

Подготовка к выполнению практических заданий

Установите ПО

- Удобный текстовый редактор (Notepad++, Sublime)
- VirtualBox
- Git
- Vagrant (<https://hashicorp-releases.yandexcloud.net/vagrant/>)
- Опционально - IDE (IntelliJ, VSCode, ...)

Установка из-под Windows+Chocolatey:

```
choco install virtualbox
choco install vagrant
choco install git
```

опционально:

```
choco install jdk8
choco install intellijidea-community
```

Под MacOS+Brew:

```
brew cask install virtualbox
brew cask install vagrant
brew install git
```

опционально:

```
brew tap adoptopenjdk/openjdk
brew cask install adoptopenjdk/openjdk/adoptopenjdk8
```

Установить плагины:

```
vagrant plugin install vagrant-hostmanager
vagrant plugin install vagrant-vbguest
```

Подключить образ (т.е. добавить образ в кэш) vagrant:

Образы находятся в C:\Install\VagrantBox. Понадобятся три образа:

- **docker-community-ubuntu-server.box** <https://disk.yandex.ru/d/I2ixvAeehdYXzA> (1.08 ГБ)
- **ubuntu.box**
- **centos7.box**

Для подключения образа в командной строке выполните команду из папки со скачанным файлом по следующему шаблону:

```
vagrant box add <название_образа> <название_файла_образа.box>
```

Иначе говоря, выполните три команды:

```
vagrant box add docker-community-ubuntu-server docker-community-ubuntu-server.box
vagrant box add ubuntu ubuntu.box
vagrant box add centos7 centos7.box
```

Занятие 4. Виртуализация

1. Развертывание виртуальных машин в GUI (опционально)

Запускаем VirtualBox.

- * Создать, выбрать тип ОС (Linux)
- * Задать объем памяти
- * Выбрать накопитель: HDD, новый, VDI, динамический, оставить файл по-умолчанию

Теперь надо установить ОС

- * Гуглим Google CentOS ISO
- * Скачиваем нужный образ (CentOS-7-x86_64-Minimal-2009.iso)

В VirtualBox:

- * Настройки
 - * Носители - IDE - Вторичный - Выбрать файл диска - выбрать файл
 - * Сеть - Включить - NAT или Мост (Bridged)

В виртуальной машине:

- * Запустить
- ** Enter (установить ОС)
- ** Язык, Сеть - включить (сверить, что IP попадает в нужную подсеть)
- ** Начать установку
- * После окончания - перезагрузить или выключить VM

В VirtualBox:

- * Настройки - Носители - Извлечь диск из привода

В Putty:

- * Подключиться к виртуалке по IP-адресу
- * Выключаем

2. Развертывание виртуальных машин в Vagrant

На локальном ПК

```
* Создаем каталог для виртуальной машины и переходим в него
mkdir centos7
cd centos7
```

```
* Создаем шаблонный файл настроек под нужный образ
vagrant init geerlingguy/centos7
```

```
* Теперь можно поднять виртуалку
vagrant up
```

```
* Подключиться к гостевой ОС
vagrant ssh
```

В гостевой ОС:

```
whoami
exit
```

На локальном ПК

```
* Выключить виртуалку
vagrant halt
```

Виртуалка выключена

Повторим с другой гостевой ОС

```
cd ..
mkdir ubuntu20
cd ubuntu20
vagrant init geerlingguy/ubuntu2004
vagrant up
vagrant ssh
exit
vagrant halt
```

Можно убедиться, что список действий идентичен

```
history
```

3. Команды и опции Vagrant

Документация по Vagrant доступна на vagrantup.com/docs

Еще о базовом управлении

```
# Чтобы удалить виртуалку с ПК
vagrant destroy

# Для перезагрузки
vagrant reload

# Статус всех машин
vagrant global-status
# Статус всех машин с вычисткой некорректных записей
vagrant global-status --prune
# Статус текущей машины (из её каталога)
vagrant status
```

