

脱VMware！ Linux KVMで始める 仮想化環境構築超入門

日本仮想化技術株式会社
代表取締役兼CEO

宮原 徹

VirtualTech Japan

講師 自己紹介

宮原 徹

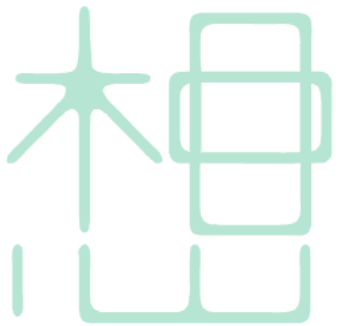
- 1972年1月 神奈川県生まれ
- 1994年3月 中央大学法学部法律学科卒業
- 1994年4月 日本オラクル株式会社入社
 - PCサーバー向けRDBMS製品マーケティングに従事
 - Linux版Oracle8の日本市場向け出荷に貢献
- 2001年1月 株式会社びぎねっと 設立
- 2006年12月 日本仮想化技術株式会社 設立
- 仮想化とクラウド活用に関するコンサルティングを中心に活動



本日のアジェンダ

- 脱VMwareの環境選定
- Proxmox活用のポイント
- Cockpit活用のポイント

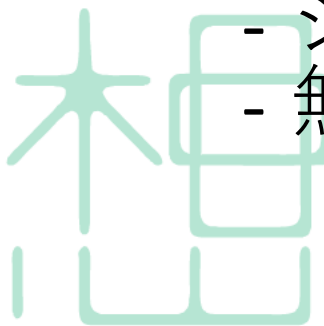
脱VMwareの環境選定



VirtualTech Japan

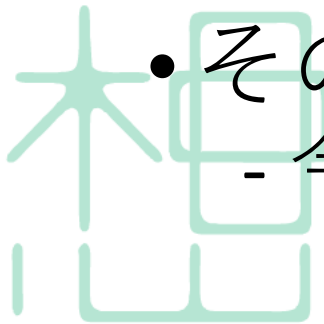
脱VMwareの検討対象

- VMwareの大規模導入のケースは除きます
 - OpenStackやK8sなどを検討？
- VMware Essentials（小規模仮想化基盤）
 - 3ノード + vSphere Server
 - ちょっとしたSANストレージ
 - HA、ライブマイグレーション
- vSphere ESXi無償版（開発用、軽量システム用）
 - シングルノード
 - 無償版提供は一度停止したが現在は再開
 - Broadcom社に対する不信感が盛り上がっていない



脱VMwareの選択肢

- Nutanix
 - 実績あるが意外と高い
- Hyper-V
 - MS製品メインでシステムを構成しているなら
- Proxmox
 - 小規模ならいける
- その他Linux KVMベースソリューション
 - 今後広まりそうな気配

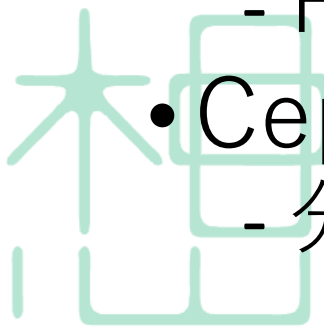


Proxmox活用のポイント



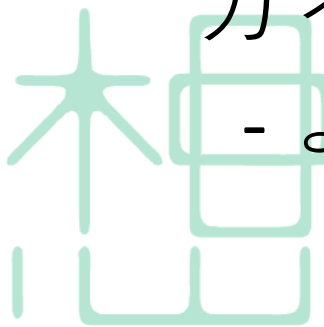
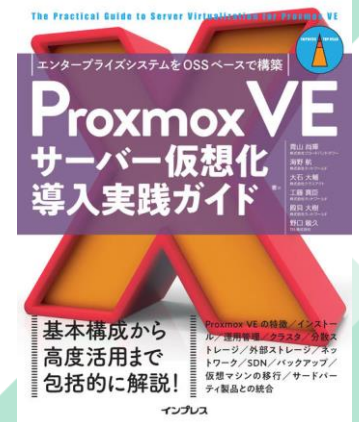
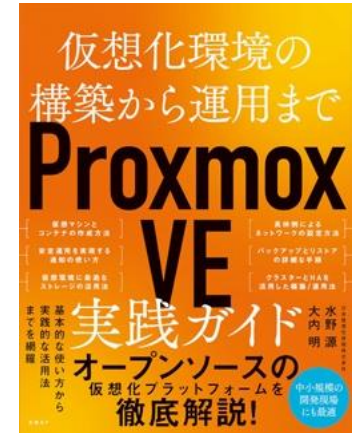
Proxmoxとは

