УДК 004.624

Введение в инструмент vagrant

Козич Полина Александровна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Научный руководитель:
Глаголев Владимир Александрович
Приамурский государственный университет имени Шолом-Алейхема
к.г.н., доцент кафедры информационных систем, математики и правовой информатики

Аннотация
В данной статье описаны основные принципы работы с инструментом vagrant. Для лучшего понимания процесс понимания идет поэтапно и последовательно. Для повторения потребуется только компьютер на базе операционной системы windows 10.
Ключевые слова: Vagrant, виртуализация, автоматизация

Vagrant tool introduction

Kozich Polina Alexandrovna
Sholom-Aleichem Priamursky State University
Student

Scientific Sypervisor:
Glagolev Vladimir Alexandrovich
Sholom-Aleichem Priamursky State University
candidate of geographical sciences, Associate Professor of the Department of Information System, Mathematics and law informatics

Abstract
This article describes the basic principles of working with the vagrant tool. For a better understanding, the process of understanding proceeds step by step and sequentially. To repeat, you only need a computer based on the windows 10 operating system.
Keywords: Vagrant, virtualization, automation

Vagrant — это инструмент с открытым исходным кодом, который позволяет создавать, настраивать и управлять блоками виртуальных машин через простой в использовании командный интерфейс. По сути, это уровень программного обеспечения, установленный между инструментом

виртуализации (таким как VirtualBox, Docker, Hyper-V) и виртуальной машиной.

Он часто используется при разработке программного обеспечения, чтобы гарантировать, что все члены команды могут работать на одной конфигурации. Он не только разделяет среды, но и код. Это позволяет коду от одного разработчика работать в системе другого, делая возможным совместную и совместную разработку.

Цель исследования – разобрать базовые принципы работы с инструментом vagrant.

Ранее этим вопросом интересовались Б.Н. Попов, И.С. Капков развивали тему «Особенности администрирования облачной инфраструктуры на основе технологии vagrant» [1] в которой описан процесс развертки окружения с помощью Vagrant. Приводится пример его использования, а также способы и средства администрирования облачной инфраструктуры. Е. Синельников с темой «Отладка и валидация сложных инфраструктурных решений с помощью vagrant» [2], а подробнее про возможность воспроизводимости проблемы для её исправления. В случае сложных инфраструктурных решений из нескольких узлов, многократное соблюдение этого условия становится всё более трудоёмким. В этом докладе представлен один из эффективных сценариев отладки развёртывания и валидации конфигурации Active Directory на базе Samba с помощью механизма управления конфигурациями виртуальных машин Vagrant. С.Н. Елушев, В.Г. Тронин опубликовали статью «Анализ качества управления проектом "vagrant"» [3] описали качество управления проектом «Vagrant» на основе системы контроля версии git. Была проанализирована активность разработчиков как перед первыми версиями продукта, так и за всю историю разработки

Прежде всего, нужно убедиться, что в системе уже есть решение для виртуализации. Решения, которые работают с Vagrant, включают VirtualBox, VMware, Docker, Hyper-V. Они представляют среду для запуска виртуальных машин. После нужно установить Vagrant, последнюю версию можно найти на сайте: https://www.vagrantup.com/downloads.html

На рисунке 1 представлен список поддерживаемых операционных систем. Нужно загрузить соответствующий файл нужной операционной системы, затем запустить программу установки.

