

仮想化

仮想化は、1章から3章までがE-Learning対象です。

講座では1章から3章までは復習として簡単に触れますが、メインは4章以降です。また、2章のIPアドレスについては、もう少し詳細に講座内で解説します。

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. LAN17
2-3. WAN18
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

第1章 仮想化概要

仮想化の基本を学ぶ

1章, 2章はe-Learningとしても提供するが、講座では一通り流して説明する。

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

・説明の流れ

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

・ポイント

仮想化は従来の物理環境とはまったく異なるということ

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

・ポイント

仮想化の種類によって大きく4タイプに分かれるということ

・補足説明

サーバ仮想化：1台のコンピュータをリソース分割し、複数のコンピュータとして振る舞うこと。様々なOSやアプリケーションを同時に実行できる。需要に応じてプロセッサやメモリ、ストレージなどのリソースを自在に変更できる。

デスクトップ仮想化：デスクトップ上に他のOS環境を作成し、同時に実行できる。主に実験環境や一時的な開発環境の構築に用いられる。

ネットワーク仮想化：ネットワークを物理的な配線に縛られず、ソフトウェアによって自由に変更できる。近年はネットワーク仮想化のOpenFlowプロトコルに対応した安価な製品が多く販売されている

。

ストレージ仮想化：サーバ仮想化と同じく、ストレージを論理分割してあたかも複数のディスクや異なった種類のストレージのように振る舞うことができる。また、他の仮想化と同じく、需要に応じて自由にリソースの変更も可能。

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

・ポイント

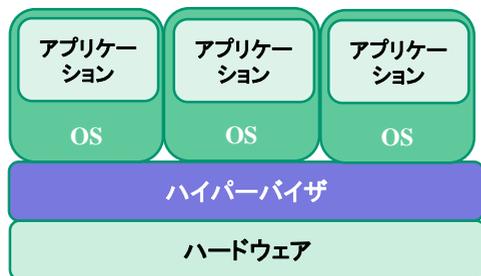
具体的な商品・サービスをここで紹介

・質問

これらの製品を使ったことがありますか？ また、名前を知っていますか？ と問いかけ。

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



・説明の流れ

図を示し、ハードウェア上で複数のシステムが動作することを中心に説明。

・補足説明

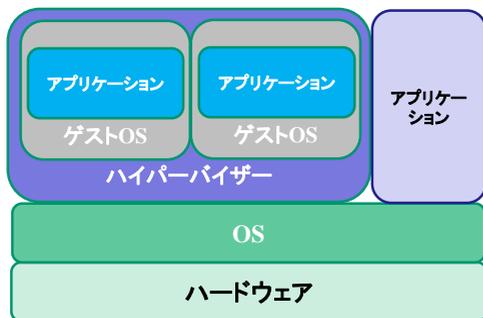
サーバ仮想化は、Type-I型、ネイティブ型、ベアメタル型、とも呼ばれる

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ーが動作

ハイパーバイザー上で仮想環境が動作



・補足説明

デスクトップ仮想化は、Type-II、ホストOS型、とも呼ばれるWindowsを普通に使用しているときに平行して他OSを使用できるなど、具体例を示す。

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

	suse-sles-12-sp3-v20171212-hvm-ssd-x86_64 - ami-8368cefb SUSE Linux Enterprise Server 12 SP3 (HVM, 64-bit, SSD-Backed) ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい
	RHEL-7.4_HVM_GA-20170808-x86_64-2-Hourly2-GP2 - ami-9fa343e7 Provided by Red Hat, Inc. ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい
	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20171121.1 - ami-0def3275 Canonical, Ubuntu, 16.04 LTS, amd64 xenial image build on 2017-11-21 ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい
	Windows_Server-2016-English-Full-Base-2017.11.29 - ami-f6d8008e Microsoft Windows Server 2016 with Desktop Experience Locale English AMI provided by Amazon ルートデバイスタイプ: ebs 仮想化タイプ: hvm ENA 有効: はい

