OTUS

help@otus.ru

Общая стоимость

60 000 руб.

В месяц: 15 000 руб.

Администратор Linux

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

Продолжительность

5 месяцев, 4 часа в неделю

Начало занятий

23 апреля

1 Архитектура Linux

1 **С чего начинается Linux**

Знакомство.

Обзор базовых инструментов, которые понадобятся в течении курса - ssh и его клиенты, vagrant, git.

Какие версии LInux бывают.
Ядро Linux. Краткое введение. Syscalls.
Версии ядра. Обновление ядра. Ручная сборка ядра.
Модули ядра. Команды: modprobe, lsmod, rmmod.

Лабораторная работа. Установка с образа, обновление ядра, включение/выключение модулей.

Домашние задания

1 Делаем собственную сборку ядра

Взять любую версию ядра с kernel.org

Подложить файл конфигурации ядра
Собрать ядро (попутно доставляя необходимые пакеты)
Прислать результирующий файл конфигурации
Прислать списк доустановленных пакетов, взять его можно из /var/log/yum.log
Устанавливать будем на следующем занятии =)

---

2 **Дисковая подсистема**

Программный и аппаратный RAID. Получение информации о дисковой системе системе с помощью dmidecode, dmesg, smartctl.

MBR и GPT. Команды gdisk/fdisk/parted/partprobe.

Лабораторная работа: управление рейд массивом с помощью mdadm: создание, съем информации. Разбиваем на партиции.

Домашние задания

1 работа с mdadm.

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 проверики принимается вывод команды lsblk до и после и описание хода решения (можно воспользовать утилитой Script).

3	**Файловые системы и LVM**	LVM - облегчаем себе жизнь управления файловыми системами. архитектура файловой системы Linux: суперблок, блоки, inodes, журналы. разбираемся в многообразии файловых систем: ext2/3/4, xfs, raiserfs, btrfs, zfs, cephfs Лабораторная работа: создаем и меняем размеры томов LVM и файловых систем. Знакомимся с mount, mkfs, fsck, resize2fs, /etc/fstab Домашние задания

1 Работа с LVM

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
- залоггировать работу можно с помощью
утилиты script

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

4 **Bash, awk, sed, grep и другие**

Изучаем основные рабочие инструменты системного администратора. Базовое программирование. Переменные, условия, циклы, однострочники. Знакомимся с командами интерпретатора bash. Знакомимся с командами awk/sed/grep/egrep/cut/find/sort/uniq и другими

Лабораторная работа: пишем скрипт

Домашние задания

1 Пишем скрипт

Подготовить свои скрипты для решения как минимум одного из следующих кейсов:
1) watchdog с перезагрузкой процесса/сервиса
2) watchdog с отсылкой емэйла
3) анализ логов веб сервера/security лога - (на взлом/скорость ответа/выявление быстрых - медленных запросов, анализ IP адресов и кол-ва запросов от них)
4) крон скрипт с защитой от мультизапуска
5) любой скрипт на ваше усмотрение

Желательно чтобы в скрипте были:
циклы
условия
регекспы
awk
наличие в скрипте трапов и функций

5 **Управление процессами**

Рассмотрим, что такое процесс, его атрибуты, жизненный цикл процесса.
Чем потоки отличаются от процессов.
Узнаем как мониторить процессы, в каком они состоянии, понимать чем они сейчас заняты.
Рассмотрим команды ps/top, подсистему /proc, а также команды gdb/strace/ltrace
Научимся менять приоритеты с мощью команд nice, ionice
Научимся посылать различные сигналы процессам.

Домашние задания

1 работаем с процессами

Задания на выбор
1) написать свою реализацию ps ax 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, оттестировать, 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. пробовать запустить с разными ionice
- Результат ДЗ - скрипт запускающий 2 процесса с разными ionice, замеряющий время выполнения и лог консоли
5) реализовать 2 конкурирующих процесса по CPU. пробовать запустить с разными nice
- Результат ДЗ - скрипт запускающий 2 процесса с разными nice и замеряющий время выполнения и лог консоли

6 **Управление пакетами. Дистрибьюция софта.**

Как устанавливать софт в Linux. Как собирать из исходников. Репозитории, yum и rpm.
Docker как средство дистрибьюции, преимущества и недостатки.

