

# KVM仮想化技術の要素

KVM(Kernel-based Virtual Machine)  
の技術の主要要素に、QEMU があります。

QEMU(キューエミュ) 仮想マシンエミュレータ  
古くからあるハードウェアエミュレーション技術だが、新しく CPU の部分の仮想マシンモニター機能をシンプルな仕組みで、linux カーネルに組み込んだ。

Qcow2(Qemu copy on write)  
仮想マシンのイメージを保存するファイル形式で、保存サイズも実サイズより小さく、スナップショットも取れる。

LVM(Logical Volume Manager)

- 物理的な Disk の存在を隠蔽して作る、論理ボリューム
- 物理的な Disk の増設や変更を隠蔽できることで、Disk デバイス管理の柔軟性が高まる。
- 仮想マシンのイメージの管理・保存に利用される。

NFS -> Gluster

NFS → ZFS

VNC

仮想マシンを表示する際に使用する通信プロトコル。

仮想マシンとのコミュニケーション --> 次世代メッセージプロトコル

# KVMとXen

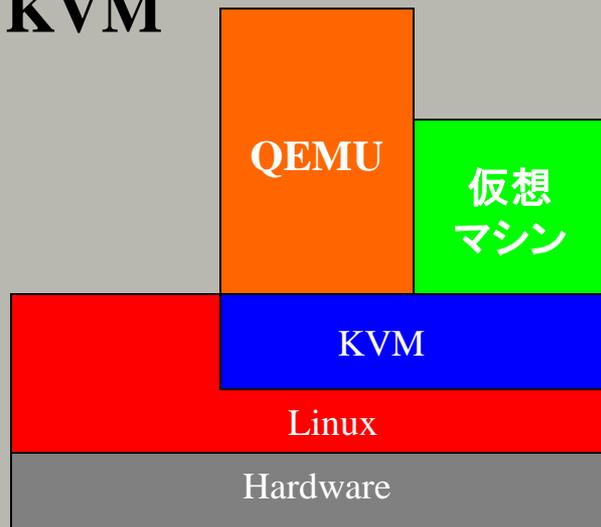
## KVMの履歴

2006年10月公開。  
直ぐにLinuxカーネルにマージ  
イスラエルのQumranetが開発  
2008年12月RedHatが買収  
IntelVT/AMD-Vが必須

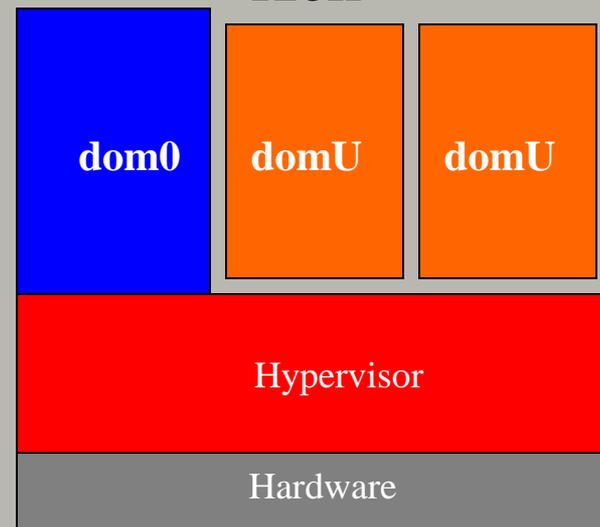
## Xenの履歴

2003年10月公開。  
ケンブリッジ大学の研究プロジェクト  
から、XenSource社へ。  
2007年 citrix が XenSourceを買収  
全仮想化も、準仮想化もサポート

