
| ディスクのイニシャライズと EXPORT | 9 |
| VXVM のディスクグループの作成 | 10 |
| VXVM のボリュームと VXFS のファイルシステムの作成 | 10 |
| 既存のサービスグループ:CVM への設定追加 | 12 |
| 5. AWSCLI 及び IAM ロールの設定 | 14 |
| 本書の構成パターンに必要な IAM ロールの詳細 | 14 |
| 6. クラスタの設定 | 15 |
| APACHE のインストールと必要な設定 | 15 |
| サービスグループの作成 | 16 |
| 異なる AZ 間でネットワーク経路を切り替える為の設定 | 17 |
| 7. 動作確認と切り替えテスト | 23 |
| アプリケーションを管理するサービスグループの手動切り替え | 23 |
| 障害によるサービスグループの自動切り替え | 24 |
| 障害ノードの復旧 | 24 |
| アプリケーション (APACHE) 障害 | 24 |
| 8. 付録 | 25 |
| MAIN.CF の内容 | 25 |

1. はじめに

本書の目的

本書は、InfoScale Enterprise 7.4.1 を用いて、2 つの Availability Zone（以下 AZ と記述）を跨いで FSS を用いたミラーリングによるデータ同期を行い、且つ OverlayIP によるルートテーブルの切り替えによってネットワークを切り替える 2 ノード稼働待機型クラスター構成の構築の理解を目的に作成されています。尚、クラスタリング対象のアプリケーションとして apache を用いています。

2. システム構成

本書の前提となるシステム構成

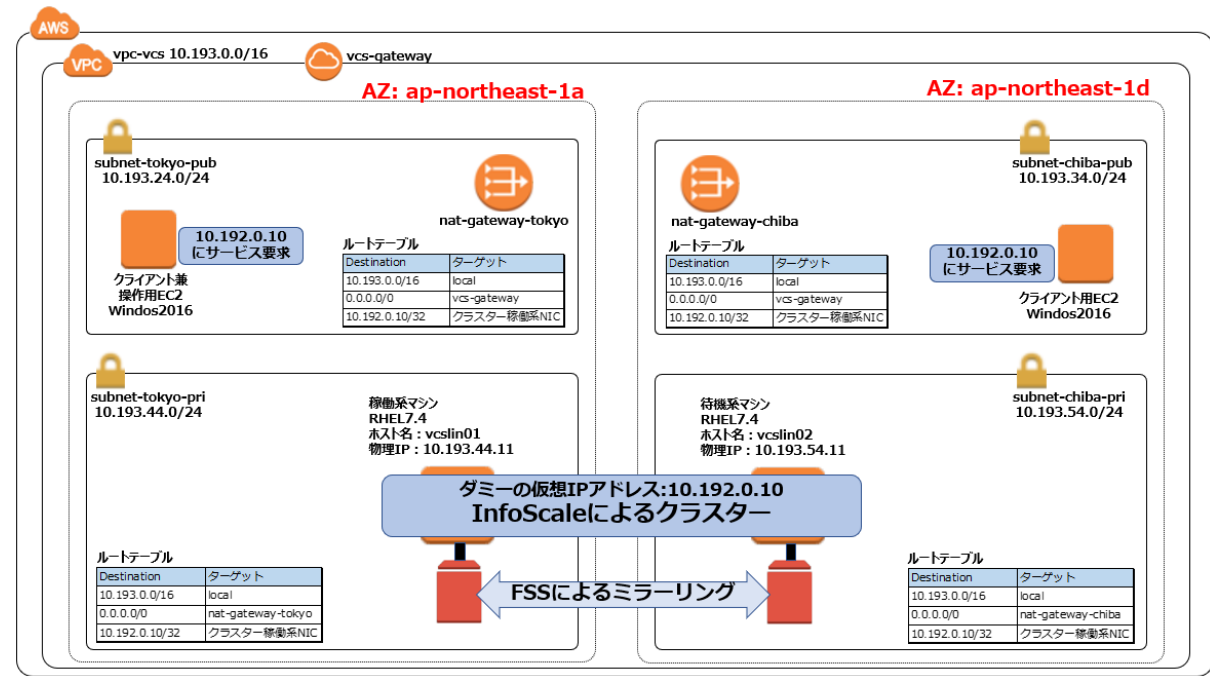


図 1InfoScale 構築後の構成イメージ

AWS 上に RHEL を構築する場合の注意点

InfoScale は、AWS 上の RHEL での稼働を保証しています。ただし、InfoScale を構築する場合は、以下の条件を満たすように RHEL を構築してください。

- **ノード間で ping が通ること**：InfoScale はインストーラー時や稼働時に、ping による相互監視を行います。しかし、AWS 上の RHEL はデフォルトで ping が通らない設定になっています。セキュリティグループの「インバウンドの設定」を変更し、ping が通るようにしてください。



- **RHEL のインスタンスに ssh を用いて root でパスワードを用いてログインできること**：InfoScale のインストール時は、1 台のノードから push install を行いますが、その際に他のノードに ssh を用いて root でパスワードを用いてログインします。AWS 上で deploy される RHEL は、デフォルトで ssh を用いて root でパスワードを用いてログインできるようになっていません。`/etc/ssh/sshd_config` や `/etc/passwd` を変更してパスワードを用いて root で ssh ができるようにしてください。
- **yum が使用できること**：AWS 上で deploy される RHEL には、InfoScale が必要とするパッケージの幾つかがインストールされていません。そのため、InfoScale のインストーラーの中で yum を用いて必要なパッケージをインストールします。適切なネットワークの設定（踏み台サーバーを経由して yum のサーバーにアクセス等）もしくは yum のリポジトリの設定を行い、yum が使用できるようにしてください。
- **AWSCLI が使用できること**：InfoScale は、プライベート IP を管理したりルートテーブルを切り替えるために AWSCLI を使用して EC2 インスタンスを IAM からコントロールします。InfoScale がインストールされるノードが NAT ゲートウェイを経由して IAM に接続できるようにし、AWSCLI が使用できるようにしてください。

