

# 普段使っているPCにLinuxをインストールして NGSデータ解析を行うには？

---

国立研究開発法人 農業・食品産業技術総合研究機構  
沼 寿隆 (NUMA Hisataka)

# 実習の概要

デスクトップPCにLinuxをインストールして午前のRNA-seq実習と  
全く同じ解析を行う

# この実習の目標

Linuxサーバで行った実習を再現できる環境をデスクトップPCに構築する

LinuxをインストールするがLinux専用PCにはしない

無料のソフトウェアだけで構築する

Windows/Mac/Linuxどのバージョンでも実現できるようにする

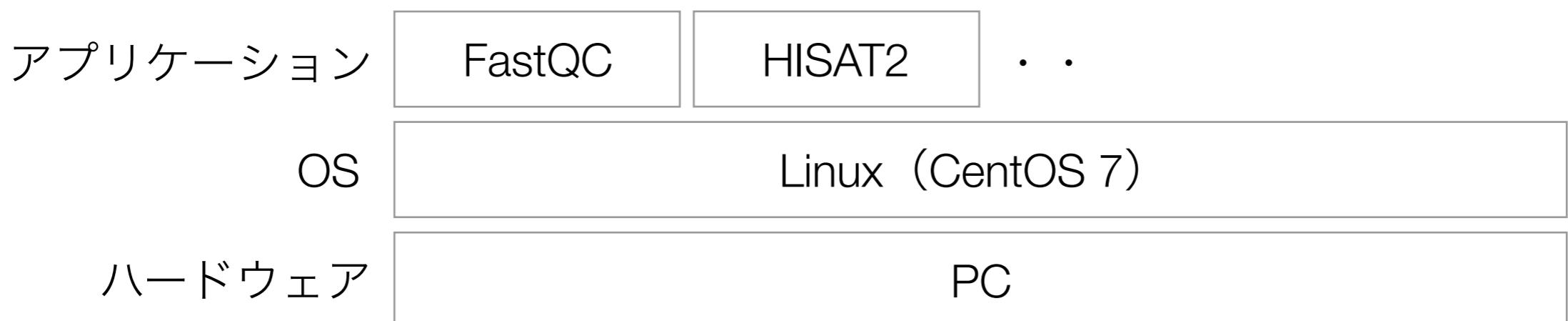
# 仮想化を利用したLinux環境の構築

仮想化でLinuxとWindows（またはMac）を共存させる

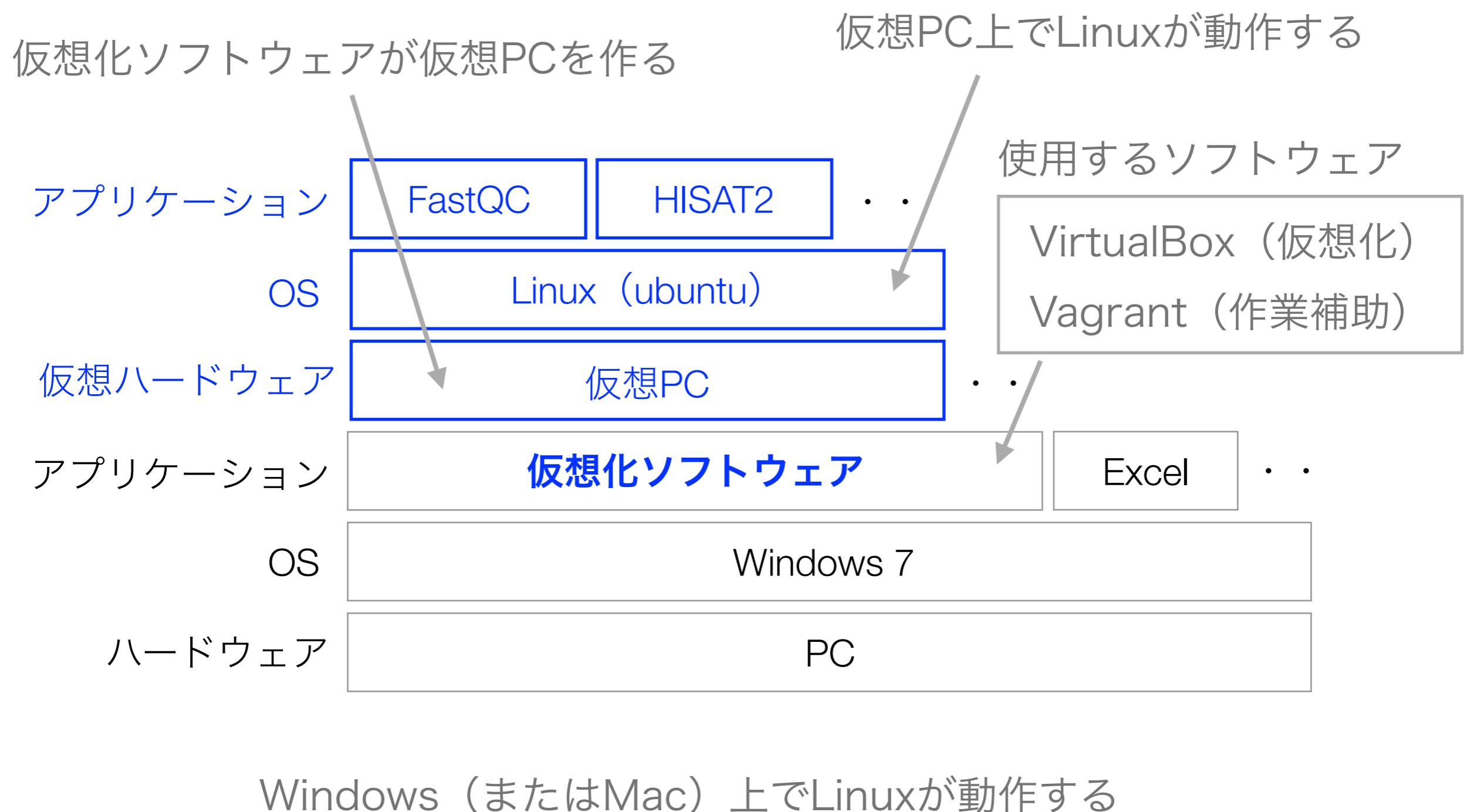
## 実習で使用しているPCの構成



## 実習で使用したLinuxサーバの構成



# この実習で構築するLinuxの構成



VirtualBox (<https://www.virtualbox.org/>)

無料の仮想化ソフトウェア

マウス操作が可能なGUIツール

使用するバージョンは5.2.18 (2018年9月5日現在の最新版)

Vagrant (<https://www.vagrantup.com/>)

仮想化ソフトウェアの操作を補助するソフトウェア

コマンド入力が必要なコマンドラインツール

使用するバージョンは2.1.4 (2018年9月5日現在の最新版)

# 仮想化ソフトウェアのインストール

VirtualBoxとVagrantをインストールする

# VirtualBoxのインストール

## 1. VirtualBoxのWebサイトからインストーラをダウンロードする



The screenshot shows the 'Download VirtualBox' page. On the left, there's a sidebar with links to 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. The main content area has a large 'VirtualBox' logo at the top. Below it, the 'VirtualBox binaries' section is visible, which includes a note about agreeing to terms and conditions, a link to 'VirtualBox 5.1 builds', and a list of 'VirtualBox 5.2.18 platform packages' for Windows hosts, OS X hosts, Linux distributions, and Solaris hosts. There are also notes about GPL version 2, a changelog, checksums, and a note about upgrading guest additions.

Windows版, Mac版, Linux版あり

<https://www.virtualbox.org/wiki/Downloads>

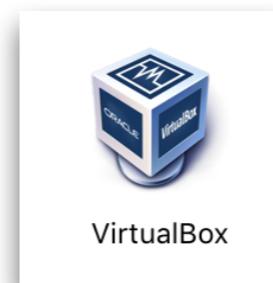
## 2. インストーラの指示に従ってインストールする

Windows



デスクトップに  
ショートカットが作成される

Mac



アプリケーションフォルダに  
インストールされる

# Vagrantのインストール

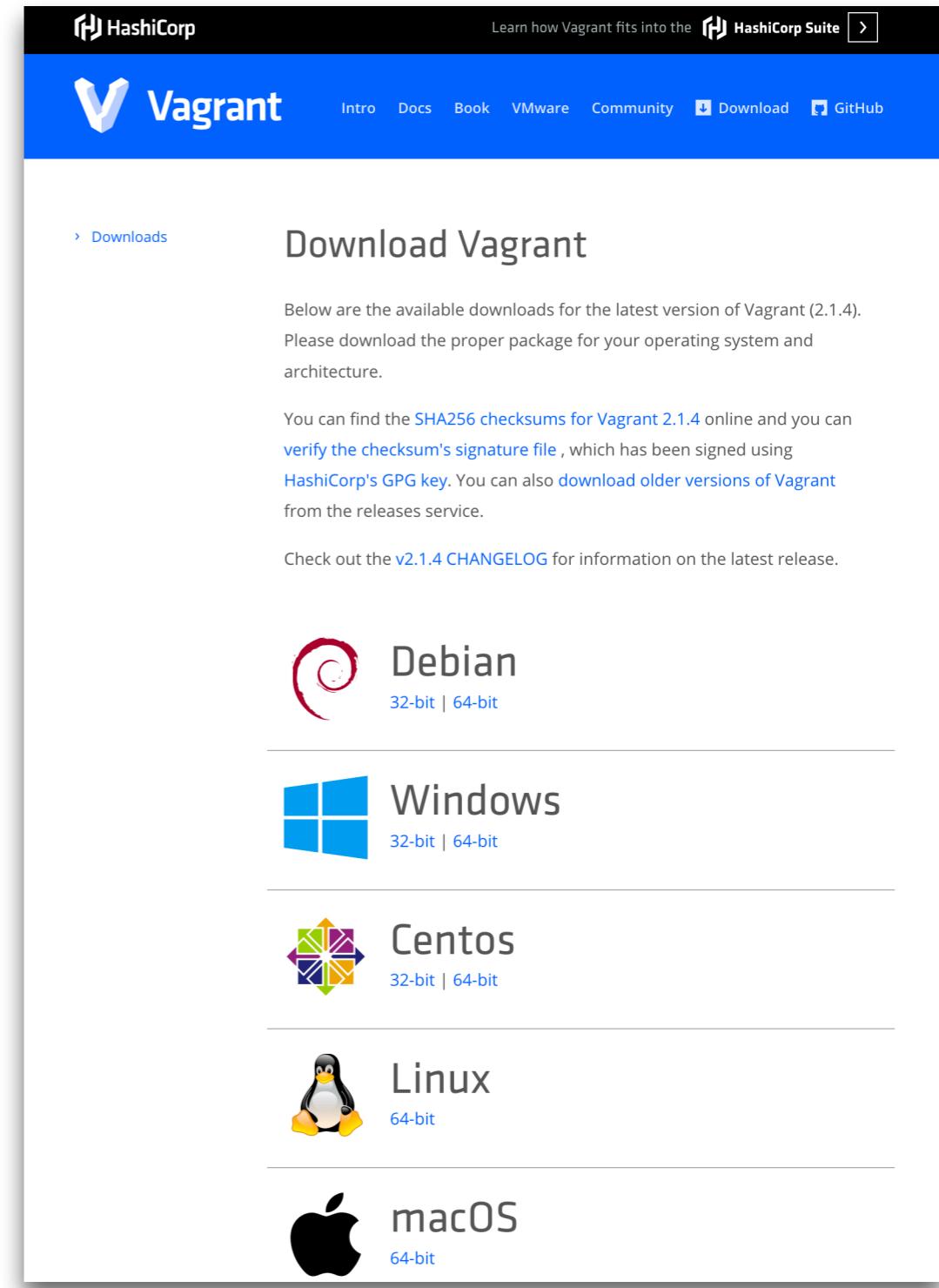
1. VagrantのWebサイトからインストーラをダウンロードする

2. インストーラの指示に従ってインストールする

コマンドツールなのでVirtualBoxとは違ってショートカットはない

Windows版, Mac版, Linux版あり

<https://www.vagrantup.com/downloads.html>



## 【参考】Windowsでの注意点

この後Linuxをインストールして起動しますが、起動できない場合があります。

### 原因

起動にはPCの仮想化支援機能（VT-x）が必要ですが、一部のPCではVT-xがオフになっている場合があります（そもそもVT-xがないPCもあります）。

### 対処法

BIOSの設定でVT-xをオンにします。方法はPCによって若干異なります。例えばHPのPCの場合は以下に方法が記載されています。

HP ワークステーション PC - BIOS で仮想化テクノロジーを有効/無効にする  
(<https://support.hp.com/jp-ja/document/c04773368>)

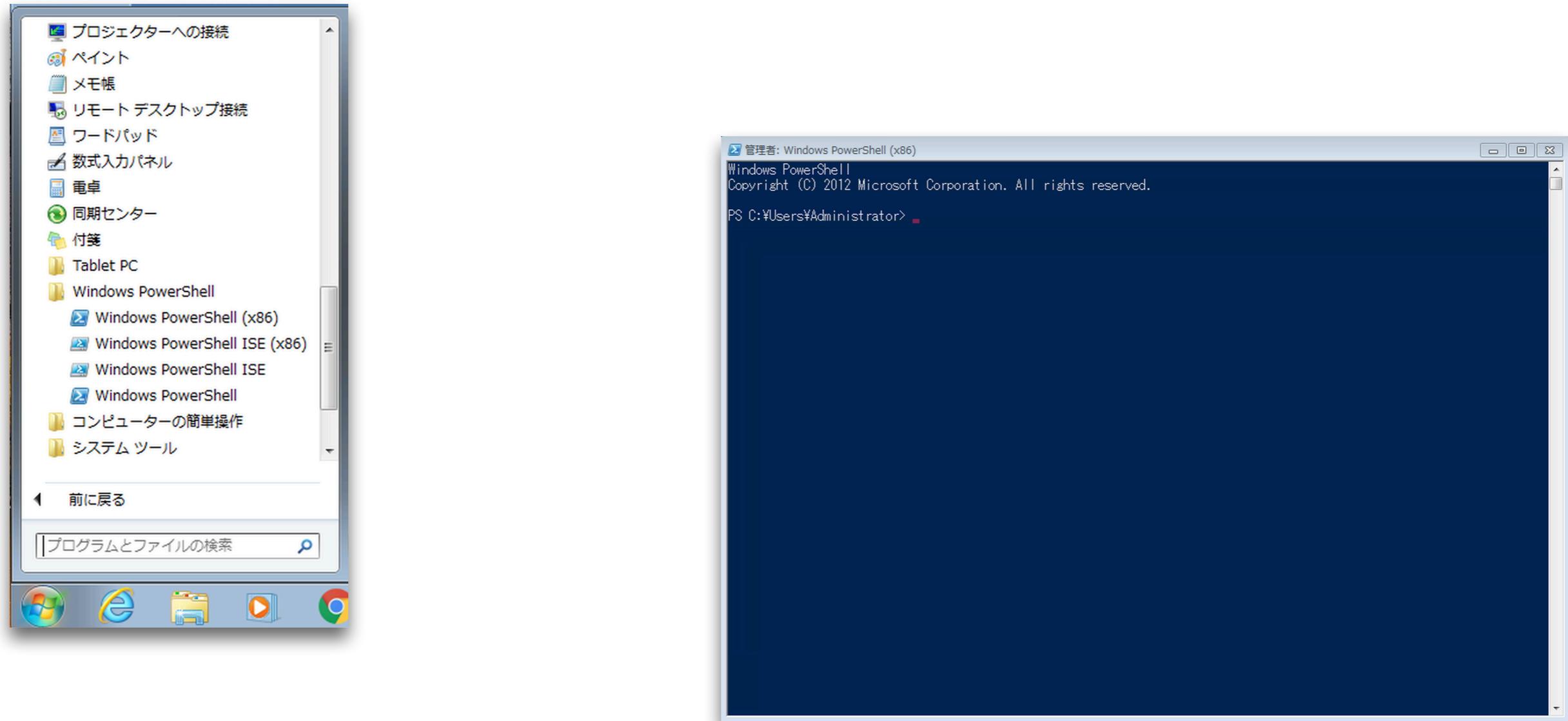
# コマンド入力のための準備

Vagrantを利用するためのコマンド入力ツールを準備する

# PowerShellの起動（Windows）

PowerShellを起動する

すべてのプログラム▶アクセサリ▶Windows PowerShell▶Windows PowerShell (x86)

