【Practical Tips for SPARC】

第 6 回：SPARC Servers / Solaris 上で IoT 分析環境構築（2/5）　～ロードバランサーゾーンの構築～

2018 年 3 月

今回は前回作成した dmz ドメインにロードバランサー機能を実装していきます。ロードバランサーは Solaris 付属の Integrated Load Balancer（以下 ILB)を使用します。

Integrated Load Balancer とは

ILB は Solaris11 付属の L3/L4 ソフトウェアロードバランサーです。ネットワーク（IP）およびトランスポート（TCP/UDP）レイヤーで動作します。仮想アドレスで受信したパケットを規則に従ってバックエンドのサーバに転送します。
動作モードとしては DSR モードと NAT モードに対応しています。
ILB はクラスタソフトを必要とすることなく、それ自身で Active-Standby 型の冗長構成とすることが可能です。そのためには ILB と同じく OS の標準機能である VRRP 機能を使って仮想アドレスを付け替えることで実現しています。Standby 側は複数台用意することも可能です。
転送先サーバの健全性検査に関しては、ping などのほかにユーザー作成スクリプトによるアプリケーションレベルの検査にも対応しています。
その他 ILB の詳しい内容はマニュアルを参照してください。
http://docs.oracle.com/cd/E62101_01/html/E62581/gmrcp.html#scrolltoc
VRRP に関しては以下を参照してください。
http://docs.oracle.com/cd/E62101_01/html/E62581/gkfkc.html#scrolltoc

ILB の利用シーン

今までも ILB 含めいくつかの L3/L4 ソフトウェアロードバランサーがありましたが、それほど多く使われてきたわけではありません。一般的にはハードウェア型のロードバランサーが使われてきました。これはハードウェア型のほうが機能的及び性能的に優れていたことがあると思われます。
一般的に負荷分散で問題となるのは SessionID を使用した Stateful なセッションの制御です。この制御には同一のセッションを確実に同一のサーバに転送する必要があるため http 層の内容を見る必要があり、そのためにはロードバランサーが L7 に対応している必要がありました。またこのような処理はほとんどが暗号化されており、http 層の解析のためには暗号化と複合化の機能も必要とされました。これらの要件を効率的に満たすためにはハードウェア型のロードバランサーのほうが良かったという状況もありました。
ただ、近年では REST などの Stateless な通信が増えてきており、必ずしも L7 での制御が必要とされなくなってきています。また暗号化に関しても、CPU の処理能力の向上により、サーバ側で暗号化/複合化を行ってもパフォーマンスに与える影響が減ってきました。
同時に、Software Designed Network（SDN)のように柔軟な構成を求める声も増えてきました。
こういったことから今後は L3/L4 ロードバランサーの利用が増えていくのではと思われます。以下に利用シーンをまとめておきます。

- シンプルな HTML ページや画像データなどの静的なコンテンツへのアクセス。
- REST やセッション管理の不要な CGI アクセスのような Stateless な通信。

ILB の設計について

ILB の構成設計を行うにあたっては、以下に留意します。

動作モードについて

ILB ではまず動作モードを決める必要があります。動作モードには DSR モードと NAT モードがありますが、どちらのモードを使用するかで Web サーバのネットワーク接続が変わります。
NAT モードの場合、Web サーバはリクエスト結果を ILB に返信するため、特殊なネットワーク設定をする必要はありません。ただし、リクエストは受信時も送信時も常に ILB を経由するため、ILB の負荷が重くなるというデメリットがあります。
NAT モードではステートフルな接続を行いますので、簡単な CGI のようなシステムに向いていると思われます。
DSR モードの場合、Web サーバはリクエスト結果をクライアントに直接返信することになります。そのため、Web サーバはインターネット LAN と DMZ 向け LAN の両方に接続し、特別な設定が必要となります。ただし、リクエスト結果の送信時に ILB を経由しないため、ILB の負荷が軽くなるというメリットがあります。DSR モードでは完全にステートレスな通信となるため、REST などのシンプルなアクセスに向いていると思われます。
今回はシンプルなリクエストを想定しているため、処理の軽い DSR モードを使用します。