- **swap があること** : AWS 上で deploy される RHEL には、swap がありません。InfoScale のインストール時は、swap が必須ですので、ファイルを作成して swap に割り当ててください。

```
# dd if=/dev/zero of=/home/swapfile bs=1024k count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 0.532228 s, 2.0 GB/s
# mkswap /home/swapfile
Setting up swapon version 1, size = 1048572 KiB
no label, UUID=f26daf30-7e86-47e3-9bf0-86c47352bac3
# swapon /home/swapfile
swapon: /home/swapfile: insecure permissions 0644, 0600 suggested.
# cat /proc/swaps
Filename                                Type              Size      Used      Priority
/home/swapfile                          file              1048572  0         -2
```

- **データ配置用の EBS が、両方のノードに割り当てられている事** : EBS は、共有ディスク構成をサポートせず、EBS は常に 1 つの EC2 インスタンスにのみ紐付けられます。また、AZ をまたいでクラスターを構成する場合は、AZ 間で EBS をやり取りする事ができません。InfoScale の FSS 構成では、各ノードのローカルディスク (EBS) をクラスターに属する全ノードで仮想的に共有し、ノードをまたいで冗長ボリュームを構成することで、AWS 上で共有ディスク構成と同じ環境を提供します。REHL を構築する際は、各ノードに、少なくとも 1 個以上のデータ格納用 EBS を割り当ててください。

3. インストール

インストールとクラスターの基本セットアップ

本書では、InfoScale Enterprise のインストールとクラスター部分の基本セットアップの説明を割愛します。インストールの詳しい手順については、既に公開済の「Veritas InfoScale Enterprise 7.4.1 for RHEL on AWS 構築手順書 EBS を用いた FSS 構成+ Private IP 切替 編」をご参照ください。これは、同一 AZ 内に 2 ノードクラスターを構築する際の手順を説明していますが、AZ 跨ぎであってもハートビート用ネットワークの疎通さえ確保できれば、手順は同じです。ドキュメントの URL は以下です。

https://www.veritas.com/content/support/en_US/doc/InfoScale7.4.1_RHEL_on_AWS_deploy_FSS_PrivateIP_JP

4. VxVM と VxFS の設定

ディスクの確認とイニシャライズ

まず、VxVM から認識されているディスクの確認とイニシャライズを行います。

オペレーションの準備

オペレーションを始める前に、両方のノードで PATH と MANPATH の設定を行ってください。

```
# PATH=$PATH:/opt/VRTS/bin export PATH
# MANPATH=$MANPATH/opt/VRTS/man export MANPATH
```

ディスクのイニシャライズと export

1 号機で認識できているディスクを確認してください。OS のディスク以外に、1 本のディスクが見えています。データ用にこのディスクを使用します。

```
# vxdisk list
DEVICE      TYPE          DISK    GROUP    STATUS
xvda        auto:none    -       -        online invalid
01_xen-vd0_1 auto:none    -       -        online invalid
```

前述のディスクをイニシャライズし、その後で export してください。

```
# vxdisksetup -i 01_xen-vd0_1
# vxdisk export 01_xen-vd0_1
```

2 号機でも、同じようにディスクのイニシャライズと export を行ってください。

```
# vxdisk list
DEVICE      TYPE          DISK    GROUP    STATUS
xvda        auto:none    -       -        online invalid
02_xen-vd0_1 auto:none    -       -        online invalid
# vxdisksetup -i 02_xen-vd0_1
# vxdisk export 02_xen-vd0_1
```

1 号機及び 2 号機で、vxctl enable でディスクの再走査を行った後に、再度 vxdisk list を実行すると、他方のノードのディスクが見えるようになります。下記は、1 号機の実出力例です。

```
# vxctl enable
# vxdisk list
DEVICE      TYPE          DISK    GROUP    STATUS
xvda        auto:none    -       -        online invalid
01_xen-vd0_1 auto:cdsdisk -       -        online remote
02_xen-vd0_1 auto:cdsdisk -       -        online exported
```

VxVM のディスクグループの作成

次に、ディスクグループを作成します。ディスクグループは、Linux の LVM で言う所の Volume Group に相当し、VxVM でボリューム（仮想デバイス）を作成して使用する場合は、ボリュームは必ずいずれかのディスクグループに所属する必要があります。本書のクラスターは、FSS 構成であるため、双方のノードの内蔵ディスクをお互いにシェアした fss ディスクグループを作成します。

ディスクグループの作成

1 号機で vxvg コマンドを用いて、以下の例に従って fssdg という名前のディスクグループを作成します。ディスクとして 01_xen-vd0_1 02_xen-vd0_1 の 2 本を使用しています。

```
# vxvg -s -o fss init fssdg 01_xen-vd0_1 02_xen-vd0_1
```

ディスクグループの作成に成功すると、vxdisk list コマンドの出力は以下のようになり、従来空欄だった "GROUP" の列にディスクグループ名である "awsdg" が表示されます。

```
# vxdisk list
DEVICE      TYPE          DISK          GROUP          STATUS
xvda        auto:none    -             -              online invalid
01_xen-vd0_1 auto:cdsdisk 01_xen-vd0_1 fssdg          online exported shared
02_xen-vd0_1 auto:cdsdisk 02_xen-vd0_1 fssdg          online shared remote
```