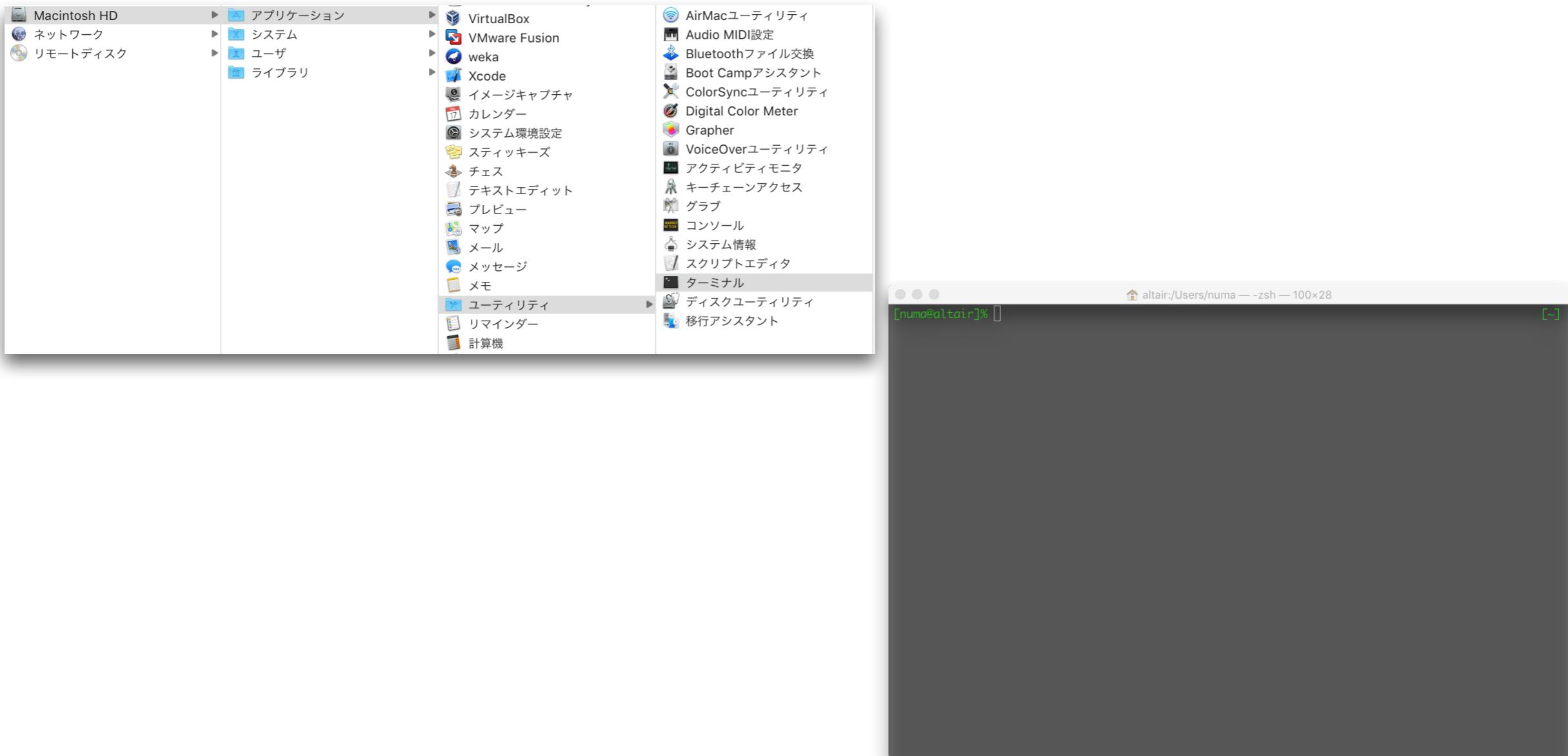


PowerShell起動画面

# ターミナルの起動 (Mac)

ターミナルを起動する

アプリケーション▶ユーティリティ▶ターミナル



ターミナル起動画面

## 【参考】Windows 7での注意点

この後Linuxをインストールして起動しますが、起動できない場合があります。

### 原因

Windows 7にインストールされているPowerShellのバージョンが古い可能性があります。  
PowerShell 3.0以上が必要です。

### 確認方法

PowerShellに以下のコマンドを入力してEnterを押します。

```
Get-Host | Select-Object Version
```

画面にPowerShellのバージョンが表示されます。

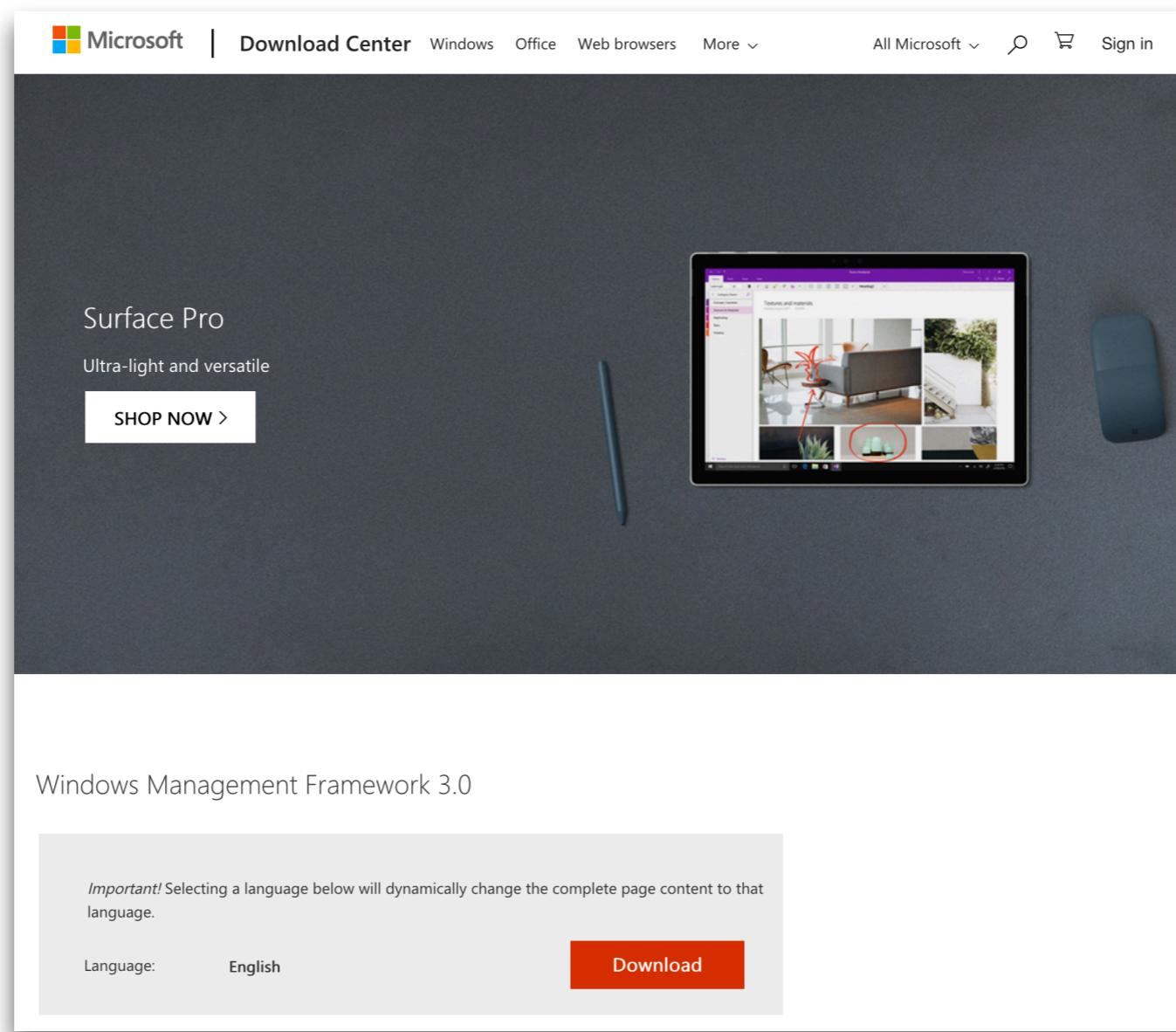
```
Version
```

```
-----
```

```
2.0
```

# 【参考】Windows 7での注意点

バージョンが3.0未満の場合はPowerShell 3.0をインストール必要があります。  
インストーラはMicrosoftダウンロードセンターからダウンロードできます。



<https://www.microsoft.com/en-us/download/details.aspx?id=34595>

# 【参考】Windows 7での注意点

Windows6.1-KB2506143-x64.msuをチェックしてNextボタンを押してダウンロードします。

Choose the download you want

File Name	Size
<input checked="" type="checkbox"/> Windows6.1-KB2506143-x64.msu	15.8 MB
<input type="checkbox"/> Windows6.0-KB2506146-x64.msu	14.4 MB
<input type="checkbox"/> Windows6.0-KB2506146-x86.msu	10.5 MB
<input type="checkbox"/> Windows6.1-KB2506143-x86.msu	11.7 MB
<input type="checkbox"/> WMF 3 Release Notes.docx	53 KB

Download Summary:  
KBMBGB

1. Windows6.1-KB2506143-x64.msu

Total Size: 15.8 MB

Next

ダウンロードしたWindows6.1-KB2506143-x64.msuを起動してインストールします。  
(インストール後に再起動が必要)

# Linuxのインストールと設定

仮想PCを作成してLinuxをインストールする

1. Linuxをダウンロードする
2. 設定ファイルなどを保存するフォルダを作成する
3. 仮想PCの設定ファイルを作成する
4. 仮想PCの作成とLinuxのインストール・設定・起動を行う
5. Linuxにログインする
6. Linuxを停止する

# コマンドの表記

%で始まる行が入力するコマンド  
(先頭の%は入力しない)

```
% vagrant -v  
Vagrant 2.1.4  
(バージョンが表示される)
```

画面に表示されるコマンドの実行結果  
(何も表示がないコマンドもある)

括弧書きの行は補足説明  
(画面には表示されない)

## 使用するフォント

```
0123456789  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz
```

(ゼロとオーの違いに注意)

# Linuxをダウンロードする

ダウンロード済みのLinuxがあるか確認する

```
% vagrant box list  
There are no installed boxes! Use `vagrant box add` to add some.
```

まだダウンロードしていないので何もない

Box : OSのインストールに必要なファイルをまとめたパッケージ

The screenshot shows the Vagrant Cloud homepage. At the top, there's a navigation bar with the HashiCorp logo, a search bar, and links for Pricing, Vagrant, Help, Create an Account, and Sign In. Below the header, a large search bar is centered with the placeholder text "Discover Vagrant Boxes". Underneath the search bar, there are filters for "Provider" (set to "any") and sorting options ("Sort by" dropdown set to "Downloads", with "Recently Created" and "Recently Updated" also available). The main content area displays a list of three Vagrant boxes:

Box Name	Provider	Downloads	Released
ubuntu/trusty64 20180824.0.7	virtualbox	30,221,414	about 8 hours ago
laravel/homestead 6.3.0	hyperv, parallels, virtualbox, vmware_desktop	13,115,068	17 days ago
hashicorp/precise64 1.1.0	hyperv, virtualbox, vmware_fusion	6,685,363	over 4 years ago

利用可能なbox一覧 (<https://app.vagrantup.com/boxes/search>)

# Linuxをダウンロードする

---

## Linuxをダウンロードする

```
% vagrant box add ubuntu/bionic64
```

(進捗状況が表示される)

```
==> box: Successfully added box 'ubuntu/bionic64' (v20180907.0.0) for 'virtualbox'!
```

完了までに5~10分程度かかる

ダウンロードするLinuxはubuntu 18.04 LTS (ubuntu/bionic64)

ubuntu/bionic64はubuntuの開発元がリリースしている公式パッケージ



## ダウンロードしたLinuxを確認する

```
% vagrant box list
```

```
ubuntu/bionic64 (virtualbox, 20180907.0.0)
```

実習で使用したLinuxサーバ

CentOS 7



**CentOS®**

<https://www.centos.org>

PCにインストールするLinux

ubuntu 18.04 LTS



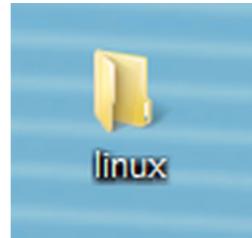
**ubuntu**

<https://www.ubuntu.com>

# 設定ファイルなどを保存するフォルダを作成する

---

デスクトップにlinuxという名前でフォルダを作成する



右クリックで普通に作成する (PowerShellやターミナルのコマンド入力でも作成できる)

任意の場所に任意の名前で作成可能

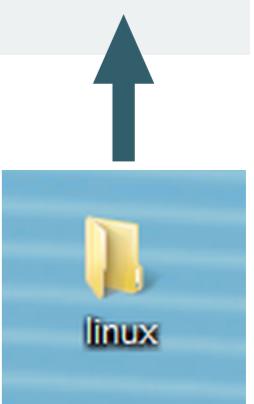
仮想PCの設定ファイルなどが保存される

# 仮想PCの設定ファイルをlinuxフォルダに作成する

cdの後にスペースを入力してからlinuxフォルダをPowerShellウィンドウ内にドラッグ&ドロップする

```
% cd
```

Enterキーはまだ押さない



linuxフォルダの場所が入力されるのでEnterキーを押す

```
% cd C:\Users\Administrator\Desktop\linux
```

linuxフォルダに移動できた

以降の作業はすべてlinuxフォルダで行う

【参考】 Macの場合は以下のような表示になる

```
% cd /Users/Administrator/Desktop/linux
```

# 仮想PCの設定ファイルをlinuxフォルダに作成する

linuxフォルダの中身を確認する

```
% ls
```

まだ何も作成してないので中身は空

設定ファイルを作成する

↓  
ダウンロードしたLinuxを指定する

```
% vagrant init ubuntu/bionic64
```

A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on `vagrantup.com` for more information on using Vagrant.

設定ファイルが作成されていることを確認する

```
% ls  
Vagrantfile
```

Vagrantfileが仮想PCの設定ファイル（テキストファイルなのでメモ帳などで編集可能）

Vagrantfileを編集することで仮想PCのカスタマイズが可能

# 仮想PCの作成とLinuxのインストール・設定・起動を行う

---

コマンド1つで一氣に行う

```
% vagrant up  
(進捗状況が表示される)
```

Vagrantfileの内容に基づいて仮想PCが作成される

ログインするための基本的な設定も同時に行われる

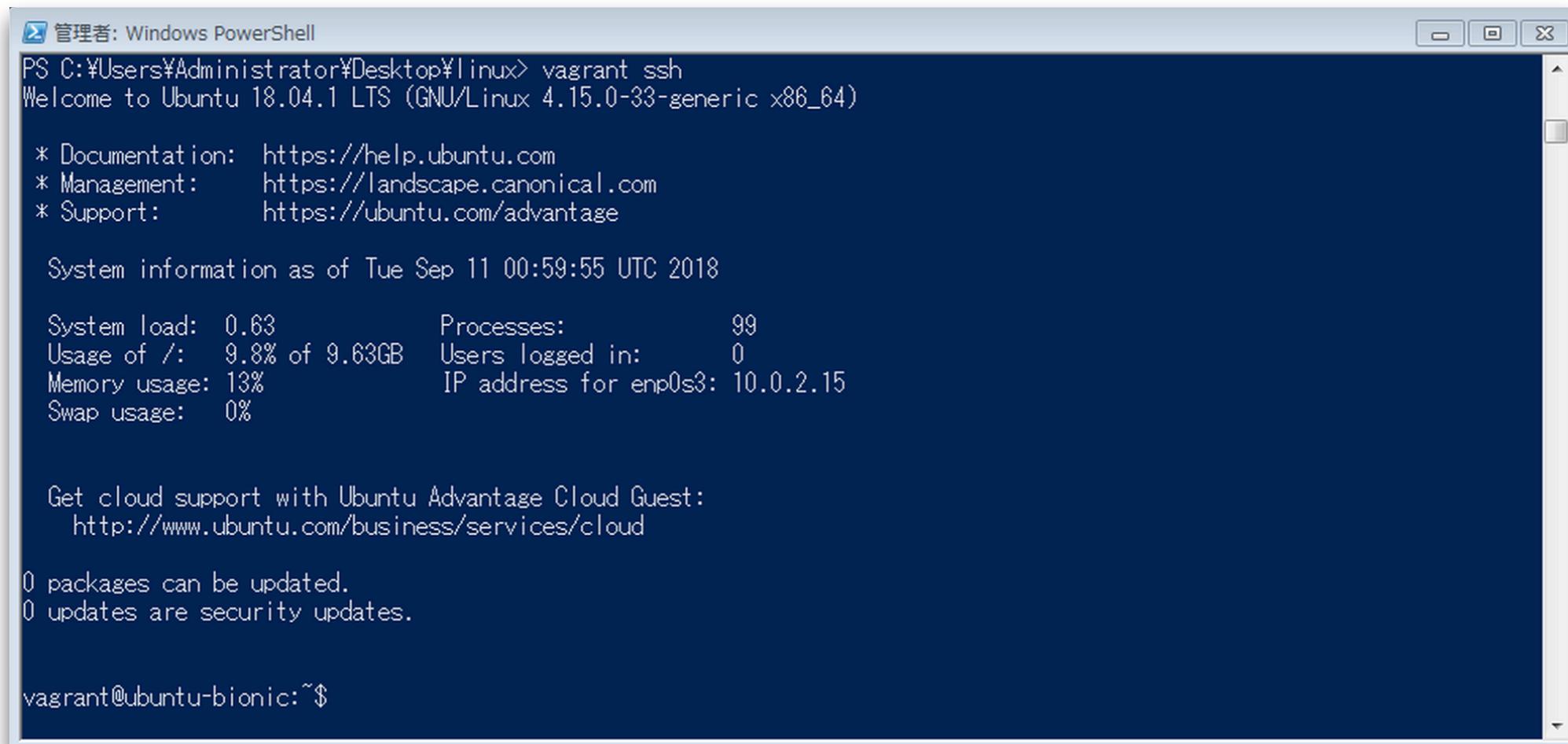
仮想PCの作成とLinuxのインストールが完了するとLinuxが起動する

# Linuxにログインする

## ログインする

```
% vagrant ssh  
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-33-generic x86_64)  
(途中のメッセージ省略)  
vagrant@ubuntu-bionic:~$
```

プロンプトが変わったことでログインしていることが確認できる



The screenshot shows a Windows PowerShell window titled "管理者: Windows PowerShell". The command "vagrant ssh" was run, resulting in the following output:

```
PS C:\Users\Administrator\Desktop\linux> vagrant ssh  
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-33-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
 System information as of Tue Sep 11 00:59:55 UTC 2018  
  
 System load: 0.63      Processes: 99  
 Usage of /: 9.8% of 9.63GB  Users logged in: 0  
 Memory usage: 13%          IP address for enp0s3: 10.0.2.15  
 Swap usage: 0%  
  
 Get cloud support with Ubuntu Advantage Cloud Guest:  
 http://www.ubuntu.com/business/services/cloud  
  
 0 packages can be updated.  
 0 updates are security updates.  
  
vagrant@ubuntu-bionic:~$
```

ログイン後の画面

# Linuxを停止する

---

## Linuxからログアウトする

```
% exit
```



Linuxにログインしていない状態でexitするとPowerShellが終了してしまう

## Linuxを停止する

```
% vagrant halt  
==> default: Attempting graceful shutdown of VM...
```

Linuxにログインしている状態で入力するコマンドと、ログインしていない状態で入力するコマンドが紛らわしいので、以降ログインしている状態で入力するコマンドには を付ける