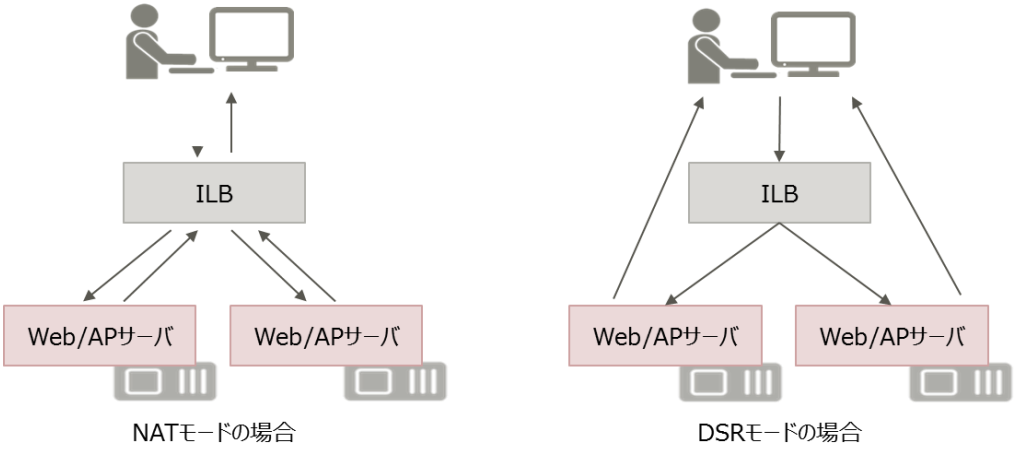

図1 動作モードイメージ

分散アルゴリズムについて

ILB には以下の 4 種類の分散アルゴリズムがあります。

- ラウンドロビン – 対象のサーバをローテーションベースでリクエストを転送します。
- src-IP ハッシュ – リクエストの発信元 IP アドレスのハッシュ値に基づいて転送するサーバを選択します。
- src-IP, port ハッシュ – リクエストの発信元 IP アドレスおよび発信元ポートのハッシュ値に基づいて転送するサーバを選択します。
- src-IP, VIP ハッシュ – リクエストの発信元 IP アドレスおよび着信先 IP アドレスのハッシュ値に基づいて転送するサーバを選択します。

分散アルゴリズムは以下を基準に選択します。
イントラネットや少数のクライアントが対象であればラウンドロビンや src-IP ハッシュでよいでしょう。
Proxy 経由でクライアントからリクエストがくるような場合は src-IP, port ハッシュを使用します。
今回は Proxy 経由での接続を想定して src-IP, port ハッシュを使用します。

冗長性について

ILB は VRRP 機能を使用して Active-Standby 形式で切り替えを行います。Standby 側は複数設定が可能です。(論理的には 254 台)　今回は 3 ノードで切り替えを行います。

Solaris の VRRP ルーターは L2 ベースの他に L3 ベースでも構築が可能です。詳しい違いはマニュアルを参照してください。

今回は IPMP での冗長化を行う場合を考慮して L3 の VRRP ルーターを使用します。

健全性検査について

ILB は転送先のサーバやサービスの状態を検査し、正常に通信ができない状況であればそのサーバやサービスへのパケットの転送を停止します。

ILB では以下の検査方法が利用可能です。

- 組み込み ping プローブ
- 組み込み TCP プローブ
- 組み込み UDP プローブ
- 健全性検査として実行できるユーザー指定のカスタムテスト

ping プローブ、TCP プローブ、UDP プローブはネットワークレベルのチェックしかできないため、サーブレットなどと通信先のアプリケーションサーバの状態を検査する場合にはユーザープログラムなどを作成する必要があります。

