

Администратор Linux

Курс об администрировании систем на базе Linux, который направлен на получение знаний и формирование навыков построения и обслуживания высоконадежных высокодоступных систем

Длительность курса: 212 академических часов

1 Архитектура Linux

1 С чего начинается Linux

после занятия вы сможете:
познакомиться с программой и группой;
объяснить, как работает ядро;
обновить ядро.

Домашние задания

1 Обновить ядро в базовой системе

Цель: Студент получит навыки работы с Git, Vagrant, Packer и публикацией готовых образов в Vagrant Cloud.

В материалах к занятию есть методичка, в которой описана процедура обновления ядра из репозитория. По данной методичке требуется выполнить необходимые действия. Полученный в ходе выполнения ДЗ Vagrantfile должен быть залит в ваш репозиторий. Для проверки ДЗ необходимо прислать ссылку на него.

Для выполнения ДЗ со * и ** вам потребуется сборка ядра и модулей из исходников.

2 Дискровая подсистема

после занятия вы сможете:
перечислить виды RAID массивов и их отличия;
получить информацию о дисковой подсистеме на любом сервере с ОС Linux;
собрать программный рейд и восстановить его после сбоя.

Домашние задания

1 Работа с mdadm

Цель: В результате выполнения ДЗ студент изменит Vagrantfile, создаст скрипт для создания рейда, конф для автосборки рейда при загрузке. Студент получил навыки работы с Vagrantfile.

добавить в Vagrantfile еще дисков
сломать/починить raid
собрать R0/R5/R10 на выбор
прописать собранный рейд в конф, чтобы рейд собирался при загрузке
создать GPT раздел и 5 партиций

в качестве проверки принимаются - измененный Vagrantfile, скрипт для создания рейда, конф для автосборки рейда при загрузке
* доп. задание - Vagrantfile, который сразу собирает систему с подключенным рейдом
** перенесети работающую систему с одним диском на RAID 1. Даунтайм на загрузку с нового диска предполагается. В качестве проверки принимается вывод команды lsblk до и после и описание хода решения (можно воспользоваться утилитой Script).

3 **Файловые системы и LVM**

после занятия вы сможете:
использовать LVM;
рассмотреть архитектуру файловой системы Linux:
суперблок, блоки, inodes, журналы;
разобрать многообразие файловых систем.

Домашние задания

1 Работа с LVM

Цель: Студент получил навыки работы с mirror, script.

на имеющемся образе
/dev/mapper/VolGroup00-LogVol00 38G 738M 37G
2% /

уменьшить том под / до 8G
выделить том под /home
выделить том под /var
/var - сделать в mirror
/home - сделать том для снэпшотов
прописать монтирование в fstab
попробовать с разными опциями и разными
файловыми системами (на выбор)
- сгенерить файлы в /home/
- снять снэпшот
- удалить часть файлов
- восстановится со снэпшота
- залогировать работу можно с помощью
утилиты script

* на нашей куче дисков попробовать поставить
btrfs/zfs - с кешем, снэпшотами - разметить здесь
каталог /opt

4 Загрузка системы

после занятия вы сможете:
оценить как происходит загрузка системы, в чем
разница между BIOS/UEFI;
настраивать GRUB2;
управлять initrd с помощью dracut;
использовать udev;
восстанавливать сломанный загрузчик.

Домашние задания

1 Работа с загрузчиком

Цель: Студент получил навыки работы с LVM.

1. Попасть в систему без пароля несколькими способами
2. Установить систему с LVM, после чего переименовать VG
3. Добавить модуль в initrd

4(*). Сконфигурировать систему без отдельного раздела с /boot, а только с LVM

Репозиторий с пропатченным grub:

https://yum.rumyantsev.com/centos/7/x86_64/

PV необходимо инициализировать с параметром
--bootloaderareasize 1m

5 Инициализация системы. Systemd и SysV.

