

LPIC 304 技術解説セミナー

2017年3月18日(土) 13:30～16:30

LPI-Japan認定トレーナー

株式会社ゼウス・エンタープライズ

鯨井 貴博



Linux・Networkの基礎から、
現場経験を活かしたトラブルシュートまで幅広い講義を行います！

[プロフィール]

LPIC レベル3・情報処理技術者資格などを保有し、
現場ではネットワーク構築・海外メーカー国内一次代理店でのテクニカルサポート業務を経験。
現在は人事管理の傍ら、講師としてIT未経験から学んできた経験を交えた講義を行っている。
他にも、LPI-Japan認定トレーナー、オープンソース技術 開発研究所、
日本Nginxユーザー会やいろんなイベントに出没中。

[保有資格]

LPIC レベル3 Core(301) Mixed Environment(302)

Security(303) Virtualization & High Availability(304)

HTML5プロフェッショナル レベル1

ACCEL(Apache CloudStack技術者認定資格 by LPI-JAPAN) ...認定第1号!!

情報セキュリティスペシャリスト

ネットワークスペシャリスト

応用情報処理技術者/基本情報処理技術者

情報セキュリティマネジメント

MCP70-640 Microfoft Windows Server 2008 Active Directory

ITIL Foundation

CCNA

MOS(Word2010/Excel2010)

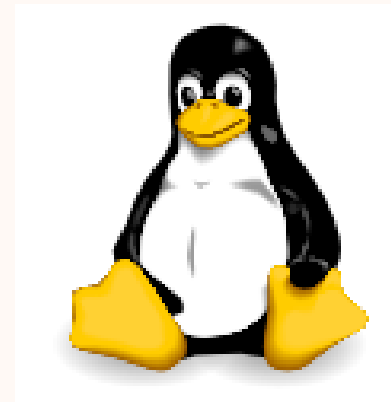




Linux開発者のLinus Torvaldsさんと私
(ゼウス・ラーニングパワーCarl先生も一緒)
in LinuxConJapan



nginx開発者のIgor Sysoevさんと私
in 日本Nginxユーザー会



鯨井貴博

BLOG <http://opensource.tech.hatenablog.jp/>

お気軽に連絡下さい！！



本日のアジェンダ

- ①LPIC304の試験概要
- ②LPIC304を学習する上でのポイント
- ③技術解説

主題330: 仮想化

330.1 仮想化の概念と理論

330.2 Xen

330.3 KVM

主題334: クラスタ管理

334.3 Pacemaker

主題335: クラスタストレージ

335.1 DRBD

- ④サンプル問題
- ⑤お知らせ
- ⑥質疑応答



①LPIC304の試験概要



LPICの構成

初級

上級

LPICレベル1 LPIC-1

サーバの
構築、運用・保守

実務に必要なLinuxの基本操作とシステム管理が行える
エンジニアであることを証明できます。

LPICレベル2 LPIC-2

ネットワークを含む、
コンピュータシステムの構築、運用・保守

Linuxのシステムデザイン、ネットワーク構築において、企
画、導入、維持、トラブルシューティング、キャパシティプ
ランニングができるエンジニアであることを証明できます。

LPICレベル3 LPIC-3

各分野の最高レベルの技術力を持つ
専門家レベル

LPIC-3 Specialty Mixed Environment

〈300試験〉:

Linux、Windows、Unixが混在するシステムの設計、構
築、運用・保守ができるエキスパートエンジニアであるこ
とを証明できます。

LPIC-3 Specialty Security

〈303試験〉:

セキュリティレベルの高いコンピュータシステムの設計、
構築、運用・保守ができるエキスパートエンジニアである
ことを証明できます。

LPIC-3 Specialty Virtualization & High Availability

〈304試験〉:

クラウドコンピューティングシステム(クラウド)の設計、構
築、運用・保守ができるエキスパートエンジニアであるこ
とを証明できます。

該当試験

101試験

102試験

201試験

202試験

300試験

303試験

304試験



LPIC304の試験概要

どのような試験か？

Virtualization & High Availabilityの名の通り、
仮想化・高可用性を理解し、
クラウド関連の構築/運用の専門家を証明するものです。

試験時間：90分

出題数：約60問

出題形式：選択/記述

受験費用：30,000円(消費税別)



主題 330: 仮想化

- 330.1 仮想化の概念と理論 (重要度: 8)
- 330.2 Xen (重要度: 9)
- 330.3 KVM (重要度: 9)
- 330.4 その他の仮想化ソリューション (重要度: 3)
- 330.5 Libvirt と関連ツール (重要度: 5)
- 330.6 クラウド関連ツール (重要度: 2)

主題 334: 高可用クラスタ管理

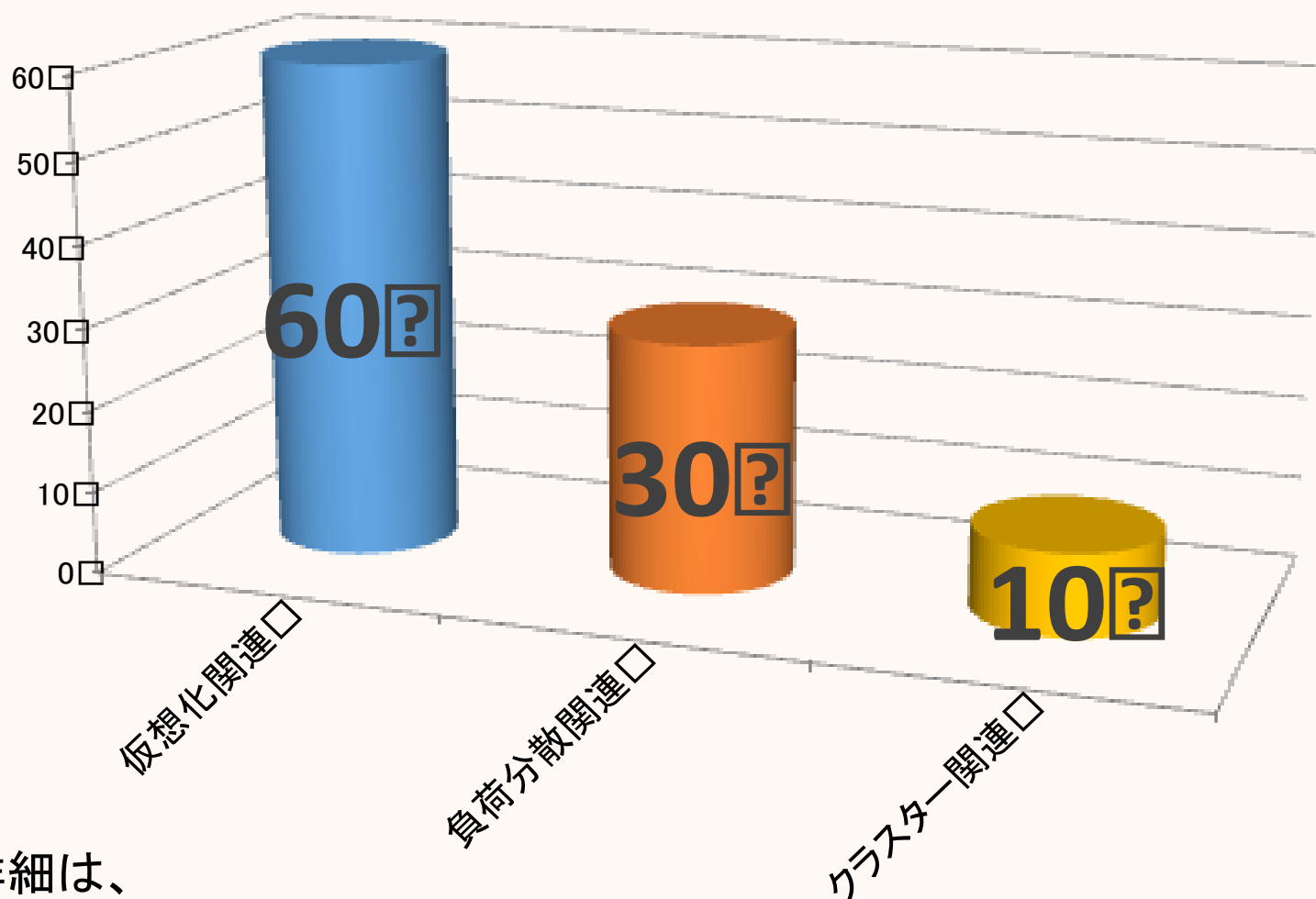
- 334.1 高可用性の概念と理論 (重要度: 5)
- 334.2 ロードバランスクラスタ (重要度: 6)
- 334.3 フェイルオーバークラスタ (重要度: 6)
- 334.4 エンタープライズ Linux ディストリビューションにおける高可用性 (重要度: 1)

主題 335: 高可用クラスタストレージ

- 335.1 DRBD / cLVM (重要度: 3)
- 335.2 クラスタファイルシステム (重要度: 3)



出題範囲の割合(%)



詳細は、

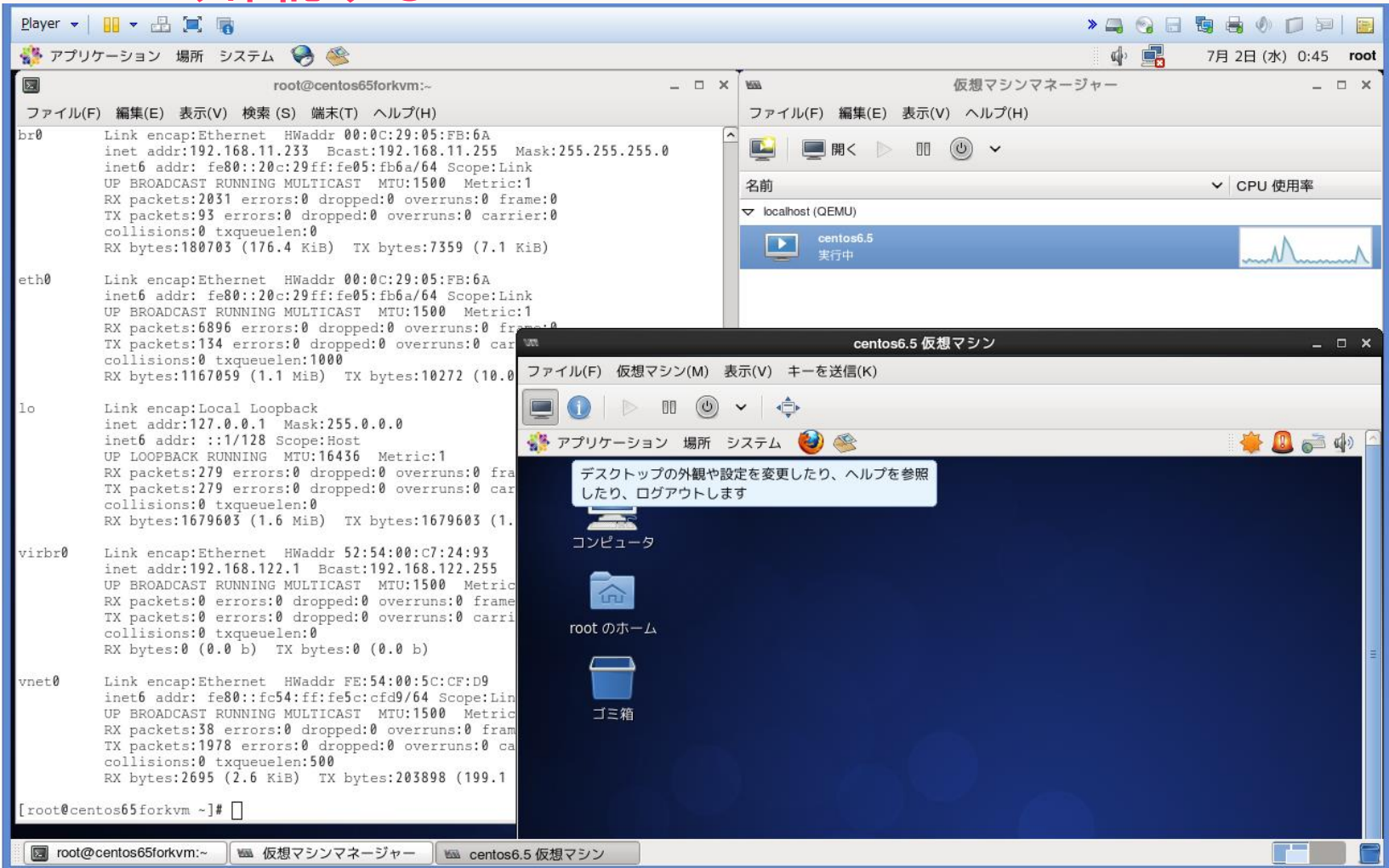
<http://www.lpi.or.jp/lpic3/range/range304.shtml>をご確認下さい。



②LPIC304を学習する上でのポイント



- 実機での操作にてどのような機能・操作方法など、
しっかり確認すること！





・注意点！！

XenやKVMなどを仮想マシン上/物理マシン上
どちらで環境構築する場合も、
64ビット・仮想化支援機能(Intel-VT/AMD-V)に
対応したCPU、2～4GB以上の搭載メモリ(目安)、
HD容量にも余裕がある など
ある程度のマシンスペックが必要となります。



教材等

各ソフトウェアのサイト(英語情報多い)を読むなどの方法があります。

<http://www.lpi.or.jp/lpic3/book.shtml>
version2.0対応の本もあります！

他にも、
Xen/KVMなどのHypervisor関連、
仮想化関連、ロードバランシング関連、
クラスタ関連の専門書にも目を通した方が
いいでしょう。





③技術解説

主題330: 仮想化



330.1 仮想化の概念と理論



仮想化のメリット

- ・複数のサーバを1台の物理マシンに統合することが出来る。
⇒業務レベルでは、消費電力やサーバ設置のスペースなどを減らすことが出来、より多くのサーバを扱える。
個人レベルでは、1台の物理マシンで複数のサーバを保持可。
- ・ハードウェアに依存しないので、サーバの追加および廃棄が楽。
- ・サーバリソースの不足にも、サーバのコピー等で容易に対応可。
⇒サーバのインストールや設定などの管理も楽になる。
- ・過去のハードウェアを仮想化することにより、最新のハードウェアで動作しないOSを動作させることが出来る。

など



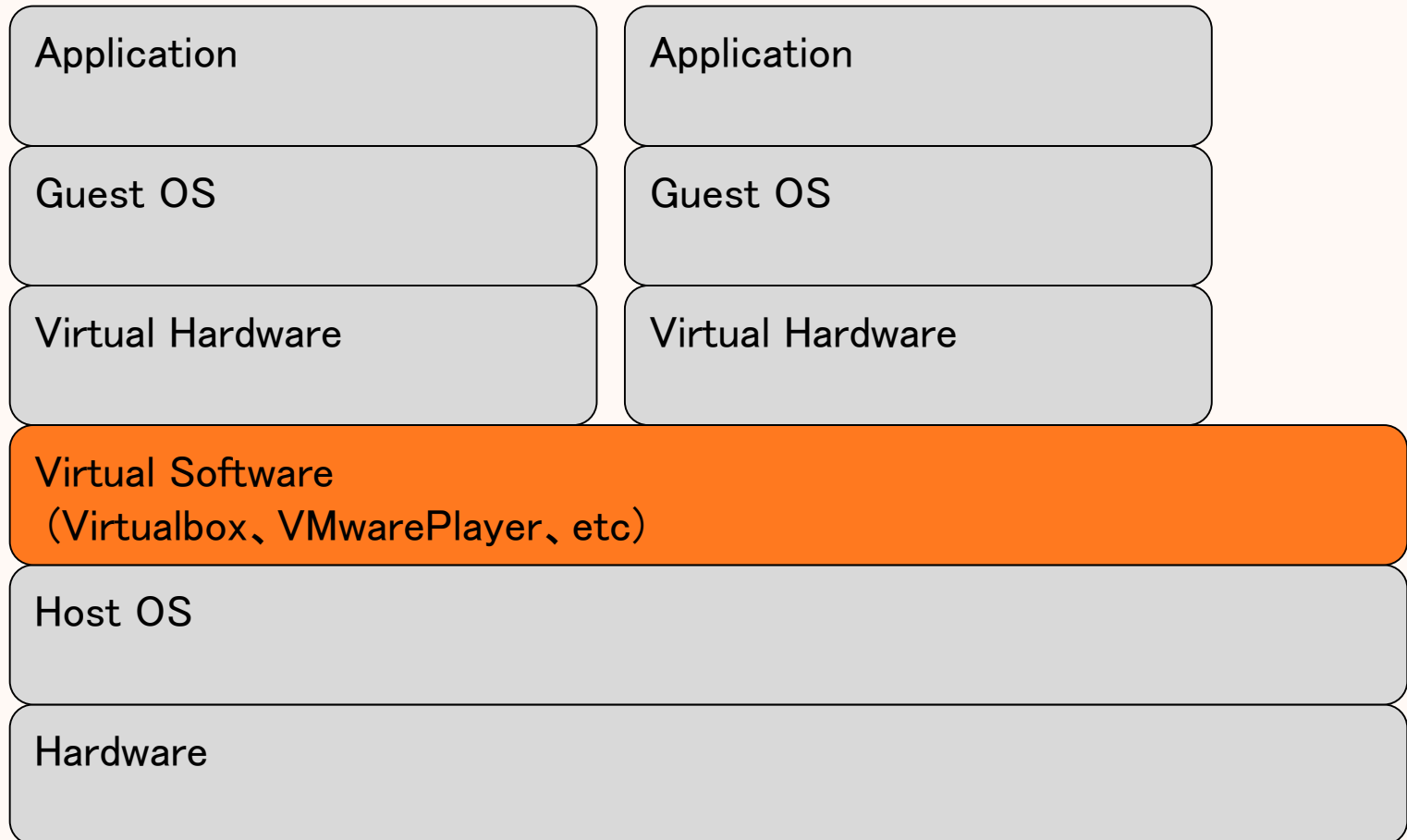
仮想化のデメリット

- ・物理サーバの障害が、格納するサーバ全体に影響を及ぼす。
⇒物理サーバのメンテナンスなどの際にも同様の事が考えられる
- ・特殊過ぎるハードウェアは利用出来ないことがある。
- ・物理マシン・仮想マシンの両方を管理する必要があり、管理が大変。

など

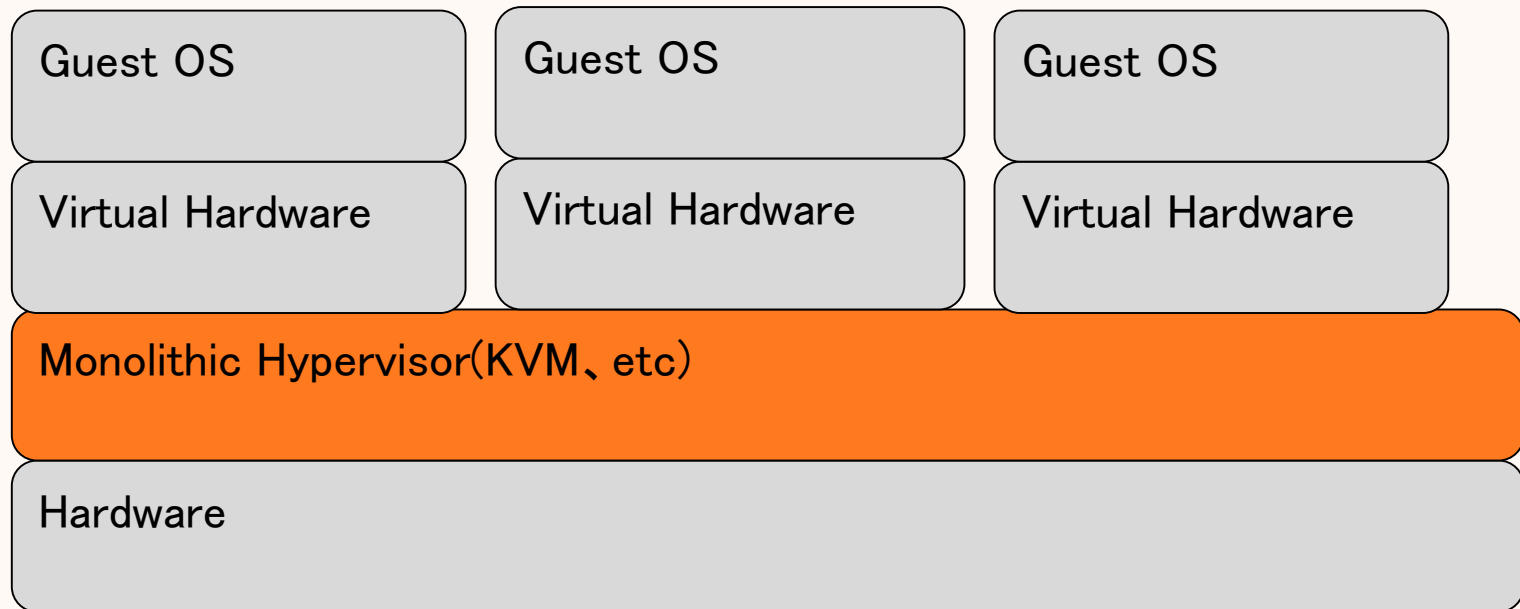


仮想マシンモニター I (Application Type)



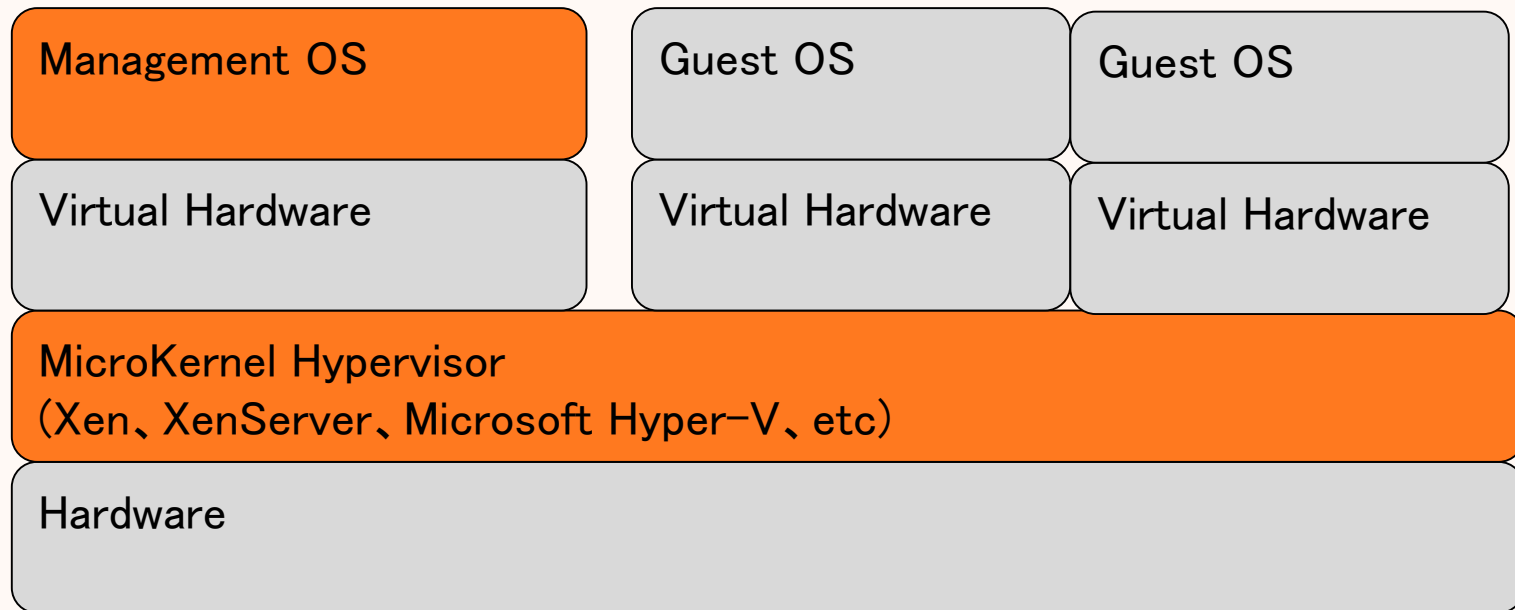


仮想マシンモニター II (Hypervisor Type1)



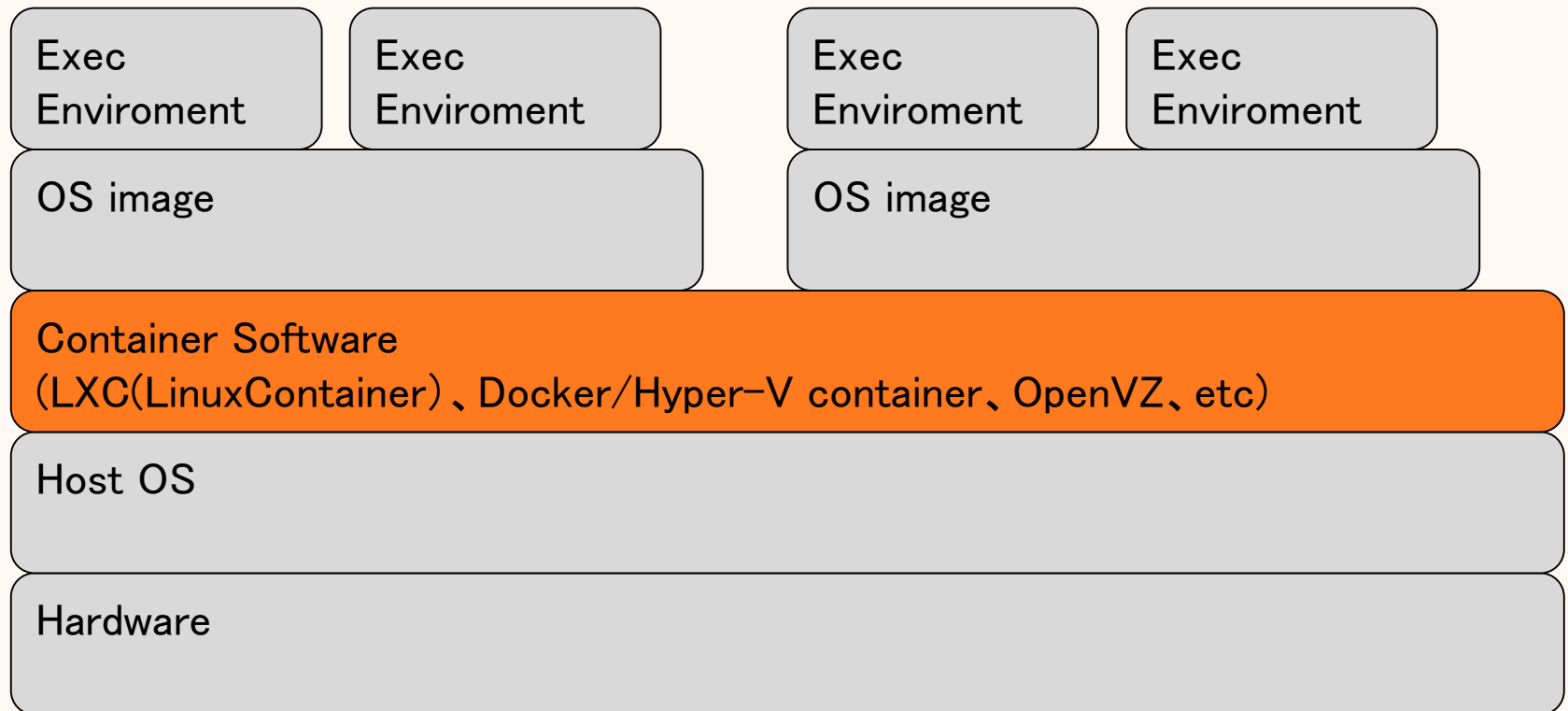


仮想マシンモニターⅢ(Hypervisor Type2)