Vagrantfile

Все настройки начинаются с ключевого слова `config` (имя класса, отличается во вложенных классах)

```
config.vm.network "private_network", ip: "..." # Статический ip
config.vm.network "public_network" # Динамический ip, можно задать и статику

# Настройки провайдера (гипервизора)
config.vm.provider "virtualbox" do |vb|
  vb.name = "my_vm"
  vb.memory = "1600" # В мегабайтах
  vb.cpus = "1"
end

# Переменные окружения
# Полезно в случае если гостевая ВМ не поддерживает локаль вашего хоста
ENV["LC_ALL"] = "en_US.UTF-8"

# Передать чувствительную информацию
config.vagrant.sensitive = ["MySecretPassword", ENV["MY_TOKEN"]]

# Задать таймаут поднятия машины (в секундах)
config.vm.boot_timeout = 900
```

4. Папки синхронизации

Vagrant будет синхронизировать все файловые структуры внутри папки машины (там, где Vagrantfile) с папкой /vagrant в гостевой ОС.

Можно использовать любую локальную папку, если описать ее в Vagrantfile.

Пример (подставьте свой путь):

```
config.vm.synced_folder "C:\\do\\VM\\syncfolders\\myshellscripts",
"/opt/scripts"
```

5. Provisioning

Так называют выполнение заданного набора команд при первом запуске гостевой ОС. В Vagrant это настраивается в последней секции шаблона Vagrantfile

В Vagrantfile новой виртуальной машины

Есть разные provisioners (Ansible, chef, puppet, ...), здесь пример на shell

```
config.vm.provision "shell", inline: <<-SHELL
    yum install httpd wget unzip -y
    mkdir /opt/devopsdir
    free -m
    uptime
SHELL
```

Затем поднимаем виртуалку:

```
vagrant up
```

Если виртуалка уже запускалась, vagrant up не будет вызывать provisioning, если не указать ключ --provision:

```
vagrant up --provision
```

или

```
vagrant reload --provision
```

Для установки апача на Ubuntu:

В Vagrantfile:

```
config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y apache2
SHELL
```

На локальном ПК:

```
vagrant ssh
```

В гостевой ОС:

```
sudo systemctl status apache2
```

Статус должен быть active(running)

На локальном ПК:

* Для проверки открыть браузер и перейти на http://<ip-адрес виртуалки>

* Наблюдаем дефолтную страничку апача

6. Автоматическое развертывание статичного веб-сайта

На локальном ПК:

```
mkdir website
cd website
```

```
vagrant init geerlingguy/centos7
```

В Vagrantfile:

```
Vagrant.configure("2") do |config|
  config.vm.box = "geerlingguy/centos7"
  config.vm.network "private_network", ip: "192.168.3.14"
  config.vm.provider "virtualbox" do |vb|
    vb.name = "my_web"
    vb.memory = "512"
    vb.cpus = 1
  end
  config.vm.provision "shell", inline: <<-SHELL
    yum install httpd wget unzip -y
    systemctl start httpd
    systemctl enable httpd
    cd /tmp/
    wget https://www.tooplate.com/zip-templates/2124_vertex.zip
    unzip 2124_vertex.zip
    cp -r 2124_vertex/* /var/www/html
    systemctl restart httpd
  SHELL
end
```

На локальном ПК:

```
vagrant up
```

* Для проверки открыть браузер и перейти на <http://<ip-адрес виртуалки>>

7. Автоматическое развертывание веб-сайта на Wordpress

На локальном ПК:

```
mkdir wordpress
cd wordpress
```

```
vagrant init geerlingguy/ubuntu2004
```

В Vagrantfile:

```
Vagrant.configure("2") do |config|
  config.vm.box = "geerlingguy/ubuntu2004"
  config.vm.network "private_network", ip: "192.168.3.15"
  config.vm.provider "virtualbox" do |vb|
    vb.name = "wordpress_auto"
    vb.memory = "1024"
    vb.cpus = 2
  end
  config.vm.provision "shell", inline: <<-SHELL
    sudo apt update
    sudo apt install apache2 \
      ghostscript \
      libapache2-mod-php \
      mysql-server \
      php \
      php-bcmath \
      php-curl \