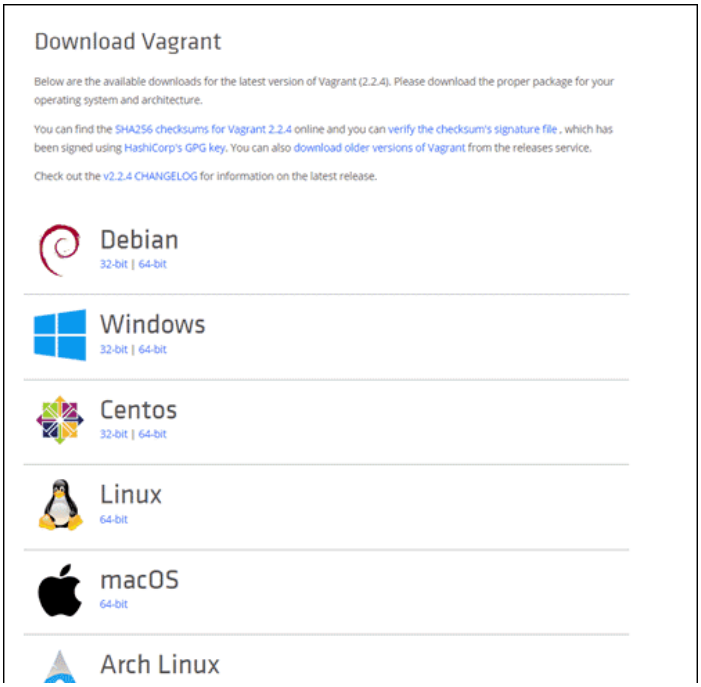

Рис. 1 Поддерживаемые операционные системы

Для проверки правильности установки есть несколько способов:

- Можно ввести в командной строке слово ***vagrant -v*** который должен показать номер версии, запущенной на компьютере.
- Можно ввести следующую команду в командной строке ***vagrant*** которая покажет список часто используемых команд, если инструмент был установлен правильно.

Настройка проекта Vagrant

1. Следует начать с создания каталога для хранения файла Vagrant:
   mkdir vagrant-test
   cd vagrant-test
2. Загрузить дистрибутив Ubuntu Trusty Tahr из общей библиотеки и создать базовый Vagrantfile команда: vagrant init ubuntu/trusty64

Vagrant работает не только с Ubuntu, по следующему адресу https://app.vagrantup.com/boxes/search можно найти множество готовый версий операционных систем под установку через vagrant.

После запуска команды init, Vagrant устанавливает box(коробку) в текущий каталог. Vagrantfile находится в том же каталоге и может быть отредактирован или скопирован.

Vagrant ящики

Базовый блок в настройке Vagrant называется «коробкой» или «Vagrantbox». Это полный, автономный образ среды операционной системы.

Vagrant Box — это клон образа базовой операционной системы. Использование клона ускоряет процесс запуска и подготовки.

Теперь вместо использования приведенной выше команды init можно просто загрузить и добавить поле с командой: vagrant box add ubuntu/trusty64

Эта команда загрузит образ виртуальной машины и сохраняет ее локально. Затем нужно настроить Vagrantfile для работы с виртуальной машиной, которую он будет обслуживать. Нужно открыть Vagrantfile любым текстовым редактором и изменить строку config.vm.box с «base» на «ubuntu / trusty64». Чтобы было так config.vm.box = "ubuntu/trusty64".

Пример такого файла представлен на рисунке 2.