## KVM



## Xen



非常に軽量  
Linuxカーネルの一部  
OS,カーネルのバージョンアップの際  
メンテナンス容易

Linuxカーネルにパッチを当てて、  
ハイパーバイザにしている。  
OS,カーネルのバージョンアップの際  
メンテナンス困難

# クラウド導入にあたって

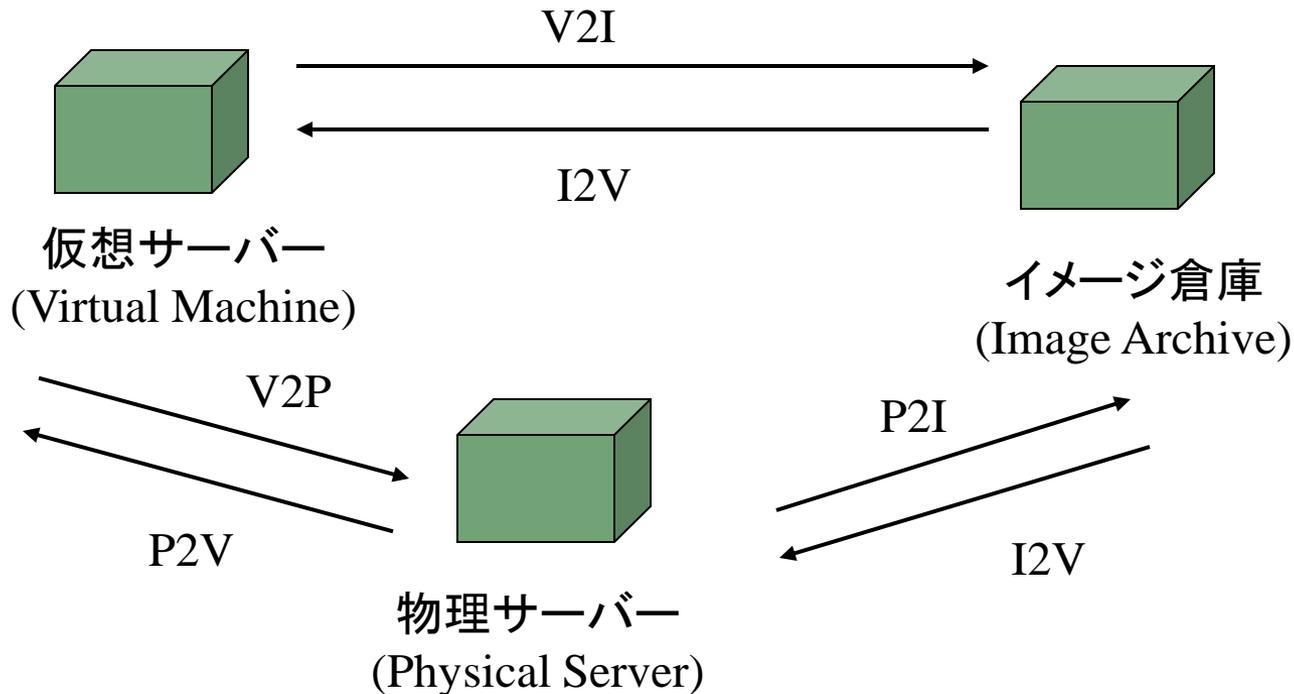
## キャパシティプランニング

システムを構成する各要素のスペックを考慮したシステム・モデルに対して、稼動時の負荷を考慮したシミュレーションを行い、システムに求められる性能要件を満たすために必要となるリソースを算出します。

## 移行シナリオ立案

- 1 新規型クラウド移行、
- 2 吸収型クラウド移行、
- 3 並存型クラウド移行などを検討

変わる運用管理 – 必要なスキルも変わるので、人材育成が大切



## 実装の体験から得た オープンソースクラウド構築技術の比較表

名称	優れている要素
OpenNebula	<p>バックに有力なヨーロッパのスポンサーが控えており、評判が高い。 国内では、クラスキャットがIBMと組んで、独自のGUIを備えた製品を持っている。 2011/03 V2.2 で待望のWeb GUIが備わった。</p>
OpenQRM	<p>多くのデータセンターで、仮想化オブジェクトの操作に便利なツールとして使用されている。 有料のセミナー(16時間)を受講しました。</p>
Enomaly ECP	<p>多くのデータセンターで、仮想化オブジェクトの操作に便利なツールとして使用されている。 日商エレクトロニクスが代理店契約を結んでいる。 パブリックドメイン内でDNS登録されたHostで構成する必要がある。 実験的なインストールでの評価は不可である。</p>
Wakame-vdc	<p>国産である。</p>
Proxmox	<p>失敗なく、最も簡単にプライベートCloudを導入できます。</p>

# Proxmox でPrivate クラウド構築 1

◎次の資料を参考にインストールします。

**Proxmox-VE(仮想環境) 日本語記事**

(<http://www.chall.ne.jp/Productes/proxmoxVE.html>)

**Proxmox-VE(仮想環境) ドイツ**

(<http://www.proxmox.com/>  
[http://pve.proxmox.com/wiki/Main\\_Page](http://pve.proxmox.com/wiki/Main_Page))