Лабораторная работа: Будем настраивать собственные репозитории и создавать собственные rpm'ки.

Домашние задания

1 Размещаем свой RPM в своем репозитории

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

---

7 **Загрузка системы**

Как происходит загрузка системы. В чем разница между BIOS/UEFI. Знакомимся GRUB2 и учимся его настраивать. Управляем initrd с помощью dracut. Знакомимся с udev. Учимся восстанавливать сломанный загрузчик

Лабораторная работа: прописываем в grub несколько конфигураций с разными ядрами. Ставим хук в initrd

Домашние задания

1 Работа с загрузчиком

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 пропатченым grub: https://yum.rumyantsev.com/centos/7/x86_64/
PV необходимо инициализировать с параметром --bootloaderareasize 1m

8 **Инициализация системы. Systemd и SysV.**

Учимся писать сценарии автозагрузки демонов. Изучаем разницу между systemd и SysV. учимся обращаться с systemctl и journalctl.

Лабораторная работа: сценарии автозапуска под systemd и Sys.V

Домашние задания

1 Systemd

1. Написать сервис, который будет раз в 30 секунд мониторить лог на предмет наличия ключевого слова. Файл и слово должны задаваться в /etc/sysconfig
2. Из epel установить spawn-fcgi и переписать init-скрипт на unit-файл. Имя сервиса должно так же называться.
3. Дополнить юнит-файл apache httpd возможностьб запустить несколько инстансов сервера с разными конфигами
4*. Скачать демо-версию Atlassian Jira и переписать основной скрипт запуска на unit-файл

2 Управление, безопасность и мониторинг

1 **Автоматизация администрирования. Ansible.**

Автоматизируем рутинные задачи администрирования. Изучаем ansible - хосты, модули, плейбуки, роли, переменные. Знакомися с другими инструментами - chef/puppet/salt.

Лабораторная работа: пишем скрипт для апгрейда системы после установки и измения конфигов.

Домашние задания

1 Первые шаги с Ansible

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 перемененными
- 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

---

2 **Пользователи и группы. Авторизация и аутентификация**

рассмотрим механизмы авторизации и аутентификации. Узнаем какие бывают права у пользовталей. Научимся управлять правами с помощью sudo, umask. sgid, suid и более сложными инструментами как PAM и ACL, PolicyKit

Лабораторная работа: даем пользователю A возможность запускать скрипт, принадлежащий пользователю B

Домашние задания

1 PAM

1. Запретить всем пользователям, кроме группы admin логин в выходные и праздничные дни
2. Дать конкретному пользователю права рута

---

3 **LDAP. Централизованная авторизация и аутентификация.**

Что такое LDAP и зачем нужен. Разбираем базовую настройку LDAP на примере.

Домашние задания

1 LDAP

1. Установить FreeIPA
2. Написать playbook для конфигурации клиента
3*. Настроить авторизацию по ssh-ключам

В git - результирующий playbook

4	**Мониторинг производительности**	Мониторим занятые ресурсы: CPU, память, диск, сеть. Изучаем инструменты ps, top , sar, htop, atop, netstat, ss, vmstat, iostat, iotop, pidstat Смотрим, что находится в /proc Узнаем, что делать с неотзывчивой системой Домашние задания 1 Linux Troubleshooting 1. Написать playbook для первоначальной настройки хоста после инсталляции по всем прошедшим лекциям - установка нужных инструментов для анализа и траблшутинга - избавление ядра и сетевых настроек от "десктопности" - установка разнообразных параметров ядра под работу в качестве сервера 2. Опубликовать ссылку на плейбук в общем чатике 3. Взять любой опубликованный плейбук, прокомментировать.
5	**SELinux - когда все запрещено.**	Разбираемся, что такое SELinux

6 **Сбор и анализ логов.**

Разбираем настройку логгирования с помощью rsyslog и logrotate.
Знакомимся с модными система логгирования - ELK, graylog

Домашние задания

1 Настраиваем 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 нжинкса

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 уходят ТОЛЬКО 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 нжинкса
во вторую систему все остальное

---

7 **Мониторинг и алертинг**

Изучаем Zabbix. Знакомимся с Prometheus

Домашние задания

1 Настройка мониторинга

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

8 **Резервное копирование.**

Обсуждаем политики и методики резерного копирования. Работаем с инструментами rsync, tar, dd и bacula.

