JailとVimageについて

2009/12/05
NISOC勉強会@新潟市市民活動支援センター研修室

ishimoto@ginzado.ne.jp

Jail

- FreeBSDに実装
- FreeBSDの中にFreeBSD
- FreeBSDの中にFreeBSDをいっぱい作る
- FreeBSD Jail(監獄)の中にFreeBSDの環境(Prisoner)を作る

- 各Prisoner(環境)はchrootされ分離している
- 各Prisoner(環境)はプロセスが分離されている

Vimage

- FreeBSD8.0Rより使えるようになりました

- 各々のJail環境(prisoner)で、個別のネットワークを持てるようになりました。
- (今までは、各prisonerでルーティングテーブルは持てませんでした)
- (今までは、IPアドレス体系を本体と合わせなければいけませんでした。)

- それぞれのJail環境(prisoner)がそれぞれ違うゲートウェイを向くということもOKになりました。

Prisonerを作成する (test1/test2/test3 の3台)

- [VIMAGEを活かしたkernelを作成]
- /usr/src/sys/***/conf/GENERIC を編集
  - options VIMAGE
  - nooptions SCTP #(SCTPはStream Control Transmission Protocol)
- cd /usr/src
  - make KERNCONF=GENERIC buildkernel
  - make KERNCONF=GENERIC installkernel

- [jailへ世界を作成する]
- cd /usr/src
  - make buildworld
  - make installworld DESTDIR=/usr/local/jails/test1
  - make distribution DESTDIR=/usr/local/jails/test1

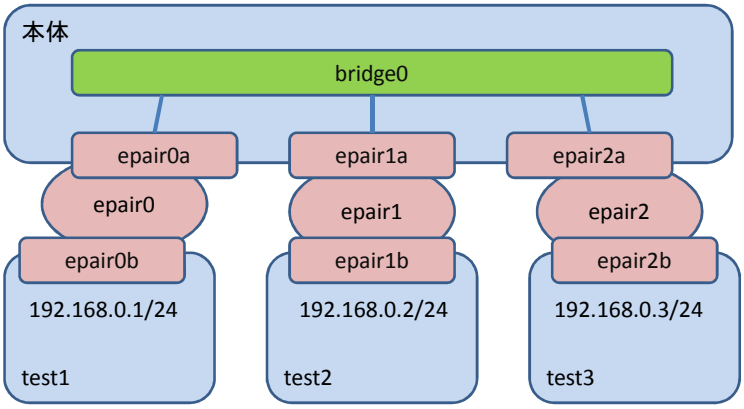
  同様にtest2,test3を作成

- [jailをvnetを使用して起動]
  - jail -c vnet host.hostname=test1.test.jp path=/usr/local/jails/test1 persist
    (persistはjailを永続させて起動する意味)

  - mount -t devfs devfs /usr/local/jails/test1/dev
  - mount -t procfs proc /usr/local/jails/test1/proc

  同様にtest2,test3を起動

テストケース1



- 本体にbridge0 I/F を作成します。
- 本体にepair0 epair1 epair2 I/F を作成します。
- 各prisonerと接続します。

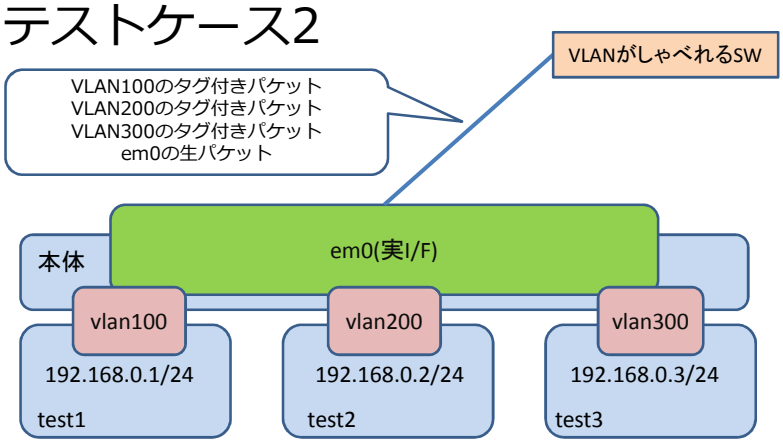
テストケース1

- [本体でbridge0を作成]
  - ifconfig bridge0 create
  (本体の中にHUBができたようなものとなります)
- [本体でepair I/Fを作成]
  - ifconfig epair0 create
  (epair0 を作成すると、epair0a , epair0b が作成され、橋渡し用のI/Fが作成されます)
- [epair I/Fとprisonerをひも付けする]
  - ifconfig epair0b vnet 1　　(vnet 1 は、test1の jail ID)
- [本体で、epair0a を bridge0 に参加させます]
  - Ifconfig bridge0 addm epair0a
  (prisoner側のepair0b の 本体側のI/Fとなる epair0a を bridge に参加させます)

  同様に、他のprisoner(test2 , test3)にも設定します。
- [prisoner(test1)に入って、epairにIPアドレスを設定します]
  - jexec 1 csh
  - % ifconfig epair0b inet 192.168.0.1/24

  同様に、他のprisoner(test2 , test3)にも設定します。

テストケース2



- 本体にVLAN100/200/300 を作成します。
- 本体のVLANを各prisonerにひも付けします。
- 実I/F(em0)は、各prisonerのVLANパケットが通ります。

テストケース2

- [本体にVLAN100,VLAN200.VLAN300を作成し、em0(実I/F)とバインドします]
  - ifconfig vlan100 create
  - ifconfig vlan200 create
  - ifconfig vlan300 create
  - ifconfig vlan100 vlan 100 vlandev em0
  - ifconfig vlan200 vlan 200 vlandev em0
  - ifconfig vlan200 vlan 200 vlandev em0

- [各VLANと各prisonerをバインドします]
  - ifconfig vlan100 vnet 1　　(vnet 1 はtest1)
  - ifconfig vlan200 vnet 2
  - ifconfig vlan300 vnet 3

  (これを実行すると、各prisoner内に各々vlan100,vlan200,vlan300 のI/Fが出現します)

- [各prisoner内に入って、vlan にIPアドレスを設定します]
  - jexec 1 csh
  - % ifconfig vlan100 inet 192.168.0.1/24
  - % ifconfig vlan100 inet6 2406:b800:beef::1/64
  - exit
  - jexec 2 csh
  - % ifconfig vlan200 inet 192.168.0.2/24
  - % ifconfig vlan200 inet6 2406:b800:beef::2/64
  - exit
  - Jexec 3 csh
  - % ifconfig vlan300 inet 192.168.0.3/24
  - % ifconfig vlan300 inet6 2406:b800:beef::3/64
  - exit

そのあとは・・・

- 各prisoner内でルーティングを個別に設定する
- 各prisoner内でルーティングデーモンを稼働させる

などが行えます。

ポイントは？

- 本体とprisonerを橋渡しする epair インターフェース
- 本体のインターフェースを直接prisonerに出現させる直接バインド
- 本体のbridge I/F は、vlan I/F とバインドできません(MTU合わず)
- IPv6は手動で設定すべき(自動取得等は安定してないようです)