ping プローブはデフォルトではその他のプローブより先に実行されます。ping プローブが成功した場合のみ、その他のプローブは実行されます。

ロードバランサーゾーンの構成

今回、図 2 のような動作を実現するようロードバランサーを構成します。ILB は各物理サーバに 1 台配置し、計 3 台で HA 構成とします。

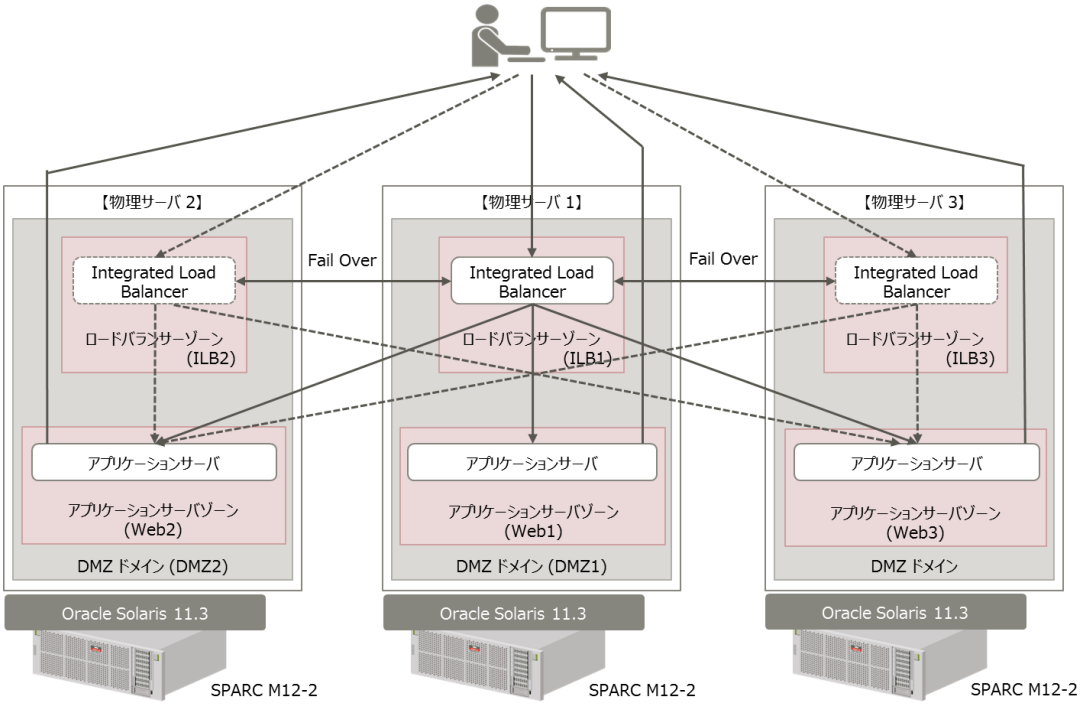

図2 論理構成イメージ

ロードバランサーゾーンの構成イメージは以下の通りです。

図3 構成図

ロードバランサーゾーンの構築

以下の手順は ilb1 だけでなく ilb2 と ilb3 も同様に設定します。

ロードバランサーゾーンのインストール

dmz ドメイン上に root ユーザーでログインします。
物理ポートとリンク名の確認を行います。ここで表示される物理ポートの vnet0 と vnet1 は、dmz ドメイン作成時に追加した vnet ではなく、dmz ドメインに追加された順番に OS によって指定される物理ポートの名前です。

```
# dladm show-phys
LINK              MEDIA             STATE      SPEED  DUPLEX    DEVICE
net1              Ethernet          up         0      unknown   vnet1
net0              Ethernet          up         0      unknown   vnet0
```

以下のようにロードバランサーゾーンを構成します。zone 名は ilb とします。ネットワーク設定は anet で GZ 上のデータリンクとロードバランサーゾーンのデータリンクを結び付けています。

```
# zonecfg -z ilb
zonecfg:ilb> create -b
zonecfg:ilb> set brand=solaris
zonecfg:ilb> set zonepath=/zones/ilb
zonecfg:ilb> set ip-type=exclusive
zonecfg:ilb> set autoboot=false
zonecfg:ilb> add dedicated-cpu
zonecfg:ilb> set ncpus=16
zonecfg:ilb> end
zonecfg:ilb> add anet
zonecfg:ilb:anet> set linkname=net0
zonecfg:ilb:anet> set lower-link=net0
zonecfg:ilb:anet> set configure-allowed-address=true
zonecfg:ilb:anet> set link-protection=mac-nospoof
zonecfg:ilb:anet> set mac-address=auto
zonecfg:ilb:anet> end
zonecfg:ilb> add anet
zonecfg:ilb:anet> set linkname=net1
zonecfg:ilb:anet> set lower-link=net1
zonecfg:ilb:anet> set configure-allowed-address=true
zonecfg:ilb:anet> set link-protection=mac-nospoof
zonecfg:ilb:anet> set mac-address=auto
zonecfg:ilb:anet> end
zonecfg:ilb> commit
zonecfg:ilb> exit
```