VxVM のボリュームと VxFS のファイルシステムの作成

次に、ボリュームを作成します。ボリュームは、Linux の LVM で言う所の Volume に相当します。ボリュームを作製したら、そこにファイルシステムを構築します。この作業は 1 号機のみで行います。

まず、VxVM のボリュームを作製します。下記例では、"fssdg" というディスクグループ内の "01_xen-vd0_0" と "02_xen-vd0_1" の 2 本のディスクを用いて、"vol01" という名前の容量 5Gbyte のミラーボリュームを作成しています。

```
# vxassist -g fssdg make vol01 5g layout=mirror 01_xen-vd0_1 02_xen-vd0_1
```

作製したボリュームは、vxprint コマンドで確認できます。

```
# vxprint

Disk group: fssdg

TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg fssdg         fssdg         -       -       -       -       -       -

dm lin01_xen-vd0_1 lin01_xen-vd0_1 - 16703232 - REMOTE - -
dm lin02_xen-vd0_1 lin02_xen-vd0_1 - 16703232 - - -

v vol01          fsgen          ENABLED 10485760 - ACTIVE - -
pl vol01-01      vol01          ENABLED 10485760 - ACTIVE - -
sd lin01_xen-vd0_1-01 vol01-01 ENABLED 10485760 0 - - -
pl vol01-02      vol01          ENABLED 10485760 - ACTIVE - -
sd lin02_xen-vd0_1-01 vol01-02 ENABLED 10485760 0 - - -
dc vol01_dco     vol01          -       -       -       -       - -
v vol01_dcl      gen            ENABLED 67840 - ACTIVE - -
pl vol01_dcl-01 vol01_dcl      ENABLED 67840 - ACTIVE - -
sd lin01_xen-vd0_1-02 vol01_dcl-01 ENABLED 67840 0 - - -
pl vol01_dcl-02 vol01_dcl      ENABLED 67840 - ACTIVE - -
sd lin02_xen-vd0_1-02 vol01_dcl-02 ENABLED 67840 0 - - -
```

作製されたボリューム上にファイルシステムを作製します。指定するデバイスパス名は /dev/vx/rdisk までが固定で、その後にディスクグループ名（下記例では fssdg）、最後にボリューム名（下記例では vol01）、というフォーマットです。

```
# mkfs -t vxfs /dev/vx/rdisk/fssdg/vol01
version 15 layout
10485760 sectors, 5242880 blocks of size 1024, log size 16384 blocks
rcq size 1024 blocks
largefiles supported
maxlink supported
WORM not supported
```

既存のサービスグループ:cvm への設定追加

本構成では、インストール時に SFCFSHA を指定しているため、自動的に cvm というサービスグループが作成されています。作成したファイルシステムを mount して使用するためには、このサービスグループにリソースを追加する必要があります。この作業は、特に断りがない限り 1 号機のみで行います。

まず、クラスターの構成を編集可能にするモード設定を行ってください。

```
[root@node01 ~]# haconf -makerw
```

下記では、使用するディスクグループ:fssdg とボリューム vol01 を管理・監視するリソース:voldg を追加しています。最後に、このリソースの online の条件が、既に構成済みのリソースである cvm_clus の online である事を設定しています。

```
# hares -add voldg CVMVolDg cvm
VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before agent
monitors
# hares -modify voldg CVMDiskGroup fssdg
# hares -modify voldg CVMActivation sw
# hares -modify voldg CVMVolume vol01
# hares -modify voldg Critical 0
# hares -modify voldg Enabled 1
# hares -link voldg cvm_clus
```

次に、ファイルシステムを mount するリソース : mount を作製します。このリソースの online の条件が、既に構成済みのリソースである voldg の online である事を設定しています。尚、このリソースを作製する際に、1 号機と 2 号機で mount point : /data を作製しておいてください。

```
# hares -add mount CFSSMount cvm
VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before agent
monitors
# hares -modify mount MountPoint /fss_aws
# hares -modify mount BlockDevice /dev/vx/dsk/fssdg/vol01
# hares -modify mount Enabled 1
# hares -link mount voldg
#
# mkdir /data
```

サービスグループ:cvm を 1 号機と 2 号機で online にします。

```
# hagr -online cvm -any
VCS NOTICE V-16-1-50735 Attempting to online group on system vcslin01
VCS NOTICE V-16-1-50735 Attempting to online group on system vcslin02
```

Online になったかどうかは、hastatus と df で確認してください。

```
# hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A vcslin01         RUNNING        0
A vcslin02         RUNNING        0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B cvm             vcslin01        Y         N              ONLINE
B cvm             vcslin02        Y         N              ONLINE

# df -h
Filesystem          Size  Used Avail Use% Mounted on
/dev/xvda2          10G  6.1G  4.0G  61% /
devtmpfs            3.8G   0  3.8G   0% /dev
tmpfs               3.9G   0  3.9G   0% /dev/shm
tmpfs               3.9G  17M  3.9G   1% /run
tmpfs               3.9G   0  3.9G   0% /sys/fs/cgroup
/dev/vx/dsk/fssdg/vol01 5.0G  46M  4.7G   1% /data
tmpfs               783M   0  783M   0% /run/user/0
```

