

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Институт информационных технологий и анализа данных
(121)»

УТВЕРЖДЕНА:

на заседании совета института ИТиАД им. Е.И.Попова

Протокол №8 от 16 февраля 2026 г.

Рабочая программа дисциплины

«ОСНОВЫ DEVOPS»

Направление: 09.03.02 Информационные системы и технологии

Информационные системы и технологии в административном управлении

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой
электронной подписью
Составитель программы:
Харахинов Владимир
Александрович
Дата подписания: 10.06.2026

Документ подписан простой
электронной подписью
Утвердил: Говорков Алексей
Сергеевич
Дата подписания: 14.06.2026

Документ подписан простой
электронной подписью
Согласовал: Аршинский
Вадим Леонидович
Дата подписания: 11.06.2026

Год набора – 2026

Иркутск, 2026 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Основы DevOps» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-3 Способен обеспечивать эффективную работу баз данных, являющихся частью различных информационных систем, включая развертывание, сопровождение, оптимизацию их функционирования	ПКС-3.3

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-3.3	Способен грамотно выбирать и применять инструменты организации взаимодействия между разработчиками, тестировщиками и системными администраторами	Знать современные средства автоматизации развертывания и управления приложениями Уметь применять средства автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации Владеть системами непрерывной разработки; системами оркестрации контейнеров

2 Место дисциплины в структуре ООП

Изучение дисциплины «Основы DevOps» базируется на результатах освоения следующих дисциплин/практик: «Операционные системы», «Технологии программирования», «Технологии разработки программных комплексов»

Дисциплина является предшествующей для дисциплин/практик: «Проектирование информационных систем», «Производственная практика: преддипломная практика»

3 Объем дисциплины

Объем дисциплины составляет – 3 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 7
Общая трудоемкость дисциплины	108	108
Аудиторные занятия, в том числе:	48	48
лекции	16	16
лабораторные работы	32	32
практические/семинарские занятия	0	0
Самостоятельная работа (в т.ч. курсовое проектирование)	60	60
Трудоемкость промежуточной аттестации	0	0

Вид промежуточной аттестации (итогового контроля по дисциплине)	Зачет с оценкой	Зачет с оценкой
--	-----------------	-----------------

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 7

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Роль и задачи DevOps в процессе разработки программного обеспечения	1	2					2	10	Отчет по лабораторной работе
2	Непрерывная интеграция и развертывание	2	2	1	10			1, 4, 5	10	Отчет по лабораторной работе
3	Контейнеризация	3	4	4	6			1, 4, 5	14	Отчет по лабораторной работе
4	Оркестрация контейнеров	4	6	6	10			1, 4, 5	18	Отчет по лабораторной работе
5	Machine Learning Operations	5	2	7	6			1, 3, 5	8	Отчет по лабораторной работе
	Промежуточная аттестация									Зачет с оценкой
	Всего		16		32				60	

4.2 Краткое содержание разделов и тем занятий

Семестр № 7

№	Тема	Краткое содержание
1	Роль и задачи DevOps в процессе разработки программного обеспечения	Типовые проблемы при разработке программного обеспечения. Инспекция кода (code review). Рабочие процессы при версионировании (workflow). Понятие DevOps и лучшие практики. 7 принципов DevOps и основные инструменты. Культура DevOps. Фазы жизненного цикла DevOps.
2	Непрерывная интеграция и развертывание	Типовые проблемы при совместной разработке. Основные понятия CI/CD. Конвейер (pipeline) и основные этапы. Введение в Jenkins. Конвейер поставки. Требования к приложениям. Форматы и версионирование артефактов. Решение проблем с изменением окружений. Обновление и откат приложений.