・ポイント

クラウドサービスの基盤上で仮想化が使用されていることを説明。

ここで示している図は、AmazonのクラウドサービスAmazon EC2でテンプレートとして用意されている仮想マシンの一例。

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

・ポイント

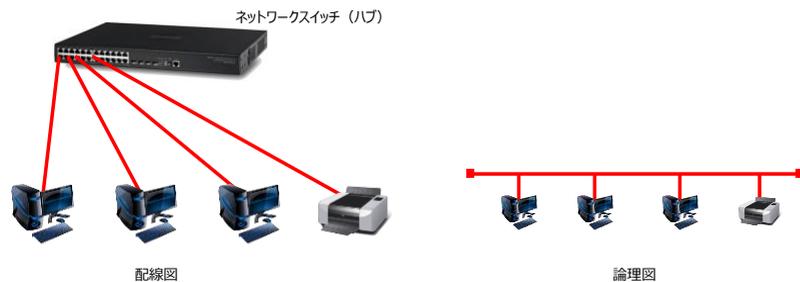
仮想化はネットワークと密接に関わってきます。

ここから先はネットワークについて少し説明しますが、ノード、リンク、エッジ、またホストなどの用語をしっかりとここで押さえてください。

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



・説明の流れ

おそらくLANという言葉聞いたことのない方はいないと思いますが、社内LANとか言いますよね、社内LANありますよね、と問いかけて説明してください。

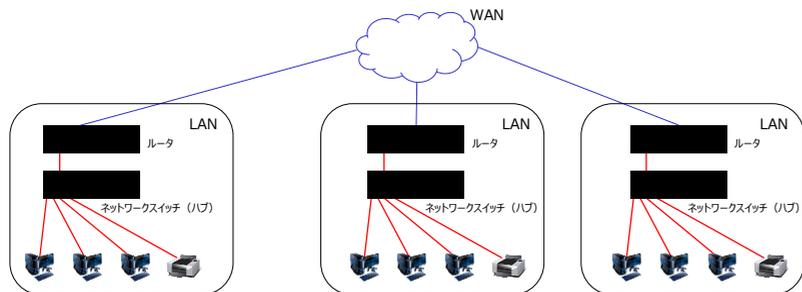
・ポイント

LANとWANはよく対比して説明されます。次のページのWANとともに受講者に説明してください。

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ことが多い。
WANを世界規模で実現しているのがインターネット。



・ポイント

LANとWANはよく対比して説明されます。前のページのLANとともに受講者に説明してください。

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

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

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

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

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

・説明の流れ

IPアドレスは聞いたことありますか、と問いかけ、そこから識別のためのもの、ユニークなもの、ネットワーク部、ホスト部、と分かれることを説明してください。

IPアドレス

192.168.1.10

1100 0000 1010 1000 0000 0001 0000 1010

2進数の並びが $8 \times 4 = 32$ ｺ



32bit

※ 8bitのかたまり = 1オクテットと呼ぶことも octet

・ポイント

IPアドレスは2進数で扱うと計算が楽なので、2進数表記もときおり使用されます。

・補足説明

2進数になじみのない方がいる場合は、2進数の説明を少しした方がよいでしょう。

192.168を「いちくに いちろっぱ」とよく呼ぶので、そのことも合わせて補足説明した方が演習等でのアドレスの読み方も参考になります。

IPアドレス (続)

- 1オクテットの範囲

0000 0000 すべて 0 ~ 1111 1111 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。