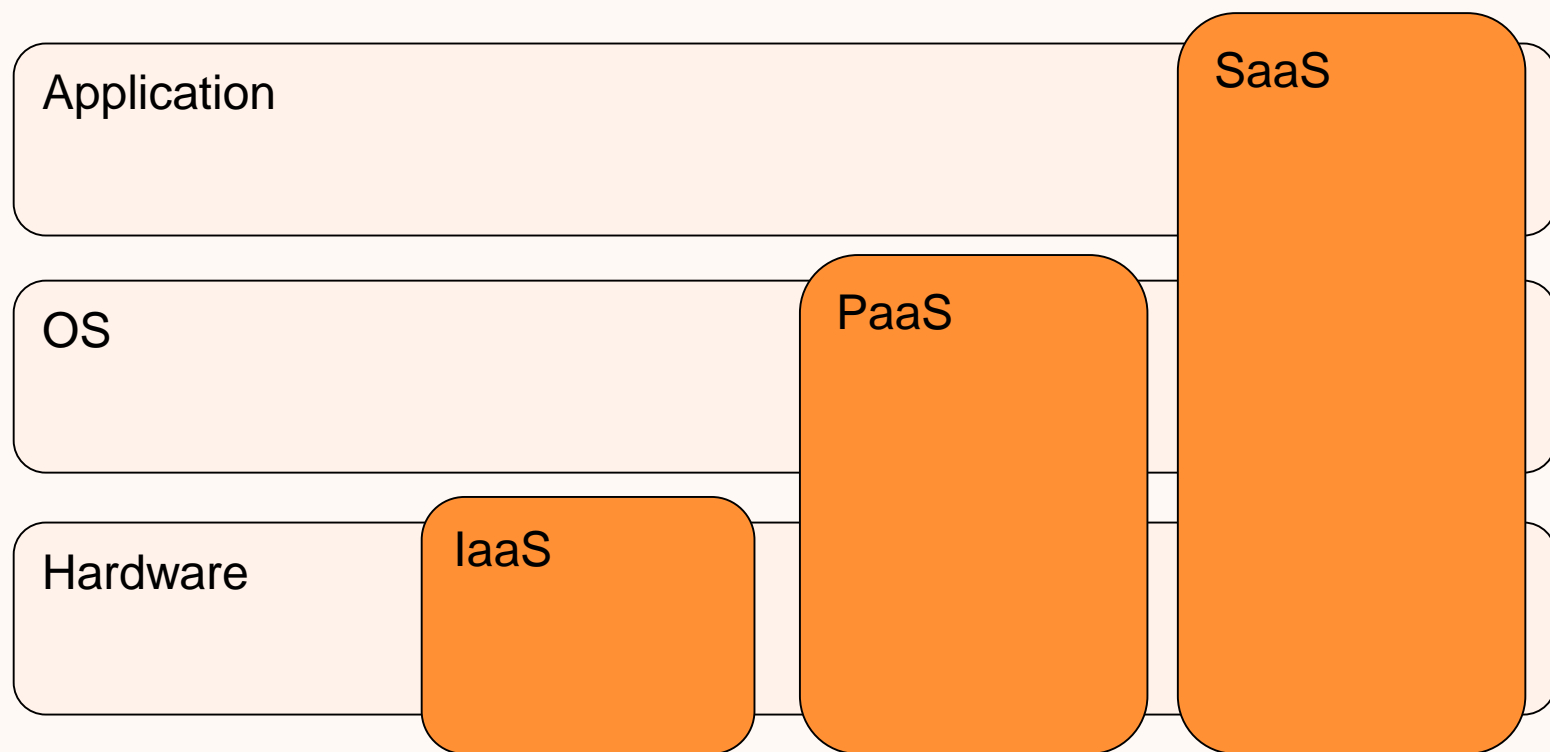


仮想マシンモニターⅣ(Container Type)



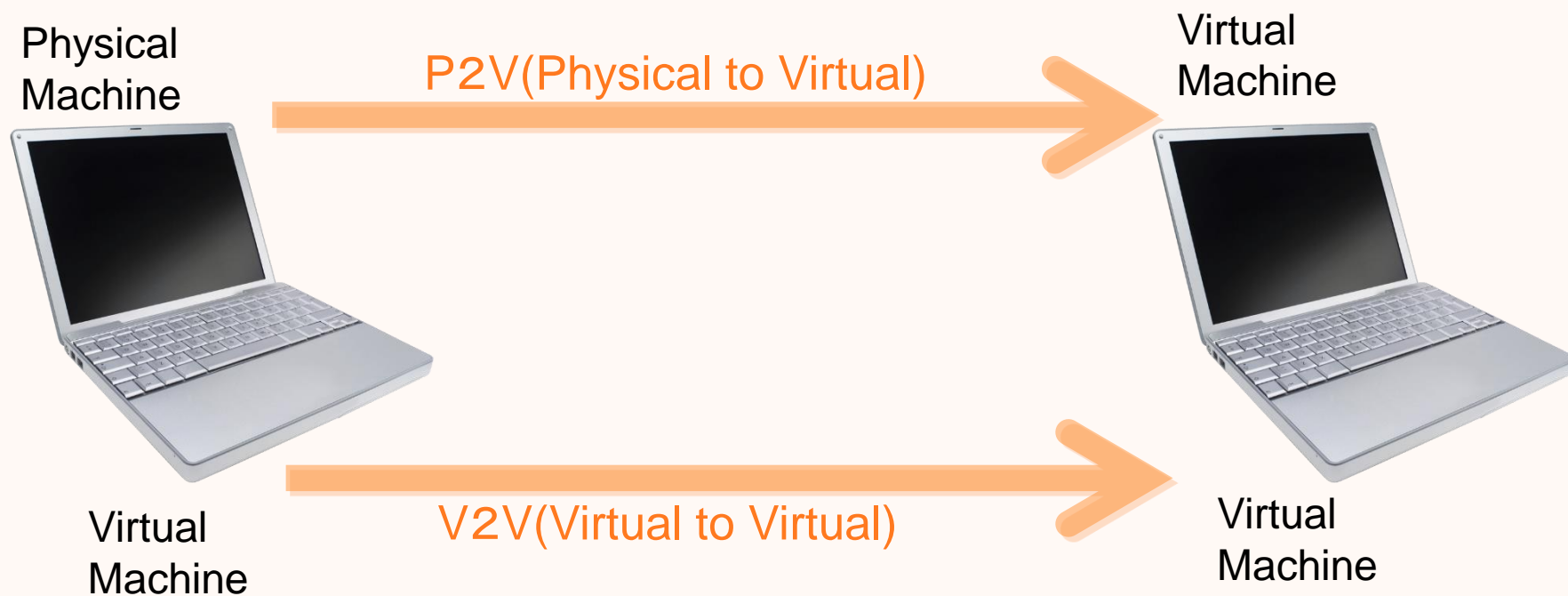


IaaS/SaaS/PaaS





P2V、V2V





CPU flags

まず、XenやKVMなどの仮想化を利用するには、
PCのCPUにおいて、
仮想化支援機能が有効になっている必要があります。

/proc/cpuinfoにて行うことが確認出来、
Intel製のCPUの場合にはvmxというflagの有無、
AMD製のCPUの場合にはsvmというflagの有無を確認します。



Intel製CPUの場合

```
[root@centos65forkvm ~]# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 58
model name     : Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz
stepping       : 9
cpu MHz        : 3192.749
cache size     : 6144 KB
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
                sse2 ss syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts xtopology tsc_reliable nonstop_
air_spinlock pni pclmulqdq vmx ssse3 cx16 pcid sse4_1 sse4_2 x2apic popcnt aes xsave avx f16c rd
hf_lm ida arat epb xsaveopt pln pts dts tpr_shadow vnmi ept vpid fsgsbase smep
bogomips       : 6385.49
clflush size   : 64
cache_alignmen : 64
address sizes   : 40 bits physical, 48 bits virtual
power managemen:
```