после занятия вы сможете:
написать сценарии автозагрузки демонов;
оценить разницу между systemd и SysV;
обращаться с systemctl и journalctl.

Домашние задания

1 Systemd

Цель: В результате выполнения ДЗ студент запустит несколько экземпляров сервера с различными конфигурационными файлами. Студент получил навыки работы с Vagrant и Vagrant shell provisioner.

Выполнить следующие задания и подготовить развёртывание результата выполнения с использованием Vagrant и Vagrant shell provisioner (или Ansible, на Ваше усмотрение):

1. Написать service, который будет раз в 30 секунд мониторить лог на предмет наличия ключевого слова (файл лога и ключевое слово должны задаваться в /etc/sysconfig);
 2. Из репозитория epel установить spawn-fcgi и переписать init-скрипт на unit-файл (имя service должно называться так же: spawn-fcgi);
 3. Дополнить unit-файл httpd (он же apache) возможностью запустить несколько экземпляров сервера с различными конфигурационными файлами;
 - 4*. Скачать демо-версию Atlassian Jira и переписать основной скрипт запуска на unit-файл.
-

6 Bash, awk, sed, grep и другие

после занятия вы сможете:
писать скрипты на языке bash.

Домашние задания

1 Пишем скрипт

Цель: В результате выполнения ДЗ студент напишет скрипт.

написать скрипт для крона
который раз в час присылает на заданную почту
- X IP адресов (с наибольшим кол-вом запросов)
с указанием кол-ва запросов с момента
последнего запуска скрипта
- Y запрашиваемых адресов (с наибольшим кол-
вом запросов) с указанием кол-ва запросов с
момента последнего запуска скрипта
- все ошибки с момента последнего запуска
- список всех кодов возврата с указанием их кол-
ва с момента последнего запуска
в письме должно быть прописан
обрабатываемый временной диапазон
должна быть реализована защита от
мультизапуска

7 Управление пакетами. Дистрибьюция софта

после занятия вы сможете:
устанавливать софт в Linux;
сбирать из исходников.
использовать репозитории, yum и rpm;
оценить Docker как средство дистрибьюции,
преимущества и недостатки.

Домашние задания

1 Размещаем свой RPM в своем репозитории

Цель: В результате выполнения ДЗ студент создаст репо.

Студент получил навыки работы с RPM.

1) создать свой RPM (можно взять свое приложение, либо собрать к примеру апач с определенными опциями)

2) создать свой репо и разместить там свой RPM реализовать это все либо в вагранте, либо развернуть у себя через nginx и дать ссылку на репо

* реализовать дополнительно пакет через docker

8 Управление процессами

после занятия вы сможете:
оценить, что такое процесс, его атрибуты,
жизненный цикл процесса;
разобраться, чем потоки отличаются от процессов;
мониторить процессы, в каком они состоянии,
понимать чем они сейчас заняты;
проанализировать команды ps/top, подсистему /proc,
а также команды gdb/strace/ltrace;
менять приоритеты с помощью команд nice, ionice;
посылать различные сигналы процессам.

Домашние задания

1 Работаем с процессами

Цель: В результате выполнения ДЗ студент запустит скрипт, запускающий два процесса.

Задания на выбор

1) написать свою реализацию ps ах используя анализ /proc

- Результат ДЗ - рабочий скрипт который можно запустить

2) написать свою реализацию lsof

- Результат ДЗ - рабочий скрипт который можно запустить

3) дописать обработчики сигналов в прилагаемом скрипте, протестировать, приложить сам скрипт, инструкции по использованию

- Результат ДЗ - рабочий скрипт который можно запустить + инструкция по использованию и лог консоли

4) реализовать 2 конкурирующих процесса по IO. попробовать запустить с разными ioprio

- Результат ДЗ - скрипт запускающий 2 процесса с разными ioprio, измеряющий время выполнения и лог консоли

5) реализовать 2 конкурирующих процесса по CPU. попробовать запустить с разными nice

- Результат ДЗ - скрипт запускающий 2 процесса с разными nice и измеряющий время выполнения и лог консоли

2 Управление, безопасность и мониторинг

1 Docker

после занятия вы сможете:
писать Dockerfile;
создать docker-compose;
запустить docker swarm.