- Debianベース
- **Linux KVM**による仮想マシン
- LXCによるコンテナ
- WebUI
- 複数ノードクラスタ構築可能
 - HA、ライブマイグレーション
- Cephストレージ対応
 - 分散ストレージ



Proxmoxの基本的な解説書

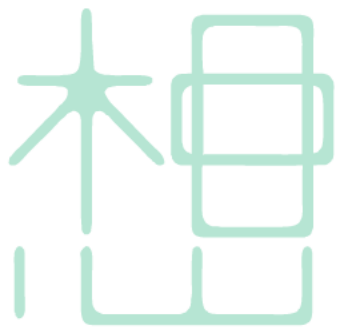
- バージョンが8から9に上がっていますが大体そのまま使えます
- 『Proxmox VE 実践ガイド』
 - 導入から基本的な使い方の解説
- 『Proxmox VEサーバー仮想化 導入実践ガイド』
 - より実践的なことについて知りたい人向け



この後のお話について

- インストールから仮想マシンを動かすまでの流れの引っかけポイントを解説します
- 簡単に環境は構築出来るので実際に触ってみてください
 - Nested VMを使える環境であれば、仮想マシンとしてProxmoxをインストールして、その上で仮想マシンを動かします
- 仮想化そのものに慣れていない場合、VirtualBoxあたりで先に遊んでみることをお勧めします

初期設定



VirtualTech Japan

インストール

- Proxmox VEのISOイメージをダウンロード
- 起動USBメモリを作成
 - ISOイメージをコピーするだけで複数のISOイメージから起動できる「Ventoy」が便利
 - <https://www.ventoy.net/en/index.html>
- インストーラーの起動（大したことはできない）
 - インストール先の選択
 - 管理者パスワードとメールアドレス設定
 - NIC選択とIPアドレス設定

初期起動後にやること

- ネットワーク設定の変更
 - ボンディングでポート冗長化
- サブスクリプションの設定
- 仮想マシン作成の準備
 - ISOイメージのアップロード
 - OVA・QCOW2イメージのアップロード

ネットワーク設定

- インストール時に設定できるのはNIC 1つに対するIPアドレス
 - 正確にはブリッジにIPアドレスを割り当て、1つのNICにブリッジされている
- ネットワーク障害（NIC障害、スイッチ障害）に耐えられるようにNICの冗長化（ボンディング）
 - 理想を言えば、NICをチップ単位で冗長化、スイッチも複数で冗長化
- 各種設定が行えるが、スイッチ側の対応が不要なbalance-tlbに設定しています（個人の見解です）

ネットワーク設定（初期状態）

PROXMOX Virtual Environment 9.1.1

検索

ドキュメント VMを作成 CTを作成 root@pam

再起動 シャットダウン シェル 一括操作 ヘルプ

サーバ表示 ノード 'ms-s1'

データセンター ms-s1

検索

サマリー

注釈

シェル

システム

ネットワーク

証明書

DNS

ホスト

オプション

時刻

システム ログ

アップデート

リポジトリ

ファイアウォール

ディスク

LVM

LVM-Thin

ディレクトリ

ZFS

Ceph

レプリケーション

名前 ↑	代替名	種別	稼働中	自動起動	VLAN a...	ポート/ス...	Bondモード	CIDR	ゲートウェイ	コメント
nic0	enx38052533db43	ネットデバイス	はい	いいえ	いいえ					
nic1	enx38052533db44	ネットデバイス	いいえ	いいえ	いいえ					
nic2	enx40a6b7da5be3	ネットデバイス	いいえ	いいえ	いいえ					
nic3		ネットデバイス	いいえ	いいえ	いいえ					
vmbr0		Linux Bridge	はい	はい	いいえ	nic0		192.168.11.20/24	192.168.11.1	
wlp99s0	wlx0ccdb40953c7	ネットデバイス	いいえ	いいえ	いいえ					

vmbr0のIPアドレスがnic0に紐付いている状態

タスク クラスタログ

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

ボンディング作成時の注意点

最初に設定したNICがボンディング対象の場合

1. vmbr0のブリッジポートを別NICに設定
2. Linux bond (bond0) を作成
3. vmbr0のブリッジポートをbond0に変更
4. 設定を適用

vmbr0のブリッジポートを一時変更

The screenshot shows the Proxmox VE 9.1.1 web interface. In the left sidebar, the 'ネットワーク' (Network) section is expanded. The main panel displays a table of network interfaces. The 'vmbr0' interface is selected, and the 'Edit: Linux Bridge' dialog is open. In this dialog, the 'ブリッジポート' (Bridge port) field is highlighted with a red box and contains the value 'nic2'. A text box overlay with the text 'ブリッジポートを未使用のnic2に一時変更' (Temporarily change bridge port to unused nic2) is positioned over the dialog.

