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

**Vagrant-стенд c PXE**

**Цель домашнего задания**

Отработать навыки установки и настройки DHCP, TFTP, PXE загрузчика и автоматической загрузки

**Описание домашнего задания**

1. Следуя шагам из документа <https://docs.centos.org/en-US/8-docs/advanced-install/assembly_preparing-for-a-network-install> установить и настроить загрузку по сети для дистрибутива CentOS 8.

В качестве шаблона воспользуйтесь репозиторием <https://github.com/nixuser/virtlab/tree/main/centos_pxe>

2. Поменять установку из репозитория NFS на установку из репозитория HTTP.

3. Настроить автоматическую установку для созданного kickstart файла (\*) Файл загружается по HTTP.

\* 4. автоматизировать процесс установки Cobbler cледуя шагам из документа <https://cobbler.github.io/quickstart/>.

*Задание со звездочкой выполняется по желанию.*

Формат сдачи ДЗ - vagrant + ansible

**Введение**

Бывают ситуации, когда ИТ-специалисту потребуется переустановить ОС на большом количестве хостов. Переустановка вручную потребует от специалиста большого количества времени. В этот момент стоит обратить внимание на PXE.

PXE (Preboot eXecution Environment) — это набор протоколов, которые позволяют загрузить хост из сети. Для загрузки будет использоваться сетевая карта хоста.

Для PXE требуется:

* Со стороны клиента (хоста на котором будем устанавливать или загружать ОС):
  + Cетевая карта, которая поддерживает стандарт PXE
* Со стороны сервера:
  + DHCP-сервер
  + TFTP-сервер

TFTP (Trivial File Transfer Protocol) — простой протокол передачи файлов, используется главным образом для первоначальной загрузки бездисковых рабочих станций. Основная задача протокола TFTP — отправка указанных файлов клиенту.

TFTP работает на 69 UDP порту. TFTP — очень простой протокол, у него нет аутентификации, возможности удаления файлов и т д. Протокол может только отправлять запросы на чтение и запись…

DHCP (Dynamic Host Configuration Protocol) — протокол динамической настройки узла, позволяет сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

Протокол DHCP пришёл на смену протоколу BOOTP. DHCP сохраняет обратную совместимость с BOOTP. Основное отличие протоколов заключается в том, что протокол DHCP помимо IP-адреса может отправлять клиенту дополнительные опции (маску подсети, адреса DNS-серверов, имя домена, адрес TFTP-сервера).

Протокол DHCP использует следующие порты:

* UDP 67 на сервере
* UDP 68 на клиенте

Также DHCP позволяет DHCP-клиенту отправить ответом опции для DHCP-сервера.

Через DHCP мы можем передать клиенту адрес PXE-сервера и имя файла, к которому мы будем обращаться.

**Функциональные и нефункциональные требования**

* ПК на Unix c 8ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.

Предварительно установленное и настроенное следующее ПО:

* Hashicorp Vagrant (<https://www.vagrantup.com/downloads>)
* Oracle VirtualBox (<https://www.virtualbox.org/wiki/Linux_Downloads>).
* Ansible (версия 2.8 и выше) - <https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html>
* Любой редактор кода, например Visual Studio Code, Atom и т.д.
* **\*(Не обязательно)** Python версии 3 и выше - (<https://python-scripts.com/install-python>)

**Инструкция по выполнению домашнего задания**

Все дальнейшие действия были проверены при использовании Vagrant 2.2.19, VirtualBox v6.1.26 r145957 и Ansible 2.9. Серьёзные отступления от этой конфигурации могут потребовать адаптации с вашей стороны.

**1. Работа с шаблоном из задания**

Скачиваем файлы, указанные в домашнем задании. Рассмотрим загруженный Vagrantfile:

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

**# export VAGRANT\_EXPERIMENTAL="disks"**

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

config**.**vm**.**define "pxeserver" **do** **|**server**|**

**server.vm.box = 'centos/8.4'**

**server.vm.disk :disk, size: "15GB", name: "extra\_storage1"**

server**.**vm**.**host\_name **=** 'pxeserver'

server**.**vm**.**network :private\_network**,**

ip: "10.0.0.20"**,**

virtualbox\_\_intnet: 'pxenet'

**# server.vm.network "forwarded\_port", guest: 80, host: 8081**

server**.**vm**.**provider "virtualbox" **do** **|**vb**|**

vb**.**memory **=** "1024"

vb**.**customize **[**"modifyvm"**,** :id**,** "--natdnshostresolver1"**,** "on"**]**

**end**

**# ENABLE to setup PXE**

**server.vm.provision "shell",**

**name: "Setup PXE server",**

**path: "setup\_pxe.sh"**

**end**

# config used from this

# https://github.com/eoli3n/vagrant-pxe/blob/master/client/Vagrantfile

config**.**vm**.**define "pxeclient" **do** **|**pxeclient**|**

**pxeclient.vm.box = 'centos/8.4'**

pxeclient**.**vm**.**host\_name **=** 'pxeclient'

pxeclient**.**vm**.**network :private\_network**,** ip: "10.0.0.21"

pxeclient**.**vm**.**provider :virtualbox **do** **|**vb**|**

vb**.**memory **=** "2048"

vb**.**customize **[**"modifyvm"**,** :id**,** "--natdnshostresolver1"**,** "on"**]**

vb**.**customize **[**

'modifyvm'**,** :id**,**

'--nic1'**,** 'intnet'**,**

'--intnet1'**,** 'pxenet'**,**

'--nic2'**,** 'nat'**,**

'--boot1'**,** 'net'**,**

'--boot2'**,** 'none'**,**

'--boot3'**,** 'none'**,**

'--boot4'**,** 'none'

