仮想化

目次

第1章 仮想化概要
1-1. 仮想化とは 9
1-2. 仮想化の種類 11
1-3. 仮想化製品 12
1-4. サーバ仮想化 13
1-5. デスクトップ仮想化 14
1-6. 仮想化とクラウド 15

目次

第2章 仮想化概要
2-1. ネットワークとは 16
2-2. LAN 17
2-3. WAN 18
2-4. IPアドレス 19
2-5. ネットワーク部とホスト部 22
2-6. IPアドレスのクラス 24

目次

第3章 仮想環境の構成要素
3-1. 仮想化のコンポーネント 27
3-2. 仮想マシン 28
3-3. 仮想CPU 29
3-4. 仮想メモリ 32
3-5. 仮想HBA, 仮想ディスク 33
3-6. 仮想NIC 34
3-7. 仮想スイッチ、仮想ルータ 35
3-8. 演習 36

目次

第4章 仮想ネットワーク
4-1. 仮想ネットワークの概要 38
4-2. ネットワークとVLAN 39
4-3. 仮想NICとネットワーク形態 40
4-4. 仮想ネットワーク Host-only 41
4-5. 仮想ネットワーク NAT 42
4-6. 仮想ネットワーク Bridge 43

目次

第5章 仮想環境の運用
5-1. 仮想環境の運用 45
5-2. 様々な構築支援ツール 46
5-3. Vagrantの概要 47
5-4. 運用とバックアップ 49
5-5. 移行（マイグレーション） 51
5-6. P2Vマイグレーション 52
5-7. V2Vマイグレーション 53
5-8. ライブマイグレーション 54
5-9. Nested VM 55
5-10. 演習 56

目次

第6章 コンテナの概要
6-1. コンテナとは 58
6-2. コンテナの動作 59
6-3. 仮想化との違い 60
6-4. Dockerの概要 61
6-5. コンテナの概要 62
6-6. Docker Hub 63
6-7. DockerとOS 64
6-8. コンテナの用途 65
6-9. コンテナのサイズ 66
6-10. Dockerのプロビジョニング 67

第7章 コンテナの実践
7-1. 演習 69

E-Learning

第1章 仮想化概要

仮想化の基本を学ぶ

1-1. 仮想化とは

プロセッサやメモリ、ディスク、通信回線など、コンピュータシステムを構成する資源（リソース）を、物理的構成に拠らず柔軟に分割したり統合したりすること

1台のサーバコンピュータをあたかも複数台のコンピュータであるかのように論理的に分割し、それぞれに別のOSやアプリケーションソフトを動作させる「サーバ仮想化」や、複数のディスクをあたかも1台のディスクであるかのように扱い、大容量のデータを一括して保存したり耐障害性を高めたりする「ストレージ仮想化」などの技術がある

IT用語辞典より転載

1-2. 仮想化の種類

仮想化は次の種類に分けられる

- サーバ仮想化
- デスクトップ仮想化
- ネットワーク仮想化
- ストレージ仮想化

1-3. 仮想化製品

代表的な製品（ハイパーバイザ）

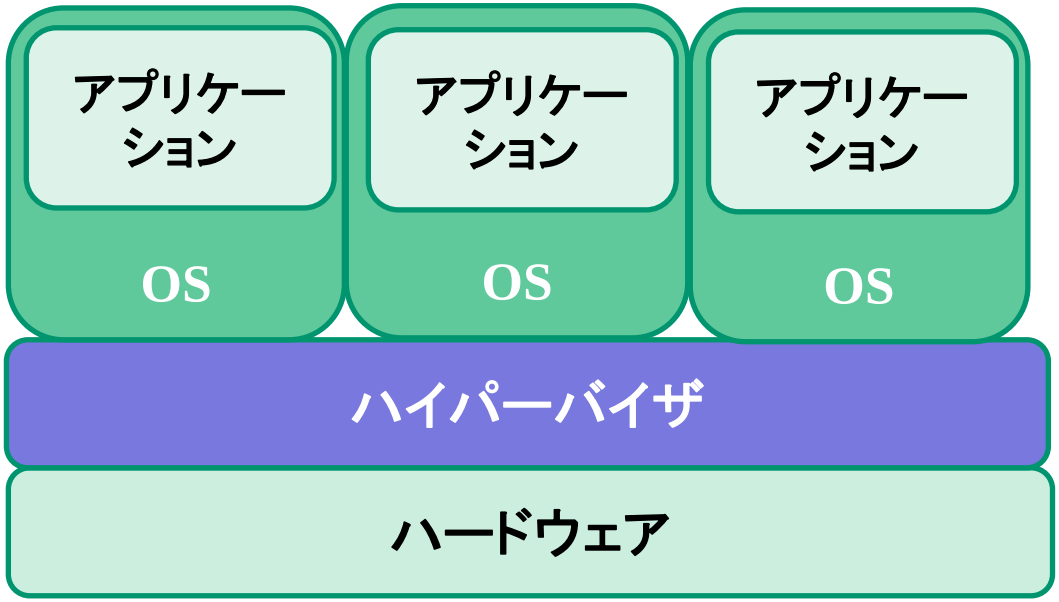
- サーバ仮想化 (Type-I)
  - VMware Sphere
  - Microsoft Hyper-V
  - Xen
- デスクトップ仮想化 (Type-II)
  - VMware Workstation, Player, Fusion
  - Microsoft Hyper-V
  - Virtual Box

1-4. サーバ仮想化

Type-I型、ネイティブ型、ベアメタル型

ハードウェア上で直接動作するハイパーバイザ

OSはすべてハイパーバイザ上で動作し、ほぼネイティブ環境に近い動作速度が得られる

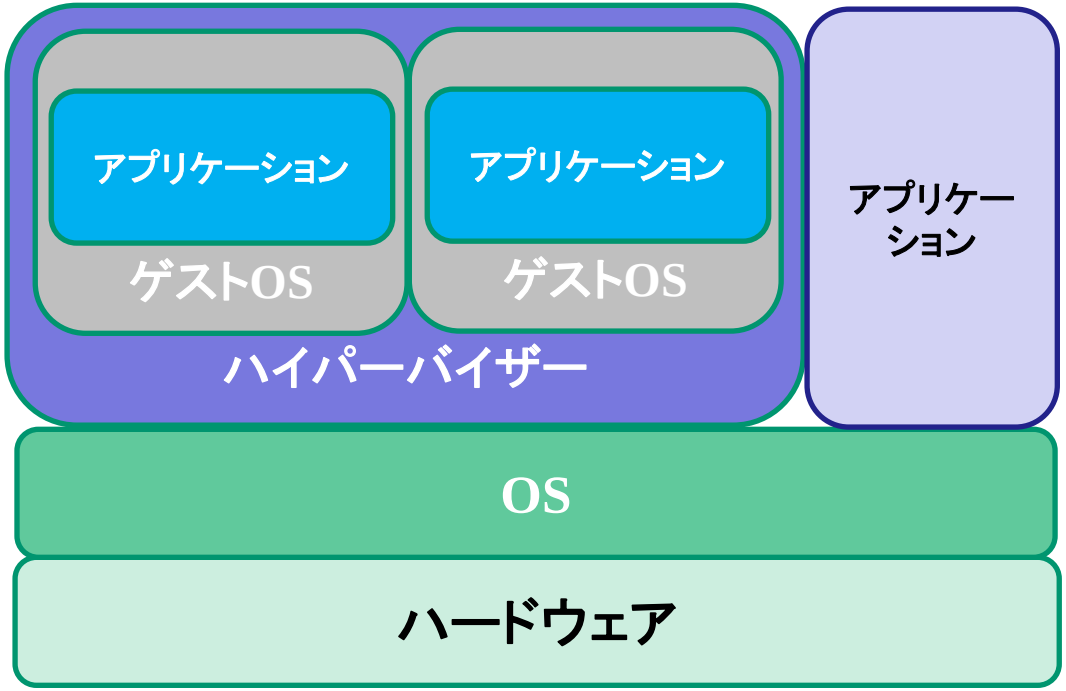


1-5. デスクトップ仮想化

Type-II、ホストOS型

ハードウェア上にまずOSが動作し、その上で1アプリケーションとしてハイパーバイザが動作

ハイパーバイザー上で仮想環境が動作



1-6. 仮想化とクラウド

クラウドサービスでは仮想化環境を提供

クラウドサービスはユーザが自由にリソースを変更可能。仮想化環境がマッチしている

E-Learning

第2章 ネットワークの基本

仮想化技術を習得する上で最低限
の知識

2-1. ネットワークとは

ネットワークとは，網という意味の英単語。複数の要素が互いに接続された網状の構造体のこと。ネットワークを構成する各要素のことを「ノード」(node)，ノード間の繋がりのことを「リンク」(link)あるいは「エッジ」(edge)と言う。

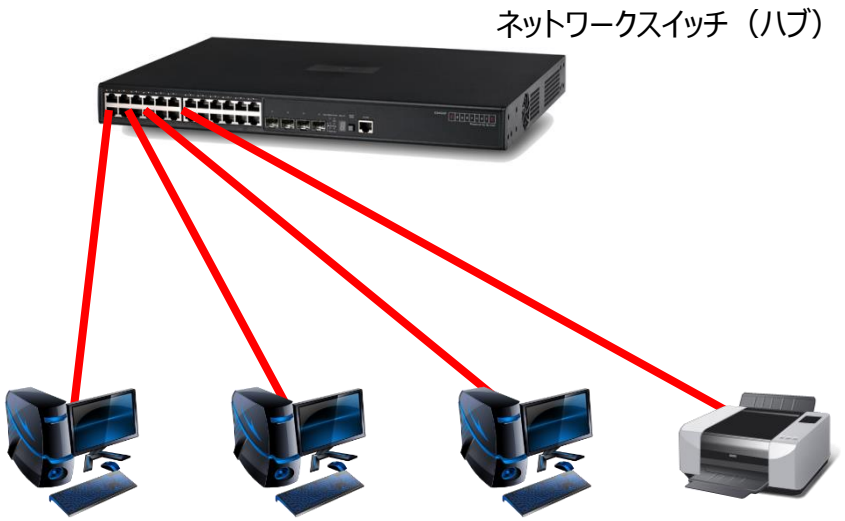
一般の外来語としては人間関係の広がりのことや，組織や集団の構造などを指すこともあるが，IT関連の分野で単にネットワークという場合は、複数のコンピュータや電子機器などを繋いで信号やデータ，情報をやりとりすることができるコンピュータネットワークあるいは通信ネットワークのことを意味することが多い。

IT用語辞典より転載

2-2. LAN

LAN (Local Area Network)

