

Linux Hyper-V で作るサーバ環境

目次

はじめに	1
1. ホスト OS とハード	1
2. Hyper-V マネージャー	1
3. Linux インストーラー	2
4. 仮想マシンの作成	2
5. Ubuntu Server のインストール	4
6. ssh 接続	9
7. サーバのその他の設定	10
7.1. 管理者権限の使用	10
7.2. 時刻 (タイムゾーン) の設定	10
7.3. ロケールの設定	10
7.4. ネットワーク / 名前解決 (DNS) の設定	12

Linux Hyper-V で作るサーバ環境

はじめに

Docker を使えば特定のサービス (DBMS、Web、Git や Jenkins 等の DevOps 他) を構成済の環境が簡単に手に入りますが、自分の好みのパッケージを組み合わせる環境を作りたい場合は Docker だとかえって手間が掛かります。また、ミドルウェアの導入試験や開発の方式確認を行うときなどは、ターミナルから自由に操作できるサーバ環境が必要になります。

昨今は Windows に仮想マシンを作るための機能が標準で用意されている¹ (Hyper-V) ので、これを使って Windows 上に Linux サーバ環境を作る手順を紹介します。

1. ホスト OS とハード

ここでは以下の仕様の PC を想定しています。

OS : Windows 10 Pro エディション

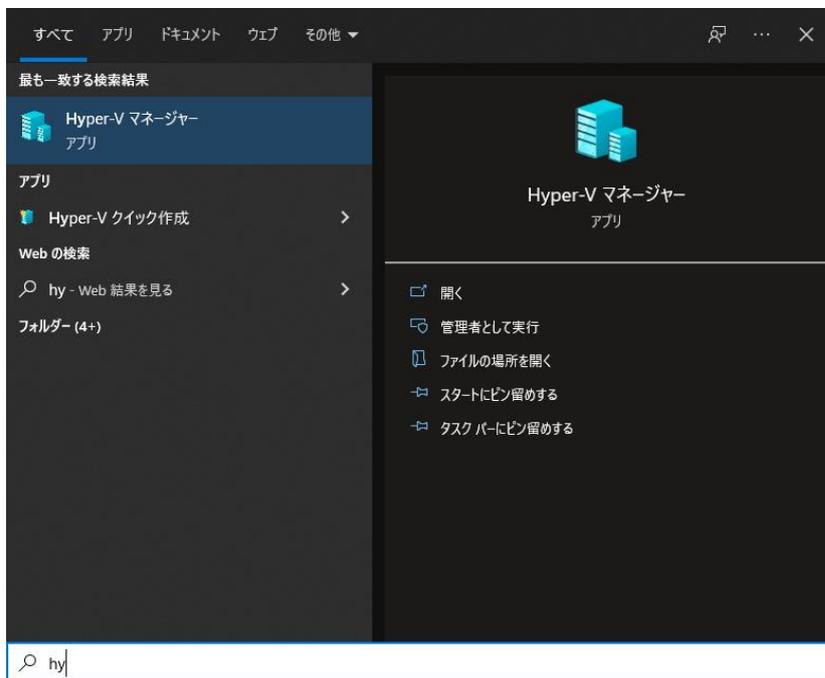
※Home エディションの Hyper-V 有効化方法も各所に公開されていますが、ここでは取り上げません。以降の説明は Hyper-V マネージャーが有効になっている前提です

メモリ : 8 Gbyte (以上)

ディスク空き : 127Gbyte (VM のディスク容量上限設定。初期占有量は 10GByte 程度で自動拡張)

2. Hyper-V マネージャー

Hyper-V が使える状態であれば、Windows のスタートボタンを押下して hyper と打鍵していくと Hyper-V マネージャーが検索結果にでてくるので、これを起動します。



¹ Linux にも KVM(Kernel-based Virtual Machine)というモジュールがあります。オラクルがオープンソース(GPLv2)化した VirtualBox というソフトウェアはマルチプラットフォームです

Linux Hyper-V で作るサーバ環境

3. Linux インストーラー

Hyper-V クイック作成も表示されますが、こちらで選択できる構成はデスクトップ (GUI) を使う構成になっています。サーバとして構成する場合は、こちらは使いません。

以下のサイトから Ubuntu²の LTF (長期サポート版) 最新版がダウンロードできます。

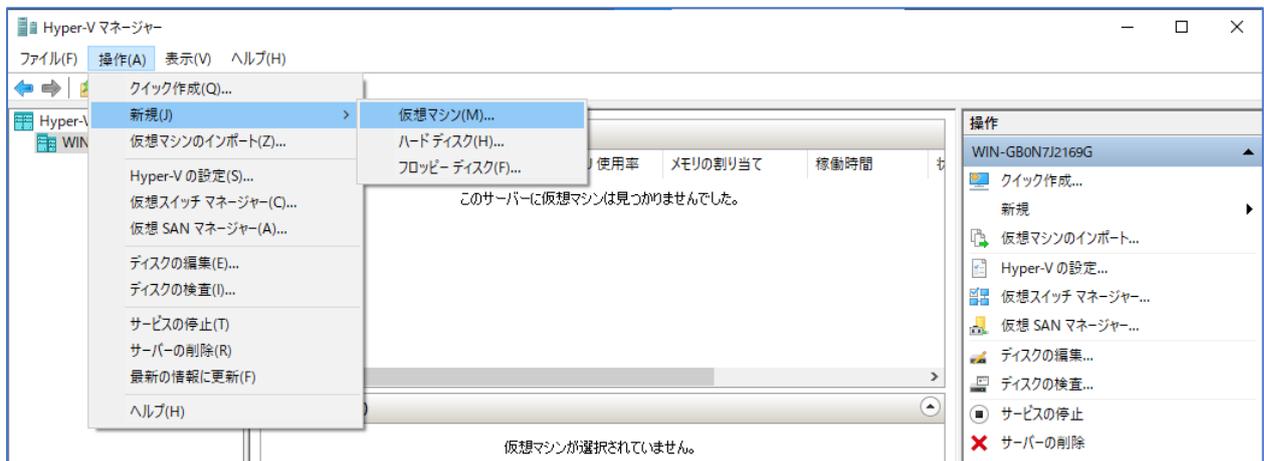
<http://jp.releases.ubuntu.com/focal/>

Desktop image と Server install image があるので「Server install image」をダウンロードします。資料作成時の対象 (最新) 資材は、ubuntu-20.04.4-live-server-amd64.iso です。