```

```

        php-imagick \
        php-intl \
        php-json \
        php-mbstring \
        php-mysql \
        php-xml \
        php-zip -y

sudo mkdir -p /srv/www
sudo chown www-data: /srv/www
curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -C
/srv/www

cp /vagrant/wordpress.conf /etc/apache2/sites-available/wordpress.conf

sudo a2ensite wordpress
sudo a2enmod rewrite
sudo a2dissite 000-default
sudo service apache2 reload

mysql -u root -e 'CREATE DATABASE wordpress;'
mysql -u root -e 'CREATE USER wordpress@localhost IDENTIFIED BY
"lesson123";'
mysql -u root -e 'GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER ON
wordpress.* TO wordpress@localhost;'
mysql -u root -e 'FLUSH PRIVILEGES;'

sudo -u www-data cp /srv/www/wordpress/wp-config-sample.php
/srv/www/wordpress/wp-config.php

sudo -u www-data sed -i 's/database_name_here/wordpress/'
/srv/www/wordpress/wp-config.php
sudo -u www-data sed -i 's/username_here/wordpress/' /srv/www/wordpress/wp-
config.php
sudo -u www-data sed -i 's/password_here/lesson123/' /srv/www/wordpress/wp-
config.php

SHELL

end

```

Создайте файл wordpress.conf рядом с Vagrantfile со следующим содержанием:

```

<VirtualHost *:80>
    DocumentRoot /srv/www/wordpress
    <Directory /srv/www/wordpress>
        Options FollowSymLinks
        AllowOverride Limit Options FileInfo
        DirectoryIndex index.php
        Require all granted
    </Directory>
    <Directory /srv/www/wordpress/wp-content>
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>

```

Поднимите виртуальную машину:
vagrant up

* Для проверки открыть браузер и перейти на <http://<ip-адрес виртуалки>>

8. Домашнее задание: конфигурация на несколько виртуалок

На локальном ПК:

```
mkdir multivm
cd multivm
```

Создайте Vagrantfile:

```
$webscript = <<-SCRIPT1
# Сюда вставьте блок из Provisioning виртуалки, как описано в п.6.
SCRIPT1

$dbscript = <<-SCRIPT2
# Сюда вставьте блок из Provisioning виртуалки, как описано в п.7.
SCRIPT2

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

  config.vm.boot_timeout = 900

  config.vm.define "web01" do |web01|
    web01.vm.box = "geerlingguy/centos7"
    web01.vm.network "public_network"
    web01.vm.provider "virtualbox" do |vb|
      vb.name = "web01"
      vb.memory = "1024"
    end
    web01.vm.provision "shell", inline: $webscript
  end

  config.vm.define "db01" do |db01|
    db01.vm.box = "geerlingguy/ubuntu2004"
    db01.vm.network "public_network"
    db01.vm.provider "virtualbox" do |vb|
      vb.name = "db01"
      vb.memory = "1024"
    end

    db01.vm.provision "shell", inline: $dbscript
  end

end
```

* Обязательно проверьте блоки в Vagrantfile на предмет наличия всех необходимых **end** и отсутствия лишних.

Поднимите виртуалки:

```
vagrant up
```

Чтобы зайти на конкретную виртуалку в конфигурации с несколькими VM нужно указать её имя:

```
vagrant ssh web01
```

* Другие команды в случае multi-VM можно использовать без имени, тогда они относятся ко всем виртуалкам, описанным в конфигурации.

Посмотрите на сетевые адаптеры виртуалок и их параметры командой `ip addr`.

Погасите (halt) виртуалки и замените в Vagrantfile опции сети с "public_network" на "private_network", `virtualbox____intnet: true`

Снова поднимите виртуалки и посмотрите на изменения в сетевых адаптерах виртуальных машин.

9. Опционально (при необходимости): Добавить образ в кэш Vagrant

Образ **gusztavvargadr/docker-linux**, который нам потребуется на занятии по контейнерам, можно скачать с личного облака преподавателя:

<https://disk.yandex.ru/d/I2ixvAeehdYXzA> (1.08 ГБ).

Чтобы он был доступен для развертывания, его необходимо добавить в кэш Vagrant-а. Для этого положите его в папку, путь к которой не содержит пробелов (например, C:\VM\VagrantBoxes\docker-community-ubuntu-server.box) и из командной строки выполните команду:

```
vagrant box add docker-community-ubuntu-server docker-community-ubuntu-server.box
```

Теперь вы сможете использовать его для развертывания ВМ, используя имя **docker-community-ubuntu-server**