- 1つのフロア，組織，部署といった，比較的狭い範囲でのネットワーク



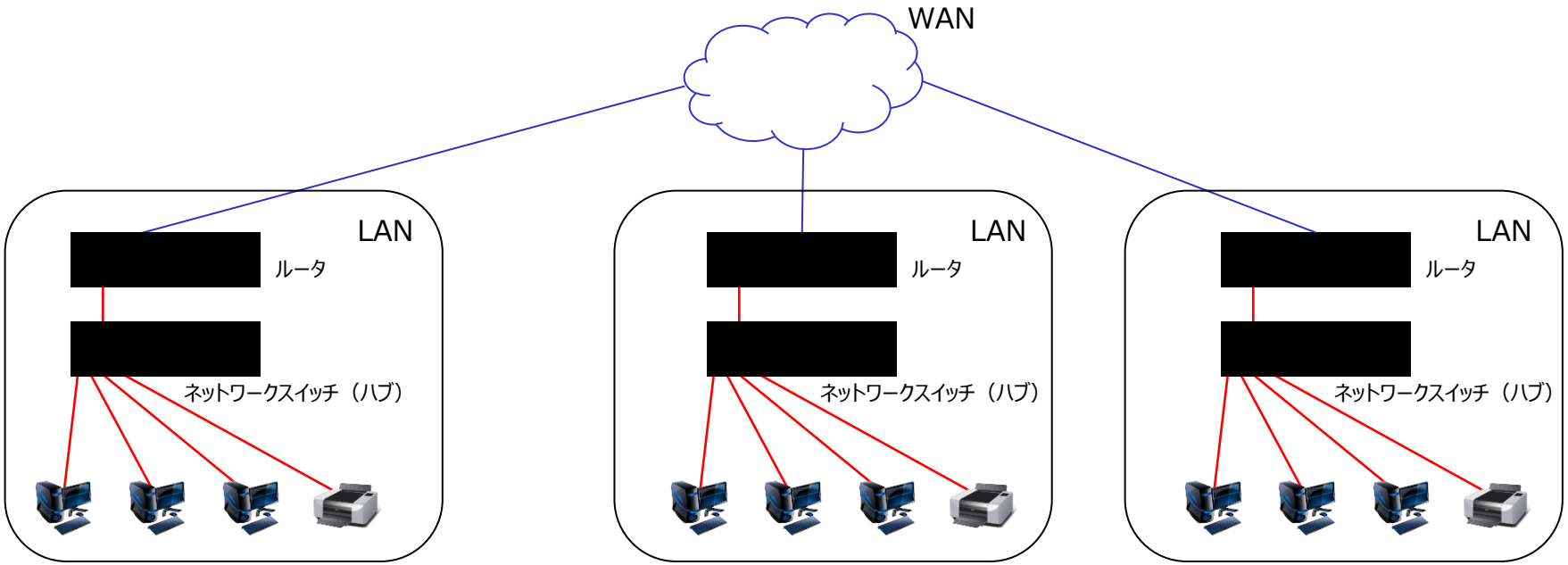
配線図

論理図

2-3. WAN

WAN (Wide Area Network)

- 広域通信網（Wide Area Network）の略。
  LANとLANを結ぶ公衆網のことを指す場合が多い。
  WANを世界規模で実現しているのがインターネット。



2-4. IPアドレス

- ネットワーク上で機器の識別をするための番号
- 32bit でIPアドレスを管理 (IPv4, Internet Protocol Version 4)
  - bit … 情報量の最小単位，1 or 0
- ネットワーク部とホスト部の存在

192.168.1.10

10進 → 2進に変換

$192_{(10)}$ = 1100 0000 (2)

$168_{(10)}$ = 1010 1000 (2)

$1_{(10)}$ = 0000 0001 (2)

$10_{(10)}$ = 0000 1010 (2)

IPアドレス

192.168.1.10

1100 0000 | 1010 1000 | 0000 0001 | 0000 1010

2進数の並びが 8×4 ＝ 32コ

⇩

32bit

※ 8bitのかたまり ＝ 1オクテットと呼ぶことも
octet

IPアドレス（続）

- 1オクテットの範囲

0000 0000 ～ 1111 1111

すべて 0　　すべて 1

2進 → 10進に変換

0000 0000 (2) ＝ 0 (10)

1111 1111 (2) ＝ 255 (10)

※ あくまでも1オクテットで表現できる範囲を10進法で表しているだけ。
※ 3桁だからといって，999.999.999.999 のような値を表現できるわけではない。