4. 仮想マシンの作成

Hyper-V マネージャーのメニュー >操作 >新規 >仮想マシン を選択します。

※クイック作成...から選択できる構成はデスクトップ (GUI) を使う構成になっています。サーバとして構成したいので、こちらは使いません。



仮想マシンの以下の項目は重要なので確認してください。他はデフォルト値でも問題ありません。

●世代の指定 : 「第2世代」を選択 (セキュアブートの設定を行うためにこちらを選びます)

●ネットワークの構成 : Default Switch

●仮想ハードディスクの接続 : 「仮想ハードディスクを作成する」を選択

●インストールオプション :

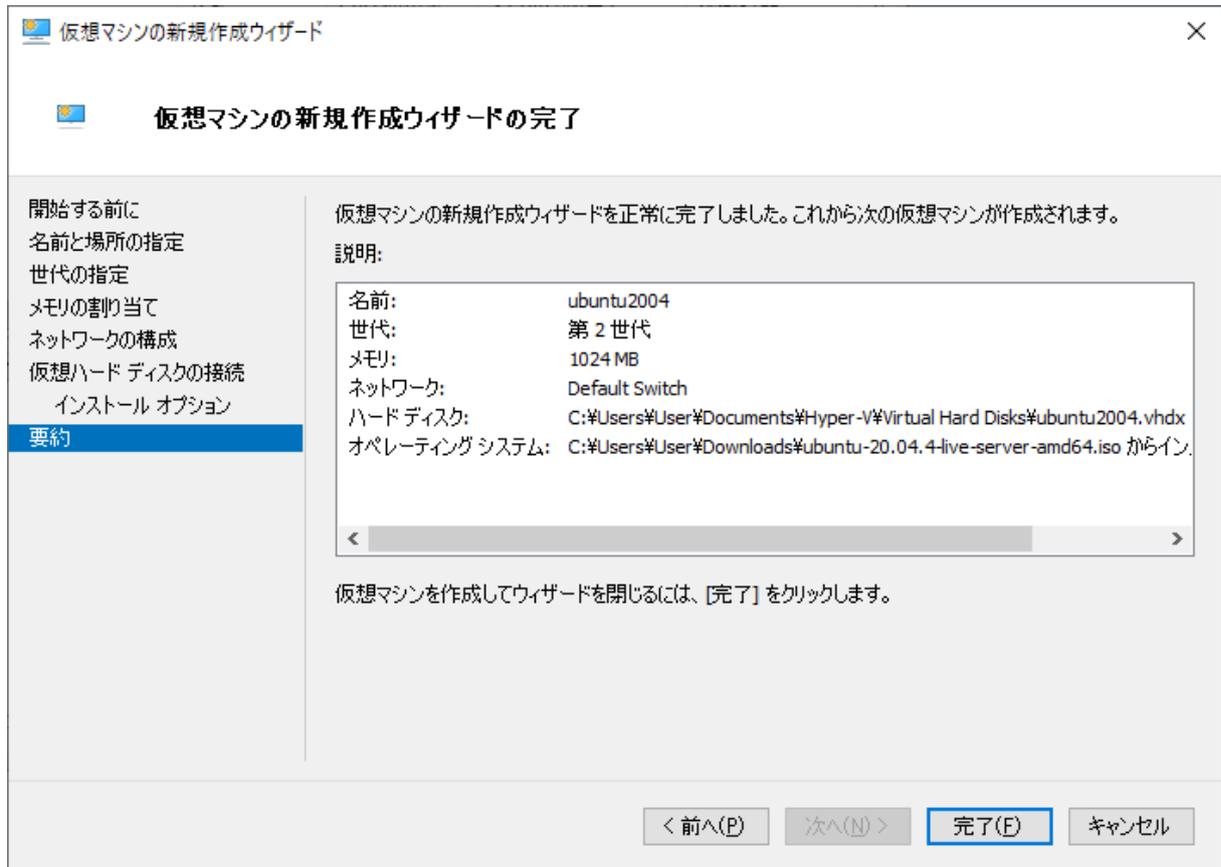
「ブートイメージファイルからオペレーティングシステムをインストールする」を選択
メディアに、項番 3. でダウンロードした iso ファイル

² Ubuntu はオープンソース、無料で使用できると明示されていますが、知的財産権等についてはディストリビュータが独自に設定しているので、以下を一読してください。

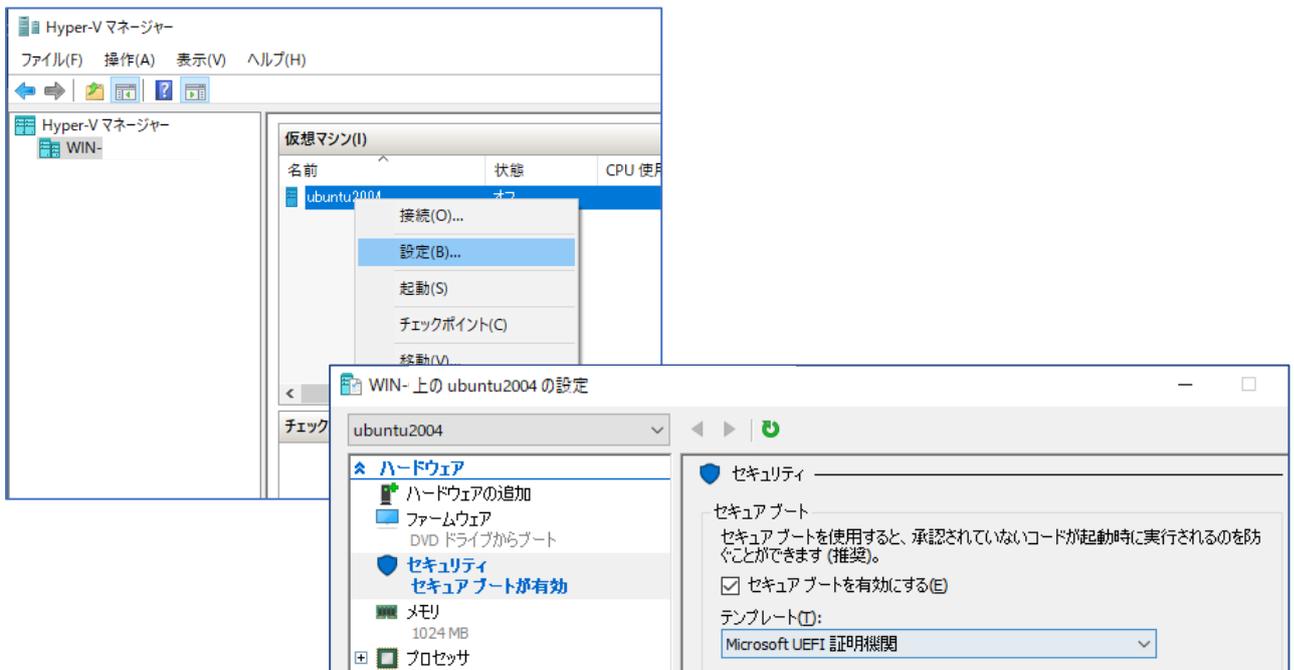
<https://ubuntu.com/legal/intellectual-property-policy>