ロードバランサーゾーンのインストールを行います。

```
# zoneadm -z ilb install
```

ロードバランサーゾーンをブートします。

```
# zoneadm -z ilb boot
```

ロードバランサーゾーンのコンソールにログインして OS の初期設定を行います。設定方法は以下を参照してください。
http://www.fujitsu.com/jp/sparc-technical/document/solaris/index.html#solaris-zone

ホスト名は図 2 を参考にしてください。
ネットワークは net0 側がインターネット向け LAN、net1 側が DMZ 向け LAN となるように設定します。初期設定時に net0 を設定した場合は、net1 の IP インターフェースの作成から行います。net1 側のデフォルトゲートウェイはファイヤーウォールゾーンに作成する vrrp の仮想アドレスを指定します。(詳細は次回解説します)
以下は初期設定時に net0 を設定した場合のネットワーク設定例です。

```
root@ilb1:~# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           --         --
   net0/v4        static     ok           --         <インターネット LAN 側 IP アドレス>/<ネットマスク>
   net0/v6        addrconf   ok           --         fe80::214:4fff:fefa:27c9/10
root@ilb1:~# ipadm create-ip net1
root@ilb1:~# ipadm create-addr -T static -a <DMZ LAN 側 IP アドレス>/<ネットマスク> net1/v4
root@ilb1:~# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           --         --
   net0/v4        static     ok           --         <インターネット LAN 側 IP アドレス>/<ネットマスク>
   net0/v6        addrconf   ok           --         fe80::214:4fff:fefa:27c9/10
net1              ip         ok           --         --
   net1/v4        static     ok           --         <DMZ LAN 側 IP アドレス>/<ネットマスク>
root@ilb1:~# route add -inet default <ファイヤーウォールゾーンに作成する vrrp の仮想アドレス>
root@ilb1:~# route -p show
persistent: route add -inet default <インターネット LAN 側デフォルトゲートウェー>
persistent: route add -inet default <ファイヤーウォールゾーンに作成する vrrp の仮想 IP アドレス>
```

/etc/inet/hosts の設定を行います。アプリケーションサーバゾーンの IP アドレスは DMZ 向け LAN 側の IP アドレスを指定します。

必要なパッケージのインストール

ロードバランサーゾーンに以下のパッケージをインストールします。network/firewall はグローバルゾーンにもインストールしておきます。

```
root@ilb1:~# pkg install system/network/routing
root@ilb1:~# pkg install system/network/routing/vrrp
root@ilb1:~# pkg install service/network/load-balancer/ilb
root@ilb1:~# pkg install network/firewall
```

ipforward を設定します。

```
root@ilb1:~# ipadm set-prop -p forwarding=on ipv4
```

ipforward の状態を確認する。IPv4 forwarding が enabled になっていることを確認します。

```
root@ilb1:~# routeadm
             Configuration   Current             Current
                    Option   Configuration       System State
---------------------------------------------------------------
              IPv4 routing   disabled            disabled
              IPv6 routing   disabled            disabled
            IPv4 forwarding  enabled             enabled
            IPv6 forwarding  disabled            disabled

          Routing services   "route:default ripng:default"

Routing daemons:

                     STATE   FMRI
                    online   svc:/network/routing/ndp:default
                  disabled   svc:/network/routing/rdisc:default
                  disabled   svc:/network/routing/route:default
                  disabled   svc:/network/routing/ripng:default
                  disabled   svc:/network/routing/legacy-routing:ipv6
                  disabled   svc:/network/routing/legacy-routing:ipv4
```

