

# DevOps для эксплуатации и разработки

01	Кому подойдёт курс	<p>Системным администраторам или инженерам по эксплуатации ПО с опытом:</p> <ul style="list-style-type: none"><li>• сможете выполнять более сложные задачи</li><li>• получите возможность перейти на новую роль</li><li>• повысите свой грейд</li></ul> <p>Разработчикам или QA-инженерам:</p> <ul style="list-style-type: none"><li>• изучите самые актуальные DevOps-инструменты и освоите новые навыки для работы над инфраструктурными задачами</li><li>• сможете браться за более сложные проекты и расти профессионально</li></ul>
02	Чему научитесь на курсе  Какие знания и навыки освоите	<p>Начнёте применять DevOps в работе:</p> <ul style="list-style-type: none"><li>• разберёте ключевые практики и подходы методологии DevOps, которые стали стандартом в современной разработке и эксплуатации ПО, и научитесь использовать их</li></ul> <p>Изучите современные инструменты:</p> <ul style="list-style-type: none"><li>• освоите актуальные подходы и инструменты для тестирования и доставки приложений, описания инфраструктуры и управления ей</li></ul> <p>Освоите Docker и Kubernetes:</p> <ul style="list-style-type: none"><li>• будете создавать и масштабировать приложения с помощью контейнерной виртуализации и оркестрации</li></ul> <p>Будете сами проектировать пайплайны:</p> <ul style="list-style-type: none"><li>• от исходного кода до отказоустойчивого и нагруженного онлайн-маркета, как инженеры в командах разработки IT-продуктов</li></ul>
03	Как проходит обучение	<ul style="list-style-type: none"><li>• теория и практика на платформе Практикума</li><li>• практические задания на готовой инфраструктуре в Яндекс.Облаке</li><li>• воркшопы</li><li>• индивидуальные консультации с наставником</li><li>• обучение из любой точки мира в удобное время</li><li>• траблшутинг в расширенном тарифе</li></ul>

## Что вас ждёт на обучении

Начнёте применять в работе принципы и инструменты DevOps	Научитесь решать проблемы на реальной инфраструктуре	Сделаете и добавите в портфолио большой самостоятельный проект
--	--	--

Обучение на этом курсе может оплатить ваш работодатель. Если у вас есть вопросы, свяжитесь с нами по телефону: +7 (495) 139-49-47. Наши менеджеры вас проконсультируют.


DevOps для эксплуатации и разработки

Сравнение тарифов

	Интенсивный тариф	Стандартный тариф	Расширенный тариф
Длительность	5 месяцев	7 месяцев	9 месяцев
Нагрузка в неделю	~15 часов	~12 часов	~10 часов
Минимальные знания на старте	<ul style="list-style-type: none"><li>Опыт работы в IT не менее трёх лет</li><li>Умение писать небольшие скрипты на любом языке</li><li>Опыт работы с UNIX подобными системами</li><li>Знание языка SQL на уровне написания простых запросов</li></ul>	<ul style="list-style-type: none"><li>Опыт работы в IT не менее трёх лет</li><li>Умение писать небольшие скрипты на любом языке</li><li>Опыт работы с UNIX подобными системами</li><li>Знание языка SQL на уровне написания простых запросов</li></ul>	<ul style="list-style-type: none"><li>Опыт работы в IT не менее трёх лет</li><li>Умение писать небольшие скрипты на любом языке</li></ul>
Помощь с трудоустройством	Карьерный курс	Карьерный курс	Карьерный курс
Дополнительные темы	✗	✗	<ul style="list-style-type: none"><li>Основы Linux</li><li>Базовое программирование на Bash</li><li>Курс по Soft skills</li></ul>
Дополнительная практика	Практические задания в «песочнице» с самопроверкой	Практические задания в «песочнице» с самопроверкой	<ul style="list-style-type: none"><li>Практические задания в «песочнице» с самопроверкой</li><li>Регулярный траблшутинг</li></ul>
Сопровождение: наставник, куратор и поддержка учебной инфраструктуры	✓	✓	✓
Готовая инфраструктура в Yandex Cloud, симулирующая реальные рабочие ситуации	✓	✓	✓
Вебинары с экспертами	✓	✓	✓
Выпускной проект	✓	✓	✓