Linux Hyper-V で作るサーバ環境

要約は以下のようになります。ハードディスクとオペレーティングシステムのパスは環境により異なります。[完了]ボタン押下で仮想



できた仮想マシンを右クリック >設定 でセキュリティ> セキュアブートの設定を行います。



セキュリティからセキュアブートを有効、テンプレートに
Microsoft UEFI 証明機関を選択します

Linux Hyper-V で作るサーバ環境

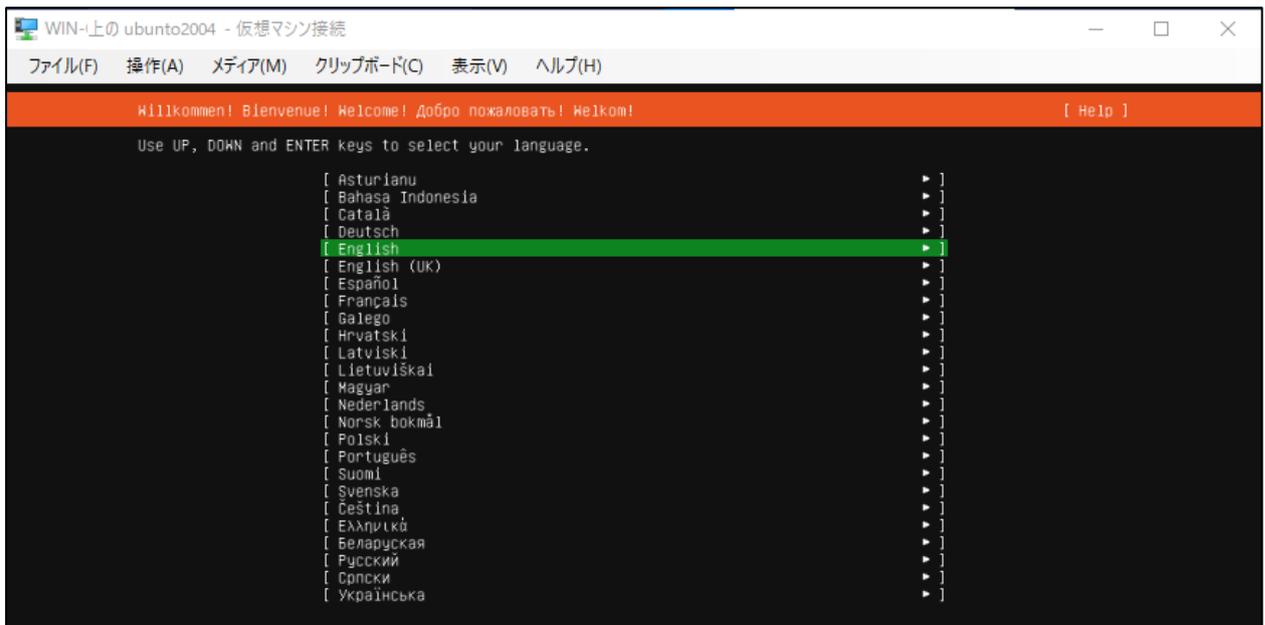
5. Ubuntu Server のインストール

以下の手順でインストールを進めます。

① Hyper-V マネージャーから項番 4. で作成した仮想マシンを右クリック >起動 を選択します。

② 仮想マシンが実行中になったら、仮想マシンを右クリック >接続 を選択します。

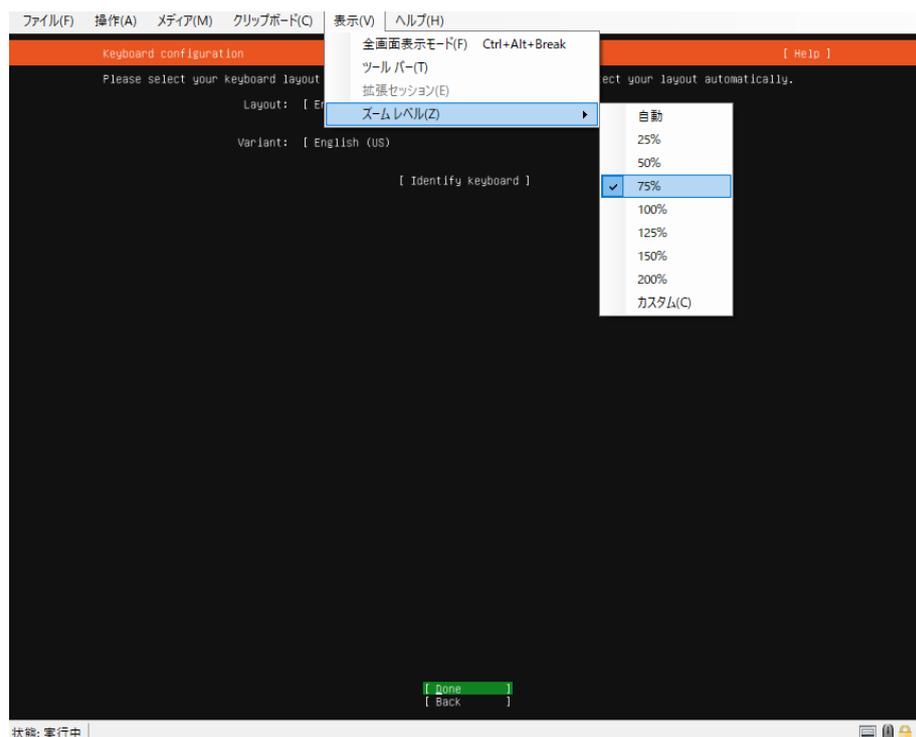
...しばらく画面上に起動ログが表示された後 (/cdrom unmount failed 等は無視してよい)、
言語選択のスクリーンがでます