Домашние задания

1 Docker, docker-compose, dockerfile

Цель: Разобраться с основами docker, с образа, эко системой docker в целом.

Описание ДЗ в документе

2 Автоматизация администрирования. Ansible

после занятия вы сможете:
автоматизировать рутинные задачи администрирования;
разобрать ansible - инвентори, модули, плейбуки, роли, переменные;
проанализировать разницу с другими инструментами - chef/puppet/salt.

Домашние задания

1 Первые шаги с Ansible

Цель: В результате выполнения ДЗ студент подготовит стенд на Vagrant.

Подготовить стенд на Vagrant как минимум с одним сервером. На этом сервере используя Ansible необходимо развернуть nginx со следующими условиями:
- необходимо использовать модуль yum/apt
- конфигурационные файлы должны быть взяты из шаблона jinja2 с переменными

- после установки nginx должен быть в режиме enabled в systemd
- должен быть использован notify для старта nginx после установки
- сайт должен слушать на нестандартном порту - 8080, для этого использовать переменные в Ansible
- * Сделать все это с использованием Ansible роли

Домашнее задание считается принятым, если:

- предоставлен Vagrantfile и готовый playbook/роль (инструкция по запуску стенда, если посчитаете необходимым)
- после запуска стенда nginx доступен на порту 8080
- при написании playbook/роли соблюдены перечисленные в задании условия

3 SELinux - когда все запрещено

после занятия вы сможете:
разобраться, что такое SELinux.

4 Механизмы изоляции и аккаунтинга Linux (namespaces и cgroups)

после занятия вы сможете:
управлять и мониторить контейнеры без использования систем управления контейнерами.

5 Пользователи и группы. Авторизация и аутентификация

после занятия вы сможете:
рассмотреть механизмы авторизации и аутентификации;
оценить какие бывают права у пользователей;
управлять правами с помощью sudo, umask, sgid, suid и более сложными инструментами как PAM и ACL, PolicyKit.

Домашние задания

1 PAM

Цель: Студент получил навыки работы с PAM.

1. Запретить всем пользователям, кроме группы admin логин в выходные (суббота и воскресенье), без учета праздников
* дать конкретному пользователю права работать с докером
и возможность рестартить докер сервис

6 Мониторинг производительности

после занятия вы сможете:
мониторить занятые ресурсы: CPU, память, диск, сеть;
освоить инструменты ps, top, sar, htop, atop, netstat, ss, vmstat, iostat, iotop, pidstat;
проанализировать, что находится в /proc;
разобраться, что делать с неотзывчивой системой.

7 Сбор и анализ логов

после занятия вы сможете:

разобрать настройку логгирования с помощью rsyslog и logrotate;
использовать модную систему логгирования - ELK, graylog.

Домашние задания

1 Настраиваем центральный сервер для сбора логов

Цель: В результате выполнения ДЗ студент настроит центральный сервер для сбора логов.

в вагранте поднимаем 2 машины web и log
на web поднимаем nginx
на log настраиваем центральный лог сервер
на любой системе на выбор

- journald
- rsyslog
- elk

настраиваем аудит следящий за изменением конфигов nginx

все критичные логи с web должны собираться и локально и удаленно
все логи с nginx должны уходить на удаленный сервер (локально только критичные)

логи аудита должны также уходить на удаленную систему

* развернуть еще машину elk
и таким образом настроить 2 центральных лог системы elk И какую либо еще в elk должны уходить только логи nginx во вторую систему все остальное

8 Мониторинг и алертинг

после занятия вы сможете:
использовать Zabbix;
разобраться с Prometheus.