・説明の流れ

このスライド含め、3枚はIPアドレスの基本的事項の確認です。
やや時間をかけて説明してください。

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

8bit : ホスト部 →

ネットワークプレフィックスと言うことも、
network prefix

ネットワーク内の端末を
識別

端末がどのネットワークに所属しているのかを判別。

・説明の流れ

色を付けてわかりやすく分離していますが、/を付けて、ネットワーク部のビット数を表記することをプレフィックス表記と呼びます。

ここでは頭から24bit分がネットワーク部なので、/24となります。

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

・ポイント

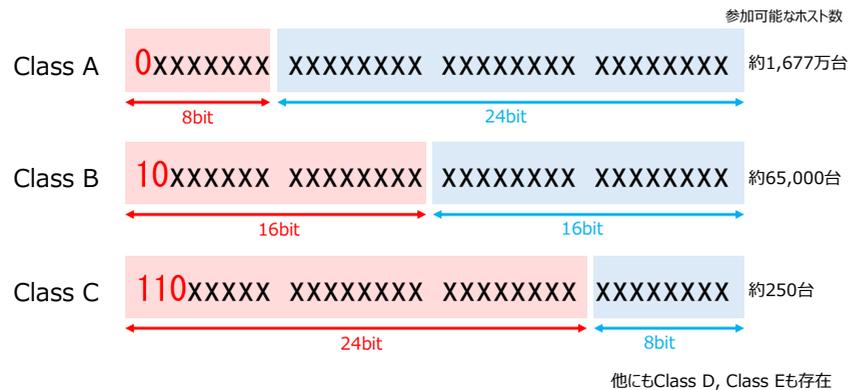
ネットワーク内では、すべてのIPアドレスがホストに割り当てられるわけではありません。

スライドにも書いてあるように、ネットワークの一番最初のアドレスはそのネットワーク自身を指すネットワークアドレス、一番最後は、ネットワーク全体にデータを送信するブロードキャストアドレスとして使用されるため、全アドレス数から-2の数がホストへの割り当て数になります。この例では、192.168.1.1から192.168.1.254までをホストに割り当てられます。

また、192.168.1.10の属するネットワークは、192.168.1.0になります。（ネットワークアドレスはこのようにして使用します）

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



・説明の流れ

IPアドレスのクラスによって、ネットワーク部、ホスト部が異なることを説明し、その結果クラスによって何が変わるのかを説明します。

ホスト部を広くとれるということは、すなわちネットワーク内のホストをより多く収容できることを併せて説明してください。

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

・ポイント

10進・2進変換をここで実際に手計算し、先頭2ビットを見てクラスBであることを再確認してください。

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

・ポイント

ここに挙げたコンポーネントはすべて物理的なリソースです。

仮想化とは、物理リソースを論理化するものだというをここで再確認してください。

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

📁 ubuntu14d	2017/11/29 22:41	ファイル フォルダ	
📄 ubuntu14-desktop.vhdx	2017/12/19 14:29	ハードディスク イメ...	9,441,280 KB

↑
仮想マシン(ハードディスクイメージ)

・補足説明

この図で示しているように、仮想マシンは単一、あるいは複数のファイルになり、それがイメージになります。

この例はHyper-Vのvhdx形式ですが、ハイパーバイザごとにフォーマットがあることも触れてください。

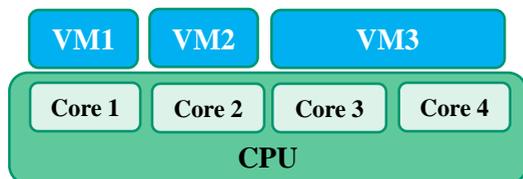
3-3. 仮想CPU

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



最近のCPUはデュアルコア、クアッドコア、さらにハイパースレッディングによって、1CPUでも論理コア数が4や8といった場合が多いので、物理CPU数とコア数は一致しないことに注意。

3-3. 仮想CPUと仮想化支援機能

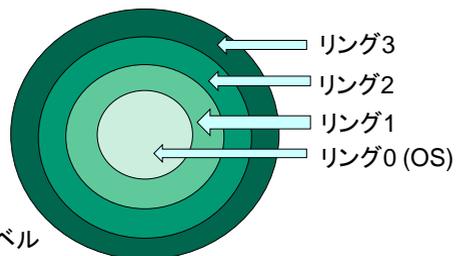
近年のCPUには、仮想CPUを物理CPUとほぼ同等に動作させる仮想化支援機能がある

