



Vagrant — это ПО с открытым исходным кодом, предназначенное для создания и поддержки переносимых виртуальных программных сред, совместимое с VirtualBox, Hyper-V, VMware, Docker и AWS.



Microsoft
Hyper-V



vmware



docker



amazon
webservices™

Почти все взаимодействие с Vagrant осуществляется через интерфейс командной строки с помощью команды **vagrant**. Эта команда имеет много подкоманд, таких как **vagrant up**, **vagrant destroy** и т. п.

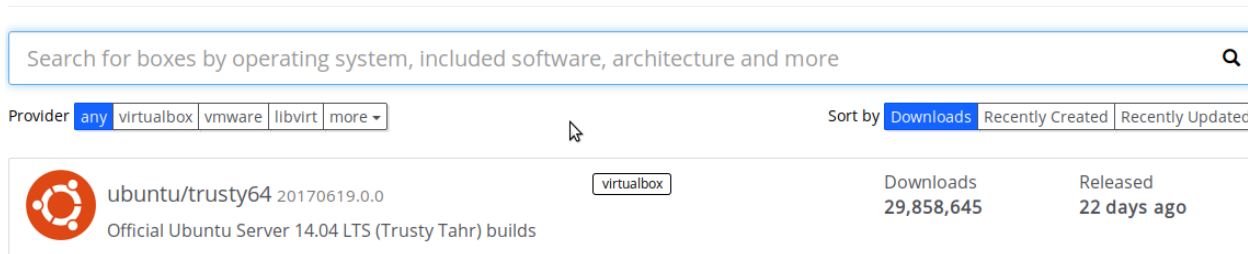
Для получения справки можно использовать ключ **-h**, например: **vagrant box -h**

Бокс — это формат пакета для окружений Vagrant, который может быть использован в любой среде, которую поддерживает Vagrant, для создания идентичного рабочего окружения. Представляет из себя архив (tar, tar.gz, zip) с расширением .box, который содержит образ виртуальной машины (VM), шаблон Vagrantfile, а также может содержать метаданные в формате json.

Простейший путь — использование готового бокса из публичного каталога Vagrant:

<https://app.vagrantup.com/boxes/search>

Discover Vagrant Boxes



The screenshot shows the 'Discover Vagrant Boxes' search interface. At the top is a search bar with the placeholder text 'Search for boxes by operating system, included software, architecture and more' and a magnifying glass icon. Below the search bar are two filter sections: 'Provider' with buttons for 'any', 'virtualbox', 'vmware', 'libvirt', and 'more -'; and 'Sort by' with buttons for 'Downloads', 'Recently Created', and 'Recently Updated'. The main content area displays a search result for 'ubuntu/trusty64 20170619.0.0' with a 'virtualbox' tag. The result includes the Ubuntu logo, the text 'Official Ubuntu Server 14.04 LTS (Trusty Tahr) builds', and statistics: 'Downloads: 29,858,645' and 'Released: 22 days ago'.

Загрузка боксов из публичного репозитория производится с помощью команды **vagrant box add <user>/<box>**, например: **vagrant box add ubuntu/trusty64**.

Также можно загрузить файл по прямой ссылке (URL) или передав путь локальному к бокс-файлу.

Для просмотра списка загруженных боксов используйте команду **vagrant box list**.

Первым шагом в настройке проекта Vagrant является создание **Vagrantfile** в директории проекта, основная функция которого описать настройки создаваемой виртуальной среды.

Для создания Vagrantfile можно использовать команду:

vagrant init ubuntu-VAGRANTSLASH-trusty64

где ubuntu-VAGRANTSLASH-trusty64 — имя бокса, Vagrantfile которого будет использован как шаблон.

В результате будет создан файл примерно следующего содержания:

```
*Vagrantfile ✕
1 # -*- mode: ruby -*-
2 # vi: set ft=ruby :
3 Vagrant.configure("2") do |config|
4   config.vm.box = "ubuntu-VAGRANTSLASH-trusty64"
5   config.vm.box_check_update = false
6   config.vm.network "forwarded_port", guest: 80, host: 8080
7   config.vm.provider "virtualbox" do |vb|
8     vb.gui = true
9     vb.memory = "1024"
10  end
11 end
```

Vagrantfile позволяет задавать имена хостов, переменные окружения, сетевые настройки, ограничивать время ожидания подготовки VM (задавать таймауты), проверять контрольные суммы и мн.др.

Vagrantfile может содержать **циклы** для создания большого количества VM.

Команды для работы с виртуальными средами необходимо запускать из каталога с Vagrantfile. Ниже представлен список наиболее часто используемых команд.

vagrant init	Инициализация среды
vagrant destroy	Уничтожение среды
vagrant up	Запуск среды
vagrant halt	Остановка среды
vagrant suspend	Приостановка среды
vagrant reload	Перезагрузка среды
vagrant ssh	Подключение к консоли ssh
vagrant powershell	Подключение к консоли powershell

Подготовка сред (provisioning) может осуществляться динамически, согласно заданным в Vagrantfile инструкциям, непосредственно в процессе выполнения **vagrant up**.