Домашние задания

1 Настраиваем бэкапы

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

3 Linux и сеть

1 **Архитектура сетей.**

Обзор Модели OSI. Протоколы ARP, IP, TCP/UDP. Протоколы прикладного уровня. Сетевые интерфейсы в Linux.
Освоим команды ip/tc/ss/nstat, вспомним ifconfig/netstat/route, заглянем в /etc/sysconfig/network-scripts, поснифферим через tcpdump и ngrep

Лабораторная работа: строим маршрутизацию между подсетями

Домашние задания

1 разворачиваем 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 получится 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 черз inetRouter
- Все сервера должны видеть друг друга
- у всех новых серверов отключить дефолт на нат (eth0), который вагрант поднимает для связи
- при нехватке сетевых интервейсов добавить по несколько адресов на интерфейс

2	**DNS/DHCP - настройка и обслуживание**	Настраиваем DHCP Узнаем как завести домен Как управлять зонами (bind/powerdns) Как обслуживать свой домен самостоятельно Разбираем dig/host/nslookup Лабораторная работа: настраиваем свой кеширующий днс (мастер/слейв) со своей локальной зоной Домашние задания 1 настраиваем 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 выключения selinux

3 **Фильтрация трафика**

Углубляемся в iptables/firewalld
разбираем цепочки и таблицы
учимся правильно защищать свою сеть
строим NAT, проксируем трафик, пробрасываем порты
Лабораторная работа: защищаем веб сервер от DOS атак

Домашние задания

1 Сценарии iptables

1) реализовать knocking port
- centralRouter может попасть на ssh inetrRouter через knock скрипт
пример в материалах
2) добавить inetRouter2, который виден(маршрутизируется) с хоста
3) запустить nginx на centralServer
4) пробросить 80й порт на inetRouter2 8080
5) дефолт в инет оставить через inetRouter

4 **Мосты, туннели и VPN**

Разбираемся в терминах и протоколах - что такое мосты, туннели, VPN, PPP, PPTP, PPoE, IPoE, GRE, IPIP, IpSec, L2TP.
Строим VPN между линуксами, разбираем нюансы подключения к Cisco и Mikrotik.
Лабораторная работа: VPN через openvpn

Домашние задания

1 VPN

1. Между двумя виртуалками поднять vpn в режимах
- tun
- tap
Прочуствовать разницу.

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

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

5 **Сетевые пакеты. VLAN'ы. LACP.**

Изучаем UniCast/MultiCast/BroadCast/AnyCast.
Изучаем протокол LACP. Учимся аггрегировать интерфейсы через teaming и bonding.
Разбираемся что такое VLAN. Знакомимся с dot1q, macvlan
Осваиваем работу с nmcli
Лабораторная работа: аггрегируем интерфейсы в режиме active/active и failover

Домашние задания

1 строим бонды и вланы

в Office1 в тестовой подсети появляется сервера
с доп интерфесами 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 inernal сеть) и объединить их в бонд
проверить работу с отключением интерфейсов

для сдачи - вагрант файл с требуемой конфигурацией
Разворачиваться конфигурация должна через ансибл

6 **Статическая и динамическая маршрутизация**

настраиваем простые маршруты с помощью route/ip/nmcli
разбираем что такое RIP/OSPF/BGP
настраиваем динамическую маршрутизацию с помощью bird и quagga
Лабораторная работа: настройка OSPF между 3мя сетями

Домашние задания

1 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

4 Сервисы на базе Linux

1 **Web сервера**

Изучаем протоколы HTTP/HTTPS, HTTP 2.0
Разбираемся с SSL
Устанавливаем и настраиваем Apache и Nginx
Настраиваем vhosts

Домашние задания

1 Простая защита от DDOS

Написать конфигурацию nginx, которая даёт доступ клиенту только с определенной cookie.
Если у клиента её нет, нужно выполнить редирект на location, в котором кука будет добавлена, после чего клиент будет обратно отправлен (редирект) на запрашиваемый ресурс.