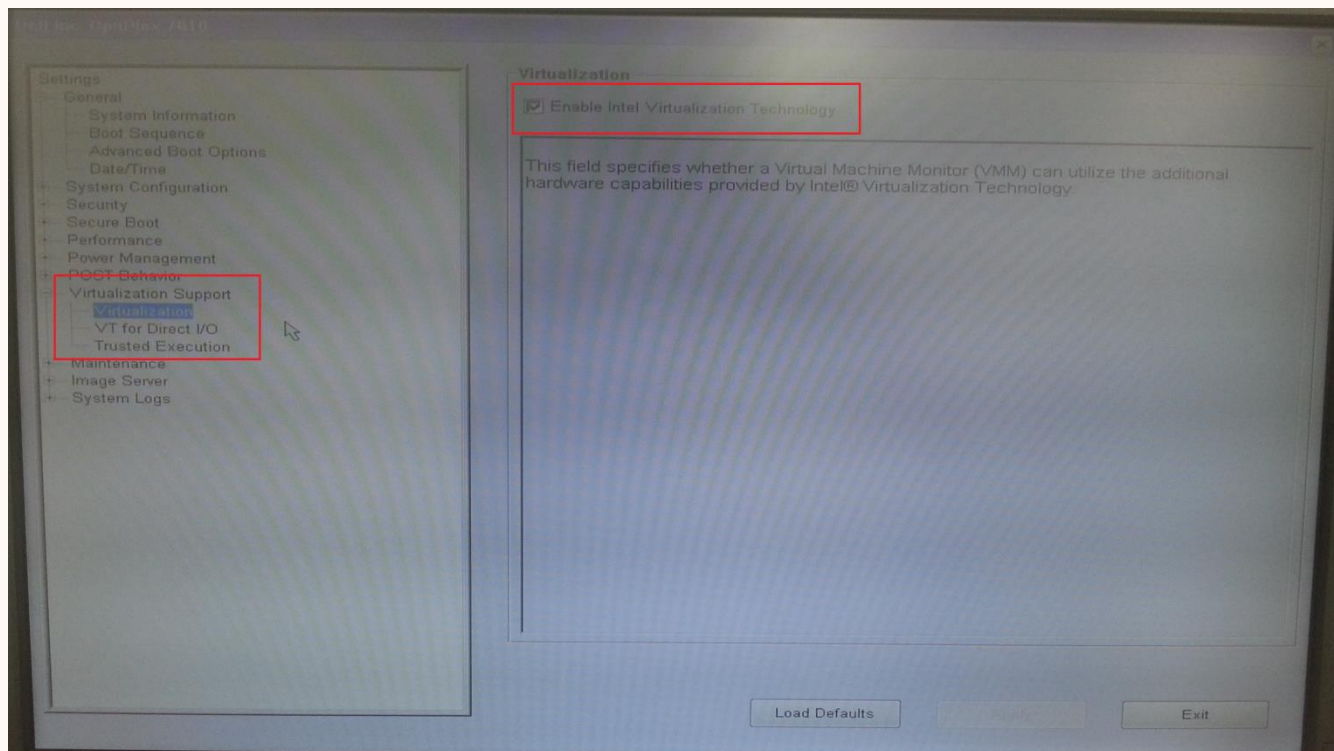
AMD製CPUの場合

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
[root@centos65forKVM ~]# cat /proc/cpuinfo
processor          : 0
vendor_id         : AuthenticAMD
cpu family       : 16
model            : 10
model name       : AMD Phenom(tm) II X6 1090T Processor
stepping         : 0
cpu MHz          : 3206.365
cache size       : 512 KB
fpu              : yes
fpu_exception    : yes
cpuid level      : 6
wp               : yes
flags            : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm
3dnowext 3dnow constant_tsc rep_good tsc_reliable nonstop_tsc aperfmperf unfair
_spinlock pnix cx16 x2apic popcnt hypervisor lahfm_lm svm extapic cr8_legacy abm s
se4a misalignsse 3dnowprefetch osvw npt svm_lock nrip_save
bogomips         : 6412.73
TLB size         : 1024 4K pages
clflush size     : 64
cache_alignment  : 64
address sizes    : 40 bits physical, 48 bits virtual
power management:
```



vmxやsvmが確認出来ない場合、
PCのBIOSやEFIなどで仮想化機能が有効化されているか確認し、
無効であれば有効にします。

※VirtualizationやVirtualization Support、Secure Virtual Machine
などという項目です。





完全仮想化(Full Virtualization)

ゲストOSのソース修正を伴わず、ハードウェアを完全に仮想化する方式
Intel-VTやAMD-VなどのCPU仮想化機能を利用する

準仮想化(Para Virtualization)

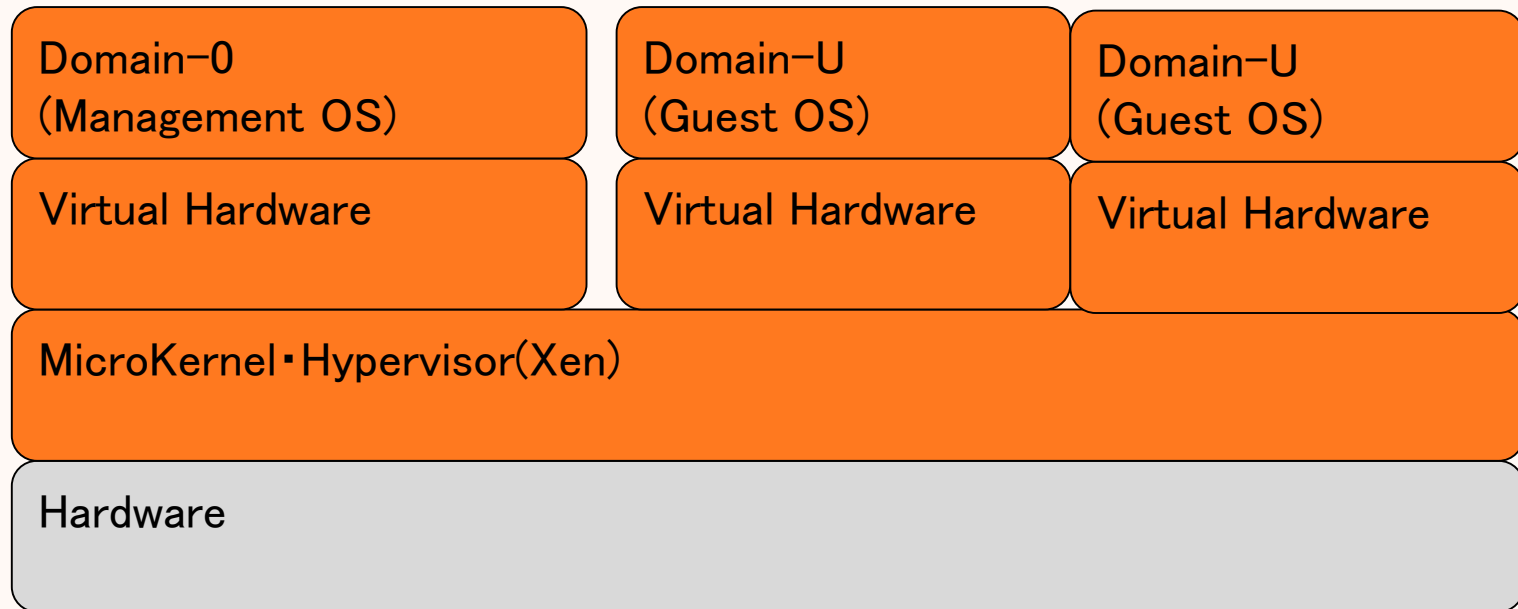
ゲストOSを一部修正処理性能を向上させる方式



330.2 Xen



Xenとは、ハイパーバイザータイプの仮想化を実現するものです。
また、管理OSの事をDomain-0、
ゲストOSの事をDomain-Uと呼びます。





Xenのインストール

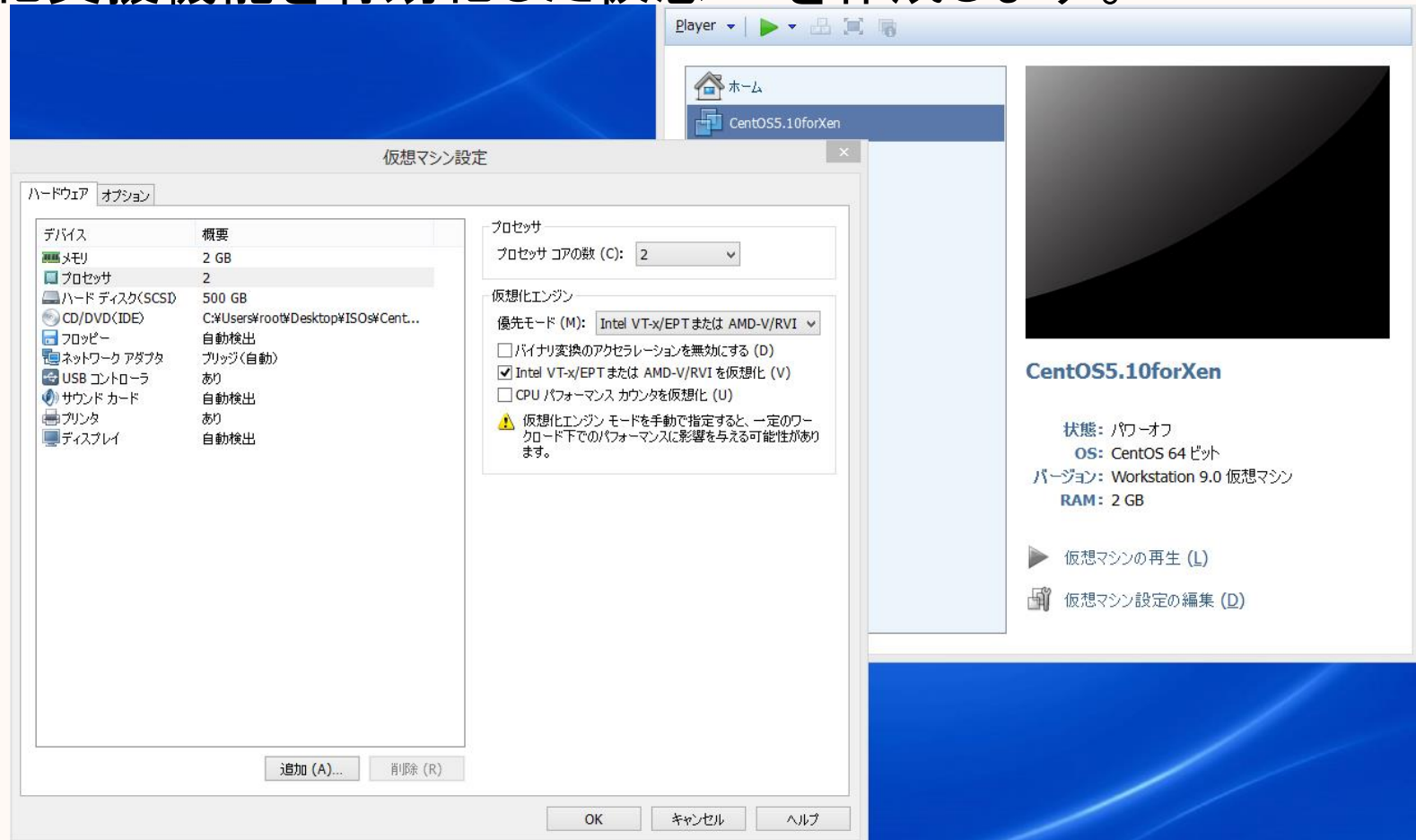
レポジトリより、
CentOS-5.11(x86_64)の
ISOイメージを取得します。

Index of /Linux/centos/5.11/isos/x86_64

Name	Last modified	Size	Description
 Parent Directory		-	
 CentOS-5.11-x86_64-bin-1of9.iso	27-Sep-2014 07:10	631M	
 CentOS-5.11-x86_64-bin-1to9.torrent	29-Sep-2014 18:43	210K	
 CentOS-5.11-x86_64-bin-2of9.iso	27-Sep-2014 07:10	627M	
 CentOS-5.11-x86_64-bin-3of9.iso	27-Sep-2014 07:10	619M	
 CentOS-5.11-x86_64-bin-4of9.iso	27-Sep-2014 07:10	609M	
 CentOS-5.11-x86_64-bin-5of9.iso	27-Sep-2014 07:10	629M	
 CentOS-5.11-x86_64-bin-6of9.iso	27-Sep-2014 07:11	631M	
 CentOS-5.11-x86_64-bin-7of9.iso	27-Sep-2014 07:11	624M	
 CentOS-5.11-x86_64-bin-8of9.iso	27-Sep-2014 07:11	629M	
 CentOS-5.11-x86_64-bin-9of9.iso	27-Sep-2014 07:11	345M	
 CentOS-5.11-x86_64-bin-DVD-1of2.iso	27-Sep-2014 06:08	4.4G	
 CentOS-5.11-x86_64-bin-DVD-1to2.torrent	29-Sep-2014 18:43	209K	
 CentOS-5.11-x86_64-bin-DVD-2of2.iso	27-Sep-2014 06:08	870M	
 CentOS-5.11-x86_64-netinstall.iso	22-Sep-2014 23:10	15M	

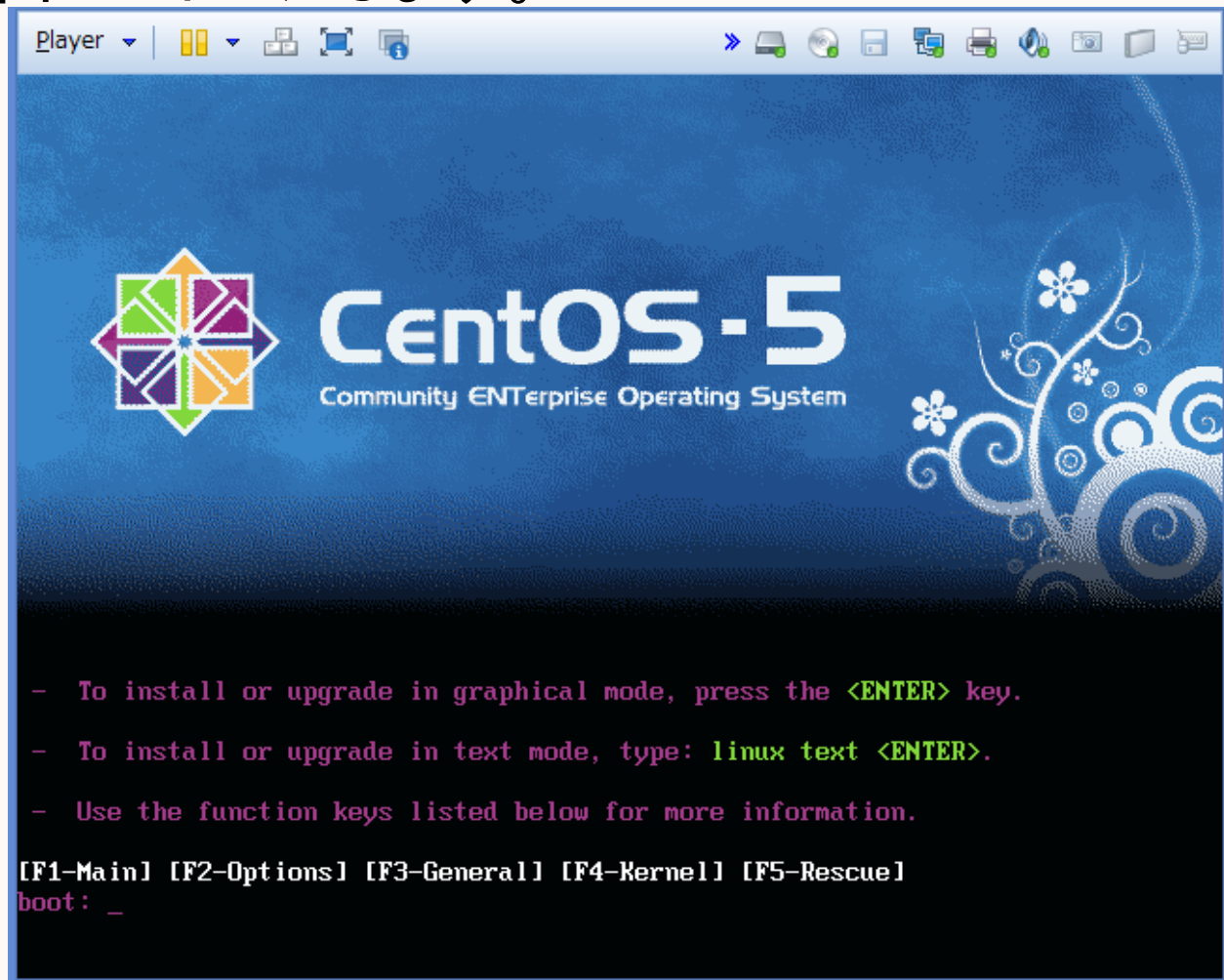


仮想化ソフトウェアにて、
CPU仮想化支援機能を有効化した仮想PCを作成します。



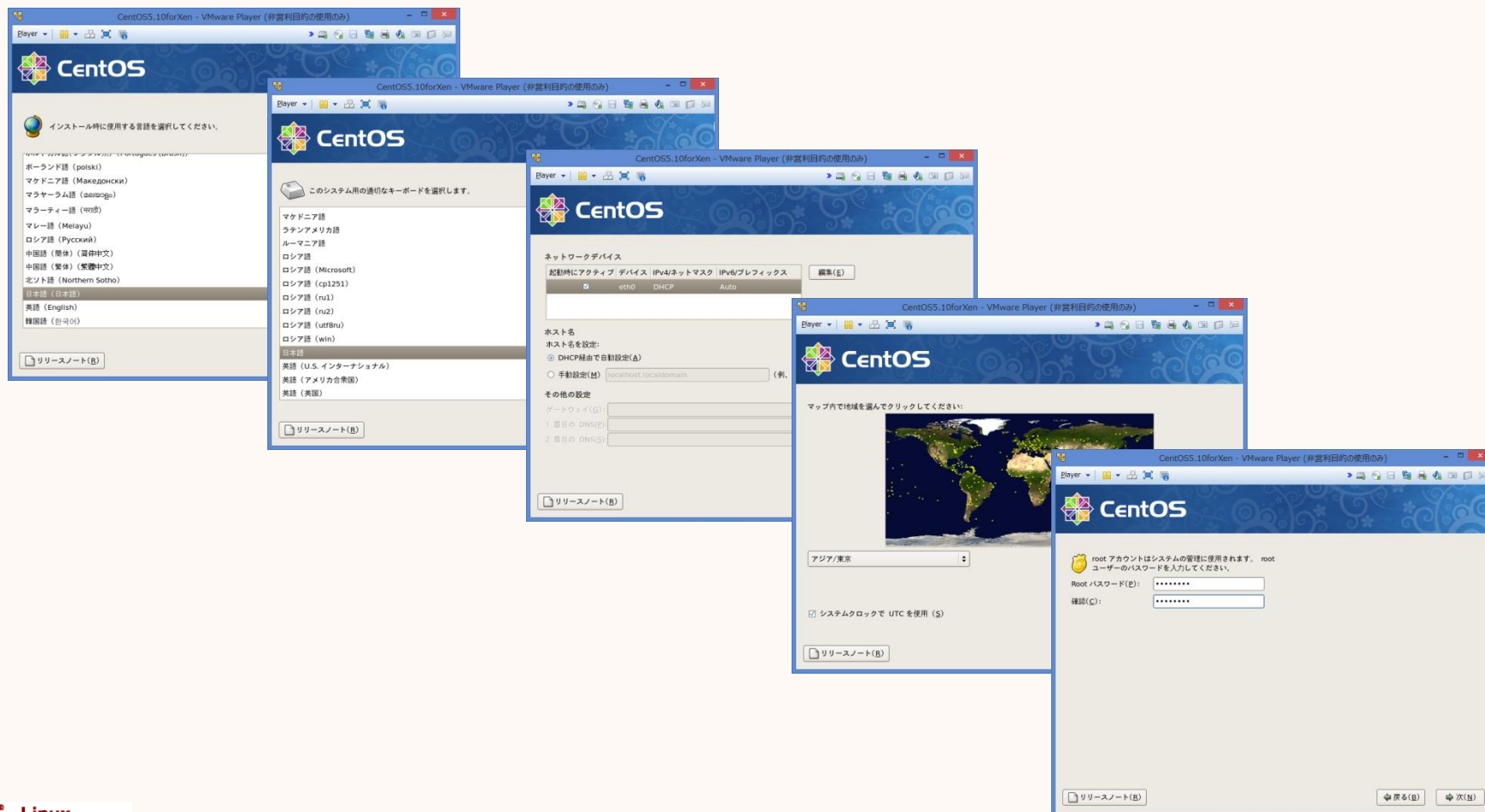


CentOS5.11 (64bits)をインストールします。





ソフトウェアの選択までは、自身の環境等に合わせて設定を行います。





ソフトウェアの選択にて、
「今すぐカスタマイズする」に
チェックを入れる。





仮想化 > Xenに
チェックを入れる。





仮想化 > Xenにより、

- libvirt
- virt-managerなどがインストールされます。



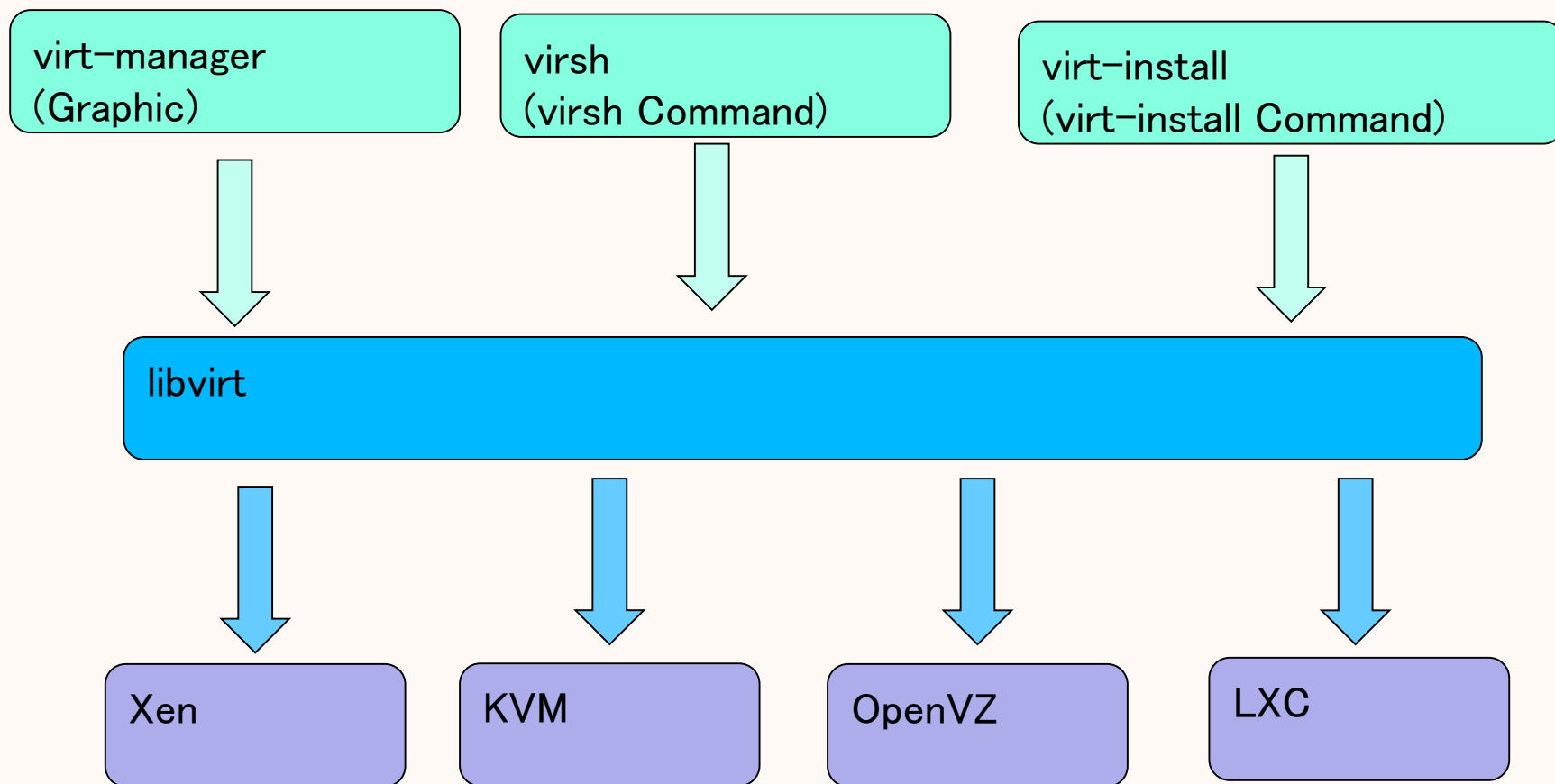


libvirtとは?

Xen、KVM、OpenVZ、LXCなどを制御するライブラリ

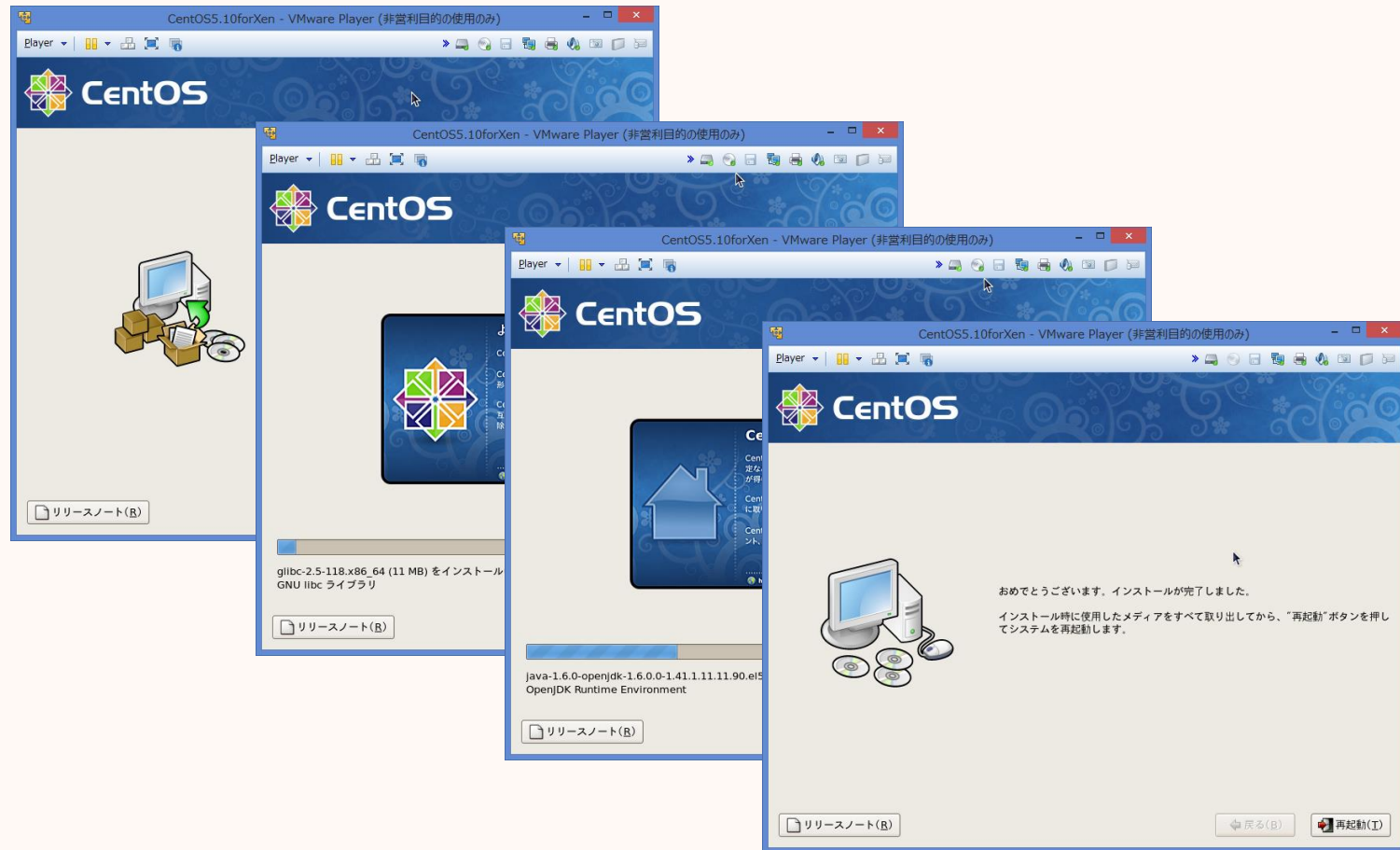
そして、libvirtは、

- virt-manager
- virshコマンド
- virt-installコマンド などからの操作を行う事が出来ます。





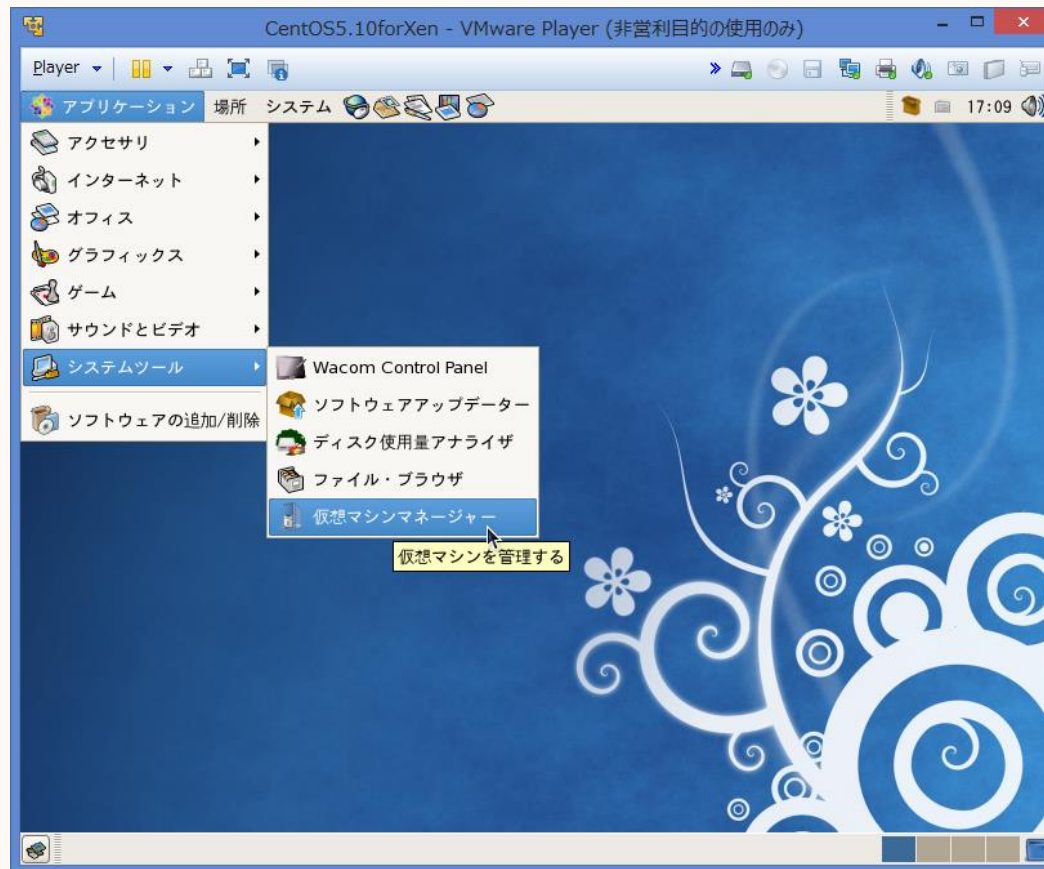
更にインストールを進めます。
SELinuxは無効にしておきます。





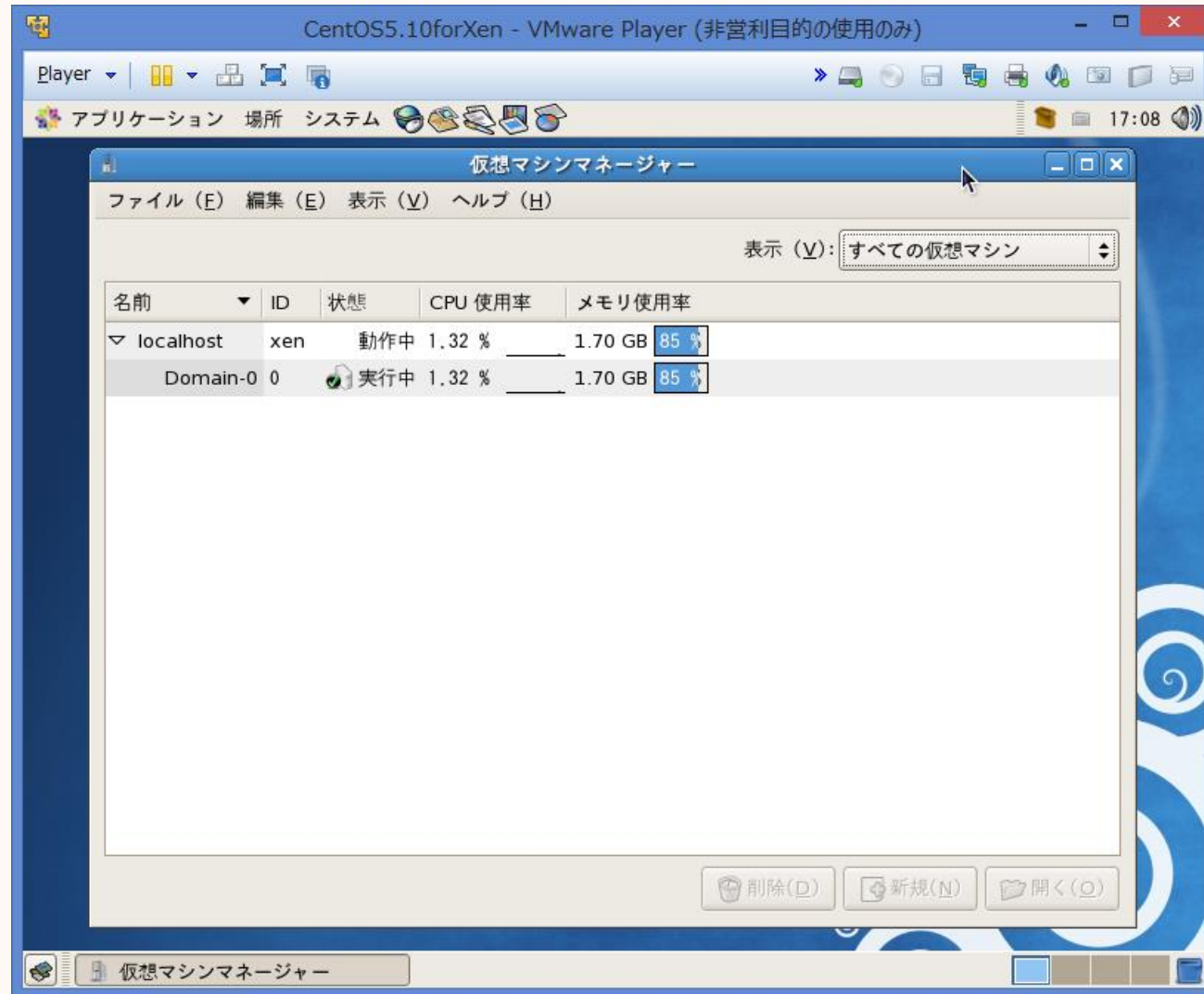
Xenを使ってみる by virt-manager

アプリケーション > システムツール > 仮想マシンマネージャーから
virt-managerを起動します。



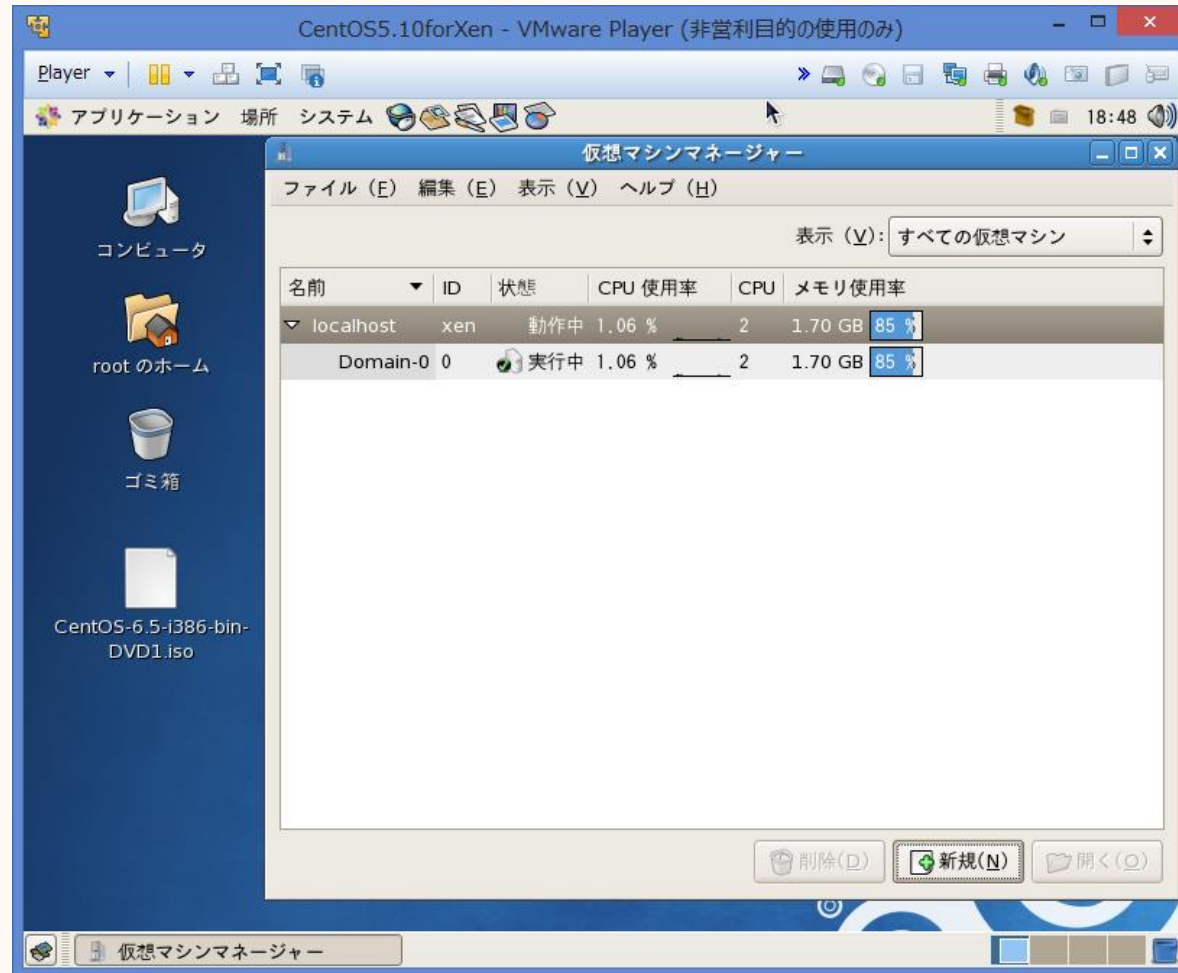


初期状態では、
Domain-0(管理OS)
のみが起動



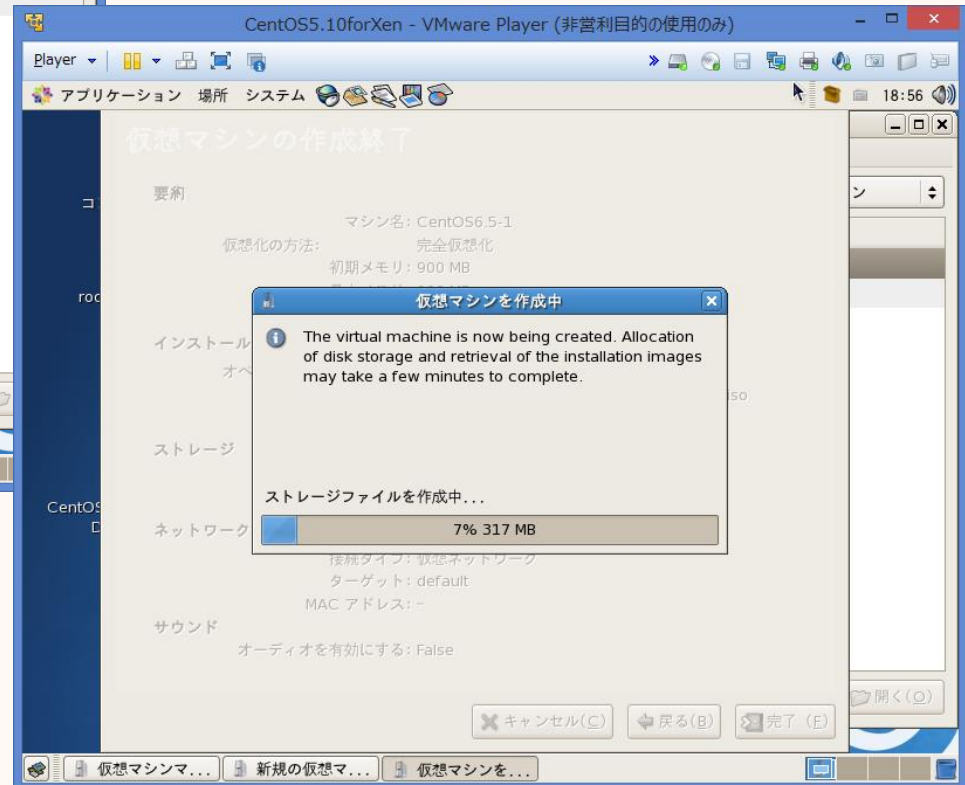
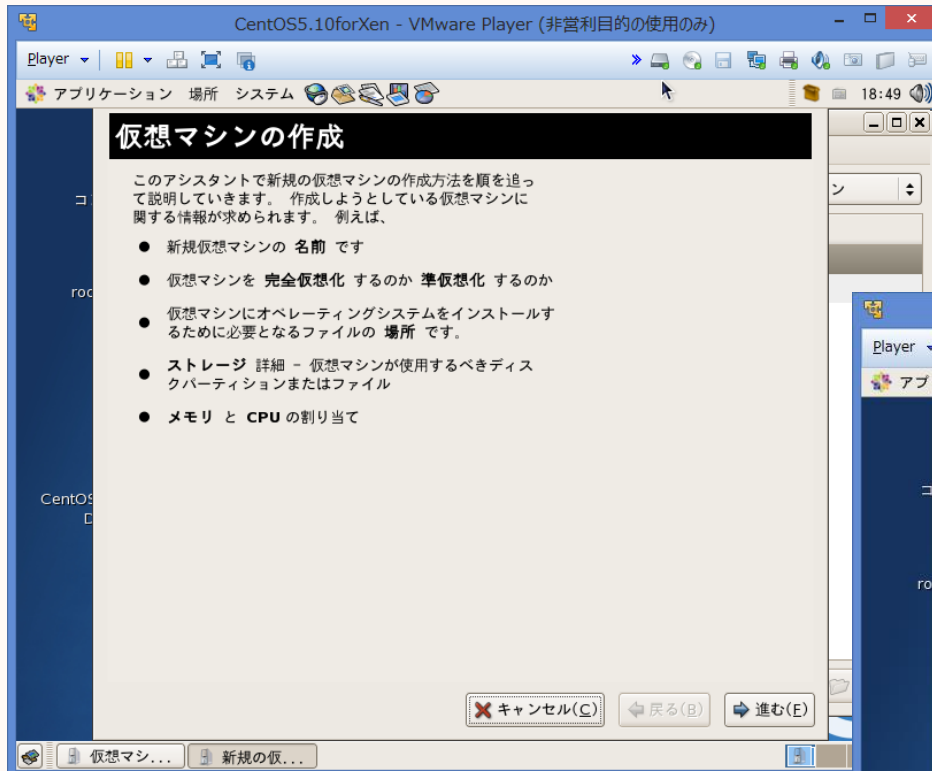


Domain-U(ゲストOS)の作成
仮想マシンマネージャより
「新規」をクリック



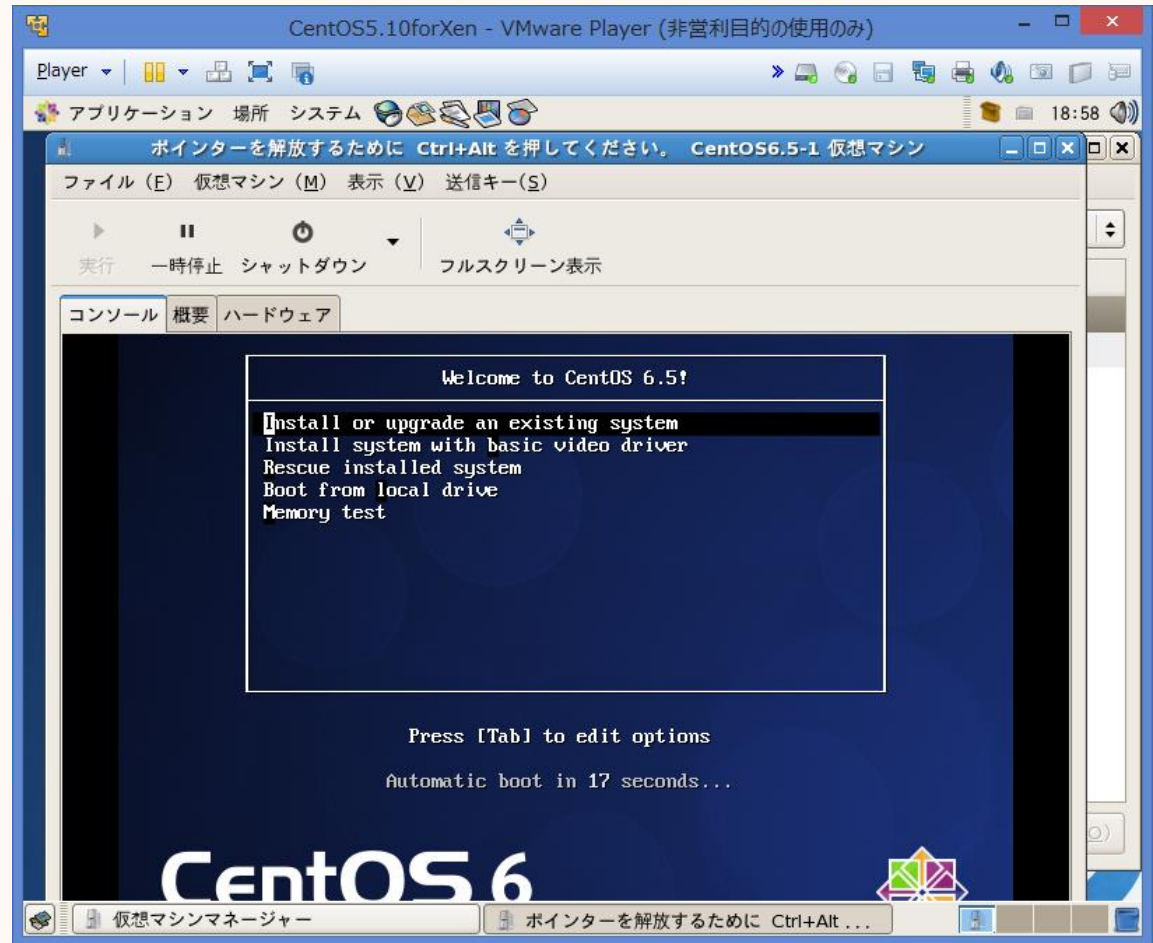


Domain-U(ゲストOS)の作成



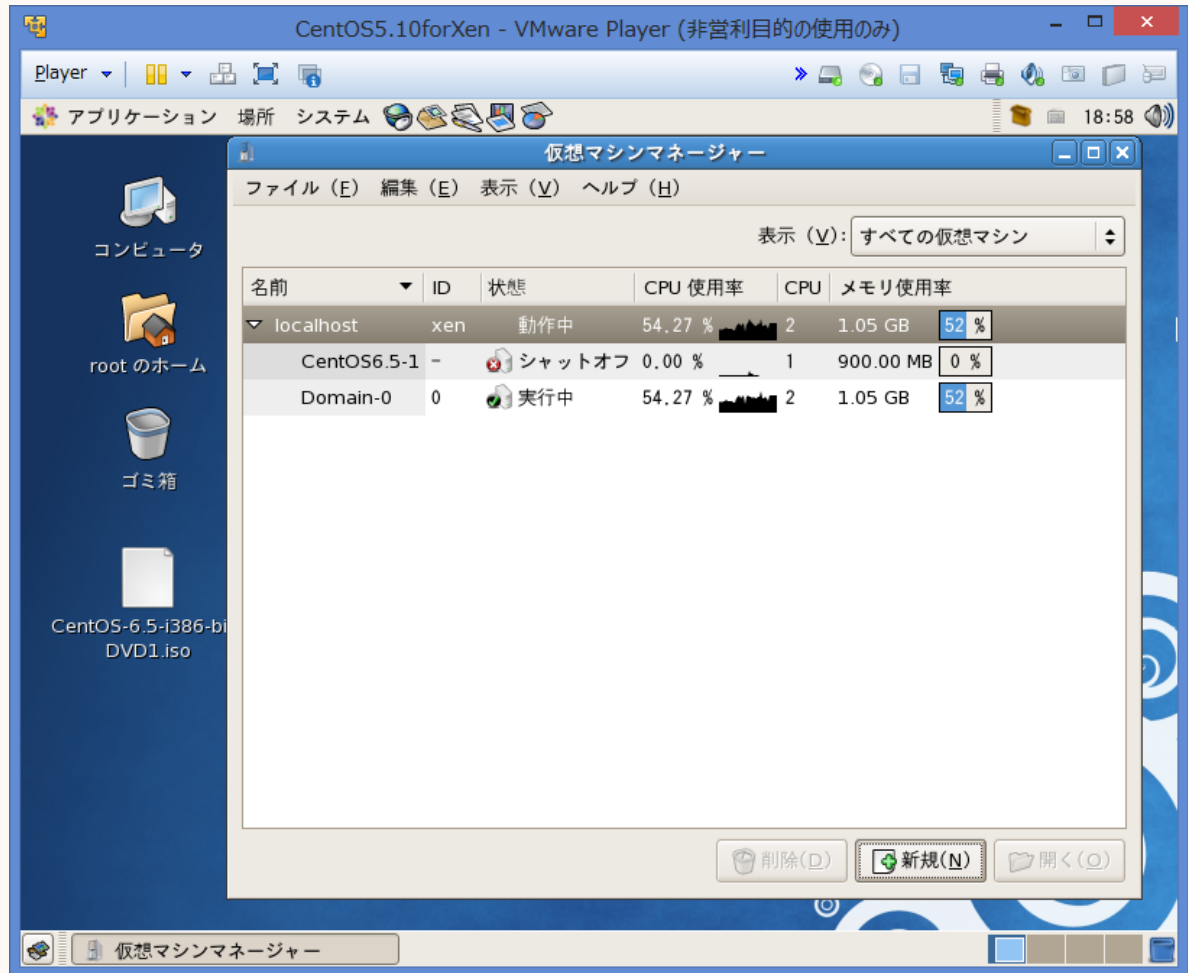


Domain-U(ゲストOS)の起動





Domain-U(ゲストOS)の起動





Xenの設定

Xenの設定は、/etc/xenディレクトリ配下にあります

