

クラウド基盤構築演習

第一部

クラウド基盤を支えるインフラ技術  
～ 第2回 Linux導入・設定演習

ver1.1 2012/05/01

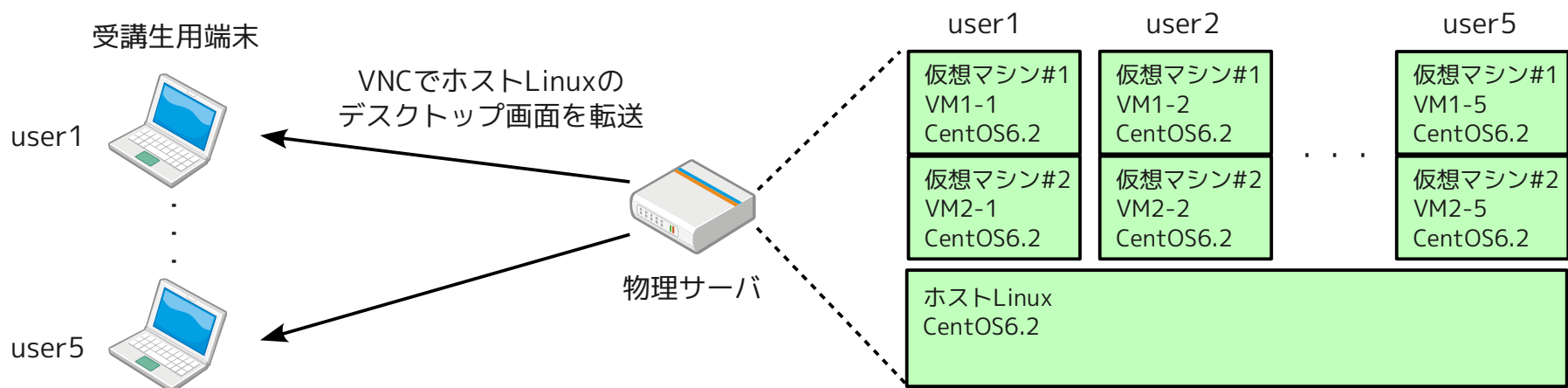
# 目次

- 演習環境の説明
- Linux導入演習
- ディスクイメージファイル利用/LVM構成演習
- iSCSI構成演習

# 演習環境の説明

# 演習環境 (1)

- 受講生（最大）5名ごとに演習用の物理サーバが割り当てられています。
  - 各受講生は、自分に割り当てられた「物理サーバ（IPアドレス）」と「ログインユーザ（user1～user5）」を確認してください。
- 各物理サーバには、ホストLinuxとして、CentOS6.2が導入されています。このホストLinuxのデスクトップ画面をVNCで受講生用端末に表示して演習を行います。
  - VNC接続の方法は、別途インストラクタよりガイドがあります。
- この演習では、Linux KVMによる仮想化環境を利用して、CentOS6.2をゲストOSとする仮想マシンを「受講生1名につき2台」作成します。
  - 各受講生は自分が作成する仮想マシンについて、「仮想マシン名、ホストネーム、IPアドレス」の割り当てルール（次ページ参照）を確認してください。

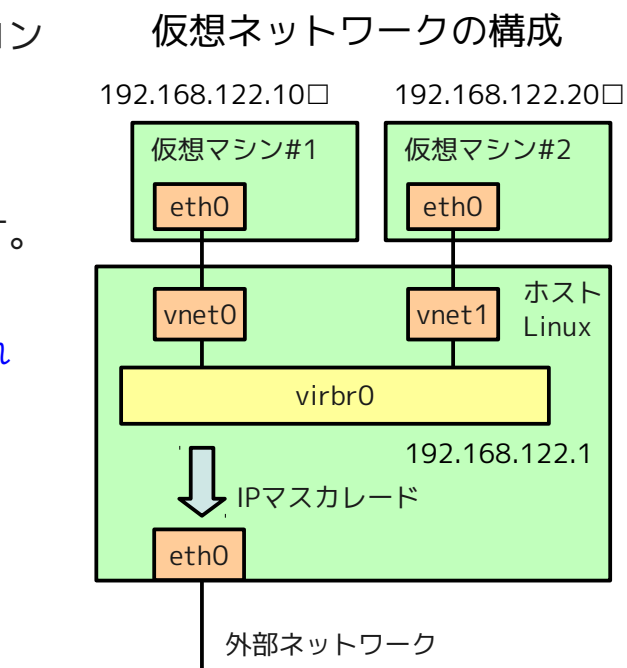


## 演習環境 (2)

- 演習で作成する仮想マシンは、ホストLinux上の仮想ブリッジによるプライベートネットワークに接続されます。
  - 仮想マシンから外部ネットワークには、IPマスカレードで接続します。外部ネットワークから仮想マシンに接続することはできません。
  - ホストLinuxから仮想マシンにログインすることは可能です。
- 仮想マシンを使用する際は、次のどちらかで接続します。
  - ホストLinuxで「virt-manager」を起動して、仮想マシンのコンソール画面を開く。
  - ホストLinuxから仮想マシンにSSHでログインする
  - 仮想マシン名、ホストネーム、IPアドレスは下表を使用します。  
□には、割り当てられたユーザ番号 (1~5) が入ります。

※演習手順において、□で示された部分も同様にユーザ番号 (1~5) を入れてください。

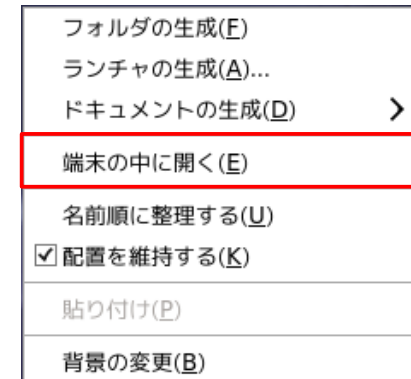
仮想マシン名	ホストネーム	IPアドレス /ネットマスク	デフォルトゲートウェイ
仮想マシン#1 VM1-□	vm1-□	192.168.122.10□ /255.255.255.0	192.168.122.1
仮想マシン#2 VM2-□	vm2-□	192.168.122.20□ /255.255.255.0	192.168.122.1



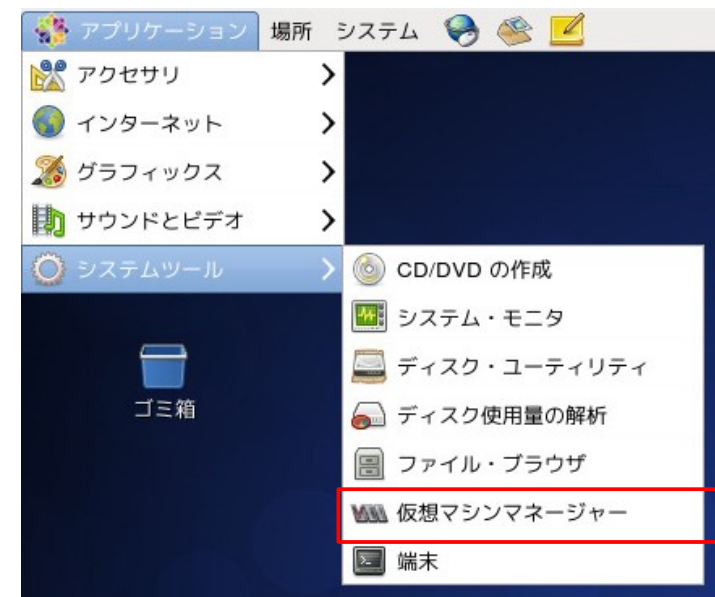
# 演習環境 (3)

- ホストLinuxでコマンド端末を開くには、デスクトップを右クリックして「端末の中を開く」を選択します。
- 「virt-manager」を起動するには、コマンド端末で「virt-manager」を実行するか、デスクトップ左上の「アプリケーション」メニューから「システムツール→仮想マシンマネージャー」を選択します。
- 「Firefox」を起動するには、コマンド端末で「firefox」を実行するか、デスクトップ上部のアイコン（「システム」メニューの右横）をクリックします。
- ホストLinux上では、CentOS6.2のインストールメディアの内容がHTTPで公開されています。ホストLinuxのFirefoxから次のURLにアクセスして、内容を確認してください。
  - <http://192.168.122.1/repo>

デスクトップの  
右クリックメニュー



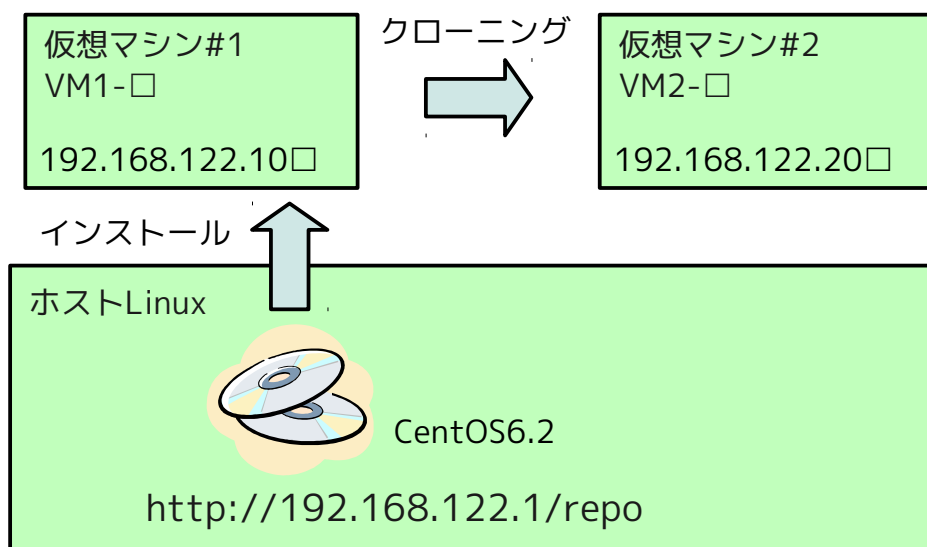
デスクトップのアプリケーションメニュー



# Linux導入演習

# 演習内容

- この演習では、次の作業を行います。
  - 仮想マシン#1 (VM1-□) を作成して、CentOS6.2を導入します。
    - 仮想マシンの構成には、virt-managerを使用します。
  - 仮想マシン#1をクローニングすることで、仮想マシン#2 (VM2-□) を作成します。
    - 仮想マシンのクローニングには、virt-managerを使用します。