Домашние задания

1 Настройка мониторинга

Цель: В результате выполнения ДЗ студент настроит мониторинг.

Настроить дашборд с 4-мя графиками

- 1) память
- 2) процессор
- 3) диск
- 4) сеть

настроить на одной из систем

- zabbix (использовать screen (комплексный экран))
- prometheus - grafana

* использование систем примеры которых не рассматривались на занятии

- список возможных систем был приведен в презентации

в качестве результата прислать скриншот экрана - дашборд должен содержать в названии имя приславшего

9 Резервное копирование

после занятия вы сможете:
объяснить политики и методики резервного копирования;
работать с инструментами rsync, tar, dd и bacula.

Домашние задания

1 Настраиваем бэкапы

Цель: В результате выполнения ДЗ студент настроит стенд Vagrant с двумя виртуальными машинами.

Настроить стенд Vagrant с двумя виртуальными машинами server и client.

Настроить политику бэкапа директории /etc с клиента:

- 1) Полный бэкап - раз в день
- 2) Инкрементальный - каждые 10 минут
- 3) Дифференциальный - каждые 30 минут

Запустить систему на два часа. Для сдачи ДЗ приложить list jobs, list files jobid=<id> и сами конфиги bacula-*

* Настроить доп. Опции - сжатие, шифрование, дедупликация

1 Архитектура сетей

после занятия вы сможете:
рассмотреть Модели OSI, Протоколы ARP, IP, TCP/UDP, Протоколы прикладного уровня, сетевые интерфейсы в Linux;
освоить команды ip/tc/ss/nstat, вспомним ifconfig/netstat/route, разобраться в /etc/sysconfig/network-scripts, познакомиться через tcpdump и nmap.

Домашние задания

1 Разворачиваем сетевую лабораторию

Цель: В результате выполнения ДЗ студент развернет сетевую лабораторию.

otus-linux

Vagrantfile - для стенда урока 9 - Network

Дано

<https://github.com/erlong15/otus-linux/tree/network>

(ветка network)

Vagrantfile с начальным построением сети

- inetRouter

- centralRouter

- centralServer

тестировалось на virtualbox

Планируемая архитектура

построить следующую архитектуру

Сеть office1

- 192.168.2.0/26 - dev

- 192.168.2.64/26 - test servers

- 192.168.2.128/26 - managers

- 192.168.2.192/26 - office hardware

Сеть office2

- 192.168.1.0/25 - dev

- 192.168.1.128/26 - test servers

- 192.168.1.192/26 - office hardware

Сеть central

- 192.168.0.0/28 - directors

- 192.168.0.32/28 - office hardware

- 192.168.0.64/26 - wifi

...

Office1 ---\

-----> Central --IRouter --> internet

Office2----/

...

Итого должны получиться следующие сервера

- inetRouter

- centralRouter

- office1Router

- office2Router

- centralServer

- office1Server

- office2Server

Теоретическая часть

- Найти свободные подсети

- Посчитать сколько узлов в каждой подсети, включая свободные

- Указать broadcast адрес для каждой подсети

- проверить нет ли ошибок при разбиении

Практическая часть

- Соединить офисы в сеть согласно схеме и настроить роутинг

- Все сервера и роутеры должны ходить в инет через inetRouter

- Все сервера должны видеть друг друга

- у всех новых серверов отключить дефолт на

нат (eth0), который вагрант поднимает для связи

- при нехватке сетевых интерфейсов добавить по несколько адресов на интерфейс

2 Фильтрация трафика

после занятия вы сможете:

настраивать фаерволл с использованием iptables/firewalld;

настраивать NAT;

пробрасывать порты;

настраивать взаимодействие с роутингом;

понимать работу таблиц и цепочек.

Домашние задания

1 Сценарии iptables

Цель: Студент получил навыки работы с centralServer, inetRouter.