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

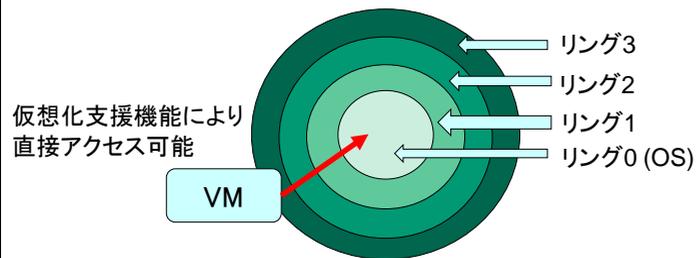
・質問

VT-xやAMD-Vという機能を聞いたことがありますか？と問いかけ。

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



・ポイント

仮想CPUは物理CPUとは異なるため、ネイティブの速度は出ないが、支援機能によって近い速度が出ることがポイントです。

・補足説明

最近のハイパーバイザは、仮想化支援機能が有効になっていないとインストールできなかつたり起動できなかつたりします。このことも併せて説明してください。

有効になっていない場合は、BIOSやUEFIから有効にします。

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

・ポイント

仮想化のタイプによって、仮想メモリがどれくらい使用できるかが異なります。

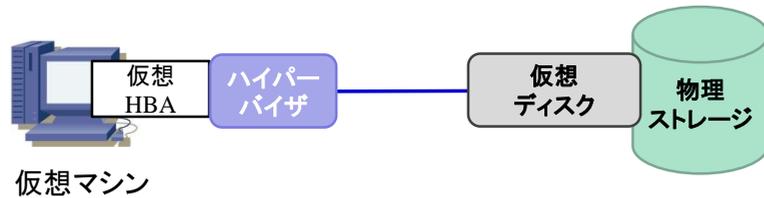
3-5. 仮想HBA, 仮想ディスク

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



・補足説明

HBAは仮想化関連でしかあまり出てこない用語です。

聞き慣れない人が多いと思いますが、スライドにあるようにPCのストレージインターフェイスのことだと説明すれば理解が早いでしょう。

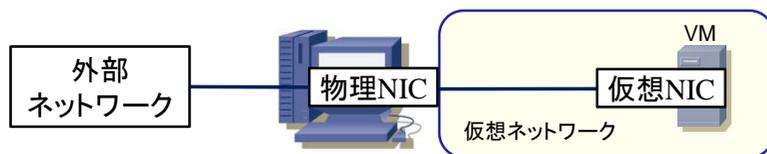
また、仮想ディスクは設定上20Gとしても、動的構成ならば実際に使用しているサイズしか使用しないこともポイントです。

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



・説明の流れ

今は無線が多いので、有線ネットワークの口を見ることがあまりないですが〜と物理NICのことをまず説明し、仮想ネットワークへの接続のために仮想NICが必要なことを説明する。