◎特徴

- Hypervisor として、OpenVZ と KVM をサポート
- インストーラが Debian Lenny のLinux OS と仮想化マネージャの全てを含めてトータルを担っているため、確実に失敗のないインストールが可能。
- ストレージを追加することで、バックアップジョブを設定できます。

◎構成パラメータ

**OS: Debian Linux 5.0 Lenny**

**URL: <http://172.29.66.107>**

**id: root**

**password: master**

**FQDN(Fully Qualified Domain Name): prox.nhk-mt.co.jp**

完全修飾ドメイン名

# Proxmox でPrivate クラウド構築 2

- ◎ Hypervisor として、OpenVZ を使用する場合は、既存のアプライアンスやテンプレートをダウンロードして利用できます。  
しかも、ネットワークをGUIで設定できます。
- ◎ KVM の場合は cdrom から新規に作成できます。  
ホストに cd をセットし、次のようにパラメータをセット後,"作成する"をクリック。  
ネットワークは VNC で設定します。

'root'としてログインしています

## PROXMOX

ホーム | ログアウト Proxmox Virtual Environment 1.8 www.proxmox.com

### VM マネージャー

- 仮想マシン
- テンプレート
- ISO イメージ

### 設定

- システム
- ストレージ
- バックアップ

### 管理

- サーバー
- ログ
- クラスター

### 仮想マシン

リスト 作成 **マイグレート**

Configuration	
タイプ:	Fully virtualized (KVM) ↓
ISO ストレージ:	local (dir) ↓
インストールメディア:	cdrom device ↓
ディスクストレージ:	local (dir) ↓
ディスク使用量 (GB):	32
名前:	ubuntu10.10Desktop
メモリ (MB):	512
VMID:	104
クラスターノード:	prox (172.29.66.107) ↓
ブート時に起動:	<input type="checkbox"/>
イメージ形式:	qcow2 ↓
ディスクタイプ:	IDE ↓
ゲストのタイプ:	Linux 2.6 ↓
CPU ソケット:	1

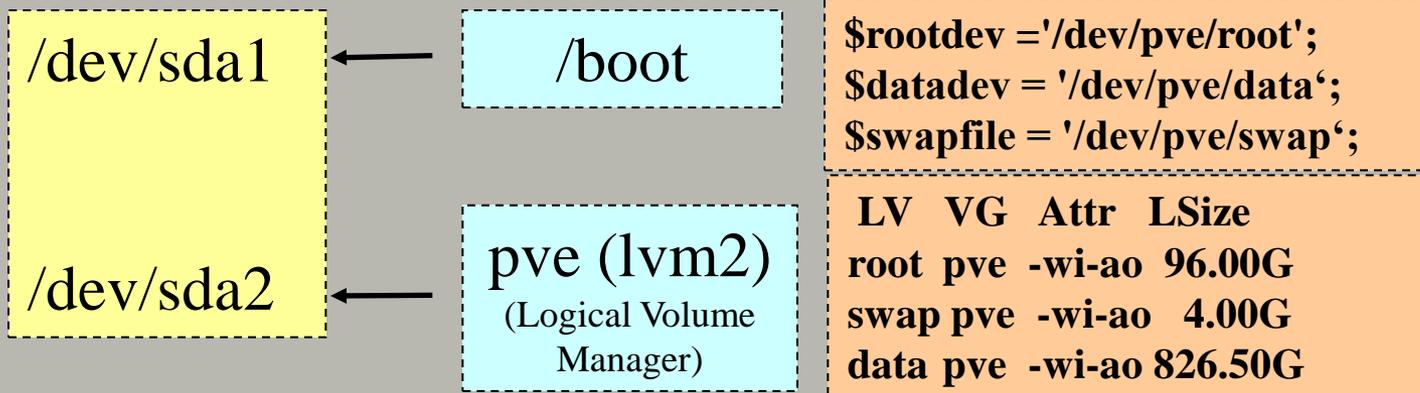
ネットワーク	
ブリッジ:	vmbro ↓
ネットワークカード:	rtl8139 ↓
MAC アドレス:	6A:A0:5B:50:F0:49