・言語選択は日本語がないので、English を選んだ ([Enter]押下) ものとして以降進めます

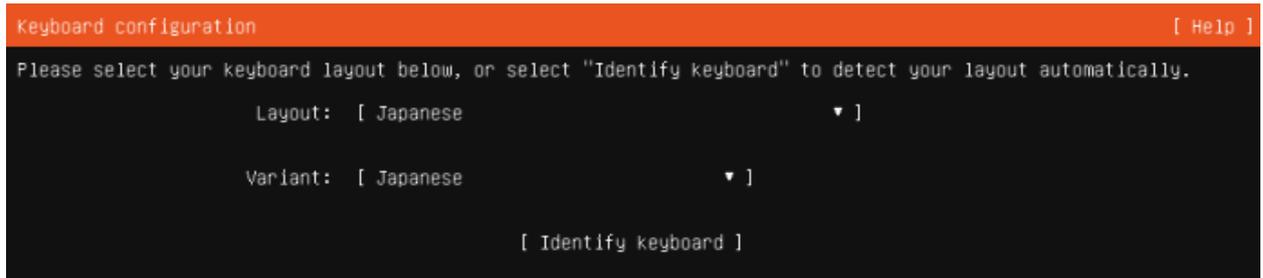
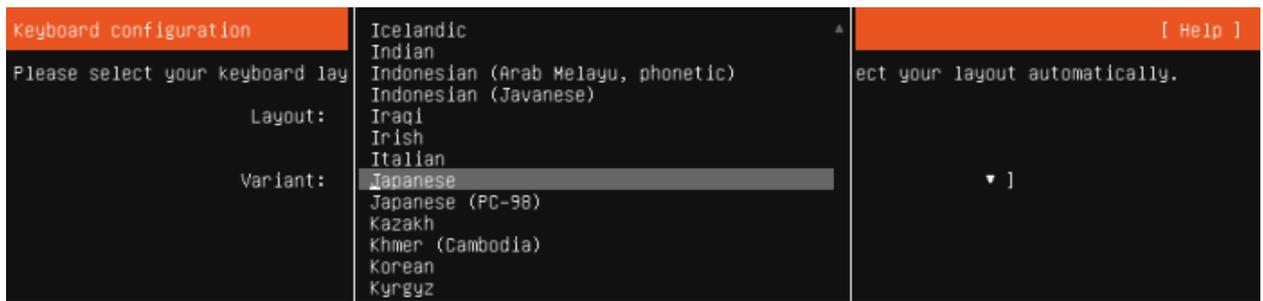
※次の画面で、解像度によっては下部に [Done][Back] が表示されない場合があります。

メニュー >表示 >ズームレベルを調節してください。



Linux Hyper-V で作るサーバ環境

- ・キーボードの選択画面がでるので、↑↓キーでLayoutを選択し、[Enter]を押下します

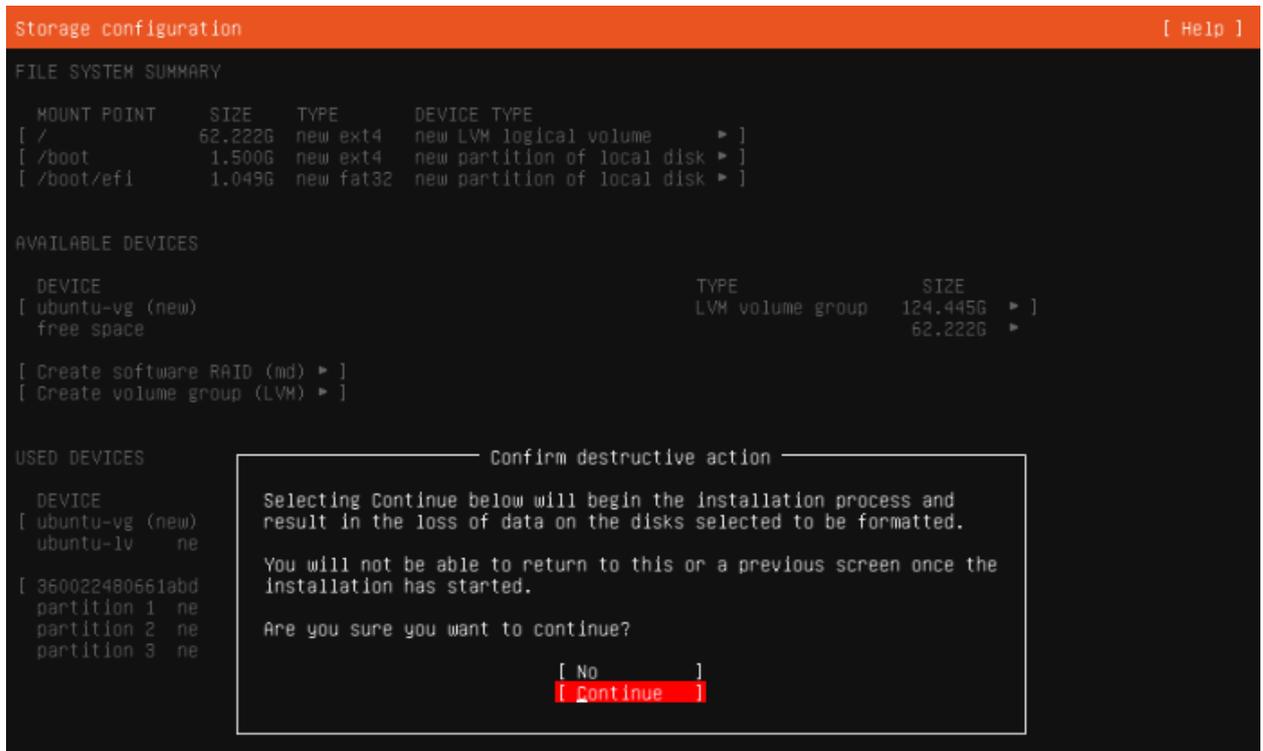


- ・Japanese の選択肢があるので、↑↓でこれを選択して[Enter]を押下 > 画面下の Done を選択して[Enter]を押下

【中略】 Network connections*1/ Configure proxy / Guided storage configuration 全てデフォルトで OK

*1:インストール中にダウンロードを行うので、ネットワーク設定は DHCP のまま にしておきます

Storage Configuration の画面がでるまで全て、画面下の Done を選択&[Enter]で進めます



- ・インストールを進めてよいか確認がでるので、Continue を選択し[Enter]

Linux Hyper-V で作るサーバ環境

Profile setup … 管理権限者の登録 (root ではない)

```
Profile setup [ Help ]
Enter the username and password you will use to log in to the system. You can configure SSH access on the next
screen but a password is still needed for sudo.

Your name: user full name
Your server's name: ubuntu2004
The name it uses when it talks to other computers.
Pick a username: adminuser
Choose a password: *****
Confirm your password: *****
```

- Your name :/etc/passwd のコメント欄に入ります (利用者氏名等、ログイン名ではない)
- Your server's name : /etc/hostname にセットされ、hostname コマンドで表示される名前です
- Pic a username :ログインで使うユーザ名です
- Choose a password :ハッシュ化して/etc/shadow に格納されます
- 以上を設定し、画面下の Done を選択して[Enter]