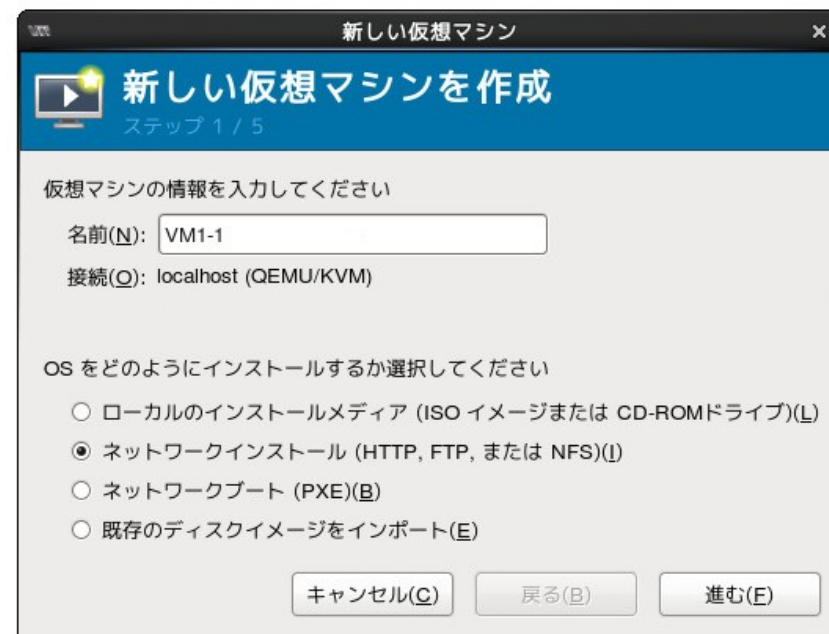
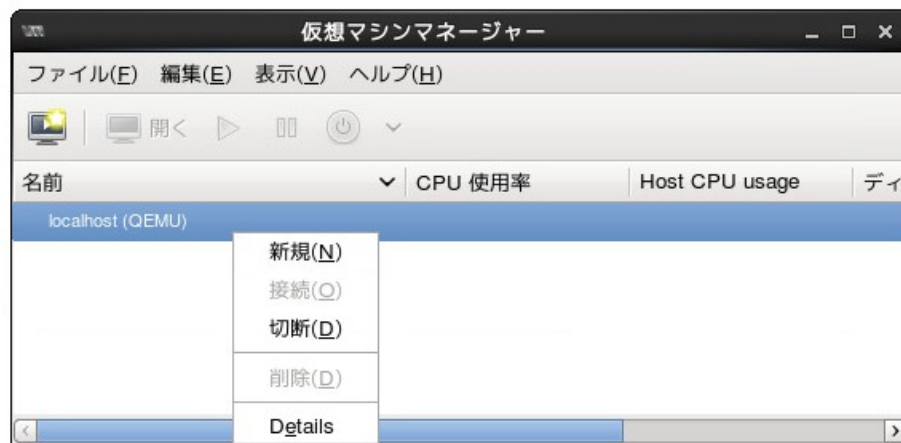
# 仮想マシン#1の作成 (1)

- virt-managerを利用して仮想マシンの作成とゲストOSの導入を行います。

- virt-managerを起動します。

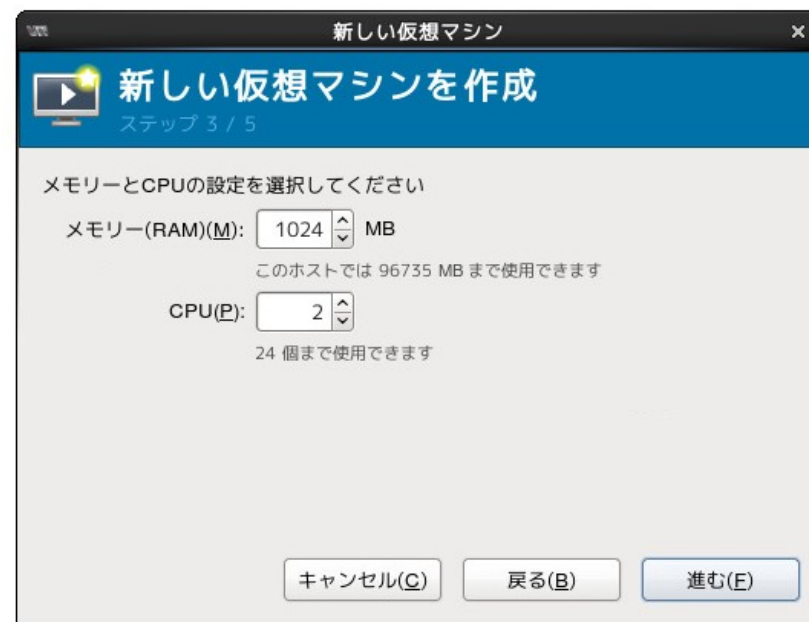
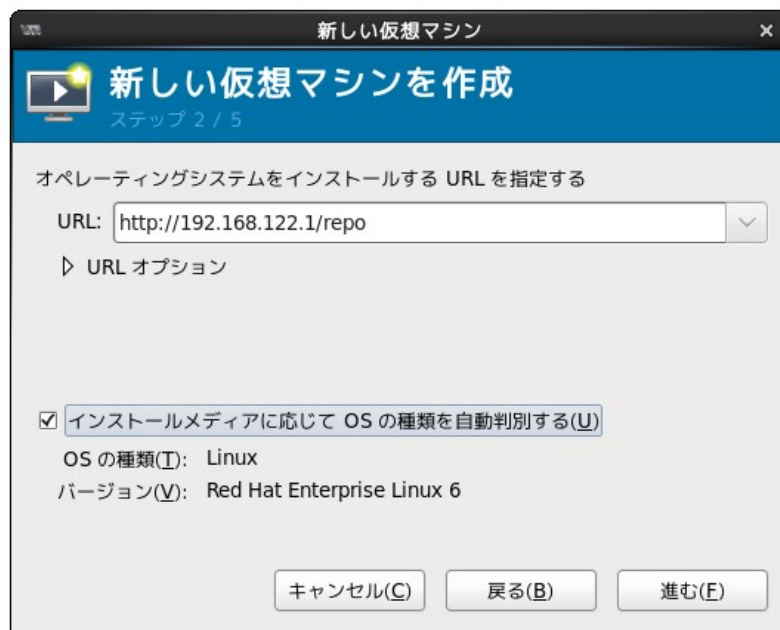
```
# virt-manager
```

- 「localhost (QEMU)」を右クリックして「新規」を選択します。
- 仮想マシン作成ウィザードが開くので、図の内容を入力、選択して次に進みます。
  - 仮想マシンの名前は「VM1-□」を入力します。インストール方法は、「ネットワークインストール」を選択します。



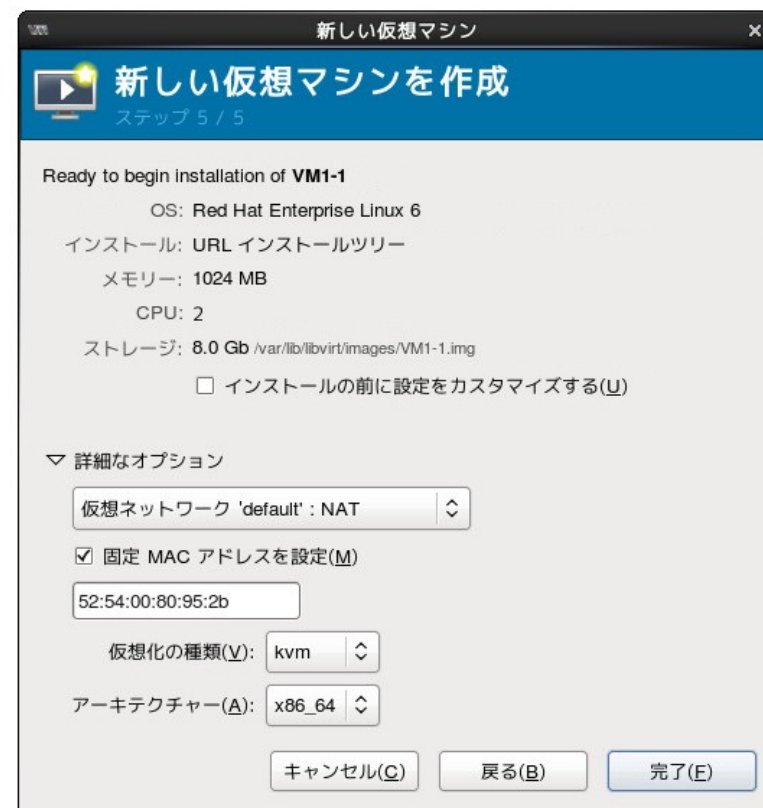
# 仮想マシン#1の作成 (2)

- URLに「http://192.168.122.1/repo」を入力します。
- 「インストールメディアに応じてOSの種類を自動判別する」のチェックボックスの選択を解除して、再度、選択します。図のように「Red Hat Enterprise Linux 6」が認識されることを確認して、次に進みます。
- メモリとCPUを図のように選択して次に進みます。