詳細は4章で、と前置きをして、仮想ブリッジや仮想ルータも使用することを触れる。

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

・質問

スイッチとルータの違いはわかりますか？ これらの機器の設定を行ったことがありますか？ と問いかけ。

ネットワーク機器について詳しい方が多ければ、4章の説明はさらっと流してよいでしょう。

・補足説明

スイッチはレイヤ2でネットワークを分割、ルータはレイヤ3でネットワークを分割しますが、ネットワークになじみのない人にはわかりにくいと思います。

IPアドレス上でのネットワーク分割はルータで行う、という事例を出しながら、少し補足説明をしてください。

3-8. 演習

演習1 : VirtualBoxによるVM作成

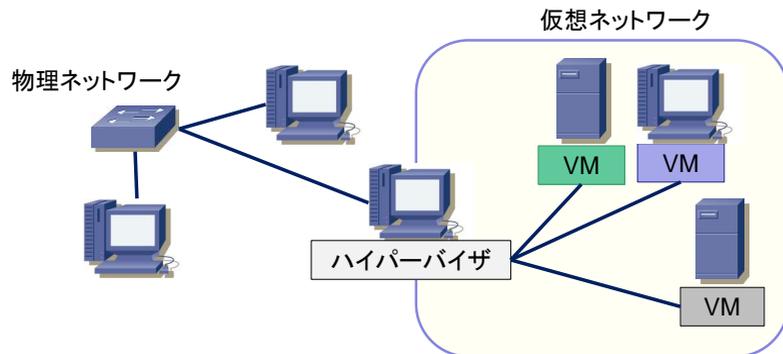
第4章 仮想化ネットワーク

仮想化特有のネットワーク構造に
ついて理解する

ここまで（1 - 3章）がE-Learningです。

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

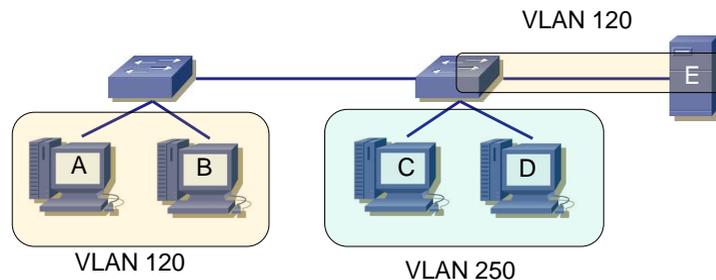


・ポイント

ここから、スライドでVM表記を多用します。仮想マシン=VMという認識を再確認してください。

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



・補足説明

必要ならばOSI 7層モデルを板書し、簡単に説明してください。

ユーザ側でVLANを意識することはあまりないのですが、自分で仮想ネットワークを構成するときを使用することがたまにあります。以下は、VLANについての詳細説明です。

VLANはVLAN対応スイッチによって論理分割されるネットワークです。スライドでも説明してあるように、本来スイッチはデータリンク層のため、レイヤ3での（IPアドレスでの）ネットワーク分割はできません。

ネットワーク分割はネットワーク層のルータによって行われますが、例外的にVLANによってネットワーク分割ができる。VLANは番号によって管理され、同じVLAN番号のホストは通信可能。異なったVLAN番号のホスト間は通信できません。

VLANを越えて通信するには、ルータによるルーティングが必要です。この例では、ホストA,B,E間はVLANが同じのため互いに通信できま

すが、ホストC,D間とはVLANが異なるため通信できません。

現在の主要なハイパーバイザは、VLAN番号を付与した通信（VLANタギング）が可能なので、VLANによるネットワーク分割をそのまま仮想ネットワークでも適用できます。

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

・ポイント

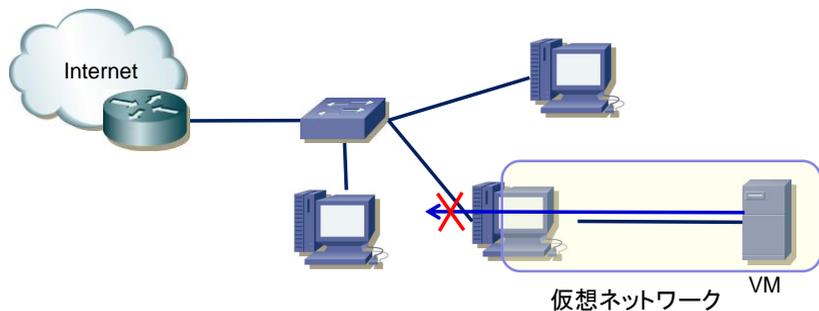
一般的な仮想ネットワークとしてこの3形態があることを押さえてください。

・補足説明

Type-I型ではブリッジが、Type-II型はNATかHost-onlyがデフォルトであることが多いです。また、ハイパーバイザによっては、この3形態すべてをサポートしていなかったり（Hyper-VはNATをオプション扱い）あるいはより細かいネットワーク形態を取っている場合もあり、必ずしもこの3形態とは限りません。

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



・説明の流れ

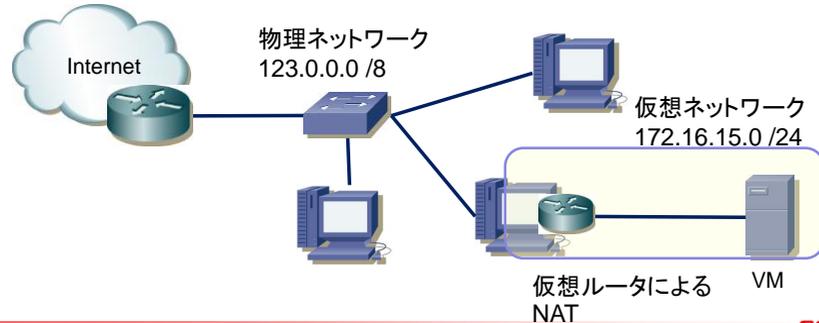
ここから、Host-only, NAT, Bridgeについて説明します。VMと物理ネットワークの接続性について着目して説明します。