次に表示される Enable Ubuntu Advantage 画面は空白のまま 先に進め (Done) てください

SSH setup … ネットワーク経由でログインできるように sshd をインストール

```
SSH Setup [ Help ]
You can choose to install the OpenSSH server package to enable secure remote access to your server.

[ ] Install OpenSSH server

Import SSH identity: [ No ]
You can import your SSH keys from GitHub or Launchpad.

Import Username:

[ ] Allow password authentication over SSH
```

Featured Server Snaps … 追加のサーバーAP を選択 (なにか必要なら...)

```
Featured Server Snaps [ Help ]
These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see more details of
the package, publisher and versions available.

[ ] microk8s          canonical✓      Kubernetes for workstations and appliances
[ ] nextcloud         nextcloud✓     Nextcloud Server - A safe home for all your data
[ ] wekan             xet7          The open-source kanban
[ ] kata-containers  katacontainers✓ Build lightweight VMs that seamlessly plug into the containers eco
[ ] docker           canonical✓     Docker container runtime
[ ] canonical-livepatch canonical✓     Canonical Livepatch Client
[ ] rocketchat-server rocketchat✓    Rocket.Chat server
[ ] mosquito         mosquito✓     Eclipse Mosquitto MQTT broker
[ ] etcd             canonical✓     Resilient Key-value store by CoreOS
[ ] powershell      microsoft-powershell PowerShell for every system!
[ ] stress-ng        cking-kernel-tools tool to load and stress a computer
[ ] sabnzbd          safihre       SABnzbd
[ ] wormhole         snapcrafters  get things from one computer to another, safely
[ ] aws-cli          aws✓          Universal Command Line Interface for Amazon Web Services
[ ] google-cloud-sdk google-cloud-sdk✓ Google Cloud SDK
[ ] slcli            softlayer     Python based SoftLayer API Tool.
[ ] doctl            digitalocean✓ The official DigitalOcean command line interface
[ ] conjure-up       canonical✓     Package runtime for conjure-up spells
[ ] postgresql10    cmd✓         PostgreSQL is a powerful, open source object-relational database s
[ ] heroku           heroku✓       CLI client for Heroku
[ ] keepalived       keepalived-project✓ High availability VRRP/BFD and load-balancing for Linux
[ ] prometheus       canonical✓     The Prometheus monitoring system and time series database
[ ] juju             canonical✓     Juju - a model-driven operator lifecycle manager for K8s and machi
```

Done でインストールが始まります。

Linux Hyper-V で作るサーバ環境

Install complete! …インストール完了画面

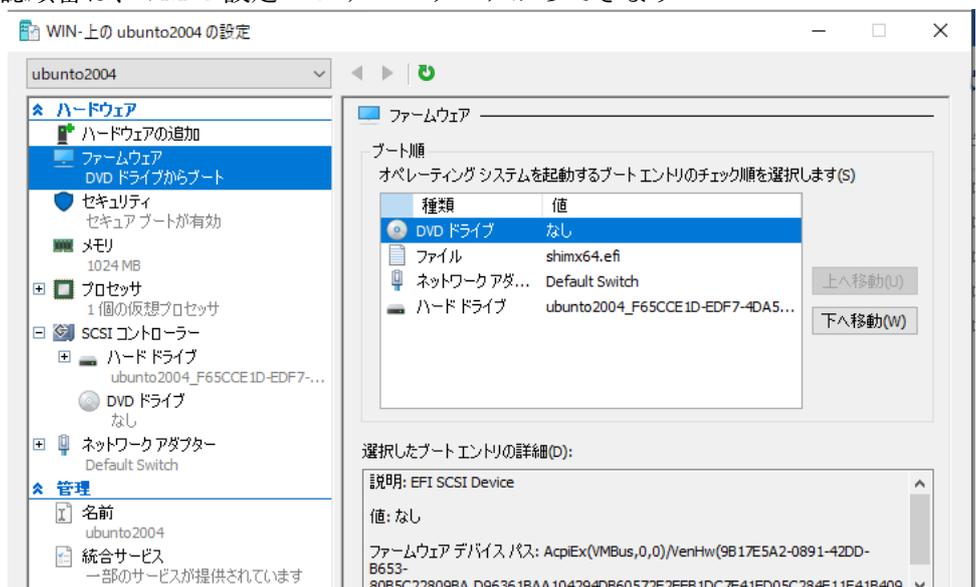
```
Install complete! [ Help ]
configuring lvm_volgroup: lvm_volgroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-2
configuring mount: mount-2
configuring mount: mount-1
configuring mount: mount-0
writing install sources to disk
running 'curtin extract'
curtin command extract
acquiring and extracting image from cp:///tmp/tmpx6Sm_uss/mount
configuring installed system
running 'mount --bind /cdrom /target/cdrom'
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
Installing packages on target system: ['efibootmgr', 'grub-efi-amd64', 'grub-efi-amd64-signed',
'shim-signed']
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
calculating extra packages to install
installing openssh-server
curtin command system-install
downloading and installing security updates
curtin command in-target
restoring apt configuration
curtin command in-target
curtin command in-target
subiquity/Late/run
[ View full log ]
[ Reboot Now ]
```

[Reboot Now]を選択し[Enter]を押下。VM を再起動します。

```
ファイル(F)  操作(A)  メディア(M)  クリップボード(C)  表示(V)  ヘルプ(H)
[FAILED] Failed unmounting /cdrom.
Please remove the installation medium, then press ENTER:
```

インストール媒体を取り外せとメッセージがでたら、そのまま[Enter]押下

<参考> 起動媒体の確認順番は、VM の設定 >ファームウェアからできます



Linux Hyper-V で作るサーバ環境

起動が完了したら、ログイン・プロンプトが表示されます。

```
ファイル(F)  操作(A)  メディア(M)  クリップボード(C)  表示(V)  ヘルプ(H)
Ubuntu 20.04.4 LTS ubuntu2004 tty1
ubuntu2004 login: _
```

インストールの Prifile setup で入力した Pic a username と Choose a password でログインできれば完成です。

```
ファイル(F)  操作(A)  メディア(M)  クリップボード(C)  表示(V)  ヘルプ(H)
Ubuntu 20.04.4 LTS ubuntu2004 tty1
ubuntu2004 login: adminuser
Password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue 19 Apr 2022 02:26:04 AM UTC

System load:  0.1          Processes:    100
Usage of /:   7.0% of 61.00GB  Users logged in:  0
Memory usage: 52%          IPv4 address for eth0: 172.27.29.0
Swap usage:  0%

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

adminuser@ubuntu2004:~$ _
```