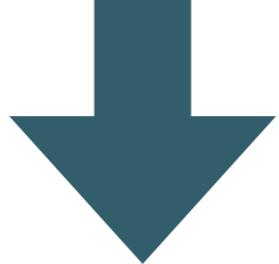
# RNA-Seqによる大量発現データ解析再び

インストールしたLinuxを使って午前の実習と同じ解析を行う

仮想PCのメモリを1GBから4GBに増やす

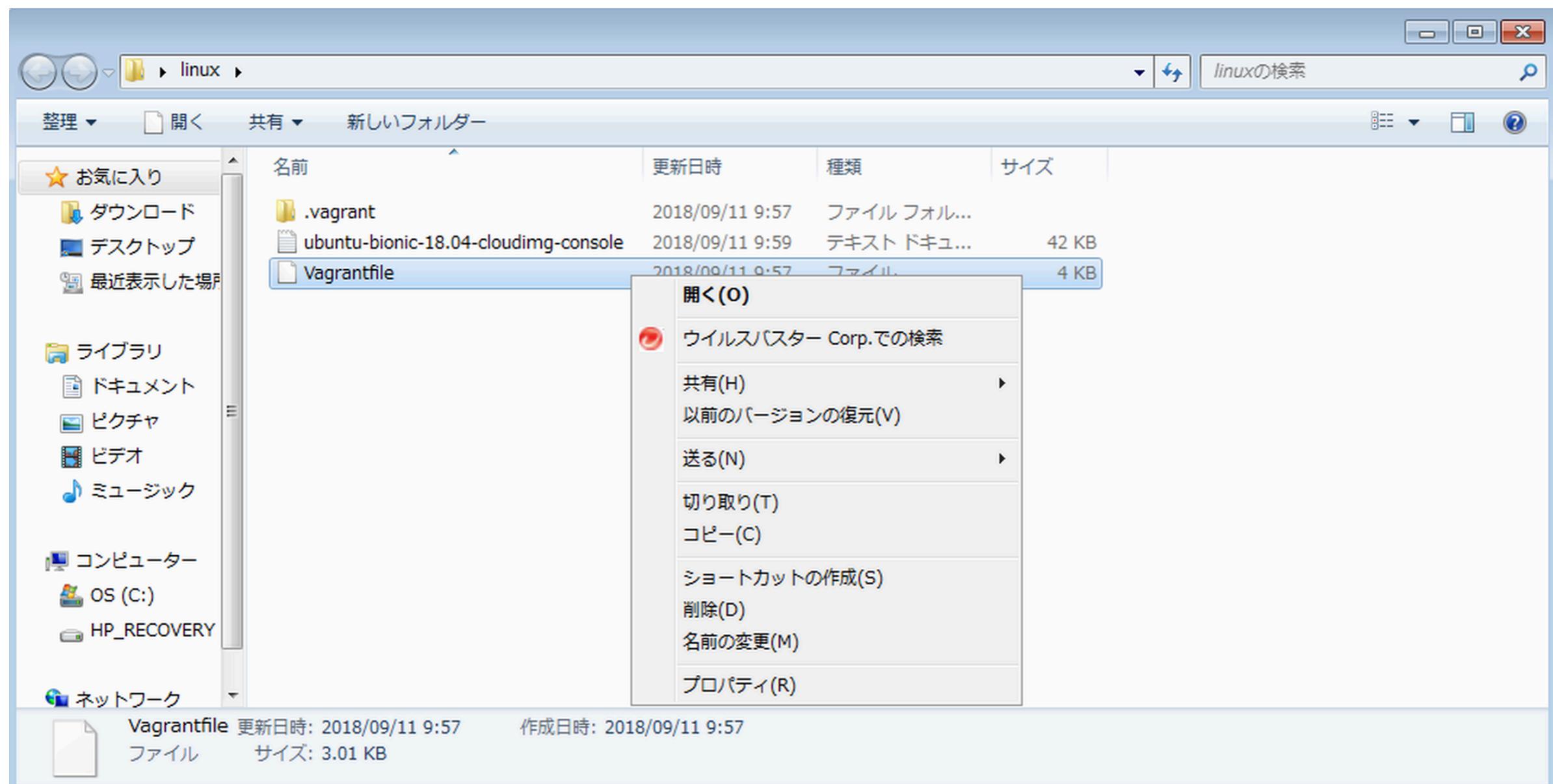
Vagrantは標準で1GBのメモリを持つ仮想PCを作成する

午前のRNA-seqの実習を行うには4GBくらいあった方が良い

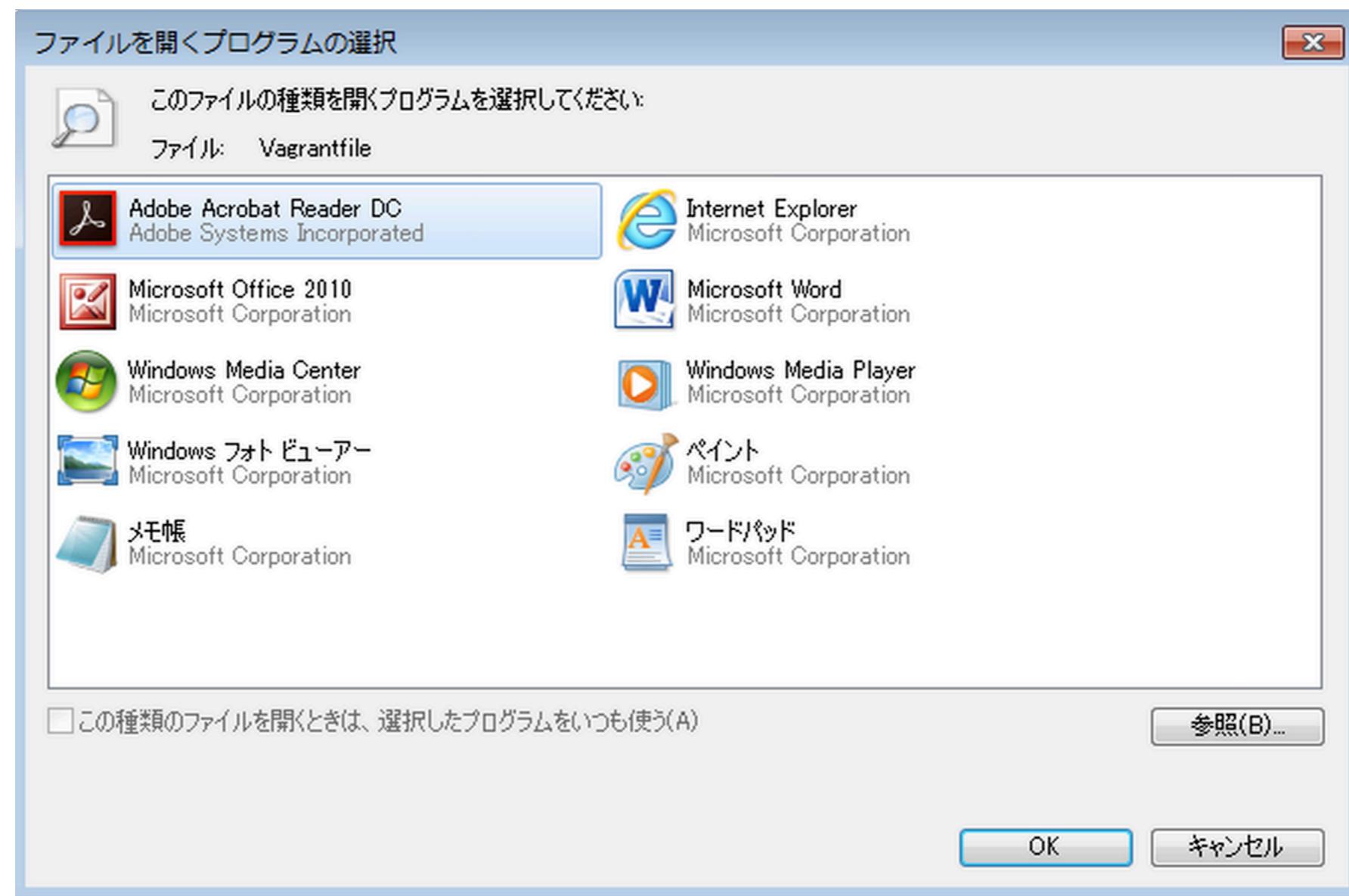


Vagrantfileを修正して仮想PCのメモリを4GBに増設する

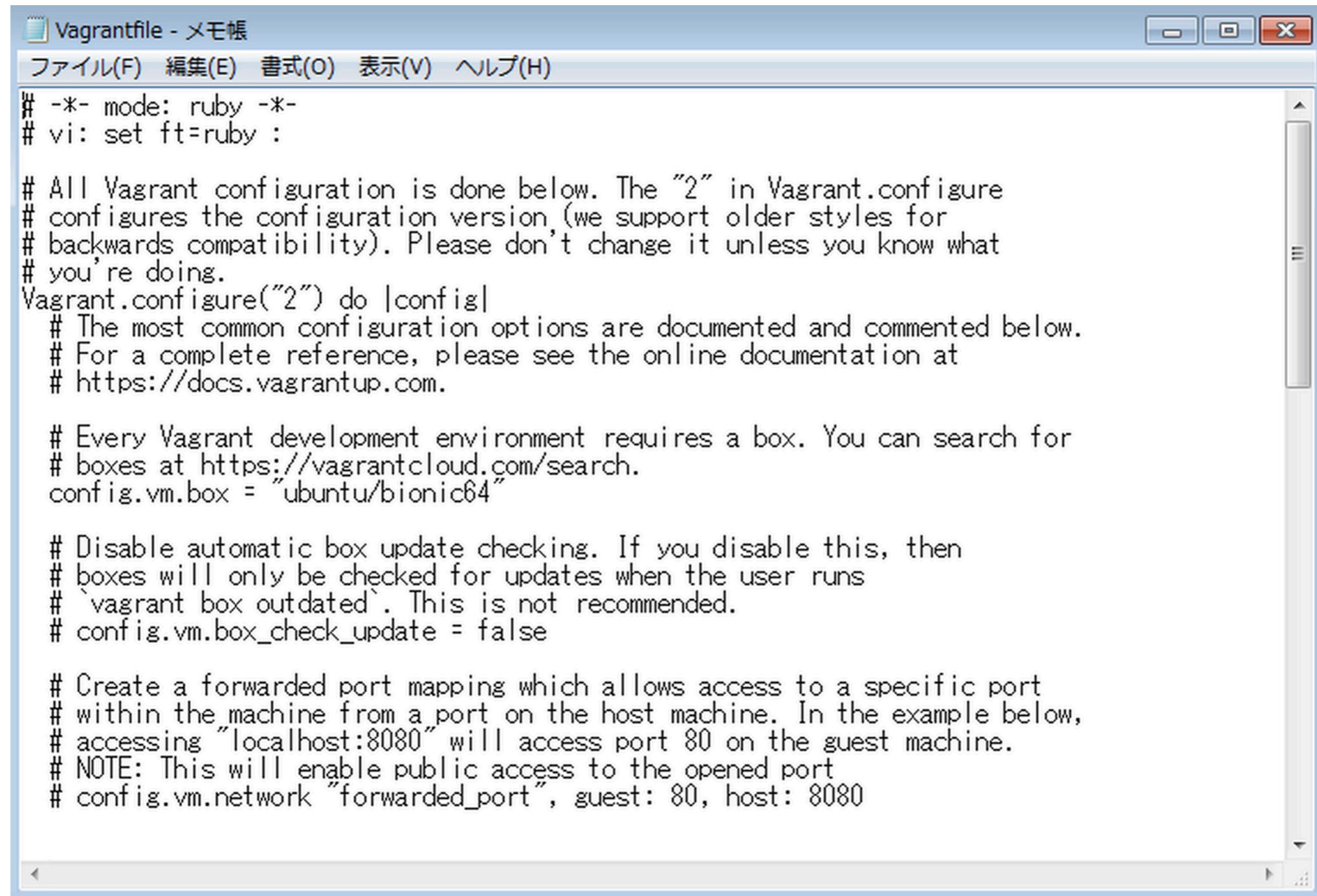
linuxフォルダのVagrantfileを右クリックしてメニューを表示させ  
開くをクリックする



メモ帳を選択してOKを押す



メモ帳が起動してVagrantfileの内容が表示される



```
# -*- mode: ruby -*-
# vi: set ft=ruby :

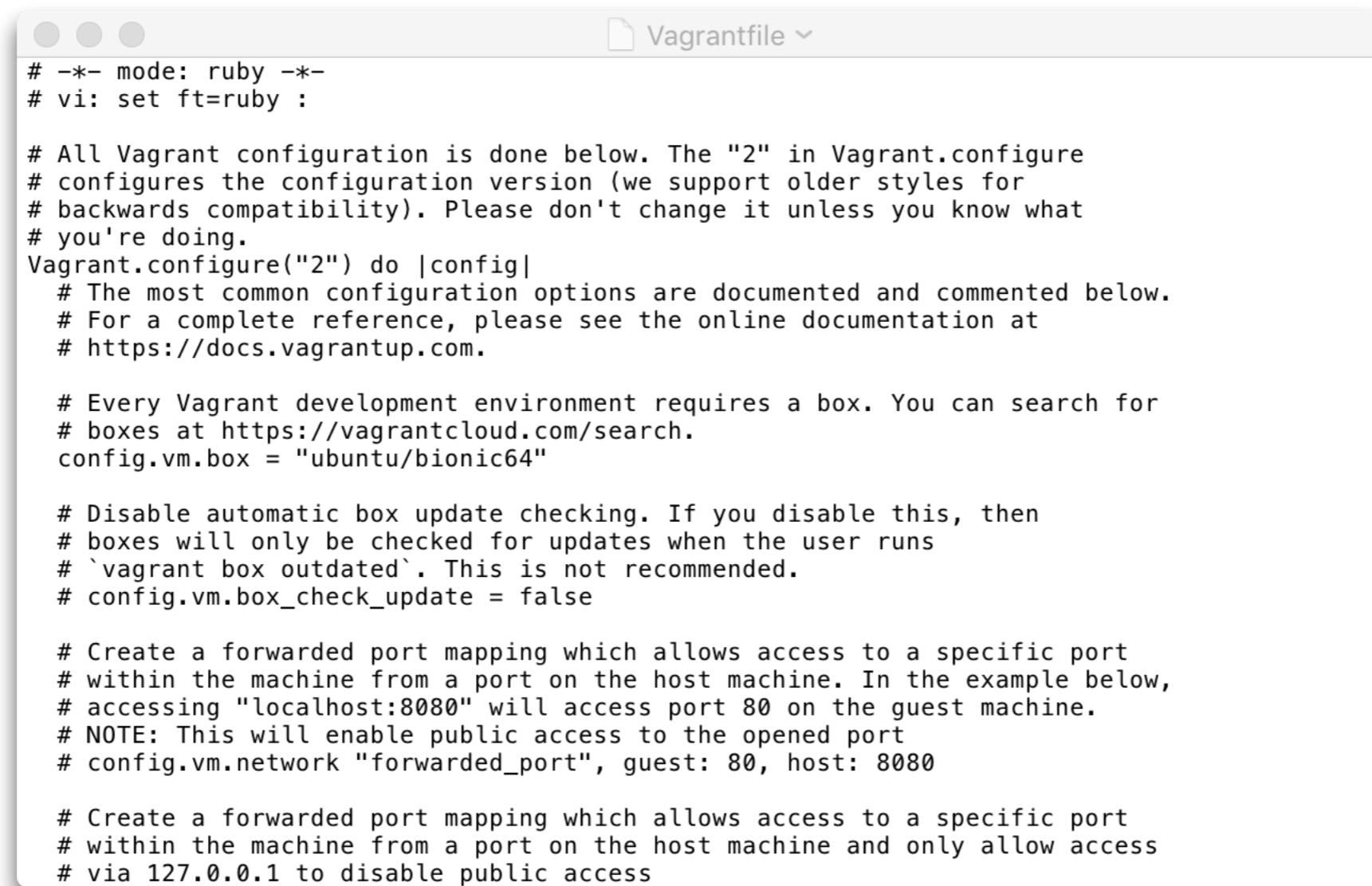
# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/bionic64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080
```

Vagrantfileを2本指クリック▶このアプリケーションで開く▶テキストエディット



The screenshot shows a Mac OS X TextEdit window with a light gray background. The title bar at the top has three circular buttons on the left and the text "Vagrantfile" followed by a dropdown arrow on the right. The main content area contains the Vagrantfile code, which is a Ruby script. The code includes comments explaining the configuration options, such as setting the Vagrant version, choosing a box, and creating forwarded port mappings. The font used is a standard monospaced font.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/bionic64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

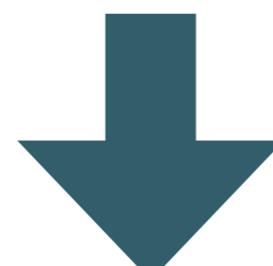
  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine and only allow access
  # via 127.0.0.1 to disable public access
```

Vagrantfileの最後の方を以下の通り4箇所を修正して保存する

① #を削除する ② #を削除する ③ #を削除する

① → # config.vm.provider "virtualbox" do |vb|  
# # Display the VirtualBox GUI when booting the machine  
# vb.gui = true  
  
#  
# # Customize the amount of memory on the VM:  
② → # vb.memory = "1024"  
③ → # end

↑  
④



④ 1024を4096にする

```
config.vm.provider "virtualbox" do |vb|
#   # Display the VirtualBox GUI when booting the machine
#   vb.gui = true
#
#   # Customize the amount of memory on the VM:
#   vb.memory = "4096"
end
```

Vagrantfileに間違いがないことを確認する

```
% vagrant validate  
Vagrantfile validated successfully.
```

次にvagrant upした時にメモリが4GBになった仮想PCが起動する

# Linuxを起動してログインする

---

デスクトップのlinuxフォルダに移動する

```
% cd C:\Users\Administrator\Desktop\linux
```

実習ではすでにlinuxフォルダに移動しているのこの操作は必要ない

Linuxを起動する

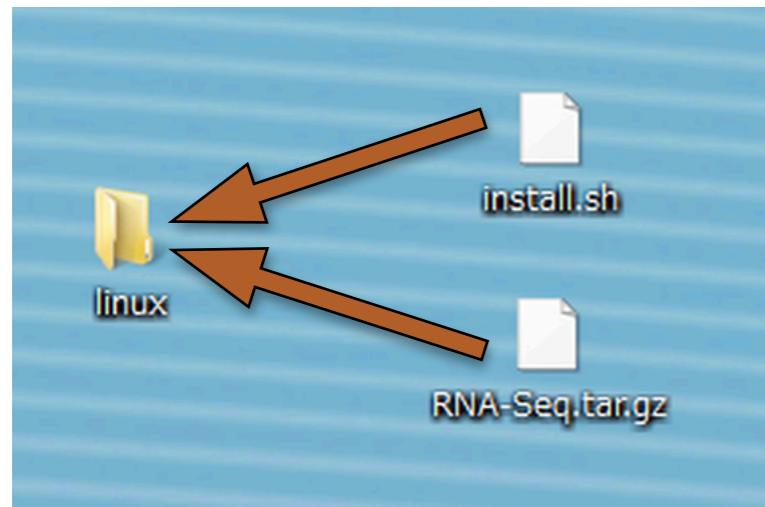
```
% vagrant up  
(進捗状況が表示される)
```

ログインする

```
% vagrant ssh  
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-33-generic x86_64)  
(途中のメッセージ省略)  
vagrant@ubuntu-bionic:~$
```

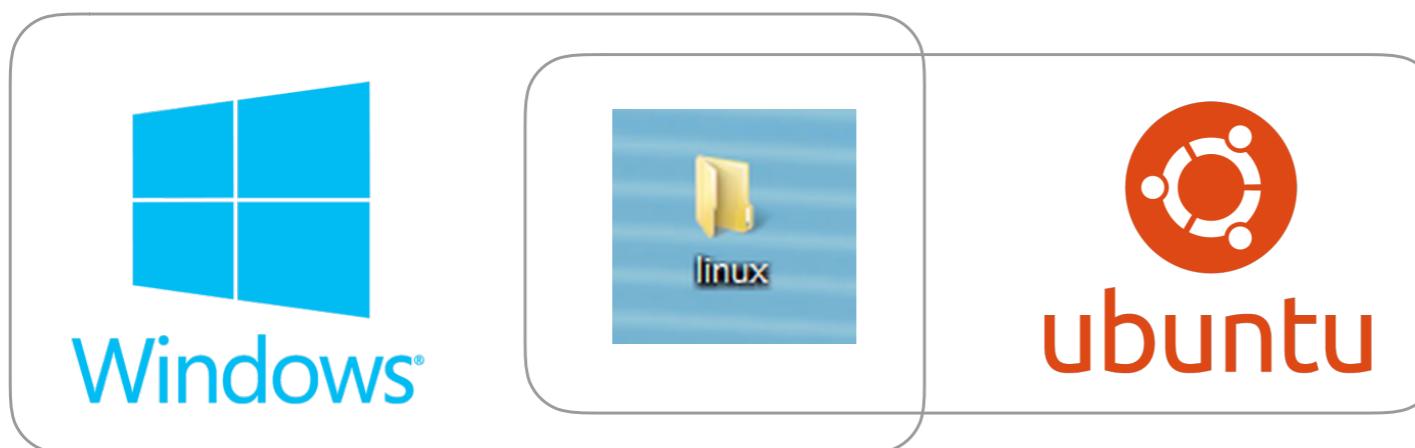
# 実習用ファイル一式をLinuxにコピーする

RNA-Seq.tar.gzをlinuxフォルダにドラッグ&ドロップする



この操作でRNA-Seq.tar.gzをLinuxから使えるようになる  
install.shファイルも同様にドラッグ&ドロップする

WindowsのlinuxフォルダはLinuxと共有されている



VirtualBoxのファイル共有機能を利用  
ファイル共有の設定はVagrantが実施

Windows上のRNA-Seq.tar.gzが共有されているだけなので  
実際にはLinuxにコピーされていない

# 共有ディレクトリを確認する

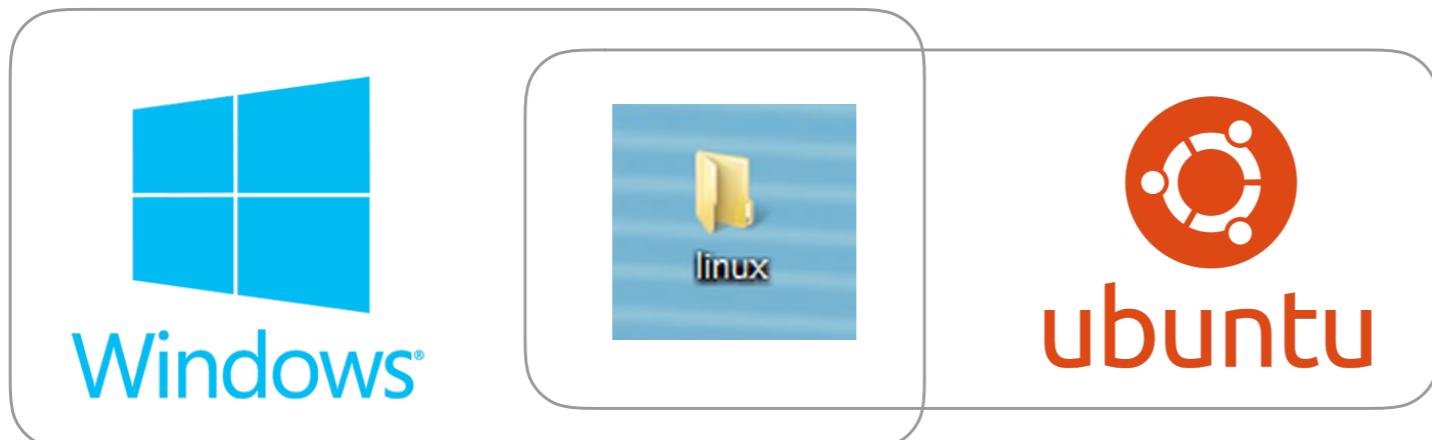
/vagrantディレクトリのファイルを表示する

```
% ls /vagrant
```

RNA-Seq.tar.gz	install.sh	 ubuntu
Vagrantfile	ubuntu-bionic-18.04-cloudimg-console.log	

先ほどドラッグ&ドロップしたRNA-Seq.tar.gzがあることが確認できる

WindowsのlinuxフォルダはLinuxからは/vagrantでアクセスできる



# 実習に必要なツールをインストールする

午前の実習では行わなかった作業（管理者がやっておいてくれた）

リファレンスを使った解析を実行するのに必要なツールをインストールする

```
% /vagrant/install.sh  
(進捗状況が表示される)
```



10分程度かかる

以下のメッセージが表示された場合はインストールが失敗している（時々失敗することがある）

```
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
```

インストールが失敗した場合は再度install.shを実行する

シェルスクリプトの中では以下のコマンドを実行している

```
sudo apt update  
sudo apt -y install unzip build-essential zlib1g-dev libbz2-dev  
libncurses5-dev liblzma-dev openjdk-8-jre python
```

**apt update** でツールのリストを更新する

**apt install ツール名** でツールがインストールできる（ツール名は複数指定可）

**-y** を付けるとインストールの途中でYes/Noを尋ねられることがない（全部Yesになる）

**sudo** を付けると管理者権限でコマンドが実行される（管理者権限でインストールするので必要）

# 【参考】 リファレンスを使わない発現解析も行う場合

リファレンスを使わない発現解析 (RNA-Seqのdenovo アセンブルと発現解析) を行うには以下のコマンドでNumPy, R, edgeRもインストールする必要がある

NumPyをインストールする

```
% sudo apt -y install python-numpy
```



最新のR version 3.5をインストールするための設定をする  
(最新版が必要なれば以下の2つのコマンドは必要ない)

```
% echo "deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/" | sudo tee /etc/apt/sources.list.d/cran.list
```

(表示の都合上2行になっているが改行せずに1行で入力する)

```
% sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9
```

(進捗状況が表示される)

(表示の都合上2行になっているが改行せずに1行で入力する)



# 【参考】 リファレンスを使わない発現解析も行う場合

Rをインストールする