名前	代替名	種別	稼働中	自動起動	VLAN a...	ポート/ス...	Bondモード	CIDR	ゲートウェイ	コメント
nic0	enx38052533db43	ネットデバイス	はい	いいえ	いいえ					
nic1	enx38052533db44	ネットデバイス	いいえ	いいえ	いいえ					
nic2	enx40a6b7da5be3	ネットデバイス	いいえ	いいえ	いいえ					
nic3		ネットデバイス	いいえ	いいえ	いいえ					
vmbr0		Linux Bridge	はい	はい	いいえ	nic0		192.168.11.20/24	192.168.11.1	
wlp99s0	wlx0ccdb40953c7	ネットデバイス	いいえ	いいえ	いいえ					

編集: Linux Bridge

名前: vmbr0 自動起動: ☒

IPv4/CIDR: 192.168.11.20/24 VLAN aware: ☐

ゲートウェイ (IPv4): 192.168.11.1 ブリッジポート: nic2

詳細設定 ☐ OK

ブリッジポートを未使用のnic2に一時変更

タスク クラスタログ

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

Linux Bondを作成

作成ボタンからLinux Bondを選択

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

Linux Bond (bond0) の設定

作成: Linux Bond

名前: bond0

IPv4/CIDR:

ゲートウェイ (IPv4):