③ 最後に ssh 接続のための ip アドレスを ip コマンドで確認します。

```
adminuser@ubuntu2004:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:15:5d:0b:05:07 brd ff:ff:ff:ff:ff:ff
   inet 172.27.29.0/20 brd 172.27.31.255 scope global dynamic eth0
       valid_lft 84316sec preferred_lft 84316sec
   inet6 fe80::215:5dff:fe0b:507/64 scope link
       valid_lft forever preferred_lft forever
adminuser@ubuntu2004:~$ _
```

ログイン後に ip address と入力し、2: eth0 の 3 行目 inet に表示されているのが IPv4 アドレスです (この例の場合、172.27.29.0)

後は、exit コマンドでログアウトし、仮想マシン接続も終了して構いません。

Linux Hyper-V で作るサーバ環境

6. ssh 接続

Hyper-V マネージャーから仮想マシン接続はコピー&ペーストに制約がある等、使い勝手がよくないので、インストール手順の SSH setup で設定しておいた ssh を使って接続します。

Windows のコマンドプロンプトから以下のように ssh コマンドを実行します。

```
ssh ユーザー名@サーバip アドレス
```

ユーザ名：インストールの Prifile setup で入力した Pic a username

サーバ ip アドレス：仮想マシン接続で実行した ip address で確認した ip アドレス

<実行例>

```
ca: コマンドプロンプト - ssh adminuser@172.27.29.0
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:¥Users¥User>ssh adminuser@172.27.29.0
The authenticity of host '172.27.29.0 (172.27.29.0)' can't be established.
ECDSA key fingerprint is SHA256:DpuVFE72f2h3LmWshenpgwn/giDWmRVyVTx4b6/NF+E.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

確認の応答が出るので、yes を入力し、その後のパスワードにはインストールの Prifile setup で設定したパスワードを応答します。

```
ca: adminuser@ubuntu2004: ~
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:¥Users¥User>ssh adminuser@172.27.29.0
The authenticity of host '172.27.29.0 (172.27.29.0)' can't be established.
ECDSA key fingerprint is SHA256:DpuVFE72f2h3LmWshenpgwn/giDWmRVyVTx4b6/NF+E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.27.29.0' (ECDSA) to the list of known hosts.
adminuser@172.27.29.0's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue 19 Apr 2022 04:48:45 AM UTC

System load:  0.02          Processes:    104
Usage of /:   7.0% of 61.00GB Users logged in: 1
Memory usage: 49%          IPv4 address for eth0: 172.27.29.0
Swap usage:  0%

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Tue Apr 19 04:35:18 2022 from 172.27.16.1
adminuser@ubuntu2004:~$ █
```

※コマンドプロンプトを使ってカーソルが消える症状が発生する場合、日本語入力 (IME) の予測入力をオフを試してください

Linux Hyper-V で作るサーバ環境

7. サーバのその他の設定

Linux(Ubuntu20.04)のインストールが終わりましたが、初期状態では時刻 (timezone) が日本の時刻と 9 時間ずれて (UTC になっている) いたり、日本語の表記になっていません。これらを調整するには追加の設定が必要です。

7.1. 管理者権限の使用

サーバの環境変更は殆どが管理者権限を必要とします。インストール時に Prifile setup で登録したユーザは管理者グループに登録されるので管理者権限を持ちますが、管理者権限は sudo コマンドを使い、自分のパスワードを入力した場合だけ効力を持ちます。

7.2. 時刻 (タイムゾーン) の設定

timedatectl コマンドで設定します。

```
adminuser@ubuntu2004:~$ sudo timedatectl set-timezone Asia/Tokyo  
[sudo] password for adminuser:
```

```
adminuser@ubuntu2004:~$ timedatectl    …パラメータ無し timedatectl コマンドで設定値確認  
      Local time: Tue 2022-04-19 14:36:26 JST    ←日本時間  
      Universal time: Tue 2022-04-19 05:36:26 UTC  
      RTC time: Tue 2022-04-19 05:36:26  
      Time zone: Asia/Tokyo (JST, +0900)  
System clock synchronized: yes  
      NTP service: active  
      RTC in local TZ: no
```

7.3. ロケールの設定

(1) インストール済ロケールの確認

日本語の表示にはロケール (言語、国、文字コード) を設定する必要があります。システムに入っているロケールは localectl list-locales コマンドで表示できます。

```
adminuser@ubuntu2004:~$ localectl list-locales  
C.UTF-8  
en_US.UTF-8  
--- 以上、デフォルトの英語、米国、UTF-8 のセットだけが入っている ---
```

(2) 日本語パッケージのインストール

パッケージは、apt コマンドでダウンロードとインストールができます。

また、日本語に必要なロケールを含むパッケージは language-pack-ja です。

```
adminuser@ubuntu2004:~$ sudo apt install language-pack-ja  
[sudo] password for adminuser:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  language-pack-ja-base
```

Linux Hyper-V で作るサーバ環境

```
The following NEW packages will be installed:
 language-pack-ja language-pack-ja-base
0 upgraded, 2 newly installed, 0 to remove and 26 not upgraded.
Need to get 1,669 kB of archives.
After this operation, 7,772 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://jp.archive.ubuntu.com/ubuntu focal-updates/main amd64 language-pack-ja-base all 1:20.04+20220211 [1,667 kB]
Get:2 http://jp.archive.ubuntu.com/ubuntu focal-updates/main amd64 language-pack-ja all 1:20.04+20220211 [1,896 B]
Fetched 1,669 kB in 4s (415 kB/s)
Selecting previously unselected package language-pack-ja-base.
(Reading database ... 71812 files and directories currently installed.)
Preparing to unpack .../language-pack-ja-base_1%3a20.04+20220211_all.deb ...
Unpacking language-pack-ja-base (1:20.04+20220211) ...
Selecting previously unselected package language-pack-ja.
Preparing to unpack .../language-pack-ja_1%3a20.04+20220211_all.deb ...
Unpacking language-pack-ja (1:20.04+20220211) ...
Setting up language-pack-ja (1:20.04+20220211) ...
Setting up language-pack-ja-base (1:20.04+20220211) ...
Generating locales (this might take a while)...
  ja_JP.UTF-8... done
Generation complete.
```

(3) システムロケールの設定

システム全体のロケール設定は、`localectl set-locale` コマンドで行います。

```
adminuser@ubuntu2004:~$ sudo localectl set-locale LANG=ja_JP.UTF-8
```

--- (*'▽') localectl コマンドでシステムロケールが変更されたことを確認できる ---