1) реализовать knocking port

- centralRouter может попасть на ssh

inetrRouter через knock скрипт

пример в материалах

2) добавить inetRouter2, который

виден(маршрутизируется (host-only тип сети

для виртуалки)) с хоста или форвардится

порт через локалхост

3) запустить nginx на centralServer

4) пробросить 80й порт на inetRouter2 8080

5) дефолт в инет оставить через inetRouter

* реализовать проход на 80й порт без маскардинга

3 **LDAP. Централизованная авторизация и аутентификация**

после занятия вы сможете:
объяснить, что такое LDAP и зачем нужен;
разобрать базовую настройку LDAP на примере.

Домашние задания

1 LDAP

Цель: В результате выполнения ДЗ студент установит FreeIPA.

1. Установить FreeIPA;
2. Написать Ansible playbook для конфигурации клиента;
- 3*. Настроить аутентификацию по SSH-ключам;
- 4**. Firewall должен быть включен на сервере и на клиенте.

В git - результирующий playbook.

4 Статическая и динамическая маршрутизация

после занятия вы сможете:
понимать принцип работы multicast;
настраивать статическую и динамическую маршрутизацию;
понимать принципы работы протоколов OSPF BGP;
настраивать динамическую маршрутизацию с помощью quagga.

Домашние задания

1 OSPF

Цель: Студент получил навыки работы с OSPF.

- Поднять три виртуалки
- Объединить их разными vlan
- 1. Поднять OSPF между машинами на базе Quagga
- 2. Изобразить ассиметричный роутинг
- 3. Сделать один из линков "дорогим", но что бы при этом роутинг был симметричным

Формат сдачи:
Vagrantfile + ansible

5 DNS/DHCP - настройка и обслуживание

после занятия вы сможете:
настроить DHCP;
объяснить как завести домен;
управлять зонами (bind/powerdns);
обслуживать свой домен самостоятельно;
разобрать dig/host/nslookup.

Домашние задания

1 Настраиваем split-dns

Цель: В результате выполнения ДЗ студент настроит split-dns.

взять стенд <https://github.com/erlong15/vagrant-bind>

добавить еще один сервер client2

завести в зоне dns.lab

имена

web1 - смотрит на клиент1

web2 смотрит на клиент2

завести еще одну зону newdns.lab

завести в ней запись

www - смотрит на обоих клиентов

настроить split-dns

клиент1 - видит обе зоны, но в зоне dns.lab

только web1

клиент2 видит только dns.lab

*) настроить все без исключения selinux

6 Мосты, туннели и VPN

после занятия вы сможете:
объяснить, что такое мосты и туннели;
настраивать мосты и туннели между сетями;
разбираться в различных типах реализаций мостов и туннелей.

Домашние задания

1 VPN

Цель: Студент получил навыки работы с VPN, RAS.

1. Между двумя виртуалками поднять vpn в режимах

- tun

- tap

Прочувствовать разницу.

2. Поднять RAS на базе OpenVPN с клиентскими сертификатами, подключиться с локальной машины на виртуалку

3*. Самостоятельно изучить, поднять oserv и подключиться с хоста к виртуалке

7 Сетевые пакеты. VLAN'ы. LACP

после занятия вы сможете:
использовать UniCast/MultiCast/BroadCast/AnyCast;
проанализировать протокол LACP;
агрегировать интерфейсы через teaming и bonding;
объяснить, что такое VLAN. Знакомимся с dot1q, macvlan;
освоить работу с nmcli.