スレーブ: nic0 nic1

モード: balance-tlb

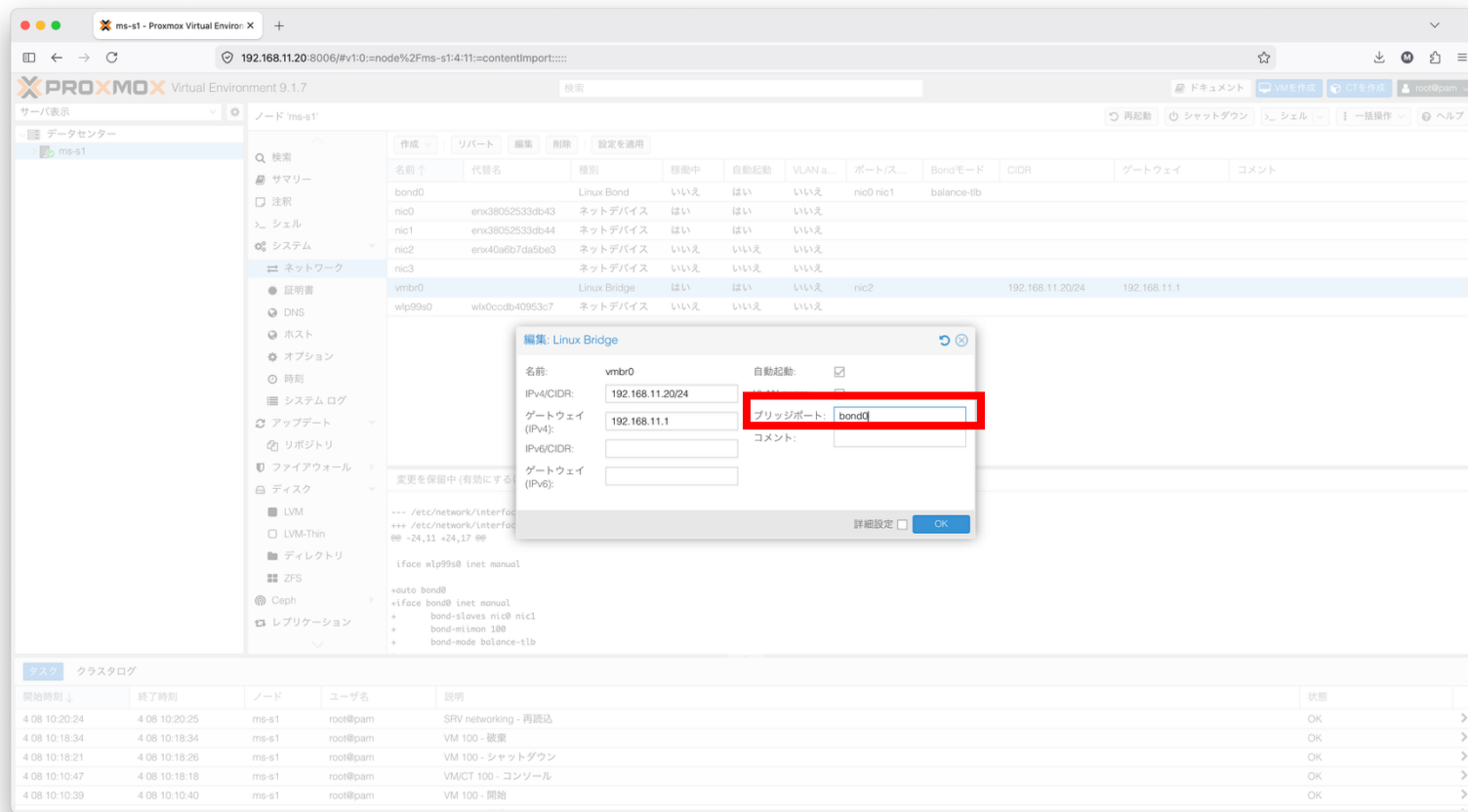
Linux Bond (bond0) のslaveにnic0とnic1を設定

モードはbalance-tlbを選択

名前↑	代替名	種別	稼働中	自動起動	VLAN a...	ポート/ス...	Bondモード	CIDR	ゲートウェイ	コメント
nic0	enx38052533db43	ネットデバイス	はい	いいえ	いいえ					
nic1	enx38052533db44	ネットデバイス	はい	いいえ	いいえ					
nic2	enx40a6b7da5be3	ネットデバイス	いいえ	いいえ	いいえ					
nic3		ネットデバイス	いいえ	いいえ	いいえ					
vmbro		Linux Bridge	はい	はい	いいえ	nic2		192.168.11.20/24	192.168.11.1	
wlp99s0	wlx0ccdb40953c7	ネットデバイス	いいえ	いいえ	いいえ					

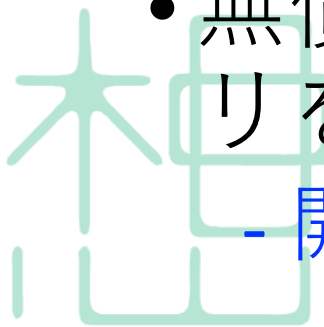
開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

vmbr0のブリッジポートをbond0に設定



サブスクリプションの設定

- アップデート時に参照するパッケージリポジトリの設定
- デフォルトでは有償サブスクリプション用リポジトリを参照
 - 安定版パッケージが提供される
- 無償で使うにはサブスクリプション無しリポジトリを有効化
 - 開発版パッケージが提供される



No-Subscriptionを追加

The screenshot shows the Proxmox VE web interface. In the left sidebar, the 'リポジトリ' (Repositories) option is selected. The main content area displays a table of repositories. A modal dialog titled '追加: リポジトリ' (Add: Repository) is open, showing a text input field with 'No-Subscription' entered. The dialog also includes a 'ヘルプ' (Help) button and an '追加' (Add) button.

Annotations in Japanese:

- ←追加 (Add)
- ←No-Subscriptionを選択 (Select No-Subscription)

The table of repositories shows the following data:

有効	Type	Suites	コンポーネント	オプション	オリジン	コメント
File: /etc/apt/sources.list	deb					
File: /etc/apt/sources.list	deb					
File: /etc/apt/sources.list	deb					

The bottom of the interface shows a 'タスク' (Tasks) section with a table of tasks:

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 10:03:38	4 08 10:03:39	ms-s1	root@pam	SRV networking - 再読み込み	OK
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

サブスクリプション設定

警告

Proxmox VE用の更新を取得します
サブスクリプションなしリポジトリを本番環境で使用することはお勧めできません!

APT リポジトリ

有効	Types	URLs	Suites	コンポーネント	オプション	オリジン	コメント
✓	deb	http://deb.debian.org/debian/	trixie	trixie-updates	main contrib non-free-fi...	Signed-By: /usr/shar... Enabled: false	Debian
✓	deb	http://security.debian.org/debian-security/	trixie-security	main contrib non-free-fi...	Signed-By: /usr/shar... Enabled: false	Debian	
✓	deb	http://download.proxmox.com/debian/pve	trixie	pve-no-subscription		Signed-By: /usr/shar... Enabled: false	Proxmox
✓	deb	http://download.proxmox.com/debian/pve	trixie	pve-enterprise		Signed-By: /usr/shar... Enabled: false	Proxmox

タスク クラスタログ

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 10:03:38	4 08 10:03:39	ms-s1	root@pam	SRV networking - 再読み込み	OK
4 08 09:56:34	4 08 09:56:34	ms-s1	root@pam	VMとコンテナの一括起動	OK
4 08 09:56:32	4 08 09:56:34	ms-s1	root@pam	アップデートパッケージデータベース	エラー: command 'apt-get u...