DevOps для эксплуатации и разработки

5, 7 или 9 месяцев  
продолжительность курса

<div>1 или 2 недели</div> <div>01</div> <div>Системы контроля версий и автоматизация сборки приложений</div>	<div>2 недели</div> <div>02</div> <div>Гибкие методологии и Continuous Integration</div>	<div>3 недели</div> <div>03</div> <div>Основы Linux</div> <div>Только в расширенном тарифе</div>	<div>2 недели</div> <div>04</div> <div>Создание скриптов на Bash</div> <div>Только в расширенном тарифе</div>
<div>2 недели</div> <div>05</div> <div>Сети и основы работы на серверах Linux</div>	<div>1 или 2 недели</div> <div>06</div> <div>Continuous Delivery и Continuous Deployment</div>	<div>2 или 3 недели</div> <div>07</div> <div>Infrastructure as Code и системы управления конфигурацией</div>	<div>2 недели</div> <div>08</div> <div>DBOps: реляционные и нереляционные базы данных</div>
<div>2 или 3 недели</div> <div>09</div> <div>Docker-контейнеризация и хранение данных</div>	<div>1 или 2 недели</div> <div>10</div> <div>Микросервисы, балансировка и кэширование</div>	<div>4 недели</div> <div>11</div> <div>Kubernetes. Деплой и обеспечение надёжности приложения</div>	<div>2 недели</div> <div>12</div> <div>Логирование и мониторинг ошибок</div>
<div>2 или 3 недели</div> <div>13</div> <div>Дипломный проект</div>	<div></div> <div>Факультативный курс. Эффективное взаимодействие с командой и управление ей</div> <div>Только в стандартном и расширенном тарифе</div>		

1 неделя (интенсивный тариф)  
2 недели (стандартный и расширенный тарифы)  
Практическая работа

Вы попадаете на работу в стартап, знакомитесь с командой и удивляетесь, как вообще можно так работать. Желая скорее всё поправить, вы разбираетесь в том, как устроена разработка программного обеспечения в наши дни, и делаете первые шаги на пути ускорения производства.

Узнаете основы концепции бережливого производства и поймёте, как они перекладываются на IT-процессы.

Научитесь автоматизировать повторяющиеся задачи при помощи Jenkins.

**Практика**

- Организация работы с ветками в Git
- Установка и настройка Jenkins, сборка проекта с помощью Maven

## Содержание

01.	Как устроен жизненный цикл ПО	
<hr/>		
02.	Системы контроля версий. Почему все выбирают Git?	<ul style="list-style-type: none"><li>• Организация командной работы с помощью Feature Branch Workflow</li><li>• Организация хранения Git-репозитория в Gitea</li></ul>
<hr/>		
03.	Бережливое производство: основы и принципы	
<hr/>		
04.	Автоматизация сборки	<ul style="list-style-type: none"><li>• Инструменты автоматизации повторяющихся задач</li><li>• Фреймворки для автоматизации сборки проекта</li></ul>
<hr/>		

2 недели  
Практическая работа

Ваш стартап покупает IT-гигант — Krupnaya Kompaniya (KK). Тимлид вашей команды становится вашим проводником на пути познания DevOps. Но проблемы с приложением никуда не уходят.

Вы узнаете, что приложение работает плохо на продуктивной среде, а в разработку постоянно приходят новые заявки на исправление дефектов, из-за чего копится работа и о выпуске нового функционала и стабильности даже не приходится мечтать.

Сможете организовать процесс CI в работе команды. Сумеете представить процесс сборки и поставки ПО в виде пайплайна, используя Gitlab CI.

Встроите в процесс поставки ПО этап анализа кода на безопасность и качество, используя анализаторы кода.

Практика

- Изучение структуры пайплайна в GitLab CI
- Настройка SonarQube, добавление в пайплайн этапов анализа качества (SonarQube) и безопасности (Gitlab SAST) кода
- Написание Jenkins Declarative Pipeline

## Содержание

01. Проблематика DevOps	<ul style="list-style-type: none"><li>• Функциональные колодцы</li><li>• Исходящая спираль</li></ul>
02. Гибкие методологии и DevOps культура	<ul style="list-style-type: none"><li>• Взаимодействие в команде</li><li>• Современные подходы. Чем обусловлено появление Agile?</li><li>• Три пути DevOps: цикл поставки ценности, петли обратной связи, эксперименты и обучение</li></ul>
03. Continuous Integration	<ul style="list-style-type: none"><li>• Организация хранения кода</li><li>• Организация сборки</li><li>• Организация тестирования</li><li>• Быстрое получение обратной связи</li></ul>
04. Сервера Continuous Integration	<ul style="list-style-type: none"><li>• Обзор интерфейса и функционала Gitlab CI</li></ul>
05. Измерение качества и статическое тестирование безопасности кода	<ul style="list-style-type: none"><li>• Инструменты статического анализа качества кода: SonarQube, Gitlab Analyzer</li><li>• Инструменты статического анализа безопасности кода: SonarQube SAST, Gitlab SAST</li></ul>

3 недели  
Практическая работа

Техлид КК выбирает вас, чтобы провести инструктаж по Linux для новых сотрудников. Но оказывается, что ваших знаний недостаточно и нужно снова вливаться в ряды стажёров. Шаг за шагом вы научитесь обращаться с Linux, как с родной операционной системой.

