

Специальная цена

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: 210 академических часов

1 Архитектура Linux

1 С чего начинается Linux

познакомиться с программой и группой,
объяснить, как работает ядро,
обновить ядро

Домашние задания

1 Делаем собственную сборку ядра

Цель: Делаем самостоятельную сборку ядра. Самостоятельно собирать ядро сейчас нужно не часто, но иногда необходимо - в случае embedded систем, либо для сборки защищенного монолитного ядра, либо маленького легковесного. В процессе разбираем какие пакеты нужны для успешной сборки, как ускорить сборку ядра с опцией `parallel`. Изучаем как конфигурируется ядро. Знакомимся к командой `make`. Результат первой ДЗ с пошаговым описанием в `README.md` выкладываем в `github`

Взять любую версию ядра с `kernel.org`

Подложить файл конфигурации ядра

Собрать ядро (попутно доставляя необходимые пакеты)

Прислать результирующий файл конфигурации

Прислать список доустановленных пакетов, взять его можно из `/var/log/yum.log`

Устанавливать будем на следующем занятии =)

2 Дискровая подсистема

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: Системный администратор обязан уметь работать дисковой подсистемой, делать это без ошибок, не допускать потерю данных. В этом задании необходимо продемонстрировать умение работать с software raid и инструментами для работы с разделами (parted, fdisk, lsblk)

добавить в Vagrantfile еще дисков
сломать/починить raid
собрать R0/R5/R10 - на выбор
создать на рейде GPT раздел и 5 партиций

в качестве проверки принимаются - измененный Vagrantfile, скрипт для создания рейда

* 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**

LVM - облегчаем себе жизнь управления файловыми системами.

архитектура файловой системы Linux: суперблок, блоки, inodes, журналы.

разбираемся в многообразии файловых систем

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**

писать скрипты на языке bash

Домашние задания

1 Пишем скрипт

Цель: В результате этого ДЗ вы научитесь писать простые скрипты, решающие нужные задачи, такие как мониторинг кол-ва ошибок в логах Освоите работу с файлами, поиском, парсингом текста, управлением блокировками

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

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 помощью команд nice, ionice

Научимся посылать различные сигналы процессам.

Домашние задания

Цель: В результате ДЗ вы разберетесь что находится в файловой системе /proc и закрепите навыки работы с bash. Зачастую, например в контейнерах, у вас нет кучу удобных утилит предоставляющих информацию о процессах, ip адресах, итд И есть только один инструмент bash и /proc

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. пробовать 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. пробовать запустить с разными nice

- Результат ДЗ - скрипт запускающий 2 процесса с разными nice и измеряющий 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.

Домашние задания

1 Размещаем свой RPM в своем репозитории

Цель: Часто в задачи администратора входит не только установка пакетов, но и сборка и поддержка собственного репозитория. Этим и займемся в ДЗ.

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

Домашние задания

1 Работа с загрузчиком

Цель: Зайти в систему без пароля рута - базовая задача сисадмина (ну и одно из заданий на любой линуксовой сертификации). Так же нужно уметь управлять поведением загрузчика. Это и будем учиться делать в ДЗ

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 --bootloaderareaseize 1m

8 Инициализация системы. Systemd и SysV.

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

Домашние задания

1 Systemd

Цель: Управление автозагрузкой сервисов происходит через systemd. Вместо cron'a тоже используется systemd. И много других возможностей. В ДЗ нужно написать свой systemd-unit.

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 возможность запустить несколько экземпляров сервера с разными конфигами
- 4*. Скачать демо-версию Atlassian Jira и переписать основной скрипт запуска на unit-файл

Задание необходимо сделать с использованием Vagrantfile и provisioner shell (или ansible, на Ваше усмотрение)

1 Автоматизация администрирования. Ansible.

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

Лабораторная работа: пишем скрипт для апгрейда системы после установки и изменения 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 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
-

2 Пользователи и группы. Авторизация и аутентификация

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

Лабораторная работа: даем пользователю А возможность запускать скрипт, принадлежащий пользователю В

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 Резервное копирование.

Обсуждаем политики и методики резервного 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

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

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 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

7 **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

8 **Docker** разбираем как писать Dockerfile
создаем docker-compose
запускаем docker swarm

Домашние задания

1 Docker, docker-compose, dockerfile

Описание ДЗ в документе

9 **SELinux - когда все запрещено.** Разбираемся, что такое SELinux

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, поснимфферим через tcpdump и ngrep

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 получиться следующие сервера

- inetRouter
- centralRouter
- office1 Router
- 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 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

Практическая 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 исключения selinux

3 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
Практическая 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 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 inetRouter
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
-

5 Мосты, туннели и VPN

Разбираемся в терминах и протоколах - что такое мосты, туннели, VPN, PPP, PPTP, PPoE, IPoE, GRE, 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

Прочувствовать 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

6 Сетевые пакеты. VLAN'ы. LACP.

Изучаем Транспортный уровень
Изучаем протокол LACP. Учимся агрегировать интерфейсы через teaming и bonding.
Разбираемся что такое VLAN. Знакомимся с dot1q, macvlan
Осваиваем работу с nmcli
Лабораторная работа: агрегируем интерфейсы в режиме active/active и failover

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 вланы

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

* реализовать teaming вместо bonding'a (проверить работу в active-active)

** реализовать работу интернета с test машин

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/nginx_http_rewrite_module.html

https://nginx.org/ru/docs/http/nginx_http_headers_module.html

2 **Динамический веб контент**

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

Домашние задания

1 Собрать стенд с 3мя проектами на выбор

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 стенд с проброшенными на локалхост портами
каждый порт на свой сайт
через nжинкс

3 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 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

- 4 **MySQL - бэкап, репликация, кластер** настраиваем политику инкрементального бэкапа
Настраиваем GTID репликацию и кластер с proxysql
Настраиваем Galera и Innodb кластера

Домашние задания

1 mysql cluster

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

- 5 **Redis, Memcached, RabbitMQ** Разбираем что это такое и зачем нужны эти сервисы.
Устанавливаем и настраиваем их
Запускаем в работу.
-

6 Почта: SMTP, IMAP, POP3

разбираем почтовые протоколы.
Устанавливаем и настраиваем Postfix и Dovecot
изучаем настройки безопасности

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"

7 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

8 PostgreSQL cluster

настраиваем кластер PostgreSQL

9 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 | Понять устройства кластера на основе pacemaker.
Построить тестовый кластер с плавающим ип и вебсервисом. |
| 3 | ISCSI, multipath и кластерные файловые системы: GFS2 | разобрать принципы работы ISCSI
разобраться что такое multipath
настроить общее хранилище для двух нод |
| 4 | Распределенные файловые системы: CEPH | понимать 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 |
| 5 | Распределенные файловые системы: GlusterFS | построить распределенное хранилище GlusterFS |
| 6 | Механизмы изоляции и аккаунтинга Linux (namespaces и cgroups) | |

7 **Защита
проектных работ** Домашние задания

1 Проектная
работа

8 **Итоговое
занятие** обсуждение организационных вопросов
фидбэк по курсу
обсуждение проекта