



# Administrator Linux.Basic

Администратор Linux. Базовый уровень

Длительность курса: 86 академических часов

## 1 Вводный урок

### Цели занятия:

познакомиться с программой курса;  
обсудить дальнейшую работу;  
ответить на вопросы студентов.

### Краткое содержание:

программа курса;  
обсуждение домашнего задания;  
обсуждение дальнейшей работы.

---

## 2 Вопрос-ответ по курсу OnlineLinux

### Цели занятия:

обсудить процесс обучения на предзаписанном курсе OnlineLinux;  
задать вопросы по темам занятий курса OnlineLinux;  
получить ответы на вопросы.

### Краткое содержание:

вопросы по темам  
операционная система - общие сведения;  
UNIX;  
структура Linux;  
простейшие команды в Linux;  
пользователи в Linux;  
работа с файлами;  
работа с потоками STDIN, STDOUT, STDERR;  
логические условия для выполнения команд;  
монтирование;  
установка ПО;  
процессы в Linux;  
сеть;  
системы инициализации;

### 1 Bash. Написание простых скриптов

#### Цели занятия:

писать простые скрипты для оболочки bash;  
автоматизировать рутинные задачи.

#### Краткое содержание:

что такое скрипт;  
как заставить ОС выполнять скрипт в нужной оболочке;  
варианты запуска скрипта.

#### Домашние задания

- 1 Необходимо написать скрипт, который проверяет систему на предмет работы службы selinux

Цель: Напишите скрипт, который проверяет систему на предмет работы службы selinux

Выполняется на CentOS.

Необходимо написать скрипт, который проверяет систему на предмет работы службы selinux а именно:

1. Проверяет включена ли на данный момент selinux.
2. Активирована ли selinux в конфиге.
3. Выдает собранную информацию в виде диалога:
  - selinux работает/не работает, в конфиге активирована/не активирована
  - включить/выключить selinux?
  - активировать/деактивировать selinux в конфиге

Предварительно скрипт проверяет возможность своей работы от пользователя, который его запустил и говорит что нужно сделать чтобы скрипт работал.

У скрипта должен быть режим справочника. Т.е. если скрипт может управлять selinux то он предлагает это сделать. Если у скрипта такой возможности нет то он сообщает что нужно сделать/изменить чтобы получить нужный результат.

---

## 2 **Вопрос-ответ по Bash**

### **Цели занятия:**

обсудить сложности выполнения домашнего задания;  
задать преподавателю вопросы по теме Bash;  
получить ответы на вопросы.

### **Краткое содержание:**

типичные ошибки и сложности при выполнении ДЗ;  
ответы на ваши вопросы.

## 1 Жёсткие диски

### Цели занятия:

понимать из чего состоит жёсткий диск; понимать функционирование таблицы разделов; понимать ограничения структуры данных; структурировать дисковое пространство для загрузки ОС;

### Краткое содержание:

hdd,  
ssd,  
адресация,  
бинарные данные,  
mbr,  
gpt,  
утилита dd,  
разбор таблицы разделов mbr.

---

## 2 Вопрос-ответ

## 1 Конфигурирование web-сервера (apache, nginx, балансировка nginx)

### Цели занятия:

- установить apache;
- установить nginx;
- настроить apache на работу в кач-ве backend;
- настройка портов, проверка работоспособности (curl);
- настроить балансировку нагрузки в nginx:
- что такое балансировка;
- некоторые алгоритмы балансировок;
- возможности nginx для балансировки;
- настройка upstream на бэкенд apache.

### Краткое содержание:

понятия FrontEnd и BackEnd;  
apache, настройка его на работу на разных порта;  
nginx, настройка балансировки.

### Домашние задания

- 1 Настроить веб-сервер с балансировкой.  
FrontEnd — nginx, BackEnd — apache

Цель: В результате выполнения ДЗ вы создадите базовый скелет web-сервера с балансировкой нагрузки.

В данном задании тренируются навыки:

- декомпозиции предметной области
- установка ПО на сервер, работа с файлами конфигурации
- построения элементарной архитектуры FrontEnd/BackEnd web-сервера с балансировкой нагрузки

Необходимо:

- установить nginx и apache
- настроить работу apache на порты отличные от порта 80
- настроить работу nginx на порт 80
- настроить upstream в nginx для BackEnd apache

- настроить перенаправление обращения nginx на upstream.

---

## 2 Установка MySQL-сервера и настройка репликации

### Цели занятия:

рассмотреть принципы репликации в MySQL;  
настроить репликацию master-slave: настройка с использованием новой схемы аутентификации (caching\_sha2\_password);  
настроить репликацию master-master.

### Краткое содержание:

репликация распределенных систем на примере MySQL.