仮想マシンの作成



仮想マシンの作成方法

A) ISOイメージからインストール

- 好きな設定でインストールできる
- 少し手間がかかる

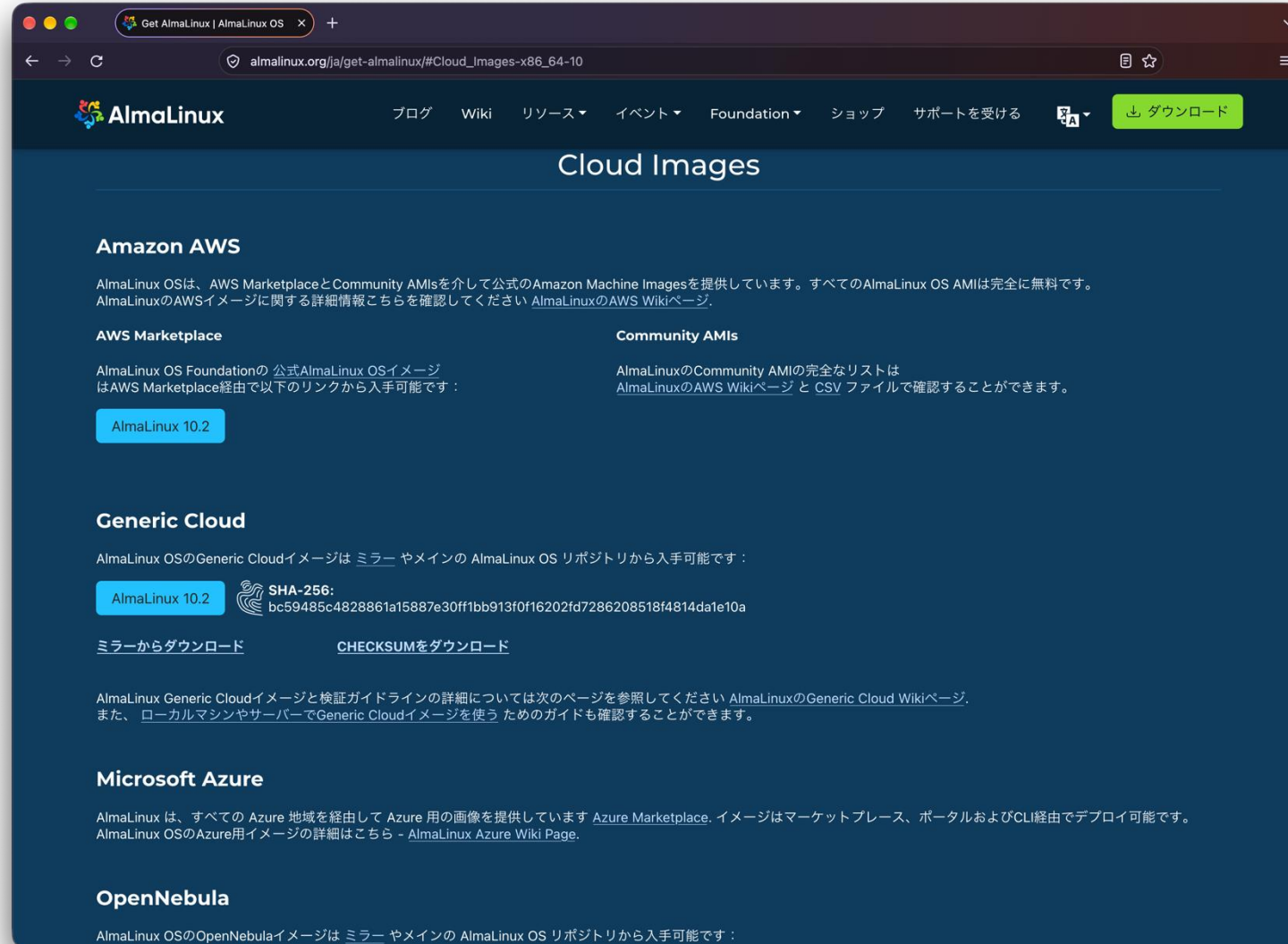
B) クラウドイメージからのインポート

- 手間がかからない
- どのような設定でイメージが作られたのか把握する必要がある
- グローバルなイメージは言語設定が英語なので注意
 - キー配列が違ってる罫