➔ 作成する

# Proxmox でPrivate クラウド構築 3 (Disk 構成)

◎ Proxmox の Disk 構成を図示します。

最初に boot パーティションを 512\*1024ブロックで作成し、残りを lvm でパーティションで作成します。



```
/etc/fstab # <file system> <mount point> <type> <options> <dump> <pass>  
/dev/pve/root / ext3 errors=remount-ro 0 1  
/dev/pve/data /var/lib/vz ext3 defaults 0 1  
UUID=e982620b-1342-4a05-a54c-2c515a4c04ff /boot ext3 defaults 0 1  
/dev/pve/swap none swap sw 0 0  
proc /proc proc defaults 0 0
```

```
df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/mapper/pve-root	99083868	726228	93324476	1%	/
tmpfs	2016928	0	2016928	0%	/lib/init/rw
udev	10240	604	9636	6%	/dev
tmpfs	2016928	0	2016928	0%	/dev/shm
/dev/mapper/pve-data	853049764	205136	852844628	1%	/var/lib/vz
/dev/sda1	516040	31632	458196	7%	/boot

# Proxmox でPrivate クラウド構築 4 (Desktop 構成)

- ◎ Proxmox をインストールすると、デフォルトでGUIの無い、サーバー構成となります。これにX-windowsをいんすとーるして、Desktop環境を作成できます。

参考

URL: [http://pve.proxmox.com/wiki/Developer\\_Workstations\\_with\\_Proxmox\\_VE\\_and\\_X11](http://pve.proxmox.com/wiki/Developer_Workstations_with_Proxmox_VE_and_X11)

ここではlxdeをdesktop, iceweaselをbrowserとしてインストールします。

Iceweaselを日本語化するためのAdd-onも利用できます。

```
aptitude install iceweasel-l10n-ja
```

- ◎これだけでは足りません。日本語フォントが必要です。

参考 「Debianで日本語を表示するには」

URL: [http://www.exacteye.com/debian\\_ja.html#debian\\_ja\\_view](http://www.exacteye.com/debian_ja.html#debian_ja_view)

システムとして日本語のロケールが作成されているか確認します。

```
# dpkg-reconfigure locales
```

TrueTypeフォントのインストール

```
# apt-get install ttf-kochi-gothic ttf-kochi-mincho ttf-kochi-gothic-naga10 ttf-sazanami-gothic ttf-sazanami-mincho
```

- ◎ Proxmox にJava をインストールします。

参考 「2010.10.18 Debian LennyにJava JDK 6をインストールする」

URL: <http://www.skullsaba.jp/?p=77>

1. /etc/apt/sources.listに次の行を追加

```
deb http://ftp.nara.wide.ad.jp/debian/ lenny main non-free
```

2. インストール

```
apt-get install sun-java6-jdk sun-java6-demo
```

# Proxmox でPrivate クラウド構築 5 (Browser 構成)

◎ **Opera 11.11**をインストールします。

Google で debian opera で検索して最初に見つけたサイト「Debian lennyでopera ブラウザを使う」  
( <http://tibiwan8-linux.blogspot.com/2010/05/debian-lennyopera.html> )

オペラブラウザのインストール

1)/etc/sources.listをルートでひらいて

gedit(テキストエディタ)で

deb <http://deb.opera.com/opera> lenny non-free

を最後の行に書き加える。

2)認証キー(ルート端末で)

wget -O - <http://deb.opera.com/archive.key> | apt-key add -

を入力してgpgキーを入手。

3)オペラのインストール

```
#apt-get update
```

```
#apt-get install opera
```

◎ **Plugin** を**Opera** に認識させます。

```
www.opera.com/docs/linux/plugins/install#java
```

```
cd /usr/lib/opera/plugins
```

```
ln -s /usr/lib/jvm/java-6-sun-1.6.0.22/jre/lib/amd64/libnjp2.so
```

結果は

```
lrwxrwxrwx 1 root root 58 Jun 21 17:14 libnjp2.so -> /usr/lib/jvm/java-6-sun-1.6.0.22/jre/lib/amd64/libnjp2.so
```

◎ **Plugin** を**Opera** の確認は:

ブラウザで URL に about:plugins と入力する。

VM に Tomcat を入れたので、URL: <http://172.29.66.130:8080/> で確認する。

# Proxmox でPrivate クラウド構築 6 (Browser 構成)

◎ **Chrome 12.0.7442.112**をインストールします。  
Google で debian Chrome で検索して得た案内に従って、  
( <http://www.google.co.jp/chrome/eula.html> )  
64ビット deb( Debian/Ubuntu ) を選択してダウンロード。  
google-chrome-stable\_current\_adm64.deb をダウンロード

## Chromeのインストール