# 仮想マシン#1の作成 (3)

- ディスクイメージのサイズを図のように選択して、「今すぐディスク全体を割り当てる」をチェックして、次に進みます。
- 「詳細なオプション」で図の仮想ネットワーク（仮想ネットワーク 'default':NAT）が選択されていることを確認して、「完了」を押します。
- 仮想マシンのコンソールが開いて、インストールが始まります。これ以降は、通常のCentOS6.2のインストール手順と同様です。次ページからの手順にしたがってインストールを進めます。



# ゲストOSのインストール (1)

- 仮想マシンにCentOS 6.2をインストールします。
  - インストールに使用する言語の選択画面では、「Japanese」を選択してます。[Tab]で「OK」を選択して、[Enter]で次に進みます。
  - テキストモードでは日本語が表示できないという警告は、[Enter]で次に進みます。

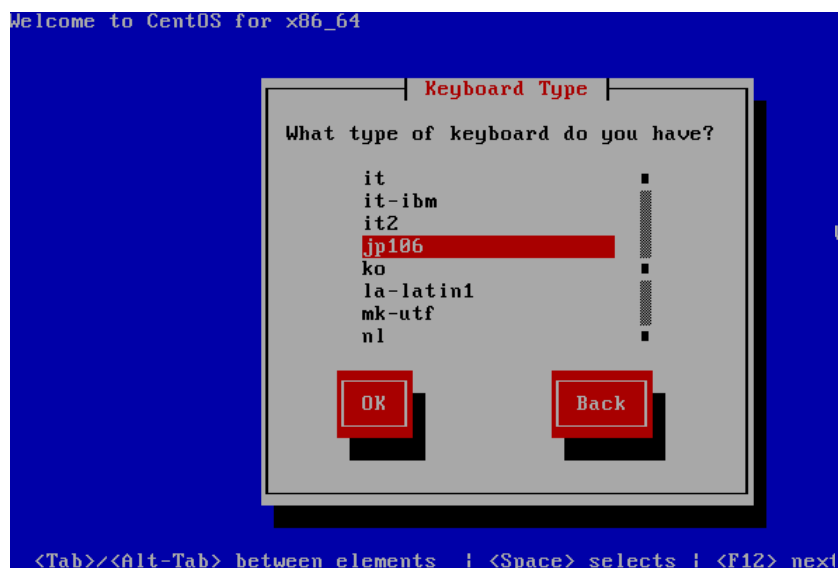


Welcome to CentOS for x86\_64

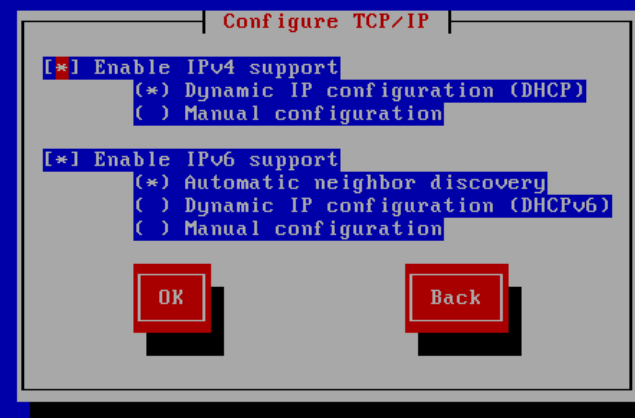


# ゲストOSのインストール (2)

- キーボードの選択画面では、「jp106」を選択してます。[Tab]で「OK」を選択して、[Enter]で次に進みます。
- TCP/IP構成画面では、[Tab]を数回押して「OK」を選択して、[Enter]で次に進みます。
- GUI画面に切り替わり、CentOS 6のロゴが表示されます。「次」を押して、次に進みます。



Welcome to CentOS for x86\_64



# ゲストOSのインストール (3)

- ストレージデバイスタイプの選択画面では、「Basic Storage Devices」が選択されていることを確認して次に進みます。
- データ消去の確認ウィンドウでは、「はい。含まれていません。どのようなデータであっても破棄してください。」を選択します。

どちらのタイプのストレージデバイスにインストールしますか?

## Basic Storage Devices

- ☒ 一般的なストレージデバイスにインストール、またはアップグレードします。どのオプションが正しいのか不明な場合は、これが適切でしょう。

## Specialized Storage Devices

- ☐ SAN (Storage Area Networks) などのエンタープライズデバイスに、インストールまたはアップグレードします。このオプションにより、FCoE / iSCSI / zFCP などのストレージデバイスを追加でき、インストーラーが無視すべきデバイスを選別します。

## ストレージデバイスの警告



以下のストレージデバイスは、データを含んでいるかもしれません。



### Virtio Block Device

8192.0 MB pci-0000:00:05.0-virtio-pci-virtio1

We could not detect partitions or filesystems on this device.

This could be because the device is **blank**, **unpartitioned**, or **virtual**. If not, there may be data on the device that can not be recovered if you use it in this installation. We can remove the device from this installation to protect the data.

Are you sure this device does not contain valuable data?

☒ Apply my choice to all devices with undetected partitions or filesystems

はい。含まれていません。どのようなデータであっても破棄してください。(Y)

いいえ。含まれています。どのようなデータであっても保護してください。(N)

# ゲストOSのインストール (4)

- ホスト名の設定画面では、「vm1-□」を入力して次に進みます。
- タイムゾーンの設定画面では、「アジア/東京」が選択されており、「システムクロックでUTCを使用」がチェックされていることを確認して次に進みます。



このコンピュータのホスト名を指定してください。ホスト名はネットワーク上でこのコンピュータを識別するために必要です。

ホスト名:

使用するタイムゾーンの中で一番近い都市を選択してください:




選択した都市: 東京, アジア

☒ システムクロックで UTC を使用 (S)



# ゲストOSのインストール (5)


- rootパスワード設定画面では、「edubase」を2回入力して次に進みます。
  - 「パスワードが弱すぎます。」という警告が表示された場合は、「とにかく使用する」を選択します。
- パーティション構成の選択画面では、「Create Custom Layout」を選択して次に進みます。

 root ユーザーはシステムの管理用に使用します。 root ユーザーのパスワードを入力してください。


root パスワード(P):


確認(C):


**パスワードが弱すぎます。**


 あなたは弱いパスワードを与えています: 辞書の単語に基づいています


どのタイプのインストールをしますか？

☐  **Use All Space**  
 選択したデバイス上のすべてのパーティションを削除します。これには、他のオペレーティングシステムで作成されたパーティションも含まれます。  
**注意:**このオプションは選択したデバイスからデータを削除します。バックアップがあることを確認してください。

☐  **Replace Existing Linux System(s)**  
 Removes only Linux partitions (created from a previous Linux installation). This does not remove other partitions you may have on your storage device(s) (such as VFAT or FAT32).  
**Tip:** This option will remove data from the selected device(s). Make sure you have backups.

☐  **Shrink Current System**  
 現在のパーティションを縮小してデフォルトレイアウト用に空き領域を作成します。

☐  **Use Free Space**  
 現在のデータとパーティションを維持して、選択したデバイスに十分な空き領域がある場合は、その未設定領域のみを使用します。

☒  **Create Custom Layout**  
 パーティション設定ツールを使用して選択したデバイス上に手動で個人設定のカスタムレイアウトを作成します。



# ゲストOSのインストール (6)

- パーティション構成画面が表示されるので、次の手順で、右表の3つのパーティションを作成します。
- 「作成」を押した後、「標準パーティション」を選択して、「作成する」を押します。

パーティション	容量	マウントポイント	タイプ
vda1	200MB	/boot	ext4
vda2	512MB	スワップ領域	swap
vda3	残り容量すべて	/	ext4



# ゲストOSのインストール (7)

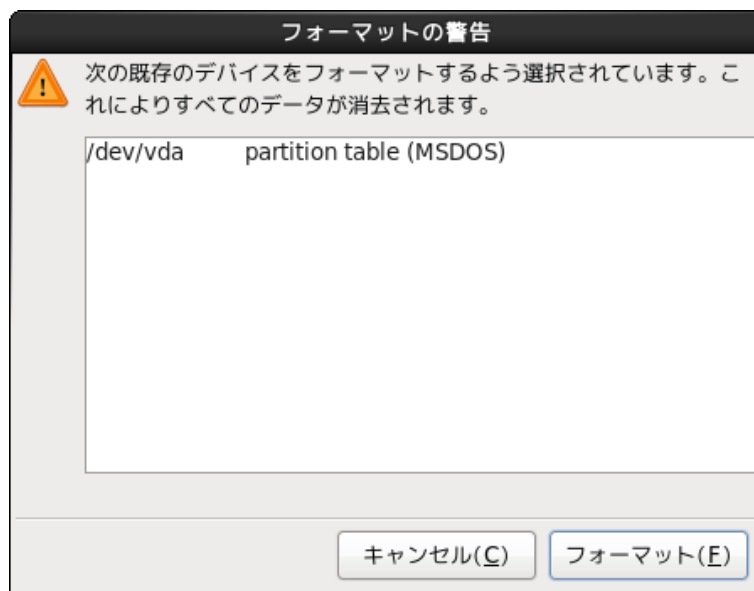
- 「パーティションの追加」ウィンドウに左下図（/bootファイルシステム）の内容を入力してOKを押します。
- 再度、「作成」を押した後、「標準パーティション」を選択して、「作成する」を押します。中下図（swap領域）の内容を入力してOKを押します。ファイルシステムタイプには「swap」を選択してください。
- 再度、同様の手順で、右下図（/ファイルシステム）の内容を入力します。「追加容量オプション」には「最大許容量まで使用」を選択してください。



# ゲストOSのインストール (8)

- 下図のパーティション構成を確認して次に進みます。「フォーマットの警告」では「フォーマット」を選択します。さらに「Writing storage configuration to disk」では「Write changes to disk」を選択します。