2-5. ネットワーク部とホスト部

192.168.1.10/24

この例だと，IPアドレスのビット列の，

先頭から24bit ＝ ネットワーク部

残りの8bit ＝ ホスト部

1100 0000 1010 1000 0000 0001 | 0000 1010

24bit：ネットワーク部

ネットワークプレフィックスと言うことも．
network prefix

端末がどのネットワークに所属しているのかを判別．

8bit：ホスト部

ネットワーク内の端末を識別

ひとつのネットワークに収容できるホスト数

ホスト部がすべて 0 ⇨ **ネットワーク自身**のことを表す

ホスト部がすべて 1 ⇨ そのネットワーク全体にデータを送る
ブロードキャストアドレスを表す

192.168.1.10 ならば…

ネットワーク自身 ⇨ **192.168.1.0**

1100 0000 1010 1000 0000 0001 | 0000 0000

すべて 0

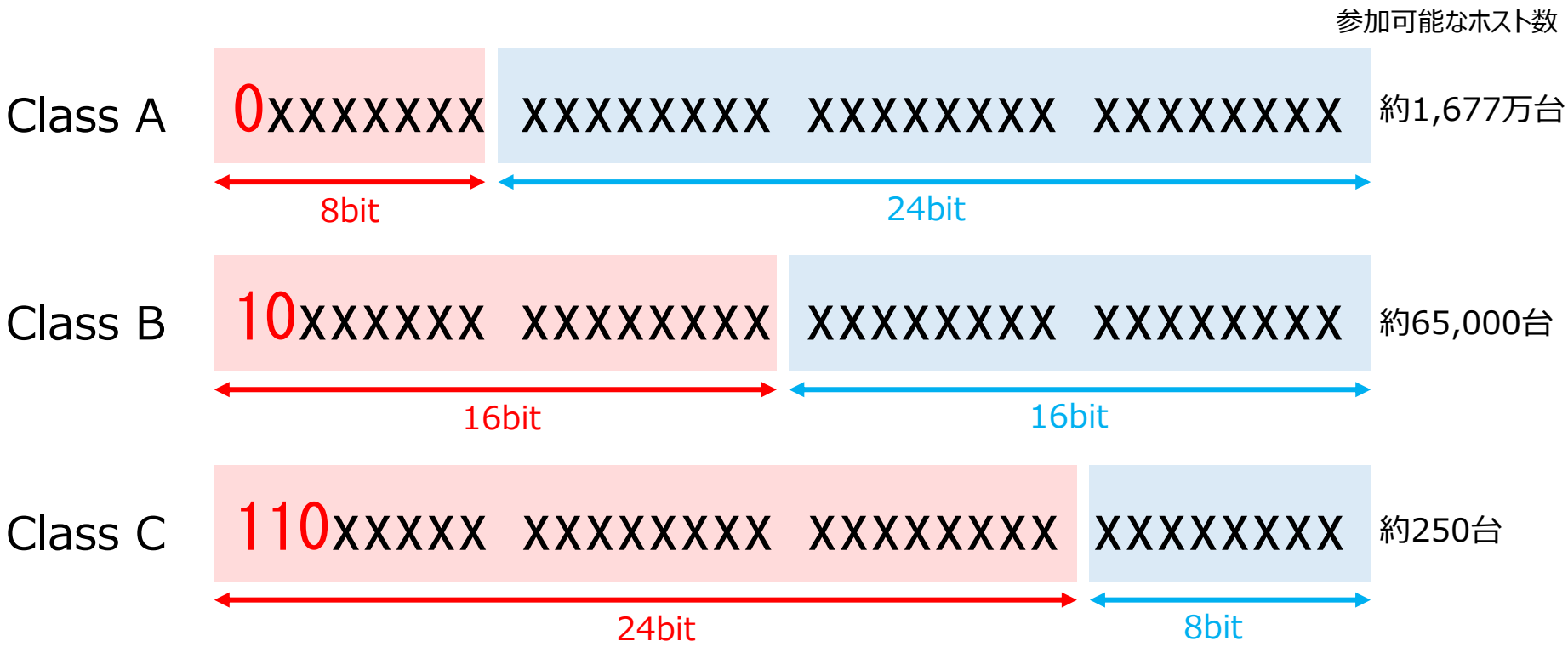
ブロードキャストアドレス ⇨ **192.168.1.255**

1100 0000 1010 1000 0000 0001 | 1111 1111

すべて 1

2-6. IPアドレスのクラス

- 32bit中，「何bitが**ネットワーク部**で，何bitが**ホスト部**を表すか」
- ネットワークの規模を識別する



他にもClass D, Class Eも存在

IPアドレスのクラス

（例題）次のIPアドレスのクラス（ネットワークの規模）は？

172.16.0.1

10進 → 2進に変換

$172_{(10)} = \boxed{1010\ \ 1100}_{(2)}$

先頭ビットが　10… なので，「**クラスB**」であることがわかる

E-Learning

第3章 仮想環境の構成要素

主要コンポーネントについて理解する

3-1. 仮想化のコンポーネント

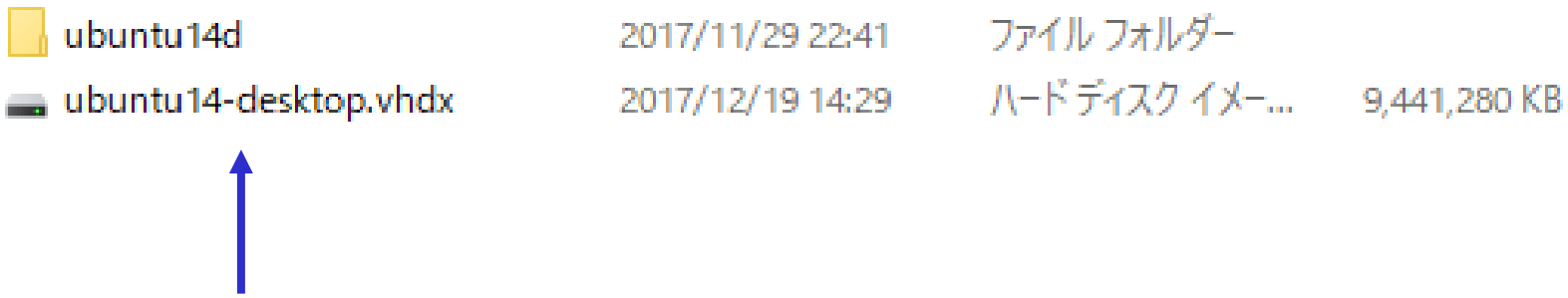
主なコンポーネント

- 仮想マシン
- 仮想CPU
- 仮想メモリ
- 仮想HBA
- 仮想ディスク
- 仮想NIC
- 仮想スイッチ

3-2. 仮想マシン

物理的なPC,サーバをハイパーバイザによって仮想的な機械に置き換えたもの
Virtual Machine：VM
構成ファイル、仮想ストレージなど構成される

Hyper-Vの例



仮想マシン（ハードディスクイメージ）

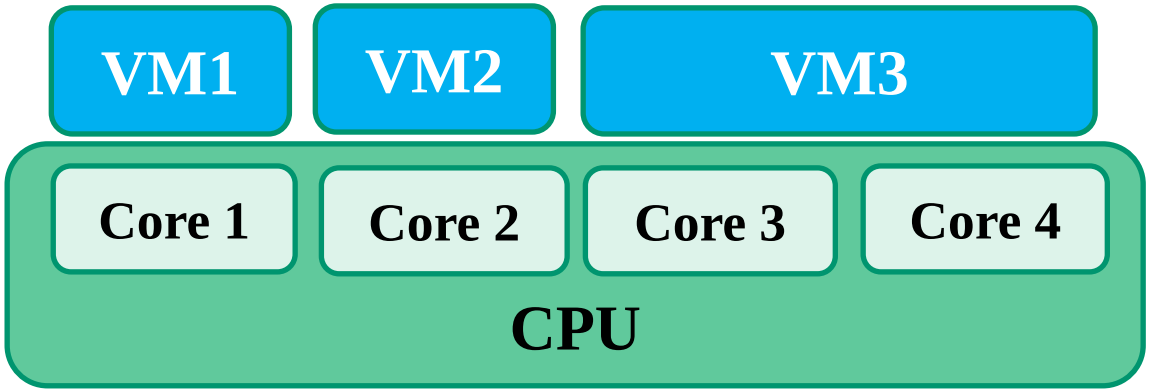
3-3. 仮想CPU

仮想マシンからは仮想CPUが物理CPUとして見える

仮想マシン1つにつき1つのCPU、もしくは1つのコアを割り振るのが理想的

重い仮想マシン（VM3）には多くのコアを割り振り、軽い仮想マシン（VM1,VM2）には少ないコアを割り振る

商用のハイパーバイザの中には、CPUのコア数、仮想CPUの総コア数でライセンスが決まるものもある



3-3. 仮想CPUと仮想化支援機能

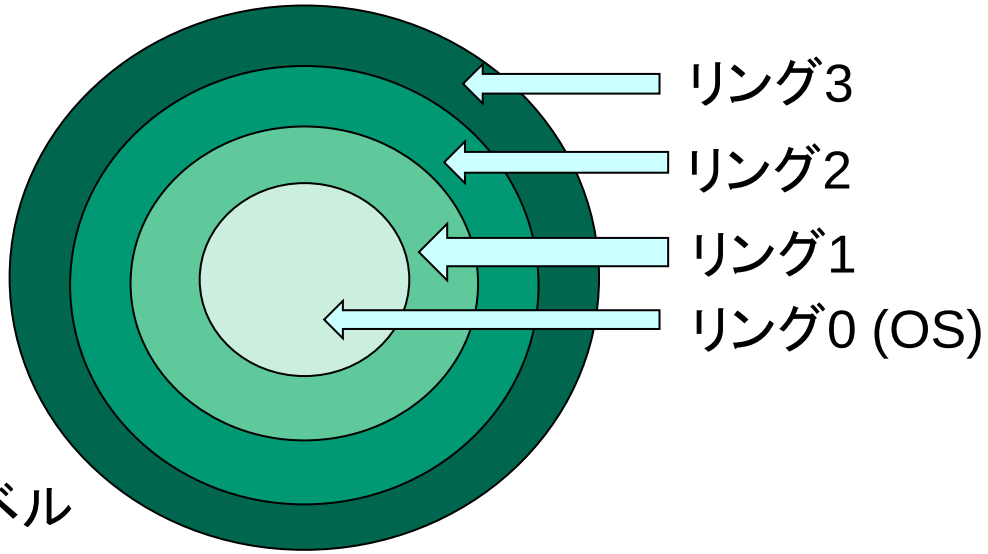
近年のCPUには、仮想CPUを物理CPUとほぼ同等に動作させる仮想化支援機能がある