**]**

vb**.**customize **[**"modifyvm"**,** :id**,** "--natdnshostresolver1"**,** "on"**]**

**end**

# ENABLE to fix memory issues

# end

**end**

**end**

Жирным шрифтом отмечены строки, в которых требуется внести изменения. Давайте рассмотрим их более подробно:

* **Pxeclient.vm.box = 'centos/8.4' и server.vm.box = 'centos/8.4' —** на данный момент в Vagrant Box нет образа с таким именем. Нам требуется образ CentOS 8.4, мы можем воспользоваться образом **bento/centos-8.4.** Плюсом этого Vagrant Box является то, что по умолчанию он создаёт ОС с размером диска 60ГБ. При использовании данного образа нам не придётся полдключать дополнительный диск.
* **# export VAGRANT\_EXPERIMENTAL="disks"** и **server.vm.disk :disk, size: "15GB", name: "extra\_storage1" —** так как нам хватает свободного места, мы можем не подключать дополнитеный диск. *Если вы планируете в своём домашнем задании подключить дополнительный диск, то команда* ***export VAGRANT\_EXPERIMENTAL="disks"*** *должна быть введена в терминале.*
* ***# server.vm.network "forwarded\_port", guest: 80, host: 8081 —*** опция проброса порта. В нашем ДЗ её рекомендуется расскомментировать. Также для удобства можно поменять порт 8081 на любой удобный Вам.
* **# ENABLE to setup PXE** — блок настройки PXE-сервера с помощью bash-скрипта. Так как мы будем использовать Ansible для настройки хоста, данный блок нам не понадобится. Его можно удалить. Далее можно будет добавить блок настройки хоста с помощью Ansible…
* Для настройки хоста через Ansible, нам потребуется добавить дополнтельный сетевой интефейс для Pxeserver. Пример добавления сетевого интефейса, с адресом 192.168.50.10:

**server.vm.network :private\_network, ip: "192.168.50.10", adapter: 3**

После внесения всех изменений запускаем наш стенд с помощью команды *vagrant up*

*Выполнение команды закончится с ошибкой, так как на Pxeclient настроена загрузка по сети.*

Теперь мы можем приступить к настройке Pxe-сервера.

Для настроки хоста с помощью Ansible нам нужно создать несколько файлов и положить их в отдельную папку (в моём примере имя папки ansible):

* Конфигурационный файл: ansible.cfg — файл описывает базовые настройки для работы Ansible:

*[defaults]*

*#Отключение проверки ключа хоста*

*host\_key\_checking = false*

*#Указываем имя файла инвентаризации*

*inventory = hosts*

*#Отключаем игнорирование предупреждений*

*command\_warnings= false*

* Файл инвентаризации host — данный файл хра
* нит информацию о том, как подключиться к хосту:

[servers]

pxeserver ansible\_host=192.168.50.10 ansible\_user=vagrant ansible\_ssh\_private\_key\_file=.vagrant/machines/inetRouter/virtualbox/private\_key

* + [servers] - в квадратных скобках указана группа хостов
  + *pxeserver —* имя нашего хоста (имена хостов и групп не могут быть одинаковые)
  + *ansible\_host —* адрес нашего хоста
  + *ansible\_user —* имя пользователя, с помощью которого Ansible будет подключаться к хосту
  + *ansible\_ssh\_private\_key —* адрес расположения ssh-ключа
* Ansible-playbook provision.yml — основной файл, в котором содержатся инструкции (модули) по настройке для Ansible.
* Дополнительно можно создать каталоги для темплейтов конфигурационных файлов (templates) и файлов с переменными (defaults)

**Настройка Web-сервера**

Для того, чтобы отдавать файлы по HTTP нам потребуется настроенный веб-сервер.