デバイス	容量 (MB)	マウントポイント/ RAID/ボリューム	タイプ	フォーマットする
▼ ハードディスク				
▼ vda (/dev/vda)				
vda1	200	/boot	ext4	✓
vda2	512		swap	✓
vda3	7479	/	ext4	✓



# ゲストOSのインストール (9)

- ブートローダの設定画面では、左下図の内容を確認して次に進みます。
- 導入ソフトウェアの選択画面では「Basic Server」を選択して次に進みます。
- OSのインストールが開始されますので、完了するまでしばらく待ちます。

☒ Install boot loader on /dev/vda. デバイスの変更(C)

☐ ブートローダーパスワードを使用(U) パスワードを変更(P)

ブートローダーのオペレーティングシステムのリスト

デフォルト	ラベル	デバイス
<input checked="" type="radio"/>	CentOS	/dev/vda3

CentOS のデフォルトインストールは最小限インストールです。オプションとして追加のソフトウェアを選択することができます。

- ☐ Desktop
- ☐ Minimal Desktop
- ☐ Minimal
- ☒ Basic Server
- ☐ Database Server
- ☐ Web Server
- ☐ Virtual Host
- ☐ Software Development Workstation

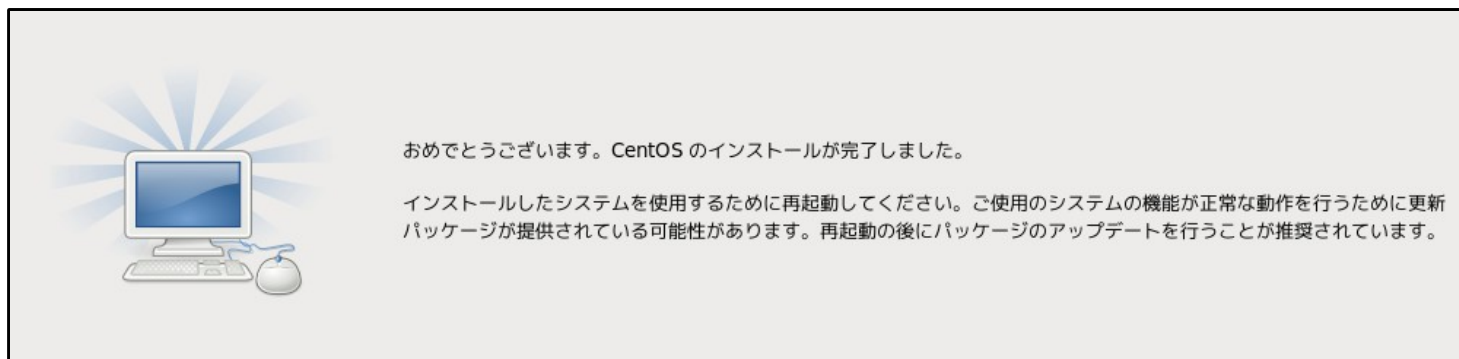
**CentOS 6**  
Community ENTERprise Operating System

完了したパッケージ: 16 個中 618 個が完了

**glibc-common-2.12-1.47.el6.x86\_64** のインストール (107 MB)  
Common binaries and locale data for glibc

# ゲストOSのインストール (10)

- インストールの完了画面が表示されたら、「再起動」を押して仮想マシンを再起動します。
- 仮想マシンの再起動後、仮想コンソールに図のログイン画面が表示されます。
- これでゲストOSのインストールは完了です。次ページからの手順にしたがって、ゲストOSの初期設定作業を行います。



```
CentOS release 6.2 (Final)
Kernel 2.6.32-220.el6.x86_64 on an x86_64

vm1-1 login: _
```

# ゲストOSの初期設定 (1)

- ゲストOSに対して、ネットワークとyumリポジトリの設定を行います。
  - virt-managerの仮想コンソールからrootユーザでログインします。(パスワードはインストール時に設定した「edubase」)
  - NICの設定ファイル「/etc/sysconfig/network-scripts/ifcfg-eth0」を次の内容に書き換えます。

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=static
NM_CONTROLLED=no
ONBOOT=yes
TYPE=Ethernet
IPADDR=192.168.122.10
NETMASK=255.255.255.0
GATEWAY=192.168.122.1
```

- iptablesを停止して、networkサービスを再起動します。

```
# chkconfig iptables off
# service iptables stop
# service network restart
```

- 192.168.122.1にpingが通ることを確認した後に、仮想コンソールからログアウトします。

```
# ping 192.168.122.1
# exit
```

- 仮想コンソールのウィンドウはここで閉じて構いません。

# ゲストOSの初期設定 (2)

- ホストLinuxのコマンド端末から、ゲストOSにSSH接続します。

```
# ssh root@192.168.122.10 ← ホストLinuxのコマンド端末で実行
```

- 「/etc/yum.repos.d」以下の既存の設定ファイルを削除します。

```
# cd /etc/yum.repos.d
# mv * /root/
```

- yumリポジトリの設定ファイル「/etc/yum.repos.d/base.repo」を次の内容で作成します。

/etc/yum.repos.d/base.repo

```
[base]
name=CetnOS6.2
baseurl=http://192.168.122.1/repo
enabled=1
gpgcheck=0
```

- yumリポジトリに接続できることを確認します。

```
# yum repolist
Loaded plugins: fastestmirror, security
base | 4.0 kB 00:00 ...
base/primary_db | 4.5 MB 00:00 ...
repo id repo name status
base CentOS6.2 6,294
repolist: 6,294
```

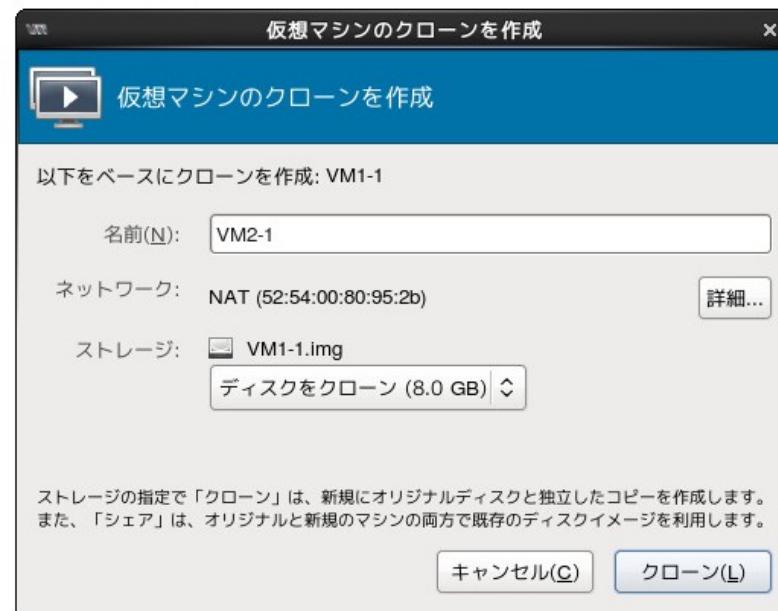
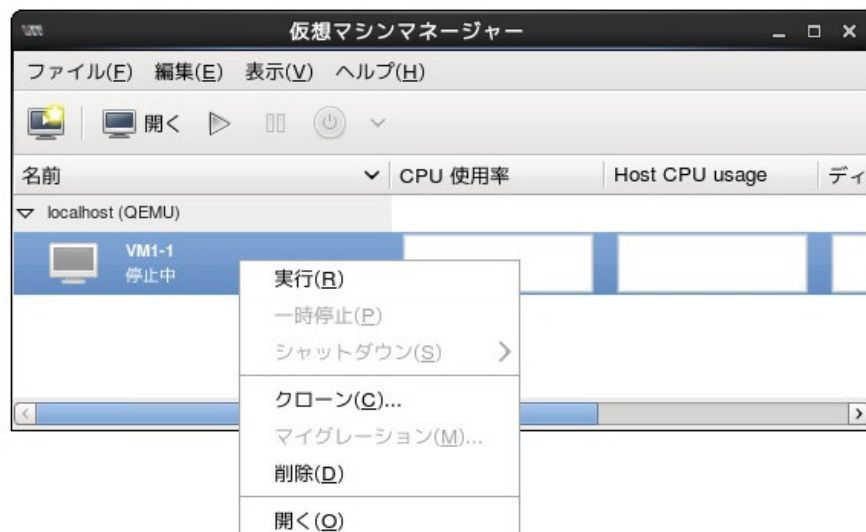
- これで「仮想マシン#1」の構成は完了です。仮想マシンのクローニングを行うために、ゲストOSを一旦、停止しておきます。

```
# poweroff
```

# 仮想マシン#2の作成 (1)

- 仮想マシンのクローニングにより「仮想マシン#2」を作成します。
  - 「仮想マシンマネージャー」のウィンドウで「VM1-□」を右クリックして、「クローン」を選択します。
  - クローン作成画面で名前に「VM2-□」を指定して、「クローン」を押します。
  - クローンの作成処理が進行するので、完了するまでしばらく待ちます。

※ 同じLinuxホストで他のユーザがクローンを行なっていると「クローンパラメータの設定中にエラーが発生しました」というメッセージが出ます。この場合は、他のユーザのクローン処理が終わってから再度、実行してください。





# 仮想マシン#2の作成 (2)

- 「仮想マシンマネージャー」のウィンドウで「VM2-□」を右クリックして、「実行」を選択します。
- 同じく「VM2-□」をダブルクリックして、仮想コンソールを開きます。ゲストOSの起動が完了するとログイン画面が表示されます。
  - ホストネームは「vm1-□」のままです。この後の手順で、ホストネームとネットワーク設定を修正します。



# 仮想マシン#2の作成 (3)

- 仮想コンソールからrootユーザでログインします。（パスワードは「edubase」）
- udev設定ファイル「/etc/udev/rules.d/70-persistent-net.rules」を次のように修正します。

この2行を削除