Освойте базу Linux, без которой невозможно представить работу DevOps.

- Практика
- Работа с командной строкой
  - Взаимодействие с файловой системой и разделами в Linux
  - Управление ресурсами ОС и распределение их по задачам
  - Траблшутинг: устранение неисправностей в «песочнице»

## Содержание

01.	Архитектура (устройство Linux)
<hr/>	
02.	Терминал и основные команды
<hr/>	
03.	Дисковая подсистема и ввод-вывод
<hr/>	
04.	Процессы и память
<hr/>	
05.	Сетевая подсистема
<hr/>	
06.	Графическая система
<hr/>	
07.	Администрирование: демоны, пользователи, загрузка операционной системы и введение в Bash
<hr/>	
08.	Траблшутинг
<hr/>	

2 недели

Практическая работа

Вы переходите на новый уровень изучения UNIX-подобных систем и погружаетесь в один из самых популярных языков для написания скриптов — Bash. И вы должны быть готовы встретиться с ним на практике.

Научитесь уверенно программировать на языке Bash и сможете писать скрипты на уровне, который необходим для прохождения курса.

Практика

- Создание полезных скриптов для закрепления синтаксиса Bash
- Взаимодействие с пользователями
- Выполнение задач на удалённых машинах

## Содержание

01.	Автоматизация рутинных операций с помощью Bash
<hr/>	
02.	Работа с потоками ввода-вывода
<hr/>	
03.	Использование Bash за пределами Linux
<hr/>	
04.	Синтаксис языка Bash
<hr/>	
05.	Основные функции языка Bash
<hr/>	

2 недели

Практическая работа

Вы знакомитесь с системным администратором КК, который рассказывает вам, как устроена сеть и как у него тут вообще всё работает. В процессе демонстрации вы видите, что админ гораздо быстрее управляется с командами и перемещениями курсора в консоли, и просите рассказать про эти приёмы. Админ предлагает вам разобраться на примере задачи по переносу сервиса со старого сервера на новый и заодно решить горячий вопрос.

Научитесь уверенно работать с сетями и серверами на Linux.

- Практика
- Работа с пользователями, сессиями и процессами
  - Назначение и управление правами пользователей
  - Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01.	Основы работы на серверах Linux, работа с командной строкой	
02.	Дисковая подсистема Linux	
03.	Права пользователей Linux	
04.	Основы сетей: проблемы с подключением к удалённому серверу	
05.	Виртуализация	<ul style="list-style-type: none"><li>• Популярные гипервизоры</li><li>• Виртуальные машины</li><li>• Виртуальные сети</li></ul>



1 неделя (интенсивный тариф)  
2 недели (стандартный и расширенный тарифы)  
Практическая работа

Получив очередное за неделю обновление приложения на смартфоне, вы заинтересовались, как часто обновления приходят пользователям. Оказалось, совсем не часто. А нужно ли чаще?

Сможете организовать часть деплоя в пайплайне и автоматизировать её, используя инструменты Gitlab CI. Сможете проводить оценку потребности в частых обновлениях ПО с точки зрения бизнеса.

Практика

- Публикация артефактов сборки проекта в хранилище Nexus
- Проектирование процесса поставки
- Добавление в пайплайн части с установкой на тестовую среду
- Откат на предыдущую версию приложения с помощью GitLab

Содержание

01.	Как устроен процесс поставки IT-продукта	<ul style="list-style-type: none"><li>• Что такое процесс поставки</li><li>• Value Stream management</li></ul>
.....		
02.	Системы хранения артефактов: Nexus	
.....		
03.	Continuous Delivery	<ul style="list-style-type: none"><li>• Визуализация этапов поставки</li><li>• Организация процесса поставки. Delivery Pipeline. Continuous Deployment</li></ul>
.....		
04.	Методология Twelve-Factor App	
.....		
05.	Бэкапирование	<ul style="list-style-type: none"><li>• Организация резервного копирования</li><li>• Виды бэкапов</li><li>• Ротации бэкапов</li></ul>
.....		