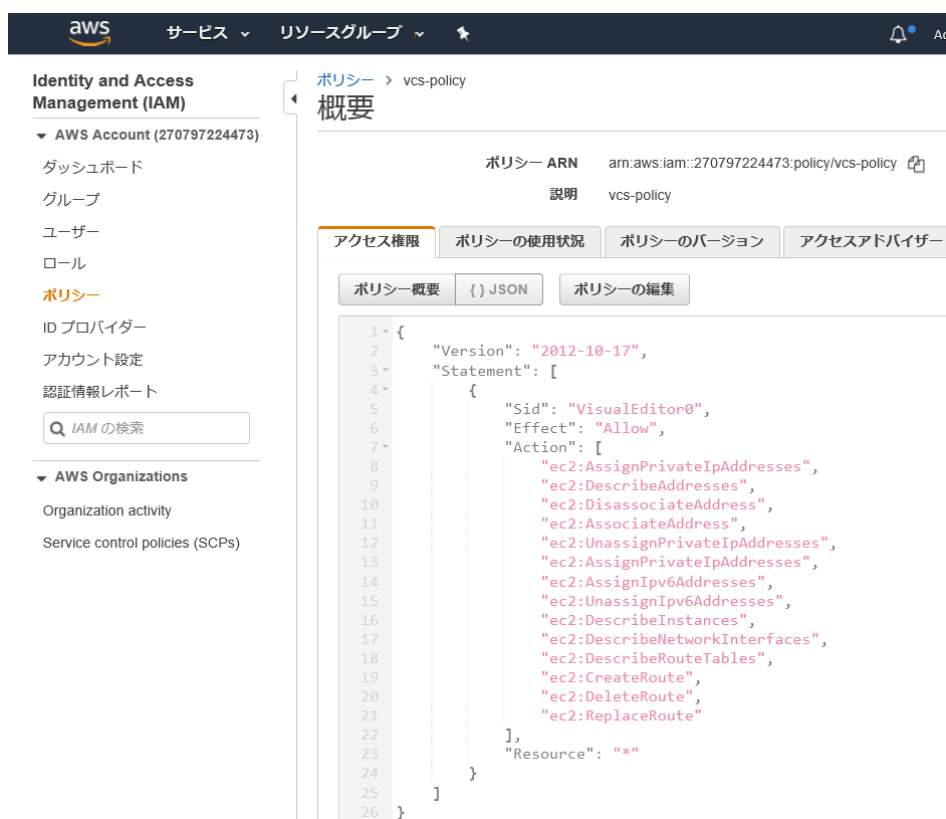
5. AWSCLI 及び IAM ロールの設定

本書では、AWSCLI の設定及び IAM ロール設定の説明を割愛します。これらの詳しい手順については、既に公開済の「Veritas InfoScale Enterprise 7.4.1 for RHEL on AWS 構築手順書 EBS を用いた FSS 構成 + Private IP 切替 編」の 19～25 ページをご参照ください。ただし、必要な IAM ロールの内容は異なりますので、その部分のみ後述します。ドキュメントの URL は以下です。

https://www.veritas.com/content/support/en_US/doc/InfoScale7.4.1_RHEL_on_AWS_deploy_FSS_PrivateIP_JP

本書の構成パターンで必要な IAM ロールの詳細

本書の構成では、プライベート IP の立ち上げとルートテーブルの更新を InfoScale から制御する必要があります。従って、以下の内容でポリシーを作成し、それに関連付けたロールを、InfoScale がインストールされたインスタンスに割り当ててください。



The screenshot shows the AWS IAM console interface. On the left, the 'Identity and Access Management (IAM)' menu is visible, with 'ポリシー' (Policies) selected. The main content area displays the details for a policy named 'vcs-policy'. The policy ARN is 'arn:aws:iam::270797224473:policy/vcs-policy' and the description is 'vcs-policy'. Below this, there are tabs for 'アクセス権限', 'ポリシーの使用状況', 'ポリシーのバージョン', and 'アクセスアドバイザー'. The 'ポリシー概要' (Policy Summary) tab is active, showing the JSON policy document. The policy grants 'Allow' permissions for a list of EC2 actions on all resources ('*').

```

1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Sid": "VisualEditor0",
6-       "Effect": "Allow",
7-       "Action": [
8-         "ec2:AssignPrivateIpAddresses",
9-         "ec2:DescribeAddresses",
10-        "ec2:DisassociateAddress",
11-        "ec2:AssociateAddress",
12-        "ec2:UnassignPrivateIpAddresses",
13-        "ec2:AssignPrivateIpAddresses",
14-        "ec2:AssignIpv6Addresses",
15-        "ec2:UnassignIpv6Addresses",
16-        "ec2:DescribeInstances",
17-        "ec2:DescribeNetworkInterfaces",
18-        "ec2:DescribeRouteTables",
19-        "ec2:CreateRoute",
20-        "ec2>DeleteRoute",
21-        "ec2:ReplaceRoute"
22-      ],
23-       "Resource": "*"
24-     }
25-   ]
26- }

```

6. クラスターの設定

いよいよ、AWS 上の RHEL 上で、アプリケーション及びその稼働に必要な個々のコンポーネントが、InfoScale によって適切に監視/リカバリされるための設定を行います。この作業は、どちらか一方ノードから行います。

apache のインストールと必要な設定

本書では、クラスタリング対象のアプリケーションとして apache を用いています。ここでは、apache のインストールとクラスタリングの為に必要な apache の微調整を行います。