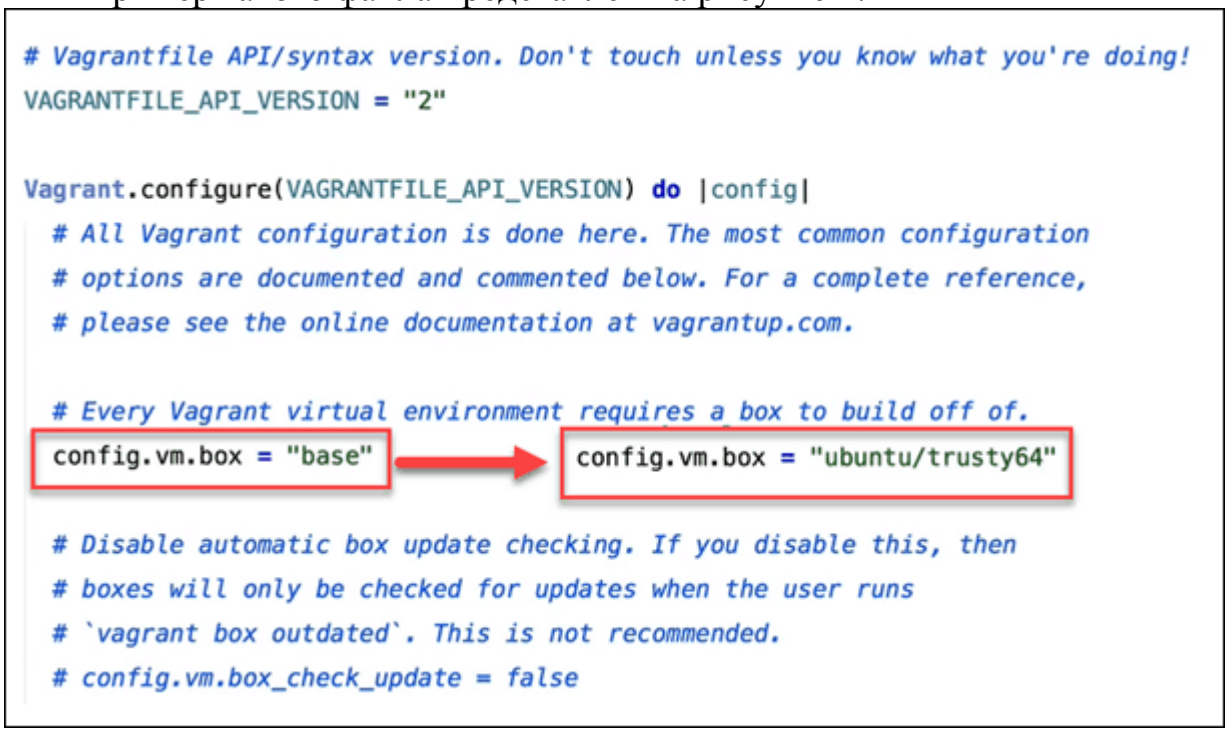

Рис. 2 Пример файла Vagrantfile.

Также можно указать версию «бокса» как показано:

config.vm.box_version = “1.0.1”

Или вообще указать URL адрес до vagrant бокса:

config.vm.box_url = https://vagrantcloud.com/ubuntu/trusty64

Для удаления бокса, нужно применить следующую команду:

vagrant box remove ubuntu/trusty64

VagrantFile

Вместо того чтобы создавать полный образ операционной системы и копировать его, Vagrant использует «Vagrantfile», чтобы указать конфигурацию бокса.

Provisioning

Иногда нужно сделать так чтобы программное обеспечение устанавливалось автоматический с установкой операционной системы. Для этого существует файл bootstrap.sh, сохраненный в том же каталоге, что и Vagrantfile.

Далее идет описание файла с командами для установки монитора ресурсов системы nmon. Файл будет содержать следующие команды.

```
#!/usr/bin/env bash
apt-get update
apt-get install -y nmon
if ! [ -L /var/www ]; then
rm -rf /var/www
ln -fs /vagrant /var/www
```

Теперь нужно сохранить файл и отредактировать Vagrantfile и добавить строку обеспечения. Это должно выглядеть следующим образом:

```
Vagrant.configure("2") do |config|

config.vm.box = "ubuntu/trusty64"

config.vm.provision :shell, path: "bootstrap.sh"

end
```

Когда Vagrant читает файл Vagrantfile, он перенаправляется на чтение файла bootstrap.sh. Этот файл начальной загрузки обновит кеш менеджера пакетов, а затем установит пакет nmon. Теперь этот пакет будет доступен при каждом запуске виртуальной машины.

Такие файлы предоставляют мощный инструмент для предварительной настройки виртуальной среды. Также можно сделать то же самое с apache2 и создать веб-сервер в виртуальной среде.

Провайдеры

Провайдерами выступают сами виртуальные оболочки. Они бывают разнообразные. Поэтому есть настройка, выбор с каким провайдером нужно запустить виртуальную систему.

Для примера так можно запустить vagrant на VMware.

vagrant up –provider=vmware_fusion

А так можно запустить Vagrant, используя Amazon Web Services.

vagrant up –provider=aws

После запуска начальной команды последующие команды будут применены к тому же провайдеру.

Запуск и подключение

Основная команда для запуска новой виртуальной среды:

vagrant up

Эта команда запустит программное обеспечение и виртуальную среду Ubuntu. Однако, даже если виртуальная машина работает, у неё нет никаких выходных данных. Vagrant не предоставляет никакого пользовательского интерфейса. Другими словами, нужно работать с виртуальной системой через консоль.