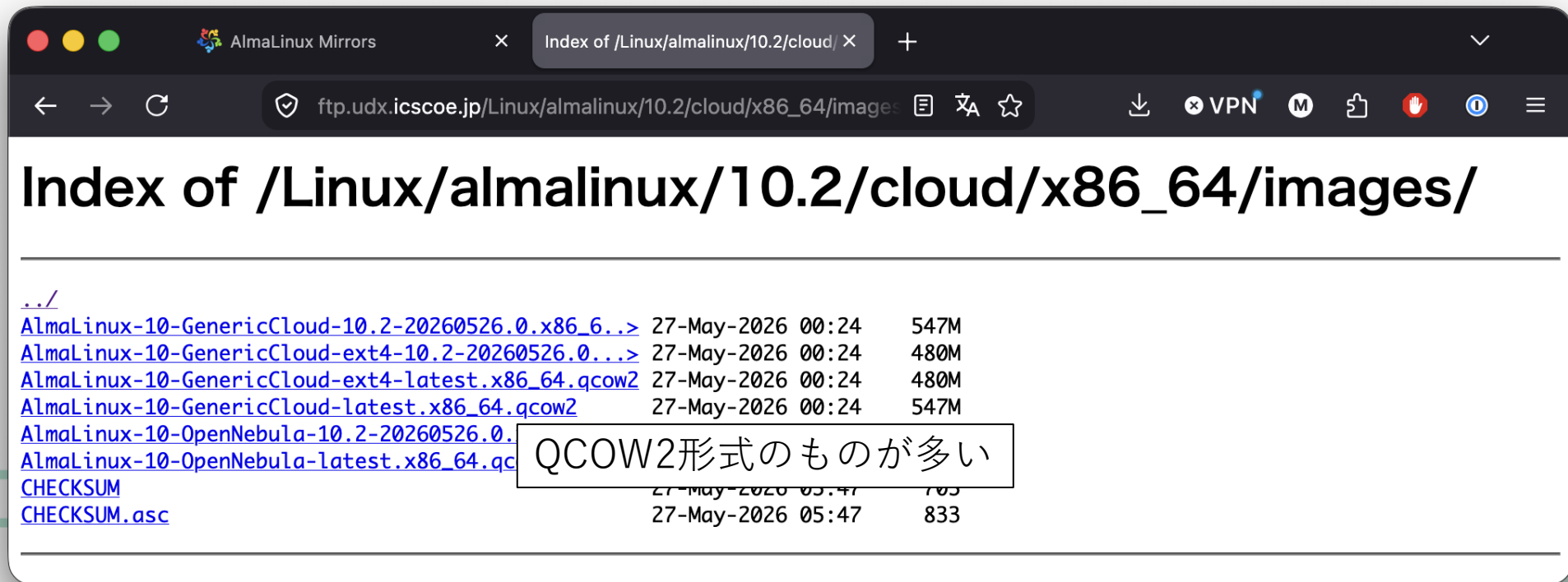
ISOイメージのダウンロード例



クラウドイメージのダウンロード例

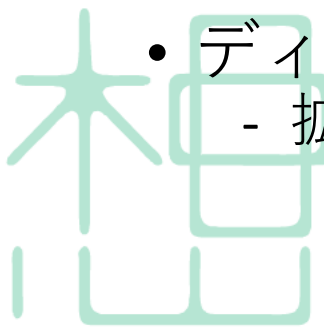


クラウドイメージのダウンロード例



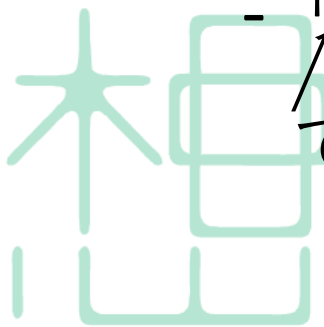
仮想マシン作成のポイント

- IDは変更できないので計画的に
 - クローンを駆使すれば変更はできるが手間
- BIOSは必要に応じてUEFIに
 - UEFI領域が要るので無理にUEFIにしなくてもよい
- Windowsはインストール時にTPMが必要
 - インストール後は外せる
 - 大きめのアップデートの時にもTPMが必要
- QEMUエージェントはチェック（後でも可）
 - ゲストOSにqemu-guest-agentのインストールが必要
- ディスクサイズは余裕をもって
 - 拡張は比較的簡単だが、OS内での拡張がLVMだと手間になる
 - あえてLVMにしないでゲストOSをインストールするのも手か



仮想マシンCPU設定のポイント

- CPUソケットは無理に増やさなくてよい
- CPUコア数は余裕をもって
 - 普段は2か4ぐらいにすることが多い
 - CPUオーバーコミットメントも可能
- CPU種別は分からなければhostに（一番下）
 - 異種CPU混在の場合、ライブマイグレーションできなくなるのできちんと調べて一番下のレベルに合わせる
 - サポートされるレベルは
`/lib64/ld-linux-x86-64.so.2 -help`
で確認できる