・ポイント

VMはハイパーバイザが稼働しているホストとのみ通信可能です。もし複数VMが稼働していると、各VM間は通信できません。

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



・ポイント

VMにはプライベートアドレスが割り当てられるということ、物理ネットワークに影響が少ないことがポイントです。

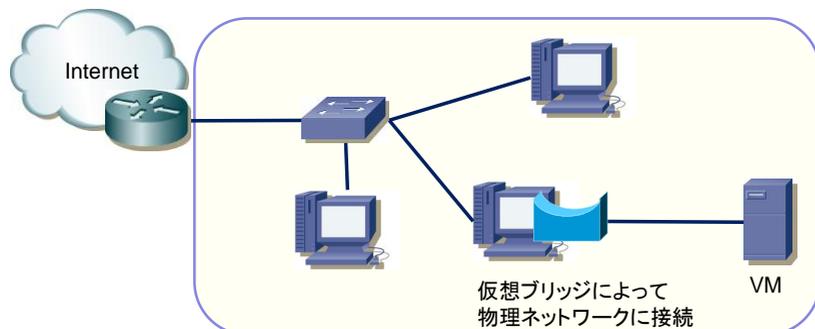
Type-II型のハイパーバイザはこの形態がデフォルトであることが多いです。

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



・ポイント

Bridge構成にすると物理ネットワークとホストを經由して直接つながりません。

Type-I型のハイパーバイザではこの形態がデフォルトであることが多いです。

4-7. 演習

演習3：仮想ネットワーク設定

演習ではここまで紹介した仮想ネットワークを実際に体験します。

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



・補足説明

Type-I型のハイパーバイザでは、多数のVMを管理することが多く、管理ツールが充実しています。

ここではHyper-Vの例を紹介していますが、VMware SphereではWebブラウザからハイパーバイザにアクセスし、様々な設定を行うことができます。

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



・説明の流れ

これらのツールを見聞きしたことはあるか、参加者に質問します。また、使ったことがあるという人がいれば、簡単に説明してもらおうのもよいでしょう。

・補足説明

ここに挙げたツールの他にもいろいろありますが、著名なものをスライドでは挙げています。

Vagrantをこの先少し紹介していますが、仮想環境を手早く構築できて大変便利なツールです。

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
```

・補足説明

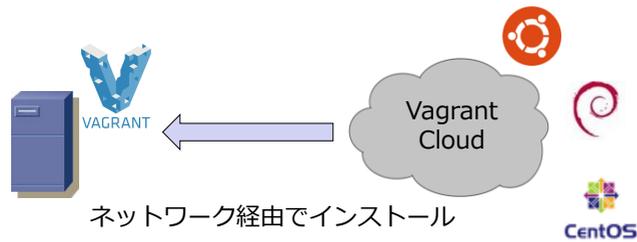
Vagrantは主要なハイパーバイザに対応しており、設定ファイルでどのハイパーバイザを使用するか指定します。

使用するハイパーバイザのことを、Vagrantではプロバイダと呼んでいて、この例ではコメントにもあるようにHyper-Vを指定しています。

Dockerはこの後6章で説明するので、コンテナとは？という疑問は一度置いておいてください。

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



・補足説明

Vagrantでは導入できる仮想イメージのことをBoxと呼び、また動作対象になるハイパーバイザのことをプロバイダと呼びます。

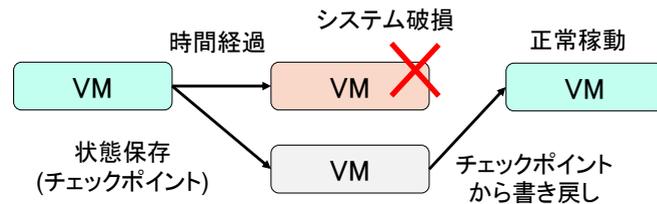
時間に余裕があれば、実際にスライドのURLにアクセスして、Box一覧を見せてください。UbuntuなどのオフィシャルBoxが多数あり、またユーザによって色々と作られたものがアップロードされています。

演習ではプロバイダをHyper-Vに設定してVagrantによる自動構築を行います。

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



・ポイント