### Домашние задания

- 1 Настроить репликацию MySQL master-slave, настроить бэкап БД на slave (потаблично с указанием позиции бинлога)

Цель: В результате выполнения ДЗ вы создадите репликацию базы данных master-slave для последующей работы с бекапами.

В данном задании тренируются навыки:  
- понимание предметной области задания  
- установка ПО на сервер, работа с файлами конфигурации  
- построения репликации master-slave с последующей настройкой бекапа"

Необходимо:

- установить mysql на двух серверах
- проверить доступность порта mysql с одного сервера на другой
- создать пользователя для репликации на сервере master
- настроить реплику slave
- написать скрипт бекапа баз с реплики

## 1 Docker

### Цели занятия:

объяснить что такое контейнер;  
объяснить для каких задач были созданы контейнеры;  
объяснить когда не нужно использовать контейнер;  
поработать с контейнерами docker:  
- архитектура (рассмотреть сетевое взаимодействие контейнеров);  
- dockerhub;  
- образы (+ слои);  
- контейнеры;  
- запуск контейнера из образа;  
- подключение к контейнеру;  
- создание образа на основании контейнера;  
- почему в образе контейнера много мусора (удаление данных в образе не уменьшает размер образа).

### Краткое содержание:

базовые понятия docker (образы, контейнеры);  
правильное применение контейнеров (когда использовать контейнер хорошо, а когда он не нужен).

### Домашние задания

- 1 Установить docker, скачать образ nginx, запустить контейнер nginx. Настроить балансировку, как в задании про веб-сервер только в кач-ве FrontEnd использовать контейнер nginx

Цель: В результате выполнения ДЗ вы получите базовые навыки работы с контейнерами docker.

В данном задании тренируются навыки:

- понимание предметной области задания
- установка ПО на сервер, работа с файлами конфигурации
- базовая работа с контейнерами"

Необходимо:

- установить docker
- найти образ nginx и скачать его
- запустить контейнер nginx на базе образа nginx



- подключить конфигурационные файлы nginx из ДЗ с web-сервером в контейнер nginx

---

**Цели занятия:**

объяснить, что такое система управления версиями;  
рассмотреть особенности git — распределенная система контроля версий (каждый клиент имеет полную копию репозитория);  
поговорить об основных понятиях git;  
установить и поработать с git (установка, создание репозитория, добавление файлов в репозиторий, коммит).

**Краткое содержание:**

базовая работа с системой контроля версий git;  
репозиторий (repository, repo);  
рабочий каталог (working directory);  
область подготовленных файлов (staged area);  
ревизия (revision);  
коммит (commit).

**Домашние задания**

- 1 Создать репозиторий на GitHub с конфигами веб-сервера (из задачи про настройку вебсервера), настроить авторизацию по ключу, настроить автодеплой конфигов на веб-сервер из репозитория

Цель: В результате выполнения ДЗ вы получите базовые навыки работы с git и github.

В данном задании тренируются навыки:  
- работы с системой управления версиями  
- установка ПО на сервер, работа с файлами конфигурации  
- базовая работа с github

Необходимо:  
- установить git  
- создать репозиторий на github  
- добавить ключ для авторизации на github  
- авторизоваться по ключу на github

## 1 Сети. Базовые знания

### Цели занятия:

объяснить почему протоколы не пишут "единым куском";  
рассмотреть модель OSI;  
объяснить, что такое стек протоколов TCP/IP (что за "стек", сколько протоколов в названии);  
объяснить, что такое Ethernet, Wi-Fi... — канальный уровень;  
объяснить, что IP — сетевой уровень;  
объяснить, что TCP, UDP... — транспортный уровень;  
рассмотреть примеры протоколов прикладного уровня (http, ssh, ftp...);  
проанализировать на соответствие стека TCP/IP с моделью OSI;  
объяснить что обрабатывается ядром ОС (IP, TCP...), что обрабатывается приложениями (http, dns...).

### Краткое содержание:

базовые принципы сетевого взаимодействия и модель OSI.

### Домашние задания

- 1 Настроить сеть на Linux. Статический IP, DHCP, роутер в кач-ве DNS, Google NS в кач-ве DNS

Цель: В результате выполнения ДЗ вы получите базовые навыки настройки сети в Linux.