```
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x1af4:0x1000 (virtio-pci)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="52:54:00:c8:8a:e
a", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x1af4:0x1000 (virtio-pci)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="52:54:00:43:a0:7
7", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```

↑  
この「eth1」を「eth0」に変更

- これは、仮想NICのMACアドレスとデバイス名「ethX」の対応を指定するものです。クローン直後は、VM1-□の設定が残っているため、VM2-□の新しい仮想NICのMACアドレスが「eth1」に指定されています。これを「eth0」に変更するのが上記の修正です。
- 上記の設定ファイルにおけるMACアドレスの部分は、環境によって変わります。

# 仮想マシン#2の作成 (3)

- ネットワーク設定ファイル「/etc/sysconfig/network」の「HOSTNAME」を「vm2□」に修正します。

/etc/sysconfig/network

```
NETWORKING=yes  
HOSTNAME=vm2-□ ← ここを修正
```

- NIC設定ファイル「/etc/sysconfig/network-scripts/ifcfg-eth0」の「IPADDR」を「192.168.122.20□」に修正します。

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0  
BOOTPROTO=static  
NM_CONTROLLED=no  
ONBOOT=yes  
TYPE=Ethernet  
IPADDR=192.168.122.20□ ← ここを修正  
NETMASK=255.255.255.0  
GATEWAY=192.168.122.1
```

- ゲストOSを再起動します。

```
# reboot
```

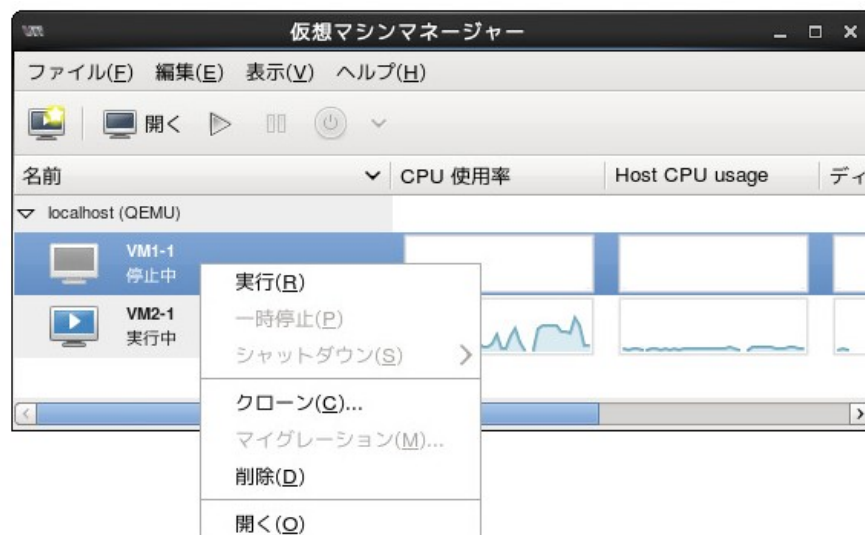
- 再起動後、ホストLinuxからゲストOSにSSHログインできることを確認します。

```
# ssh root@192.168.122.20□ ← ホストLinuxのコマンド端末で実行
```

- 仮想コンソールのウィンドウはここで閉じて構いません。

## 仮想マシン#2の作成 (3)

- 停止していた「VM1-□」を起動しておきます。「仮想マシンマネージャー」のウィンドウで「VM1-□」を右クリックして、「実行」を選択します。
- これで「仮想マシン#2」の構成は完了です。



- 以上で「Linux導入演習」は完了です。

メモとしてお使いください

[illegible]

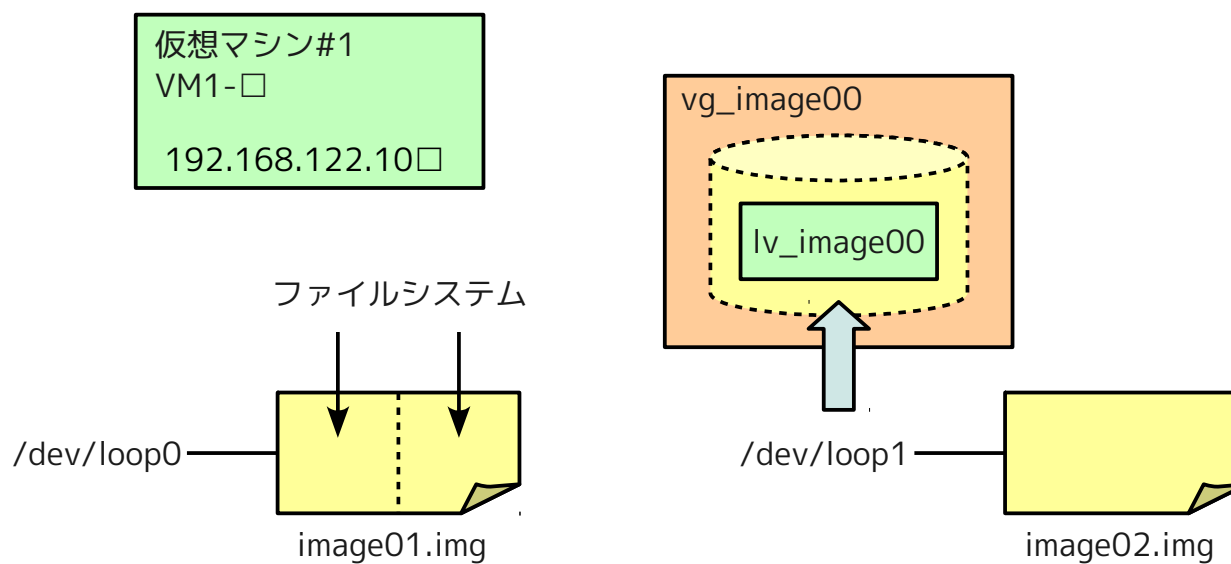
メモとしてお使いください

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# ディスクイメージファイル利用/LVM構成演習

# 演習内容

- この演習では、次の作業を行います。
  - 仮想マシン#1 (VM1-□) で、ディスクイメージファイル「image01.img」を作成して、イメージファイル内部にパーティションを作成します。作成したパーティション内にファイルシステムを作成して、VM1-□上にマウントした上でファイルを保存します。
  - 仮想マシン#1 (VM1-□) で、ディスクイメージファイル「image02.img」を作成して、これを物理ボリューム (PV) とする論理ボリューム (LV) を作成します。作成したLVにファイルシステムを作成して、VM1-□上にマウントした上でファイルを保存します。さらに、LVおよびファイルシステムの拡張を行います。





# ディスクイメージファイルの作成 (1)

- ディスクイメージファイルを作成して、内部にディスクパーティションを作成します。

- ホストLinuxのコマンド端末からVM1-□にログインします。

```
# ssh root@192.168.122.10□ ← ホストLinuxのコマンド端末で実行
```

- 200MBのディスクイメージファイル「image01.img」「image02.img」を作成します。

```
[root@vm1-□ ~]# mkdir /root/work  
[root@vm1-□ ~]# cd /root/work  
[root@vm1-□ work]# fallocate -l 200000000 image01.img  
[root@vm1-□ work]# fallocate -l 200000000 image02.img
```

- fallocateは、ext4ファイルシステム上で大容量の空ファイルを高速に作成するコマンドです。

- イメージファイルが作成されたことを確認します。

```
[root@vm1-□ work]# ls -l  
合計 390632  
-rw-r--r--. 1 root root 200000000 1月 16 18:51 2012 image01.img  
-rw-r--r--. 1 root root 200000000 1月 16 18:51 2012 image02.img
```

- 「image01.img」をループバックデバイスに紐づけます。

```
[root@vm1-□ work]# losetup -v /dev/loop0 image01.img  
Loopデバイスは /dev/loop0 です
```

- 次のコマンドで、「/dev/loop0」に紐づけられたことを確認します。

```
[root@vm1-□ work]# losetup -a  
/dev/loop0: [fc03]:263571 (/root/work/image01.img)
```

# ディスクイメージファイルの作成 (2)

- ループバックデバイス「/dev/loop0」に対してfdiskコマンドでパーティションを作成します。

```
[root@vm1-□ work]# fdisk /dev/loop0

警告: GPT (GUID パーティションテーブル) が '/dev/loop0' に検出されました! この fdisk ユーティリティは
GPT をサポートしません。GNU Parted を使ってください。
. . .

コマンド (m でヘルプ): n
コマンドアクション
  e   拡張
  p   基本パーティション (1-4)
p
パーティション番号 (1-4): 1
最初 シリンダ (1-24, 初期値 1): 1
Last シリンダ, +シリンダ数 or +size{K,M,G} (1-24, 初期値 24): 12

コマンド (m でヘルプ): n
コマンドアクション
  e   拡張
  p   基本パーティション (1-4)
p
パーティション番号 (1-4): 2
最初 シリンダ (13-24, 初期値 13): 13
Last シリンダ, +シリンダ数 or +size{K,M,G} (13-24, 初期値 24): 24

コマンド (m でヘルプ): wq
パーティションテーブルは変更されました !