まず、以下の例に従って両方のノードで apache をインストールしてください。

```
# yum install httpd
Loaded plugins: amazon-id, product-id, rhui-lb, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.6-90.el7 will be installed
--> Processing Dependency: httpd-tools = 2.4.6-90.el7 for package: httpd-2.4.6-90.el7.x86_64
--> Running transaction check
--> Package httpd-tools.x86_64 0:2.4.6-90.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version        Repository      Size
=====
Installing:
httpd                   x86_64        2.4.6-90.el7   rhel-7-server-rpms    1.2 M
Installing for dependencies:
httpd-tools             x86_64        2.4.6-90.el7   rhel-7-server-rpms    91 k
=====

Transaction Summary
=====
Install 1 Package (+1 Dependent package)
:
以下略
:
```

apache のコンテンツは、デフォルトで /var/www 配下に格納されます。しかし、クラスタリングする場合は、クラスターの仮想共有ストレージ内に格納されべきです。まず、1号機で /var/www ディレクトリを仮想共有ストレージ領域である /data 配下に移動し、/var/www にアクセスすると /data/www を参照するようにシンボリックリンクを設定します。

```
# mv /var/www /data
# ln -s /data/www /var/www
# ls -l /var | grep www
lrwxrwxrwx. 1 root root 9 Nov 15 05:49 www -> /data/www
```

2号機でも同じ設定が必要ですが、既上記の作業によって仮想共有ストレージ領域である /data 配下に www ディレクトリが存在しています。従って、/var/www を消去し /var/www にアクセスすると /data/www を参照するようなシンボリックリンクの設定を行ってください。

```
# rm -rf /var/www
# ln -s /data/www /var/www
# ls -l /var | grep www
lrwxrwxrwx. 1 root root 9 Nov 15 05:57 www -> /data/www
```

インストールと apache 関連の微調整の後に、以下の例に従って apache の動作確認を行います。まず、"systemctl start httpd.service" で apache が立ち上がり、プロセス ID が /etc/httpd/run/httpd.pid に格納される事を確認します。次に、"systemctl stop httpd.service" で apache が停止し、プロセス ID ファイル：/etc/httpd/run/httpd.pid が削除される事を確認します。この作業は、両方のノートで行ってください。

```
# ps -ef | grep httpd
root 29275 17759 0 05:58 pts/0 00:00:00 grep --color=auto httpd

# systemctl start httpd.service

# ps -ef | grep httpd
root 29484 1 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 29485 29484 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 29486 29484 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 29487 29484 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 29488 29484 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
apache 29489 29484 0 05:59 ? 00:00:00 /usr/sbin/httpd -DFOREGROUND
root 29504 17759 0 05:59 pts/0 00:00:00 grep --color=auto httpd
# cat /etc/httpd/run/httpd.pid
29484

# systemctl stop httpd.service

# ps -ef | grep httpd
root 29767 17759 0 06:00 pts/0 00:00:00 grep --color=auto httpd

# cat /etc/httpd/run/httpd.pid
cat: /etc/httpd/run/httpd.pid: No such file or directory
```

サービスグループの作成

アプリケーション及びその稼働に必要な個々のコンポーネントの集合体を「サービスグループ」と呼びます。切り替えは、このサービスグループ単位で行います。

まず、クラスターの構成を編集可能にするためのモード設定を行ってください。

```
# haconf -makerw
```

サービスグループ：websg を作製します。このサービスグループは、クラスターが立ち上がる際に、自動的にオンラインになる設定にしています。自動的にオンラインにたくない場合は、「hagrp -modify websg AutoStartList vcslin01 vcslin02」を省略してください。また、サービスグループ: testsg が online になる条件として、同じノードでサービスグループ:cvm である事を設定しています。

```
# hagrp -add websg
VCS NOTICE V-16-1-10136 Group added; populating SystemList and setting the Parallel
attribute recommended before adding resources
# hagrp -modify websg Parallel 0
# hagrp -modify websg SystemList vcslin01 0 vcslin02 1
# hagrp -modify websg AutoStartList vcslin01 vcslin02
# hagrp -link websg cvm online local firm
```

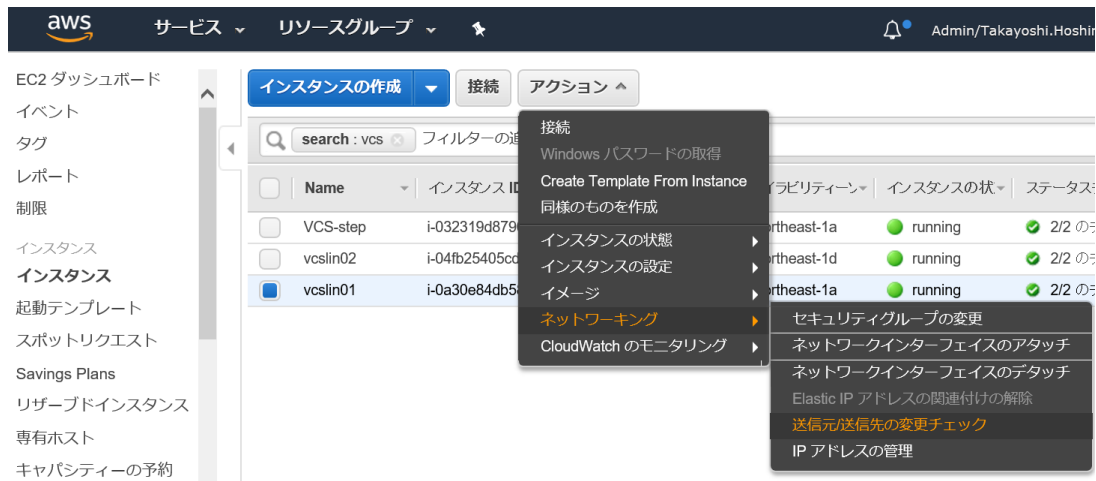
以下の例に従って、作成したサービスグループ：websg 内に apache を制御するリソース:web を作製します。apache を起動する際に実行するコマンドとして “/bin/systemctl start httpd.service” を、apache を停止する際に実行するコマンドとして “/bin/systemctl stop httpd.service” を、apache の稼働確認をプロセス ID ファイル：/etc/httpd/run/httpd.pid によって行うように設定しています。

```
# hares -add app Application websg
VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before agent
monitors
# hares -modify app StartProgram "/bin/systemctl start httpd.service"
# hares -modify app StopProgram "/bin/systemctl stop httpd.service"
# hares -modify app PidFiles /etc/httpd/run/httpd.pid
# hares -modify app Enabled 1
```