```
% sudo apt update  
(進捗状況が表示される)  
% sudo apt -y install r-base-core  
(進捗状況が表示される)
```



edgeRをインストールする

```
% R  
(Rが起動する)
```



```
> source("https://bioconductor.org/biocLite.R") ← 入力する  
(進捗状況が表示される)  
Would you like to use a personal library instead? (yes/No/cancel) ← yes  
Would you like to create a personal library  
'~/R/x86_64-pc-linux-gnu-library/3.5'  
to install packages into? (yes/No/cancel) ← yes  
(進捗状況が表示される)  
  
> biocLite("edgeR") ← 入力する  
(進捗状況が表示される)  
  
> q() ← 入力する  
Save workspace image? [y/n/c]: ← n
```

# 午前の実習と同じ解析ができるようになった！

実習ファイルを解凍・展開して実習用ディレクトリに移動する

```
% tar xfz /vagrant/RNA-Seq.tar.gz  
% cd RNA-Seq
```



ubuntu

ツールのインストールを行う

```
% bash install_tools.sh
```



ubuntu

リファレンスマッピング～発現解析を実行する

```
% cd useref
```



ubuntu

```
% bash step1_preprocessing.sh
```



ubuntu

```
% bash step2_HISAT2-build.sh
```



ubuntu

```
% bash step3_HISAT2.sh
```



ubuntu

```
% bash step4_StringTie-abundance_estimation.sh
```



ubuntu

# 【参考】 GATKをインストールするには

以下のコマンドでツールをインストール際にGATKのインストールに失敗する

```
% bash install_tools.sh
```



インストールが失敗したことを見せるメッセージ

```
installing GATK...
cp: cannot stat '/work/NGSworkshop/tool/gatk-4.0.8.1.zip': No such file or directory
unzip:  cannot find or open gatk-4.0.8.1.zip, gatk-4.0.8.1.zip.zip or gatk-4.0.8.1.zip.ZIP.
rm: cannot remove 'gatk-4.0.8.1.zip': No such file or directory
```

GATKをインストールするにはinstall\_tools.shの最後から5行目と6行目を  
以下のように修正しなくてはならない

```
# wget https://github.com/broadinstitute/gatk/releases/download/4.0.8.1/gatk-4.0.8.1.zip
cp /work/NGSworkshop/tool/gatk-4.0.8.1.zip ./
```

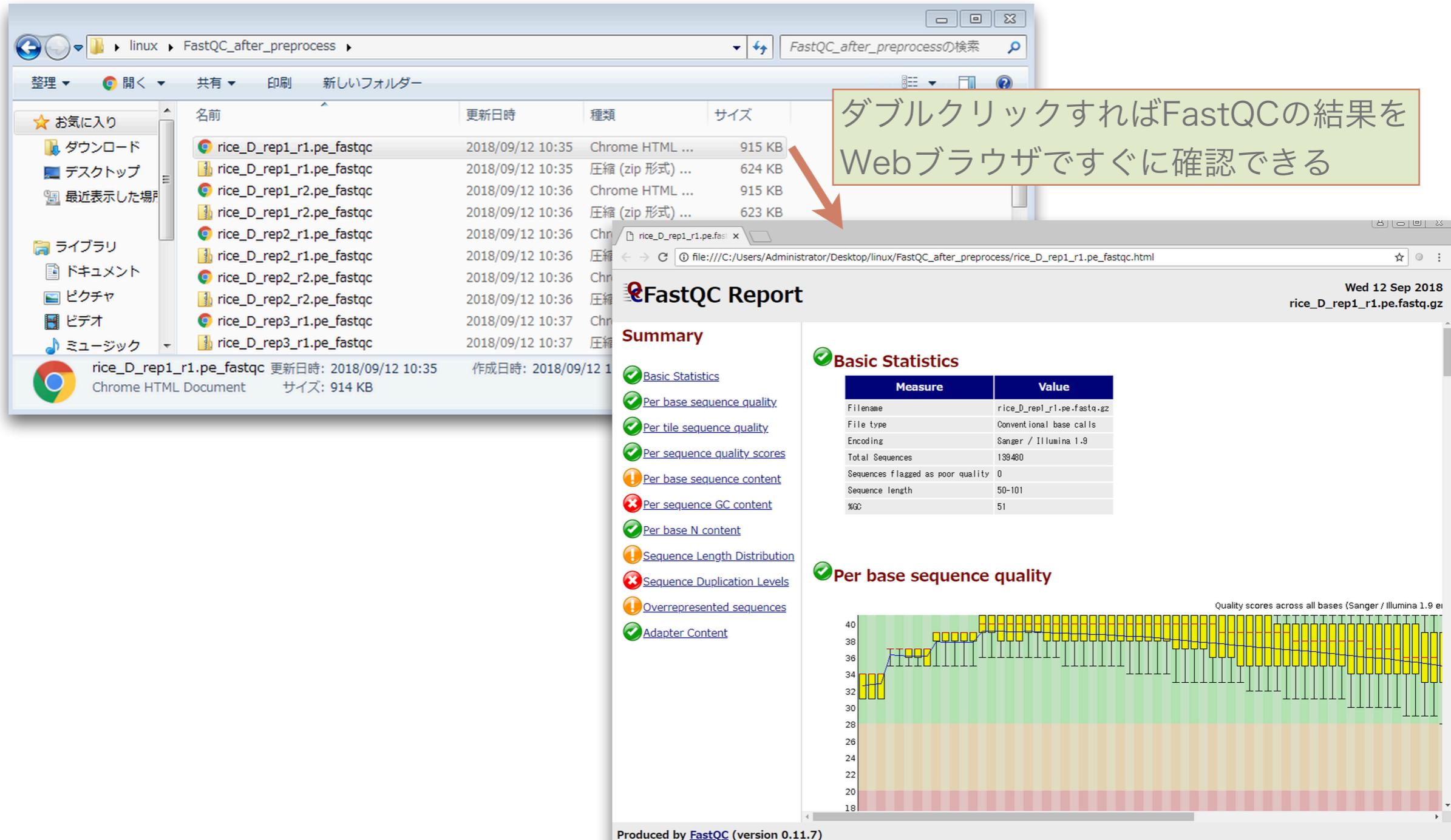


```
wget https://github.com/broadinstitute/gatk/releases/download/4.0.8.1/gatk-4.0.8.1.zip
```

(# wgetで始まる行の#を削除し, cpで始まる行を削除する)

GATKのダウンロードには時間がかかる。午前の実習ではあらかじめダウンロードしておいたファイルを  
使うことで時間を節約している。自分でインストールしたLinuxの場合にはGATKをダウンロードする  
必要があるため、上記のようにスクリプトを修正しなくてはならない。

フォルダが共有できるので解析結果をWindowsやMacとやりとりするのが簡単



ただしその前に/vagrantにファイルをコピーしておく必要がある

```
% cp -a FastQC_after_preprocess /vagrant
```



## 【参考】 /vagrantで解析を実行しなかった理由

/vagrantで解析を実行すれば解析結果をわざわざ/vagrantにコピーしなくてもWindowsと共有できたのでは？

### 理由1

実習で使用したRNA-Seq解析用スクリプトは\$HOME/RNA-Seq/userefで実行することを前提に作られている

→ スクリプトを修正すればどのディレクトリでも実行できるようになる

### 理由2

共有フォルダは遅い

→ 必要なデータだけを共有ディレクトリに  
コピーした方が速い

step1\_preprocessing.shの実行にかかる時間

/vagrant	\$HOME/RNA-Seq/useref
46分16秒	3分43秒



共有フォルダで実行すると46分もかかる！

# 仮想PCをコピーするには？

作成した仮想PCを別のPCで起動するための手順

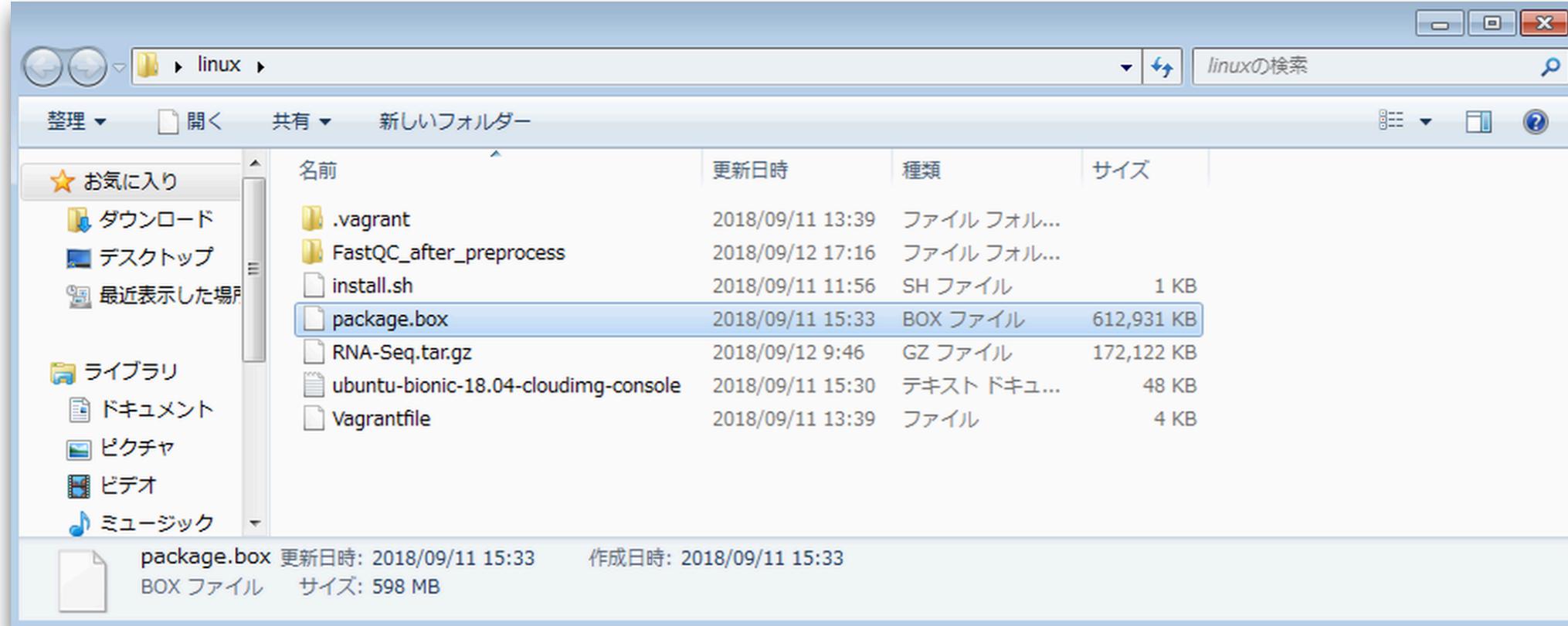
1. Linuxからログアウトする (exit)
2. 仮想PCを停止する (vagrant halt)
3. 仮想PCをパッケージ化したBoxファイルを作成する (vagrant package)
4. Boxファイルを別のPCにコピーする
5. 別のPCにVirtualBoxとVagrantをインストールする
6. コピーしたBoxファイルを追加する (vagrant box add)
7. 追加したBoxを使って仮想PCを作成する (vagrant init / vagrant up)

# 仮想PCをパッケージ化したBoxファイルを作成する

```
% vagrant package  
==> default: Clearing any previously set forwarded ports...  
==> default: Exporting VM...  
==> default: Compressing package to: C:/Users/user01/Desktop/linux/package.box
```

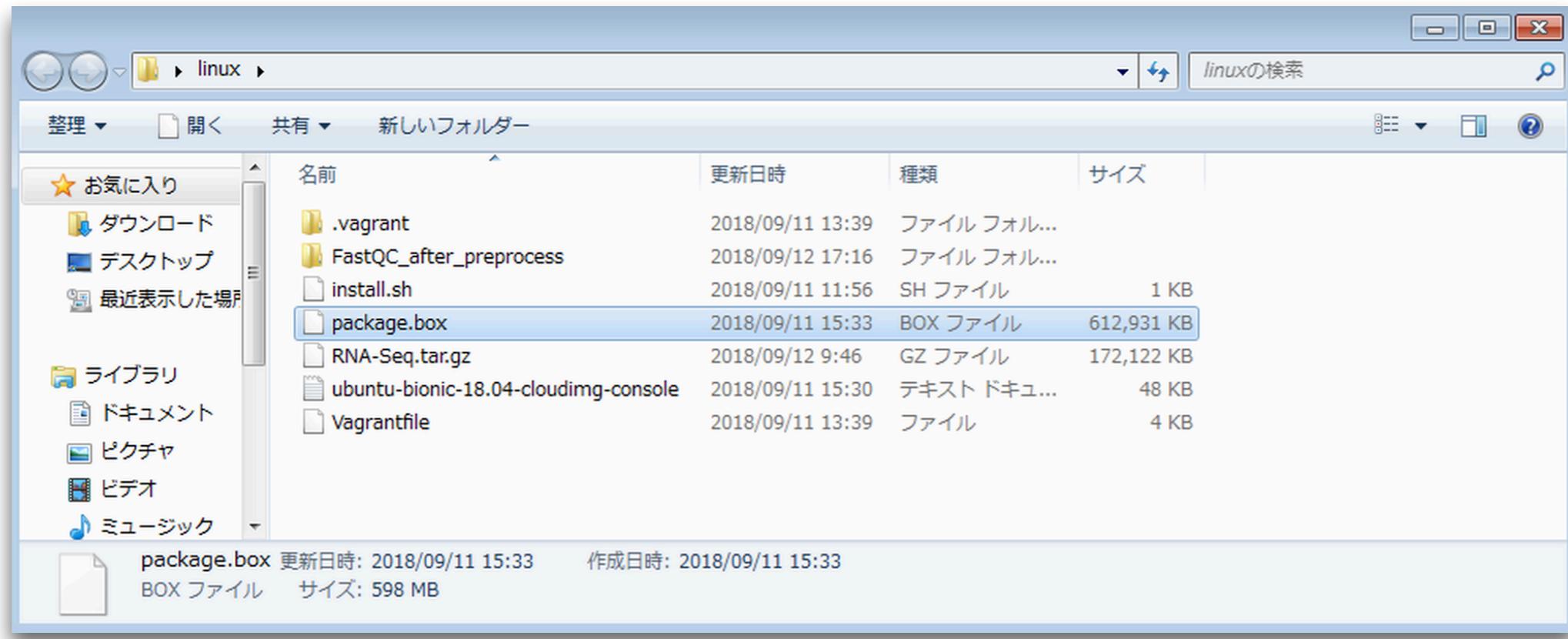
Vagrantfileがあるフォルダで実行する（今回の場合はデスクトップのlinuxフォルダ）

Boxファイル（package.box）が作成されていることを確認する



package.boxに仮想PCを復元するための全データが含まれている

# Boxファイルを別のPCにコピーする



package.boxを別のPCにコピーする

linuxフォルダのファイルはBoxファイルには含まれないので必要なら併せてコピーする

Vagrantfileもコピーして再利用することが出来るが一部書き換えが必要になる場合がある

# コピーしたBoxファイルを追加する

---

package.boxを別のPCのデスクトップにコピーしたとする（コピー先はどこでも良い）

ここからはコピー先での作業

PowerShell（またはターミナル）を起動してデスクトップに移動してから以下のコマンドでBoxファイルを追加する

```
% vagrant box add --name ubuntu package.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'ubuntu' (v0) for provider:
    box: Unpacking necessary files from: file:///Users/user02/Desktop/package.box
==> box: Successfully added box 'ubuntu' (v0) for 'virtualbox'!
```

**--name** オプションでBoxの名前を指定する（上のコマンドではubuntu）

Boxの名前は何でも良い

Boxを追加した後はpackage.boxは削除して良い

# 追加したBoxを使って仮想PCを作成する

---

設定ファイルを保存するフォルダとしてデスクトップにubuntuフォルダを作成し、そこに移動する

```
% cd C:\Users\user02\Desktop  
% mkdir ubuntu  
% cd ubuntu
```

設定ファイルを保存するフォルダはどこに作成しても良い

フォルダ名はubuntuでなくても良い

追加したBoxの名前（ubuntu）を指定してVagrantfileを作成する

```
% vagrant init ubuntu  
A `Vagrantfile` has been placed in this directory. You are now  
ready to `vagrant up` your first virtual environment! Please read  
the comments in the Vagrantfile as well as documentation on  
`vagrantup.com` for more information on using Vagrant.
```

仮想PCのメモリを4GBにした設定は消えているので必要なら同様にVagrantfileを修正する

## 【参考】 コピーしてきたVagrantfileを利用する場合

vagrant init コマンドを実行せずにこれまで使っていたVagrantfileをubuntuフォルダにコピーして再利用することができる

その場合はVagrantfileのBoxを指定している箇所を以下のように修正しなくてはならない

```
config.vm.box = "ubuntu/bionic64"
```



```
config.vm.box = "ubuntu"
```

**config.vm.box**で使用するBox名を指定する

最初に使用したBoxはubuntu/bionic64なので、そのように指定されている

今回使用するBoxはubuntuなので、Boxの指定をubuntuに変更する必要がある

# 追加したBoxを使って仮想PCを作成する

---

仮想PCの作成とLinuxのインストール・起動を行う

```
% vagrant up  
(進捗状況が表示される)
```

ログインする

```
% vagrant ssh  
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-33-generic x86_64)  
(途中のメッセージ省略)  
vagrant@ubuntu-bionic:~$
```

全く同じ仮想PCが起動した！

## その他の便利な機能

スナップショットや仮想PCの削除

# 現在の状態を保存するスナップショット

---

現在の仮想PCの状態を保存して後から復元できるようにする機能

仮想PCを停止する

```
% vagrant halt  
==> default: Attempting graceful shutdown of VM...
```

スナップショットを作成する

```
% vagrant snapshot save s1  
==> default: Snapshotting the machine as 's1'...  
==> default: Snapshot saved! You can restore the snapshot at any time by  
==> default: using `vagrant snapshot restore`. You can delete it using  
==> default: `vagrant snapshot delete`.
```

s1という名前のスナップショットを作成する（名前は何でもよい）

スナップショットを確認する

```
% vagrant snapshot list  
s1
```

## 仮想PCを起動して何か操作をする

```
% vagrant up  
(進捗状況が表示される)  
% vagrant ssh  
(途中のメッセージ省略)  
vagrant@ubuntu-bionic:~$ ls  
RNA-Seq  
vagrant@ubuntu-bionic:~$ rm -rf RNA-Seq ← 解析結果を丸ごと削除してみる  
vagrant@ubuntu-bionic:~$ ls ← 削除されたので何も表示されない  
vagrant@ubuntu-bionic:~$ exit
```

## スナップショットから復元する

```
% vagrant snapshot restore s1  
==> default: Forcing shutdown of VM...  
==> default: Restoring the snapshot 's1'...  
(途中のメッセージ省略)
```

## ログインして確認する

```
% vagrant ssh  
(途中のメッセージ省略)  
vagrant@ubuntu-bionic:~$ ls  
RNA-Seq ← 削除した解析結果が復活している！
```

# 仮想PCとBoxを削除したい場合

---

仮想PCを削除する

```
% vagrant destroy  
  default: Are you sure you want to destroy the 'default' VM? [y/N] y  
==> default: Destroying VM and associated drives...
```

yを入力してEnterを押す



Vagrantfileがあるフォルダで実行する（今回の場合はデスクトップのlinuxフォルダ）

Boxを削除する

```
% vagrant box remove ubuntu/bionic64  
Removing box 'ubuntu/bionic64' (v20180907.0.0) with provider 'virtualbox'...
```

# ハードディスクの拡張

仮想PCのハードディスクの容量を増やす

実習で作成した仮想PCのハードディスクは10GB



NGSのデータを扱っていると10GBを使い切ってしまう



仮想PCのハードディスクを100GBに拡張する

この後の作業は仮想PCを停止させた状態で行う  
(vagrant haltを実行しておく)

# ハードディスクを操作するツールをダウンロードする

GPartedのWebサイトからGPartedの起動ディスクイメージをダウンロードする  
(2018年9月26日現在の最新版はgparted-live-0.32.0-1-amd64.iso)

GParted Live CD/USB/HD/PXE Bootable Image			
<b>Stable Releases</b>			
<a href="#">Download gparted-live-0.32.0-1-i686.iso</a>	<a href="#">Release notes</a>	<a href="#">Changelog</a>	<a href="#">Checksums</a> <a href="#">Checksums.gpg</a> (Signed by <a href="#">GParted live maintainer Steven Shiau</a> , updated on <a href="#">Sep/18/2017</a> ).
<a href="#">Download gparted-live-0.32.0-1-amd64.iso</a>			
<a href="#">Stable directory (.iso/.zip)</a> (for i686, i686-pae and amd64 architectures)			
<b>Testing Releases</b>			
<a href="#">Testing directory (.iso/.zip)</a> (for i686, i686-pae and amd64 architectures)	Release notes and checksums in same directory.		

<https://gparted.sourceforge.io/download.php>

仮想PCのデータが保存されているフォルダに移動する

Windowsの場合