VMはファイルベースなので、バックアップを容易にとれる、またハイパーバイザや管理ツールからある時点での状態（チェックポイント、スナップショット）を容易にバックアップ・復元できます。

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は同じではない

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

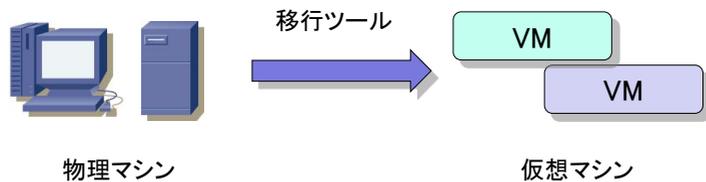
5-6. P2Vマイグレーション

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

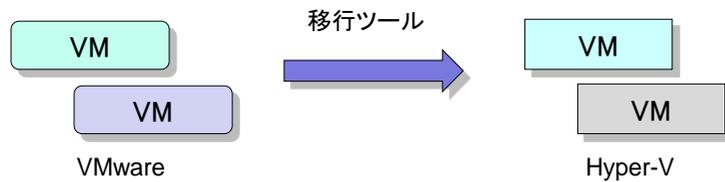


5-7. V2Vマイグレーション

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

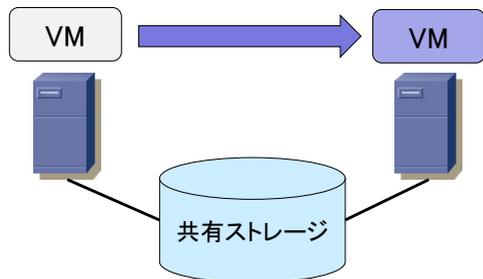
多くの場合互換性が有り、変換ツールも用意されている



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



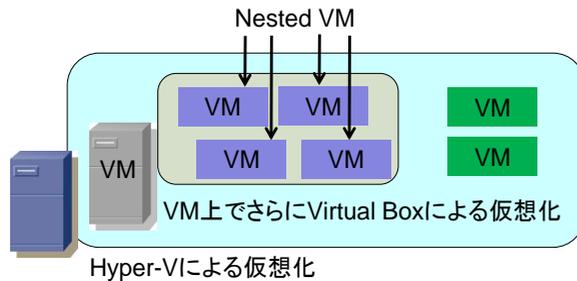
5-9. Nested VM

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働している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



・ポイント

VMの中にさらにVMを立てることが現在では容易にできます。

現在のハイパーバイザではほとんどがデフォルトでNested VMを有効にしていますが、スライドの説明にあるように、Hyper-Vではデフォルトは無効で、設定メニューには項目がないのでコマンドで有効にする必要があります。

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



・説明の流れ

コンテナは元々Linuxのコンテナ機能から来ていますが、なじみのない人にはわかりにくいでしょう。

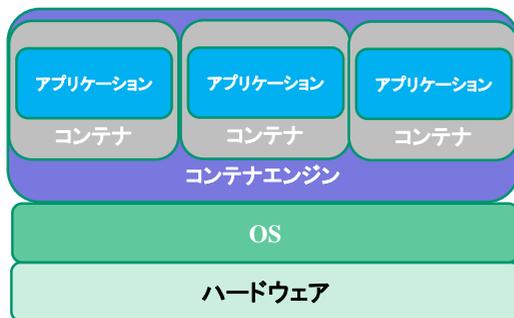
まずコンテナについて知っているかどうか質問し（おそらく大半の人が知らないと思いますが）説明を行います。

また、Dockerという名前について聞いたことがあるかも質問してみましょう。

Dockerは開発環境としてすでに定着しており、名前は聞いたことがある、という人もこれからの説明と演習で概要はつかめるようになるということをわかってもらえるでしょう。

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

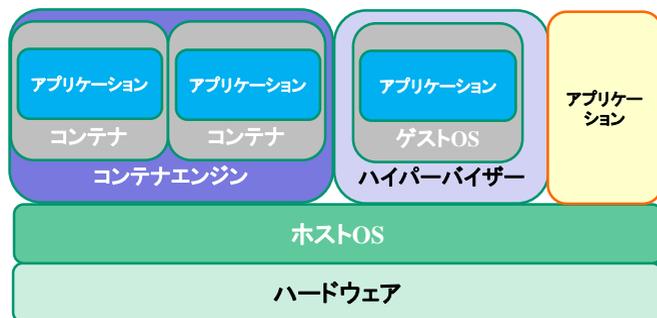


・ポイント

このスライドと次のスライドで、コンテナと仮想化の違いをおさえます。
ハードウェアリソースの消費が非常に少ないという点が重要です。

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



・ポイント

前のスライドと併せて仮想化との違いおさえていきます。

コンテナは隔離・分離されているということ、また仮想化と異なり、OSやハードウェアはホストと共通のものを使用することが重要です。