```
adminuser@ubuntu2004:~$ localectl
```

```
System Locale: LANG=ja_JP.UTF-8
VC Keymap: n/a
X11 Layout: jp
X11 Model: pc105
```

--- (°Д°) locale で環境変数の値を確認すると、変わっていない... ---

```
adminuser@ubuntu2004:~$ locale
```

```
LANG=en_US.UTF-8
LANGUAGE=
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
```

(以下略)

Linux Hyper-V で作るサーバ環境

ロケールは環境変数に反映されないと効力を持たないため、set-locale を行っても次にログインするまでは効果ができません。

一方で、LANG=ja_JP.UTF-8 と打鍵したセッションには即時効果が得られます。同様に、ユーザのホームディレクトリにある bash の初期化ファイル (.bashrc 等) で LANG 環境変数に設定すれば set-locale とは関係なく個人毎にロケールを変えることができます。

--- ログインしなおすと... ---

```
adminuser@ubuntu2004:~$ locale
LANG=ja_JP.UTF-8
LANGUAGE=
LC_CTYPE="ja_JP.UTF-8"
LC_NUMERIC="ja_JP.UTF-8"
LC_TIME="ja_JP.UTF-8"
LC_COLLATE="ja_JP.UTF-8"
LC_MONETARY="ja_JP.UTF-8"
LC_MESSAGES="ja_JP.UTF-8"
LC_PAPER="ja_JP.UTF-8"
LC_NAME="ja_JP.UTF-8"
LC_ADDRESS="ja_JP.UTF-8"
LC_TELEPHONE="ja_JP.UTF-8"
LC_MEASUREMENT="ja_JP.UTF-8"
LC_IDENTIFICATION="ja_JP.UTF-8"
LC_ALL=
```

7.4. ネットワーク／名前解決 (DNS) の設定

ネットワークの構成に Default Switch を指定するだけでホスト OS を経由したネットワーク接続ができる環境が作られます。ここに至るまでのインストールが正常に終わっているのであればネットワークは正常に構成されています。

(1) デフォルトの状態

仮想マシンはインターネット接続共有 (ICS : Internet Connection Sharing) という機能によりホスト OS から IP アドレスと DNS サーバの情報を受け取り (DHCP) mshome.net ドメインに参加します。ホスト OS と仮想マシンはドメイン内で通信ができ、ICS のネットワーク アドレス変換 (NAT) によりインターネットに接続します。

この環境はホストマシン (Windows) に作られる %systemroot%\system32\drivers\etc\hosts.ics に定義されます。

Linux Hyper-V で作るサーバ環境

※ICS の定義は hosts.ics で確認できます。

```
C:¥Users¥User>more %systemroot%¥system32¥drivers¥etc¥hosts.ics
# Copyright (c) 1993-2001 Microsoft Corp.
#
# This file has been automatically generated for use by Microsoft Internet
# Connection Sharing. It contains the mappings of IP addresses to host names
# for the home network. Please do not make changes to the HOSTS.ICS file.
# Any changes may result in a loss of connectivity between machines on the
# local network.
#
```

```
172.27.28.118 ubuntu-server.mshome.net #
172.27.29.0 ubuntu2004.mshome.net # 2022 4 2 26 2 21 55 3
172.27.16.1 ホストマシン名.mshome.net # 2027 4 0 18 2 21 55 398
----- hosts.ics はこの上の行まで -----
```

※ここに書かれたホスト名で仮想マシンとホストは相互に通信できます。

但し、Hyper-V マネージャの操作で **hosts.ico** が破損することがあります。正常であれば、以下手順で接続できるはずですが

<Windows 側から仮想マシン側に ping コマンドを実行>.

```
C:¥Users¥User>ping ubuntu2004.mshome.net
```

```
ubuntu2004.mshome.net [172.27.29.0]に ping を送信しています 32 バイトのデータ:
172.27.29.0 からの応答: バイト数 =32 時間 <1ms TTL=64
172.27.29.0 からの応答: バイト数 =32 時間 <1ms TTL=64
172.27.29.0 からの応答: バイト数 =32 時間 <1ms TTL=64
172.27.29.0 からの応答: バイト数 =32 時間 =1ms TTL=64
(以下略)
```

<仮想マシンからホストマシン側に ping コマンドを実行>.

```
adminuser@ubuntu2004:~$ ping ホストマシン名.mshome.net
PING ホストマシン名.mshome.net (172.27.16.1) 56(84) バイトのデータ
64 バイト応答 送信元 WIN-GBON7J2169G.mshome.net (172.27.16.1): icmp_seq=1 ttl=128 時間=0.529 ミリ秒
64 バイト応答 送信元 WIN-GBON7J2169G.mshome.net (172.27.16.1): icmp_seq=2 ttl=128 時間=0.879 ミリ秒
64 バイト応答 送信元 WIN-GBON7J2169G.mshome.net (172.27.16.1): icmp_seq=3 ttl=128 時間=0.885 ミリ秒
--- WIN-ホストマシン名.mshome.net ping 統計 ---
送信パケット数 4, 受信パケット数 4, パケット損失 0%, 時間 3005 ミリ秒
rtt 最小/平均/最大/mdev = 0.529/0.801/0.914/0.158 ミリ秒
```

Linux Hyper-V で作るサーバ環境

<ip コマンドによる、仮想マシンのネットワーク構成確認>

```
adminuser@ubuntu2004:~$ ip -r address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet localhost/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ip6-localhost/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:0b:05:07 brd ff:ff:ff:ff:ff:ff
    inet ubuntu2004.mshome.net/20 brd 172.27.31.255 scope global dynamic eth0
        valid_lft 51912sec preferred_lft 51912sec
    inet6 ubuntu2004/64 scope link
        valid_lft forever preferred_lft forever
```

<nslookup コマンドによる、DNS 構成確認>

```
adminuser@ubuntu2004:~$ nslookup ホストマシン名.mshome.net
Server:          127.0.0.53
Address:         127.0.0.53#53
```

Non-authoritative answer:

```
Name:   ホストマシン名.mshome.net
Address: 172.27.16.1
```

※nslookup コマンドで表示される Server は DNS の参照先ですが、実際は 127.0.0.53#53 は Ubuntu が使うキャッシュサーバで、初めての名前は以下の DNS サーバを参照します

```
adminuser@ubuntu2004:~$ systemd-resolve --status
```

Global

(中略)

Link 2 (eth0)