Смысл: умные боты попадаются редко, тупые боты по редиректам с куками два раза не пойдут

Для выполнения ДЗ понадобятся
https://nginx.org/ru/docs/http/ngx_http_rewrite_module.html
https://nginx.org/ru/docs/http/ngx_http_headers_module.html

---

2 **Динамический веб контент**

разбираем CGI/FCGI/WSGI/mod_xxx
настраиваем uwsgi/php_fpm
разбираемся с python/perl/php/ruby

3	**Почта: SMTP, IMAP, POP3**	разбираем почтовые протоколы. Устанавливаем и настраиваем Postfix и Dovecot Домашние задания 1 установка почтового сервера 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	**Почта: безопасность и другие задачи**	

5	**PostgreSQL**	Учимся администрировать PostgreSQL Установка, настройка, создаем пользователей и роли выдаем права, создаем базы, мониторим, делаем бэкапы Домашние задания 1 PostgreSQL - Настроить hot_standby репликацию с использованием слотов - Настроить правильное резервное копирование Для сдачи присылаем postgresql.conf, pg_hba.conf и recovery.conf А так же конфиг barman, либо скрипт резервного копирования
6	**PostgreSQL cluster**	настраиваем кластер PostgreSQL

7 **Mysql**

Учимся администрировать mysql
Устанавливаем, запускаем, мониторим
Создаем схемы, делаем простые запросы
Учимся делать бэкап и репликацию

Домашние задания

1 развернуть базу из дампа и настроить репликацию

В материалах приложены ссылки на вагрант для репликации
и дамп 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

8	**MySQL - бэкап, репликация, кластер**	Настраиваем политику инкрементального бэкапа Настраиваем GTID репликацию и кластер с proxysql Настраиваем NDB кластер Домашние задания 1 простая связь через sip/pjsip установить астериск на сервере для ус тановки воспользоваться ролью https://github.com/erlong15/tls-asterisk14-ansible при установке создаются 3 номер 1100, 1101, 1102 подключить два телефона (можно использовать transport-tls, transport-udp, transport-tcp) сделать звонок в качестве ДЗ принимается лог SIP сессии при использовании вагранта и внешних телефонов можно использовать варианты - виртуалки с телефонами в одной приват сети с астериском - бридж интерфейс для проброса во вне
9	**Redis, Memcached, RabbitMQ**	Разбираем что это такое и зачем нужны эти сервисы. Устанавливаем и настраиваем их Запускаем в работу.

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

Строим файловое хранилище на основе Linux

Домашние задания

1 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

5 Проектная работа

1	**Балансировка нагрузки на основе HAproxy и VRRP (keepalived)**	Изучаем протокол VRRP Изучаем работу с haproxy Настраиваем балансировку для вебсерверов Разбираем конфиги Проверяем работу балансировки на стенде
2	**строим кластер - Pacemaker, Corosync, Heartbeat**	
3	**ISCSI, multipath и кластерные файловые системы: GFS2**	настраиваем ISCSI настраиваем multipath Лабораторная работа: разбираем работу системы на стенде, 2 таргета, 2 клиента, общая файловая система
4	**Распределенные файловые системы: CEPH**	
5	**Распределенные файловые системы: GlusterFS**	
6	**Контейнеры - cgroups, LXC, docker**	

7	**Docker**	разбираем как писать Dockerfile создаем docker-compose запускаем docker swarm
8	**Итоговое занятие**	обсуждение вопросов обсуждение проекта
9	**Защита проектных работ**	Домашние задания 1 Проектная работа