```
#dpkg -i google-chrome-stable_current_adm64.deb
```

パッケージの依存関係エラーが出るので、次のコマンドで解消します。

```
#apt-get install -f
```

◎ **Plugin** を **Chrome** に認識させます。

```
www.opera.com/docs/linux/plugins/install#java
```

```
cd /opt/google/chrome
```

```
mkdir pluguns
```

```
cd plugins
```

```
ln -s /usr/lib/jvm/java-6-sun-1.6.0.22/jre/lib/amd64/libnjp2.so
```

結果は

```
lrwxrwxrwx 1 root root 58 Jun 21 17:14 libnjp2.so -> /usr/lib/jvm/java-6-sun-1.6.0.22/jre/lib/amd64/libnjp2.so
```

◎ **Plugin** を **Chrome** の確認は:

ブラウザで URL に about:plugins と入力する。

# Proxmox でPrivate クラウド構築 7 (変換)

## VMWare - - > Promxmox( KVM )

◎ VMWare イメージを qcow2 形式のKVM イメージに変換します。

参考

[http://pve.proxmox.com/wiki/Migration\\_of\\_servers\\_to\\_Proxmox\\_VE#Prepare\\_the\\_Windows\\_operating\\_system\\_2](http://pve.proxmox.com/wiki/Migration_of_servers_to_Proxmox_VE#Prepare_the_Windows_operating_system_2)  
qemu-img convert -f vmdk ide.vmdk -O qcow2 ide.qcow2

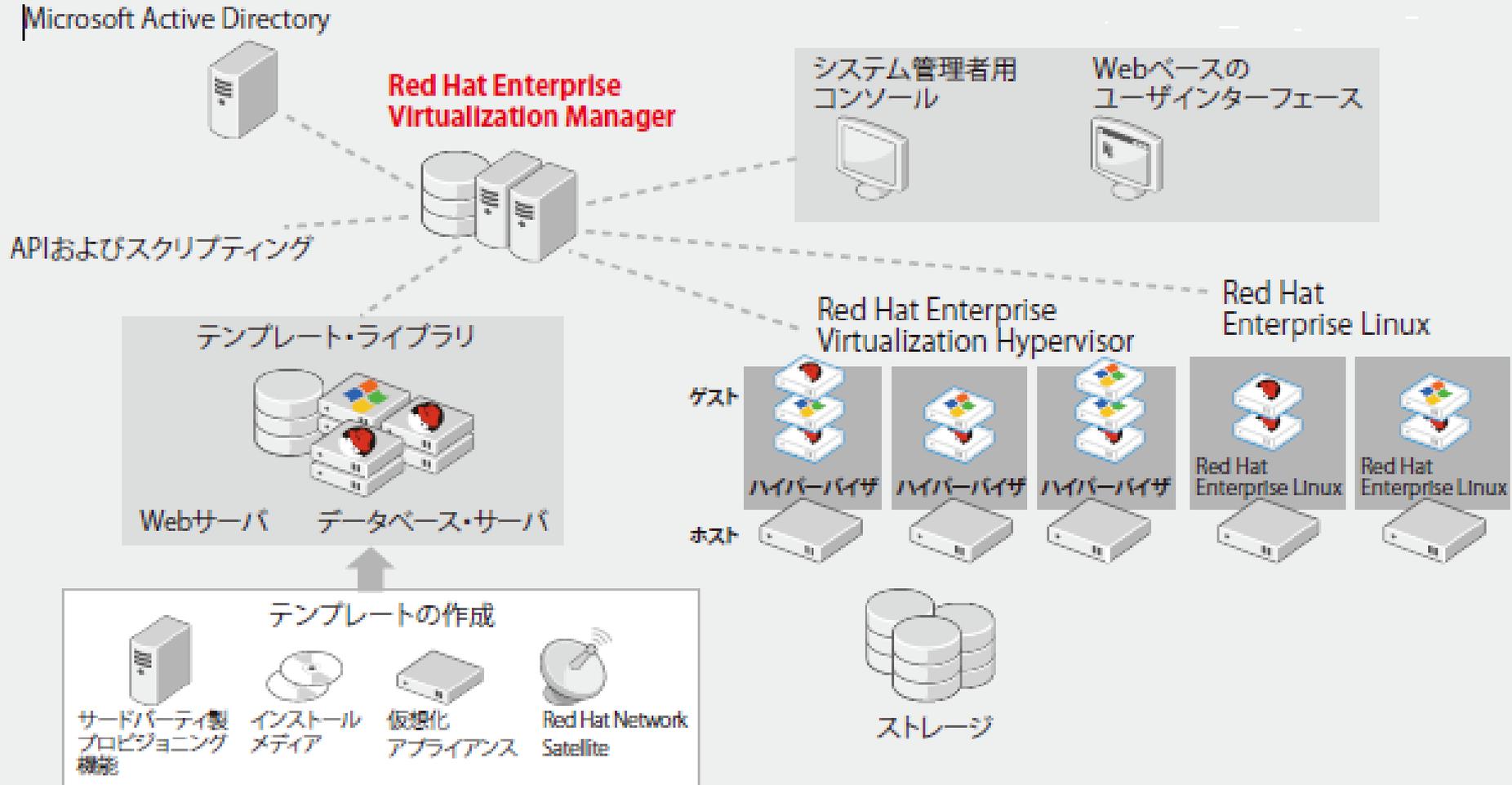
◎ ide.qciw2 を管理端末(Windwos) にコピーして、管理端末からProxmoxの web ui から  
テンプレートとしてアップロードを試みても、次のエラーで進みません。

エラー: 名称 'ide.qcow2' はテンプレートの命名スキームに準拠していません  
(<OS>-<OSVERSION>-<NAME>-<VERSION>\_(i386|amd64).tar.gz)

◎ そこで、イメージを qcow2 形式から iso 形式に変換して使用します。

qemu-img convert -f qcow2 ide.qcow2 -O raw Chrome\_OS\_Linux.i686-1.3.iso

# RHEV ( RED HAT EnterPrise Virtualization ) 2.2 ->3.0



RHEL6.0 が Enterprise Virtualization(RHEV) の制御下に入るのは V3.0以降だそうです。