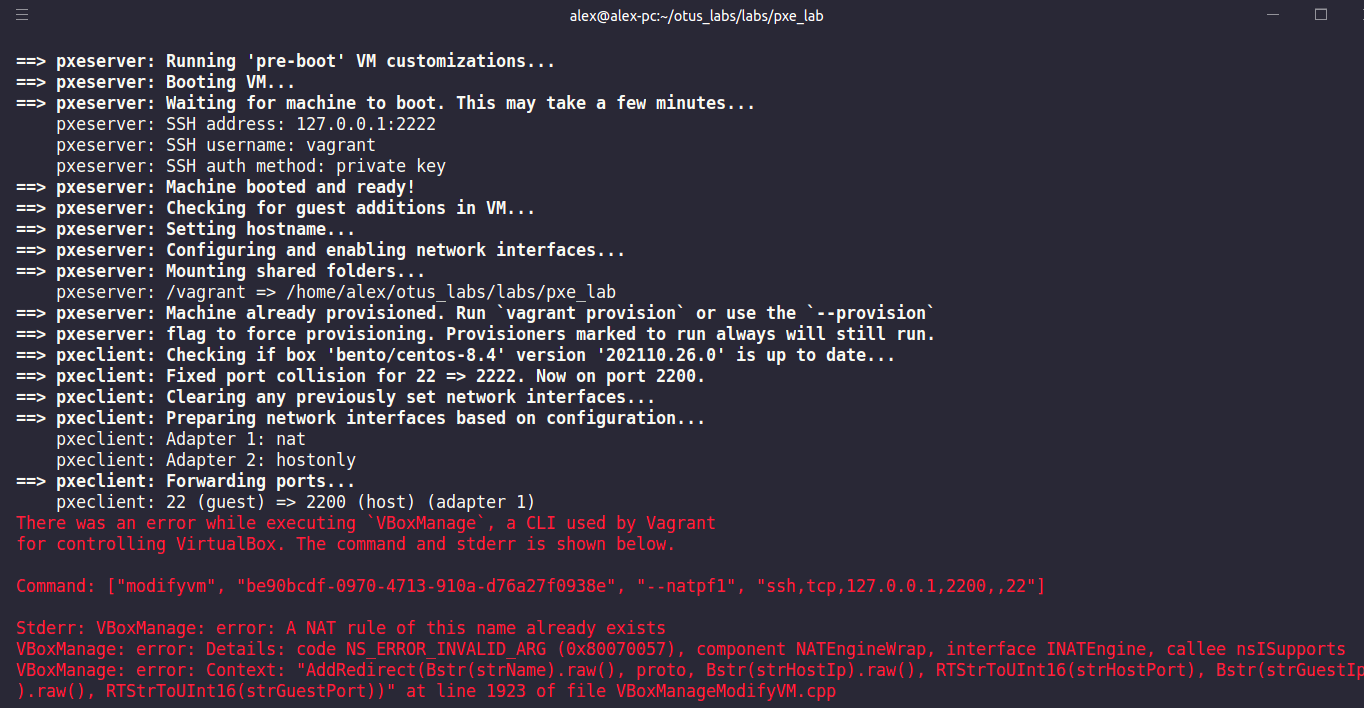
Процесс настройки вручную:

0. Так как у CentOS 8 закончилась поддержка, для установки пакетов нам потребуется поменять репозиторий. Сделать это можно с помощью следующих команд:

*sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-Linux-\**

*sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-Linux-\**

1. Устанавливаем Web-сервер Apache: *yum install httpd*

2. Далее скачиваем образ CentOS 8.4.2150:

*wget https://mirror.sale-dedic.com/centos/8.4.2105/isos/x86\_64/CentOS-8.4.2105-x86\_64-dvd1.iso*

*Размер образа больше 9ГБ, скачивание может занять продолжительное время.*

3. Монтируем данный образ:

*mount -t iso9660 CentOS-8.4.2105-x86\_64-dvd1.iso /mnt -o loop,ro*

4. Создаём каталог /iso и копируем в него содержимое данного каталога:

*mkdir /iso*

*cp -r /mnt/\* /iso*

5. Ставим права 755 на каталог /iso: *chmod -R 755 /iso*

6. Настраиваем доступ по HTTP для файлов из каталога /iso:

* Создаем конфигурационный файл: *vi /etc/httpd/conf.d/pxeboot.conf*
* Добавляем следующее содержимое в файл:

*Alias /centos8 /iso*

*#Указываем адрес директории /iso*

*<Directory /iso>*

*Options Indexes FollowSymLinks*

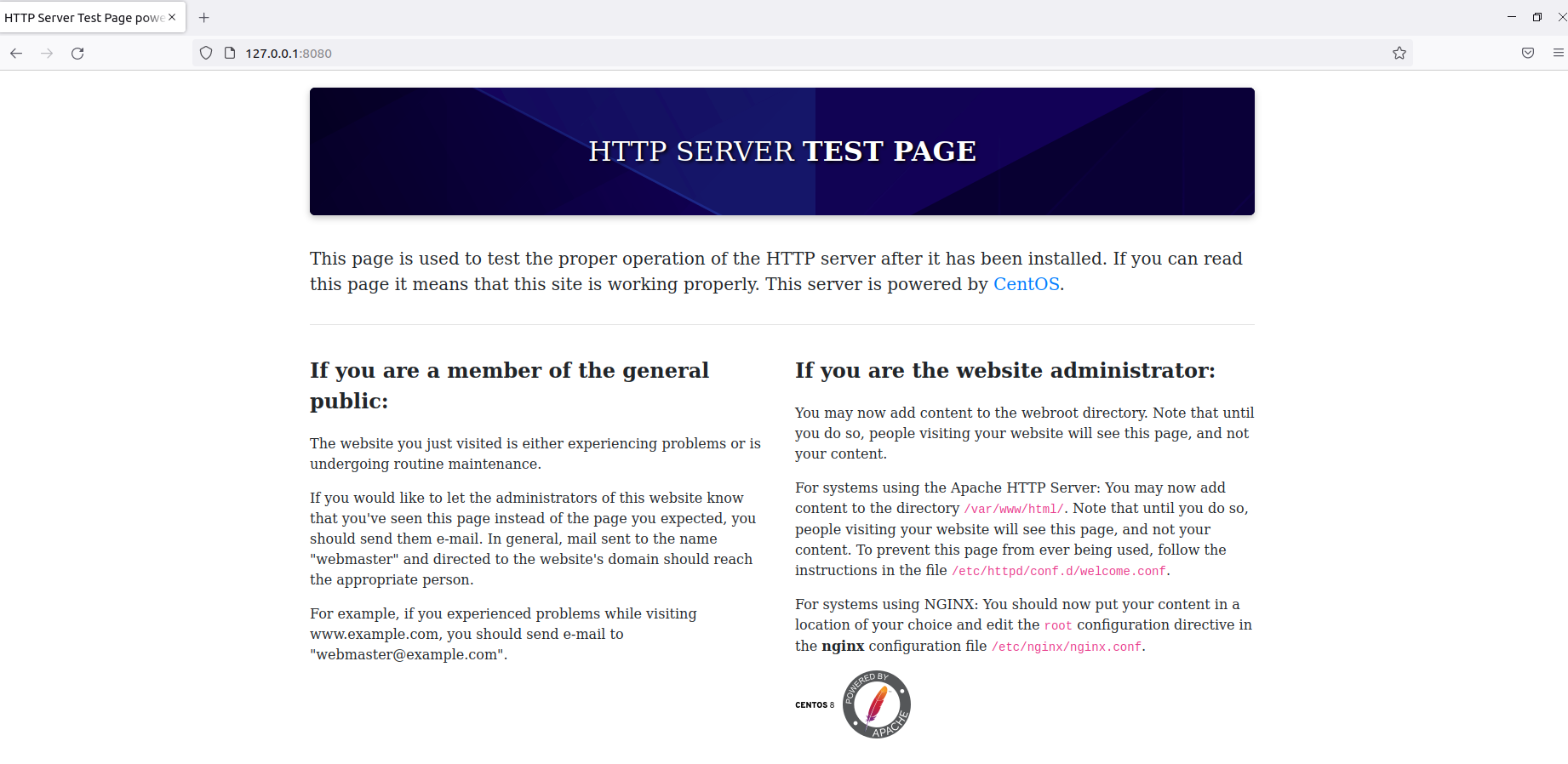
*#Разрешаем подключения со всех ip-адресов*