Intel CPU: VT-x, AMD CPU:AMD-V

CPUの特権レベルが最も高いリング0にアクセスできるのはOSのみ

アプリケーションやハイパーバイザはリング3で動作

リング3からリング0へのアクセスは例外が発生する



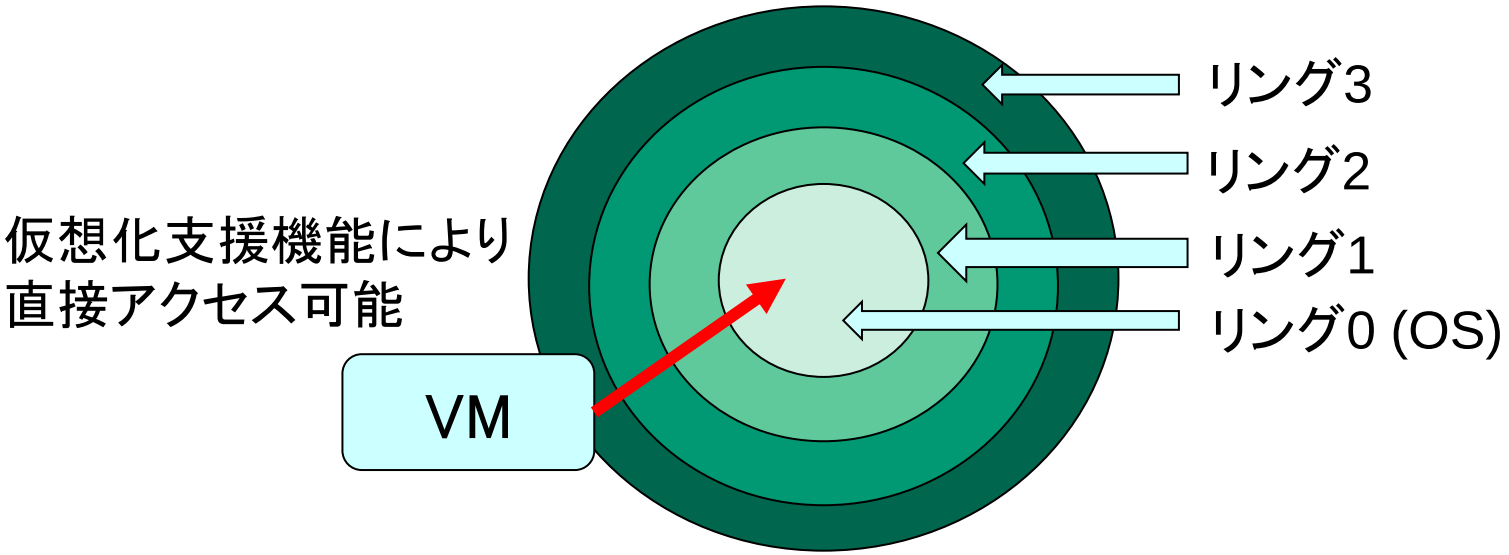
CPUの特権レベル

3-3. 仮想CPUと仮想化支援機能

VT-xやAMD-Vに対応したハイパーバイザではVMからリング0へ例外なしにアクセス可能

I/Oもほぼ直接アクセスできるため、物理環境に近い速度で動作

デフォルトで仮想化支援機能が無効になっているPCがあるので注意



3-4. 仮想メモリ

仮想マシンの動作には当然メモリが必要

ハイパーバイザによって最大メモリ量が異なる

仮想マシンへの仮想メモリとして割り当てる際に、完全固定として割り当てる方法（スタティック）と、最小容量と最大容量を定めておき、動的に変更できる方法がある

Type-I ハイパーバイザ：仮想マシンがほぼすべてのメモリを使用可能

Type-II ハイパーバイザ：他アプリケーションが使用するメモリ容量を考慮して仮想メモリを設計する

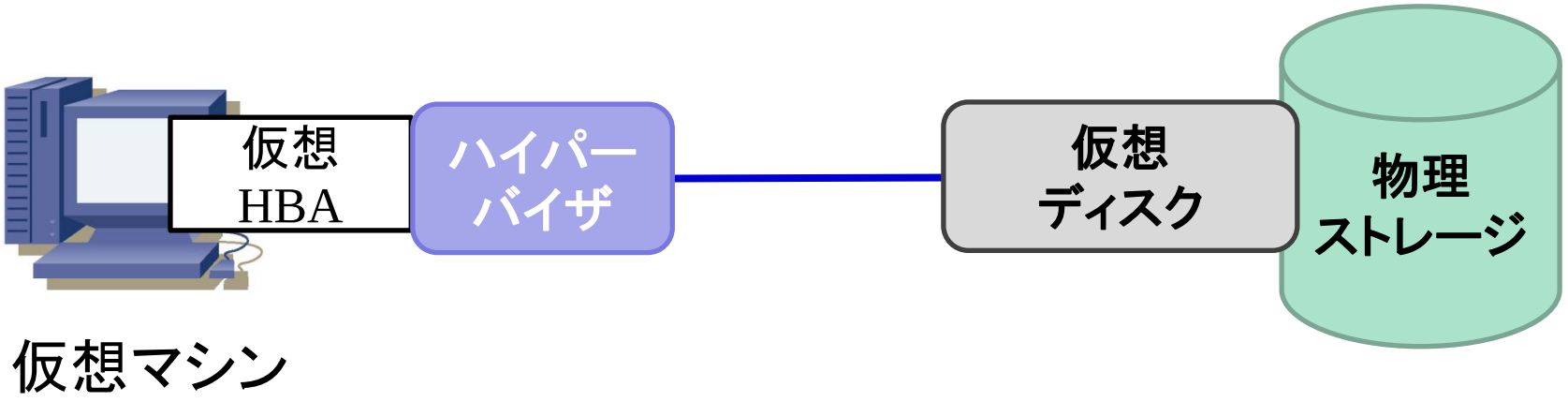
3-5. 仮想HBA, 仮想ディスク

仮想マシンは仮想ディスクをストレージとして使用する

仮想マシンには、ホストバスアダプタ（HBA）としてIDEやSCSIのインターフェイスが接続され、各方式の内蔵ディスクとして見える

仮想ディスクは動的構成を取ることができる

設定上120Gの仮想ディスクを使用しても、実際に使用しているサイズのみ物理ストレージ上では消費する

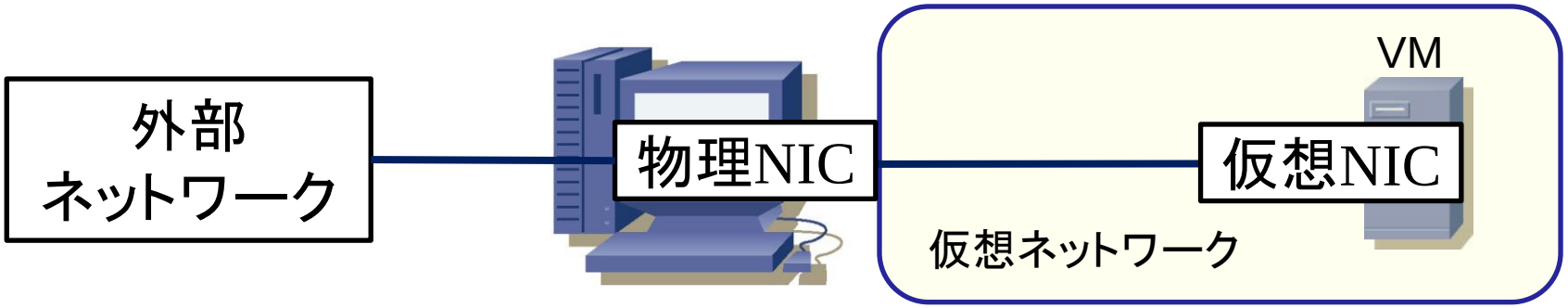


3-6. 仮想NIC

ハイパーバイザによって仮想的なインターフェイスである仮想NICが作成され、仮想ネットワークも作成される

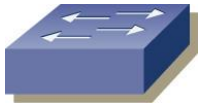
物理NICと仮想NICを接続しないと、ホストOSと仮想OSは通信できない

物理NICと仮想NICの間に仮想ブリッジや仮想ルータを挟むことで、仮想ネットワーク形態が変わる

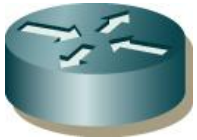


3-7. 仮想スイッチ、仮想ルータ

仮想スイッチ：Virtual Switch (vSwitch)



仮想ルータ：Virtual Router (vRouter)



仮想ネットワークにおいて、それぞれ物理機器と同じ役割を果たす
仮想ネットワークをNATにする場合はvRouterが
VLANを作成する場合はvSwitchが必要

3-8. 演習

演習1：VirtualBoxによるVM作成

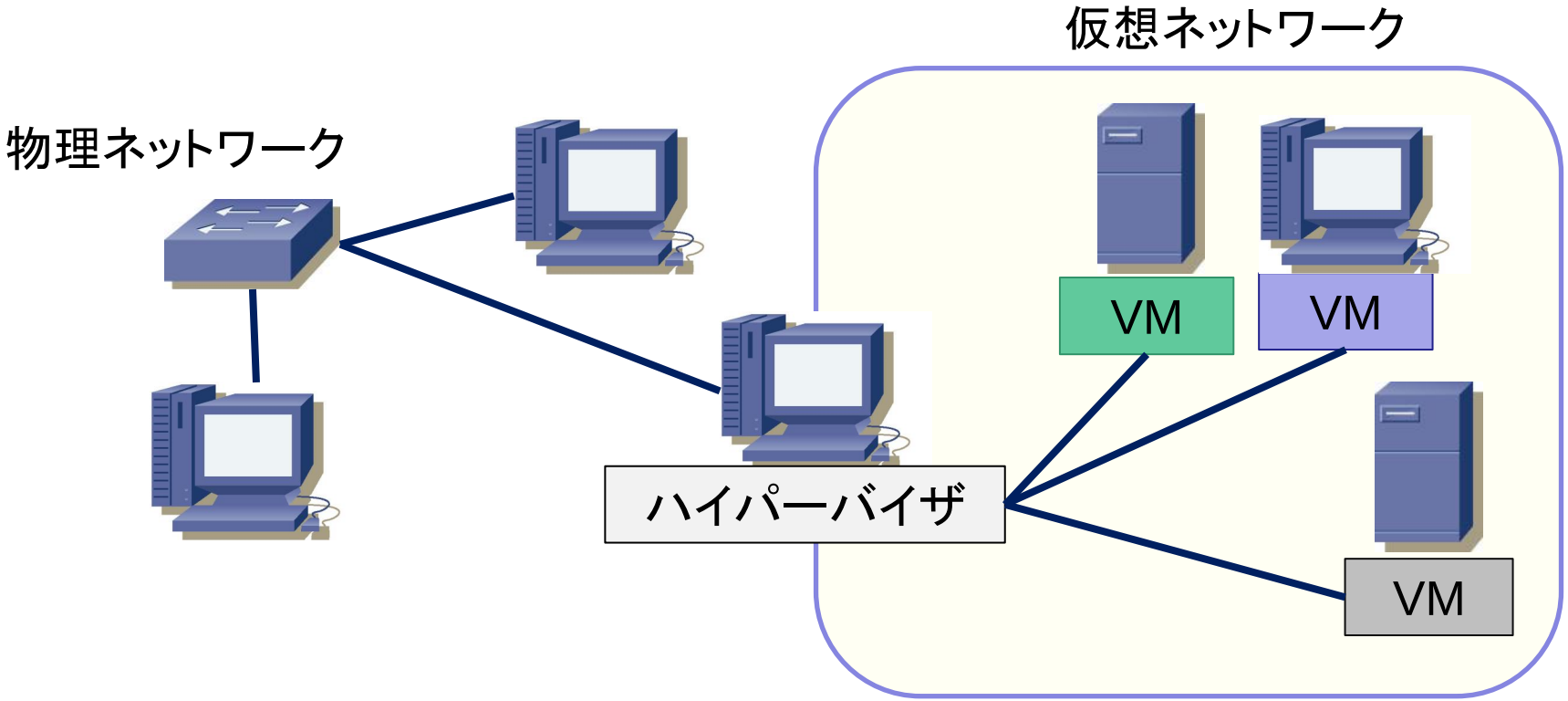
第4章 仮想化ネットワーク

仮想化特有のネットワーク構造について理解する

4-1. 仮想ネットワークの概要

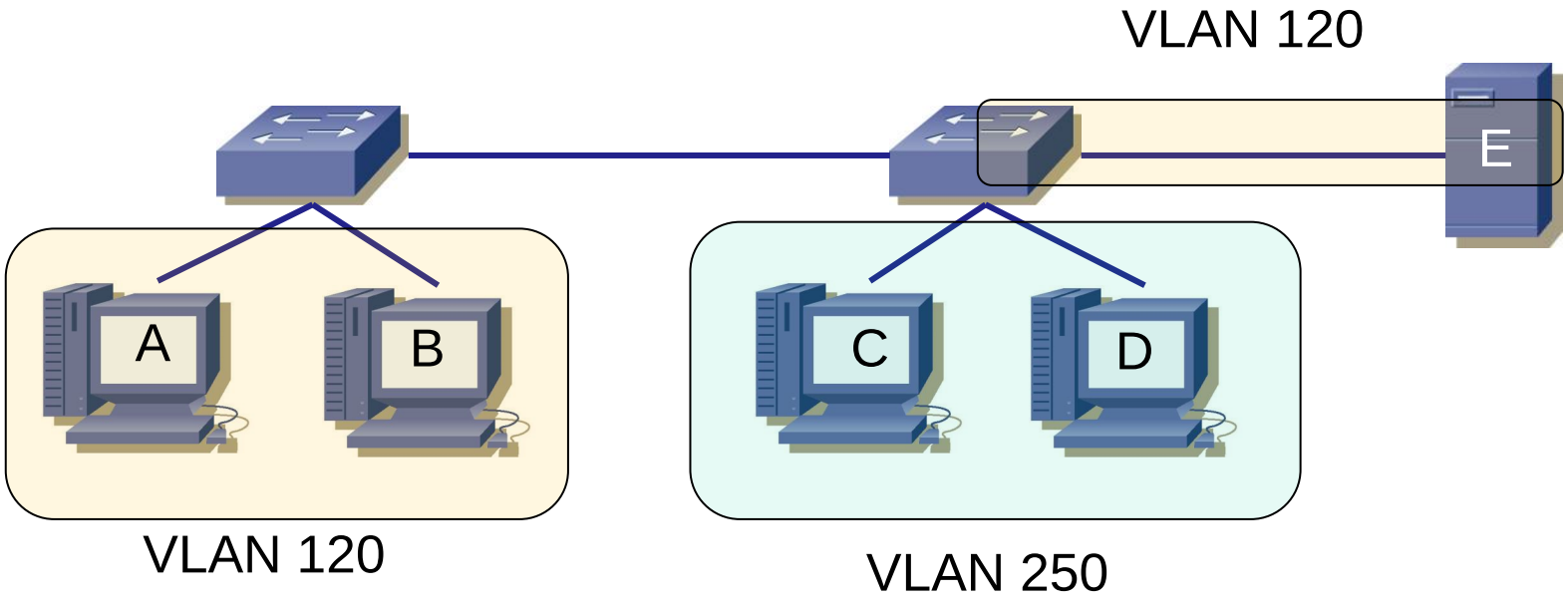
ハイパーバイザによって仮想的なネットワークの作成が可能

ハイパーバイザの設定によって、仮想ネットワークを独立させたり、物理ネットワークと接続したり、様々なトポロジ構成が可能



4-2. ネットワークとVLAN

データリンク層（Ethernet）は本来ネットワークを分割できない
VLAN対応スイッチによって提供されるVLANによって分割可能
仮想ネットワーク内でも仮想スイッチによるVLAN分割が可能