```
    Current Scopes: DNS
DefaultRoute setting: yes
    LLMNR setting: yes
MulticastDNS setting: no
    DNSOverTLS setting: no
    DNSSEC setting: no
    DNSSEC supported: no
    Current DNS Server: 172.27.16.1    ←DNS サーバ
    DNS Servers: 172.27.16.1
    DNS Domain: mshome.net
```

※hosts.ics が破損 (同一ホスト名がある、途中で切れている等) している場合はバックアップを取った後、notepad (メモ帳) を「管理者として実行」で開き改修してください

Linux Hyper-V で作るサーバ環境

(2) DHCP/固定 IP について

Hyper-V でネットワークに Default Switch を指定した仮想マシンは自動的にインターネット接続共有を使うように構成され、NAT と DHCP が有効になります。仮想マシン内では、以下の定義が作られて dhcp によりネットワーク構成（ドメイン名、IP アドレス、DNS サーバ IP アドレス）を取得します。

※この Linux ディストリビューションではネットワークの設定に netplan を使っており、ネットワーク構築時に/etc/netplan/xxxx.yml（xxxx 部分は任意文字列）ファイルが参照されます

```
adminuser@ubuntu2004:~$ cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    eth0:
      dhcp4: true
  version: 2
```

※dhcp4 を false にし、IP アドレスとデフォルト・ゲートウェイを指定すると固定 IP にできますが、再起動すると NAT が正常に動作しなくなる可能性があります。

固定 IP を使いたい場合は、仮想（内部）スイッチと NAT ゲートウェイを作成³し、仮想マシンを接続します（作業自体はコマンドを幾つか叩く程度です）。

(3) 名前解決の簡易な方法

Hyper-V (Windows) を使うと手軽に個人的な開発・試験用の Linux サーバを作ることができますが、デフォルトでは DHCP による IP アドレスの変化が問題になるかもしれません。

Linux サーバに Samba をインストールするとホスト名（Linux の hostname）でサーバを検索できるようになります。Samba は Unix/Linux に Windows サーバの機能（ファイル・プリンタ共有、ドメインコントローラ）を持たせるソフトウェアで、名前解決に Windows で使っている NetBIOS を使います。インストールは apt-get install コマンドで行います。

```
adminuser@ubuntu2004:~$ sudo apt-get install samba
```

```
[sudo] adminuser のパスワード:
```

```
パッケージリストを読み込んでいます... 完了
```

```
依存関係ツリーを作成しています
```

```
状態情報を読み取っています... 完了
```

```
以下の追加パッケージがインストールされます:
```

```
attr ibverbs-providers libavahi-client3 libavahi-common-data libavahi-common3 libboost-iostreams1.71.0
libboost-thread1.71.0 libcephfs2 libcups2 libibverbs1 libjansson4 libldb2 libnl-route-3-200 librados2 librdmacm1
libtalloc2 libtevent0 libwbclient0 python3-crypto python3-dnspython python3-gpg python3-ldb python3-markdown
python3-packaging python3-pygments python3-pyparsing python3-samba python3-talloc python3-tdb samba-common
samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
```

```
提案パッケージ:
```

```
cups-common python-markdown-doc python-pygments-doc ttf-bitstream-vera python-pyparsing-doc bind9 bind9utils ctdb
ldb-tools ntp | chrony smbldap-tools winbind heimdal-clients
```

```
以下のパッケージが新たにインストールされます:
```

```
attr ibverbs-providers libavahi-client3 libavahi-common-data libavahi-common3 libboost-iostreams1.71.0
libboost-thread1.71.0 libcephfs2 libcups2 libibverbs1 libjansson4 libldb2 libnl-route-3-200 librados2 librdmacm1
```

³ マイクロソフト > NAT ネットワークの作成 <https://docs.microsoft.com/ja-jp/virtualization/hyper-v-on-windows/user-guide/setup-nat-network>

Linux Hyper-V で作るサーバ環境

```
libtalloc2 libtevent0 libwbclient0 python3-crypto python3-dnspython python3-gpg python3-ldb python3-markdown
python3-packaging python3-pygments python3-pyparsing python3-samba python3-talloc python3-tdb samba samba-common
samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
アップグレード: 0 個、新規インストール: 36 個、削除: 0 個、保留: 28 個。
17.3 MB のアーカイブを取得する必要があります。
この操作後に追加で 102 MB のディスク容量が消費されます。
続行しますか? [Y/n] y ←Yまたは空[Enter]でインストールを開始します
```

インストールが終わり `systemctl status` コマンドで `nmbd` が active なのを確認できれば正常です。

```
adminuser@ubuntu2004:~$ sudo systemctl status nmbd
```

```
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-04-22 11:47:20 JST; 42s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 20364 (nmbd)
    Status: "nmbd: ready to serve connections..."
     Tasks: 1 (limit: 988)
    Memory: 2.7M
    CGroup: /system.slice/nmbd.service
            └─20364 /usr/sbin/nmbd --foreground --no-process-group
```

```
4月 22 11:47:20 ubuntu2004 systemd[1]: Starting Samba NMB Daemon...
```

```
4月 22 11:47:20 ubuntu2004 systemd[1]: Started Samba NMB Daemon.
```

この後、Linux サーバを再起動すると...

- ① DHCP の IP 再取得により `hosts.ics` が書き換わります

```
# Copyright (c) 1993-2001 Microsoft Corp.
#
# This file has been automatically generated for use by Microsoft Internet
# Connection Sharing. It contains the mappings of IP addresses to host names
# for the home network. Please do not make changes to the HOSTS.ICS file.
# Any changes may result in a loss of connectivity between machines on the
# local network.
#
```

```
172.27.26.104 ubuntu-server.mshome.net # 2022 4 2 26 1 26 13 787
```

```
172.27.16.163 ubuntu2004.mshome.net # 2022 4 5 29 3 4 32 346
```

```
172.27.16.1 WIN-GBON7J2169G.mshome.net # 2027 4 3 21 3 4 32 346
```

```
--- ubuntu2004.mshome.net の再起動前の IP アドレス ---
```

```
172.27.29.0 ubuntu2004.mshome.net # 2022 4 2 26 2 21 55 3
```

- ② Linux サーバの `hostname`

```
adminuser@ubuntu2004:~$ hostname
```

```
ubuntu2004
```

- ③ Linux サーバの IP アドレスが変わっても `hostname` で正常に通信ができます

```
C:%Users%User>ping ubuntu2004
```

```
ubuntu2004 [172.27.16.163]に ping を送信しています 32 バイトのデータ:
```

```
172.27.16.163 からの応答: バイト数 =32 時間 <1ms TTL=64
```

```
172.27.16.163 からの応答: バイト数 =32 時間 =1ms TTL=64
```

以上