異なる AZ 間でネットワーク経路を切り替える為の設定

次に、異なる AZ 間でネットワーク経路を切り替える為の設定を行います。AZ が異なる場合はサブネットも異なりますので、AZ 間で同じ PrivateIP を切り替えることはできません。ElasticIP を切り替えることはできますが、通常、クラスタリングを行うような重要なシステムに ElasticIP を振る事は一般的ではありません。このような条件下で、クラスターが切り替わったことを、クライアントに意識させずに業務を継続するには、特別な手法が必要です。InfoScale は、AWS のルートテーブルを切り替える事によって、この課題を解決します。これから説明する例では、クラスターが配置される VPC 内のクライアントは 10.192.0.10 という IP を用いて、Web サーバーである apache にアクセスを試みます。InfoScale は、クラスターが配置される VPC 内の全ルートテーブルを更新し、10.192.0.10 のターゲットを、クラスターの稼働系ノードのサービス用 NIC にリダイレクトします。クラスターが切り替わると、新たに稼働系になったノードのサービス用 NIC に 10.192.0.10 向けの packets がリダイレクトされるように、ルートテーブルを更新します。また、クラスターの稼働系では、10.192.0.10 が立ち上がっており、クライアントに packets を送信する際の送信元アドレスになります。ここで使用する IP：10.192.0.10 を OverlayIP とよび、この方式を OverlayIP 方式とよびます。尚、**OverlayIP は必ず VPC に関連付けられた CIDR ブロックに含まれない IP にしてください。**本書の例では、OverlayIP を VPC に関連付けられた CIDR ブロック：10.193.0.0/16 に含まれないような IP：10.192.0.10 にしています。

OverlayIP 方式を使用するには、クラスターがインストールされるインスタンスの設定の変更が必要です。ここでは、その設定変更を行いますクラスターがインストールされるインスタンスを選び、“ネットワーク” から “送信元/送信先の変更チェック” を指定します。



デフォルトでは、インスタンスのソース/宛先チェックが有効になっていますので、無効にします。



これで、OverlayIP 方式を使用するための AWS 側の設定は完了です。

Overlay 方式を使用して、異なる AZ 間でネットワーク経路を切り替える為に、フェイルオーバーサービスグループ（今回の例では websg）内に、リソースを 3 つ作成します。設定が完了すると、クライアントのブラウザから OverlayIP にアクセスすると、クラスターの稼働系のサービス用 NIC に繋がり、Web にアクセスできます。Apache によって、仮想共有ストレージ領域である /data 配下のコンテンツを使用した Web サービスを楽しむ事ができます。また、クラスターの切り替えが発生した場合、クライアントは切り替え発生を意識することなく、切り替え先のノードから Web サービスを楽しむことができます。

まず、作成したサービスグループ：websg 内にリソース：nic を作製します。このリソースは、サービス用 OverlayIP アドレスを立ち上げる NIC を監視します。監視対象 NIC は eth0 です。

```
# hares -add nic NIC websg
VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before agent monitors
# hares -modify nic Device eth0
# hares -modify nic Enabled 1
```

次にリソース : ip を作製します。このリソースは、サービスで使用する OverlayIP アドレスを立ち上げ、それを監視します。IP アドレスは 10.192.0.10 です。繰り返しになりますが、OverlayIP は必ず VPC に関連付けられた CIDR ブロックに含まれない IP にしてください。このリソースの online の条件が、既に構成済みのリソースである nic の online である事を設定します。

```
# hares -add ip IP websg
# hares -modify ip Device eth0
# hares -modify ip Address 10.192.0.10
# hares -modify ip NetMask 255.255.255.0
# hares -modify ip Enabled 1
# hares -link ip nic
```

次のリソース設定には、同一 VPC 内で、クラスターにサービス要求を行うクライアントが属する全てのサブネットに紐付けられたルートテーブルの ID が必要になります。AWS の管理コンソールで、クラスターにサービス要求を行うクライアントが属するサブネットを指定し、それに紐付けられたルートテーブルの ID を確認します。本書の例の場合、クラスターにサービス要求を行うクライアントが属するサブネットは、subnet-tokyo-pub と subnet-chiba-pub であり、それらに紐付けられたルートテーブルの ID は、それぞれ rtb-043db90c88de26141 と rtb-0c0ff5a2d59917dd1 である事が確認できました。



ルートテーブル ID が確認出来たら、ルートテーブルを更新する次にリソース：awsip を作製します。OverlayIP には 10.192.0.10/32 を指定します。**アドレスの最後に "/32" を付ける事に注意してください**。Device には eth0 を指定します。AWSBinDir には AWSCLI を構成する際に設定した「aws コマンドが格納された絶対パス」を正確に指定します。RouteTableIds には、全ページで確認してルートテーブル ID：rtb-043db90c88de26141 と rtb-0c0ff5a2d59917dd1 を指定してください。最後に、このリソースの online の条件が、既に構成済みのリソースである ip の online である事を設定します。

```
# hares -add awsip AWSIP websp
# hares -modify awsip OverlayIP 10.192.0.10/32
# hares -modify awsip Device eth0
# hares -modify awsip AWSBinDir /usr/local/bin
# hares -modify awsip RouteTableIds rtb-043db90c88de26141 rtb-0c0ff5a2d59917dd1
# hares -modify awsip Enabled 1
# hares -link awsip ip
```