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



・補足説明

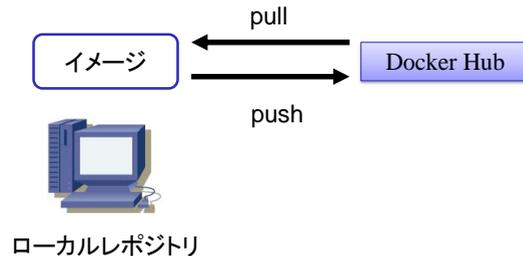
ここからDocker自体の説明に入ります。

Dockerは2017年にスライドにあるようにDocker CE/EEにエディションが分かれました。最近のことなので、少し前のWebサイトや書籍では単にDockerとしか書かれておらず、また詳細も異なります。

もし他に資料を当たる場合、2017年を一つの区切りとして考え、CE/EE以前か以降かを意識してください。

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)



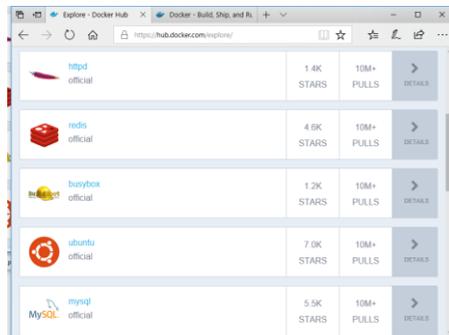
・補足説明

Dockerとコンテナの関係は、VagrantとBoxの関係と似ています。
Dockerはネットワーク経由でDocker Hubに登録されたコンテナをダウンロードして使用します。

もちろん、自分でコンテナを作成することもできますが、最初はず
Docker Hubに登録された様々なサンプルをダウンロードして
Dockerに慣れるところから始めます。

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



・補足説明

Docker Hub上のコンテナは無料で自由に使用できます。

有料でプライベートHubにすると、非公開にできます。自社でのみ使用するコンテナを共有したりするにはDocker EEで非公開Hubを作成します。

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

・補足説明

コンテナは実質Linux専用と考えてもよいでしょう。Win/Macの場合もこのように使用できますが、Windowsの場合は仮想化環境上ですし、MacもLinuxを一度作成して使用します。そのため、ハードウェアリソースの消費が非常に少ない、OSやホストを共有、というコンテナのメリットが活かせません。

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

・ポイント

コンテナは単純に軽いというだけでなく、環境をコンテナ上に隔離できるため、様々なメリットがあります。

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
```

・説明の流れ

ここで、コンテナサイズに注目する。

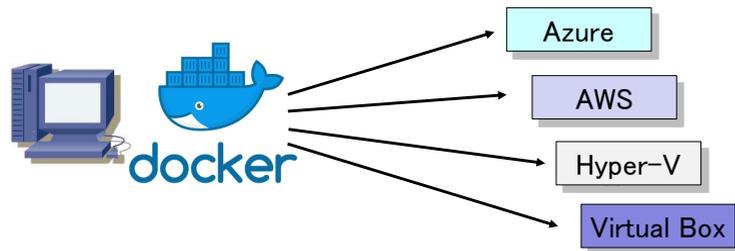
“Hello from Docker!” と表示するだけのコンテナhello-worldは
動作確認に使用されるサンプルコンテナだが、ベースはubuntuで、
ubuntu上に構築すると、ほとんどの部分がベースと共通なため、差
違の表示部分のみがコンテナとして消費する。そのため、驚異的なサ
イズに収まる。

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



・補足説明

プロビジョニングは様々なところで自動構築の用語として使用されているが、dockerでも同様。

クラウド上への展開もサポートされており、AzureやAWSなど主要なクラウドでも独自のコンテナを展開して使用できる。

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