VRRP の設定

VRRP ルーターを作成します。VRRP ルーターの構成は ILB の動作モードが NAT モードか DSR モードかによって必要となる構成が変わります。今回は DSR モードをベースに記載しています。NAT モードでも使用可能です。
Solaris の VRRP ルーターは L2 ベースの他に L3 ベースでも構築が可能です。詳しい違いはマニュアルを参照してください。
今回は IPMP での冗長化を行う場合を考慮して L3 の VRRP ルーターを使用します。
VRRP は vrrpadm コマンドを使用して作成します。以下のオプションで詳細な設定を行います。

- -V　仮想ルーター識別子(VRID)です。アドレスファミリー内に一意に指定します。
- -I　仮想 IP アドレスを設定するネットワークインタフェースを指定します。
- -p　Priority を表しており、1-255 の範囲で指定します。MASTER のルーターには 255 を指定します。その他の BACKUP ルーターは複数指定が可能ですが、MASTER の障害時には Priority に設定した値が大きいものから順番に切り替わっていきます。
- -A　inet (IPV4)または inet6(IPV6)を指定します。
- -T　-T は L2 ベースか L3 ベースかを指定します。今回は L3 ベースですので l3 を設定します。

今回は ilb1 を MASTER とし、ilb2 と ilb3 を BACKUP とする 3 台構成です。ilb1 -> ilb2 -> ilb3 の順番で切り替わります。
<ilb1 の場合>

```
root@ilb1:~# vrrpadm create-router -V 1 -I net0 -p 255 -A inet -T l3 vrrp1
```

< ilb2 の場合>

```
root@ilb2:~# vrrpadm create-router -V 1 -I net0 -p 150 -A inet -T l3 vrrp1
```

＜ilb3 の場合＞

```
root@ilb3:~# vrrpadm create-router -V 1 -I net0 -p 100 -A inet -T l3 vrrp1
```

ロードバランサーが使用する仮想 IP アドレスを VRRP ルーターに指定します。

```
root@ilb1:~# ipadm create-addr -T vrrp -n vrrp1 -a <仮想 IP アドレス>/<ネットマスク> net0/vrrp
```

IP の状態を確認します。

（例）

```
root@ilb1:~# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           --         --
   net0/v4        static     ok           --         <インターネット LAN 側 IP アドレス>/<ネットマスク>
   net0/v6        addrconf   ok           --         fe80::214:4fff:fefa:27c9/10
   net0/vrrp      vrrp       ok           --         <仮想 IP アドレス>/<ネットマスク>
net1              ip         ok           --         --
   net1/v4        static     ok           --         <DMZ LAN 側 IP アドレス>/<ネットマスク>
```

VRRP サービスを online にします。

```
root@ilb1:~# svcadm enable vrrp
root@ilb1:~# svcs vrrp
STATE          STIME    FMRI
online         15:45:52 svc:/network/vrrp:default
```

ルーターの確認をします。

<ilb1 の場合>

```
root@ilb1:~# vrrpadm show-router
NAME    VRID TYPE IFNAME   AF    PRIO ADV_INTV MODE  STATE VNIC
vrrp1   1    L3   net0     IPv4 255  1000     eopa- MASTER --
```

<ilb2 の場合>

```
root@ilb2:~# vrrpadm show-router
NAME    VRID TYPE IFNAME   AF    PRIO ADV_INTV MODE  STATE VNIC
vrrp1   1    L3   net0     IPv4 150  1000     e-pa- BACKUP --
```

<ilb3 の場合>

```
root@ilb3:~# vrrpadm show-router
NAME    VRID TYPE IFNAME   AF    PRIO ADV_INTV MODE  STATE VNIC
vrrp1   1    L3   net0     IPv4 100  1000     e-pa- BACKUP --
```

健全性検査の設定

今回は健全性検査用に「OK」という文字列を返すサーブレットを用意し、その文字列をチェックすることで健全性検査を行います。（サーブレットプログラムは次回解説します）
以下の内容で/usr/local/bin ディレクトリ配下に hcs という名前でプログラムを作成します。
健全性検査用プログラムは以下のような引数を伴って ILB によって定期的に実行されます。