*Require all granted*

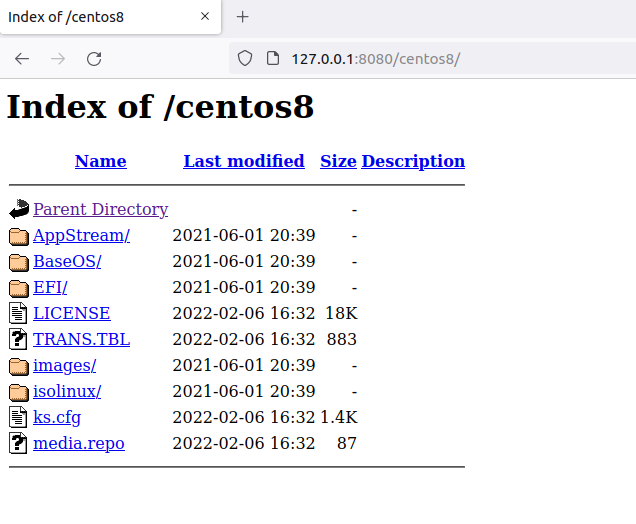
* Перезапускаем веб-сервер: *systemctl restart httpd*
* Добавляем его в автозагрузку: *systemctl enable httpd*

7. Проверяем, что веб-сервер работает и каталог /iso доступен по сети:

* С вашего компьютера сначала подключаемся к тестовой странице Apache:

Если страница открылась, значит веб-сервер запустился

* Далее проверям доступность файлов по сети:

Если файлы доступны, значит веб-сервер настроен корректно

**Пример настройки Веб-сервера в Ansible:**

#Начало файла provision.yml

**name**: Set up PXE Server

#Указываем имя хоста или группу, которые будем настраивать

**hosts**: pxeserver

#Параметр выполнения модулей от root-пользователя

**become**: **yes**

#Указание файла с дополнителыми переменными (понадобится при добавлении темплейтов)

**vars\_files**:

- defaults/main.yml

**tasks**:

#sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-Linux-\*

**- name**: set up repo

**replace**:

**path**: "{{ item }}"

**regexp**: 'mirrorlist'

**replace**: '#mirrorlist'

**with\_items**:

- /etc/yum.repos.d/CentOS-Linux-AppStream.repo

- /etc/yum.repos.d/CentOS-Linux-BaseOS.repo

#sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-Linux-\*

**- name**: set up repo

**replace**:

**path**: "{{ item }}"

**regexp**: '#baseurl=http://mirror.centos.org'

**replace**: 'baseurl=http://vault.centos.org'

**with\_items**:

- /etc/yum.repos.d/CentOS-Linux-AppStream.repo

- /etc/yum.repos.d/CentOS-Linux-BaseOS.repo

#Установка пакета httpd (дополнительно сразу ставятся все пакеты, которые потребуются в данном задании)

**- name**: install softs on CentOS

**yum**:

**name**:

- vim

- wget

- epel-release

- httpd

- tftp-server

- dhcp-server

**state**: present

**update\_cache**: **true**

#Скачивание образа CentOS-8.4.2105-x86\_64-dvd1.iso

***#Скачиваться будет долго, размер файла больше 9 ГБ.***

**- name**: Download ISO image CentOS 8.4.2105

**get\_url**:

**url**: https://mirror.sale-dedic.com/centos/8.4.2105/isos/x86\_64/CentOS-8.4.2105-x86\_64-dvd1.iso

**dest**: ~/CentOS-8.4.2105-x86\_64-dvd1.iso

**mode**: '0755'

#Создание каталога /iso и назначение прав 755

**- name**: Create ISO directory

**file**:

**path**: /iso

**state**: directory

**mode**: '0755'

#Монтируем образ в каталог /mnt

**- name**: Mount ISO image

**mount**:

**path**: /mnt

**src**: /root/CentOS-8.4.2105-x86\_64-dvd1.iso

**fstype**: iso9660

**opts**: ro,loop

**state**: mounted

#Копируем все файлы из каталога /mnt в /iso

**- name**: copy ALL files from /mnt to /iso

**copy**:

**src**: /mnt/

**dest**: /iso

**remote\_src**: **yes**

**directory\_mode**: **yes**

#Копируем конфгурационный файл pxeboot.conf (Файл должен быть предварительно создан в каталаге templates)