Домашние задания

1 Строим бонды и вланы

в Office1 в тестовой подсети появляется сервера с доп интерфейсами и адресами в internal сети testLAN

- testClient1 - 10.10.10.254
- testClient2 - 10.10.10.254
- testServer1- 10.10.10.1
- testServer2- 10.10.10.1

равести вланами

testClient1 <-> testServer1

testClient2 <-> testServer2

между centralRouter и inetRouter

"пробросить" 2 линка (общая internal сеть) и объединить их в бонд

проверить работу с отключением интерфейсов

для сдачи - вагрант файл с требуемой конфигурацией

Разворачиваться конфигурация должна через ансибл

8 IPv6

после занятия вы сможете:

- объяснить как работает протокол Ipv6;
- применить правила формирования адреса;
- отличить типы адресов;
- оценить особенности работы и преимущества использования;
- настраивать маршрутизацию и файрвалл.

1 Web сервера

после занятия вы сможете:
объяснить протоколы HTTP/HTTPS, HTTP 2.0;
разобраться с SSL;
устанавливать и настраивать Apache и Nginx;
настраивать vhosts.

Домашние задания

1 Простая защита от DDOS

Цель: Разобраться в базовых принципах конфигурирования nginx. Рассмотреть хорошие\плохие практики конфигурирования, настройки ssl.

https://gitlab.com/otus_linux/nginx-antiddos-example

Следуйте инструкциям из README

2 PostgreSQL

после занятия вы сможете:
администрировать PostgreSQL;
установить, настроить, создать пользователей и роли;
выдать права, создать базы;
мониторить, делать бекапы.

Домашние задания

1 PostgreSQL

Цель: Студент получил навыки работы chot_standby.

- Настроить hot_standby репликацию с использованием слотов
- Настроить правильное резервное копирование

Для сдачи работы присылаем ссылку на репозиторий, в котором должны обязательно быть Vagrantfile и плейбук Ansible, конфигурационные файлы postgresql.conf, pg_hba.conf и recovery.conf, а так же конфиг barman, либо скрипт резервного копирования. Команда "vagrant up" должна поднимать машины с настроенной репликацией и резервным копированием. Рекомендуется в README.md файл вложить результаты (текст или скриншоты) проверки работы репликации и резервного копирования.

3 **Динамический веб контент**

после занятия вы сможете:
разбираться в CGI/FCGI/WSGI/mod_xxx;
настраивать uwsgi/php_fpm;
понимать Ajax/Long Pooling/Websockets.

Домашние задания

1 Роль для настройки веб сервера

Варианты стенда

nginx + php-fpm (laravel/wordpress) + python
(flask/django) + js(react/angular)
nginx + java (tomcat/jetty/netty) + go + ruby
можно свои комбинации

Реализации на выбор

- на хостовой системе через конфиги в /etc
- деплой через docker-compose

Для усложнения можно попросить проекты у коллег с курсов по разработке

К сдаче принимается

vagrant стэнд с проброшенными на локалхост портами
каждый порт на свой сайт
через нжинкс

4 **PostgreSQL cluster**

после занятия вы сможете:
настраивать кластер PostgreSQL.

5 Mysql

после занятия вы сможете:
устанавливать, настраивать, тестировать mysql;
понимать, на что влияют различные настройки;
понимать, как работает СУБД;
понимать какие файлы и их назначение
используются mysql;
настраивать базовую репликацию.

Домашние задания

- 1 Развернуть базу из дампа и настроить репликацию

Цель: В результате выполнения ДЗ студент развернет базу из дампа и настроит репликацию.

В материалах приложены ссылки на вагрант для репликации

и дампы базы bet.dmp

базу развернуть на мастере

и настроить чтобы реплицировались таблицы

| bookmaker |

| competition |

| market |

| odds |

| outcome

* Настроить GTID репликацию

варианты которые принимаются к сдаче

- рабочий вагрантафайл

- скрины или логи SHOW TABLES

* конфиги

* пример в логе изменения строки и появления строки на реплике

**6 MySQL - бэкап,
репликация,
кластер**

после занятия вы сможете:
настраивать политику инкрементального бэкапа;
настраивать GTID репликацию и кластер с proxysql;
настраивать Innodb и Galera кластера.