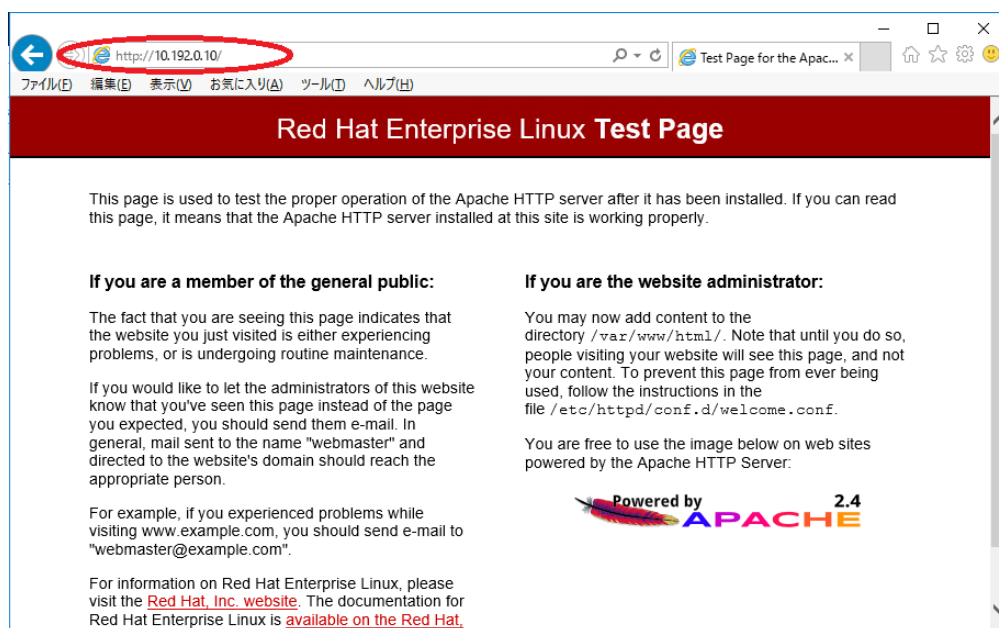
ネットワーク関連のリソース設定が終了したら、apache の online の条件が ルートテーブルの更新、つまり awsip の online である事を設定します。

```
# hares -link app awsip
```

これで、サービスグループ: websp の設定は完了です。早速、このサービスグループを 1 号機で online にします。

```
# hagrps -online websp -sys vcslin01
```

クライアントのブラウザから、OverlayIP：10.192.0.10 を指定して Web アクセスできることを確認してください。



InfoScale が稼働しているノードで、サービスグループが 1 号機で online になっており、1 号機で OverlayIP : 10.192.0.10 が立ち上がっており、且つ apache が動作していることを確認します。

```
# hastatus -sum

-- SYSTEM STATE
-- System          State          Frozen
A vcslin01         RUNNING       0
A vcslin02         RUNNING       0

-- GROUP STATE
-- Group           System          Probed         AutoDisabled   State
B cvm             vcslin01        Y              N               ONLINE
B cvm             vcslin02        Y              N               ONLINE
B websg          vcslin01        Y              N               ONLINE
B websg          vcslin02        Y              N               OFFLINE

# ifconfig -a | grep 10.192.0.10
    inet 10.192.0.10 netmask 255.255.255.0 broadcast 10.192.0.255

# ps -ef | grep httpd
root      28405      1  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    28406  28405  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    28407  28405  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    28408  28405  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    28409  28405  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    28410  28405  0 07:45 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    29006  28405  0 07:47 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
root      30434  17759  0 07:51 pts/0    00:00:00 grep --color=auto httpd
```

AWS のコンソールでルートテーブルの 10.192.0.10/32 のターゲットの NIC の ID が、1 号機の eth0 の NIC ID になっている事を確認してください。



最後に、必ずクラスターの構成をディスクに反映させ、編集不可にするモード設定を行ってください。これを行わないと、サーバー再起動やクラスターの停止時に、そこまでの設定変更が無効になってしまいます。

```
# haconf -dump -makero
```

これで、全ての設定は完了です。

7. 動作確認と切り替えテスト

最後に、AWS 上の RHEL 上で、apache 及びその稼働に必要な個々のコンポーネントが、InfoScale によって適切に監視/リカバリーされるかを確認するための簡単なテストを行います。

アプリケーションを管理するサービスグループの手動切り替え

下記は、GUI から websg のサービスグループを、手動で vcslin02 に切り替えるオペレーション例です。

```
# hagr -switch websg -to vcslin02
```

2号機への切り替えが正常に完了すると、hastatus の結果は以下のようになります。

```
# hastatus -sum
```

| -- SYSTEM STATE | | |
|-----------------|---------|--------|
| -- System | State | Frozen |
| A vcslin01 | RUNNING | 0 |
| A vcslin02 | RUNNING | 0 |

| -- GROUP STATE | | | | |
|----------------|----------|--------|--------------|---------|
| -- Group | System | Probed | AutoDisabled | State |
| B cvm | vcslin01 | Y | N | ONLINE |
| B cvm | vcslin02 | Y | N | ONLINE |
| B websg | vcslin01 | Y | N | OFFLINE |
| B websg | vcslin02 | Y | N | ONLINE |

2号機への切り替えが正常に完了すると、hastatus の結果は以下のようになります。

クライアントのブラウザから、OverlayIP：10.192.0.10 を指定して Web アクセスできることを確認すると共に、AWS のコンソールでルートテーブルの 10.192.0.10/32 のターゲットの NIC の ID が変更されている事を確認してください。

The screenshot shows the AWS Management Console interface for configuring a route table. The 'Route Table' tab is selected, and a route is shown with destination 10.192.0.10/32 and target eni-0d6cead587761bcc4. Red circles highlight the subnet name 'subnet-tokyo-pub', the 'Route Table' tab, and the route details.