```
% cd '~\VirtualBox\ VMs\vagrant_default_1537939156546_34206'
```

Macの場合

```
% cd ~/VirtualBox\ VMs/vagrant_default_1537939156546_34206
```

上の例では**vagrant\_default\_1537939156546\_34206**フォルダに移動しているが  
フォルダ名の数字の部分はインストールした状況によって変わる

その中にある**ubuntu-bionic-18.04-cloudimg.vmdk**ファイルがハードディスクの本体

# ハードディスクを拡張する

ハードディスク（の内容が保存されているファイル）をコピーする

```
% VBoxManage clonemedium disk ubuntu-bionic-18.04-cloudimg.vmdk  
ubuntu-bionic-18.04-cloudimg.vdi --format VDI
```

表示の都合上改行しているが改行せずに入力する

コピーしたハードディスクを100GBに拡張する

```
% VBoxManage modifymedium disk ubuntu-bionic-18.04-cloudimg.vdi  
--resize 102400
```

表示の都合上改行しているが改行せずに入力する

拡張前のハードディスクをVirtualBoxの管理下から除外する

```
% VBoxManage closemedium disk ubuntu-bionic-18.04-cloudimg.vmdk
```

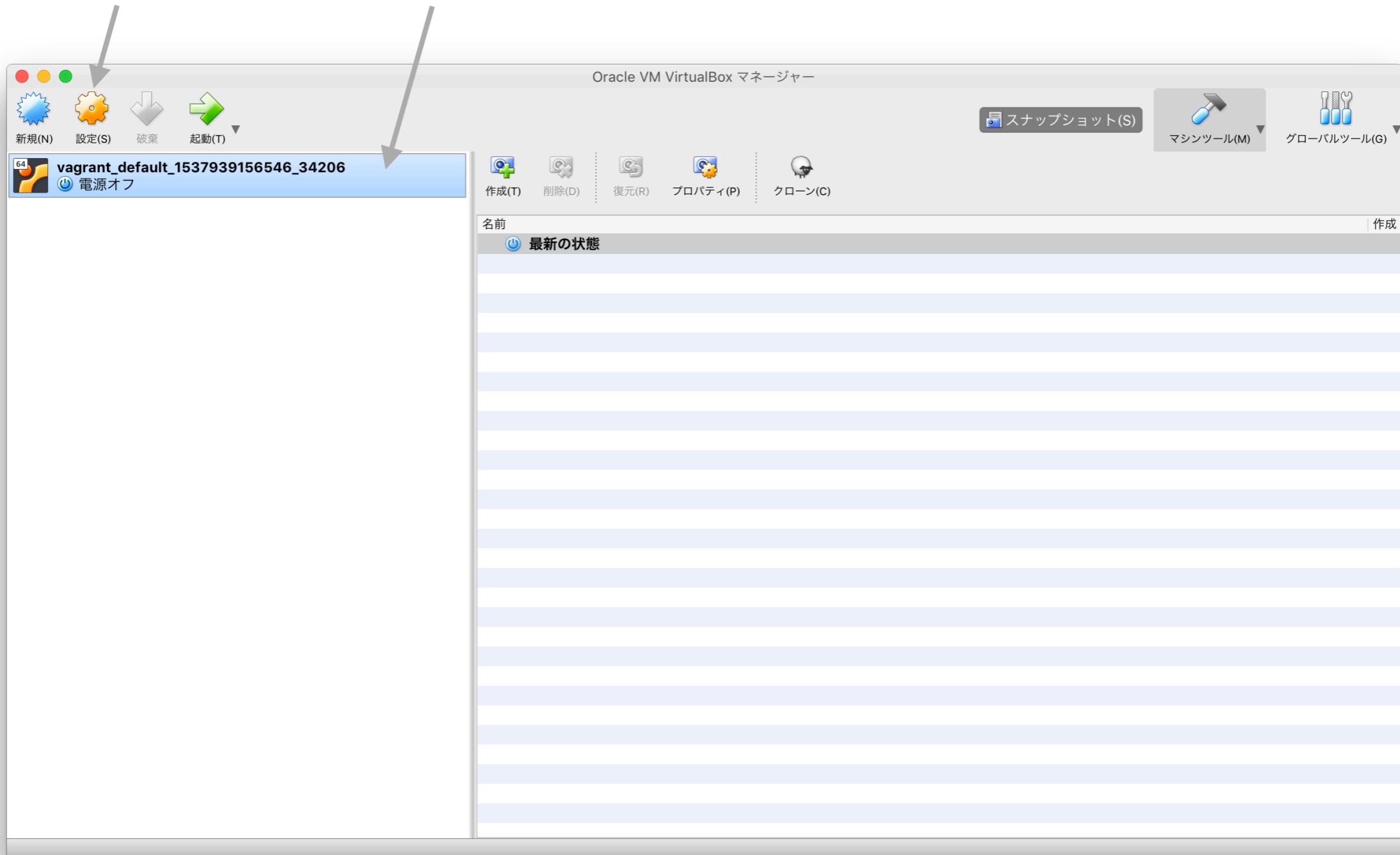
拡張前のハードディスク（の内容が保存されているファイル）を削除する