. . .
```

- 省略部分「. . .」に警告メッセージが表示されますが、これらは無視して構いません。

# ディスクイメージファイルの作成 (3)

- ディスクパーティションをデバイスファイルにマッピングします。

```
[root@vm1-□ work]# kpartx -av /dev/loop0
add map loop0p1 (253:0): 0 192717 linear /dev/loop0 63
add map loop0p2 (253:1): 0 192780 linear /dev/loop0 192780
/dev/mapper/loop0p1: mknod for loop0p1 failed: File exists
```

- 最後の警告メッセージは無視して構いません。
- デバイスファイル「/dev/mapper/loop0p1」「/dev/mapper/loop0p2」が作成されていることを確認します。

```
[root@vm1-□ work]# ls -l /dev/mapper/loop0p*
lrwxrwxrwx. 1 root root 7 1月 16 19:06 2012 /dev/mapper/loop0p1 -> ../dm-0
lrwxrwxrwx. 1 root root 7 1月 16 19:06 2012 /dev/mapper/loop0p2 -> ../dm-1
```

- それぞれのパーティションにext4ファイルシステムを作成します。

```
# mkfs.ext4 /dev/mapper/loop0p1
mke2fs 1.41.12 (17-May-2010)
. . .
# mkfs.ext4 /dev/mapper/loop0p2
mke2fs 1.41.12 (17-May-2010)
. . .
```

- 1つめのパーティションを「/mnt/loop01」にマウントして、ファイルを作成します。

```
[root@vm1-□ work]# mkdir /mnt/loop01
[root@vm1-□ work]# mount /dev/mapper/loop0p1 /mnt/loop01
[root@vm1-□ work]# echo "This is image01.img" >> /mnt/loop01/file01.txt
[root@vm1-□ work]# cat /mnt/loop01/file01.txt
This is image01.img
```

- 以上で「ディスクイメージファイルの作成」は完了です。

# ディスクイメージによるLVMの構成 (1)

- ディスクイメージファイルを物理ボリューム (PV) として、論理ボリューム (LV) を作成します。

- 「image02.img」をループバックデバイスに紐づけます。

```
[root@vm1-□ work]# losetup -v /dev/loop1 image02.img
Loopデバイスは /dev/loop1 です
[root@vm1-□ work]# losetup -a
/dev/loop0: [fc03]:263571 (/root/work/image01.img)
/dev/loop1: [fc03]:263565 (/root/work/image02.img)
```

- 「image01.img」の紐づけを解除していないので、「image02.img」は「/dev/loop1」に紐づけられています。
- 「/dev/loop1」をPVとして定義した上で、これを用いて、VG (vg\_image00) と LV (lv\_image00) を作成します。

```
[root@vm1-□ work]# pvcreate /dev/loop1
Writing physical volume data to disk "/dev/loop1"
Physical volume "/dev/loop1" successfully created
[root@vm1-□ work]# vgcreate vg_image00 /dev/loop1
Volume group "vg_image00" successfully created
[root@vm1-□ work]# lvcreate -L 100M -n lv_image00 vg_image00
Logical volume "lv_image00" created
```

- ここでは、LVのサイズは100MBを指定しています。
- 作成したLVにext4ファイルシステムを作成します。

```
[root@vm1-□ work]# mkfs.ext4 /dev/vg_image00/lv_image00
mke2fs 1.41.12 (17-May-2010)
...
```

# ディスクイメージによるLVMの構成 (2)

- LV上のファイルシステムを「/mnt/loop02」にマウントしてファイルを作成します。

```
[root@vm1-□ work]# mkdir /mnt/loop02
[root@vm1-□ work]# mount /dev/vg_image00/lv_image00 /mnt/loop02
[root@vm1-□ work]# echo "This is image02.img" >> /mnt/loop02/file01.txt
[root@vm1-□ work]# cat /mnt/loop02/file01.txt
This is image02.img
```

- 現在のファイルシステムのサイズとLVの構成情報を確認します。

```
[root@vm1-□ work]# df
Filesystem            1K-ブロック  使用  使用可  使用% マウント位置
/dev/vda3              7538064   2101384    5053756    30% /
tmpfs                  510344         0    510344     0% /dev/shm
/dev/vda1             198337     27747    160350    15% /boot
/dev/mapper/loop0p1    93307      5648     82842     7% /mnt/loop01
/dev/mapper/vg_image00-lv_image00
                      99150      5648     88382     7% /mnt/loop02

# lvsdisplay
--- Logical volume ---
LV Name                /dev/vg_image00/lv_image00
VG Name                vg_image00
LV UUID                0afzXu-vfQL-gcZN-7nng-tt1U-qD0G-fNsmla
LV Write Access        read/write
LV Status              available
# open                 1
LV Size                100.00 MiB
Current LE             25
Segments              1
Allocation              inherit
Read ahead sectors    auto
- currently set to    256
Block device          253:0
```

- pvdisplayコマンド、vgdisplayコマンドの出力も確認してみてください。

# ディスクイメージによるLVMの構成 (3)

- LVとファイルシステムのサイズを拡張します。

```
[root@vm1-□ work]# lvextend -L+50MB /dev/vg_image00/lv_image00
Rounding up size to full physical extent 52.00 MiB
Extending logical volume lv_image00 to 152.00 MiB
Logical volume lv_image00 successfully resized

[root@vm1-□ work]# resize2fs /dev/vg_image00/lv_image00
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/vg_image00/lv_image00 is mounted on /mnt/loop02; on-line resizing
required
old desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/vg_image00/lv_image00 to 155648 (1k) blocks.
The filesystem on /dev/vg_image00/lv_image00 is now 155648 blocks long.
```

- lvextendコマンドでLVのサイズを50MB拡張しています。次のresize2fsコマンドでは、それに合わせて、LV上のファイルシステムのサイズを拡張しています。

# ディスクイメージによるLVMの構成 (4)

- LVとファイルシステムのサイズが拡張されたことを確認します。

```
[root@vm1-□ work]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg_image00/lv_image00
VG Name                vg_image00
LV UUID                0afzXu-vfQL-gcZN-7nng-tt1U-qD0G-fNsm1a
LV Write Access        read/write
LV Status              available
# open                 1
LV Size                152.00 MiB
Current LE             38
Segments              1
Allocation             inherit
Read ahead sectors    auto
- currently set to    256
Block device          253:0

[root@vm1-□ work]# df
Filesystem            1K-ブロック  使用  使用可  使用%  マウント位置
/dev/vda3              7538064  2101392   5053748   30% /
tmpfs                  510344      0   510344    0% /dev/shm
/dev/vda1             198337    27747   160350   15% /boot
/dev/mapper/loop0p1    93307     5648    82842    7% /mnt/loop01
/dev/mapper/vg_image00-lv_image00
                      150904     5648   137682    4% /mnt/loop02
```

- 以上で「ディスクイメージによるLVMの構成」は完了です。次ページからの手順で「後片付け」を行います。

# ループバックデバイスの解放

- 使用したファイルシステムをアンマウントして、ループバックデバイスへの紐づけを解除します。

- ファイルシステムをアンマウントします。

```
[root@vm1-□ work]# umount /mnt/loop01  
[root@vm1-□ work]# umount /mnt/loop02
```

- VGを非アクティブ化して、「/dev/loop1」への紐づけを解除します。

```
[root@vm1-□ work]# vgchange -an  
  0 logical volume(s) in volume group "vg_image00" now active  
[root@vm1-□ work]# losetup -d /dev/loop1
```

- 同じく、「/dev/loop0」のパーティションマッピングを解除して、紐づけを解除します。

```
[root@vm1-□ work]# kpartx -d /dev/loop0  
[root@vm1-□ work]# losetup -d /dev/loop0
```

- ループバックデバイスに対するすべての紐づけが解除されていることを確認します。

```
[root@vm1-□ work]# losetup -a
```

- 以上で「ループバックデバイスの解放」は完了です。

- 以上で「ディスクイメージファイル利用/LVM構成演習」は完了です。

- ここで作成したディスクイメージファイル「image01.img」「image02.img」は次の演習で使用します。削除せずに残しておいてください。



メモとしてお使いください

[illegible]

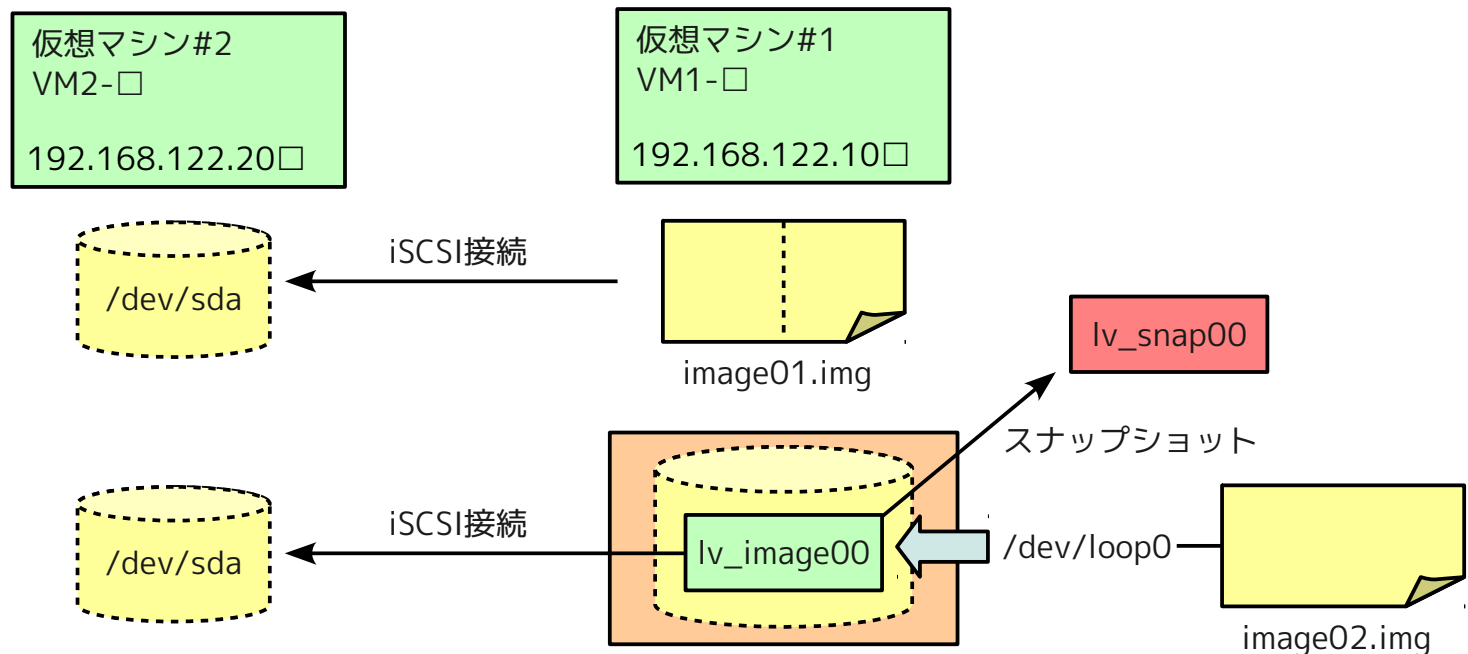
メモとしてお使いください

[illegible]

# iSCSI構成演習

# 演習内容

- この演習では、次の作業を行います。
  - 仮想マシン#1 (VM1-□) のディスクイメージファイル「image01.img」をiSCSIターゲットとして公開します。これを仮想マシン#2 (VM2-□) にiSCSIディスクとして接続して、使用します。
  - 仮想マシン#1 (VM1-□) のディスクイメージファイル「image02.img」から作成した論理ボリューム (LV) 「lv\_image00」をiSCSIターゲットとして公開します。これを仮想マシン#2 (VM2-□) にiSCSIディスクとして接続して、使用します。
  - 仮想マシン#1 (VM1-□) で「lv\_image00」のスナップショット「lv\_snap00」を作成します。



# iSCSIターゲットの構成 (1)

- VM1-□でディスクイメージファイルをiSCSIターゲットとして公開します。

- ホストLinuxのコマンド端末からVM1-□にログインします。