Домашние задания

1 Развернуть InnoDB кластер в docker

Цель: В результате выполнения ДЗ студент развернет InnoDB кластер в docker.

развернуть InnoDB кластер в docker
* в docker swarm

в качестве ДЗ принимает репозиторий с docker-compose
который по кнопке разворачивает кластер и
выдает порт наружу

**7 Redis,
Memcached,
RabbitMQ**

после занятия вы сможете:
объяснить, что это такое и зачем нужны эти сервисы;
устанавливать настраивать их;
запускать в работу.

1 **Файловые хранилища - NFS, SMB, FTP**

после занятия вы сможете:
строить файловое хранилище на основе Linux.

Домашние задания

1 Vagrant стенд для NFS или SAMBA

Цель: В результате выполнения ДЗ студент получит Vagrant стенд для NFS или SAMBA.

NFS или SAMBA на выбор:

vagrant up должен поднимать 2 виртуалки:
сервер и клиент
на сервер должна быть расшарена директория
на клиента она должна автоматически монтироваться при старте (fstab или autofs)
в шаре должна быть папка upload с правами на запись
- требования для NFS: NFSv3 по UDP,
включенный firewall

* Настроить аутентификацию через
KERBEROS

2 **Балансировка нагрузки на основе HAProxy и VRRP (keepalived)**

после занятия вы сможете:
оценить протокол VRRP;
объяснить работу с haproxy;
настраивать балансировку для вебсерверов;
разбирать конфиги;
проверять работу балансировки на стенде.

3 **Почта: SMTP, IMAP, POP3**

после занятия вы сможете:
разбирать почтовые протоколы;
устанавливать и настраивать Postfix и Dovecot;
оценить настройки безопасности.

Домашние задания

1 Установка почтового сервера

Цель: В результате выполнения ДЗ студент установит почтовый сервер.

1. Установить в виртуалке postfix+dovecot для приёма почты на виртуальный домен любым обсужденным на семинаре способом
2. Отправить почту телнетом с хоста на виртуалку
3. Принять почту на хост почтовым клиентом

Результат

1. Полученное письмо со всеми заголовками
2. Конфиги postfix и dovecot

Всё это сложить в git, ссылку прислать в "чат с преподавателем"

4 **Строим кластер - Pacemaker, Corosync**

после занятия вы сможете:
оценить устройства кластера на основе pacemaker;
построить тестовый кластер с плавающим ип и вебсервисом.

5 **ISCSI, multipath и кластерные файловые системы: GFS2**

после занятия вы сможете:
разобрать принципы работы ISCSI;
разобраться, что такое multipath;
настроить общее хранилище для двух нод;

6 **Распределенные файловые системы: CEPH**

после занятия вы сможете:
оценить устройство распределенного хранилища CEPH и требования для его развертывания;
получить практический опыт создания кластера с нуля, его использования с точки зрения клиентского доступа;
протестировать отказоустойчивость, надежность и скорость работы.

7 **Распределенные файловые системы: GlusterFS**

после занятия вы сможете:
построить распределенное хранилище GlusterFS.

8 **Виртуализация**

после занятия вы сможете:
разбираться в технологиях виртуализации, чтобы в нужный момент; сделать правильный выбор инструмента для решения инфраструктурных задач.

9 **Защита проектных работ**

защитить проект и получить рекомендации от экспертов

Домашние задания

1 Проектная работа

Заключительный месяц курса посвящен проектной работе. Свой проект это то, что интересно писать студенту. То, что можно создать на основе знаний, полученных на курсе.

При этом не обязательно закончить его за месяц. В процессе написания по проекту можно получить консультации преподавателей.

10 **Подведение итогов курса**

узнать, как получить сертификат об окончании курса, как взаимодействовать после окончания курса с OTUS и преподавателями, какие вакансии и позиции есть для выпускников (опционально - в России и за рубежом) и на какие компании стоит обратить внимание.