```
% rm ubuntu-bionic-18.04-cloudimg.vmdk
```

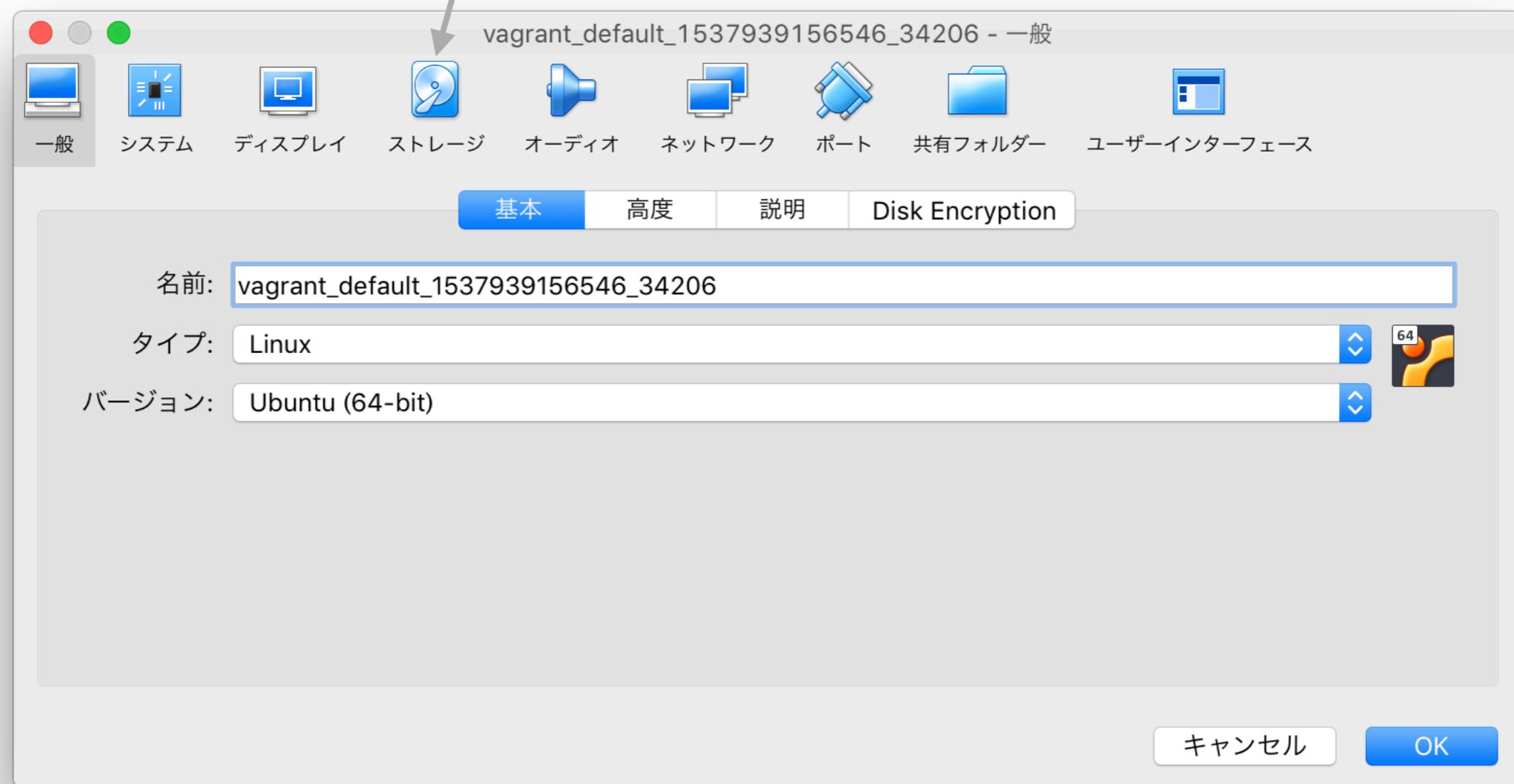
# VirtualBoxを起動する

設定を押す

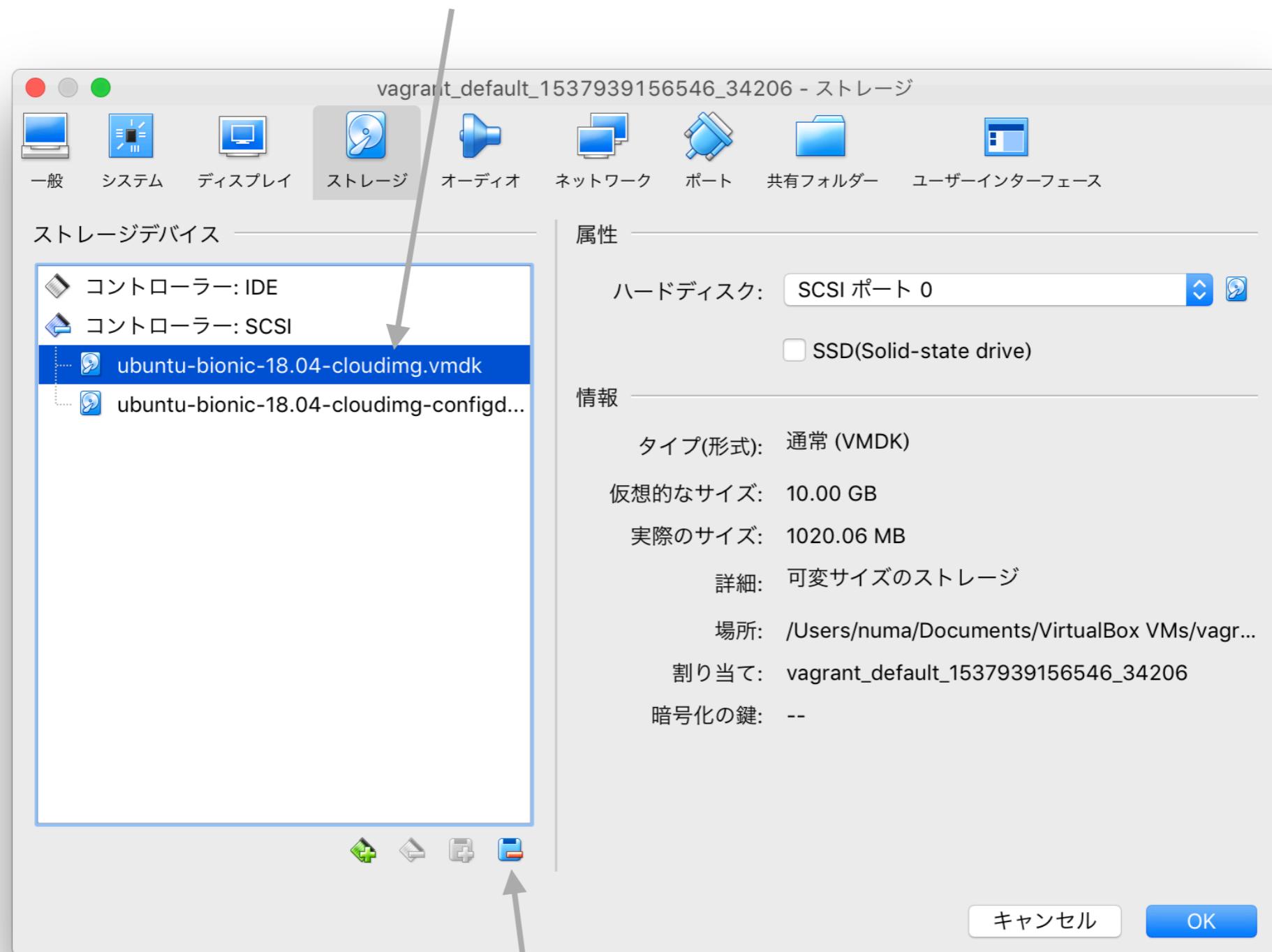
仮想PCが選択されていることを確認する



ストレージを押す

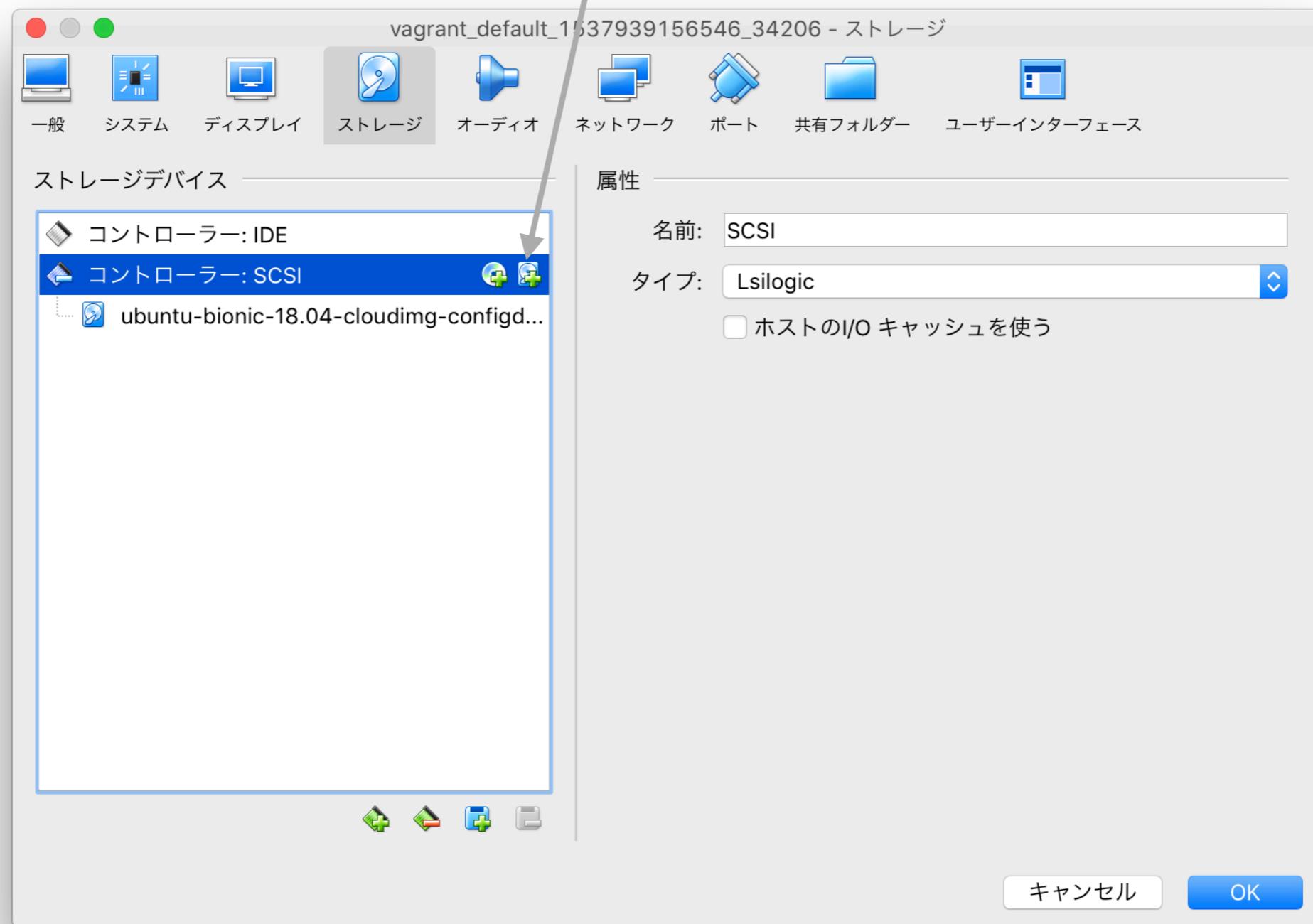


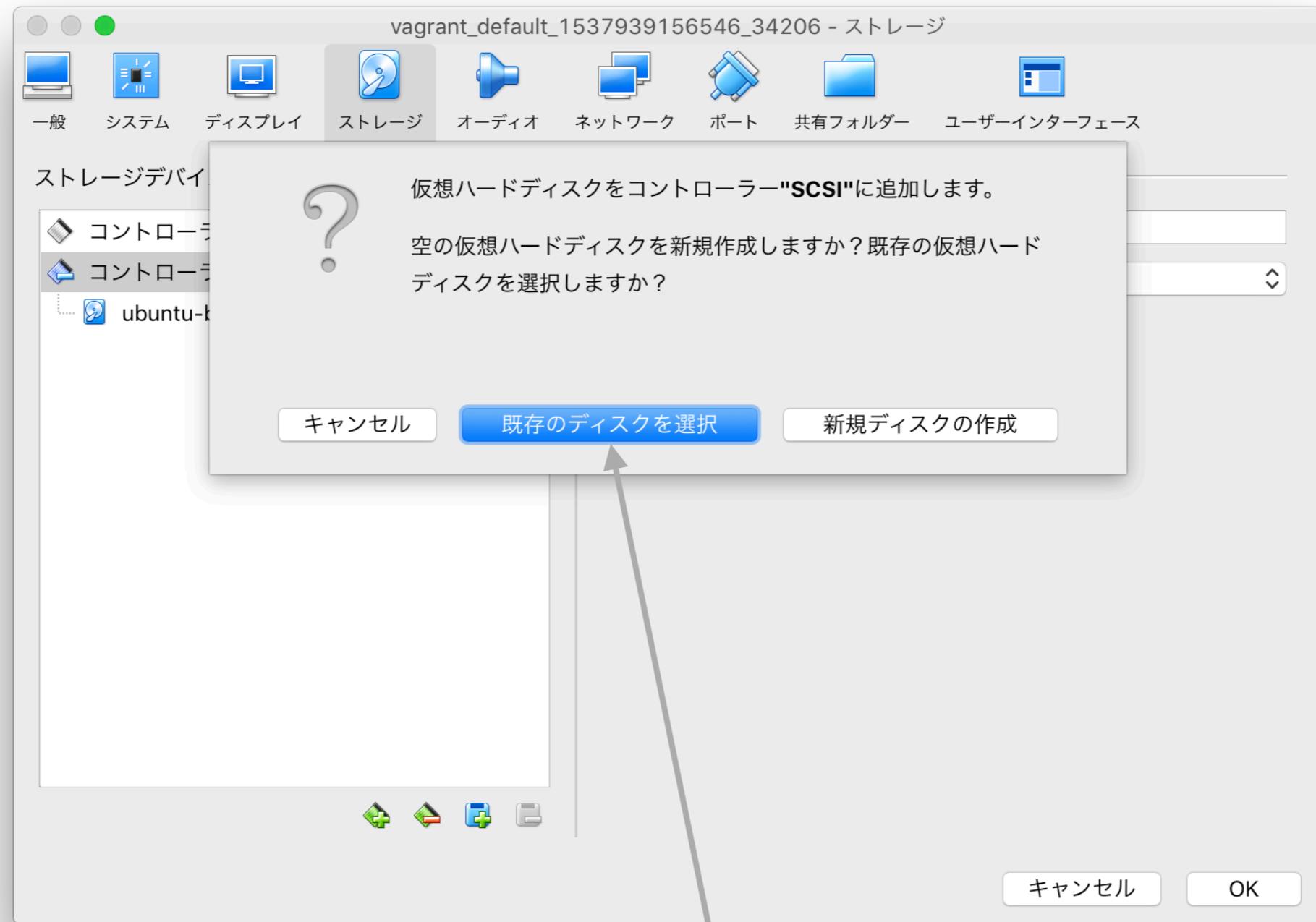
## ubuntu-bionic-18.04-cloudimg.vmdkを選択する



削除アイコンを押す

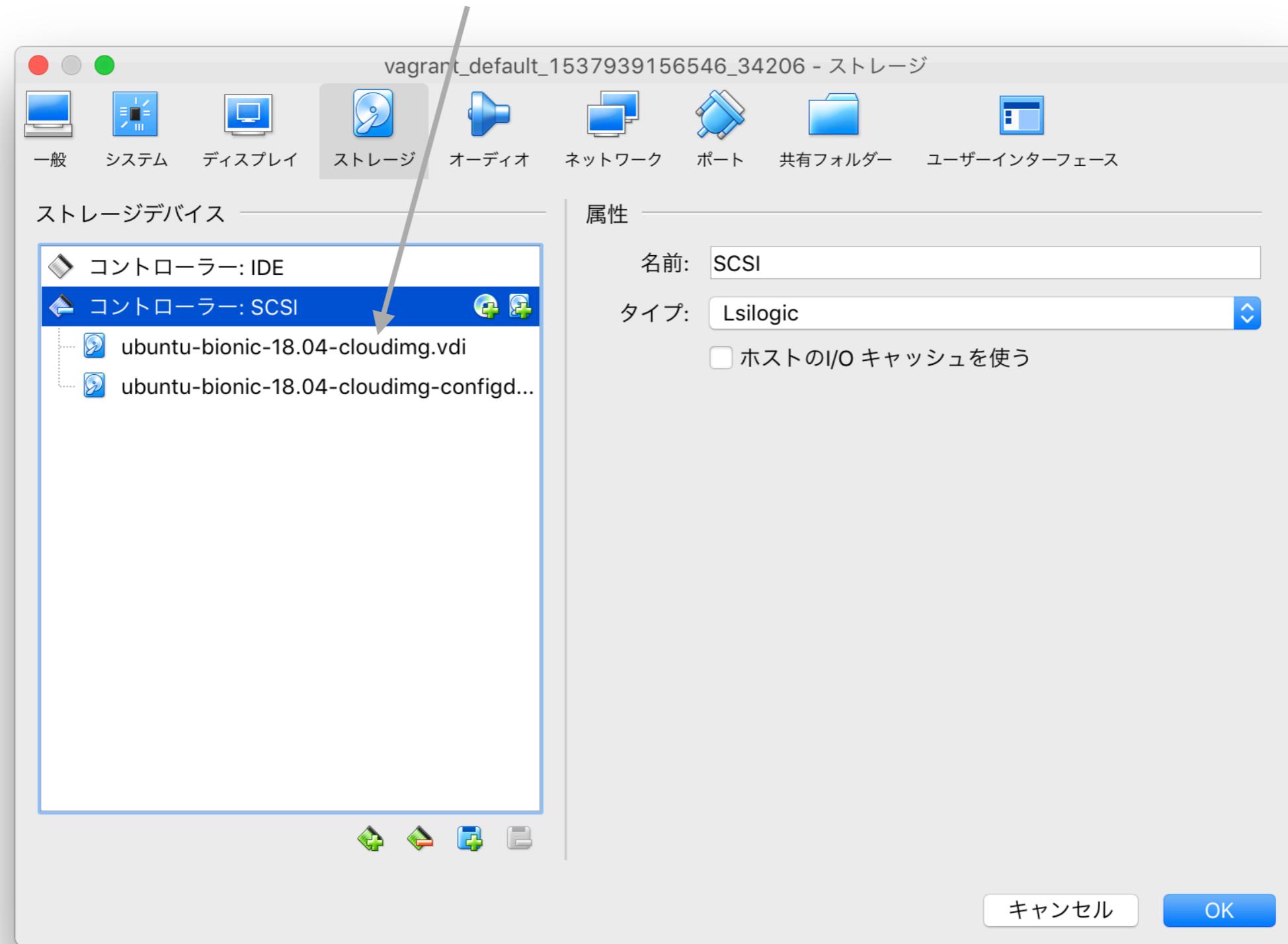
ハードディスク追加アイコンを押す



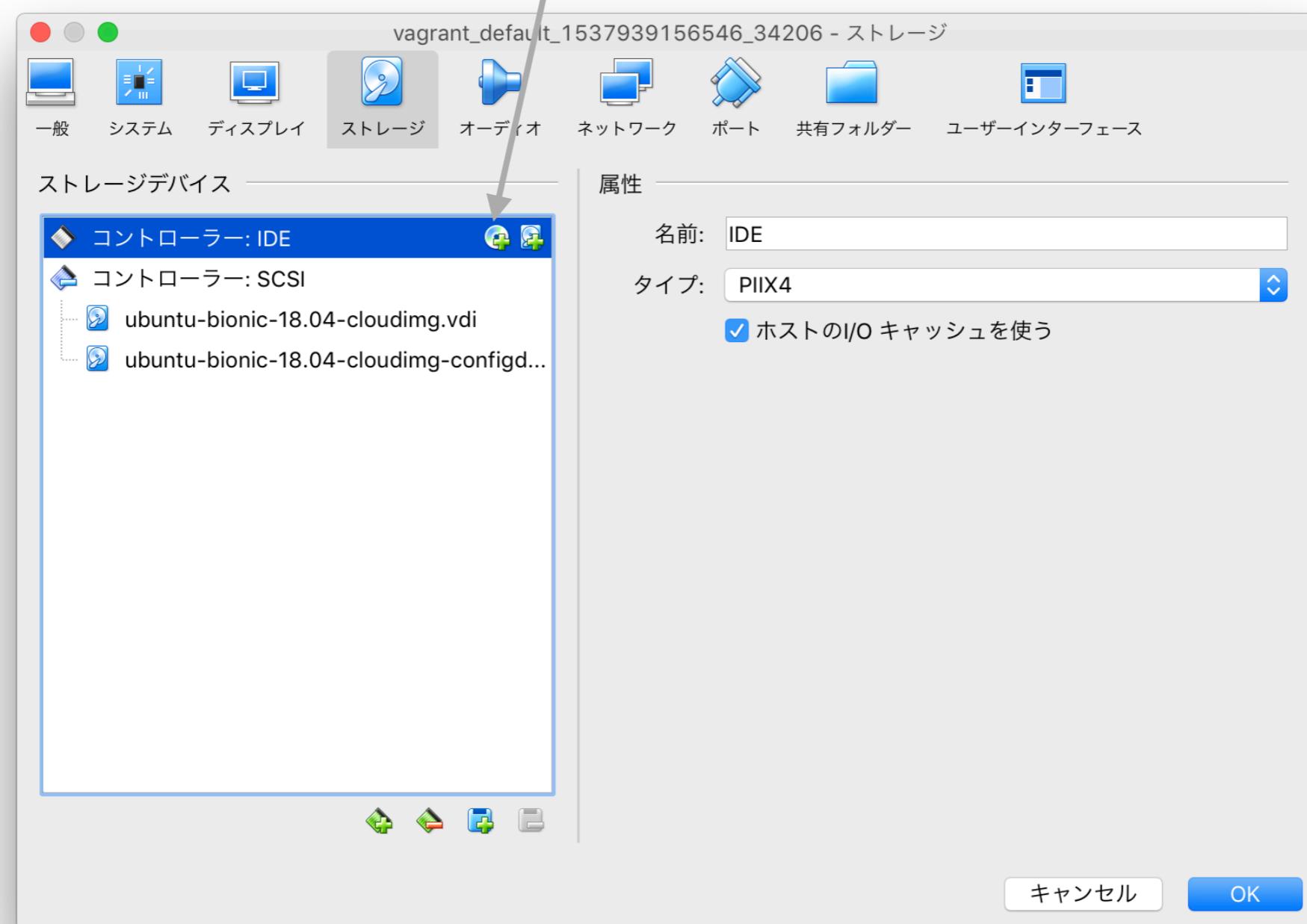


既存のディスクを選択を押してubuntu-bionic-18.04-cloudimg.vdiを選択する

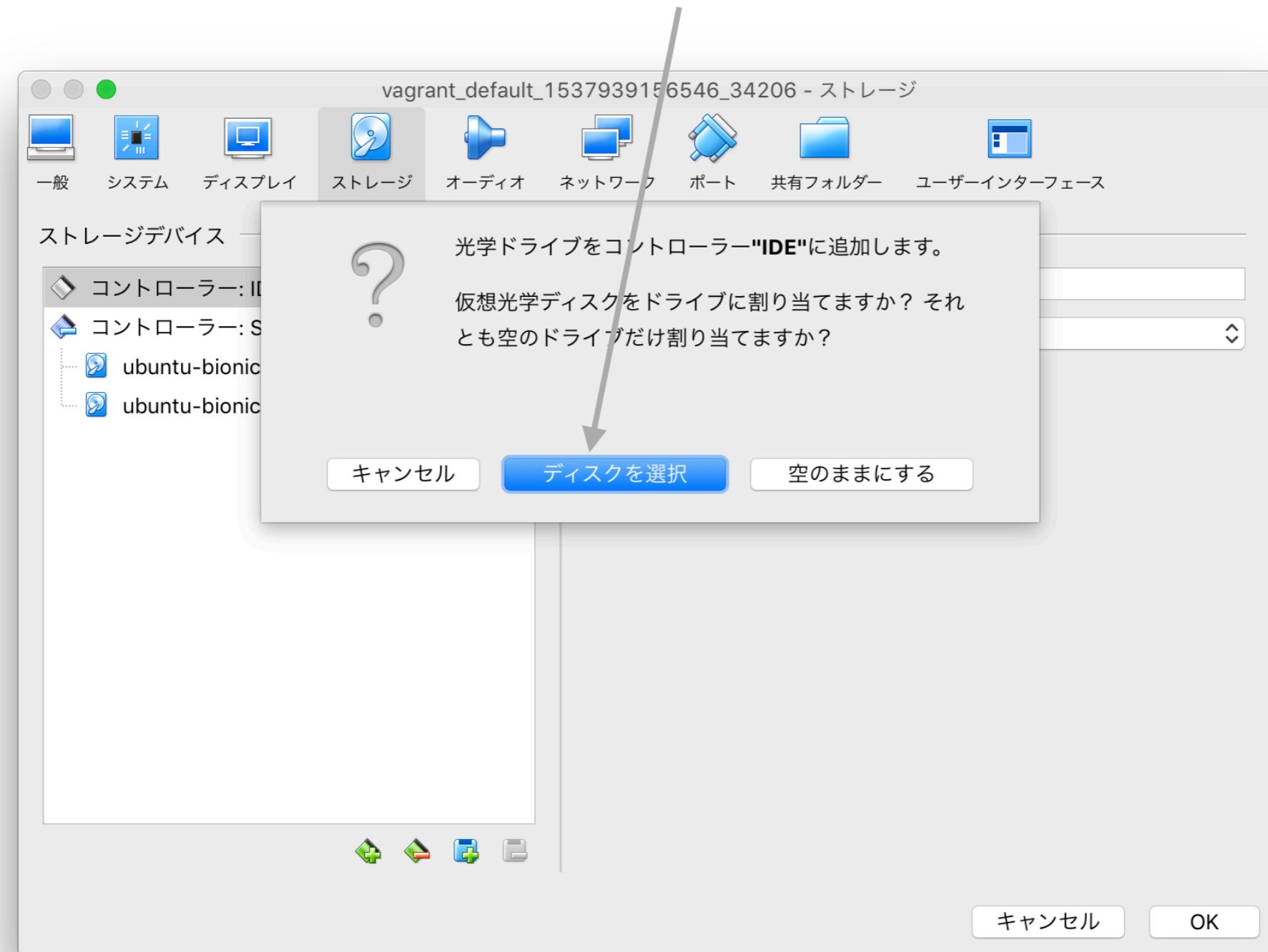
## 拡張したハードディスクが追加される



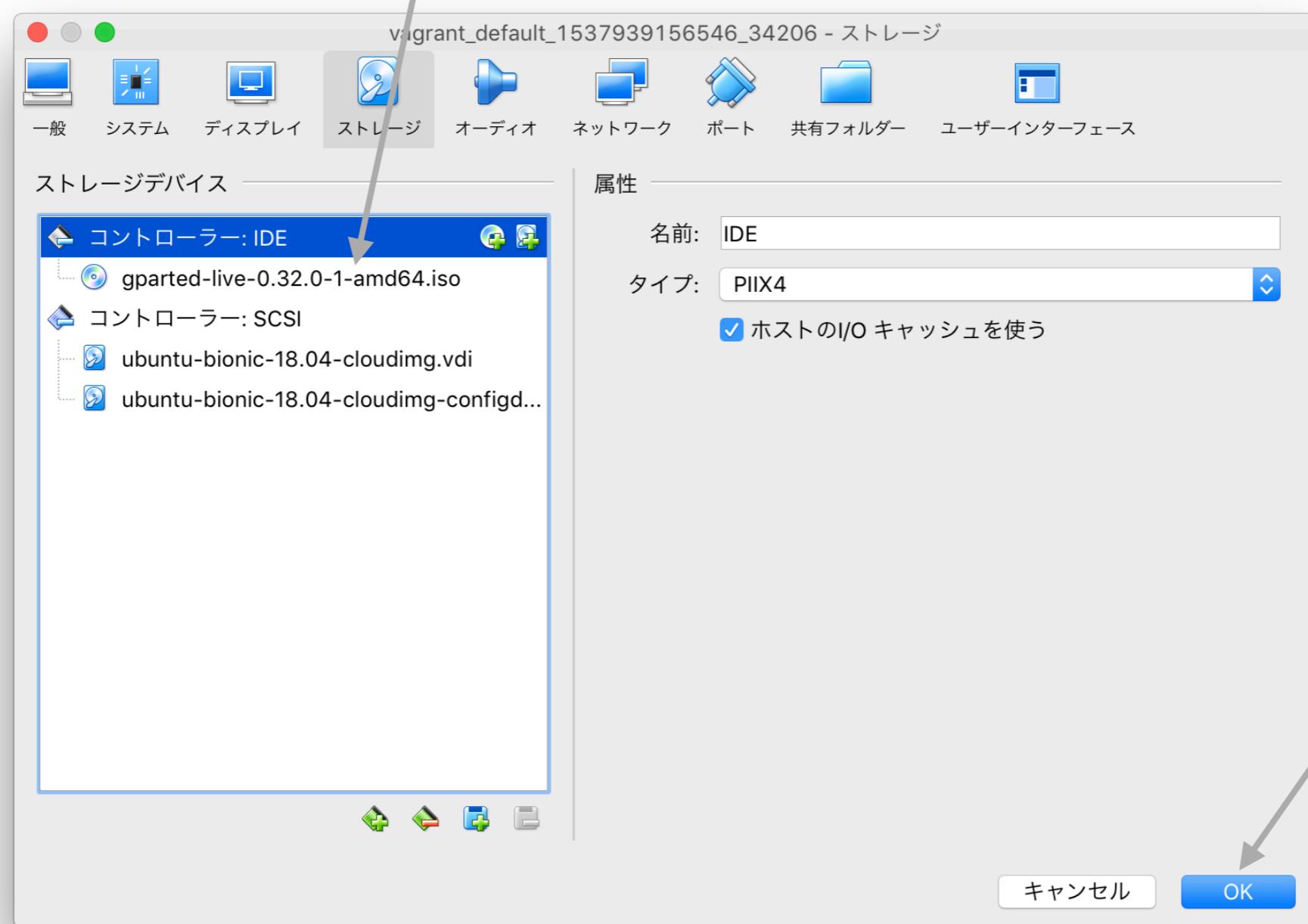
CDアイコンを押す



ディスクを選択を押してダウンロードしたgparted-live-0.32.0-1-amd64.isoを選択する

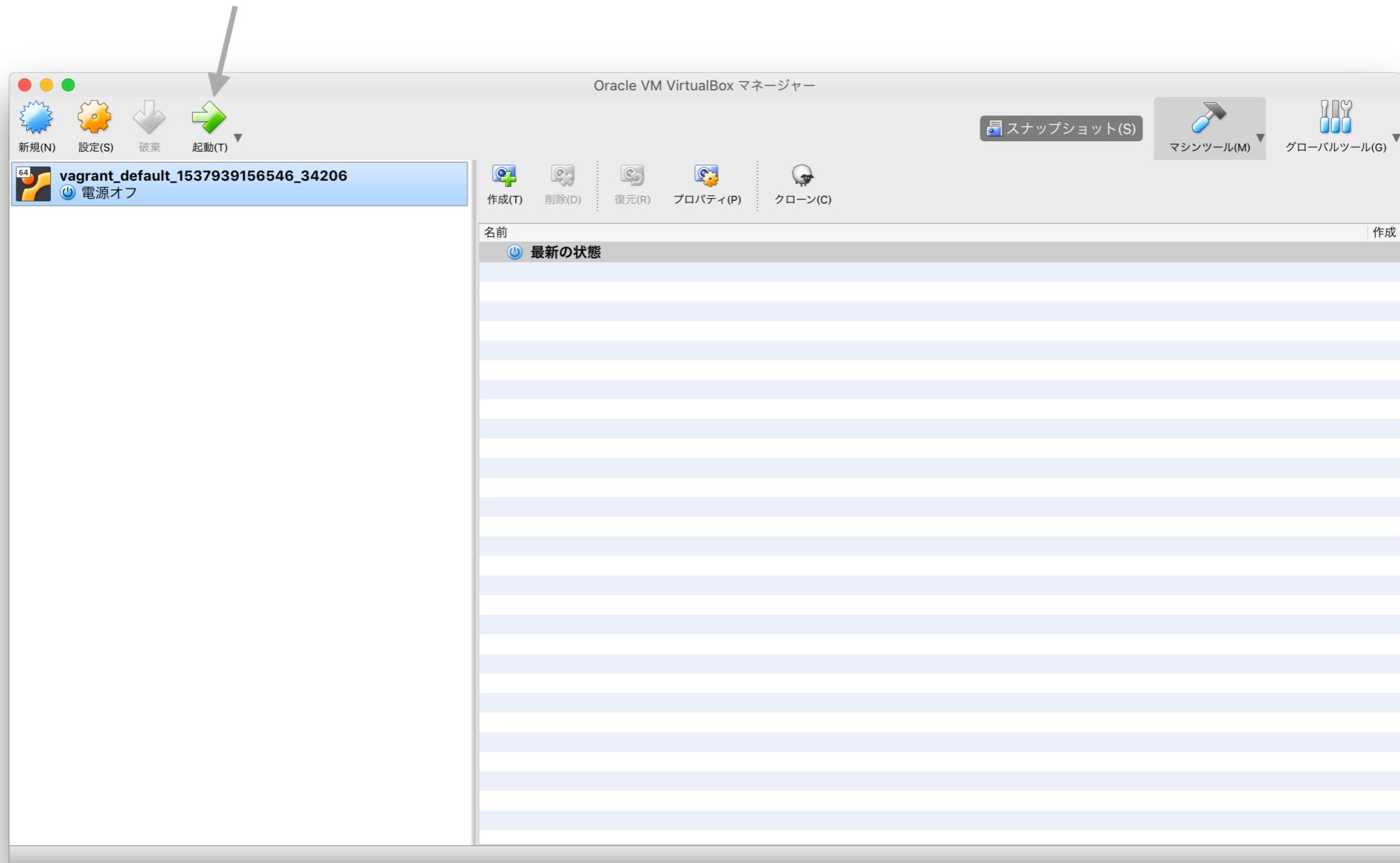


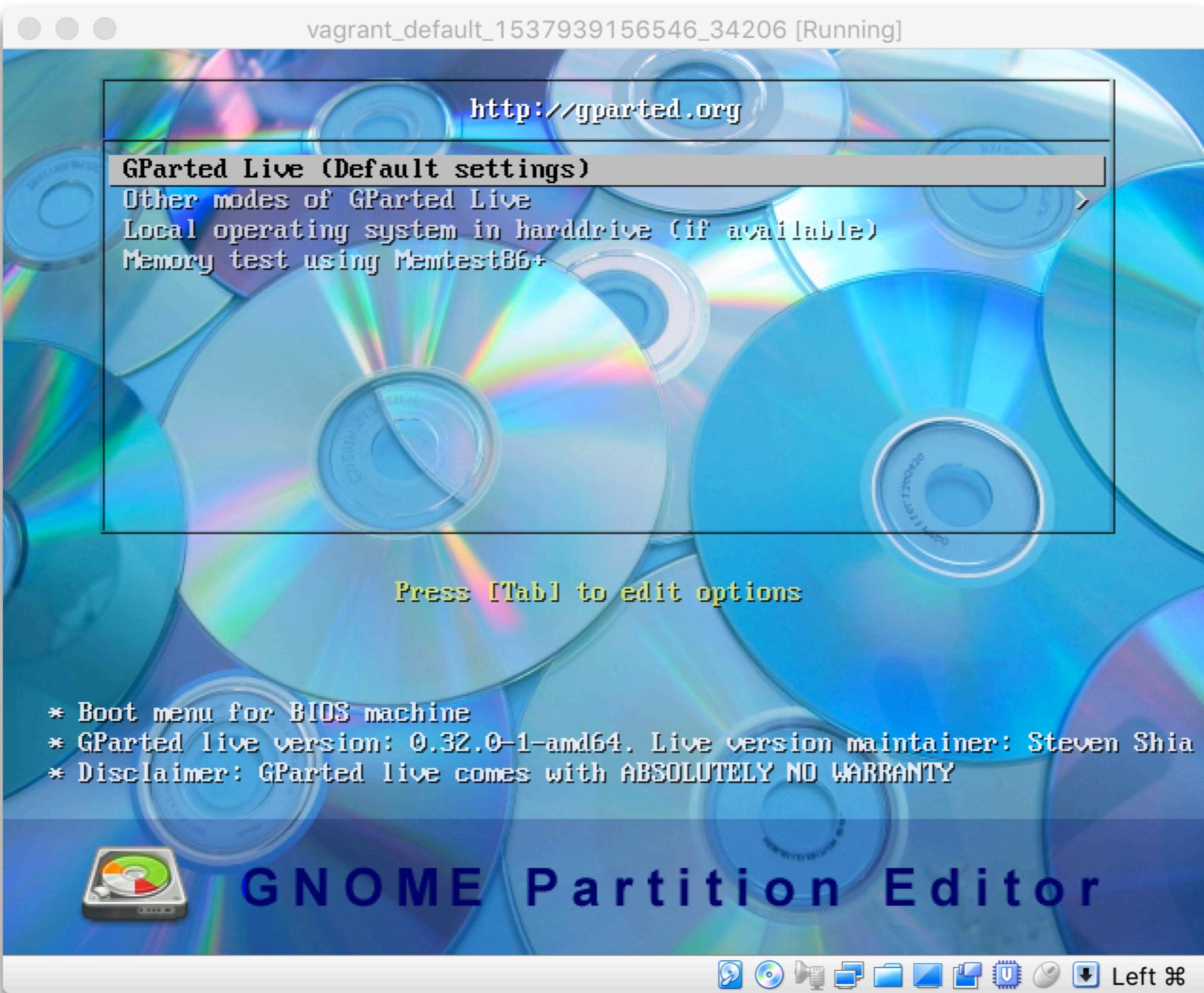
ハードディスク操作ツールのCDが挿入された



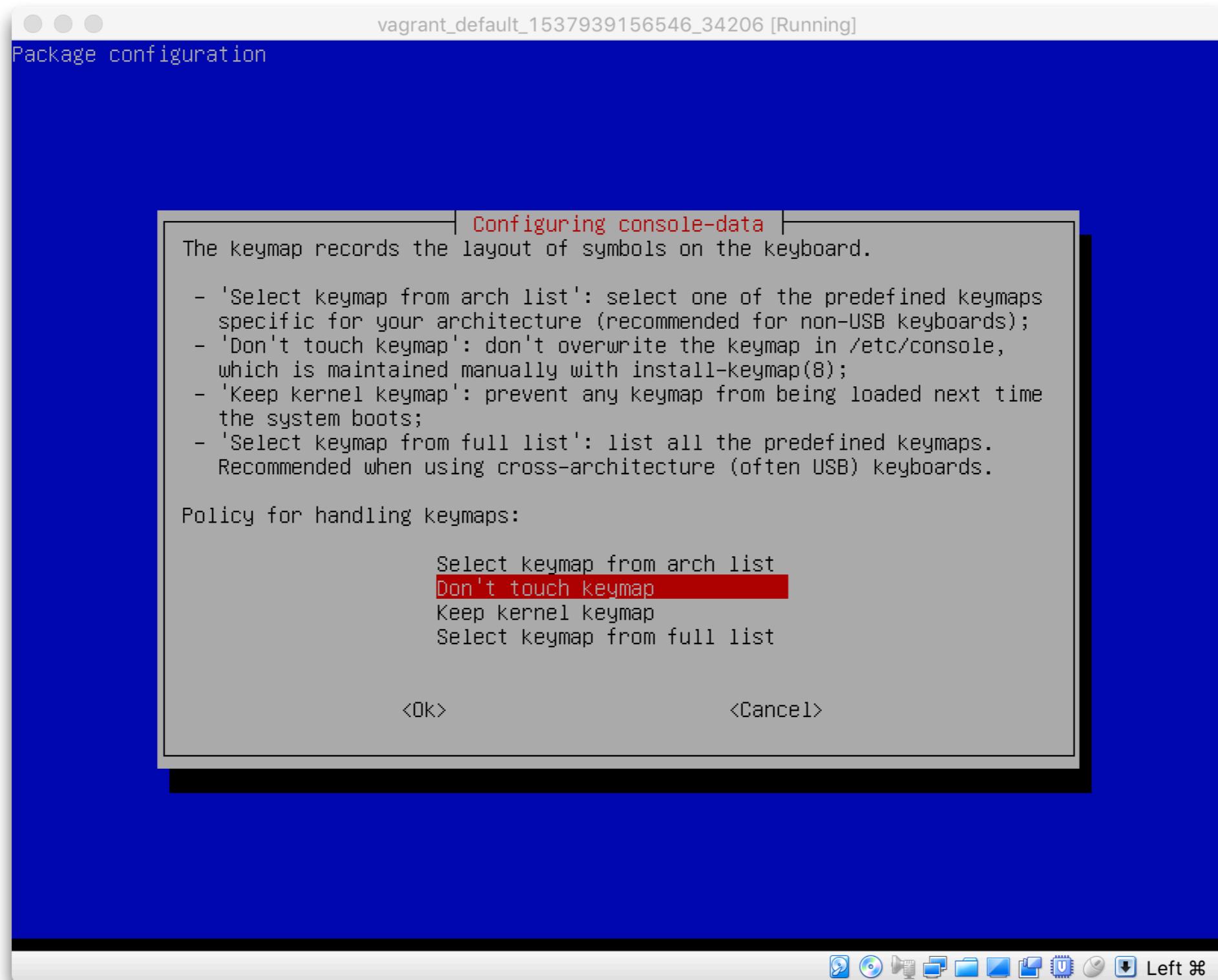
OKを押す

起動する（ハードディスク操作ツールが起動する）

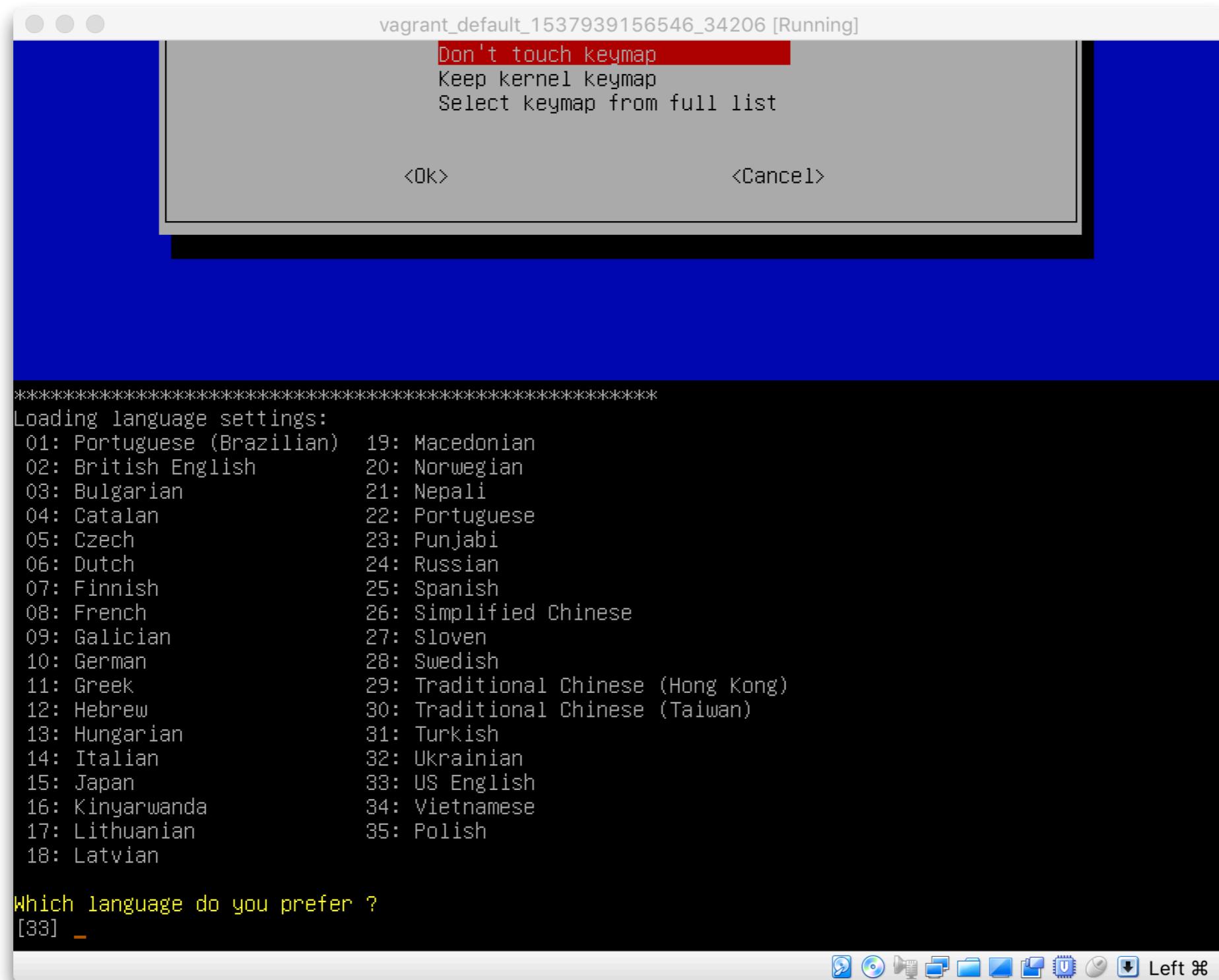




GParted Live (Default settings)が選択されていることを確認してEnterを押す



Enterを押す



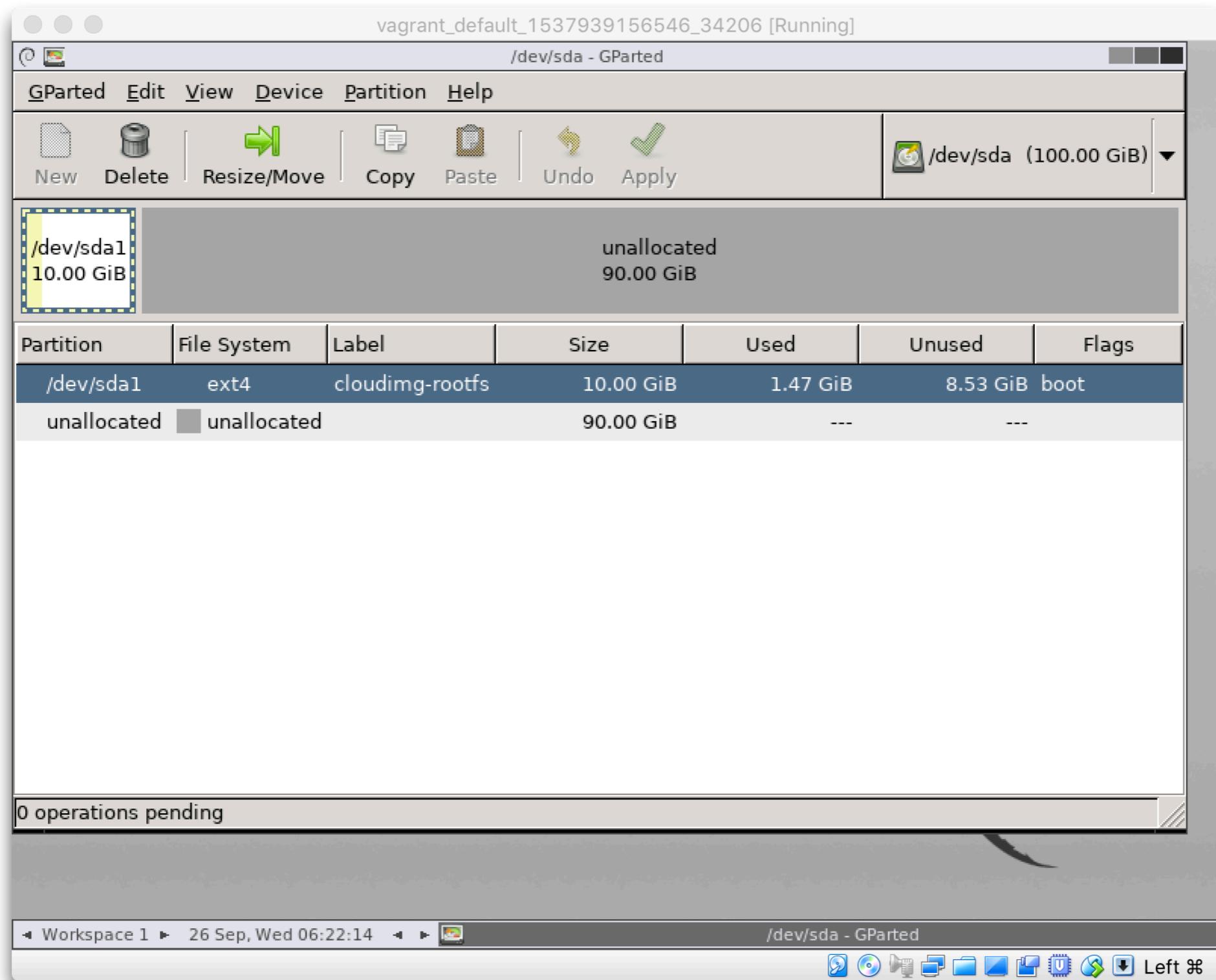
Enterを押す