```
# ssh root@192.168.122.10□ ← ホストLinuxのコマンド端末で実行
```

- iSCSIターゲットを構成するために必要なパッケージを導入します。

```
[root@vm1-□ ~]# yum install scsi-target-utils
. . .
=====
Package                        Arch      Version      Repository    Size
=====
Installing:
scsi-target-utils             x86_64     1.0.14-4.el6      base         162 k
Installing for dependencies:
libibverbs                    x86_64     1.1.5-3.el6       base          44 k
librdmacm                     x86_64     1.0.14.1-3.el6    base          27 k
perl-Config-General           noarch     2.44-1.el6        base          68 k

Transaction Summary
=====
Install      4 Package(s)

Total download size: 300 k
Installed size: 735 k
Is this ok [y/N]: y
. . .
Complete!
```

- 導入パッケージの確認には「y [Enter]」を入力します。

## iSCSIターゲットの構成 (2)

- 先の演習で作成したイメージファイル「imgae01.img」をiSCSIターゲットのバックエンドファイル用ディレクトリに移動します。

```
[root@vm1-□ ~]# mkdir /var/lib/tgtd  
[root@vm1-□ ~]# restorecon /var/lib/tgtd  
[root@vm1-□ ~]# cp /root/work/image01.img /var/lib/tgtd/
```

- iSCSIターゲットのデーモン (tgtd) はSELinuxでセキュリティ保護されているため、適切なセキュリティコンテキストが設定されたイメージファイルのみを公開することができます。restoreconコマンドは、ディレクトリ「/var/lib/tgtd」に対して、「tgtdが公開可能なファイルを保存する」ためのセキュリティコンテキストを設定します。
- tgtdの設定ファイル「/etc/tgt/targets.conf」の末尾に次の内容を追加します。

/etc/tgt/targets.conf

```
<target iqn.2012-07.com.example.vm1□:tgt01>  
    backing-store /var/lib/tgtd/image01.img  
</target>
```

- tgtdサービスを開始します。

```
[root@vm1-□ ~]# chkconfig tgtd on  
[root@vm1-□ ~]# service tgtd start  
SCSI target daemon を起動中:
```

[ OK ]

# iSCSIターゲットの構成 (3)

- ディスクイメージファイル「/var/lib/tgtd/image01.img」がiSCSIターゲットとして公開されていることを確認します。

```
[root@vm1-□ ~]# tgt-admin -s
Target 1: iqn.2012-07.com.example.vm1□:tgt01
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
  LUN information:
    LUN: 0
      Type: controller
      SCSI ID: IET      00010000
      SCSI SN: beaf10
      Size: 0 MB, Block size: 1
      Online: Yes
      Removable media: No
      Readonly: No
      Backing store type: null
      Backing store path: None
      Backing store flags:
    LUN: 1
      Type: disk
      SCSI ID: IET      00010001
      SCSI SN: beaf11
      Size: 200 MB, Block size: 512
      Online: Yes
      Removable media: No
      Readonly: No
      Backing store type: rdwr
      Backing store path: /var/lib/tgtd/image01.img
      Backing store flags:
  Account information:
  ACL information:
    ALL
```

- 以上で「iSCSIターゲットの構成」は完了です。

# iSCSIディスクの接続 (1)

- VM1-□で公開したiSCSIターゲットをVM2-□に接続して使用します。

- ホストLinuxのコマンド端末からVM2-□にログインします。

```
# ssh root@192.168.122.20□ ← ホストLinuxのコマンド端末で実行
```

- iSCSIイニシエータに必要なパッケージを導入します。

```
[root@vm2-□ ~]# yum install iscsi-initiator-utils
```

```
...
```

Package	Arch	Version	Repository	Size
Installing:				
iscsi-initiator-utils	x86_64	6.2.0.872-34.el6	base	614 k

```
Transaction Summary
```

```
=====
Install      1 Package(s)
```

```
Total download size: 614 k
```

```
Installed size: 2.2 M
```

```
Is this ok [y/N]: y
```

```
...
```

```
Complete!
```

- 導入パッケージの確認には「y [Enter]」を入力します。



# iSCSIディスクの接続 (2)

- VM1-□のiSCSIターゲットの存在をVM2-□に登録します。

```
[root@vm2-□ ~]# iscsiadm -m discovery --type sendtargets --portal 192.168.122.10□
iscsid を起動中: [ OK ]
192.168.122.10□:3260,1 iqn.2012-07.com.example.vm1□:tgt01
```

- iscsiサービスを起動して、登録したiSCSIターゲットに接続します。

```
[root@vm2-□ ~]# chkconfig iscsi on
[root@vm2-□ ~]# service iscsi start
iscsi を起動中: [ OK ]
```

- 「iscsid」サービスではなく、「iscsi」サービスなので間違えないように注意してください。

- iSCSIディスクが「/dev/sda」として認識されていることを確認します。

```
[root@vm2-□ ~]# fdisk -l /dev/sda

ディスク /dev/sda: 200 MB, 200000000 バイト
ヘッド 255, セクタ 63, シリンダ 24
Units = シリンダ数 of 16065 * 512 = 8225280 バイト
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O size (minimum/optimal): 512 bytes / 512 bytes
ディスク識別子: 0x94eaa5c5
```

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sda1		1	12	96358+	83	Linux
/dev/sda2		13	24	96390	83	Linux

- 先の演習でイメージファイルに作成したパーティションが認識されていることが分かります。

# iSCSIディスクの接続 (3)

- 1つめのパーティションのファイルシステムをマウントして内容を確認します。

```
[root@vm2-□ ~]# mkdir /mnt/iscsi01
[root@vm2-□ ~]# mount /dev/sda1 /mnt/iscsi01
[root@vm2-□ ~]# ls -l /mnt/iscsi01/
合計 14
-rw-r--r--. 1 root root    20  1月 16 22:35 2012 file01.txt
drwx-----. 2 root root 12288  1月 16 22:34 2012 lost+found
[root@vm2-□ ~]# cat /mnt/iscsi01/file01.txt
This is image01.img
```

- 先の演習で作成したファイルが残っていることが分かります。

- 一度、ファイルシステムをアンマウントして、iSCSIターゲットの接続を解除します。

```
[root@vm2-□ ~]# umount /mnt/iscsi01
[root@vm2-□ ~]# service iscsi stop
iscsi を停止中: [ OK ]
[root@vm2-□ ~]# fdisk -l /dev/sda
```

- 最後のfdiskコマンドで何も表示されないことを確認します。

- 以上で「iSCSIディスクの接続」は完了です。

# 論理ボリュームによるiSCSIターゲットの構成 (1)

- VM1-□で論理ボリュームをiSCSIターゲットとして公開します。

- ホストLinuxのコマンド端末からVM1-□にログインします。

```
# ssh root@192.168.122.10□ ← ホストLinuxのコマンド端末で実行
```

- 先の演習でPVとして構成したイメージファイル「image02.img」をループバックデバイスに紐づけます。

```
[root@vm1-□ ~]# losetup -v /dev/loop0 /root/work/image02.img  
Loopデバイス は /dev/loop0 です
```

- PVを再認識して、VGをアクティブ化します。

```
[root@vm1-□ ~]# pvscan  
PV /dev/loop0 VG vg_image00 lvm2 [188.00 MiB / 36.00 MiB free]  
Total: 1 [188.00 MiB] / in use: 1 [188.00 MiB] / in no VG: 0 [0 ]  
[root@vm1-□ ~]# vgchange -ay  
1 logical volume(s) in volume group "vg_image00" now active
```

# 論理ボリュームによるiSCSIターゲットの構成 (2)

- VGに含まれるLVが有効化されていることを確認します。