4-3. 仮想NICとネットワーク形態

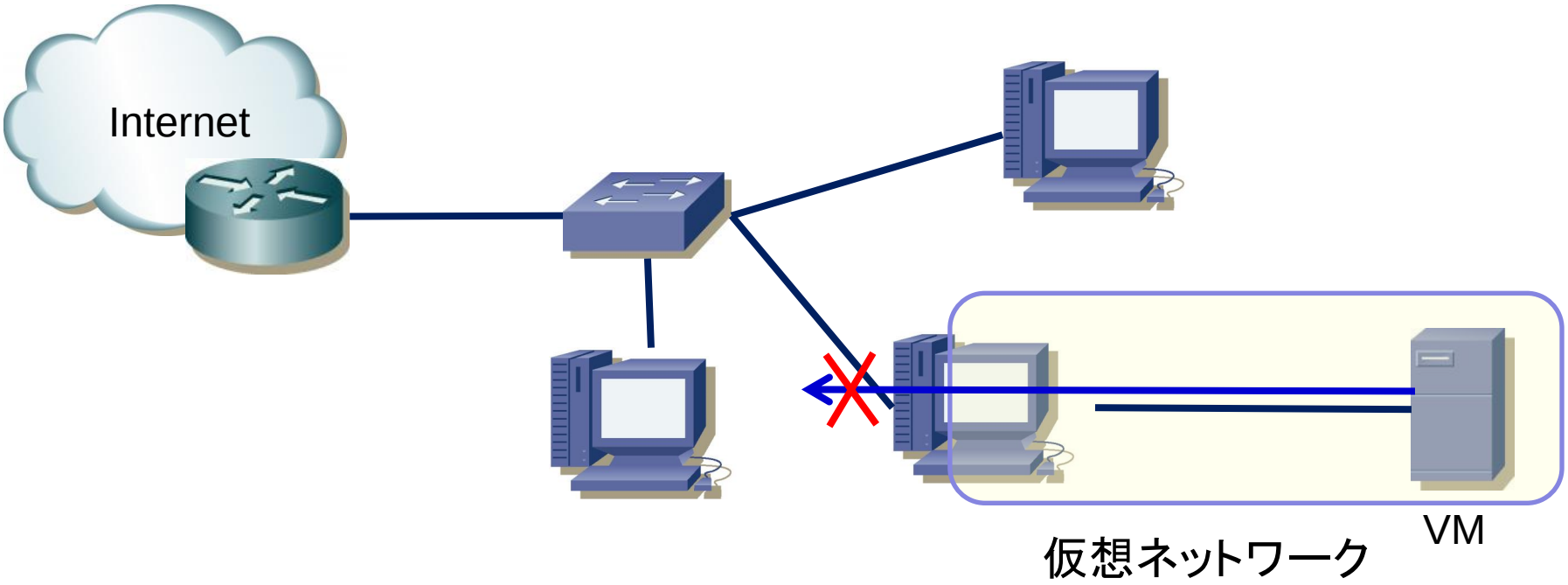
ハイパーバイザによって仮想的なインターフェイスである仮想NICが作成され、仮想ネットワークも作成される

次の3形態

- Host-only
- NAT
- Bridge

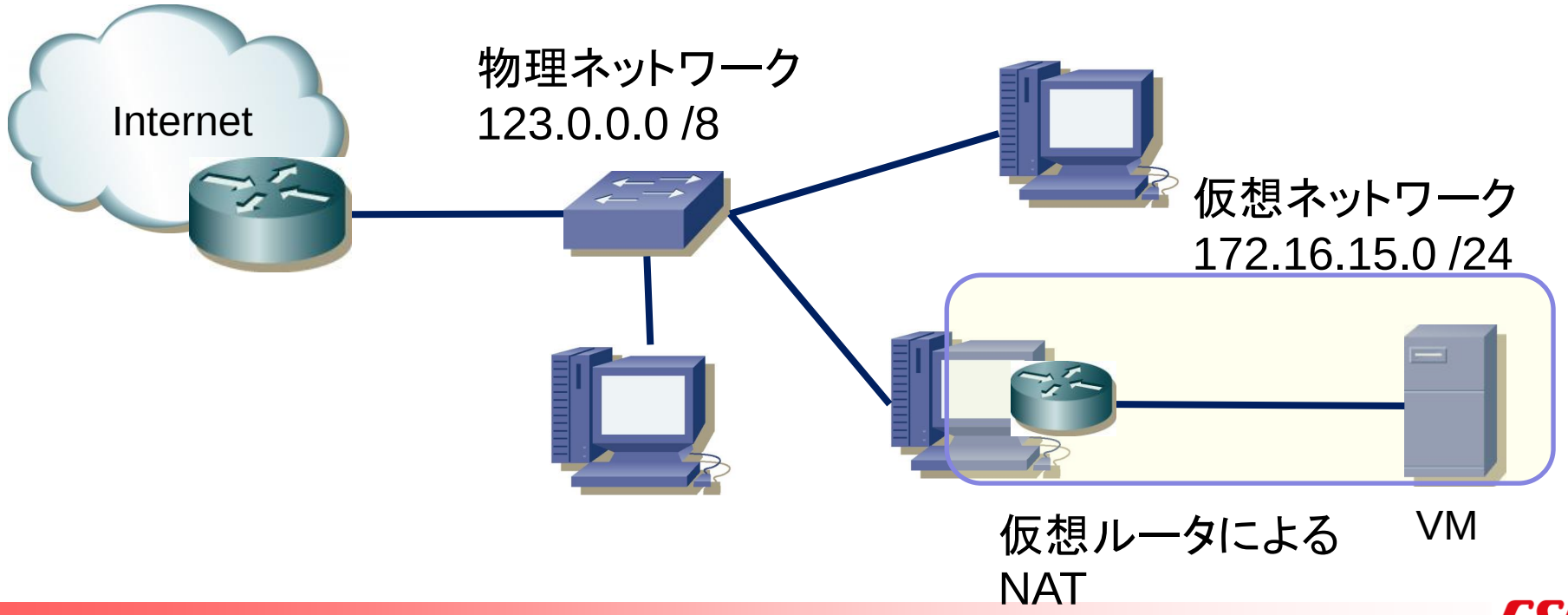
4-4. 仮想ネットワーク Host-only

ハイパーバイザが動作しているホストとのみ通信できるネットワーク
実ネットワーク内の物理マシンや外部ネットワークとは通信できない



4-5. 仮想ネットワーク NAT

仮想ネットワークにはプライベートアドレスが与えられる
ハイパーバイザが提供する仮想ルータによってアドレス変換が行われる
物理ネットワークに影響が少ない

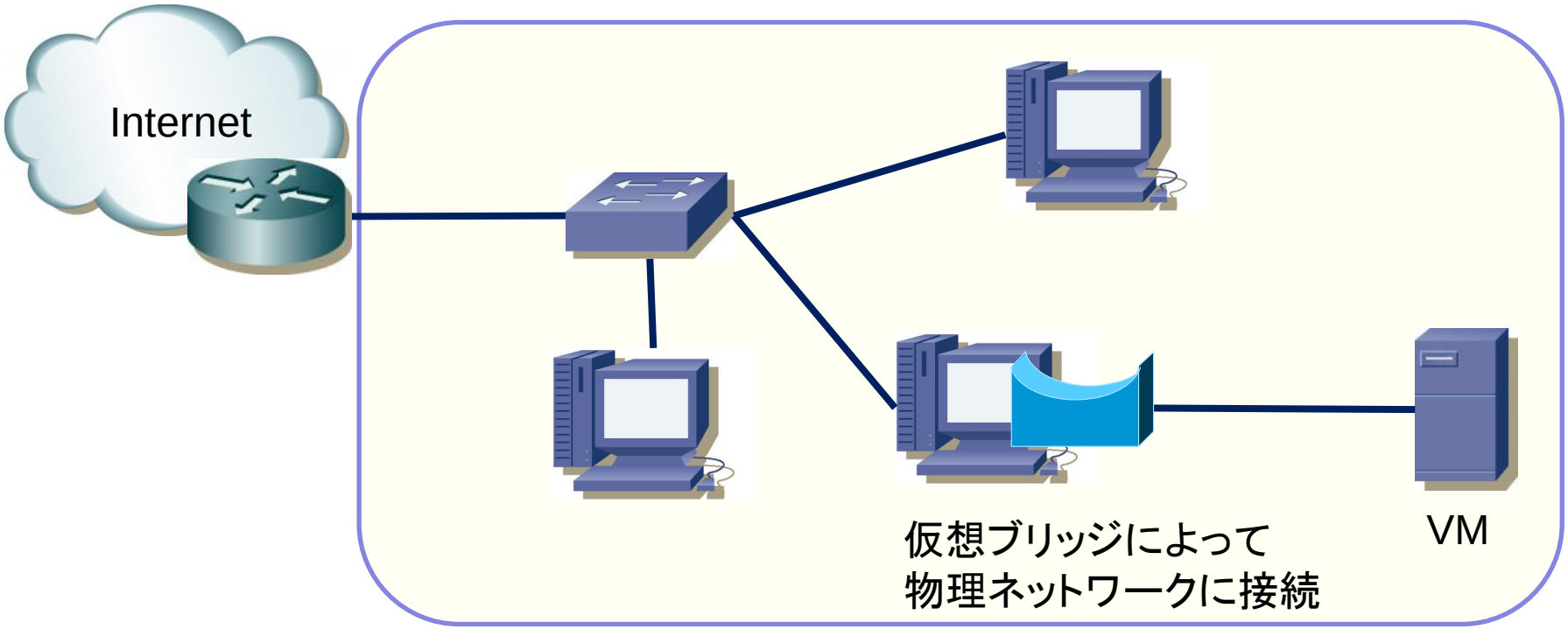


4-6. 仮想ネットワーク Bridge

ハイパーバイザーの提供する仮想ブリッジ（仮想スイッチ）を介して実ネットワークに直接接続

物理マシンと直接通信可能

物理ネットワークに影響を与えるおそれがある



4-7. 演習

演習3：仮想ネットワーク設定

第5章 仮想環境の運用

支援ツールや移行について学ぶ

5-1. 仮想環境の運用

ハイパーバイザの操作：GUI or コマンド
ハイパーバイザ独自のシェル環境など
Hyper-VはPowerShellによるコマンド、Hyper-Vマネージャ

5-2. 様々な構築支援ツール

実験的に1台構築するだけでも、様々な設定を手動で行うのは煩雑
仮想環境を自動構築するための支援ツールを使用

- Vagrant：仮想環境構築ツール
- Puppet：構成管理ツール
- Chef：サーバ設定・更新自動化ツール
- libvirt：仮想化環境共通API群

5-3. Vagrantの概要

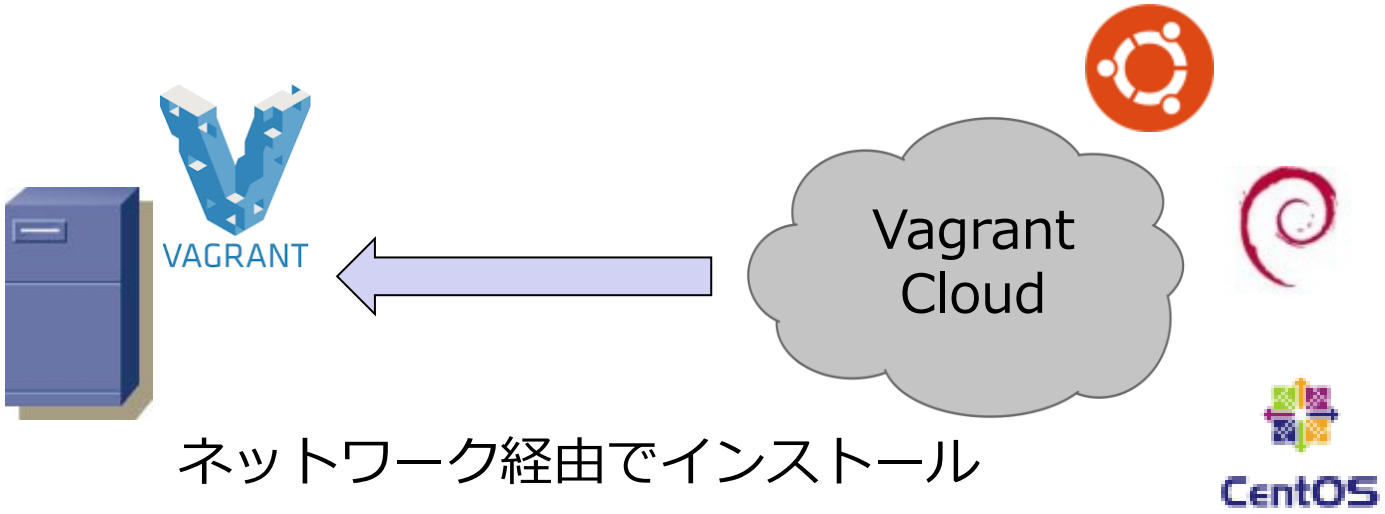
オープンソースの仮想環境構築ソフトウェア、MITライセンス
ファイル(Vagrantfile)に設定を記述し、仮想環境を自動的に構築
VirtualBox, VMware, Hyper-Vなどに対応
Amazon EC2といったクラウドにも対応
コンテナDockerにも対応している

