



Методическое пособие по выполнению домашнего задания по курсу «Администратор Linux. Professional»

Занятие 1. Vagrant-стенд для обновления ядра и создания образа системы

Цель домашнего задания

Научиться обновлять ядро в ОС Linux. Получение навыков работы с Vagrant.

Описание домашнего задания

- 1) Запустить ВМ с помощью Vagrant.
- 2) Обновить ядро ОС из репозитория ELRepo.
- 3) Оформить отчет в README-файле в GitHub-репозитории.

Дополнительные задания:

- Ядро собрано из исходников
- В образе нормально работают VirtualBox Shared Folders

Введение

Данные методические рекомендации помогут разобраться с инструментом Vagrant, а также получить базовые навыки работы с системами контроля версий. Эти технологии помогут Вам в выполнении следующих домашних работ.

Рассмотрим подробнее данные инструменты:

- **Vagrant** – ПО для создания и конфигурирования виртуальной среды.
- **Git** – система контроля версий.
- **GitHub** – веб-сервис для хостинга IT-проектов и их совместной разработки.

В современном мире операционные системы постоянно обновляются, при обновлении ОС закрываются уязвимости программного обеспечения и исправляются ошибки. Обновление операционной системы является одним из важных навыков администратора Linux.

В данном руководстве рассмотрен процесс обновления ядра Linux в ОС.

Функциональные и нефункциональные требования

- ПК на Linux с 16 ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.
- Созданный аккаунт на GitHub – <https://github.com/>
- Если Вы находитесь в России, для корректной работы с ресурсами компании Hashicorp Вам может потребоваться VPN или зеркало в России.

Предварительно установленное и настроенное следующее ПО:

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

- Oracle VirtualBox (https://www.virtualbox.org/wiki/Linux_Downloads) .
- Любой редактор кода, например Visual Studio Code, Atom и т.д.

Инструкция по выполнению домашнего задания

Все дальнейшие действия были проверены при использовании Vagrant 2.4.0, VirtualBox 7.0.14, хостовая ОС: Ubuntu 22.04 Desktop. В лабораторной работе используются Vagrant boxes с CentOS 8 Stream. Серьёзные отступления от этой конфигурации могут потребовать адаптации с вашей стороны.

Важно! При копировании из текстового редактора (в том числе из этого документа) возможны искажения символов, например, для дефисов, тире, пробелов и т.д. Поэтому копирование кода лучше проводить из репозитория (https://github.com/Nickmob/vagrant_kernel_update) с примерами непосредственно в редакторах кода (VSCode, nano, vim, mcedit и других).

Текст конфигов здесь представлен для ознакомления.

Создадим Vagrantfile, в котором будут указаны параметры нашей ВМ:

```
# Описываем Виртуальные машины
MACHINES = {
  # Указываем имя ВМ "kernel update"
  :"kernel-update" => {
    #Какой vm box будем использовать
    :box_name => "generic/centos8s",
    #Указываем box_version
    :box_version => "4.3.4",
    #Указываем количество ядер ВМ
    :cpus => 2,
    #Указываем количество ОЗУ в мегабайтах
    :memory => 1024,
  }
}

Vagrant.configure("2") do |config|
  MACHINES.each do |boxname, boxconfig|
    # Отключаем проброс общей папки в ВМ
    config.vm.synced_folder ".", "/vagrant", disabled: true
    # Применяем конфигурацию ВМ
    config.vm.define boxname do |box|
      box.vm.box = boxconfig[:box_name]
      box.vm.box_version = boxconfig[:box_version]
      box.vm.host_name = boxname.to_s
      box.vm.provider "virtualbox" do |v|
        v.memory = boxconfig[:memory]
        v.cpus = boxconfig[:cpus]
      end
    end
  end
end
```

После создания Vagrantfile, запустим виртуальную машину командой `vagrant up`. Будет создана виртуальная машина с ОС CentOS 8 Stream, с 2-мя ядрами CPU и 1ГБ ОЗУ.

Обновление ядра

Подключаемся по ssh к созданной виртуальной машины. Для этого в каталоге с нашим Vagrantfile вводим команду `vagrant ssh`

Перед работами проверим текущую версию ядра:

```
[vagrant@kernel-update ~]$ uname -r  
4.18.0-277.el8.x86_64
```

Далее подключим репозиторий, откуда возьмём необходимую версию ядра:

```
sudo yum install -y  
https://www.elrepo.org/elrepo-release-8.el8.elrepo.noarch.rpm
```

В репозитории есть две версии ядер:

- `kernel-ml` – свежие и стабильные ядра
- `kernel-lt` – стабильные ядра с длительной версией поддержки, более старые, чем версия `ml`.

Установим последнее ядро из репозитория `elrepo-kernel`:

```
sudo yum --enablerepo elrepo-kernel install kernel-ml -y
```

Параметр `--enablerepo elrepo-kernel` указывает что пакет ядра будет запрошен из репозитория `elrepo-kernel`.

Уже на этом этапе можно перезагрузить нашу виртуальную машину и выбрать новое ядро при загрузке ОС.

Если требуется, можно назначить новое ядро по-умолчанию вручную:

1) Обновить конфигурацию загрузчика:

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

2) Выбрать загрузку нового ядра по-умолчанию:

```
sudo grub2-set-default 0
```

Далее перезагружаем нашу виртуальную машину с помощью команды `sudo reboot`

После перезагрузки снова проверяем версию ядра (версия должна стать новее):

```
[vagrant@kernel-update ~]$ uname -r  
5.18.3-1.el8.elrepo.x86_64
```

На этом обновление ядра закончено.

Критерии оценивания

Статус «Принято» ставится при выполнении следующих условий:

1. Ссылка на репозиторий GitHub в котором находятся файлы указанные в последующих пунктах.
2. Vagrantfile, который будет разворачивать виртуальную машину в которой вы обновляли ядро.
3. Документация по каждому заданию:

Создайте файл README.md и снабдите его следующей информацией:

- название выполняемого задания;
- текст задания;
- ;
- особенности проектирования и реализации решения,
- заметки, если считаете, что имеет смысл их зафиксировать в репозитории.

Рекомендуемые источники

- Репозиторий vagrant_kernel_update –
https://github.com/Nickmob/vagrant_kernel_update
- Статья о GitHub – <https://ru.wikipedia.org/wiki/GitHub>
- Elrepo HomePage – <http://elrepo.org/tiki/HomePage>