**- name**: set up httpd config

**template**:

**src**: pxeboot.conf

**dest**: /etc/httpd/conf.d/pxeboot.conf

**owner**: root

**group**: root

**mode**: 0640

#Перезупускаем httpd и добавляем службу в автозагрузку

**- name**: restart httpd

**service**:

**name**: httpd

**state**: restarted

**enabled**: **true**

На этом настройка веб-сервера завершена.

Настройка TFTP-сервера

TFTP-сервер потребуется для отправки первичных файлов загрузки (vmlinuz, initrd.img и т. д.)

1. Устанавливаем tftp-сервер: y*um install tftp-server*

2. Запускаем службу: *systemctl start tftp.service*

3. Проверяем, в каком каталоге будут храниться файлы, которые будет отдавать TFTP-сервер:

[root@pxeserver ~]# systemctl status tftp.service

● tftp.service - Tftp Server

**Loaded**: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor preset: disabled)

**Active**: active (running) since Sun 2022-02-06 20:53:28 UTC; 4s ago

**Docs**: man:in.tftpd

**Main PID**: 7732 (in.tftpd)

**Tasks**: 1 (limit: 4953)

**Memory**: 248.0K

**CGroup**: /system.slice/tftp.service

└─7732 /usr/sbin/in.tftpd **-s /var/lib/tftpboot**

**Feb 06 20**:53:28 pxeserver systemd[1]: Started Tftp Server.

[root@pxeserver ~]#

В статусе видим, что рабочий каталог **/var/lib/tftpboot**

4. Созаём каталог, в котором будем хранить наше меню загрузки:

*mkdir /var/lib/tftpboot/pxelinux.cfg*

5. Создаём меню-файл: vi */var/lib/tftpboot/pxelinux.cfg/default*

*default menu.c32*

*prompt 0*

*#Время счётчика с обратным отсчётом (установлено 15 секунд)*

*timeout 150*

*#Параметр использования локального времени*

*ONTIME local*

*#Имя «шапки» нашего меню*

*menu title OTUS PXE Boot Menu*

*#Описание первой строки*

*label 1*

*#Имя, отображаемое в первой строке*

*menu label ^ Graph install CentOS 8.4*

*#Адрес ядра, расположенного на TFTP-сервере*

*kernel /vmlinuz*

*#Адрес файла initrd, расположенного на TFTP-сервере*

*initrd /initrd.img*

*#Получаем адрес по DHCP и указываем адрес веб-сервера*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8*

*label 2*

*menu label ^ Text install CentOS 8.4*

*kernel /vmlinuz*

*initrd /initrd.img*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8 text*

*label 3*

*menu label ^ rescue installed system*

*kernel /vmlinuz*

*initrd /initrd.img*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8 rescue*

Label 1-3 различаются только дополнительными параметрами:

* label 1 — установка вручную в графическом режиме
* label 2 — установка вручную в текстовом режиме
* label 3 — восстановление системы

6. Распакуем файл syslinux-tftpboot-6.04-5.el8.noarch.rpm:

*rpm2cpio /iso/BaseOS/Packages/syslinux-tftpboot-6.04-5.el8.noarch.rpm | cpio -dimv*

7. После распаковки в каталоге пользователя root будет создан каталог tftpboot из которого потребуется скопировать следующие файлы:

- pxelinux.0

- ldlinux.c32

- libmenu.c32

- libutil.c32

- menu.c32

- vesamenu.c32

*cd tftpboot*

*cp pxelinux.0 ldlinux.c32 libmenu.c32 libutil.c32 menu.c32 vesamenu.c32 /var/lib/tftpboot/*

8. Также в каталог /var/lib/tftpboot/ нам потребуется скопировать файлы initrd.img и vmlinuz, которые располагаются в каталоге /iso/images/pxeboot/:

*cp /iso/images/pxeboot/{initrd.img,vmlinuz} /var/lib/tftpboot/*

9. Далее перезапускаем TFTP-сервер и добавляем его в автозагрузку:

*systemctl restart tftp.service*

*systemctl enable tftp.service*

**Настройка TFTP-сервера в Ansible:**

#Создаём каталог /var/lib/tftpboot/pxelinux.cfg

**- name**: Create TFTP directory

**file**:

**path**: ыныеу

**state**: directory

**mode**: '0755'

#Копируем файл-меню на сервер

**- name**: set up pxelinux

**template**:

**src**: default

**dest**: /var/lib/tftpboot/pxelinux.cfg/default

**owner**: root

**group**: root

**mode**: 0644

#Извлекаем файлы из RPM-пакета

**- name**: extract packages syslinux

**shell**: rpm2cpio /iso/BaseOS/Packages/syslinux-tftpboot-6.04-5.el8.noarch.rpm | cpio -dimv

#Копируем файлы в каталог /var/lib/tftpboot/

**- name**: copy files to TFTP share

**copy**:

**src**: /home/vagrant/tftpboot/{{ item }}

**dest**: /var/lib/tftpboot/{{ item }}

**mode**: '0644'

**remote\_src**: **true**

**with\_items**:

- pxelinux.0

- ldlinux.c32

- libmenu.c32

- libutil.c32

- menu.c32

- vesamenu.c32

#Копируем файлы в каталог /var/lib/tftpboot/

**- name**: copy initrd and vmlinuz files to TFTP share

**copy**:

**src**: /iso/images/pxeboot/{{ item }}

**dest**: /var/lib/tftpboot/{{ item }}

**mode**: '0755'

**remote\_src**: **true**

**with\_items**:

- initrd.img

- vmlinuz

#Перезапускаем TFTP-сервер и добавляем его в автозагрузку

**- name**: restart tftp-server

**service**:

**name**: tftp.service

**state**: restarted

**enabled**: **true**

**Настройка DHCP-сервера**

1. Устанавливаем DHCP-сервер: *yum install dhcp-server*

2. Правим конфигурационный файл: *vi /etc/dhcp/dhcpd.conf*

*option space pxelinux;*

*option pxelinux.magic code 208 = string;*

*option pxelinux.configfile code 209 = text;*

*option pxelinux.pathprefix code 210 = text;*

*option pxelinux.reboottime code 211 = unsigned integer 32;*

*option architecture-type code 93 = unsigned integer 16;*

*#Указываем сеть и маску подсети, в которой будет работать DHCP-сервер*

*subnet 10.0.0.0 netmask 255.255.255.0 {*

*#Указываем шлюз по умолчанию, если потребуется*

*#option routers 10.0.0.1;*

*#Указываем диапазон адресов*

*range 10.0.0.100 10.0.0.120;*

*class "pxeclients" {*

*match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";*

*#Указываем адрес TFTP-сервера*

*next-server 10.0.0.20;*

*#Указываем имя файла, который надо запустить с TFTP-сервера*

*filename "pxelinux.0";*

*}*

**Настройка DHCP-сервера в Ansible**

#Копирование файла конфигурации DHCP-сервера

**- name**: set up dhcp-server

**template**:

**src**: dhcpd.conf

**dest**: /etc/dhcp/dhcpd.conf

**mode**: '0644'

#Перезапуск службы и добавление в автозагрузку

**- name**: restart dhcp-server

**service**:

**name**: dhcpd

**state**: restarted

**enabled**: **true**

При копировании файла конфигурации можно использовать Jinja2 Template, который будет обращаться к переменным из файла defaults/main.yml

**Пример файла dhcpd.conf**

*option space pxelinux;*

*option pxelinux.magic code 208 = string;*

*option pxelinux.configfile code 209 = text;*

*option pxelinux.pathprefix code 210 = text;*

*option pxelinux.reboottime code 211 = unsigned integer 32;*

*option architecture-type code 93 = unsigned integer 16;*

*subnet* ***{{ dhcp\_network }}*** *netmask* ***{{ dhcp\_mask }}*** *{*

*range* ***{{ dhcp\_range\_min }} {{ dhcp\_range\_max }}****;*

*class "pxeclients" {*

*match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";*

*next-server* ***{{ pxe\_server }}****;*

*filename "pxelinux.0";*

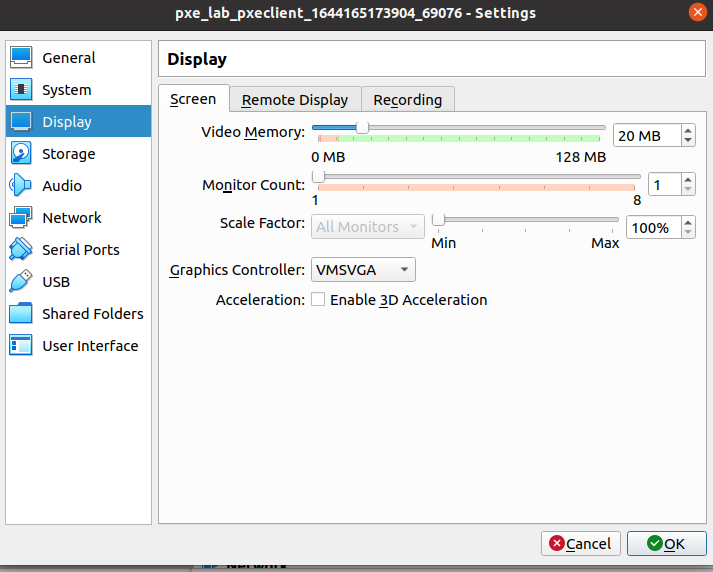
*}*

*}*

*Переменные, которые заключены в двойные скобки будут описаны в файле defaults/main.yml*

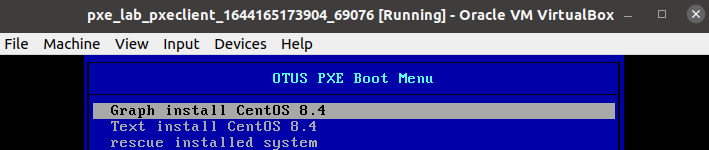
На данном этапе мы закончили настройку PXE-сервера для ручной установки сервера. Давайте попробуем запустить процесс установки вручную, для удобства воспользуемся установкой через графический интерфейс:

В настройках виртуальной машины pxeclient рекомендуется поменять графический контроллер на **VMSVGA** и добавить видеопамяти. **Видеопамять** должна стать **20 МБ** или больше.