Vagrantfileの例

```
config.vm.box = “generic/ubuntu1604”          #仮想イメージ:ubuntu16.04
 config.vm.provider “hyperv” do |h|           #プロバイダ:Hyper-V
  h.cpus = “2”                                #仮想CPU:2
  h.maxmemory=”4096”                          #最大メモリ:4096M
  h.enable_virtualization_extensions=true     #仮想化支援機能:有効
 end
```

5-3. Vagrantの概要(続き)

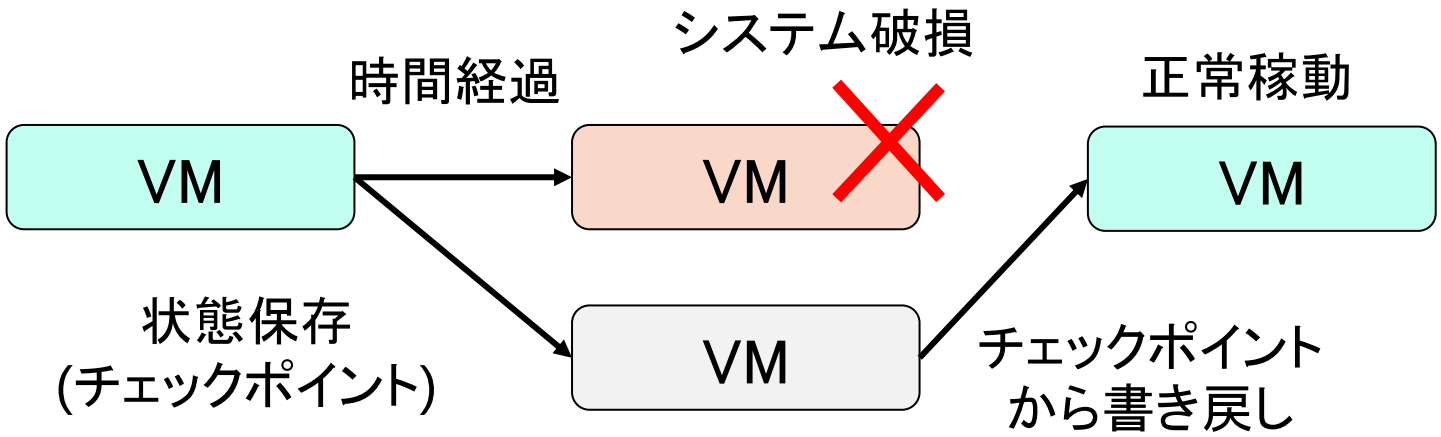
仮想イメージ：Box, 動作ハイパーバイザ：プロバイダ
コマンドでVagrant CloudからBoxを導入可能
https://app.vagrantup.com/
独自Boxの作成も可能



5-4. 運用とバックアップ

仮想マシンの運用には、Vagrantなどの運用ツールの他、ハイパーバイザ用のシェル言語などもある。例) Hyper-VはPowerShell

仮想マシンなら、ファイルベースなのでバックアップも容易
Hyper-Vではある状態を保存することをチェックポイントと呼ぶ



5-5. 移行（マイグレーション）

現在の環境をそのまま仮想環境へ移す、あるいは新規に仮想マシンを作成する、など様々な移行方法がある

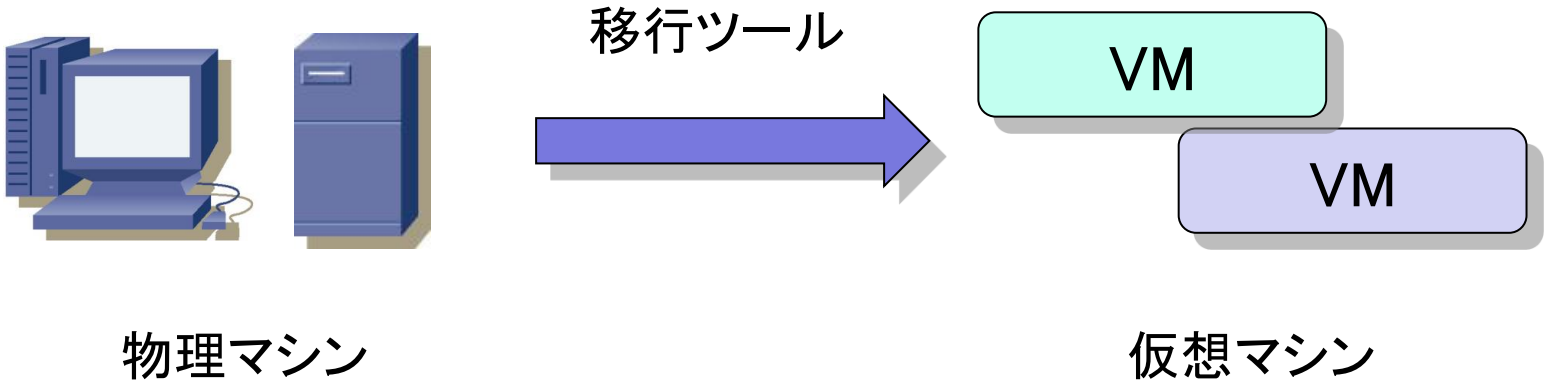
仮想マシンは物理マシンと完全に同じではない

ネットワークやハイパーバイザーによっては、仮想環境に対応していないOS、アプリケーション、ハードウェアがある

仮想環境、クラウドで使用するすべてのハードウェア、ソフトウェアについて対応状況をチェックする

5-6. P2Vマイグレーション

Physical to Virtual : P2V
現在動作している物理マシンを仮想マシンへ移行するツールを使う
無停止で行うホットクローニング
物理マシンのシャットダウンを伴うコールドクローニング