```
[root@localhost ~]# cd /etc/xen
[root@localhost xen]# ls -l
合計 108
-rw----- 1 root root 539 7月 10 18:57 CentOS6.5-1
drwxr-xr-x 2 root root 4096 5月 21 2013 auto
-rwxr-xr-x 1 root root 482 5月 21 2013 qemu-ifup
drwxr-xr-x 2 root root 4096 7月 11 01:02 scripts
-rw-r--r-- 1 root root 7206 5月 21 2013 xend-config.sxp
-rw-r--r-- 1 root root 1256 5月 21 2013 xend-pci-permissive.sxp
-rw-r--r-- 1 root root 4129 5月 21 2013 xend-pci-quirks.sxp
-rw-r--r-- 1 root root 7264 5月 21 2013 xmexample.hvm
-rw-r--r-- 1 root root 5315 5月 21 2013 xmexample.vti
-rw-r--r-- 1 root root 5463 5月 21 2013 xmexample1
-rw-r--r-- 1 root root 6746 5月 21 2013 xmexample2
```



Xenの設定

xendの設定ファイルは、xend-config.sxpです

```
[root@localhost xen]# cat xend-config.sxp | grep '"^('
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-relocation-hosts-allow '^localhost$ ^localhost¥¥.localdomain$')
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 256)
(dom0-cpus 0)
(vncpasswd '')
(keymap 'en-us')
```



Xenの設定

仮想マシン設定はそれぞれの名前の付いたファイルが作成されます

```
[root@localhost xen]# cat CentOS6.5-1
name = "CentOS6.5-1"
uuid = "28bc59a0-5212-6230-0520-3d74e9fa0ee0"
maxmem = 900
memory = 900
vcpus = 1
builder = "hvm"
kernel = "/usr/lib/xen/boot/hvmloader"
boot = "c"
pae = 1
acpi = 1
apic = 1
localtime = 0
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
device_model = "/usr/lib64/xen/bin/qemu-dm"
sdl = 0
vnc = 1
vncunused = 1
keymap = "ja"
disk = [ "file:/var/lib/xen/images/CentOS6.5-1.img,hda,w", ",hdc:cdrom,r" ]
vif = [ "mac=00:16:3e:0a:82:21,bridge=virbr0,script=vif-bridge" ]
parallel = "none"
serial = "pty"
```



Xenのユーティリティ

xm list(稼働中のドメインの一覧表示)

※ドメインとは、仮想マシン名の事

xm create ドメイン(ドメインの起動)

xm uptime(ドメインの稼働時間表示)

```
[root@localhost xen]# xm list
Name                               ID Mem(MiB) VCPUs State   Time(s)
Domain-0                           0   1074     2 r----- 145.0
[root@localhost xen]# xm create CentOS6.5-1
Using config file "/CentOS6.5-1".
Started domain CentOS6.5-1
[root@localhost xen]# xm list
Name                               ID Mem(MiB) VCPUs State   Time(s)
CentOS6.5-1                        5    908     1 r-----   1.6
Domain-0                           0   1074     2 r----- 147.1
[root@localhost xen]# xm uptime
Name                               ID Uptime
CentOS6.5-1                        5 0:00:30
Domain-0                           0 0:41:12
```



Xenのユーティリティ

xm info

(Xenの環境情報表示)

```
[root@localhost xen]# xm info
host                : localhost.localdomain
release             : 2.6.18-371.el5xen
version             : #1 SMP Tue Oct 1 09:15:30 EDT 2013
machine             : x86_64
nr_cpus              : 2
nr_nodes             : 1
sockets_per_node     : 2
cores_per_socket     : 1
threads_per_core     : 1
cpu_mhz              : 3191
hw_caps              : 0fabfbff:28100800:00000000:00000940:f6ba2223:00000000:0
00000001
total_memory         : 2047
free_memory          : 1
node_to_cpu          : node0:0-1
xen_major             : 3
xen_minor            : 1
xen_extra             : .2-371.el5
xen_caps              : xen-3.0-x86_64 xen-3.0-x86_32p hvm-3.0-x86_32 hvm-3.0-x
86_32p hvm-3.0-x86_64
xen_pagesize          : 4096
platform_params       : virt_start=0xffff800000000000
xen_changeset         : unavailable
cc_compiler           : gcc version 4.1.2 20080704 (Red Hat 4.1.2-54)
cc_compile_by         : mockbuild
cc_compile_domain     : (none)
cc_compile_date       : Tue Oct 1 08:26:28 EDT 2013
xend_config_format    : 2
```



Xenのユーティリティ

xm top(稼働中のドメイン情報をtopコマンド風に表示)

※xentopコマンドでも同様の事は可能です

```
xentop - 21:48:56 Xen 3.1.2-371.el5
2 domains: 2 running, 0 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 2096636k total, 2094692k used, 1944k free CPUs: 2 @ 3191MHz
```

NAME	STATE	CPU(sec)	CPU(%)	MEM(k)	MEM(%)	MAXMEM(k)	MAXMEM(%)	VCPUS	NETS	NETTX(k)	NETRX(k)	VBDS	VBD_OO	VBD_RD	VBD_WR	SSID
CentOS6.5-1	-----r	278	98.1	929664	44.3	937984	44.7	1	1	0	0	1	0	0	0	0
Domain-0	-----r	159	4.7	1099776	52.5	no limit	n/a	2	4	300	107	0	0	0	0	0



Xenのユーティリティ
xm shutdown
(ドメインの終了)
xm destroy
(ドメインの強制終了)
xm reboot
(ドメインの再起動)

```
[root@localhost xen]# xm list
Name                                ID Mem(MiB) VCPUs State   Time(s)
CentOS6.5-1                         5    908      1 r----- 347.8
Domain-0                           0   1074      2 r----- 163.0
[root@localhost xen]# xm shutdown CentOS6.5-1
[root@localhost xen]# xm list
Name                                ID Mem(MiB) VCPUs State   Time(s)
Domain-0                           0   1074      2 r----- 165.1
[root@localhost xen]#
[root@localhost xen]# xm create CentOS6.5-1
Using config file "/CentOS6.5-1".
Started domain CentOS6.5-1
[root@localhost xen]# xm list
Name                                ID Mem(MiB) VCPUs State   Time(s)
CentOS6.5-1                         6    908      1 r-----  3.1
Domain-0                           0   1074      2 r----- 167.5
[root@localhost xen]#
[root@localhost xen]# xm destroy CentOS6.5-1
[root@localhost xen]# xm list
Name                                ID Mem(MiB) VCPUs State   Time(s)
Domain-0                           0   1074      2 r----- 168.9
[root@localhost xen]#
[root@localhost xen]#
[root@localhost xen]# xm create CentOS6.5-1
Using config file "/CentOS6.5-1".
Started domain CentOS6.5-1
[root@localhost xen]# xm list
Name                                ID Mem(MiB) VCPUs State   Time(s)
CentOS6.5-1                         7    908      1 r-----  1.2
Domain-0                           0   1074      2 r----- 171.0
[root@localhost xen]# xm reboot CentOS6.5-1
```



Xenのユーティリティ

xm pause(ドメインの一時停止)

xm unpause(一時停止中のドメインの再開)

```
[root@localhost xen]# xm list
```

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
CentOS6.5-1	8	908	1	r-----	5.4
Domain-0	0	1074	2	r-----	175.9

```
[root@localhost xen]# xm pause CentOS6.5-1
```

```
[root@localhost xen]# xm list
```

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
CentOS6.5-1	8	908	1	--p---	15.0
Domain-0	0	1074	2	r-----	176.8

```
[root@localhost xen]#
```

```
[root@localhost xen]# xm unpause CentOS6.5-1
```

```
[root@localhost xen]# xm list
```

Name	ID	Mem(MiB)	VCPUs	State	Time(s)
CentOS6.5-1	8	908	1	r-----	18.3
Domain-0	0	1074	2	r-----	177.7



Xenのユーティリティ

他のコマンドについても、`xm help`(利用可能コマンド表示)で確認出来ます

```
[root@localhost ~]# xm help
Usage: xm <subcommand> [args]

Control, list, and manipulate Xen guest instances.

xm full list of subcommands:

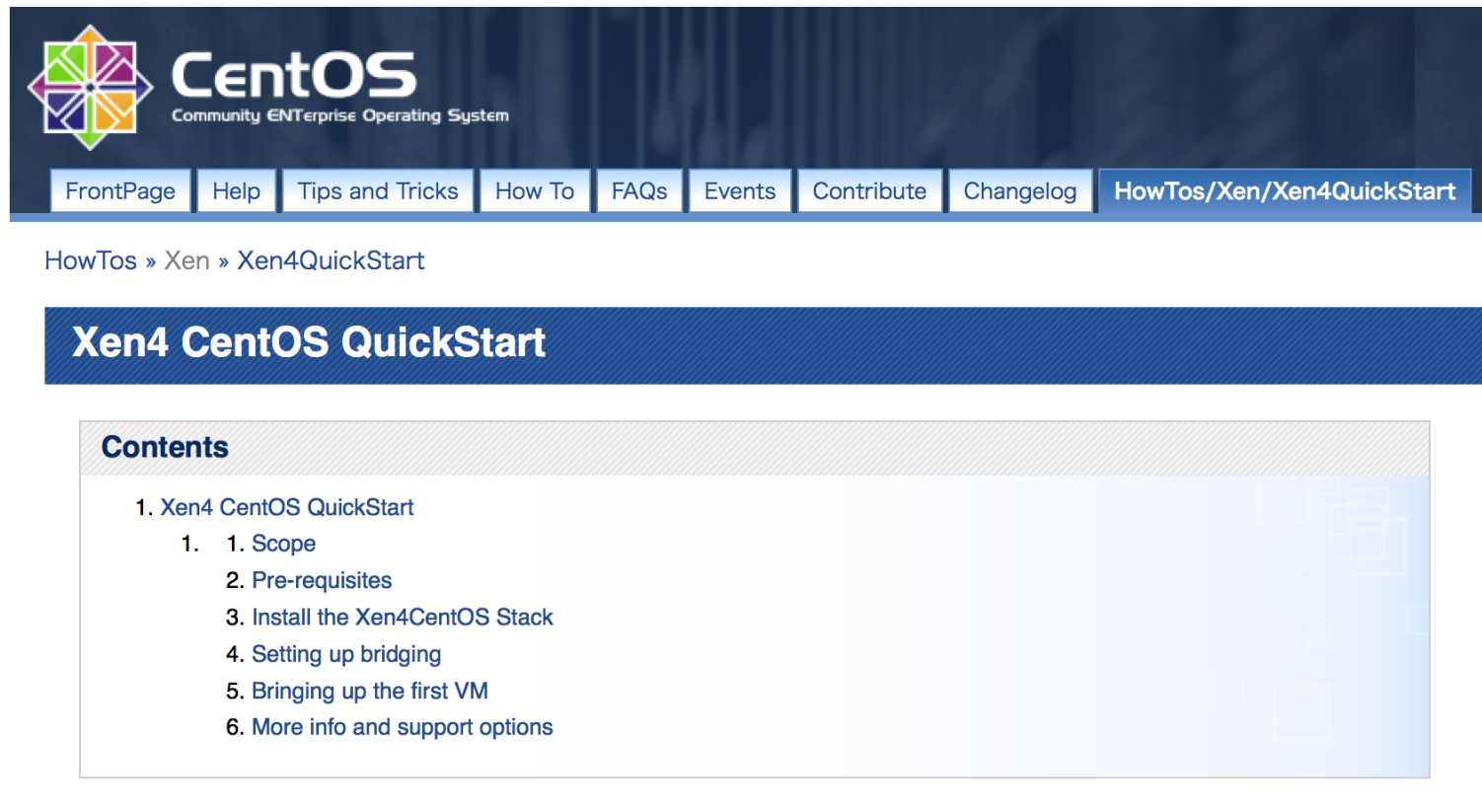
console      Attach to <Domain>'s console.
create       Create a domain based on <ConfigFile>.
destroy      Terminate a domain immediately.
domid        Convert a domain name to domain id.
domname      Convert a domain id to domain name.
dump-core    Dump core for a specific domain.
list         List information about all/some domains.
mem-max      Set the maximum amount reservation for a domain.
mem-set      Set the current memory usage for a domain.
migrate      Migrate a domain to another machine.
pause        Pause execution of a domain.
reboot       Reboot a domain.
rename       Rename a domain.
restore      Restore a domain from a saved state.
save         Save a domain state to restore later.
shutdown    Shutdown a domain.
sysrq        Send a sysrq to a domain.
trigger      Send a trigger to a domain.
top          Monitor a host and the domains in real time.
```



Xen on CentOS5以外にも

Xen on Centos6

(<https://wiki.centos.org/HowTos/Xen/Xen4QuickStart>)や



CentOS
Community ENTERprise Operating System

FrontPage | Help | Tips and Tricks | How To | FAQs | Events | Contribute | Changelog | **HowTos/Xen/Xen4QuickStart**

HowTos » Xen » Xen4QuickStart

Xen4 CentOS QuickStart

Contents

1. Xen4 CentOS QuickStart
 1. Scope
 2. Pre-requisites
 3. Install the Xen4CentOS Stack
 4. Setting up bridging
 5. Bringing up the first VM
 6. More info and support options



XenServer(<http://xenserver.org/>)を使って学習出来ます。
※xlコマンドを使います。

The screenshot shows the XenServer website's 'SOFTWARE' section, specifically the 'DOWNLOAD' page. The page title is 'DOWNLOAD - XENSERVER'. Below the title, it says 'SUPPORTED RELEASES'. A paragraph explains that having a supported platform is crucial for mission critical applications and that using the installers in this section qualifies for support services. The section 'DOWNLOAD XENSERVER 6.5 AND XENSERVER 6.5 SP1' states that XenServer 6.5 was released in January of 2015 and was available in pre-release form under the project name of Creedence. All new XenServer installations should be made using XenServer 6.5. Under the heading 'INSTALLERS [REQUIRED FOR BOTH XENSERVER 6.5 AND 6.5 SP1]', there is a list of three items: 'XenServer Installation ISO', 'XenCenter Windows Management Console (English)', and 'XenCenter Windows Management Console (Localized - Japanese and Simplified Chinese)'.

XenServer®
Open Source Virtualization

HOME SOFTWARE SUPPORT COMMUNITY ABOUT LOGIN/REGISTER

Home / Software / Download

Blog | Code | Wiki | Download

DOWNLOAD - XENSERVER

SUPPORTED RELEASES

Having a supported platform is crucial for mission critical applications, and for those who are interested in obtaining [product support for XenServer from Citrix](#), use of the installers in this section will qualify for support services.

DOWNLOAD XENSERVER 6.5 AND XENSERVER 6.5 SP1

XenServer 6.5 was released in January of 2015. It was available in pre-release form under the project name of Creedence. All new XenServer installations should be made using XenServer 6.5.

INSTALLERS [REQUIRED FOR BOTH XENSERVER 6.5 AND 6.5 SP1]

- [XenServer Installation ISO](#)
- [XenCenter Windows Management Console \(English\)](#)
- [XenCenter Windows Management Console \(Localized - Japanese and Simplified Chinese\)](#)



330.3 KVM



仮想化ソフトウェアにて、CPU仮想化支援機能を有効化した仮想PCを作成します。

仮想マシン設定

ハードウェア オプション

デバイス	概要
メモリ	4 GB
プロセッサ	2
ハード ディスク(SCSI)	50 GB
CD/DVD(IDE)	C:\Users\root\Desktop\ISOs\Cent...
フロッピー	自動検出
ネットワーク アダプタ	ブリッジ(自動)
USB コントローラ	あり
サウンド カード	自動検出
プリンタ	あり
ディスプレイ	自動検出

追加 (A)... 削除 (R)

プロセッサ
プロセッサ コアの数 (C): 2

仮想化エンジン
優先モード (M): Intel VT-x/EPTまたは AMD-V/RVI
☐ バイナリ変換のアクセラレーションを無効にする (D)
☒ Intel VT-x/EPTまたは AMD-V/RVI を仮想化 (V)
☒ CPU パフォーマンス ガガントを仮想化 (U)
⚠ 仮想化エンジン モードを手動で指定すると、一定のワークロード下でのパフォーマンスに影響を与える可能性があります。

CentOS6.5forKVM2

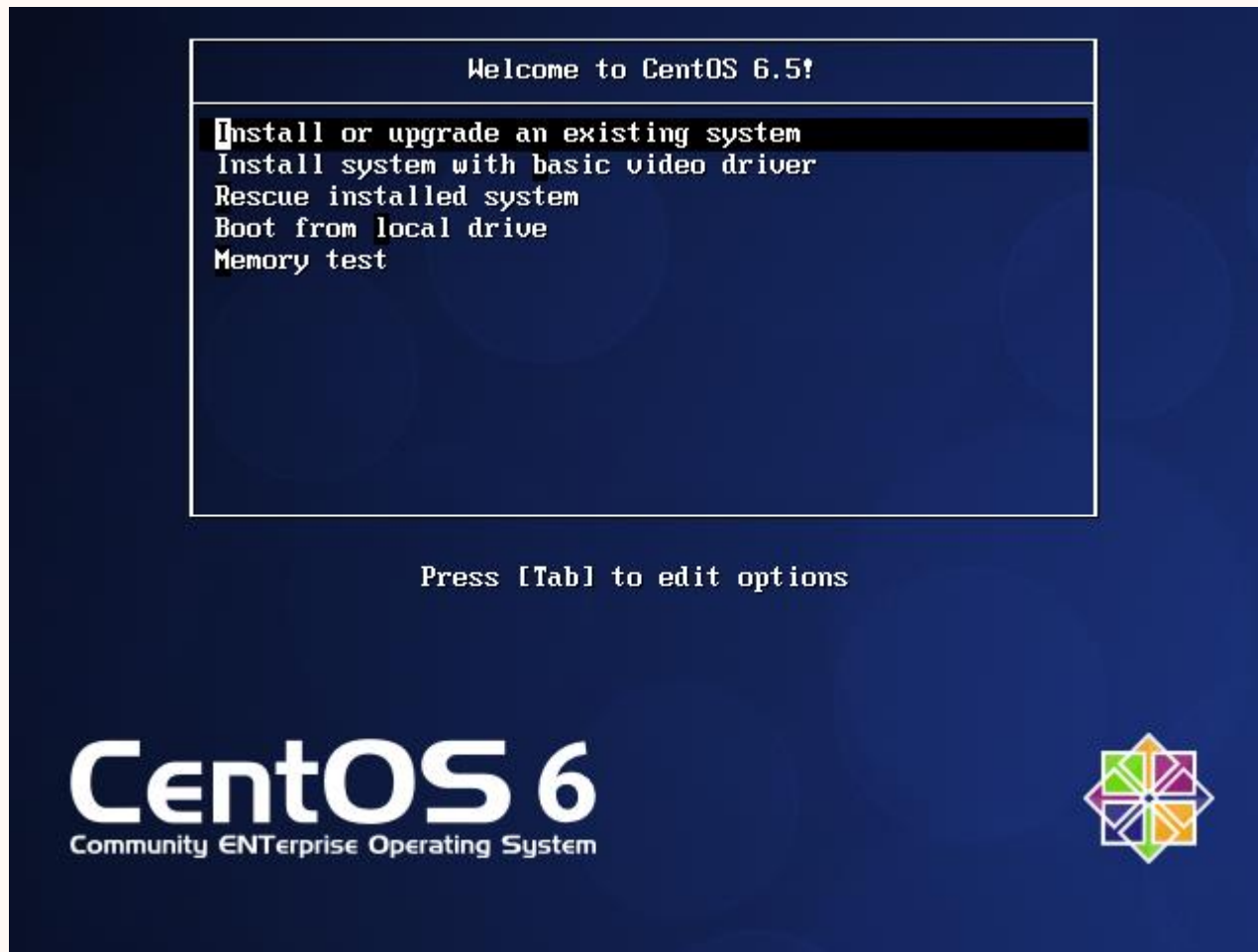
状態: パワーオフ
OS: CentOS 64 ビット
バージョン: Workstation 9.0 仮想マシン
RAM: 4 GB

▶ 仮想マシンの再生 (L)
🔧 仮想マシン設定の編集 (D)

OK キャンセル ヘルプ

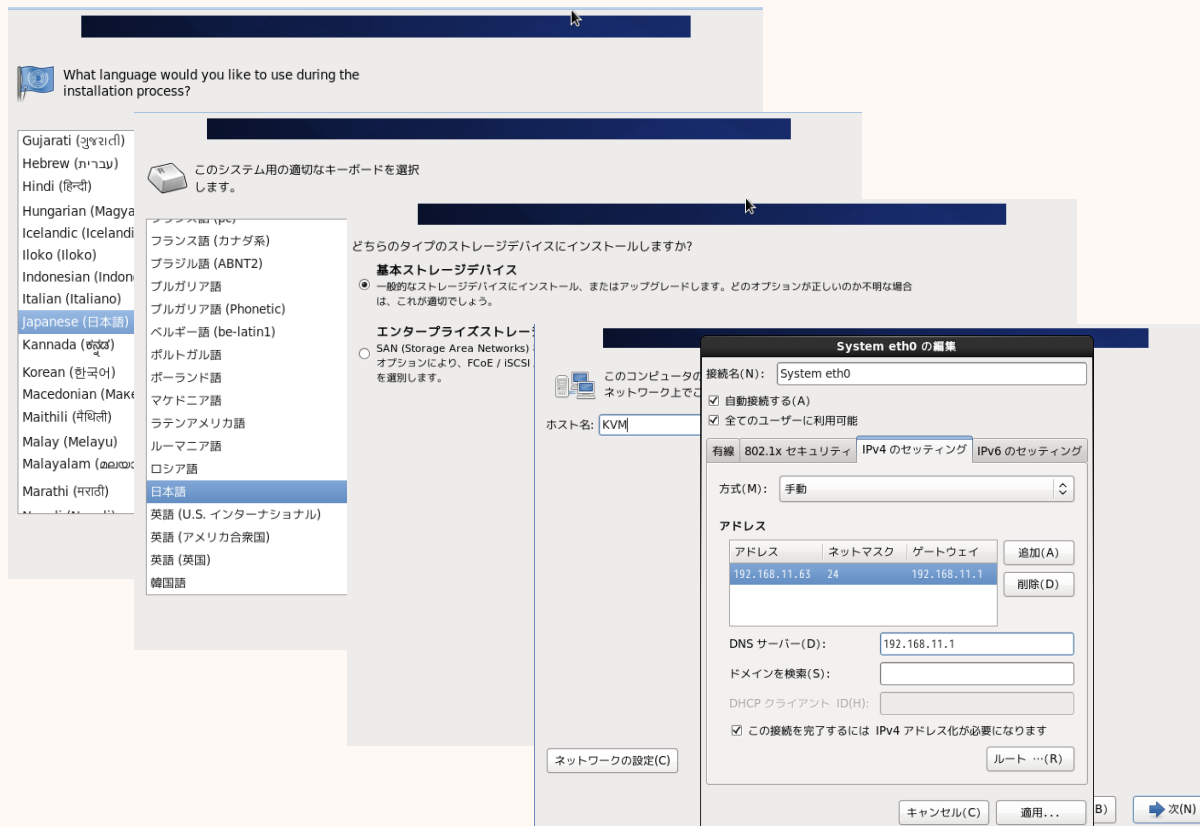


KVMのインストール(CentOS6.7 64bitsを使用)



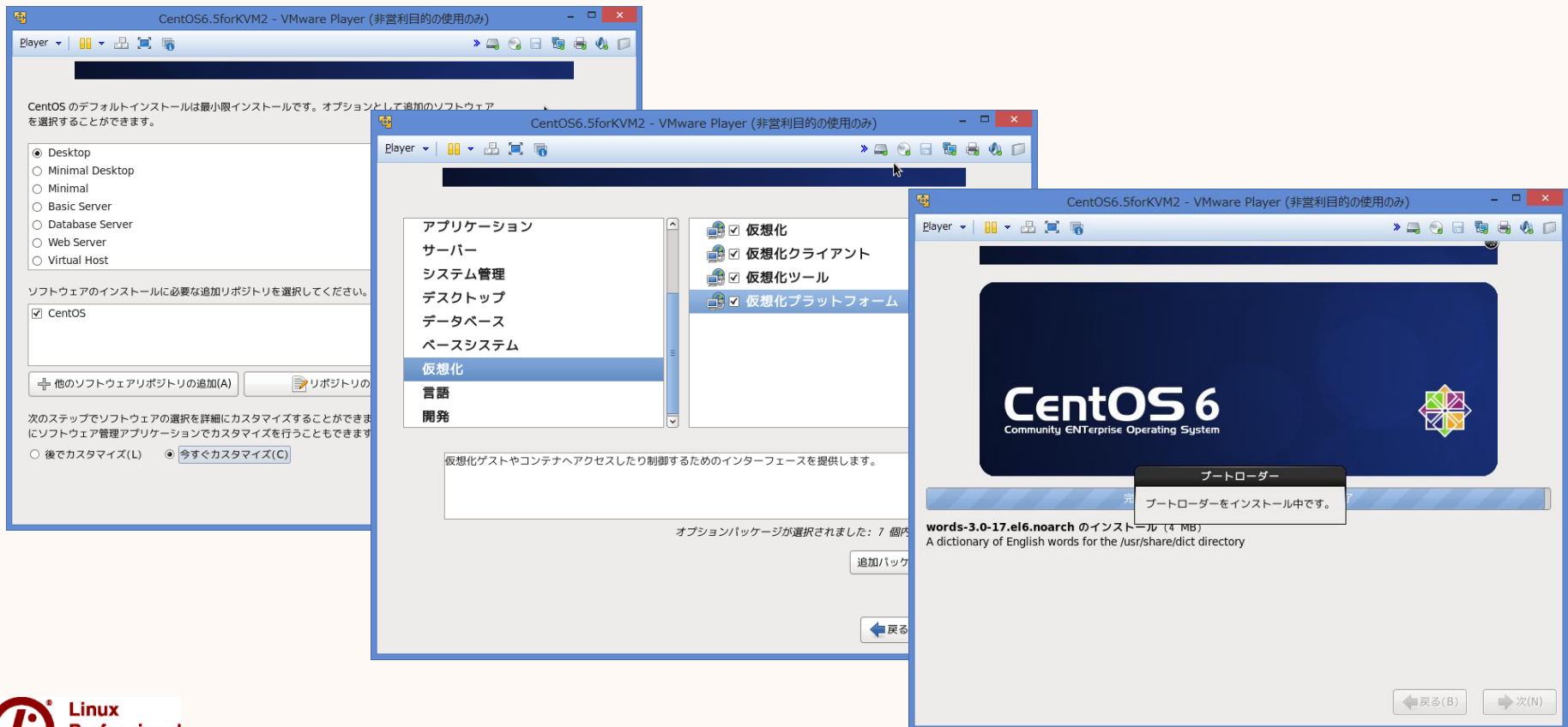


ソフトウェアの選択までは、自身の環境等に合わせて設定を行います。





ソフトウェアの選択では、「今すぐカスタマイズ」から、
仮想化 > 仮想化、仮想化クライアント、仮想化ツール、仮想化プラットフォームに
チェックを入れます。





起動後、コンソールからCPUの仮想化機能対応を確認します。(vmx or svm)

cat /proc/cpuinfo