しかも、現在のところ、RHEV Managerは Windows Server でしか動作しません。

年内に Java 版に切り替えて、自社サーバーに載せられるように開発中。

他社と比較して数分の一のコストで実用的なプライベートクラウドを構築できます。

- ◎ REDHATは仮想化で KVM に命を懸ける事にしました。  
これからの新バージョンの Linux (例えばRHEL6.0)はKVM のみの搭載。
- ◎ REDHATの仮想化製品は他社と同様に、  
サーバ仮想化と、デスクトップ仮想化をサポートします。
- ◎ デスクトップ仮想化において、SPICE (リモートディスプレイ・テクノロジー)を採用。

### 参考

サーバ仮想化ソフトウェア主要4製品 導入コスト比較！ ～本番環境における試算  
<http://techtarget.itmedia.co.jp/tt/news/1102/22/news01.html>

VMware、仮想化の機能競争でMicrosoft、Citrix、Red Hatを依然リード  
<http://techtarget.itmedia.co.jp/tt/news/1104/12/news02.html>

さくらインターネットが KVM 採用  
完全仮想化を実現した、新世代の仮想専用サーバ  
<http://vps.sakura.ad.jp/>

# Proxmox でPrivate クラウド構築 6 (Browser 構成)

◎ **Firefox 3.6.17** をインストールします。

```
cd /usr/local/share
```

```
tar -xvjf /home/steria/firefox-3.6.17.tar.bz2
```

Now you'll have a directory named "firefox" where you just unpacked it -- /usr/local/src in the example above. Now you just need to symlink it /usr/local/bin (you should still be root).

?

```
ln -s /usr/local/share/firefox/firefox /usr/local/bin
```

◎ **Firefox 3.6.17** をインストールします。

動かそうとすると

```
./firefox-bin: error while loading shared libraries: libgtk-x11-2.0.so.0: cannot open shared object file: No such file or directory
```

で起動失敗する。

32bitなバイナリが64bitのライブラリをリンクしてくれないのが原因。

32bitなライブラリは ia32-libs-gtk にあるらしい

```
$ sudo apt-get install ia32-libs-gtk
```

でOKのはずだったが、立ち上がらない。