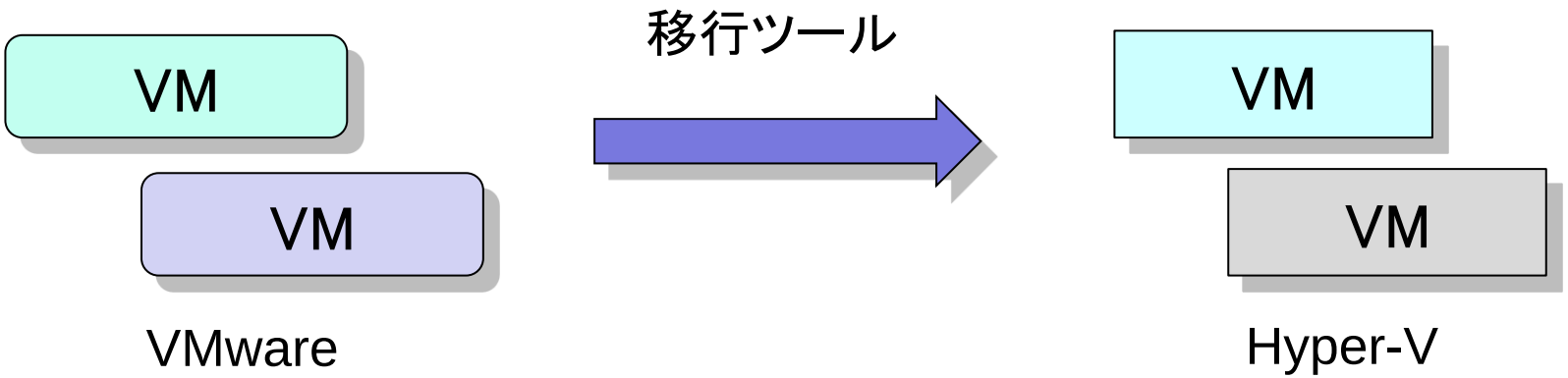


5-7. V2Vマイグレーション

Virtual to Virtual：V2V

ハイパーバイザによって仮想マシンのフォーマットが異なるため、変換が必要

多くの場合互換性が有り、変換ツールも用意されている

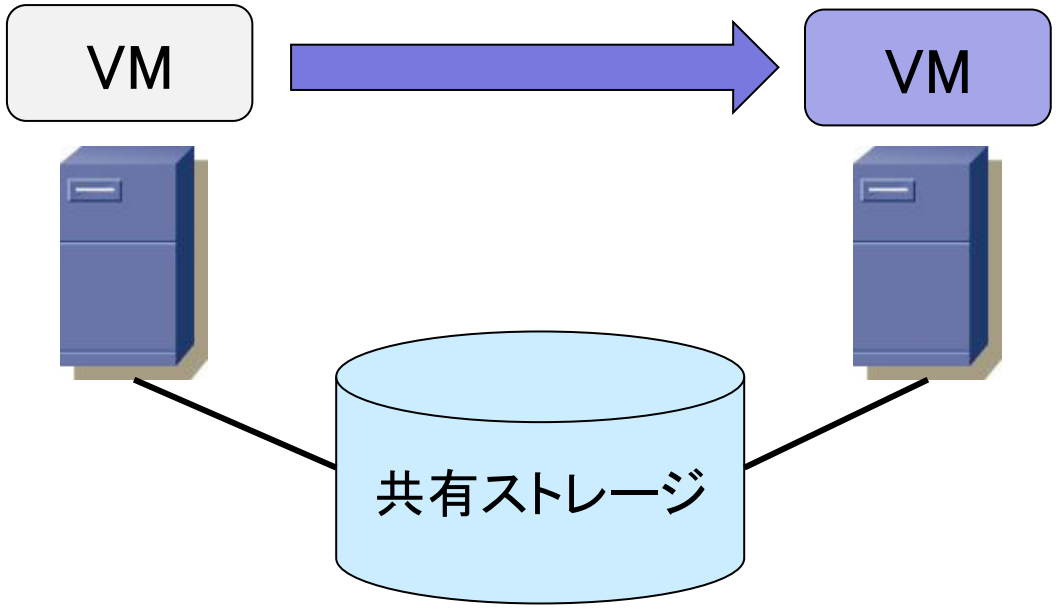


5-8. ライブマイグレーション

仮想マシンをある物理サーバから他の物理サーバに移行する際に、ノンストップで行う技術がライブマイグレーション

VMwareがVMotionとして実装。現在のハイパーバイザはほぼ対応している

VMのストレージに共有ストレージを用いることで実装されている



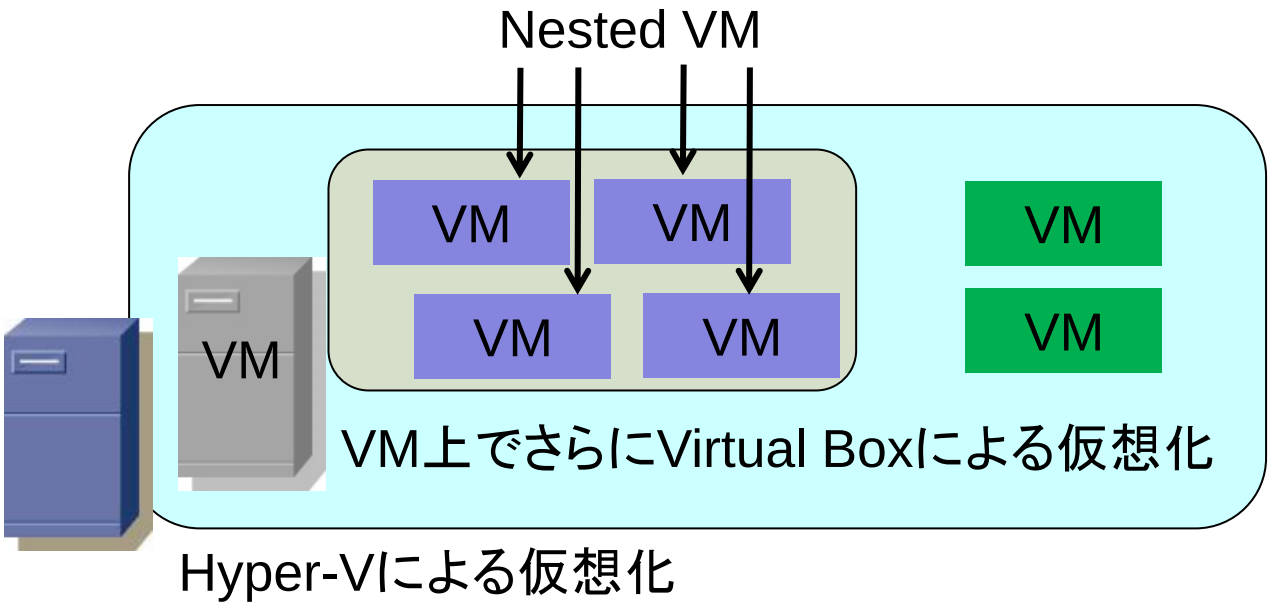
5-9. Nested VM

仮想マシン上でさらに仮想マシンを動かすこと、またはそれによって稼動しているVMのこと。入れ子VMなどとも呼ぶ

主要ハイパーバイザは対応済み。ただしオプション扱いも

Hyper-VではPowerShell上から以下のコマンドが必要

Set-VMProcessor -VMName <VM名> -ExposeVirtualizationExtensions $true



第6章 コンテナの概要

コンテナシステムについて概要を
理解する

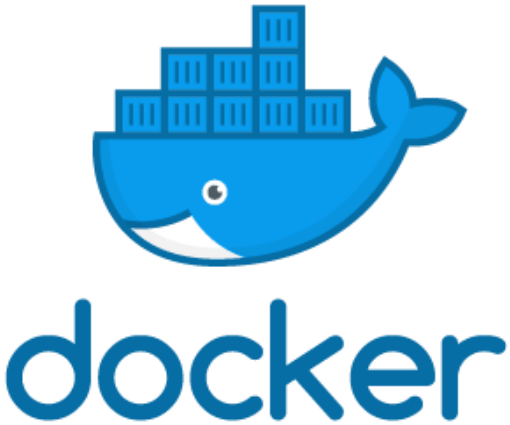
6-1. コンテナとは

Linuxカーネルの「コンテナ」機能を利用した分離環境

プロセスのように、コンテナ分離

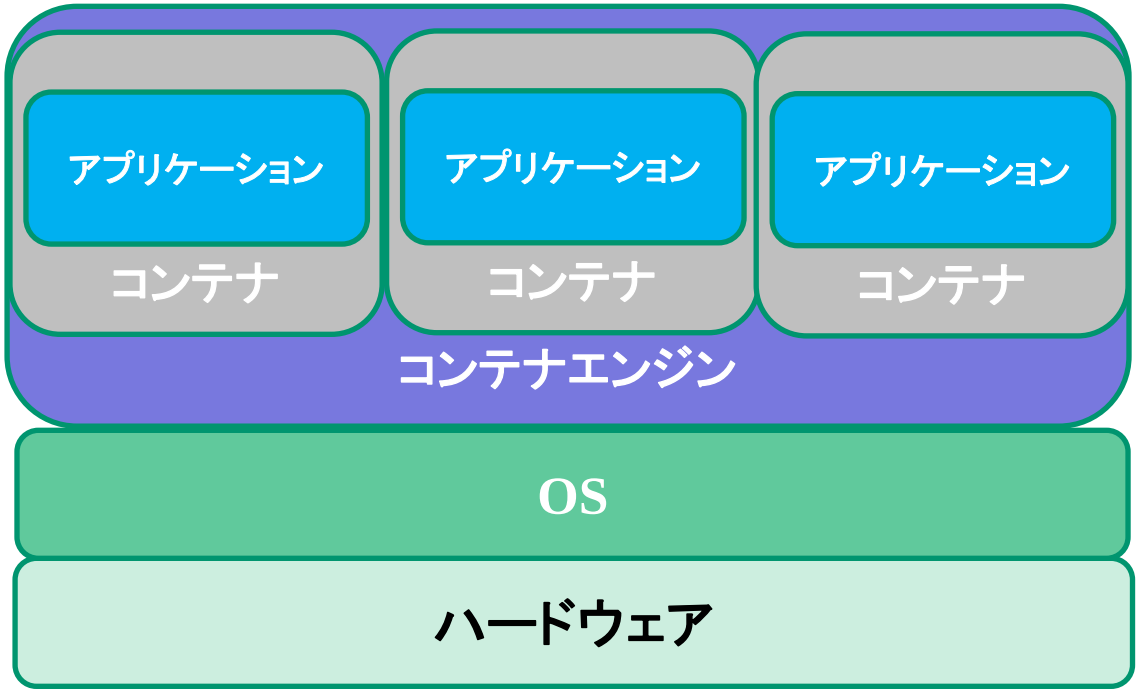
ホスト名、ファイルシステム、ユーザ名、プロセスID、ネットワークなどを、コンテナごとに独自設定可能

オープンソースのコンテナエンジンDockerが人気



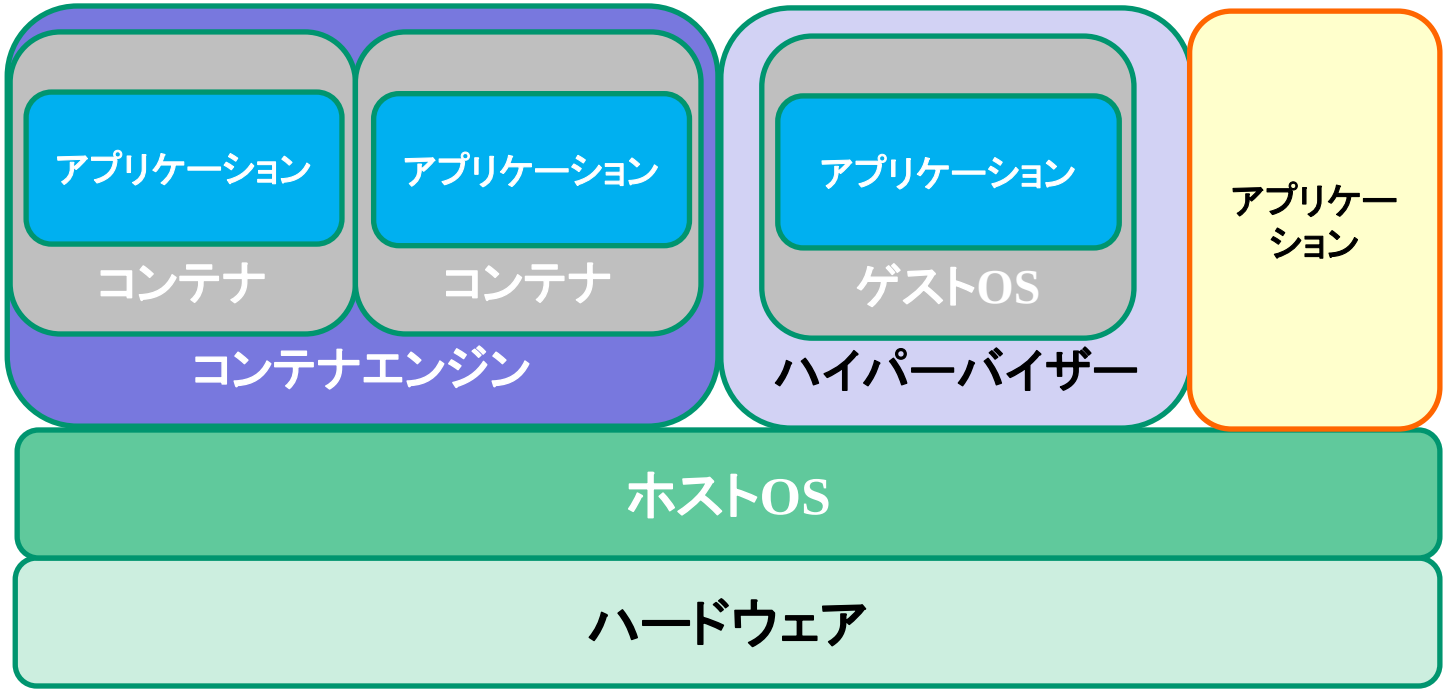
6-2. コンテナの動作

アプリケーションはコンテナ単位で独立
1つのOS上に複数のアプリケーションコンテナが動作
ハードウェアリソースの消費が非常に少ない
現在多く使用されているコンテナエンジン：Docker