Vagrant SSH

У vagrant есть возможность подключиться к виртуальной машине (и убедиться, что она работает), используя соединение SSH:

vagrant ssh

Это открывает безопасное соединение оболочки с новой виртуальной машиной. Командная строка изменится на vagrant@trusty64, чтобы указать, что авторизация прошла успешно.

После изучения виртуальной машины чтобы выйти, нужно нажать CTRL-D. Виртуальная машина всё еще будет работать в фоновом режиме, но соединение SSH будет закрыто.

Чтобы удалить виртуальную машину, нужно ввести следующую команду:

vagrant destroy

Загруженный файл чистой операционной системы останется, но все данные с виртуальной машины будут уничтожены.

Синхронизированные папки

Vagrant автоматически синхронизирует содержимое в каталоге проекта со специальным каталогом в гостевой (виртуальной) системе.

При запуске виртуальной машины, по умолчанию она запускается в каталоге /home/vagrant/. В другом каталоге /vagrant/ содержатся те же файлы, что и на хост-системе.

Это удобный способ управления файлами в гостевой ОС без использования сеанса SSH.

Сети

Vagrant включает опции для размещения виртуальной машины в сети. В конце Vagrantfile файла, непосредственно перед end командой, нужно вставить инструкцию config.vm.network, чтобы указать настройки сети.

Пример:

config.vm.network “forwarded_port”, guest: 80, host: 8080

Чтобы изменения вступили в силу, нужно сохранить и перезагрузить Vagrant с помощью команды:

vagrant reload

Эта инструкция перенаправленный порт для гостевой системы. Также можно определить частные сети, публичные сети и другие более сложные параметры.

Vagrant Share

Vagrant имеет удобную функцию, которую используют чтобы поделиться своей средой Vagrant, используя пользовательский URL.

Когда среда Vagrant запущена, нужно использовать следующую команду:

vagrant share

Система создаст сеанс Vagrant Share, а затем сгенерирует URL. Этот URL можно скопировать и отправить другому лицу. Если в сеансе Vagrant настроен Apache, любой, кто использует этот URL, может увидеть страницу конфигурации Apache. Этот URL-адрес изменяется при изменении содержимого общей папки.

Также можно закрыть сеанс обмена с помощью CTRL-C .

Vagrant завершение

Когда пользователь заканчивает работу с гостевой системой, у него есть несколько вариантов завершения сеанса.

1. Чтобы остановить машину и сохранить ее текущее состояние, нужно выполнить команду:

vagrant suspend

Можно продолжить, запустив vagrant up снова. Это очень похоже на перевод машины в спящий режим.

2. Для выключения виртуальной машины нужно использовать команду:

vagrant halt

Опять же, vagrant up перезагрузится, и снова можно продолжать работу. Это очень похоже на выключение обычной машины.

3. Чтобы удалить все следы виртуальной машины из системы, нужно ввести:

vagrant destroy

Все данные на виртуальной машине, будут удалены.

В следующий раз нужно будет снова устанавливать систему. Это очень похоже на форматирование жесткого диска в системе, а затем установку нового образа.

Вывод

В результате статьи были разобраны основные компоненты популярного инструмента vagrant. Описаны его ключевые особенности с приведением примеров кода. Рассказано как его установить и как начать использовать.

Библиографический список

1. Попов Б.Н., Капков И.С. Особенности администрирования облачной инфраструктуры на основе технологии vagrant // Информационные технологии и системы: управление, экономика, транспорт, право. 2014. № 3. С. 253-260. URL: https://elibrary.ru/item.asp?id=22600373 (Дата обращения: 11.12.2019)
2. Синельников Е. Отладка и валидация сложных инфраструктурных решений с помощью vagrant. // В книге: Четырнадцатая конференция разработчиков свободных программ Тезисы докладов. Ответственный редактор В.Л. Черный. 2017. С. 79-83. URL: https://elibrary.ru/item.asp?id=30044716 (Дата обращения: 11.12.2019)
3. Елушев С.Н., Тронин В.Г. Анализ качества управления проектом "vagrant" // В сборнике: Прикладные информационные системы Сборник научных трудов. 2016. С. 395-401. URL: https://elibrary.ru/item.asp?id=28308802 (Дата обращения: 11.12.2019)