CPUの種別はhostに

作成: 仮想マシン

ソケット: 1 種類: host

コア: 4 合計コア数: 4

モデル	ベンダ
Westmere-IBRS	Intel
kvm32	QEMU
kvm64	QEMU
max	QEMU
qemu32	QEMU
qemu64	QEMU
x86-64-v2	QEMU
x86-64-v2-AES	QEMU
x86-64-v3	QEMU
x86-64-v4	QEMU
host	Host

リストの一番下に

タスク クラスタログ

開始時刻 ↓	終了時刻	ノード	ユーザ名	説明	状態
4 08 10:08:27	4 08 10:08:27	ms-s1	root@pam	データのコピー	OK
4 08 10:07:57	4 08 10:07:59	ms-s1	root@pam	データのコピー	OK
4 08 10:04:46	4 08 10:06:50	ms-s1	root@pam	シェル	OK
4 08 10:04:36	4 08 10:04:39	ms-s1	root@pam	アップデートパッケージデータベース	OK
4 08 10:03:38	4 08 10:03:39	ms-s1	root@pam	SRV networking - 再読み込み	OK

サポートアーキテクチャレベルの確認

```
$ /lib64/ld-linux-x86-64.so.2 --help
```

```
Usage: /lib64/ld-linux-x86-64.so.2 [OPTION]... EXECUTABLE-  
FILE [ARGS-FOR-PROGRAM...]
```

(中略)

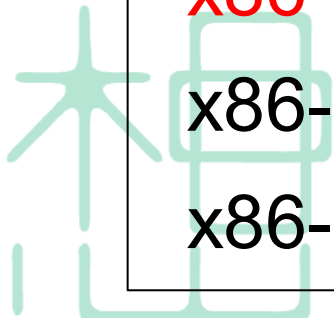
Subdirectories of glibc-hwcap directories, in priority order:

x86-64-v4 (supported, searched)

Supportedになっている最上位レベルを見る

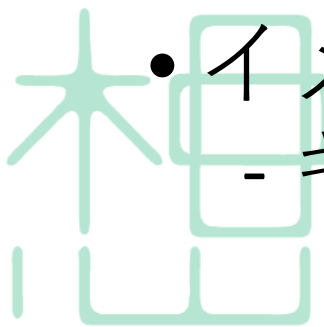
x86-64-v3 (supported, searched)

x86-64-v2 (supported, searched)



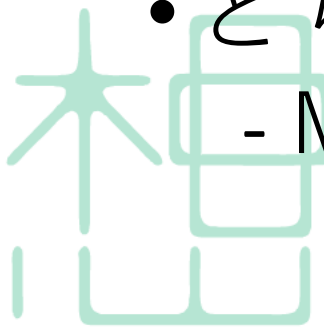
イメージからのインポートの注意点

- CPUの種類は分かなければhostに（一番下）
 - 異なるアーキテクチャ混在のクラスタでは注意
- SCSIコントローラをVirtIO SCSI singleに
 - 並列負荷が高い場合はVirtIO SCSI
- ネットワークインターフェースのモデルをVirtIO（準仮想化）に
- イメージの言語設定に注意
 - キーボードが英語配列だったり



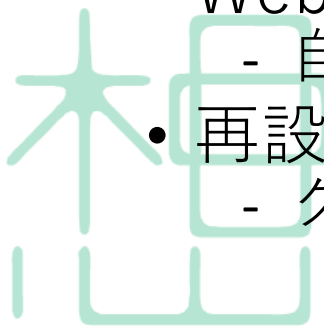
テンプレートとクローン

- テンプレート化すると起動できなくなる
 - イメージ内をいじれない
 - 修正はデプロイしたVMを再度テンプレート化
 - ハードウェア構成等は変更可能
- クローンは稼働可能なVMをそのままクローン
- どちらも基本的な設定項目はそのまま引き継ぐ
 - MACアドレスは再生成



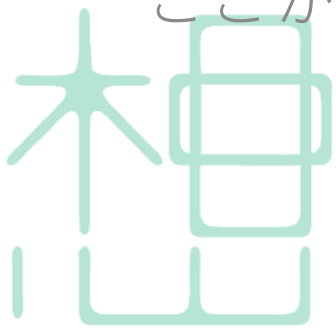
Cloud-initの利用

- ゲストOSにcloud-initパッケージのインストールが必要
- 仮想マシンにCloud-initデバイスの追加
- Cloud-initの設定項目
 - ユーザー名
 - SSH公開鍵（パスワード使いたい場合はパスワード）
 - IPアドレス（DHCP or 固定IPアドレス）
 - パッケージをアップグレードは必要に応じて
- 設定後にイメージ再作成を忘れない
- Webで設定できるパラメーター以外はいれられないのが難点
 - 自分で直接いじればいいけど、Web設定でクリアされる
- 再設定するとSSHホストキーが再作成されるので注意
 - クライアントのknown_hostsからホストキーを削除