В данном задании тренируются навыки:  
- работа с файлами конфигурации ОС Linux  
- проверка работоспособности конфигурации  
- работа с базовыми командами для настройки сети в Linux"

Необходимо:

- назначить IP адрес для интерфейса с помощью команды ip
- назначить адрес dns сервера
- добавить маршрут по умолчанию

## 2 Сети. iptables

### Цели занятия:

- Узнать как в linux работает фильтрация пакетов.
- Поработать с iptables — утилитой командной строки, управляющей netfilter - сетевым фильтром, работающим на уровне ядра;
- Рассмотреть порядок прохождения пакета (packet flow) (пакет попадает в первое по порядку подходящее правило);
- Разобраться что такое таблицы, цепочки, правила;
- Рассмотреть основные цепочки таблицы filter (INPUT, FORWARD, OUTPUT);
- Рассмотреть примеры настройки правил фильтрации (по назначению, по источнику, по порту, по протоколу и т.д.);

### Краткое содержание:

- Утилита iptables, синтаксис, основные команды, написание правил

### Домашние задания

- 1 Снять дампы обращения к веб-серверу, проанализировать пакеты (начиная с первого). Описать на примере снятого дампа, как устанавливается сессия TCP. Настроить в iptables политику DROP, прописать разрешения только на нужные порты и протоколы

Цель: В результате выполнения ДЗ вы получите базовые навыки работы с iptables и tcpdump.

В данном задании тренируются навыки:

- анализ пакетов данных
- фильтрация трафика
- восстановление конфигурации фильтрации трафика после перезагрузки ОС"

Необходимо:

- собрать дампы пакетов с определенного IP и проанализировать их

- проанализировать нужные порты назначения и адреса источников для дальнейшей фильтрации траффика
  - настроить разрешения в iptables
  - сделать автовосстановление правил фильтрации после перезагрузки ОС
- 

### 3 Сети. tcpdump

#### **Цели занятия:**

- Научиться работать с утилитой tcpdump: извлекать пакеты по различным параметрам, сохранять результат для просмотра

#### **Краткое содержание:**

- Работа с tcpdump — утилитой, позволяющей "вытащить" пакет из пространства ядра в пространство пользователя;
- Рассмотреть базовые понятия и принцип работы (BPF);
- Проанализировать пакеты (по интерфейсу, по назначению, по источнику, по порту, по протоколу...).

## 1 Мониторинг

### Цели занятия:

- Узнать что такое мониторинг и для чего он нужен;
- Сравнить системы мониторинга: zabbix, icinga, cacti, grafana, prometheus;
- Рассмотреть варианты мониторинга: по событию (работает/не работает), сбор статистики (график загрузки сервера);
- узнать что такое предсказательный мониторинг;
- рассмотреть виды доставки данных для систем мониторинга.

### Краткое содержание:

- Утилиты локального мониторинга системы
- Варианты мониторинга;
- Настройка программного комплекса Prometheus;
- Подключение экспортера состояния "железа" node\_exporter;
- Визуализация данных с помощью Grafana.

### Домашние задания

- 1 Настроить сервер prometheus, настроить сбор метрик веб-сервера

Цель: В результате выполнения ДЗ вы настроите систему мониторинга prometheus.

В данном задании тренируются навыки:

- настройка ПО
- анализ метрик web-сервера
- постановка системы на мониторинг

Необходимо:

- установить и настроить prometheus
  - установить агент на web-сервер
  - настроить сбор метрик с web-сервера
  - настроить графическое отображение метрик в prometheus
-

### Цели занятия:

понимание базовых понятий логирования и централизованного сбора логов.

### Краткое содержание:

- что такое лог;
- как работает лог;
- зачем логируют информацию;
- чем плох бинарный лог и в чем его плюсы;
- зачем хранить логи в базе данных;
- логирование ssh
- примеры логов (`/var/log/secure`, `/var/log/dmesg...`);
- `journald`, `rsyslog`, `logrotate`.

### Домашние задания

#### 1 Настроить централизованный сбор логов в ELK

Цель: В результате выполнения ДЗ вы настроите систему сбора логов ELK.

В данном задании тренируются навыки:

- установка и настройка ПО;
- анализ сервера на основании данных логов

Необходимо:

- установить Elasticsearch, Logstash, Kibana
- настроить ELK Stack
- настроить сбор логов с web-сервера nginx

## 1 Консультация по проектам и домашним заданиям

### Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

### Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;

затруднения при выполнении ДЗ;

вопросы по программе.

### Домашние задания

#### 1 Итоговый проект

Цель: Выбрать тему проекта и закрепить её в чате с преподавателем.

Собрать воедино все знания и умения курса.

- создать репозитории в GitHub (конфиги, скрипты, cron файлы и т.д.)
  - настроить вебсервер с балансировкой нагрузки
  - настроить MySQL репликацию (master-slave)
  - установить CMS (на выбор: joomla/wordpress/wiki...)
  - написать скрипт для бэкапа БД со slave сервера (потаблично с указанием позиции бинлога, скрипт хранить в GitHub)
  - настроить систему мониторинга (конфиги хранить в GitHub)
  - разработать план аварийного восстановления (на основании скриптов, конфигов, cron файлов и бэкапов в максимально короткие сроки настроить новый сервер "с нуля")
  - продемонстрировать аварийное восстановление (на чистом сервере за короткий промежуток времени получить полностью настроенную рабочую систему)
-



**2 Защита  
проектных  
работ**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.