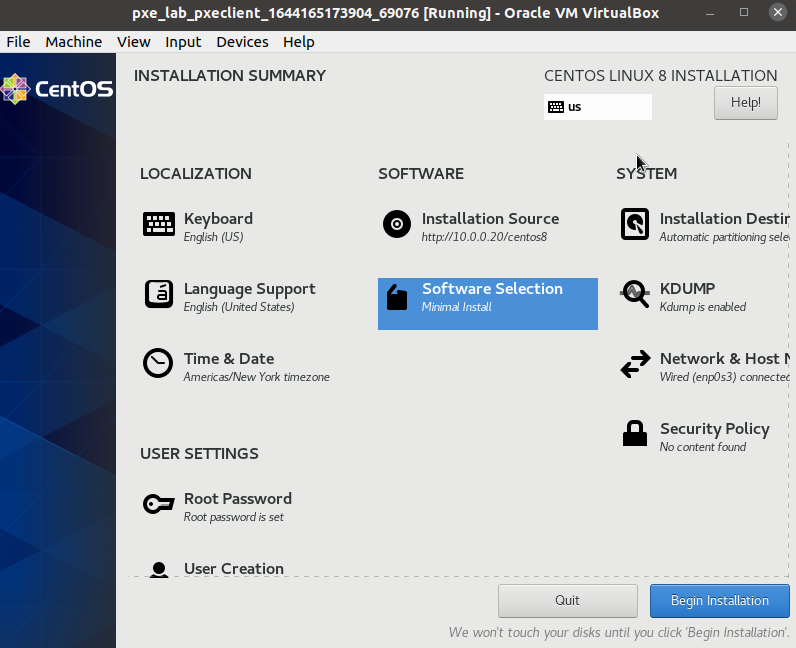
С такими настройками картинка будет более плавная и не будет постоянно мигать.

Нажимаем **ОК**, выходим из настроек ВМ и запускаем её.

Выбираем графическую установку

После этого, будут скачаны необходимые файлы с веб-сервера

Как только появится окно установки, нам нужно будет поочереди пройти по всем компонентам и указать с какими параметрами мы хотим установить ОС:



Иногда с разделом **Installation Source** случаются проблемы, и репозиторий не подтягивается автоматически. В этом случае нужно руками указать адрес репозитория: <http://10.0.0.20/centos8/BaseOS>

После установки всех, нужных нам параметров нажимаем **Begin installation**

После этого начнётся установка системы, после установки всех компонентов нужно будет перезагрузить ВМ и запуститься с диска.

Если нам не хочется вручную настраивать каждую установку, то мы можем автоматизировать этот процесс с помощью файла автоматиеской установки (kickstart file)

**Настройка автоматической установки с помощью Kickstart-файла**

1. Создаем kickstart-файл и кладём его в каталог к веб-серверу: *vi /iso/ks.cfg*

*#version=RHEL8*

*#Использование в установке только диска /dev/sda*

*ignoredisk --only-use=sda*

*autopart --type=lvm*

*#Очистка информации о партициях*

*clearpart --all --initlabel --drives=sda*

*#Использование графической установки*

*graphical*

*#Установка английской раскладки клавиатуры*

*keyboard --vckeymap=us --xlayouts='us'*

*#Установка языка системы*

*lang en\_US.UTF-8*

*#Добавление репозитория*

*url —url=http://10.0.0.20/centos8/BaseOS/*

*#Сетевые настройки*

*network --bootproto=dhcp --device=enp0s3 --ipv6=auto --activate*

*network --bootproto=dhcp --device=enp0s8 --onboot=off --ipv6=auto --activate*

*network --hostname=otus-pxe-client*

*#Устанвка пароля root-пользователю (Указан SHA-512 hash пароля 123)*

*rootpw --iscrypted $6$sJgo6Hg5zXBwkkI8$btrEoWAb5FxKhajagWR49XM4EAOfO/Dr5bMrLOkGe3KkMYdsh7T3MU5mYwY2TIMJpVKckAwnZFs2ltUJ1abOZ.*

*firstboot --enable*

*#Не настраиваем X Window System*

*skipx*

*#Настраиваем системные службы*

*services --enabled="chronyd"*

*#Указываем часовой пояс*

*timezone Europe/Moscow --isUtc*

*user --groups=wheel --name=val --password=$6$ihX1bMEoO3TxaCiL$OBDSCuY.EpqPmkFmMPVvI3JZlCVRfC4Nw6oUoPG0RGuq2g5BjQBKNboPjM44.0lJGBc7OdWlL17B3qzgHX2v// --iscrypted --gecos="val"*

*%packages*

*@^minimal-environment*

*kexec-tools*

*%end*

*%addon com\_redhat\_kdump --enable --reserve-mb='auto'*

*%end*

*%anaconda*

*pwpolicy root --minlen=6 --minquality=1 --notstrict --nochanges --notempty*

*pwpolicy user --minlen=6 --minquality=1 --notstrict --nochanges --emptyok*

*pwpolicy luks --minlen=6 --minquality=1 --notstrict --nochanges --notempty*

*%end*

2. Добавляем параметр в меню загрузки:

*vi /var/lib/tftpboot/pxelinux.cfg/default*

*default menu.c32*

*prompt 0*

*timeout 150*

*ONTIME local*

*menu title OTUS PXE Boot Menu*

*label 1*

*menu label ^ Graph install CentOS 8.4*

*kernel /vmlinuz*

*initrd /initrd.img*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8*

*label 2*

*menu label ^ Text install CentOS 8.4*

*kernel /vmlinuz*

*initrd /initrd.img*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8 text*

*label 3*

*menu label ^ rescue installed system*

*kernel /vmlinuz*

*initrd /initrd.img*

*append ip=enp0s3:dhcp inst.repo=http://10.0.0.20/centos8 rescue*

***label 4***

***menu label ^ Auto-install CentOS 8.4***

***#Загрузка данного варианта по умолчанию***

***menu default***

***kernel /vmlinuz***

***initrd /initrd.img***

***append ip=enp0s3:dhcp inst.ks=http://10.0.0.20/centos8/ks.cfg inst.repo=http://10.0.0.20/centos8/***

*В append появляется дополнительный параметр* ***inst.ks****, в котором указан адрес kickstart-файла.*

Если вы хотите сгенерировать хэш другого пароля, то сделать это можно с помощью команды: *python3 -c 'import crypt,getpass; print(crypt.crypt(getpass.getpass(), crypt.mksalt(crypt.METHOD\_SHA512)))'*

**Отправка kickstart-файла с помощью Ansible**

**- name**: copy ks.cfg

**template**:

**src**: ks.cfg

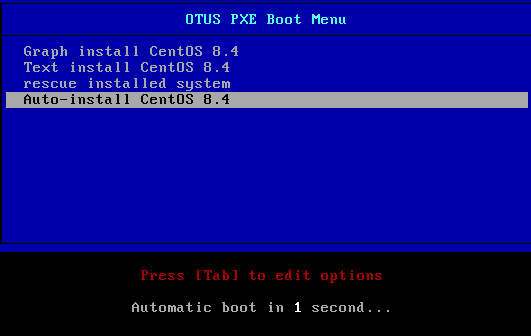
**dest**: /iso/ks.cfg

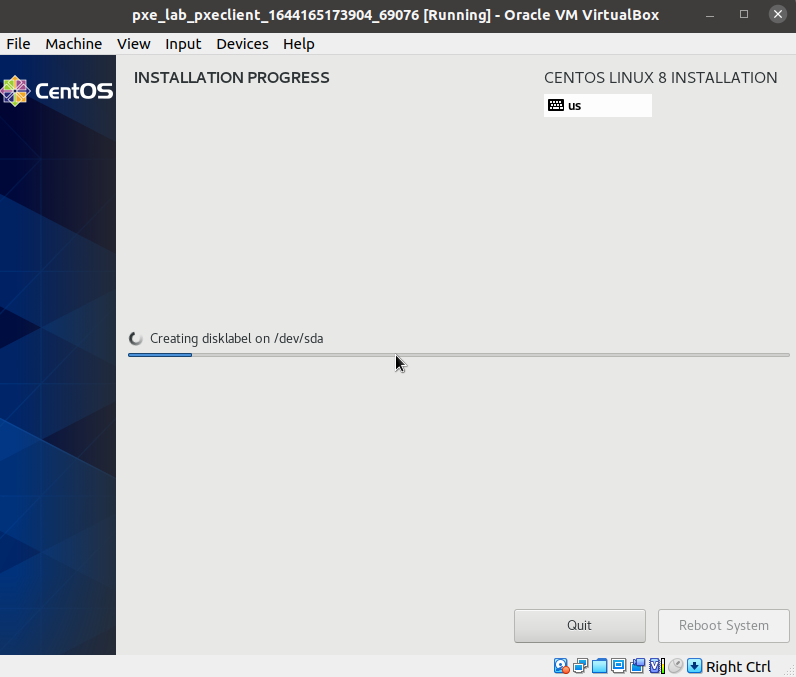
**owner**: root

**group**: root

**mode**: 0755

После внесения данных изменений, можем перезапустить нашу ВМ pxeclient и проверить, что запустится процесс автоматической установки ОС.





Для запуска Ansible сразу из Vagrant нужно добавить следующий код в описание ВМ pxeserver:

server**.**vm**.**provision "ansible" **do** **|**ansible**|**

ansible**.**playbook **=** "ansible/provision.yml"

ansible**.**inventory\_path **=** "ansible/hosts"

ansible**.**host\_key\_checking **=** "false"

ansible**.**limit **=** "all"

**end**

**Критерии оценивания**

Статус "Принято" ставится при выполнении следующих условий:

1. Ссылка на репозиторий github.

2. Vagrantfile с шагами установки необходимых компонентов

3. Исходный код для настройки сервера (Ansible-playbook)

4. Если какие-то шаги невозможно или сложно автоматизировать, то  инструкции по ручным шагам для настройки

Задание со звездочкой выполняется по желанию.

**Рекомендуемые источники**

* Статья о TFTP - https://ru.wikipedia.org/wiki/Trivial\_File\_Transfer\_Protocol
* Статья о DHCP - <https://ru.wikipedia.org/wiki/DHCP>
* Статья «Накатываем ось. Разбираемся с сервером установки в Cobbler» - <https://xakep.ru/2016/06/15/cobbler/>
* Оффициальная документация Cobbler - https://cobbler.readthedocs.io/en/latest/quickstart-guide.html
* Статья «PXE Boot : Configure PXE Server» - <https://www.server-world.info/en/note?os=CentOS_8&p=pxe&f=1>
* Статья «Configure IPv4 UEFI PXE Boot with Kickstart [RHEL/CentOS 8]» - <https://www.golinuxcloud.com/configure-uefi-pxe-boot-with-kickstart/>