2 недели (интенсивный и стандартный тарифы)  
3 недели (расширенный тариф)  
Практическая работа

Вы впервые сталкиваетесь с проблемой в инфраструктуре. Кто-то изменил конфигурацию серверов разработчиков, на которых они проводили эксперименты, и теперь сервера работают настолько неправильно, что больше невозможно их использовать.

Научитесь развёртывать и управлять инфраструктурой приложения согласно подходу IaC, используя Terraform.

Практика

- Развёртывание инфраструктуры через Terraform
- Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01. Infrastructure as Code: описываем инфраструктуру кодом, перенимаем опыт разработки

02. Системы управления конфигурацией

- Ansible (roles, tasks, inventory, vault, awx)
- Знакомство с другими системами - Chef, Salt, Puppet

2 недели  
Практическая работа

Вы встречаетесь с DBA. К вам приходят грозные комментарии от пользователей, что сайт тормозит. Эмпирическим путём вы обнаруживаете, что последнее обновление сильно замедлило работу базы данных.

Научитесь применять практики DevOps к администрированию баз данных.

Практика

- Поднятие PostgreSQL и перенос данных в новую БД
- Переподключение приложения на вновь поднятую MongoDB
- Настройка автоматической миграции
- Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01.	Основы теории баз данных	<ul style="list-style-type: none"><li>• Реляционные базы данных: PostgreSQL</li><li>• Основные SQL-запросы</li><li>• Как ускорить выполнение запросов в БД: оптимизация, кэширование, рост мощностей, индексация</li></ul>
02.	Организация высокой доступности БД	<ul style="list-style-type: none"><li>• Принципы DBOps</li><li>• Миграция БД с помощью Flyway</li></ul>
03.	Нереляционные базы данных: MongoDB	

2 недели (интенсивный и стандартный тарифы)  
3 недели (расширенный тариф)  
Практическая работа

Вы окажетесь меж двух огней: с одной стороны — разработчики, у которых локально всё работает, а у кого-то из них, возможно, и нет; с другой — отдел эксплуатации, у которого вообще ничего не работает. Разбираемся, как сделать так, чтобы у всех было всё одинаково хорошее и наконец-то запускаем приложение в проде.

Научитесь разворачивать приложение, используя Docker, и поддерживать его работоспособность.

Практика

- Написание Docker-файла, сборка образа, публикация в Nexus, использование в пайплайне и развёртывание приложения на стенде с помощью Docker
- Установка MinIO, загрузка данных в хранилище
- Установка Vault, публикация секретов в Vault
- Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01.	Контейнеризация	<ul style="list-style-type: none"><li>• Обзор Linux- и Docker-контейнеров</li><li>• Устройство Docker: слои, образы, контейнеры, Docker-файл, Registry</li><li>• Альтернативы Docker (containerd, podman, LXC, cri-o)</li></ul>
02.	Хранение данных: работа с S3, MinIO vs Ceph	
03.	Хранение и передача чувствительных данных: Содержание уроков главы: большой обзор Vault	

1 неделя (интенсивный тариф)  
2 недели (стандартный и расширенный тарифы)  
Практическая работа

Вы все готовитесь к выходу новой версии приложения. Вы хотите учесть ошибки прошлого, когда ваш сервис не выдержал трафика и лежал три дня. На этот раз к вам пришёл бизнес и потребовал, чтобы подобного не случилось! Вы настолько погрузитесь в процесс, что привнесёте новых крутых фич и стратегий поставки.

Научитесь организовывать высоконагруженную систему, используя инструменты балансировки и кэширования.

- Практика
- Поднять Nginx и несколько экземпляров сервиса
  - Развёртывание локального кластера k8s и managed k8s кластера в Облаке
  - Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01.	Балансировка и кэширование	
<hr/>		
02.	Микросервисы	<ul style="list-style-type: none"><li>• Подходы к организации архитектуры приложения</li><li>• Инструменты для управления микросервисными приложениями: Docker-compose</li></ul>
<hr/>		
03.	Стратегии поставки и как выбрать подходящую	<ul style="list-style-type: none"><li>• Какие пользователи бывают</li><li>• Сине-зелёный деплой</li><li>• Канареечный деплой</li></ul>
<hr/>		

4 недели  
Практическая работа

Вы создаёте сервисную платформу для обеспечения качественной работы всего цикла поставки.

Научитесь использовать k8s для деплоя и обеспечения надёжности приложения. Познакомитесь с практиками GitOps, научитесь организовывать поставку приложений с помощью инструментов Flux, ArgoCD.