- $1　　ロードバランサーが使用する仮想 IP アドレス（リテラルの IPv4 または IPv6 アドレス）
- $2　　検査するノードの IP アドレス（リテラルの IPv4 または IPv6 アドレス）
- $3　　プロトコル（文字列としての UDP、TCP）
- $4　　負荷分散モード（文字列としての DSR、NAT、HALF_NAT）
- $5　　ポート番号
- $6　　メソッドが失敗を返すまでに待機する最大時間（秒単位）

健全性検査用プログラムが成功したかどうかは、以下のいずれかを標準出力に出力することによって ILB に通知します。終了ステータスでは通知しないことに注意してください。

- マイクロ秒単位のラウンドトリップ時間(RTT)
- RTT を返さない場合で、検査が正常であった場合
- 検査が失敗した場合

```
#!/usr/bin/sh

/usr/bin/curl http://$2:$5/healthcheck/healthcheck 2>/dev/null | grep OK > /dev/null
status=$?
if [ $status -eq 0 ] ; then
  echo 0
else
  echo -1
fi
```

実行可能となるようにパーミッションを変更します。

```
root@ilb1:~# chmod 755 /usr/local/bin/hcs
```

プログラムの準備ができたら、"ilbadm create-hc"コマンドを使用して、ILB に健全性検査を設定します。この例では健全性検査名は"hc"としています。
-h オプションに続いて以下の設定を行います。

- hc-test　　PING か TCP か作成したプログラムを指定します。プログラムはフルパスで指定します。
- hc-timeout　　タイムアウト時間（デフォルト 5 秒）
- hc-count　　失敗とするまでのリトライ回数（デフォルト 3 回）
- hc-interval　　hc-test の呼び出し間隔（デフォルト 30 秒）

以下では hc-interval を 60 秒に設定しています。

```
root@ilb1:~# ilbadm create-hc -h hc-interval=60,hc-test=/usr/local/bin/hcs hc
```

以下のように健全性検査を確認します。

```
root@ilb1:~# ilbadm show-healthcheck
HCNAME        TIMEOUT COUNT   INTERVAL DEF_PING TEST
hc            5       3       60       Y        /usr/local/bin/hcs
```

ILB のルールの設定

ilb の SMF を online にします。ipv4 の forwarding を必ず先にしておいてください。

```
root@ilb1:~# svcadm enable ilb
root@ilb1:~# svcs ilb
STATE          STIME    FMRI
online         15:14:24 svc:/network/loadbalancer/ilb:default
```

ILB のルールを設定する前に、"ilbadm create-sg"コマンドを使用してサーバグループを作成します。この例ではサーバグループ名は"sg1"としています。
-s オプションのキーワード "server" にリクエストを転送するサーバとポート番号を指定します。

```
root@ilb1:~# ilbadm create-sg -s server=web1:8080,web2:8080,web3:8080 sg1
```

以下のようにサーバグループを確認します。

```
root@ilb1:~# ilbadm show-sg
SGNAME         SERVERID            MINPORT MAXPORT IP_ADDRESS
sg1            _sg1.0              8080    8080    <web1 の IP アドレス>
sg1            _sg1.1              8080    8080    <web2 の IP アドレス>
sg1            _sg1.2              8080    8080    <web3 の IP アドレス>
```

サーバグループの設定が完了したら、"ilbadm create-rule"コマンドを使用して、リクエストの転送ルールを作成します。この例ではルール名は"rule1"としています。ルールは ILB の冗長性を構成するすべてのノードで設定します。
以下は今回指定したオプションとそのキーワードの説明です。

- -e　　ルールを有効にします。
- -p　　クライアントからのリクエストを複数のセッションで同一のサーバに転送するようにします。
- -i　　受信するパケットの条件を以下のキーワードを使用して指定します。
  - vip　　VRRP の設定で使用した仮想 IP アドレス
  - port　　ポートの番号
- -m　　パケットの処理方法を以下のキーワードを使用して指定します。
  - lbalg　　分散アルゴリズムを指定します。今回はラウンドロビン(rr)を指定します。
  - type　　動作モードを指定します。今回は DSR を指定します。
  - pmask　　クライアントと転送先サーバの紐付けに使用します。今回は 32 を指定します。
- -o　　パケットの転送先を以下のキーワードを使用して指定します。
  - servergroup 転送先サーバグループ名を指定します。今回は sg1 を指定します。