```
vagrant_default_1537939156546_34206 [Running]

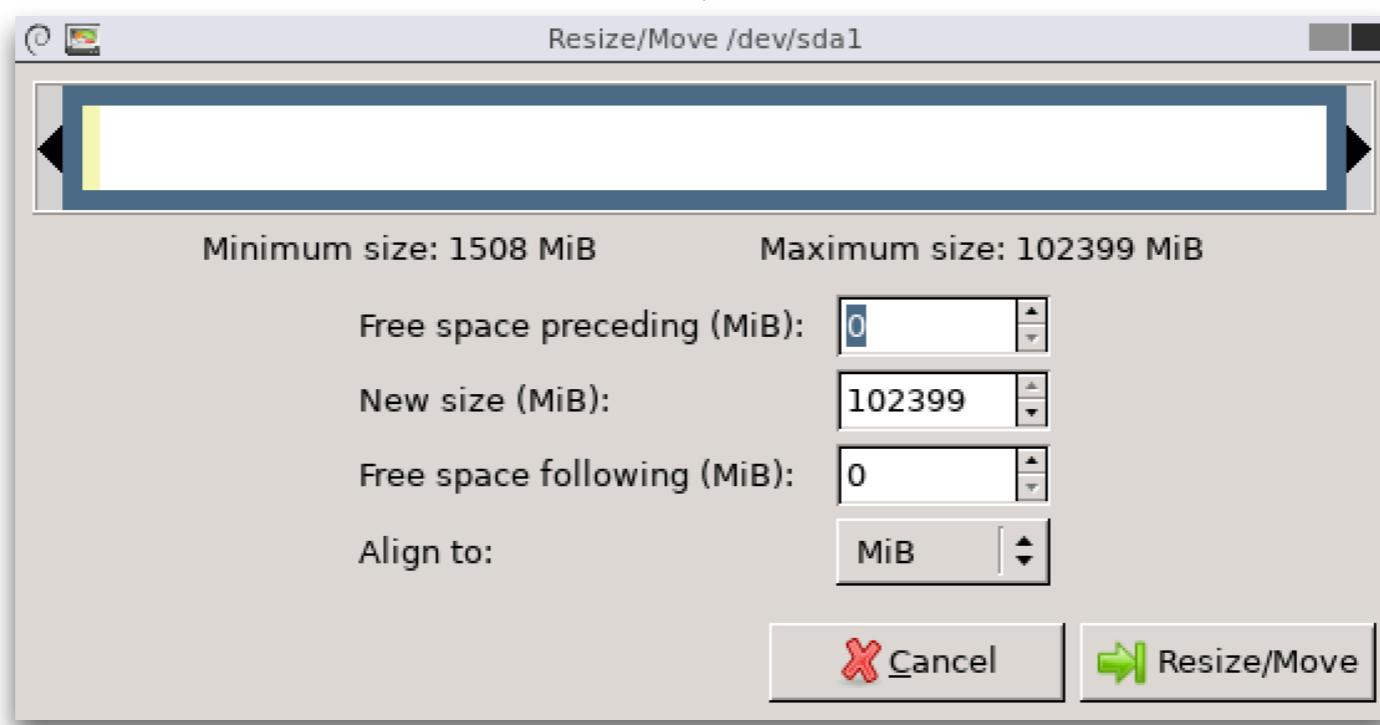
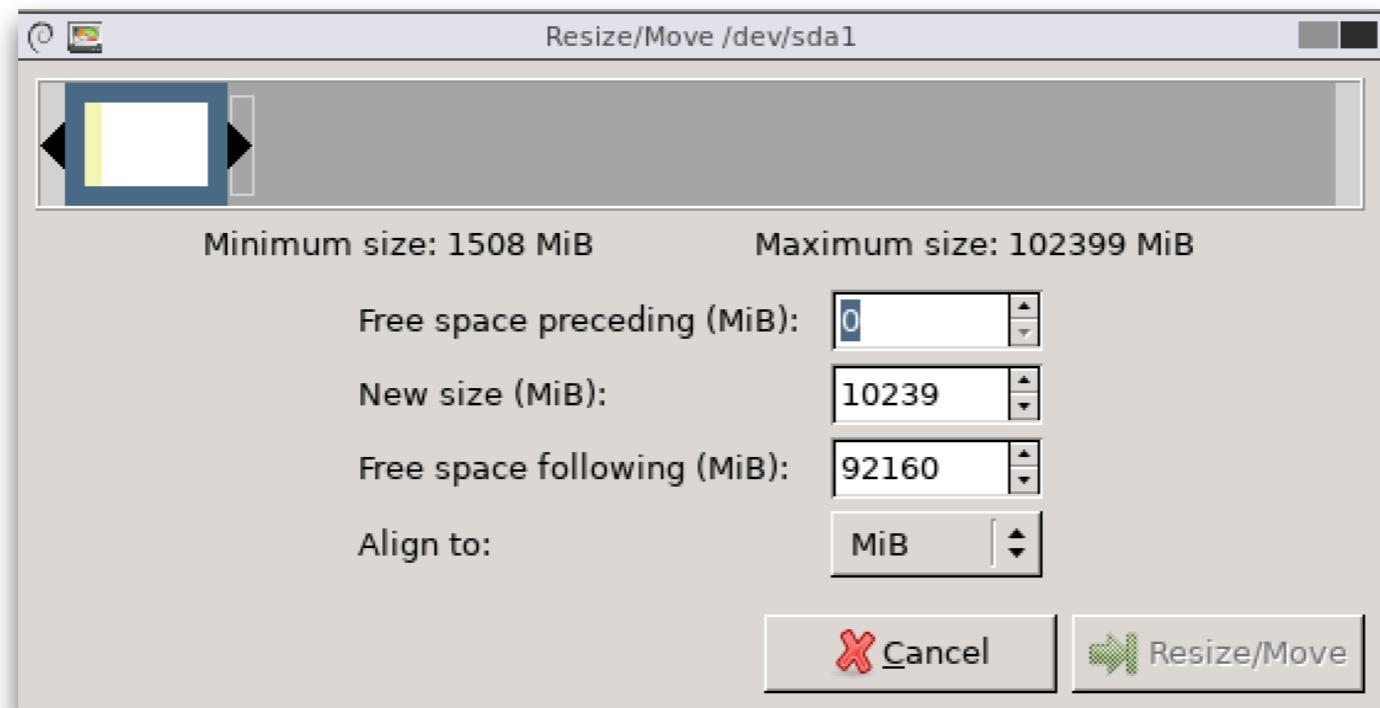
03: Bulgarian          21: Nepali
04: Catalan            22: Portuguese
05: Czech              23: Punjabi
06: Dutch              24: Russian
07: Finnish            25: Spanish
08: French             26: Simplified Chinese
09: Galician           27: Sloven
10: German              28: Swedish
11: Greek              29: Traditional Chinese (Hong Kong)
12: Hebrew              30: Traditional Chinese (Taiwan)
13: Hungarian          31: Turkish
14: Italian             32: Ukrainian
15: Japan               33: US English
16: Kinyarwanda         34: Vietnamese
17: Lithuanian          35: Polish
18: Latvian