Пример простого bootstrap-скрипта, который осуществляет установку и настройку веб-сервера apache2:

```
bootstrap.sh ✕
1 #!/usr/bin/env bash
2 apt-get update
3 apt-get install -y apache2
4 if ! [ -L /var/www ]; then
5     rm -rf /var/www
6     ln -fs /vagrant /var/www
7 fi
```

где /vagrant — синхронизируемая директория, которая может храниться локально в хостовой системе; на NFS, SMB или синхронизироваться RSync.

Пример запуска bootstrap-скрипта и подключения синхронизируемой директории "/vagrant" в гостевой системе, к той же директории хостовой системы, где хранится Vagrantfile.

```
3 Vagrant.configure("2") do |config|
4   config.vm.box = "ubuntu-VAGRANTSLASH-trusty64"
5   config.vm.box_check_update = false
6   config.vm.network "forwarded_port", guest: 80, host: 8080
7   config.vm.provider "virtualbox" do |vb|
8     vb.memory = "512"
9   end
10  config.vm.synced_folder ".", "/vagrant"
11  config.vm.provision :shell, path: "./bootstrap.sh"
12 end
```

Если гостевая система уже запущена, достаточно выполнить **vagrant reload --provision**, для быстрого перезапуска гостевой системы с выполнением этапа подготовки среды, минуя длительную процедуру импорта на этапе init.

Для подготовки среды можно использовать Ansible Playbook (стр. 11-13).

```
3 Vagrant.configure("2") do |config|
4   config.vm.box = "ubuntu-VAGRANTSLASH-trusty64"
5   config.vm.box_check_update = false
6   config.vm.network "forwarded_port", guest: 80, host: 8080
7   config.vm.provider "virtualbox" do |vb|
8     vb.memory = "512"
9   end
10  config.vm.synced_folder ".", "/vagrant"
11  config.vm.provision :ansible do |ansible|
12    ansible.playbook = "playbook.yml"
13  end
14 end
```

```
playbook.yml x
1 ---
2 - hosts: all
3   tasks:
4     - name: ensure nginx is at the latest version
5       apt: pkg=nginx state=latest
6       become: true
7       notify:
8         - start nginx
9   handlers:
10    - name: start nginx
11    service: name=nginx state=started
```

Плейбук, который устанавливает и запускает nginx.



Для взаимодействия с хостами на базе ОС **Linux, Unix и Solaris** используется **SSH**, с родной командной оболочкой, например **bash**.



Для взаимодействия с **Windows** в качестве основного средства используется **WinRM**, также возможно использование **WinSSH-коммуникатора**. В качестве командной оболочки используется PowerShell 3.0 или старше.

При запуске Vagrant под управлением ОС Windows, полный путь к домашнему каталогу пользователя, от имени которого запущен Vagrant, должен содержать только символы латинского алфавита и цифры.

Существует специальный вид боксов называемых **базовыми боксами**, которые создаются с нуля, не на базе существующих боксов. Перед упаковкой VM в бокс нужно выполнить некоторые настройки, необходимые для функционирования Vagrant, такие как:

- создать пользователь `vagrant` с паролем `vagrant`, с правами на выполнение `sudo` без пароля;
- для Windows в качестве гостевой системы, дополнительно потребуется отключить UAC, требования к сложности пароля, активировать и настроить WinRM.

Подробнее в [официальной документации](#), на [GitHub](#), а также вот [еще одна неплохая инструкция](#).

Для упаковки VM в бокс используется команда **vagrant package**:

```
vagrant package [options] [name|id]
```

Опции:

<code>--base NAME</code>	имя VM в <code>virtualbox</code>
<code>--output NAME</code>	имя файла назначения
<code>--include FILE...</code>	дополнительные файлы для упаковки в бокс
<code>--vagrantfile FILE</code>	путь к вагрант файлу

Например: `vagrant package --base custom_vm --output custom_vm.box --vagrantfile ./Vagrantfile`

Последовательность действий для того чтобы воссоздать все что сделано на курсе у себя на домашней машине*:

1. Зайти в папку с мастером (D: (C:)\ansible_course\boxes\master и запустить команду:

```
vagrant package mycourse_master
```

Она создаст mycourse_master.box прямо в этой же папке (виртуальная машина будет остановлена!). Этот файл вы можете закатать к себе на flash носитель (размер примерно 1,5 GB)

2. То же самое можно повторить для управляемой машины (D: (C:)\ansible_course\boxes\slave_nix) только с другим именем. Ну и если есть желание с slave_win (будьте внимательны его размер будет около 4GB!)

3. Скачать *.box файлы в произвольную папку на домашне машине и в этой папке для каждого бокса написать команду:

```
vagrant box add <имя_файла_box>
```

Это добавит боксы в локальный репозиторий Vagranta у вас на машине. Список всех доступных локальных боксов можно будет глянуть так:

```
vagrant box list
```

4. На домашней машине нужно создать папки где угодно *\master *\slave_nix (и *\slave_win если нужно).

5. После этого вы можете инициализировать боксы (как описано выше) и начинать пользоваться вашими тестовыми Ansible стендами.

* должны быть установлены VirtualBox 5.1+ и Vagrant 1.9.6. А также у машины должно быть примерно 10 GB RAM