- Практика
- Работа с Minikube
  - Подготовка манифестов Kubernetes для деплоя в кластер Kubernetes и push в репозиторий GitLab
  - Модификация файла .gitlab-ci.yml и изменение шагов деплоя приложения
  - Деплой и передеплой через ArgoCD
  - Доведение YAML-файла до нового релиза по ТЗ
  - Написание Helm чарта для деплоя приложения
  - Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

## Содержание

01. Оркестрация контейнеров	<ul style="list-style-type: none"><li>• Запуск приложения в современной инфраструктуре</li><li>• Для чего нужны системы оркестрации и какие они бывают</li><li>• Kubernetes: причины и цели разработки</li></ul>
02. Запуск Kubernetes кластера для тестирования и разработки	<ul style="list-style-type: none"><li>• Как получить Kubernetes: описание 5 основных способов установки</li><li>• Разбор простых команд kubectl</li></ul>
03. Архитектура Kubernetes	<ul style="list-style-type: none"><li>• Основные компоненты и их взаимодействие: scheduler, kubelet, kube-proxy, controller manager, etcd и api-сервер</li></ul>
04. Основные сущности Kubernetes	<ul style="list-style-type: none"><li>• Nodes, Podes, Services, Persistent Volumes, Persistent Volume Claim</li><li>• Манифесты Kubernetes</li><li>• Использование kubectl на уровне подключения к кластеру</li><li>• Просмотр конфигурации Node, Pod'ов, Secrets и т.д.</li></ul>
05. Продвинутые сущности Kubernetes	<ul style="list-style-type: none"><li>• ReplicaSet, Deployment, DaemonSet и StatefulSet</li><li>• Liveness, Readiness probe + Init контейнеры</li><li>• Дополнение про kubectl: edit, apply, delete, create</li><li>• Сетевое взаимодействие в кластере</li></ul>
06. Продвинутая работа с утилитой kubectl	<ul style="list-style-type: none"><li>• Дополнительные команды kubectl, флаги, представление сущностей Kubernetes в виде YAML или JSON/YAML файлов</li><li>• Деплой приложения двумя способами: с помощью команды и через YAML-файл</li></ul>

07.	Деплой приложений в Kubernetes	<ul style="list-style-type: none"><li>• Деплой фронтенда приложения</li><li>• Деплой и передеплой через ArgoCD</li></ul>
08.	Стратегии деплоя	<ul style="list-style-type: none"><li>• Связь Kubernetes и Twelve-Factor App</li><li>• Стратегии деплоя в Kubernetes</li><li>• Как в Kubernetes реализован механизм rollback</li></ul>
09.	Практики GitOps для работы с инфраструктурой	
10.	Написание Helm чарта для деплоя приложения	
11.	Настройка ArgoCD	
12.	Использование шаблонизаторов	

Логирование и мониторинг ошибок

12

2 недели  
Практическая работа

Ваша сервисная платформа внезапно перестаёт работать. Научитесь настраивать систему сбора логов с помощью инструментов Loki, Sentry. Научитесь настраивать систему мониторинга, используя GAP стек.

- Практика
- Создание системы логирования Loki для вашего приложения. Построение графика работоспособности приложения
  - Поставка и настройка GAP стека для приложения, настройка алертов
  - Траблшутинг: устранение неисправностей в «песочнице» (только в расширенном тарифе)

Содержание

01.	Логирование и мониторинг ошибок	<ul style="list-style-type: none"><li>• Логи Linux</li><li>• Loki</li><li>• Sentry</li></ul>
02.	Мониторинг	<ul style="list-style-type: none"><li>• Типы метрик, типовые аномалии</li><li>• Алерты</li><li>• GAP стек (Grafana, Prometheus, Alertmanager)</li><li>• Golden Signals</li><li>• Метрики уровня приложения</li></ul>

03. АРМ системы и  
распределённый трейсинг

04. C.A.L.M.S

Дипломный проект

13

2 недели (интенсивный  
тариф)  
3 недели (стандартный  
и расширенный тарифы)  
Практическая работа

В финальном проекте вам предстоит подготовить инфраструктуру и настроить конвейер деплоя: системы автоматической сборки, автоматического тестирования и проверки кода на ошибки и уязвимости. Вы выстроите логику развёртывания приложения для разных окружений, настроите системы для мониторинга и логирования приложения.

Факультативный курс.  
Эффективное взаимодействие  
с командой и управление ей

Только в стандартном и расширенном тарифах

Содержание

01. Приоритеты и делегирование
02. Настройка коммуникации  
в команде
03. Настройка процессов  
и эффективность команды
04. Управление коммуникацией  
в IT-командах
05. Взаимодействие с командой  
разработки