Which language do you prefer ?
[33]
The default language US English is used
Language selected en_US
Setting locale in /etc/default/locale...
done!
*****
//NOTE/// Later we will enter graphical environment if you choose '0'. However, if graphical environment (X-window) fails to start, you can:
Run "sudo Forcevideo" to configure it again. Choose 1024x768, 800x600 or 640x480 as your resolution and the driver for your VGA card, etc. Most of the time you can accept the default values if you have no idea about them.
If failing to enter graphical environment, and it does not return to text mode, you can reboot again, and choose '1' here to config X manually.
-----
Which mode do you prefer ?
(0) Continue to start X to use GParted automatically
(1) Run 'Forcevideo' to config X manually
(2) Enter command line prompt
[0] -
```

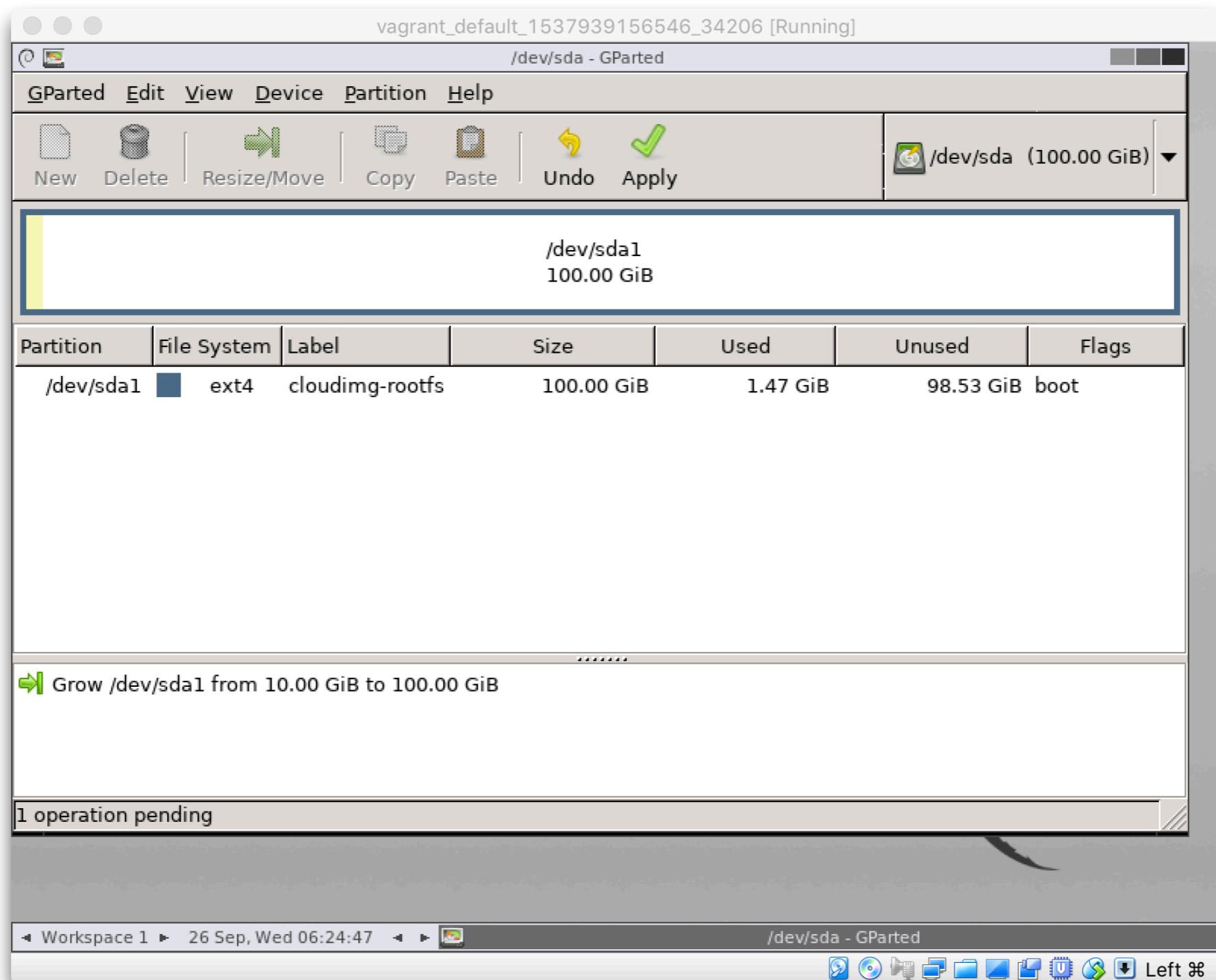
# Enterを押す



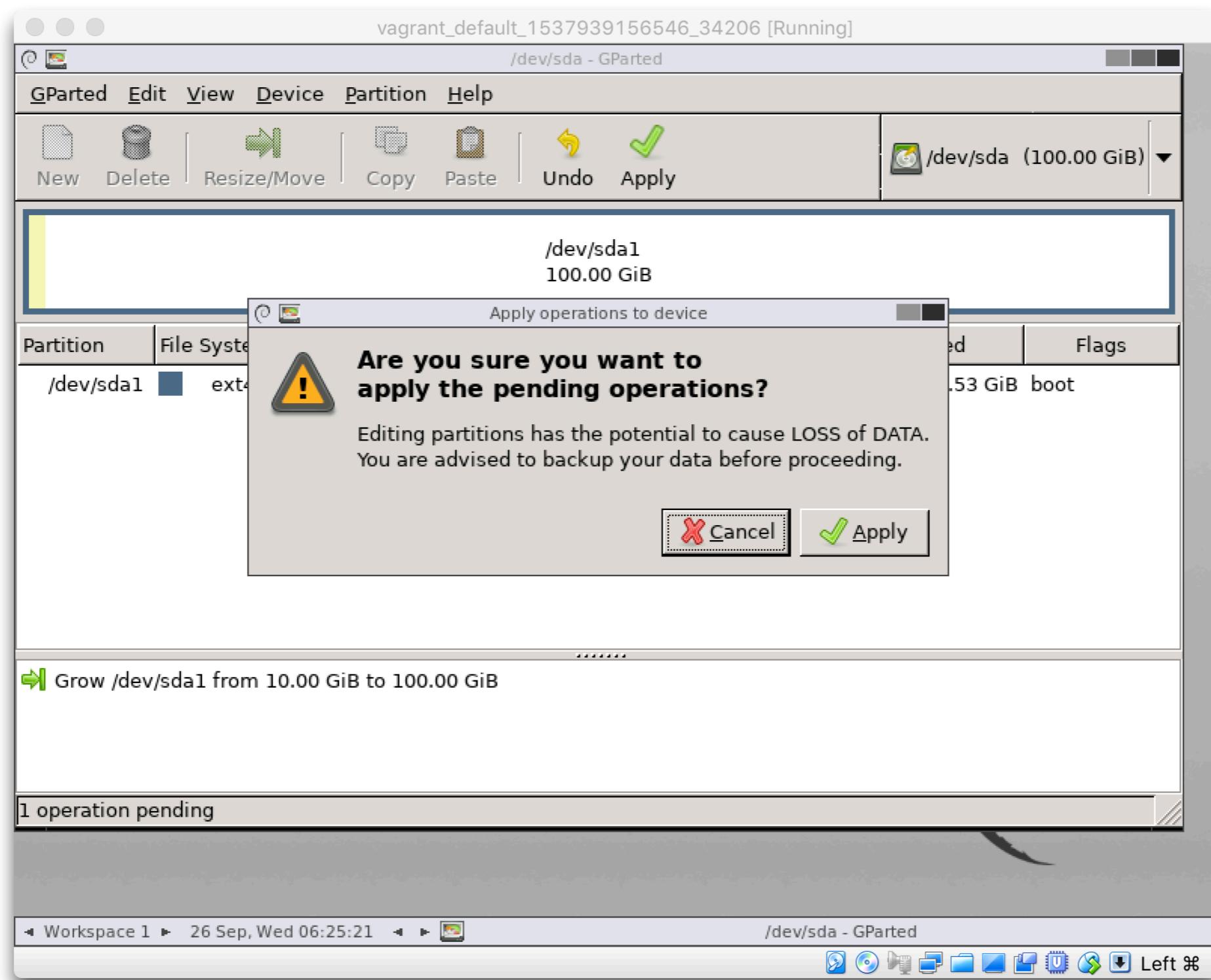
/dev/sda1が選択されていることを確認してからResize/Moveを押す



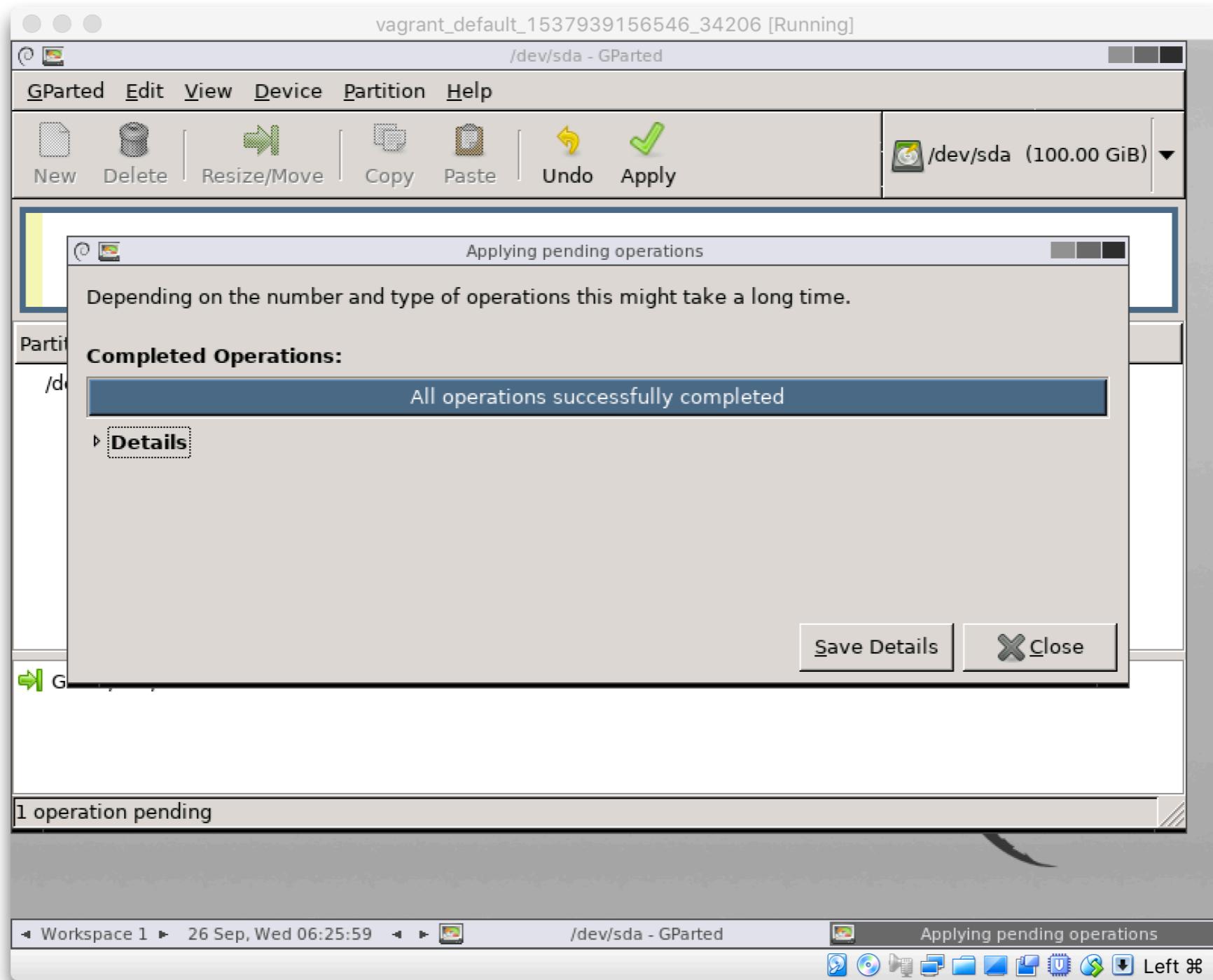
右矢印をドラッグして右端まで広げてからResize/Moveを押す



Applyを押す

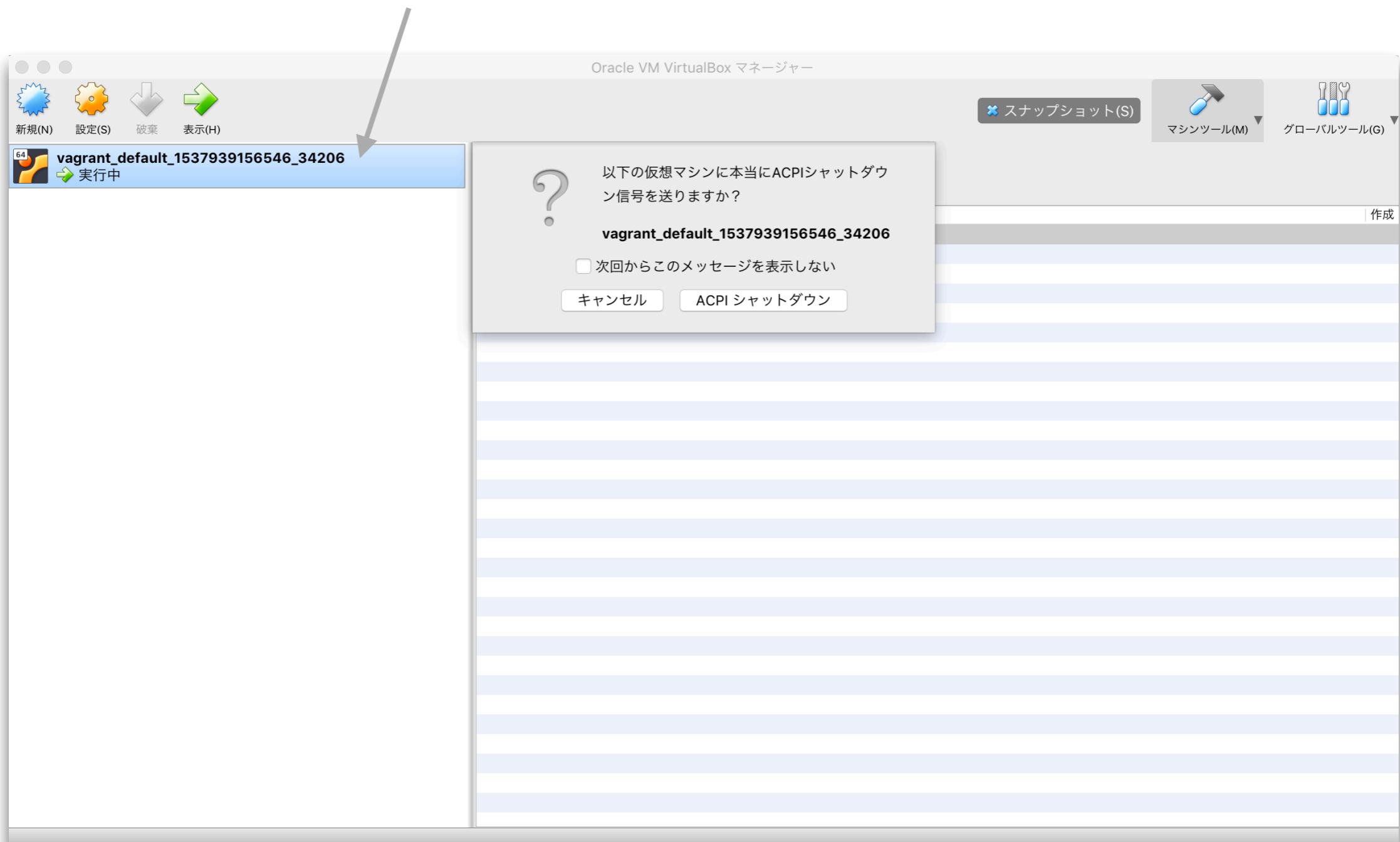


Applyを押す

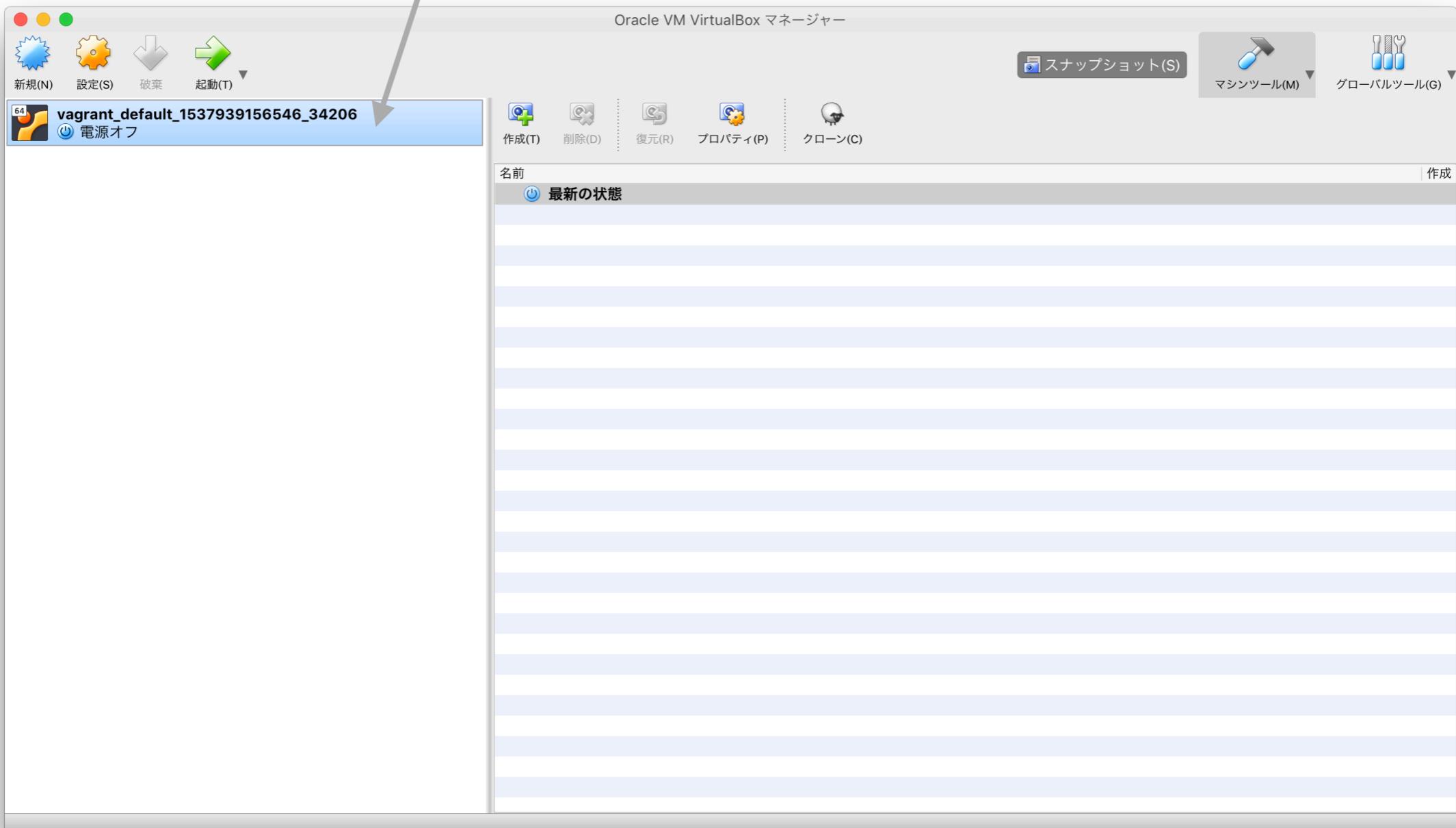


Closeを押す

右クリックして閉じる→ACPIシャットダウンを選択する



仮想PCが停止する



仮想PCが停止した後にVirtualBoxを終了する

# Linuxにログインして確認する

```
% cd C:\Users\Administrator\Desktop\linux
```

```
% vagrant up
```

(メッセージ省略)

```
% vagrant ssh
```

(メッセージ省略)

```
vagrant@ubuntu-bionic:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
------------	------	------	-------	------	------------

(途中省略)

/dev/sda1	97G	966M	96G	1%	/
-----------	-----	------	-----	----	---



10GBから約100GBに拡張された！