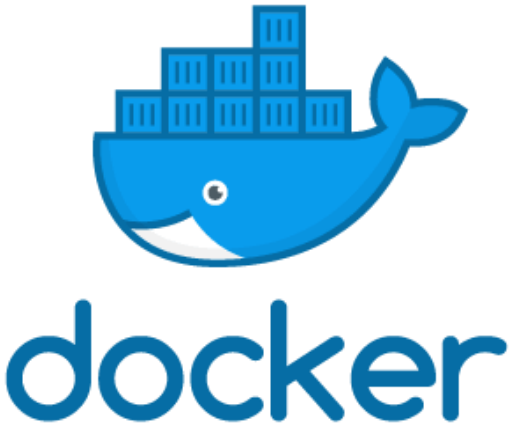
6-3. 仮想化との違い

VMはゲストOSの容量が必要、ハードウェアのオーバーヘッド大
コンテナはOSやハードウェアリソースはホストOSと共通
VM内アプリケーションはすべて同一ネットワーク
コンテナはそれぞれ隔離され、ネットワークも独立可能



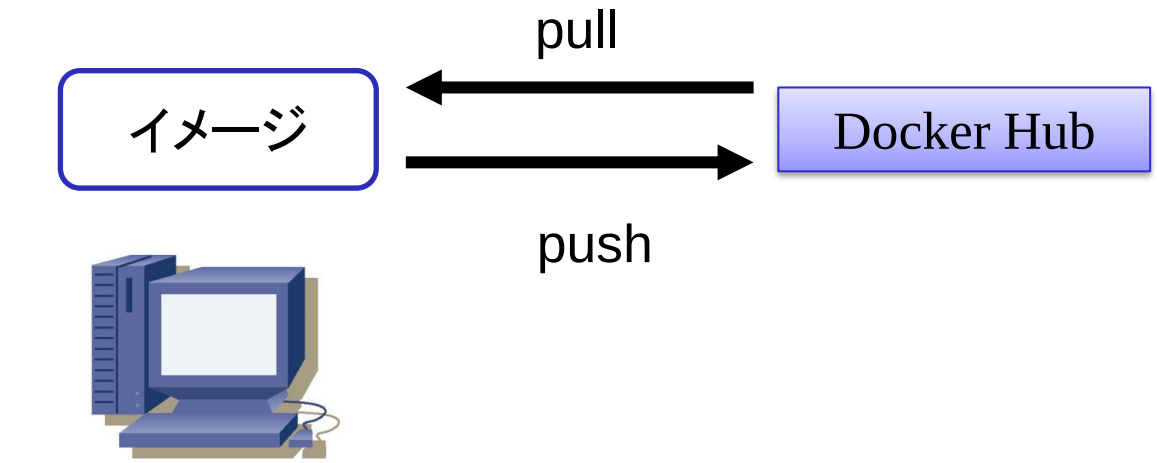
6-4. Dockerの概要

Docker社によるオープンソースのコンテナエンジン
無償版：Docker Community Edition (Docker CE)
商用版：Docker Enterprise Edition (Docker EE)
基本はLinuxだが、Windows, macOSにも対応



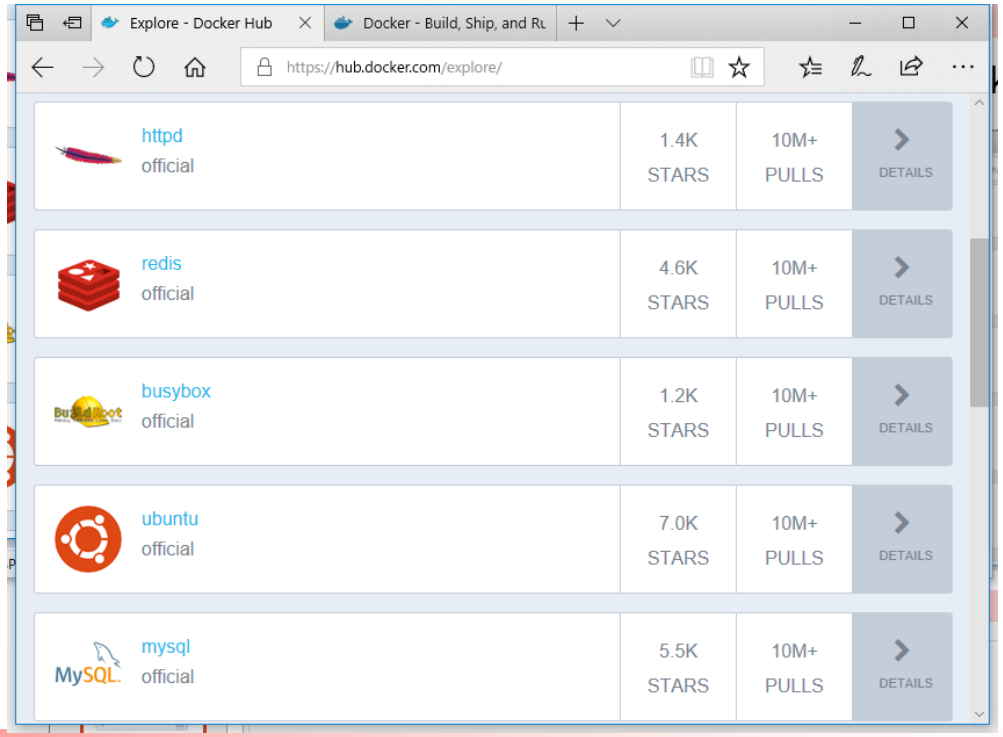
6-5. コンテナの概要

コンテナはDocker Hubに登録されている
コマンドでDocker Hubからダウンロードし実行 (pull)
実機内にはローカルレポジトリが構築される
自分が作成したコンテナイメージをDocker Hubにアップロード可能 (push)



6-6. Docker Hub

ユーザが作成したコンテナイメージを自由に公開・共有できるサービス
ローカルにないコンテナイメージはここからダウンロードして実行される
利用は無料だが、アップロードしたコンテナイメージは原則公開
商用版(Docker Enterprise Edition)の利用なら、非公開のプライベートHubとして利用可能



6-7. DockerとOS

DockerはLinuxのコンテナ技術を使用している
そのため、Dockerの対応OSはLinux
WindowsやmacOSでも動作するが、仮想環境と組み合わせている

Windows: Hyper-V上にLinuxを作成し、その上でコンテナを構築
Hyper-Vが必要なため、Windows 10 Pro以上で動作

macOS: 10.10.3(Yosemite)以降に搭載された仮想環境フレームワーク
Hypervisor.frameworkを利用して、Linuxを作成。その上でコンテナを構築

6-8. コンテナの用途

コンテナは仮想化と異なりコンパクト、起動も速い
アプリケーション単位で隔離されるため、仮想化で実現するには大がかりなことが簡単に行える

同一ツールの複数バージョンを同居：Pythonのバージョン違いを同居
分離されたデーモンプロセスの複数起動: httpdの複数起動
開発環境や検証環境をコンテナで構築し、配布することで環境の同一を保つ
……他にも様々な用途がある

6-9. コンテナのサイズ

Ubuntuのコンテナは111MByte

“Hello from Docker!” と表示するだけのhello-worldは、わずか1.85kByte

ベースとなるOSとの差異がアプリケーションコンテナとして格納されているので、一般にコンテナのファイルサイズは小さい

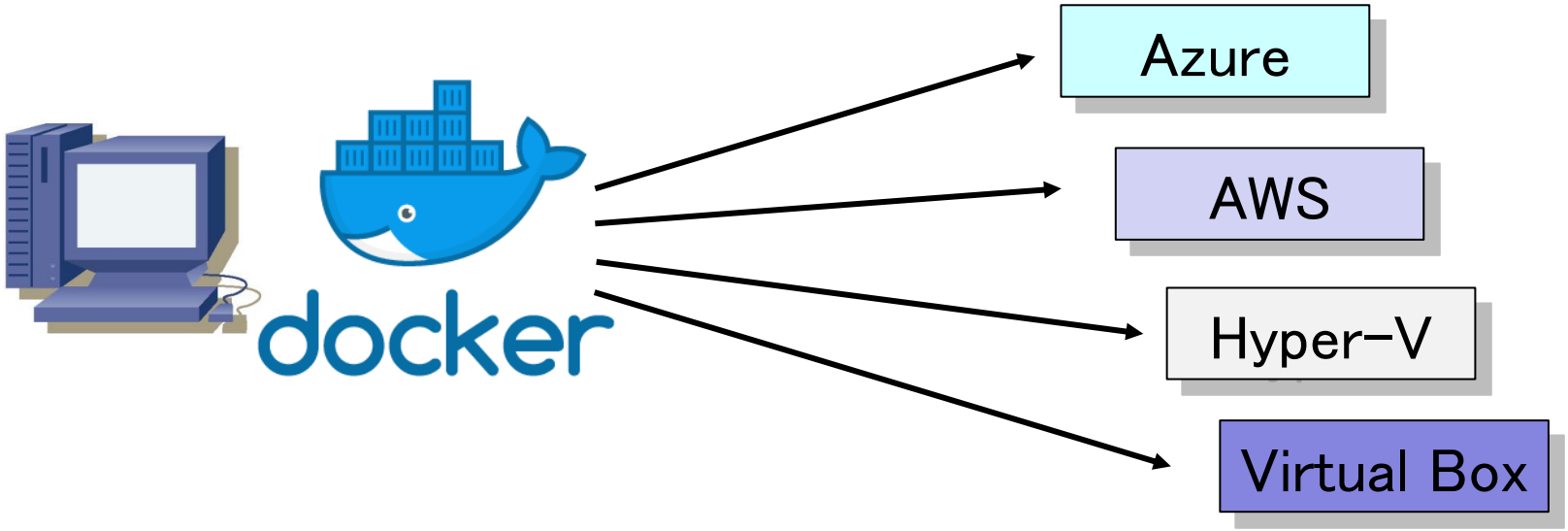
```
$ docker images
REPOSITORY      TAG      IMAGE ID          CREATED          SIZE
ubuntu          latest   00fd29ccc6f1      11 days ago      111MB
hello-world     latest   f2a91732366c      5 weeks ago      1.85kB
```

6-10. Dockerのプロビジョニング

プロビジョニング：必要に応じてコンピュータ・リソースを提供・準備すること、または自動構築すること

Docker Machineによって仮想環境のHyper-VやVirtual Box, クラウド環境のAmazon EC2やMicrosoft Azureなどに展開可能

ローカルだけではなく、クラウド環境でもサポートされている



第7章 コンテナの実践

Dockerを使用してコンテナを学ぶ

7-1. 演習

演習4：Dockerのインストール&Hello Docker
演習5：コンテナのカスタマイズ、独自コンテナの実施
演習6：Docker Hubへ独自コンテナの公開