障害によるサービスグループの自動切り替え

apache が稼働している方の RHEL インスタンスを、AWS の EC2 のコンソールから停止してください。その後で、生き残ったノードに正しくサービスグループが切り替わって、クライアントから 10.192.0.10 に Web アクセスできる事を確認します。また、AWS のコンソールでルートテーブルが更新された事も確認してください。

障害ノードの復旧

上記のテストで停止したノードを AWS の EC2 のコンソールから再立ち上げし、立ち上がったノードにサービスグループを手動で切り替えてください。この前と後で、各ノードの ifconfig や df や hastatus の表示及びルートテーブルの内容がどう変わったか確認してください。

```
# hagrpswitch webmsg -to vcslin02
```

アプリケーション (apache) 障害

apache が稼働しているノードで、以下のコマンドで apache を強制終了してください。その後、障害が検知され、他方のノードに正しくサービスグループが切り替わっていることを確認してください。

```
# systemctl stop httpd.service
```

8. 付録

main.cf の内容

本書の手順で構成した場合、main.cf ファイルは以下の内容となります。尚、main.cf のパスは、`/etc/VRTSvcs/conf/config/main.cf` です。

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CSSD.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"
include "SybaseTypes.cf"

cluster aws (
    UserNames = { admin = dqrJqIQnrMrrPzrLqo }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
)

system vcslin01 (
)

system vcslin02 (
)

group cvm (
    SystemList = { vcslin01 = 0, vcslin02 = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { vcslin01, vcslin02 }
)

CFSMount mount (
    MountPoint = "/data"
    BlockDevice = "/dev/vx/dsk/fssdg/vol01"
    MountOpt = "cluster"
)

CFSfsckd vxfsckd (
)

CVMCluster cvm_clus (
    CVMClustName = aws
    CVMNodeid = { vcslin01 = 0, vcslin02 = 1 }
    CVMTransport = gab
    CVMTimeout = 200
)

CVMVolDg voldg (
    Critical = 0
    CVMDiskGroup = fssdg
    CVMVolume = { vol01 }
    CVMActivation = sw
)

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
```

```

        CVMVxconfigdArgs = { syslog }
    )

    ProcessOnOnly vxattachd (
        Critical = 0
        PathName = "/bin/sh"
        Arguments = "- /usr/lib/vxvm/bin/vxattachd root"
        RestartLimit = 3
    )

    cvm_clus requires cvm_vxconfigd
    mount requires voldg
    voldg requires cvm_clus
    vxfsckd requires cvm_clus

// resource dependency tree
//
//     group cvm
//     {
//     CFSMount mount
//     {
//         CVMVoldg voldg
//         {
//             CVMCluster cvm_clus
//             {
//                 CVMVxconfigd cvm_vxconfigd
//             }
//         }
//     }
//     ProcessOnOnly vxattachd
//     CFSfsckd vxfsckd
//     {
//         CVMCluster cvm_clus
//         {
//             CVMVxconfigd cvm_vxconfigd
//         }
//     }
// }

group websg (
    SystemList = { vcslin01 = 0, vcslin02 = 1 }
    AutoStartList = { vcslin01, vcslin02 }
)

AWSIP awsip (
    OverlayIP = "10.192.0.10/32"
    Device = eth0
    RouteTableIds = { rtb-043db90c88de26141, rtb-0c0ff5a2d59917dd1 }
    AWSBinDir = "/usr/local/bin"
)

Application app (
    StartProgram = "/bin/systemctl start httpd.service"
    StopProgram = "/bin/systemctl stop httpd.service"
    PidFiles = { "/etc/httpd/run/httpd.pid" }
)

IP ip (
    Device = eth0
    Address = "10.192.0.10"
    NetMask = "255.255.255.0"
)

NIC nic (
    Device = eth0

```

```
)

requires group cvm online local firm
app requires awsip
awsip requires ip
ip requires nic

// resource dependency tree
//
//   group websp
//   {
//   Application app
//   {
//     AWSIP awsip
//     {
//       IP ip
//       {
//         NIC nic
//       }
//     }
//   }
// }
// }
```

ベリタステクノロジーズについて

Veritas Technologies はエンタープライズデータ管理のグローバルリーダーです。複雑化した IT 環境においてデータ管理の簡素化を実現するために、世界の先進企業 50,000 社以上、Fortune 500 企業の 90 パーセントが、ベリタスのソリューションを導入しています。ベリタスのエンタープライズ・データサービス・プラットフォームは、お客様のデータ活用を推進するため、データ保護の自動化とデータリカバリを実現して、ビジネスに不可欠なアプリケーションの可用性を確保し、複雑化するデータ規制対応に必要なインサイトを提供します。ベリタスのソリューションは信頼性とスケーラビリティに優れ、500 以上のデータソースと 50 のクラウドを含む 150 以上のストレージ環境に対応しています。



ベリタステクノロジーズ合同会社

<https://www.veritas.com/ja/jp>

〒107-0052 東京都港区赤坂 1-11-44 赤坂インターシティ 4F

ベリタスセールスインフォメーションセンター（法人のお客様向け製品購入に関する相談窓口）

■電話受付時間：10:00～12:00, 13:00～17:00（土、日、祝日、年末年始を除く）

■電話番号：0120-907-000（IP 電話からは 03-4531-1799）