```
[root@KVM ~]# cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 58
model name     : Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz
stepping       : 9
cpu MHz        : 3192.743
cache size     : 6144 KB
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clfl
ush dts mmx fxsr sse sse2 ss syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts xtopology
tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq vmx sse3 cx16 pcid sse4_1
sse4_2 x2apic popcnt aes xsave avx f16c rdrand hypervisor lahf_lm ida arat epb xsaveopt pln pt
s dts tpr_shadow vnmi ept vpid fsgsbase smep
bogomips       : 6385.48
clflush size   : 64
cache_alignme  : 64
address sizes   : 40 bits physical, 48 bits virtual
power management:
```



また、kvm用のモジュールがカーネルでロードされているか確認します。

`lsmod | grep kvm`

なお、Intel製CPUの場合、kvmとkvm_intel

AMD製CPUの場合、kvmとkvm_amdとなります。

```
[root@KVM ~]# lsmod | grep kvm
kvm_intel          54285      3
kvm                333172     1 kvm_intel
[root@KVM ~]#
```



qemu-kvm(KVMコマンドライン)、
libvirt、bridge-utils(ブリッジ管理)、
python-virtinst(virt-installなどのツール)のインストールを確認。

```
root@KVM:~  
ファイル(F) 編集(E) 表示(V) 検索 (S) 端末(T) ヘルプ(H)  
[root@KVM ~]# rpm -q qemu-kvm libvirt bridge-utils python-virtinst  
qemu-kvm-0.12.1.2-2.415.el6.x86_64  
libvirt-0.10.2-29.el6.x86_64  
bridge-utils-1.2-10.el6.x86_64  
python-virtinst-0.600.0-18.el6.noarch  
[root@KVM ~]#
```



KVMの設定

仮想マシンをブリッジ接続する為に、ifcfg-br0を作成します。

```
cd /etc/sysconfig/network-scripts
```

```
cp -p ifcfg-eth0 ifcfg-br0
```

```
[root@KVM ~]# cd /etc/sysconfig/network-scripts/  
[root@KVM network-scripts]# cp -p ifcfg-eth0 ifcfg-br0
```

ifcfg-eth0の編集

NM_CONTROLLED=no

BRIDGE=br0

```
[root@KVM network-scripts]# vi ifcfg-eth0  
[root@KVM network-scripts]# cat ifcfg-eth0  
DEVICE=eth0  
TYPE=Ethernet  
UUID=b8dc0ffd-9bc4-490f-ab95-b6285d8c653b  
ONBOOT=yes  
NM_CONTROLLED=no  
BOOTPROTO=none  
HWADDR=00:0C:29:5C:61:8A  
IPADDR=192.168.11.63  
PREFIX=24  
GATEWAY=192.168.11.1  
DNS1=192.168.11.1  
DEFROUTE=yes  
IPV4_FAILURE_FATAL=yes  
IPV6INIT=no  
NAME="System eth0"  
BRIDGE=br0
```



ifcfg-br0の編集

DEVICE=br0

TYPE=Bridge

NM_CONTROLLED=no

#NAME="System eth0"

```
[root@KVM network-scripts]# vi ifcfg-br0
[root@KVM network-scripts]# cat ifcfg-br0
DEVICE=br0
TYPE=Bridge
UUID=b8dc0ffd-9bc4-490f-ab95-b6285d8c653b
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=none
HWADDR=00:0C:29:5C:61:8A
IPADDR=192.168.11.63
PREFIX=24
GATEWAY=192.168.11.1
DNS1=192.168.11.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
#NAME="System eth0"
```



ネットワークサービスを再起動します。

/etc/init.d/network restart

```
[root@KVM network-scripts]# /etc/init.d/network restart
インターフェース br0 を終了中: [ OK ]
ループバックインターフェースを終了中 [ OK ]
ループバックインターフェースを呼び込み中 [ OK ]
インターフェース eth0 を活性化中: device eth0 is already a member of a bridge; can't enslave it to bridge br0.
[ OK ]
インターフェース br0 を活性化中: Determining if ip address 192.168.11.63 is already in use for device br0...
[ OK ]
```




ifconfigで確認

```
[root@KVM network-scripts]# ifconfig
br0      Link encap:Ethernet  HWaddr 00:0C:29:5C:61:8A
         inet addr:192.168.11.63  Bcast:192.168.11.255  Mask:255.255.255.0
         inet6 addr: fe80::20c:29ff:fe5c:618a/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:4219 errors:0 dropped:0 overruns:0 frame:0
         TX packets:1616 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:3182451 (3.0 MiB)  TX bytes:112430 (109.7 KiB)

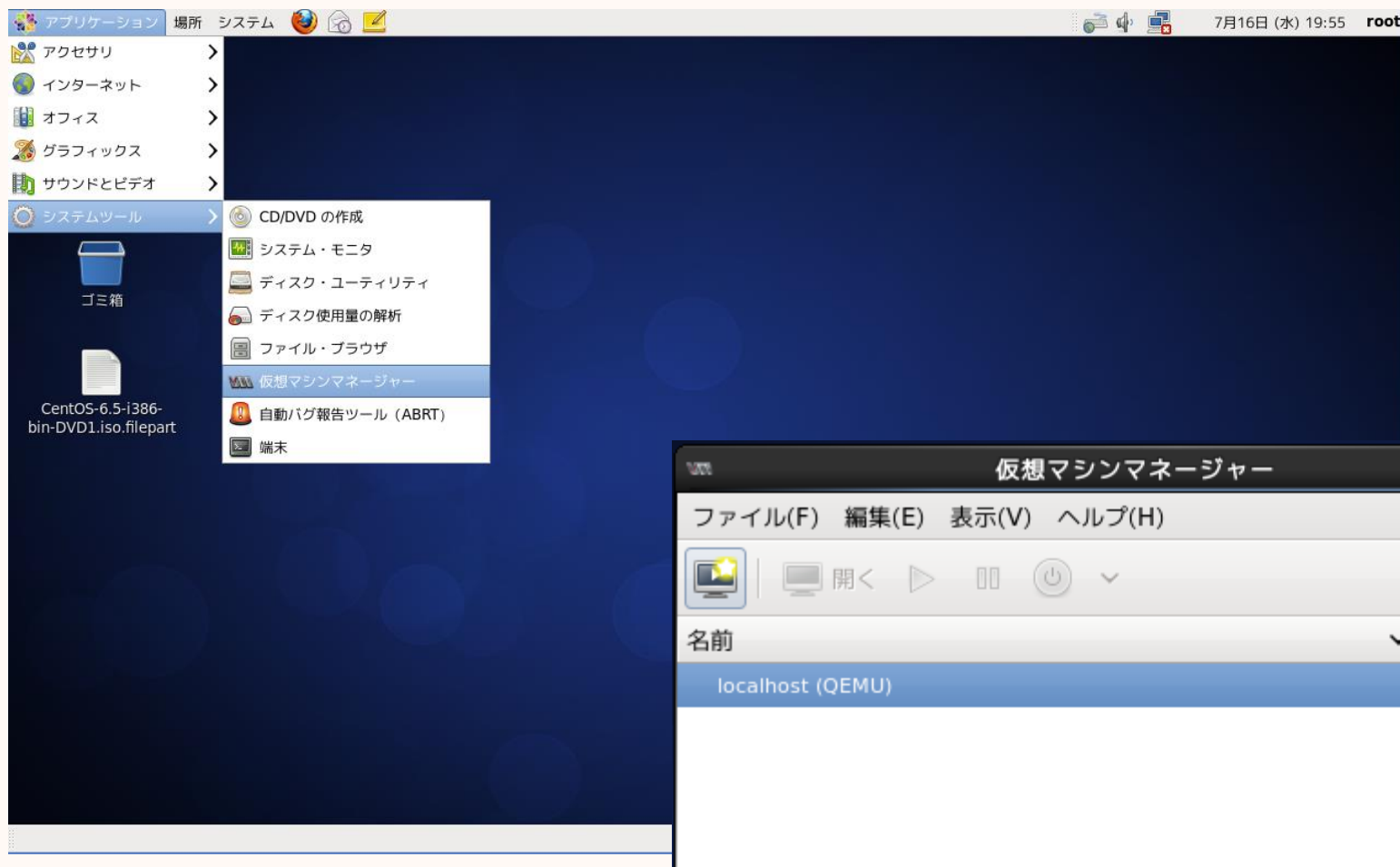
eth0     Link encap:Ethernet  HWaddr 00:0C:29:5C:61:8A
         inet6 addr: fe80::20c:29ff:fe5c:618a/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:5002622 errors:0 dropped:0 overruns:0 frame:0
         TX packets:846083 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:4136527211 (3.8 GiB)  TX bytes:54901858 (52.3 MiB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:16 errors:0 dropped:0 overruns:0 frame:0
         TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:960 (960.0 b)  TX bytes:960 (960.0 b)

virbr0   Link encap:Ethernet  HWaddr 52:54:00:E6:3B:05
         inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```



(仮想マシンマネージャによるゲストOS作成)





LPIC 304 技術解説セミナー 330.3

The image displays three overlapping screenshots of the Virtual Machine Manager (VMM) interface, illustrating the steps to create a new virtual machine.

Left Screenshot (Step 1/5): The "新しい仮想マシン" (New Virtual Machine) window shows the initial setup. The name is "CentOS6.5" and the connection is "localhost (QEMU/KVM)". The OS is selected as "CentOS 6.5".

Middle Screenshot (Step 2/5): The "新しい仮想マシン" window shows the selection of the installation media. The "ISO イメージを使用(I):" (Use ISO image) option is selected, and the path "/root/デスクトップ/CentOS-6.5-i386-bin-DVD1.iso" is entered. The OS type is "Linux" and the version is "Generic 2.6.x kernel".

Right Screenshot (Step 5/5): The "新しい仮想マシン" window shows the final configuration. The OS is "CentOS 6.5". The installation is set to "ローカル CDROM/ISO" (Local CDROM/ISO). The memory is "1024 MB", the CPU is "1", and the storage is "8.0 GB /var/lib/libvirt/images/CentOS6.5.img". The "インストールの前に設定をカスタマイズする(U)" (Customize settings before installation) checkbox is unchecked. The "詳細なオプション" (Advanced Options) section shows the host device as "eth0 (Bridge 'br0')", the fixed MAC address is "52:54:00:da:5a:1c", the virtualization type is "kvm", and the architecture is "i686".

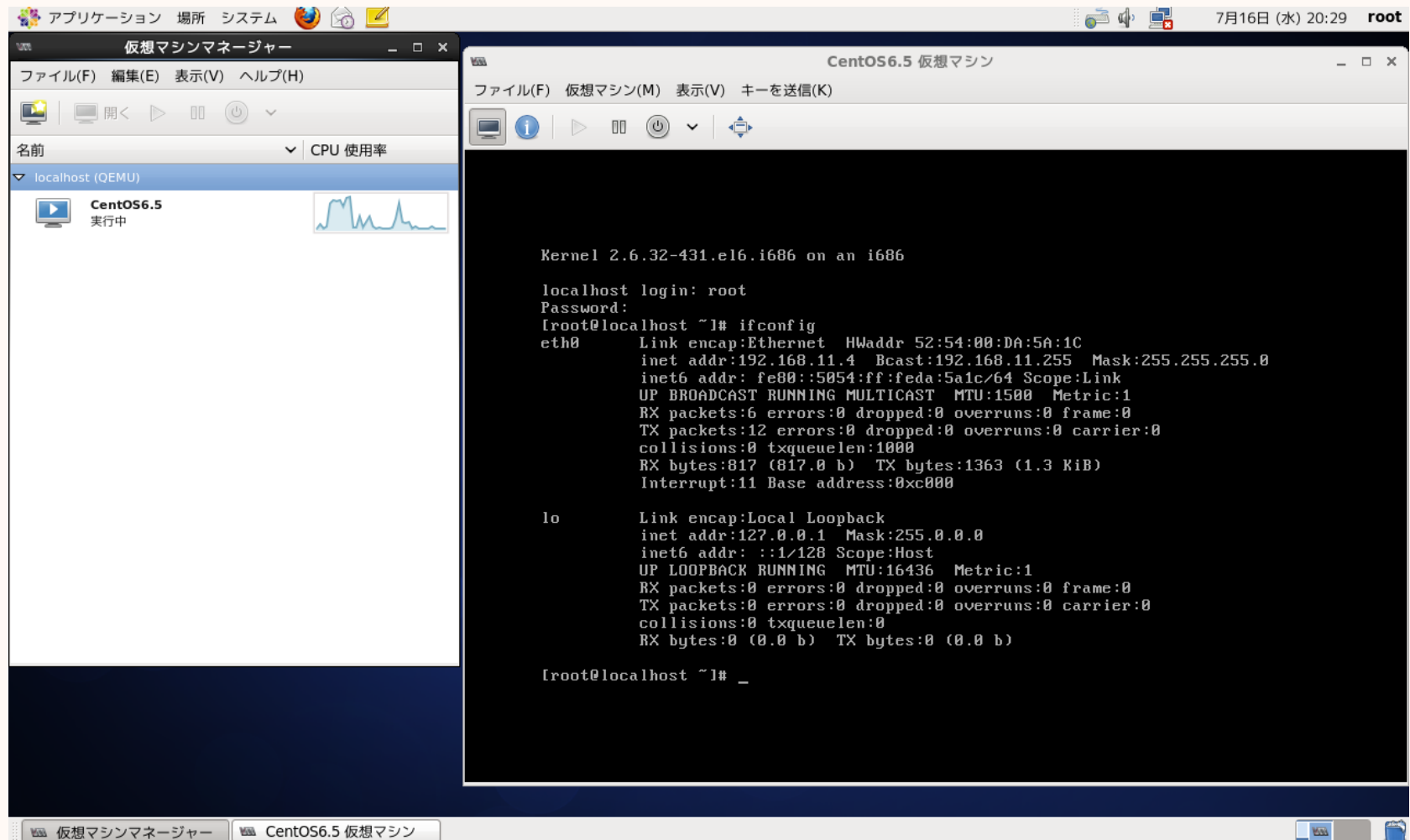


(仮想マシンの起動)





KVMのユーティリティ(仮想マシンマネージャ)





(virshコマンド)

virsh version

(virshのバージョン確認)

virsh list

(稼働中マシンリスト表示)

virsh dominfo

(仮想マシン情報表示)

virsh domstate

(仮想マシン状態表示)

```
[root@KVM ~]# virsh version
コンパイル時に使用したライブラリ: libvirt 0.10.2
使用中のライブラリ: libvirt 0.10.2
使用中の API: QEMU 0.10.2
実行中のハイパーバイザー: QEMU 0.12.1

[root@KVM ~]# virsh list
  Id      名前                                     状態
-----
  2       CentOS6.5                               実行中

[root@KVM ~]# virsh dominfo CentOS6.5
Id:                2
名前:              CentOS6.5
UUID:              57aae801-ff60-6bb3-03b6-9d10b2fb77ce
OS タイプ:         hvm
状態:              実行中
CPU:               1
CPU 時間:          124.3s
最大メモリー:     1048576 KiB
使用メモリー:     1048576 KiB
永続:              はい (yes)
自動起動:          無効にする
管理済み保存:      いいえ (no)
セキュリティモデル: selinux
セキュリティ DOI:  0
セキュリティラベル: system_u:system_r:svirt_t:s0:c211,c928 (enforcing)

[root@KVM ~]# virsh domstate CentOS6.5
実行中
```



virsh console

(仮想マシンへの接続)

virsh shutdown

(仮想マシンの終了)

virsh start

(仮想マシンの開始)

virsh reboot

(仮想マシンの再起動)

virsh suspend

(仮想マシンの一時停止)

virsh resume

(仮想マシンの再開)

virsh destroy

(仮想マシンの強制終了)

```
[root@KVM ~]# virsh console 2
ドメイン CentOS6.5 に接続しました
エスケープ文字は ^] です

[root@KVM ~]# virsh shutdown CentOS6.5
ドメイン CentOS6.5 をシャットダウンしています

[root@KVM ~]# virsh start CentOS6.5
ドメイン CentOS6.5 が起動されました

[root@KVM ~]# virsh reboot CentOS6.5
ドメイン CentOS6.5 を再起動しています

[root@KVM ~]# virsh suspend CentOS6.5
ドメイン CentOS6.5 は一時停止されました

[root@KVM ~]# virsh resume CentOS6.5
ドメイン CentOS6.5 が再開されました

[root@KVM ~]# virsh destroy CentOS6.5
ドメイン CentOS6.5 は強制停止されました
```



他のコマンドについても、
virsh helpで確認出来ます。

```
[root@KVM ~]# virsh help
グループ別コマンド:

Domain Management (ヘルプのキーワード 'domain'):
```

attach-device	XML ファイルによるデバイスの接続
attach-disk	ディスクデバイスの接続
attach-interface	ネットワークインターフェースの接続
autostart	ドメインの自動起動
blkdeviotune	ブロックデバイスの I/O チューニングパラメーターの設定または取得
blkiotune	ブロック I/O パラメーターの取得または設定
blockcommit	ブロックのコミット操作の開始
blockcopy	ブロックコピー操作の開始
blockjob	アクティブなブロック操作の管理
blockpull	既存のイメージからのディスクの配置
blockresize	ドメインのブロックデバイスの容量変更
change-media	CD またはフロッピードライブのメディアの変更
console	ゲストのコンソールへの接続
cpu-baseline	ベースライン CPU の計算
cpu-compare	XML ファイルに記載された CPU とホスト CPU の比較
cpu-stats	ドメインの CPU 統計情報の表示
create	XML ファイルによるドメインの作成
define	XML ファイルによるドメインの定義 (起動なし)
desc	ドメインの説明またはタイトルの表示または設定
destroy	ドメインの強制停止
detach-device	XML ファイルによるデバイスの切断
detach-disk	ディスクデバイスの切断
detach-interface	ネットワークインターフェースの切断
domdisplay	ドメインのディスプレイ接続 URI
domhostname	ドメインのホスト名の表示
domid	ドメイン名または UUID からドメイン ID への変換



ところで、XenやKVMなどのハイパーバイザーは
どんな場面で利用されているんだろう？

検証環境や物理サーバからの移行先？？



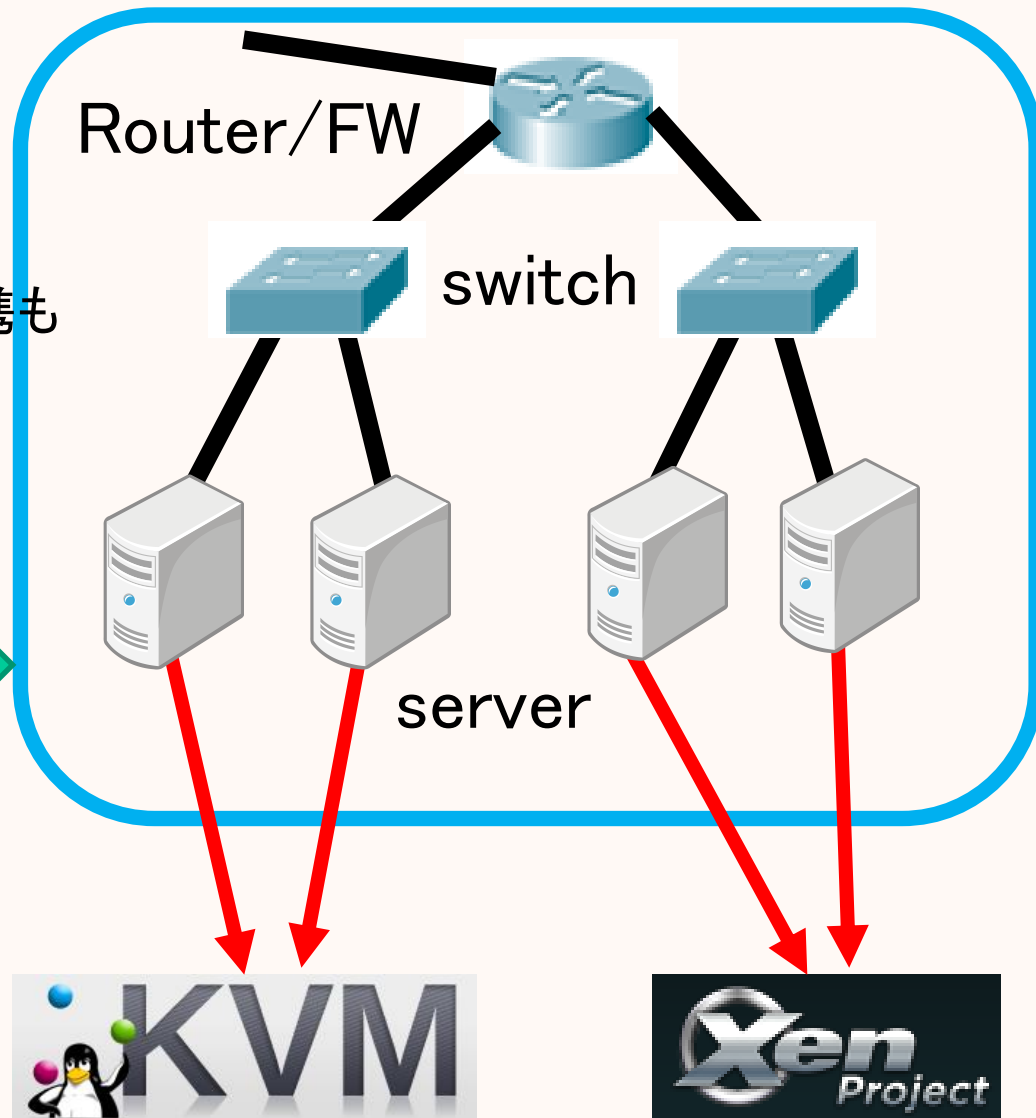
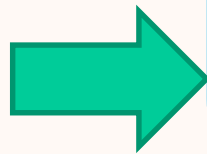
それ以外にも、
クラウド基盤ソフトウェア(IaaS)の
サーバ格納先などとして
利用する事が出来ます。
※その他、SDNコントローラーとの連携も
可能。

Apache CloudStack™
Open Source Cloud Computing™



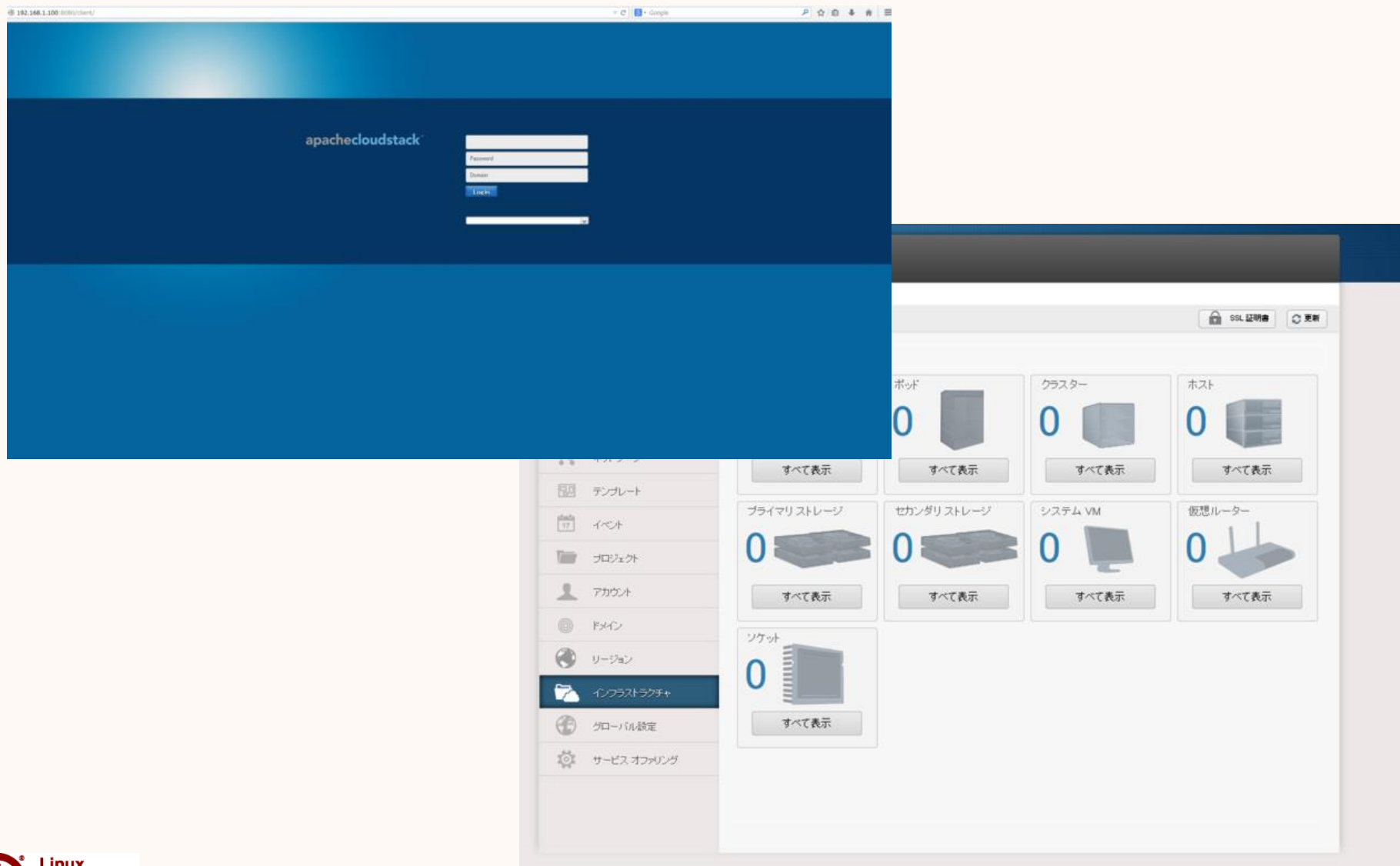
openstack™
CLOUD SOFTWARE

EUCALYPTUS





CloudStack





openstack

プロジェクト
コンピュート
ネットワーク

ネットワークポロジ

コンパクト 標準

public

client network

192.168.111.0/24

インスタンス名	イメージ名	IP アドレス	サイズ	キーペア	ステータス	アベイラビリティゾーン	タスク	電源状態	作成後経過時間	アクション
<input type="checkbox"/> 1st_instance	cirros	192.168.111.103	m1.tiny	keypair	稼働中	nova	なし	実行中	2 分	スナップショットの作成 ▼

1件表示

Openstack

フィルター インスタンスの起動 インスタンスの削除 その他のアクション ▼

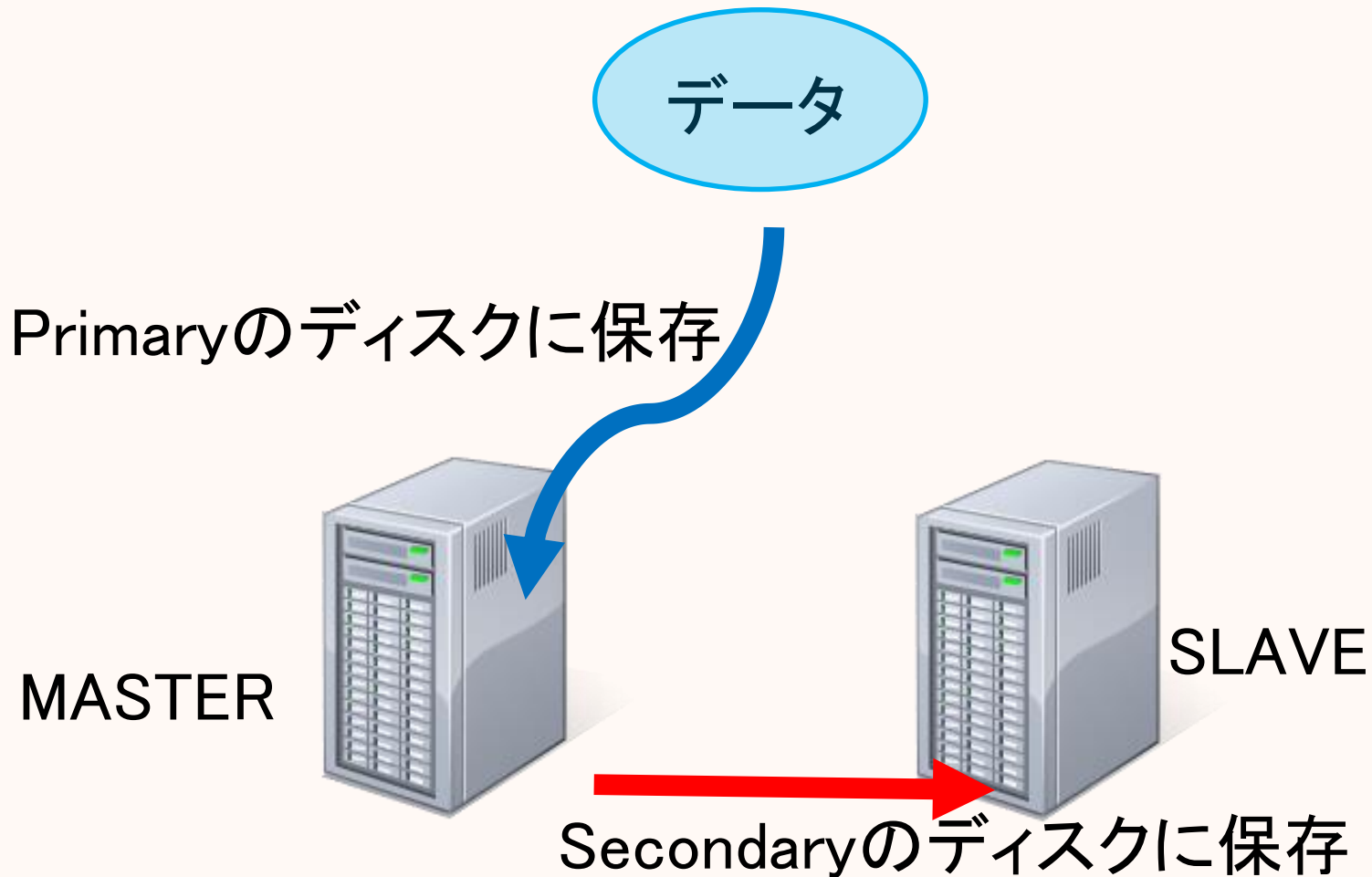


③技術解説

主題333: クラスタストレージ



クラスタストレージとは

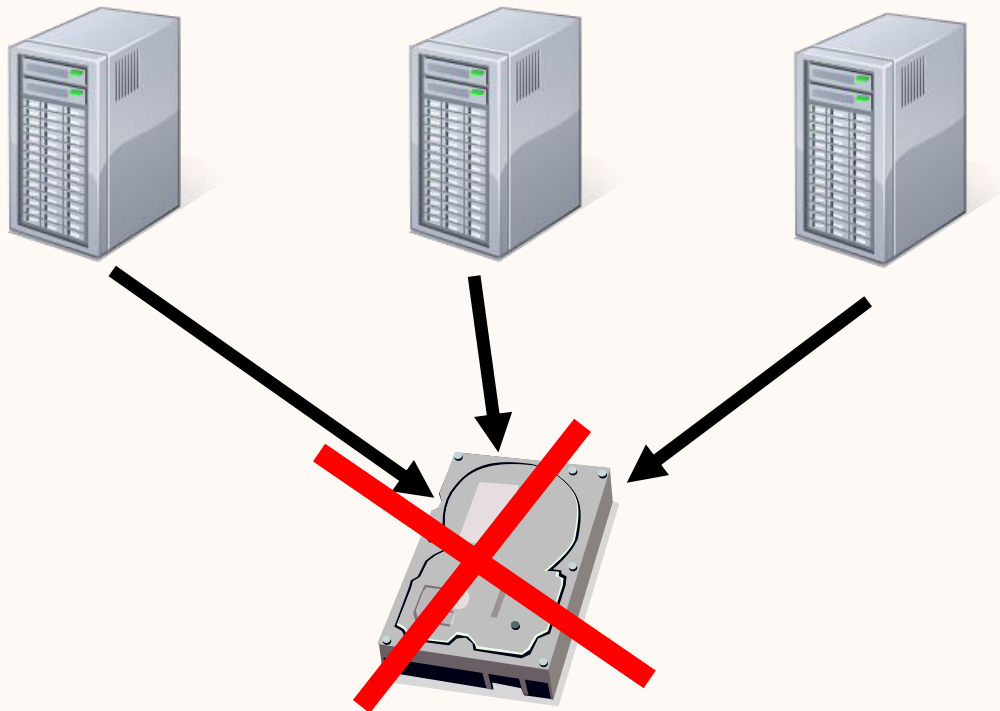




クラスタストレージとは

ディスクを冗長化するRAIDなどと同様に、データのコピーを行うものである。

また、共有ストレージのようにSPF(Single Point Failuer)とならない。






DRBD(Distributed Replicated Block Device)

<http://www.drbd.org/en/>

<https://www.3ware.co.jp/>



Software Development for High Availability Clusters

A PRODUCT BY LINBIT

download : docs : site

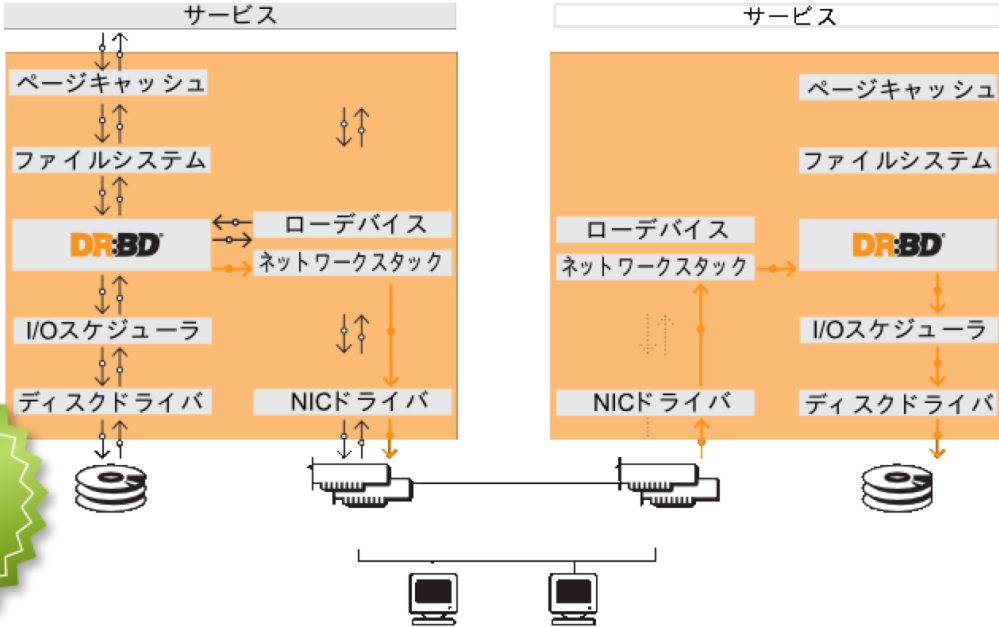
Choose a language

60

HomeドキュメントダウンロードサポートとトレーニングManagement Console利用状況

Home

- » DRBDの世界によろこ
- » HAとは何ですか？
- » DRBDの働き
- » リカバリ
- » オープンソース
- » 機能一覧
- » リリース
- » ロードマップ
- » 参考資料
- » 他の情報リソース
- » メーリングリスト
- » IRC



DRBD®はハイアベイリティ(HA)クラスタの構成要素として利用できるブロックデバイスです。このデバイスは、ネットワークを利用して、ブロックデバイス全体をミラーします。DRBDはネットワークを介したRAID 1と考えることができます。



データ

Primaryのディスクに保存

MASTER
(192.168.11.201)



SLAVE
(192.168.11.202)



Secondaryのディスクに同期



クラスタストレージの構築

ホスト名の確認(`uname -n`)

iptablesの無効化(`iptables -F & iptables-save > /etc/sysconfig/iptables`)

SELinuxの無効化(`setenforce 0 & /etc/sysconfig/selinux`でSELINUX=Disable)

```
[[root@MASTER ~]# uname -n
MASTER
[[root@MASTER ~]# iptables -F
[[root@MASTER ~]# iptables-save > /etc/sysconfig/iptables
[[root@MASTER ~]# setenforce 0
[[root@MASTER ~]# vi /etc/sysconfig/selinux
[[root@MASTER ~]# cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```



2台目のハードディスクにパーティション(/dev/sdb1)を追加

※ファイルシステムは作成しません。

```
[root@MASTER ~]# ls -l /dev/sdb
brw-rw---- 1 root disk 8, 16  1月  5 14:03 2016 /dev/sdb
[root@MASTER ~]# fdisk /dev/sdb
```

デバイスは正常な DOS 領域テーブルも、Sun, SGI や OSF ディスクラベルも含んでいません

新たに DOS ディスクラベルをディスク識別子 0x577036d6 で作成します。
あなたが書き込みを決定するまで、変更はメモリ内だけに残します。
その後はもちろん以前の内容は修復不可能になります。
警告: 領域テーブル 4 の不正なフラグ 0x0000 は w(書き込み)によって正常になります

警告: DOS互換モードは廃止予定です。このモード (コマンド 'c') を止めることを強く推奨します。 and change display units to sectors (command 'u').

```
[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 (1-1044, 初期値 1):
初期値 1 を使います
```



elrepoレポジトリの追加

```
[[root@MASTER ~]# rpm -ivh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm
http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm を取得中
警告: /var/tmp/rpm-tmp.75mA0m: ヘッダ V4 DSA/SHA1 Signature, key ID baadae52: NO KEY
準備中... ##### [100%]
      1:elrepo-release ##### [100%]
[[root@MASTER ~]# ls -l /etc/yum.repos.d/
合計 28
-rw-r--r--. 1 root root 1991  8月  4 01:13 2015 CentOS-Base.repo
-rw-r--r--. 1 root root  647  8月  4 01:13 2015 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  630  8月  4 01:13 2015 CentOS-Media.repo
-rw-r--r--. 1 root root 6259  8月  4 01:13 2015 CentOS-Vault.repo
-rw-r--r--. 1 root root  289  8月  4 01:13 2015 CentOS-fasttrack.repo
-rw-r--r--  1 root root 2150  2月 10 08:27 2014 elrepo.repo
```



drbd84-utilsとkmod-drbd84のインストール

```
[[root@MASTER ~]# yum -y install drbd84-utils kmod-drbd84
```

読み込んだプラグイン:fastestmirror, security

インストール処理の設定をしています

base	3.7 kB	00:00
base/primary_db	4.6 MB	00:08
elrepo	2.9 kB	00:00
elrepo/primary_db	716 kB	00:02
extras	2.9 kB	00:00
extras/primary_db	33 kB	00:00
updates	3.4 kB	00:00
updates/primary_db	3.3 MB	00:03

依存性の解決をしています

--> トランザクションの確認を実行しています。

---> Package drbd84-utils.x86_64 0:8.9.2-1.el6.elrepo will be インストール

---> Package kmod-drbd84.x86_64 0:8.4.6-1.el6.elrepo will be インストール

--> 依存性解決を終了しました。

依存性を解決しました

```
=====
パッケージ                アーキテクチャ                バージョン                リポジトリ   容量
=====
インストールしています:
drbd84-utils                x86_64                8.9.2-1.el6.elrepo        elrepo        470 k
kmod-drbd84                  x86_64                8.4.6-1.el6.elrepo        elrepo        194 k
=====
```



設定ファイル(/etc/drbd.conf)の確認

⇒ /etc/drbd.d内のglobal_common.confと*.resがインクルードされていることを確認。

```
[root@MASTER ~]# cat /etc/drbd.conf
# You can find an example in /usr/share/doc/drbd.../drbd.conf.example

include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```



/etc/drbd.d/global_common.confを編集

・・・Primaryで作成後、scpなどでSecondaryにコピーする

usage-count

(<http://usage.drbd.org>での

統計情報への報告)

protocol

(ローカル/リモート両ディスクにデータ保存で書込完了)

local-io-error

(物理デバイスのIOエラー発生時shutdown)

```
[[root@MASTER ~]# cat /etc/drbd.d/global_common.conf
global {
    usage-count no;
}

common {
    protocol C;

    handlers {
        local-io-error "/usr/lib/drbd/notify-io-error.sh; /usr/lib/drbd/notify-
emergency-shutdown.sh; echo o > /proc/sysrq-trigger ; halt -f";
    }
}
```




<http://usage.drbd.org>での統計情報(参考情報)

Usage DRBD.org

On this site we try to estimate the number of DRBD installations. In case you deploy DRBD, please consider to take part in this process.

When you start drbd for the first time, its tools will ask you if you like to participate in this process. In case you agree a random number gets created on your machine, and this random number is sent with DRBD's version number to <http://usage.drbd.org> and is stored on your machine. In case you update your DRBD installation later, the same random number is sent with the new version number.

When you create DRBD's meta data on a block device, you also have the chance to take part in this process. As there are random numbers for the machine there are also random numbers for the devices.

The benefits for you are:

- As a response to your data, the server will tell you how many users before you have installed this version.
- With a high counter the DRBD developers have a high motivation to continue development of the software.

The numbers in the grey boxes are generated from the database every time you load the page, they are always up-to-date.

I assume that any serious user of DRBD will upgrade his cluster nodes at least once. So to estimate the number of DRBD installations, take the "Number of nodes that were upgraded at least once", and multiply it by 10 (see the comparison with tgz download count below).

DRBD Statistics

Number of node count events	747381
Number of resources	850554
Number of nodes that were upgraded at least once	144777
node with most resources, number	388134467724613651, 95442

The 15 most recent installations/updates

Usually you will find here the developer's machines and early adopters. In the column "nth update" you can see how many times a new

most recent installations/updates

node	revision	installed/updated on	nth update
15163823438512078334	8.4.6	2016-01-05 05:39:08	1
3316116078449478076	8.4.6	2016-01-05 05:39:04	1
16009564512369232236	8.3.12	2016-01-05 05:35:41	1
3159672300244200060	87039E49...	2016-01-05 05:33:33	2
11701894452430469402	8.4.6	2016-01-05 05:29:32	1
6426397233009520002	8.4.6	2016-01-05 05:29:31	1
15636449166766489613	87039E49...	2016-01-05 05:21:04	2
7458510341311504351	B06F4B40...	2016-01-05 05:15:31	1
10558036122239240468	8.4.3	2016-01-05 05:15:27	1
13269176974203113983	B06F4B40...	2016-01-05 05:14:14	1
1386602260668103906	8.3.11	2016-01-05 04:48:39	2
6063271686485318015	DE923608...	2016-01-05 04:42:19	2
5756266443405401991	DE923608...	2016-01-05 04:42:16	2
2624711962056832067	8.3.12	2016-01-05 04:40:45	1
8168720540992055659	B06F4B40...	2016-01-05 04:18:25	2



リソース設定ファイル(r0.res)の作成

Primaryで作成後、scpなどでSecondaryにコピーする

```
[root@MASTER ~]# cat /etc/drbd.d/r0.res
resource r0 {
    protocol C;
    device /dev/drbd0;
    disk /dev/sdb1;
    meta-disk internal;
    net {
        cram-hmac-alg sha256;
        shared-secret "password";
    }
    on MASTER {
        address 192.168.11.201:7777;
    }

    on SLAVE {
        address 192.168.11.202:7777;
    }
}
```

※/proc/cryptで使用可能な認証アルゴリズムが確認出来ます。



r0.resの設定内容

cram-hmac-alg

(Master/Slave間の認証アルゴリズム)

shared-secret

(Master/Slave間の共通秘密鍵)

device

(DRBDリソースでのブロックデバイス名)

disk

(DRBD用に追加したブロックデバイス名)

meta-disk

(メタデータの格納に関する設定)

※メタデータとは、レプリケートするデータに関する情報の事

on DRBD1

(uname -nで表示される名前を設定)

address

(対向ノードからの接続を受付けるIPアドレスとポート)

<https://drbd.linbit.com/users-guide/re-drbdconf.html>に詳細説明



メタデータ領域の初期作成

drbdadm create-md r0

```
[[root@MASTER ~]# drbdadm create-md r0
initializing activity log
NOT initializing bitmap
Writing meta data...
New drbd meta data block successfully created.
```



drbdに起動(Primary/Secondaryで実施)
/etc/init.d/drbd start

```
[root@MASTER ~]# /etc/init.d/drbd start
Starting DRBD resources: [
    create res: r0
    prepare disk: r0
    adjust disk: r0
    adjust net: r0
]
.....
*****
DRBD's startup script waits for the peer node(s) to appear.
- If this node was already a degraded cluster before the
  reboot, the timeout is 0 seconds. [degr-wfc-timeout]
- If the peer was available before the reboot, the timeout
  is 0 seconds. [wfc-timeout]
  (These values are for resource 'r0'; 0 sec -> wait forever)
To abort waiting enter 'yes' [ 21]:
.
WARN: nothing stacked for this host (MASTER), nothing to do in stacked mode!
```



また、起動状況の確認も行う(Primary/Secondaryで実施)

```
netstat -tan | grep 7777
```

```
ps aux | grep drbd
```

```
[root@MASTER ~]# netstat -tan | grep 7777
tcp        0      0 192.168.11.201:46166      192.168.11.202:7777      ESTABLISHED
tcp        0      0 192.168.11.201:7777      192.168.11.202:40213     ESTABLISHED
[root@MASTER ~]# ps aux | grep drbd
root      12734  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd-reissue]
root      12741  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd_submit]
root      12749  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd_w_r0]
root      12753  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd_r_r0]
root      12767  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd_a_r0]
root      12768  0.0  0.0      0      0 ?        S      15:15   0:00 [drbd_ack_sender]
root      12772  0.0  0.0 103316  916 pts/1    S+     15:17   0:00 grep drbd
```



DRBDの状態を確認

/etc/init.d/drbd status

```
[root@MASTER ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res  cs          ro          ds          p  mounted  fstype
0:r0    Connected  Secondary/Secondary  Inconsistent/Inconsistent  C
```

cs(Connected)より、2ノード間での接続を確認

ro(Secondary/Secondary)より、Primaryが存在しないことを確認

Ds(Inconsistent/Inconsistent)より、データの未同期を確認

ちなみに、cat /proc/drbdでも同じような情報が確認出来ます。

```
[root@MASTER ~]# cat /proc/drbd
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r-----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:8385604
```



初期状態では2台ともSecondary状態になっているので、
Primaryとする方を強制的にPrimaryへと変更し、データの同期を行います。

```
[root@MASTER ~]# drbdadm -- --overwrite-data-of-peer primary r0
[root@MASTER ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res cs          ro          ds          p mounted fstype
0:r0 SyncSource Primary/Secondary UpToDate/Inconsistent C
... sync'ed:      0.2%          (8176/8188)M
```

Primary/Secondaryとなり、
データ同期中(PriorityはSyncSourceと表示)



Secondaryでの状態確認

```
[[root@SLAVE ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res cs ro ds p mounted fstype
0:r0 SyncTarget Secondary/Primary Inconsistent/UpToDate C
... sync'ed: 0.4% (8164/8188)M
```

Primary/Secondaryとなり、データ同期中(SecondaryはSyncTgargetと表示)

```
[[root@SLAVE ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res cs ro ds p mounted fstype
0:r0 Connected Secondary/Primary UpToDate/UpToDate C
```




/dev/drbd0にファイルシステム(今回はext4を使用)を作成し、マウント

※Primaryのみ

※/mnt/drbd0がマウントポイント

```
[[root@MASTER ~]# mkfs.ext4 /dev/drbd0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2096401 blocks
104820 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 21 mounts or
180 days, whichever comes first. Use tune2fs
```

```
[[root@MASTER ~]# mkdir /mnt/drbd0
[[root@MASTER ~]# mount -t ext4 /dev/drbd0 /mnt/drbd0
[[root@MASTER ~]# mount | grep drbd0
/dev/drbd0 on /mnt/drbd0 type ext4 (rw)
```



Primaryにて、DRBDデバイスにファイル(testfile)作成
アンマウント後、Secondary状態に切り替える
drbdadm secondary r0

```
[root@MASTER ~]# echo "1st message:Add by Primary(192.168.11.201)" > /mnt/drbd0/testfile
[root@MASTER ~]# cat /mnt/drbd0/testfile
1st message:Add by Primary(192.168.11.201)
[root@MASTER ~]# umount /mnt/drbd0
[root@MASTER ~]# drbdadm secondary r0
[root@MASTER ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res cs r0 ds p mounted fstype
0:r0 Connected Secondary/Secondary UpToDate/UpToDate C
```



SLAVEをPrimaryに昇格させる

```
[root@SLAVE ~]# mkdir /mnt/drbd0
[root@SLAVE ~]# drbdadm primary r0
[root@SLAVE ~]# /etc/init.d/drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00
m:res  cs          ro          ds          p  mounted  fstype
0:r0   Connected Primary/Secondary UpToDate/UpToDate C
[root@SLAVE ~]# mount -t ext4 /dev/drbd0 /mnt/drbd0
```

すると、testfileが同期されていることが確認出来ます

```
[root@SLAVE ~]# ls -l /mnt/drbd0/
合計 20
drwx----- 2 root root 16384 1月  5 15:34 2016 lost+found
-rw-r--r-- 1 root root    43 1月  5 15:39 2016 testfile
[root@SLAVE ~]# cat /mnt/drbd0/testfile
1st message:Add by Primary(192.168.11.201)
```



SLAVEでtestfileへ書き込みを行った後、
secondaryに降格させる

```
[root@SLAVE ~]# echo "2nd message:Add by Primary(192.168.11.202)" > /mnt/drbd0/testfile
[root@SLAVE ~]# cat /mnt/drbd0/testfile
2nd message:Add by Primary(192.168.11.202)
[root@SLAVE ~]# umount /mnt/drbd0
```

MASTERを再度、primaryに昇格させデータ同期を確認

```
[root@MASTER ~]# mount /dev/drbd0 /mnt/drbd0
[root@MASTER ~]# ls -l /mnt/drbd0
合計 20
drwx----- 2 root root 16384 1月 5 15:34 2016 lost+found
-rw-r--r-- 1 root root 43 1月 5 15:47 2016 testfile
[root@MASTER ~]# cat /mnt/drbd0/testfile
2nd message:Add by Primary(192.168.11.202)
```

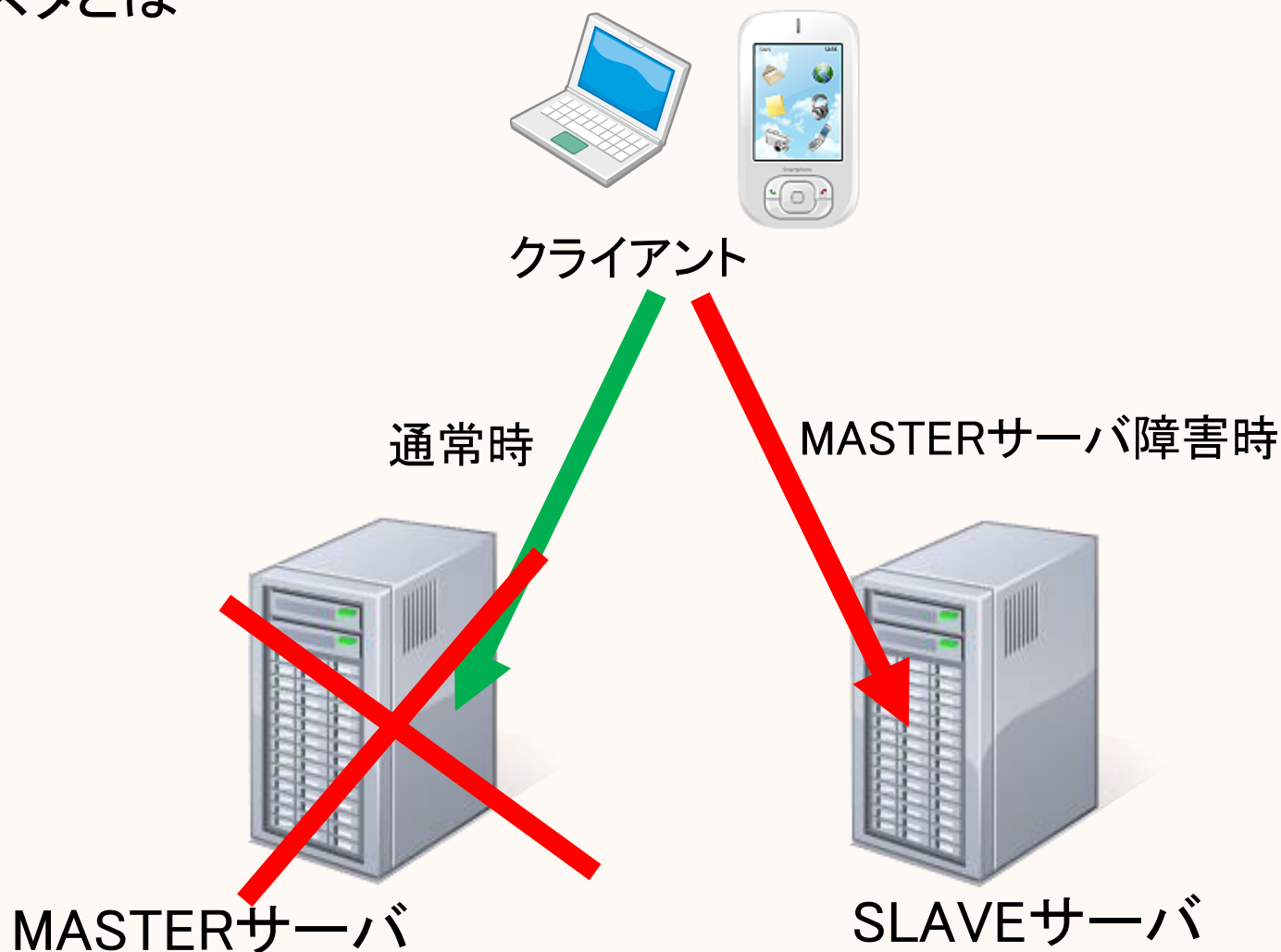


③技術解説

主題332: クラスタ管理

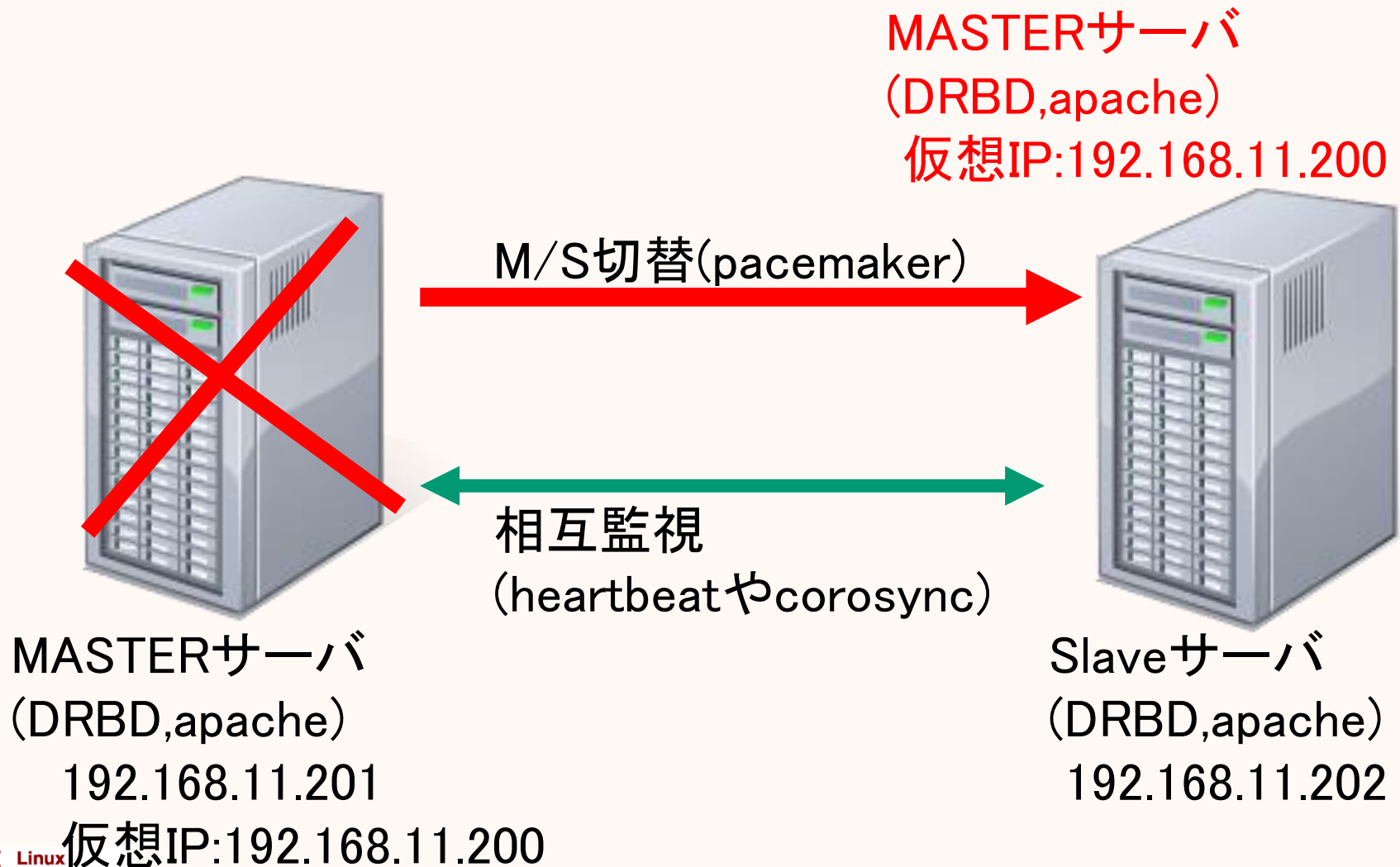


クラスタとは





heartbeat, corosync, pacemakerとは





クラスタの構築

apacheのインストール

yum -y install httpd

```
[root@MASTER ~]# yum -y install httpd
読み込んだプラグイン:fastestmirror, security
インストール処理の設定をしています
Determining fastest mirrors
 * base: www.ftp.ne.jp
 * elrepo: ftp.ne.jp
 * extras: www.ftp.ne.jp
 * updates: www.ftp.ne.jp
base | 3.7 kB | 00:00
elrepo | 2.9 kB | 00:00
extras | 2.9 kB | 00:00
updates | 3.4 kB | 00:00
updates/primary_db | 3.3 MB | 00:02
依存性の解決をしています
--> トランザクションの確認を実行しています。
---> Package httpd.x86_64 0:2.2.15-47.el6.centos.1 will be インストール
```




/var/www/htmlと/mnt/r0/htmlをリンク

Primary操作

```
[root@MASTER ~]# mv /var/www/html /mnt/r0
[root@MASTER ~]# ls -l /mnt/r0/
合計 4
drwxr-xr-x 2 root root 4096 12月 16 00:51 2015 html
[root@MASTER ~]# ln -s /mnt/r0/html /var/www/html
[root@MASTER ~]# ls -l /var/www/html
lrwxrwxrwx 1 root root 12 1月 12 20:39 2016 /var/www/html -> /mnt/r0/html
```

Secondary操作

```
[root@SLAVE ~]# rm -rf /var/www/html
[root@SLAVE ~]# ln -s /mnt/r0/html /var/www/html
```



Primaryにて、httpdのコンテンツファイル(index.html)を作成

```
[root@MASTER ~]# touch /var/www/html/index.html
[root@MASTER ~]# vi /var/www/html/index.html
[root@MASTER ~]# cat /var/www/html/index.html
<!doctype html>
<html>
<head>
<title>Test Cluster HTML</title>
</head>
<body>
<p>Test Cluster html's document!</p>
</body>
</html>
[root@MASTER ~]# ls -l /mnt/r0/html/
合計 4
-rw-r--r-- 1 root root 131  1月 12 20:43 2016 index.html
```



heartbeatのインストール

```
rpm -ivh http://ftp-srv2.kddilabs.jp/Linux/packages/CentOS/6.7/extras/x86_64/Packages/epel-release-6-8.noarch.rpm
```

```
yum install heartbeat
```

```
[root@MASTER ~]# rpm -ivh http://ftp-srv2.kddilabs.jp/Linux/packages/CentOS/6.7/extras/x86_64/Packages/epel-release-6-8.noarch.rpm
http://ftp-srv2.kddilabs.jp/Linux/packages/CentOS/6.7/extras/x86_64/Packages/epel-release-6-8.noarch.rpm を取得中
準備中... ##### [100%]
1:epel-release ##### [100%]
```

```
[root@MASTER ~]# yum install heartbeat
読み込んだプラグイン:fastestmirror, security
インストール処理の設定をしています
Loading mirror speeds from cached hostfile
epel/metalink | 7.7 kB 00:00
* base: www.ftp.ne.jp
* elrepo: ftp.ne.jp
* epel: ftp.tsukuba.wide.ad.jp
* extras: www.ftp.ne.jp
* updates: www.ftp.ne.jp
epel | 4.3 kB 00:00
epel/primary_db | 5.7 MB 00:05
依存性の解決をしています
--> トランザクションの確認を実行しています。
--> Package heartbeat.x86_64 0:3.0.4-2.el6 will be インストール
--> 依存性の処理をしています: resource-agents のパッケージ: heartbeat-3.0.4-2.el6.x86_64
```



rpm -qi heartbeatで情報確認

```
[root@MASTER ~]# rpm -qi heartbeat
Name           : heartbeat                      Relocations: (not relocatable)
Version        : 3.0.4                        Vendor: Fedora Project
Release        : 2.el6                        Build Date: 2013年12月03日 01時37分21秒
Install Date: 2016年01月12日 20時48分58秒      Build Host: buildvm-14.phx2.fedoraproject.org
Group          : System Environment/Daemons    Source RPM: heartbeat-3.0.4-2.el6.src.rpm
Size           : 269152                        License: GPLv2 and LGPLv2+
Signature      : RSA/8, 2013年12月03日 07時59分13秒, Key ID 3b49df2a0608b895
Packager       : Fedora Project
URL            : http://linux-ha.org/
Summary        : Messaging and membership subsystem for High-Availability Linux
Description    :
heartbeat is a basic high-availability subsystem for Linux-HA.
It will run scripts at initialization, and when machines go up or down.
This version will also perform IP address takeover using gratuitous ARPs.

Heartbeat contains a cluster membership layer, fencing, and local and
clusterwide resource management functionality.
```



http://sourceforge.jp/projects/linux-ha/releases/より、
pacemaker-1.0.13-2.1.el6.x86_64.repo.tar.gzを/tmpにダウンロード

sourceforge.jp/projects/linux-ha/releases/60151

SOURCEFORGE.JP ダウンロード Magazine 開発 コピペ アカウント ヘルプ

SourceForge.JP > ソフトウェアを探す > システム > クラスタリング/分散ネットワーク > Linux-HA Japan > ダウンロードファイル一覧 > パッケージ [1-01] Pacemakerリポジトリパッケージ (RHEL6) > リリース 1.0.13-1.2

カテゴリ: ソフトウェア

Linux-HA Japan Fork

概要 ダウンロード ソースコード チケット 文書 コミュニケーション ニュース

ダウンロードリスト

プロジェクト概要
本ページはLinux-HA Japan 開発者向けサイトです。プロジェクトのメインサイトはこちらです http://linux-ha.sourceforge.jp/ 成果物のダウンロードはこちらです http://sourceforge.jp/projects/linux-ha/releases/ 現在開発中のソースコードはこちらです https://github.com/linux-ha-japan Nightlyビルドはこちらです http://linux-ha.sourceforge.jp/nightly/ Linux-HA Japanプロジェクトは、Linux上で高可用クラスタシステムを構築するための部品として、オープンソースの、クラスタリソースマネージャ、クラスタ通信レイヤ、ブロックデバイス複製、その他、さまざまなアプリケーションに対応するための数多くのリソースエージェント、などを、日本国内向けに維持管理、支援等を行っているプロジェクトです。主な製品として、Pacemaker、Heartbeat、Corosync、DRBD等を取り扱っています。※現在、Linux-HA Japanの独自開発機能は個別にリリースしておりません。独自開発機能が必要な場合は、Pacemakerリポジトリパッケージに同梱されているRPMをご利用ください。

レビュー
平均評価 4.1
投票数 17
5つ星 13
4つ星 0
3つ星 0
2つ星 0
1つ星 4
あなたの評価 ☆☆☆☆☆ レビューする

[1-01] Pacemakerリポジトリパッケージ (RHEL6) 1.0.13-1.2 リリース時刻: 2013-12-25 16:59

名前	サイズ	MD5	日付	ダウンロード数
pacemaker-debuginfo-1.0.13-1.2.el6.x86_64.repo.tar.gz	4.8 MB	76df5376306a6...	2013-12-25 17:05	84
pacemaker-debuginfo-1.0.13-1.2.el6.i686.repo.tar.gz	4.8 MB	77cb2c7b78e23...	2013-12-25 17:05	25
pacemaker-1.0.13-1.2.el6.x86_64.repo.tar.gz	8.9 MB	edde5138db6f2...	2013-12-25 17:05	1816
pacemaker-1.0.13-1.2.el6.srpm.tar.gz	21.2 MB	b77abfc6bb1d8...	2013-12-25 17:05	132
pacemaker-1.0.13-1.2.el6.i686.repo.tar.gz	3.9 MB	edae559294c2...	2013-12-25 17:05	308

[1-01] Pacemakerリポ...
1.0.13-1.2
1.0.13-1.1
1.0.12-1.3
1.0.12-1.2
1.0.12-1.1
1.0.11-1.2.2
1.0.11-1.2.1
[1-02] Pacemakerリポ...
[1-09] pm_ctl : 運用管...
[1-10] pm_logconv-cs : ...
[2-01] Heartbeat (RedH...
[2-02] hb-monitor: クラ...
[2-03] hb-extras: 拡張リ...

```
[root@MASTER ~]# ls /tmp
pacemaker-1.0.13-2.1.el6.x86_64.repo.tar.gz yum.log
```



ダウンロードファイルを解凍すると、
作成されたディレクトリがローカルレポジトリとなっていることが確認出来る。

```
[root@MASTER tmp]# ls -lR pacemaker-1.0.13-2.1.el6.x86_64.repo
pacemaker-1.0.13-2.1.el6.x86_64.repo:
合計 12
-rw-r--r-- 1 root root 106 8月 5 17:34 2014 pacemaker.repo
drwxr-xr-x 2 root root 4096 8月 5 17:34 2014 repodata
drwxr-xr-x 2 root root 4096 8月 5 17:34 2014 rpm

pacemaker-1.0.13-2.1.el6.x86_64.repo/repodata:
合計 92
-rw-r--r-- 1 root root 7202 8月 5 17:34 2014 6c33169a95a00b231af6851de97dfd2b3c99f4abf2259c44e6cedddb8a7d756f-other.sqlite.bz2
-rw-r--r-- 1 root root 14644 8月 5 17:34 2014 709acd74199f00e3ab64ad2e3409e6557b5c6c0f5021dfa7680904e83d07c293-filelists.xml.gz
-rw-r--r-- 1 root root 19564 8月 5 17:34 2014 9f995e55a6e4dc027a1ea0781dd47c58b5b1b156266c941008d5fe5c5866b1d5-filelists.sqlite.bz2
-rw-r--r-- 1 root root 9687 8月 5 17:34 2014 b2b70521073e08b165c4b290bfc5eb44d119c83aab1fb9647afb4245e85904bd-primary.xml.gz
-rw-r--r-- 1 root root 4018 8月 5 17:34 2014 b6197845dfde21dfda930852ad054ec200227329f628747a9bca2750aad2c3ef-other.xml.gz
-rw-r--r-- 1 root root 25018 8月 5 17:34 2014 f8e3a69e166db1f29370d4961b069b4f078ac5e58099409cef978457dbeef925-primary.sqlite.bz2
-rw-r--r-- 1 root root 2989 8月 5 17:34 2014 repomd.xml

pacemaker-1.0.13-2.1.el6.x86_64.repo/rpm:
合計 9292
-rw-r--r-- 1 root root 266856 8月 5 17:34 2014 cluster-glue-1.0.11-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 112256 8月 5 17:34 2014 cluster-glue-libs-1.0.11-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 110624 8月 5 17:34 2014 cluster-glue-libs-devel-1.0.11-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 168788 8月 5 17:34 2014 corosync-1.4.6-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 146228 8月 5 17:34 2014 corosynclib-1.4.6-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 173340 8月 5 17:34 2014 corosynclib-devel-1.4.6-1.el6.x86_64.rpm
-rw-r--r-- 1 root root 165944 8月 5 17:34 2014 heartbeat-3.0.5-1.1.el6.x86_64.rpm
```



pacemakerのインストール

```
yum -c /tmp/pacemaker-1.0.13-2.1.el6.x86_64.repo/pacemaker.repo install pacemaker-1.0.13  
pm_extras
```

```
[root@MASTER tmp]# yum -c /tmp/pacemaker-1.0.13-2.1.el6.x86_64.repo/pacemaker.repo install pacemaker-1.0.13 pm_extras  
インストール処理の設定をしています  
base | 3.7 kB 00:00  
base/primary_db | 4.6 MB 00:06  
elrepo | 2.9 kB 00:00  
elrepo/primary_db | 716 kB 00:03  
epel/metalink | 7.7 kB 00:00  
epel | 4.3 kB 00:00  
epel/primary_db | 5.7 MB 00:05  
extras | 2.9 kB 00:00  
extras/primary_db | 33 kB 00:00  
pacemaker | 2.9 kB 00:00  
pacemaker/primary_db | 24 kB 00:00  
updates | 3.4 kB 00:00  
updates/primary_db | 3.3 MB 00:03  
依存性の解決をしています  
--> トランザクションの確認を実行しています。  
---> Package pacemaker.x86_64 0:1.0.13-2.el6 will be インストール
```




pacemakerのインストール情報を確認

Rpm -qi pacemaker

```
[root@MASTER tmp]# rpm -qi pacemaker
Name           : pacemaker                      Relocations: (not relocatable)
Version        : 1.0.13                        Vendor: Linux-HA Japan
Release        : 2.el6                         Build Date: 2014年08月05日 15時43分53秒
Install Date: 2016年01月12日 21時03分57秒      Build Host: build-rh6-x64
Group          : System Environment/Daemons     Source RPM: pacemaker-1.0.13-2.el6.src.rpm
Size           : 9032996                       License: GPLv2+ and LGPLv2+
Signature      : (none)
URL            : http://www.clusterlabs.org
Summary       : Scalable High-Availability cluster resource manager
Description   :
Pacemaker is an advanced, scalable High-Availability cluster resource
manager for Linux-HA (Heartbeat) and/or OpenAIS.

It supports "n-node" clusters with significant capabilities for
managing resources and dependencies.

It will run scripts at initialization, when machines go up or down,
when related resources fail and can be configured to periodically check
resource health.
```




heartbeatの設定ファイル及び認証用ファイルの雛形ファイルをコピーする

```
cp -p /usr/share/doc/heartbeat-3.0.4/ha.cf /etc/ha.d/ha.cf
```

```
cp -p /usr/share/doc/heartbeat-3.0.4/authkeys /etc/ha.d/authkeys
```

```
[root@MASTER tmp]# ls -l /usr/share/doc/heartbeat-3.0.4/
合計 144
-rw-r--r-- 1 root root 3701 12月 3 01:37 2013 AUTHORS
-rw-r--r-- 1 root root 17989 12月 3 01:37 2013 COPYING
-rw-r--r-- 1 root root 26532 12月 3 01:37 2013 COPYING.LGPL
-rw-r--r-- 1 root root 58752 12月 3 01:37 2013 ChangeLog
-rw-r--r-- 1 root root 2935 12月 3 01:37 2013 README
-rw-r--r-- 1 root root 1873 12月 3 01:37 2013 apphbd.cf
-rw-r--r-- 1 root root 645 12月 3 01:37 2013 authkeys
-rw-r--r-- 1 root root 10502 12月 3 01:37 2013 ha.cf
-rw-r--r-- 1 root root 5905 12月 3 01:37 2013 haresources
```

```
[root@MASTER tmp]# cp -p /usr/share/doc/heartbeat-3.0.4/ha.cf /etc/ha.d/ha.cf
```

```
[root@MASTER tmp]# cp -p /usr/share/doc/heartbeat-3.0.4/authkeys /etc/ha.d/authkeys
```



heartbeatの設定ファイルを編集

※scpなどでSLAVEにコピーする

```
[root@MASTER tmp]# vi /etc/ha.d/ha.cf
[root@MASTER tmp]# grep -v "^#" /etc/ha.d/ha.cf
pacemaker yes
logfacility      local1
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport 694
bcast   eth0           # Linux
auto_failback on
node    MASTER
node    SLAVE
```