```
[root@vm1-□ ~]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg_image00/lv_image00
VG Name                 vg_image00
LV UUID                 0afzXu-vfQL-gcZN-7nng-tt1U-qD0G-fNsm1a
LV Write Access         read/write
LV Status                available
# open                  0
LV Size                 152.00 MiB
Current LE              38
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:0
```

- 「LV status」が「available」であることを確認します。また、LVのサイズが約150MBであることを覚えておきます。
- tgttdの設定ファイル「/etc/tgt/targets.conf」の末尾を次のように修正します。

/etc/tgt/targets.conf

```
<target iqn.2012-07.com.example.VM1-□:tgt01>
  backing-store /dev/vg_image00/lv_image00 ← この行を修正
</target>
```

- tgttdサービスを再起動して、設定変更を反映します。

```
[root@vm1-□ ~]# service tgttd restart
SCSI target daemon を停止中:      [ OK ]
SCSI target daemon を起動中:      [ OK ]
```

# 論理ボリュームによるiSCSIターゲットの構成 (3)

- 論理ボリューム「/dev/vg\_image00/lv\_image00」がiSCSIターゲットとして公開されていることを確認します。

```
[root@vm1-□ ~]# tgt-admin -s
Target 1: iqn.2012-07.com.example.VM1-□:tgt01
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
  LUN information:
    LUN: 0
      Type: controller
      SCSI ID: IET      00010000
      SCSI SN: beaf10
      Size: 0 MB, Block size: 1
      Online: Yes
      Removable media: No
      Readonly: No
      Backing store type: null
      Backing store path: None
      Backing store flags:
    LUN: 1
      Type: disk
      SCSI ID: IET      00010001
      SCSI SN: beaf11
      Size: 159 MB, Block size: 512
      Online: Yes
      Removable media: No
      Readonly: No
      Backing store type: rdwr
      Backing store path: /dev/vg_image00/lv_image00
      Backing store flags:
  Account information:
  ACL information:
    ALL
```

- 以上で「論理ボリュームによるiSCSIターゲットの構成」は完了です。

# 論理ボリュームのスナップショット作成 (1)

- VM1-□で公開したLVによるiSCSIターゲットをVM2-□に接続した状態で、VM1-□上でLVのスナップショットを作成します。

- ホストLinuxのコマンド端末からVM2-□にログインします。

```
# ssh root@192.168.122.20□ ← ホストLinuxのコマンド端末で実行
```

- iSCSIサービスを起動して、iSCSIターゲットに接続します。

```
[root@vm2-□ ~]# service iscsi start  
iscsi を起動中: [ OK ]
```

- iSCSIディスクが「/dev/sda」として認識されていることを確認します。

```
[root@vm2-□ ~]# fdisk -l /dev/sda  
  
ディスク /dev/sda: 159 MB, 159383552 バイト  
ヘッド 5, セクタ 61, シリンダ 1020  
Units = シリンダ数 of 305 * 512 = 156160 バイト  
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
ディスク識別子: 0x00000000  
  
ディスク /dev/sda は正常なパーティションテーブルを含んでいません
```

- ディスク容量が先に確認したLVのサイズとほぼ等しいことを確認します。

# 論理ボリュームのスナップショット作成 (2)

- ISCSIディスクのファイルシステムをマウントして内容を確認します。

```
[root@vm2-□ ~]# mount /dev/sda /mnt/iscsi01
[root@vm2-□ ~]# ls -l /mnt/iscsi01/
合計 14
-rw-r--r--. 1 root root    20  1月 16 23:57 2012 file01.txt
drwx-----. 2 root root 12288  1月 16 23:56 2012 lost+found
[root@vm2-□ ~]# cat /mnt/iscsi01/file01.txt
This is image02.img
```

- 先の演習で作成したファイルが残っていることが分かります。

- さらに新しいファイルを作成して、一旦、ファイルシステムをアンマウントします。

```
[root@vm2-□ ~]# echo "This is written by VM2-□" > /mnt/iscsi01/file02.txt
[root@vm2-□ ~]# cat /mnt/iscsi01/file02.txt
This is written by VM2-□
[root@vm2-□ ~]# umount /mnt/iscsi01
```

- 続いて、VM1-□側でLVのスナップショットを作成します。
- ホストLinuxのコマンド端末からVM1-□にログインします。

```
# ssh root@192.168.122.10□ ← ホストLinuxのコマンド端末で実行
```

- 論理ボリューム「/dev/vg\_image00/lv\_image00」からスナップショット・論理ボリューム「lv\_snap00」を作成します。

```
[root@vm1-□ ~]# lvcreate -s -L 10M -n lv_snap00 /dev/vg_image00/lv_image00
Rounding up size to full physical extent 12.00 MiB
Logical volume "lv_snap00" created
```

- ここでは、差分の保存領域に10MBの容量を確保しています。

# 論理ボリュームのスナップショット作成 (3)

- スナップショット論理ボリュームが作成されたことを確認します。

```
[root@vm1-□ ~]# lvdisplay
...
--- Logical volume ---
LV Name                /dev/vg_image00/lv_snap00
VG Name                vg_image00
LV UUID                7NxsS9-lnBo-whZa-10gd-ZcJK-e0Rh-S02nx1
LV Write Access        read/write
LV snapshot status      active destination for /dev/vg_image00/lv_image00
LV Status               available
# open                 0
LV Size                152.00 MiB
Current LE              38
COW-table size          12.00 MiB
COW-table LE            3
Allocated to snapshot   0.00%
Snapshot chunk size     4.00 KiB
Segments                1
Allocation              inherit
Read ahead sectors      auto
 - currently set to     256
Block device            253:1
```

- スナップショットに含まれるファイルシステムをマウントして内容を確認します。

```
[root@vm1-□ ~]# mkdir /mnt/snap00
[root@vm1-□ ~]# mount /dev/vg_image00/lv_snap00 /mnt/snap00/
[root@vm1-□ ~]# ls -l /mnt/snap00/
合計 16
-rw-r--r--. 1 root root    20  1月 16 23:57 2012 file01.txt
-rw-r--r--. 1 root root    24  1月 17 16:30 2012 file02.txt
drwx-----. 2 root root 12288  1月 16 23:56 2012 lost+found
[root@vm1-□ ~]# cat /mnt/snap00/file02.txt
This is written by VM2-□
```



# 論理ボリュームのスナップショット作成 (4)

- この後は、VM2-□で再度、ファイルシステムをマウントして使用を継続することができます。

```
[root@vm2-□ ~]# mount /dev/sda /mnt/iscsi01
[root@vm2-□ ~]# ls -l /mnt/iscsi01/
合計 16
-rw-r--r--. 1 root root    20  1月 16 23:57 2012 file01.txt
-rw-r--r--. 1 root root    24  1月 17 16:30 2012 file02.txt
drwx-----. 2 root root 12288  1月 16 23:56 2012 lost+found
[root@vm2-□ ~]# echo "This is yet another file from VM2-□" >> /mnt/iscsi01/file03.txt
```

- VM2-□での書き込みは、スナップショットの内容には影響しないので、VM1-□側では独立して、スナップショットからファイルをバックアップするなどの作業が可能です。
- 以上で「論理ボリュームのスナップショット作成」は終了です。次ページからの手順で「後片付け」を行います。

# iSCSIディスク接続の解除

- VM2-□のiSCSIディスク接続を解放します。

- VM2-□でファイルシステムをアンマウントします。

```
[root@vm2-□ ~]# umount /mnt/iscsi01
```

- VM2-□が認識しているiSCSIターゲットの情報を確認します。

```
[root@vm2-□ ~]# iscsiadm -m node  
192.168.122.10□:3260,1 iqn.2012-07.com.example.VM1-□:tgt01
```

- 確認した情報を元にして、次のコマンドでiSCSIターゲットの認識を削除します。

```
[root@vm2-□ ~]# iscsiadm -m node -T iqn.2012-07.com.example.VM1-□:tgt01 -p 192.168.122.10□ --logout  
Logging out of session [sid: 5, target: iqn.2012-07.com.example.VM1-□:tgt01, portal: 192.168.122.10□,3260]  
Logout of [sid: 5, target: iqn.2012-07.com.example.VM1-□:tgt01, portal: 192.168.122.10□,3260] successful.
```

```
[root@vm2-□ ~]# iscsiadm -m node -o delete -T iqn.2012-07.com.example.VM1-□:tgt01 -p 192.168.122.10□
```

- 1つめのコマンドは、iSCSIターゲットへの接続を解除しています。2つめのコマンドは、iSCSIターゲットの認識を削除しています。

- VM2-□から認識しているiSCSIターゲットがなくなっていることを確認します。

```
[root@vm2-□ ~]# iscsiadm -m node  
iscsiadm: No records found
```

- 以上で「iSCSIディスク接続の解除」は完了です。

# iSCSIターゲットの公開停止

- VM1-□のスナップショットを削除して、iSCSIターゲットの公開停止、ループバックデバイスの紐づけ解除などを行います。

- VM1-□でスナップショットのファイルシステムをアンマウントして、スナップショットを削除します。

```
[root@vm1-□ ~]# umount /mnt/snap00  
[root@vm1-□ ~]# lvremove /dev/vg_image00/lv_snap00  
Do you really want to remove active logical volume lv_snap00? [y/n]: y  
Logical volume "lv_snap00" successfully removed
```

- tgtサービスを停止して、iSCSIターゲットの公開を停止します。

```
[root@vm1-□ ~]# service tgt stop  
SCSI target daemon を停止中: [ OK ]
```

- VGを非アクティブ化して、ループバックデバイスへの紐づけを解除します。

```
[root@vm1-□ ~]# vgchange -an  
0 logical volume(s) in volume group "vg_image00" now active  
[root@vm1-□ ~]# losetup -d /dev/loop0
```

- 以上で「iSCSIターゲットの公開停止」は完了です。

- 以上で「iSCSI構成演習」は完了です。

メモとしてお使いください

[illegible]

メモとしてお使いください

[illegible]

メモとしてお使いください

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# Top SE

---

EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS