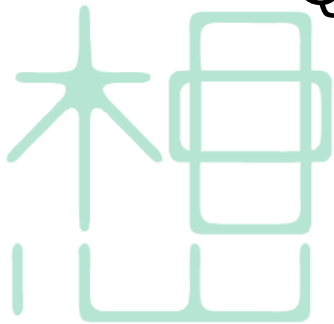
複数台によるProxmox環境

ここからはおまけ



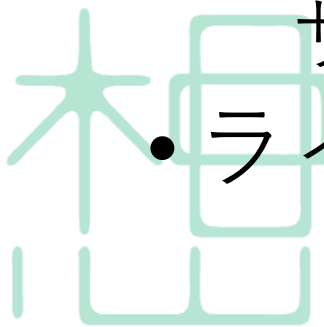
HAクラスタの構成

- 基本は3台構成
- 2台構成の場合、クォーラムの投票のみを行う
QDeviceとQNetdを動かす
 - https://pve.proxmox.com/wiki/Cluster_Manager#corosync_external_vote_support
 - QNetdデバイスにSSHでroot接続する設定が必要
 - sshdのPermitRootLoginをyesに設定しsshd再起動
 - rootユーザーのパスワードを設定



共有ストレージとの接続

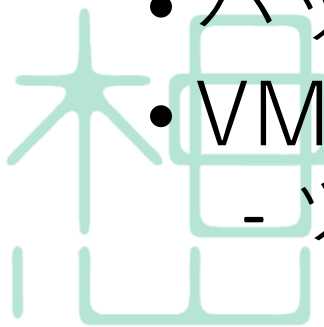
- HAクラスタを構築しても、仮想ディスクイメージが共有されていないと別ノードで起動できない
- いろいろなプロトコルが使えるがSMBが手軽
 - 普通のNASでOK
- ストレージレプリケーションも使える
 - 同期間隔が長いので、データ保存が重要じゃないサービス向け
- ライブマイグレーションも可能



その他検討事項

- Cephストレージ
 - 分散ストレージとして構築
 - ノード間接続に速度が必要
- PCIパススルーによるGPU等の利用
 - バージョン9からかなり簡単に
 - AMD GPUの場合には二手間ほど必要
- バックアップ
- VMwareからのVM移行
 - ツールの他、OVAエクスポートからのインポートなど

このあたりになると
完全にお仕事なので
詳細はお問い合わせ
ください



Cockpit活用のポイント

ごめんなさい、こっちは薄めです



Linux KVM+CockpitによるVM管理

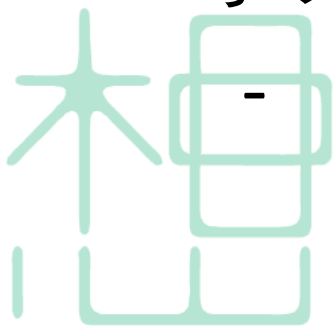
- AlmaLinuxユーザー会勉強会の渡邊さんの資料が参考になります
 - <https://speakerdeck.com/koedoyoshida/almalinux-plus-kvm-plus-cockpit-deshi-meruoshou-qing-jia-xiang-hua-ji-pan-kai-fa-huan-jing-nadodenoli-yong-woxiang-ding-site>

Cockpitで仮想マシンを管理するには

- 「アプリケーション」から「マシン」をインストール
 - 「アプリケーション」はプラグインのようなもの
 - 「マシン」は仮想マシン管理用
- ネットワークの設定
 - ブリッジを作成して仮想マシンと物理NICを接続

Cockpitのネットワーク

- ブリッジを作成して仮想マシンから接続
 - ブリッジを作成するとホストOSのデバイスは見えなくなる
 - ブリッジデバイスにIPアドレスが割り当てられる
 - Proxmoxのvbr0と同じ
- ホストOSのデバイス（物理NIC）にブリッジ接続
 - 「Direct attachment」を選択



その他検討事項

- 複数ノード構成
 - ライブマイグレーションは可能
 - Proxmoxのようなクラスタ設定はないので手設定
- バックアップ
 - 特別なバックアップ手法は無し
- PCIパススルーによるGPU等の利用は大変そう
 - VMの設定ファイルに手で設定が必要

まとめ

- 数台のサーバーでVMwareライクな環境を作りたいならProxmox
 - WebUIに癖があるので慣れが必要
- 素のLinuxで軽くVMを動かしたいならCockpit
 - VMwareよりLinuxに詳しい人向け
 - CockpitはあくまでWeb管理ツール
 - VirtualBoxと競合

あなたの企業に、最適な未来を。