なお、nodeで記載する名前については、`uname -n`と一致させる

```
[root@MASTER tmp]# uname -n
MASTER
```

```
[root@SLAVE tmp]# uname -n
SLAVE
```



ha.cfの設定内容

pacemaker.....pacemakerの使用
logfacility.....出力ログのファシリティ
keepalive.....相互確認の間隔
deadtime.....相手ノードがダウンと判断する時間
warntime.....相手ノードがダウンした警告を出力するまでの時間
initdead.....起動時にheartbeatによる監視を行うまでの時間
 (OS起動正常起動までの十分な時間を指定)
udpport.....使用ポート
bcast.....heartbeatをブロードキャストで行う場合の
 インターフェイス指定
auto_failback.....自動フェイルバックの設定
node.....クラスタを組むノード名(uname -nで表示されるもの)



rsyslogの設定追加 及びrsyslogの再起動

```
[root@MASTER tmp]# vi /etc/rsyslog.conf
[root@MASTER tmp]# grep "local1" /etc/rsyslog.conf
local1.*                                /var/log/ha.log
[root@MASTER tmp]# /etc/init.d/rsyslog restart
システムロガーを停止中:                [ OK ]
システムロガーを起動中:                [ OK ]
```

これで、HAのログがtail -f /var/log/ha.logで監視可能となる。



authkeysの編集及び権限変更(600)
※scpなどでSecondaryにコピーする

```
[root@MASTER ha.d]# vi /etc/ha.d/authkeys
[root@MASTER ha.d]# grep -v "^#" /etc/ha.d/authkeys
auth 2
2 sha1 password
```

```
[root@MASTER tmp]# ls -l /etc/ha.d/authkeys
-rw-r--r-- 1 root root 661 1月 12 21:17 2016 /etc/ha.d/authkeys
[root@MASTER tmp]# chmod 600 /etc/ha.d/authkeys
[root@MASTER tmp]# ls -l /etc/ha.d/authkeys
-rw----- 1 root root 661 1月 12 21:17 2016 /etc/ha.d/authkeys
```



Primary、Secondaryの順でheartbeatを起動する

```
[[root@MASTER ~]# /etc/init.d/heartbeat start  
Starting High-Availability services: Done.
```

```
[[root@SLAVE ~]# /etc/init.d/heartbeat start  
Starting High-Availability services: Done.
```

すると、/var/lib/heartbeat/crm配下にcib.xmlなどが生成される

```
[[root@MASTER ~]# ls -l /var/lib/heartbeat/crm/  
合計 8  
-rw----- 1 hacluster haclient 232 1月 13 17:40 2016 cib.xml  
-rw-r--r-- 1 hacluster haclient 32 1月 13 17:40 2016 cib.xml.sig
```



Primaryにて、crm_mon -rfAにて、クラスタ状態を確認

```
=====
Last updated: Wed Jan 13 17:41:14 2016
Stack: Heartbeat
Current DC: slave (36327bb2-e13d-48bf-abd4-16a5af745e33) - partition with quorum

Version: 1.0.13-a83fae5
2 Nodes configured, unknown expected votes
0 Resources configured.
=====

Online: [ master slave ]

Full list of resources:

Node Attributes:
* Node master:
* Node slave:

Migration summary:
* Node master:
* Node slave:
```



Secondaryでheartbeatを停止すると、状態が**OFFLINE**に変更される。

```
[root@SLAVE ~]# /etc/init.d/heartbeat stop
Stopping High-Availability services: Done.
```

```
=====
Last updated: Wed Jan 13 17:42:18 2016
Stack: Heartbeat
Current DC: master (fc55d32e-db6f-4989-9ab2-fcb7433f13bf) - partition with quorum
Version: 1.0.13-a83fae5
2 Nodes configured, unknown expected votes
0 Resources configured.
=====
Online: [ master ]
Offline: [ slave ]
Full list of resources:

Node Attributes:
* Node master:

Migration summary:
* Node master:
```




また、cluster2のheartbeatのみが起動している場合は、
以下ようになる

```
=====
Last updated: Wed Jan 13 17:43:39 2016
Stack: Heartbeat
Current DC: slave (36327bb2-e13d-48bf-abd4-16a5af745e33) - partition with quorum

Version: 1.0.13-a83fae5
2 Nodes configured, unknown expected votes
0 Resources configured.
=====

Online: [ slave ]
OFFLINE: [ master ]

Full list of resources:

Node Attributes:
* Node slave:

Migration summary:
* Node slave:
```



また、ha.cfに

respawn root /usr/lib64/heartbeat/ifcheckd

(実行する外部プログラムと実行ユーザーを指定)を追加すると、

※heartbeatを再起動します。

```
[root@MASTER ~]# vi /etc/ha.d/ha.cf
[root@MASTER ~]# grep -v "^#" /etc/ha.d/ha.cf
pacemaker yes
logfacility      local1
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport 694
bcast    eth0                # Linux
auto_failback on
node     MASTER
node     SLAVE
respawn  root /usr/lib64/heartbeat/ifcheckd
```



eth0の監視状態が追加される。

```
=====
Last updated: Wed Jan 13 17:52:26 2016
Stack: Heartbeat
Current DC: slave (36327bb2-e13d-48bf-abd4-16a5af745e33) - partition with quorum

Version: 1.0.13-a83fae5
2 Nodes configured, unknown expected votes
0 Resources configured.
=====

Online: [ master slave ]

Full list of resources:

Node Attributes:
* Node master:
    + slave-eth0                : up
* Node slave:
    + master-eth0               : up

Migration summary:
* Node master:
* Node slave:
```



続いて、apacheに関する監視設定を追加します。

pacemakerのhttpd監視では、server-statusを使用するので、httpd.confの以下の部分を有効化します。

```
<Location /server-status>  
    SetHandler server-status  
    Order deny,allow  
    Deny from all  
    Allow from 127.0.0.1  
</Location>
```



pacemaker全体の設定

```
[[root@MASTER ~]# crm
[crm(live)# configure
INFO: building help index
[crm(live)configure# property stonith-enabled="false"
[crm(live)configure# property no-quorum-policy="ignore"
[crm(live)configure# property default-resource-stickiness="200"
```

リソースの設定(DRBD res_drbd0)

```
[crm(live)configure# primitive res_drbd0 ocf:linbit:drbd params drbd_resource="r0" drbdconf="/etc/drbd.conf" op m
onitor interval="20s"
WARNING: res_drbd0: default timeout 20s for start is smaller than the advised 240
WARNING: res_drbd0: default timeout 20s for stop is smaller than the advised 100
WARNING: res_drbd0: action monitor not advertised in meta-data, it may not be supported by the RA
[crm(live)configure# ms ms_drbd0 res_drbd0 meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="
1" notify="true"
```



リソースの設定(ファイルシステム res_filesystem)

```
[crm(live)configure# primitive res_filesystem ocf:heartbeat:Filesystem params device="/dev/drbd0" fstype="ext4" directory="/mnt/r0" op monitor interval="20s"  
WARNING: res_filesystem: default timeout 20s for start is smaller than the advised 60  
WARNING: res_filesystem: default timeout 20s for stop is smaller than the advised 60  
WARNING: res_filesystem: default timeout 20s for monitor is smaller than the advised 40
```

リソースの設定(apache res_httpd)

```
[crm(live)configure# primitive res_httpd ocf:heartbeat:apache params configfile="/etc/httpd/conf/httpd.conf" port="80" op monitor interval="20s"  
WARNING: res_httpd: default timeout 20s for start is smaller than the advised 40s  
WARNING: res_httpd: default timeout 20s for stop is smaller than the advised 60s
```

リソースの設定(仮想IP res_ip)

```
[crm(live)configure# primitive res_ip ocf:heartbeat:IPaddr2 params ip="192.168.11.200" cidr_netmask="24" nic="eth0" op monitor interval="10s"
```



リソースグループの設定(rg_http)

```
[crm(live)configure# group rg_http res_ip res_filesystem res_httpd
```

プログラムの起動順番と、リソースの関連性に関する設定 (c_rg_http_on_drbd0 & o_drbd_before_rg_http)

```
[crm(live)configure# colocation c_rg_http_on_drbd0 inf: rg_http ms_drbd0:Master  
[crm(live)configure# order o_drbd_before_rg_http inf: ms_drbd0:promote rg_http:start
```



設定内容のチェック、コミット(確定)

```
[crm(live)configure# show
node $id="36327bb2-e13d-48bf-abd4-16a5af745e33" slave
node $id="fc55d32e-db6f-4989-9ab2-fcb7433f13bf" master
primitive res_drbd0 ocf:linbit:drbd \
    params drbd_resource="r0" drbdconf="/etc/drbd.conf" \
    op monitor interval="20s"
primitive res_filesystem ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" fstype="ext4" directory="/mnt/r0" \
    op monitor interval="20s"
primitive res_httpd ocf:heartbeat:apache \
    params configfile="/etc/httpd/conf/httpd.conf" port="80" \
    op monitor interval="20s"
primitive res_ip ocf:heartbeat:IPaddr2 \
    params ip="192.168.11.200" cidr_netmask="24" nic="eth0" \
    op monitor interval="10s"
group rg_http res_ip res_filesystem res_httpd
ms ms_drbd0 res_drbd0 \
    meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"
colocation c_rg_http_on_drbd0 inf: rg_http ms_drbd0:Master
order o_drbd_before_rg_http inf: ms_drbd0:promote rg_http:start
property $id="cib-bootstrap-options" \
    dc-version="1.0.13-a83fae5" \
    cluster-infrastructure="Heartbeat" \
    stonith-enabled="false" \
    no-quorum-policy="ignore" \
    default-resource-stickiness="200"
```




設定内容のチェック、コミット(確定)

```
[crm(live)configure# verify
WARNING: res_drbd0: default timeout 20s for start is smaller than the advised 240
WARNING: res_drbd0: default timeout 20s for stop is smaller than the advised 100
WARNING: res_drbd0: action monitor not advertised in meta-data, it may not be supported by the RA
WARNING: res_filesystem: default timeout 20s for start is smaller than the advised 60
WARNING: res_filesystem: default timeout 20s for stop is smaller than the advised 60
WARNING: res_filesystem: default timeout 20s for monitor is smaller than the advised 40
WARNING: res_httpd: default timeout 20s for start is smaller than the advised 40s
WARNING: res_httpd: default timeout 20s for stop is smaller than the advised 60s
```

```
[crm(live)configure# commit
WARNING: res_drbd0: default timeout 20s for start is smaller than the advised 240
WARNING: res_drbd0: default timeout 20s for stop is smaller than the advised 100
WARNING: res_drbd0: action monitor not advertised in meta-data, it may not be supported by the RA
WARNING: res_filesystem: default timeout 20s for start is smaller than the advised 60
WARNING: res_filesystem: default timeout 20s for stop is smaller than the advised 60
WARNING: res_filesystem: default timeout 20s for monitor is smaller than the advised 40
WARNING: res_httpd: default timeout 20s for start is smaller than the advised 40s
WARNING: res_httpd: default timeout 20s for stop is smaller than the advised 60s
[crm(live)configure# exit
bye
```



設定内容変更の確認

```
=====
Last updated: Thu Jan 14 01:54:50 2016
Stack: Heartbeat
Current DC: master (fc55d32e-db6f-4989-9ab2-fcb7433f13bf) - partition with quorum
Version: 1.0.13-a83fae5
2 Nodes configured, unknown expected votes
2 Resources configured.
=====

Online: [ master slave ]

Full list of resources:

Master/Slave Set: ms_drbd0
  Masters: [ master ]
  Slaves: [ slave ]
Resource Group: rg_http
  res_ip      (ocf::heartbeat:IPaddr2):      Started master
  res_filesystem (ocf::heartbeat:Filesystem): Started master
  res_httpd   (ocf::heartbeat:apache):       Started master

Node Attributes:
* Node master:
  + master-res_drbd0:0      : 10000
  + slave-eth0              : up
* Node slave:
  + master-eth0              : up
  + master-res_drbd0:1      : 1000

Migration summary:
* Node master:
* Node slave:
```



フェイルオーバー動作の確認

crm resource migrateを使用

```
[root@DRBD1 tmp]# crm
crm(live)# resource
crm(live)resource# migrate rg_http
WARNING: Creating rsc_location constraint 'cli-standby-rg_http' with a score of -INFINITY for resource rg_http on drbd1.
    This will prevent rg_http from running on drbd1 until the constraint is removed using the 'crm_resource -U' command or manually with cibadmin
    This will be the case even if drbd1 is the last node in the cluster
    This message can be disabled with -Q
```



SLAVEがPrimaryとなる

```
=====
Last updated: Tue Jul 15 22:24:15 2014
Stack: Heartbeat
Current DC: drbd2 (6b8910e9-00ca-43b4-8836-a10956005982) - partition with quorum
Version: 1.0.13-30bb726
2 Nodes configured, unknown expected votes
2 Resources configured.
=====

Online: [ drbd1 drbd2 ]

Full list of resources:

Master/Slave Set: ms_drbd0
Masters: [ drbd2 ]
Slaves: [ drbd1 ]
Resource Group: rg_httpd
res_ip      (ocf::heartbeat:IPaddr2):      Started drbd2
res_filesystem (ocf::heartbeat:Filesystem): Started drbd2
res_httpd   (ocf::heartbeat:apache):       Started drbd2

Node Attributes:
* Node drbd1:
  + drbd2-eth1                : up
  + master-res_drbd0:0        : 10000
* Node drbd2:
  + drbd1-eth1                : up
  + master-res_drbd0:1        : 10000

Migration summary:
* Node drbd1:
* Node drbd2:
```



フェイルオーバー動作の確認が出来たら、
MASTERをクラスタに戻しておきます。

crm resource unmigrateを使用

```
crm(live)resource# unmigrate rg_http
crm(live)resource# exit
bye
```

なお、chkconfigでdrbd/httpdは自動起動しないようにしておく。
※pacemaker/heartbeatによる切替が実施される為

```
[[root@MASTER crm]# chkconfig httpd off
[[root@MASTER crm]# chkconfig drbd off
[[root@MASTER crm]# chkconfig --list | grep -E "(http|drbd)"
drbd          0:off    1:off    2:off    3:off    4:off    5:off    6:off
httpd         0:off    1:off    2:off    3:off    4:off    5:off    6:off
```



スプリットブレインとは
「相互監視が出来なくなった場合に、
両方とも**Masterサーバ**になってしまう現象」

Masterサーバ
(DRBD,apache)
仮想IP:192.168.11.200

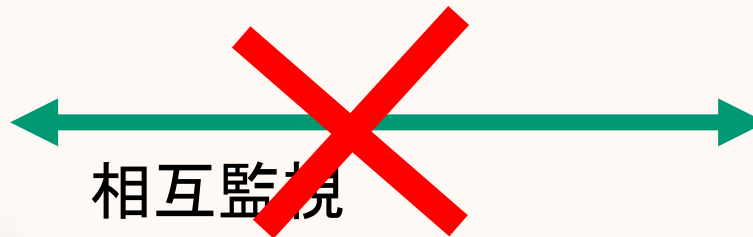


Masterサーバ
(DRBD,apache)
192.168.11.201

Masterサーバ
(DRBD,apache)
仮想IP:192.168.11.200



Slaveサーバ
(DRBD,apache)
192.168.11.202



相互監視



STONITHとは

「スプリットブレインにならないように、

相互監視が出来なくなったら相手を強制停止等を行う機能」

※専用デバイス等が必要となります。

Masterサーバ

(DRBD,apache)

仮想IP:192.168.11.200



Masterサーバ
(DRBD,apache)

192.168.11.201



Slaveサーバ
(DRBD,apache)

192.168.11.202



前述までの要素を全て考慮すると、
以下のような構成になる。





heartbeat/pacemakerについてもっと詳しく知るには、
Linux-HAのページが非常に役に立ちます。

<http://linux-ha.sourceforge.jp/wp/>

※マニュアルには、今回解説していないpacemaker+corosyncのやり方もあり。



LINUX-HA JAPAN

High-Availability Clustering on Linux

[Posts](#)
[Comments](#)

[HOME](#)
[メーリングリスト](#)
[ダウンロード&インストール](#)
[マニュアル](#)
[デスクトップテーマ・壁紙等](#)
[コミュニティ概要](#)
[関連リンク](#)

[その他](#)
[ニュース](#)
[イベント情報](#)
[読み物](#)
[WEBラジオ](#)

Linux-HA Japan プロジェクト



[ツイート](#)
127

[Check](#)

Linux上で**高可用(HA)クラスタシステム**を構築するための 部品として、オープンソースの、クラスタリソースマネージャ、クラスタ通信レイヤ、ブロックデバイス複製、その他、さまざまなアプリケーションに対応するための数多くのリソースエージェント等を、日本国内向けに維持管理、支援等を行っているプロジェクトです。

今は主に Pacemaker , Heartbeat , Corosync , DRBD等を扱ってます。

Linux-HA Japan 成果物ダウンロード	RHEL/CentOS向けPacemaker RPMパッケージ(yumのリポジトリ形式)や設定ファイル(crm)作成支援ツール、ディスク監視機能などをダウンロードできます。とりあえずRHELもしくはCentOS等のRHEL互換OSにインストールしてみたい場合は こちら 。インストール後にとりあえず何か動かしてみたい場合は こちら を参考にしてみてください。
マニュアル	本家コミュニティ提供の公式マニュアルやLinux-HA Japan提供の翻訳マニュアル。 マニュアル読んでもよくわからない場合は、過去の カンファレンス や 勉強会 等の発表資料も参考に。
メーリングリスト	インストール方法や設定方法等の質問はMLまで。 ※投稿するにはメールアドレスの登録が必要です。
イベント情報	カンファレンスへの出展や講演、勉強会開催情報、講演時のスライド公開など。
開発者向けサイト	Linux-HA Japan開発者向けサイトです。Linux-HA Japan独自開発機能のソースコードやバイナリのダウンロード等。

最近の投稿

- OSC2014 Hokkaido セミナー資料公開 2014/06/18
- 製品呼称に関する注意喚起 2014/04/01
- OSC2014 Tokyo/Spring セミナー資料・アンケート結果公開 2014/03/05
- 動かして理解する Pacemaker ~CRM設定編~ その3 2014/01/14
- Pacemaker-1.0.13-1.1 + RHEL5 における IPAddr2 の不具合について 2013/12/27
- Pacemakerリポジトリパッケージ1.0.13-1.2リリース 2013/12/26
- OSC2013 Tokyo/Fall セミナー資料・アンケート結果公開 2013/10/22
- 動かして理解する Pacemaker ~CRM設定





④サンプル問題



問題1

/proc/drbdの状態の表示について正しいものを選びなさい。

1. roには、ローカルのハードディスクに書き込まれたデータ量が表示される
2. csには、ノードの接続状態が表示される
3. dsには、ノードの状態が表示される
4. dwには、ハードディスクの状態が表示される



問題2

以下の仮想化ソリューションのうち、
ホストOS型に分類されるものを選びなさい。

1. Xen
2. KVM
3. OpenVZ
4. VirtualBox



問題3

クラウド管理ツールの説明として間違っているものを選びなさい。

1. クラウド管理ツールは仮想マシンやネットワークなどを管理する
2. OpenStackやCloudStackを使って、独自のクラウド環境を構築できる
3. Eucalyptusは多機能な独自APIを提供するクラウド管理ツールである
4. OpenNebulaはXenやKVMなどに対応したクラウド管理ツールである



問題4

Xenの説明として間違っているものを選びなさい。

1. Domain 0が仮想マシンの操作やデバイスとのI/Oなどを管理している
2. Domain Uでは準仮想化に対応したカーネルが動作する
3. Xen上でゲストOSとして動作するのはカーネルが対応している
Linuxだけである
4. XenはGPLでソースコードが公開されている



問題5

KVMの説明として正しくないものを選びなさい。

1. KVM本体はカーネルモジュールとして実装されている
2. Linuxが動作すればKVMが利用できる
3. 仮想マシンはプロセスとして管理される
4. KVMの仮想マシンはQEMUの各種機能を利用している



サンプル問題の詳細は、
<http://www.lpi.or.jp/ex/304/> で確認出来ます。



⑤お知らせ



LPIC 304 技術解説セミナー

Zeus IT Campからのお知らせ

土日開講 無料体験 カードOK フリータイム 教材費無料 カウンセリング相談無料 増税以降も受講料据え置き 給付金対象講座アリ

●受講者の声 ●よくあるご質問 ●アクセス ●お問合せ

Linuxに強い！そのワケとは…？

Point 1
LPI-Japan 公認
パートナーインストラクター
が講師を務めています。

当スクールは、Linux/OSS 技術者の技術力の認定制度「LPIC」を運営する【LPI-Japan】のアカデミック認定校としてお墨付き。
そして LPI-Japan 公認パートナーインストラクターとして厳選された講師が授業を行っています。

Point 2
ゼウス・ラーニングパワーは
【LPI-Japan】の
アカデミック認定校です！

LPI Approved Training Partner

だから、Linuxに強いんです。

**教育訓練給付金
指定講座認定記念特別キャンペーン!!**

雇用保険に加入し、
勤務していた期間の
合計が1年以上ある。

YES 当てはまる！
…という方は
CUCK!

Linux in English コース



Carl Stevens

LPI-Japan/パートナーインストラクター

This is your chance to study Linux with a native speaker of English while in Japan. I have helped numerous students to pass the LPIC exams. Study with me and gain the English and IT skills you need to succeed!

プロフィール

ニュージーランドより来日し講師として16年のキャリアを持つベテラン講師。
Linux in Englishのカリキュラムは他のスクールにはない新しいスタイルのコースです。

保有資格

LPIC 302 (Mixed Environment)
LPIC 301 (Core)
CCNA
Japanese Proficiency Test, Level 1 (日本語能力試験1級)

LPI-Japan 協理理事との
対談も掲載！

資料請求
WEB資料・紙資料をご用意しています

**LPIC合格率は
驚異の98%**

98%
86%
12%

Linuxの認定資格であるLPICは「ITエンジニアにとって当たり前の」王道資格と呼ばれる事も、LPIC1の合格率は、受験者全体で65%ほどと言われ難関資格とも言えるでしょう。
当スクールでは、本物のサーバ(等)を実際に操作することで理解を深め、正しい知識を吸収することができるため155人の受講者の合格率はなんと98%！(初回受験の合格率は86%)
短期集中でしっかりと学習して試験突破を目標そう！

企業の貴重な【人材】育成をお手伝いします！

**IT研修
お任せください**

助成金のご相談承ります。

女性に人気の講座です

**MS Office
基礎実践コース**

パソコンを使ったお仕事を目指そう！

日本で唯一、
ネイティブスピーカーが教えるLinux！！
(日本語もしゃべるので安心)



ゼウス・エンタープライズからのお知らせ

ここがポイント！

- ✓ 一次面接にお越し頂いた方に交通費1,000円支給！！
- ✓ 1日で内定も可能！『スピード選考会』も実施中！！
- ✓ リーマンショックを解雇者「ゼロ」で乗り切った会社
- ✓ 設立当初から全社員の「終身雇用」を掲げている会社
- ✓ 入社当日から100%正社員雇用をしている会社
- ✓ 年間を通じて社内イベントが多く社員同士の交流が盛ん



未経験者の方も大歓迎
新入社員の採用を積極的に行っています！

詳しくは、<http://www.zeus-enterprise.co.jp/>



⑥質疑応答



ご清聴、ありがとうございました。

皆様のLPIC304合格と、
皆様との再会を楽しみにしております！