- -h　　健全性検査を指定します。
  - hc-name　健全性検査名を指定します。今回は hc を指定します。
  - hc-port　健全性検査用プログラムで使用するポート番号を指定します。今回は 8080 を指定します。
- -t　　各種タイマーを設定します。
  - persist-timeout　クライアントと転送先サーバの紐付けを解除する時間を指定します。今回は 660 秒を指定しています。

```
root@ilb1:~# ilbadm create-rule -e -p -i vip=<インターネット LAN 向け仮想 IP アドレス>,port=8080 -m
lbalg=rr,¥
type=DSR,pmask=32 -h hc-name=hc,hc-port=8080 -t persist-timeout=660 -o servergroup=sg1 rule1
```

ルールを確認します。

```
root@ilb1:~# ilbadm show-rule -f rule1
       RULENAME: rule1
         STATUS: E
           PORT: 8080
       PROTOCOL: TCP
          LBALG: roundrobin
           TYPE: DSR
      PROXY-SRC: --
          PMASK: /32
        HC-NAME: hc
        HC-PORT: 8080
     CONN-DRAIN: 0
    NAT-TIMEOUT: 0
PERSIST-TIMEOUT: 660
    SERVERGROUP: sg1
            VIP: =<インターネット LAN 向け仮想 IP アドレス>
        SERVERS: _sg1.0,_sg1.1,_sg1.2
```

Packet Fileter の設定

セキュリティのためにファイヤーウォールの設定を行い、不要なポートを閉じます。
まず、SMF を Online にします。

```
root@ilb1:~# svcadm enable svc:/network/firewall:default
```

"pfconf"コマンドを使用して、ファイヤーウォールの設定します。以下の例は 8080 番と 443 番(https)のポートのみを開放した場合です。
"224.0.0.18/32"は VRRP が相互にハートビート監視を行うためのブロードキャストアドレスです。VRRP を構成するすべてのノード間でこのアドレスへの通信を許可します。

```
root@ilb1:~# pfconf
block in on net0 all
pass in on net1 all
pass out all
pass in on net0 proto tcp from any to any port = 8080
pass in on net0 proto tcp from any to any port = 443
pass in on net0 proto icmp from any to any
pass out quick on net0 from <ilb1 の IP アドレス>/32 to 224.0.0.18/32
pass in quick on net0 from <ilb1 の IP アドレス>/32 to 224.0.0.18/32
pass out quick on net0 from <ilb2 の IP アドレス>/32 to 224.0.0.18/32
pass in quick on net0 from <ilb2 の IP アドレス>/32 to 224.0.0.18/32
pass out quick on net0 from <ilb3 の IP アドレス>/32 to 224.0.0.18/32
pass in quick on net0 from <ilb3 の IP アドレス>/32 to 224.0.0.18/32
```

正しく設定が行われていることを確認します。

```
root@ilb1:~# pfctl -s rules
block drop in on net0 all
pass in on net0 proto tcp from any to any port = 8080 flags S/SA
pass in on net0 proto tcp from any to any port = 443 flags S/SA
pass in on net0 proto icmp all
pass in on net1 all flags S/SA
pass out all flags S/SA
pass out quick on net0 inet from <ilb1 の IP アドレス> to 224.0.0.18 flags S/SA
pass out quick on net0 inet from <ilb2 の IP アドレス> to 224.0.0.18 flags S/SA
pass out quick on net0 inet from <ilb3 の IP アドレス> to 224.0.0.18 flags S/SA
pass in quick on net0 inet from <ilb1 の IP アドレス> to 224.0.0.18 flags S/SA
pass in quick on net0 inet from <ilb2 の IP アドレス> to 224.0.0.18 flags S/SA
pass in quick on net0 inet from <ilb3 の IP アドレス> to 224.0.0.18 flags S/SA
```

以上でロードバランサーゾーンの作成は完了です。

次回はファイヤーウォールゾーンとアプリケーションサーバゾーンの作成を行います。