3	Контейнеризация	Определение контейнеризации приложений, цели и задачи. Сравнение контейнеризации и виртуализации. Введение в Docker. Основные понятия: демон, образ, контейнер, репозиторий. Запуск приложений в контейнерах. Запуск служб в контейнерах. Построение образов Docker. Составление Dockerfile. Основные способы обмена данными с контейнером. Базовые сети host, none, bridge. Docker-compose и синтаксис YAML. Многоконтейнерные приложения.
4	Оркестрация контейнеров	Трудности при работе с контейнерами. Контейнерная оркестрация. Введение в Kubernetes. Задачи и архитектура кластера Kubernetes. Установка кластера Kubernetes. Запуск приложений в Kubernetes. Сетевой доступ к приложениям в Kubernetes. Рассмотрение альтернатив Kubernetes.
5	Machine Learning Operations	Реализация базового конвейера, на базе открытой платформы MLflow, обеспечивающего непрерывную доставку, развертывание и обучение нейросетевых моделей

4.3 Перечень лабораторных работ

Семестр № 7

№	Наименование лабораторной работы	Кол-во академических часов
1	Установка и администрирование Jenkins. Реализация конвейера CI/CD	10
4	Установка Docker и применение технологии для CI/CD	6
6	Установка Kubernetes и основы оркестрации. Мониторинг Kubernetes кластера	10
7	Создание конвейера для модели нейронной сети	6

4.4 Перечень практических занятий

Практических занятий не предусмотрено

4.5 Самостоятельная работа

Семестр № 7

№	Вид СРС	Кол-во академических часов
1	Оформление отчетов по лабораторным и практическим работам	8
2	Подготовка к зачёту	10
3	Подготовка к практическим занятиям	4
4	Подготовка к практическим занятиям (лабораторным работам)	22

5	Подготовка к сдаче и защите отчетов	16
---	-------------------------------------	----

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Интерактивная лекция, публичная презентация, интервью

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по лабораторным работам:

Методические указания содержатся в электронном курсе в ЭОС ИРНИТУ:
Аршинский В.Л Основы DevOps: электронный курс / В.Л. Аршинский, В.А. Харахинов
<https://el.istu.edu/course/view.php?id=7615>

5.1.2 Методические указания для обучающихся по самостоятельной работе:

Методические указания содержатся в электронном курсе в ЭОС ИРНИТУ:
Аршинский В.Л Основы DevOps: электронный курс / В.Л. Аршинский, В.А. Харахинов
<https://el.istu.edu/course/view.php?id=7615>

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 7 | Отчет по лабораторной работе

Описание процедуры.

Выполнение каждой лабораторной работы состоит из двух частей: практической части в ходе которой необходимо выполнить индивидуальное задание и подготовка и защита отчета по лабораторной работе. Подготовка отчета по лабораторной работе необходима для закрепления и переосмысления полученных знаний и навыков. Для успешной сдачи лабораторной работы необходимо продемонстрировать выполненное задание, устранить выявленные преподавателем недостатки, подготовить и защитить отчет. Требования к содержанию отчета указаны в соответствующем разделе методических указаний для каждой лабораторной работы.

Вопросы для контроля:

К лабораторной работе №1:

1. Что такое Jenkins и для чего он используется?
2. Опишите процесс установки Jenkins на операционную систему Windows/Linux
3. Какие шаги необходимо выполнить для первого запуска Jenkins?
4. Как создать новый проект (job) в Jenkins?
5. Объясните, как настроить систему сборки проекта на основе Git.
6. Как провести резервное копирование конфигурации Jenkins?

К лабораторной работе №2:

1. Что такое Docker и какие основные преимущества его использования?
2. Опишите процесс установки Docker на вашей операционной системе?
3. Какой командой можно проверить, что Docker запущен и работает корректно?
4. Что такое Docker-контейнер?

5. Каковы основные команды для управления контейнерами Docker? Приведите примеры.
6. Какая команда используется для запуска нового контейнера из образа?
7. Как можно остановить и удалить контейнер?
8. Что такое Docker-образ и как его создать?
9. Опишите процесс создания Docker-образа.** Какие команды для этого используются?
10. Как посмотреть список всех существующих контейнеров и образов?
11. Объясните, что такое сеть в Docker и как ею управлять.
12. Как настраиваются тома (volumes) в Docker и зачем они нужны?
13. Что произойдет с данными в контейнере, если контейнер будет удален?
14. Как можно управлять сетевыми настройками контейнеров?
15. Что такое Docker Compose и для чего он используется?
16. Что такое многослойные образы в Docker и как они работают?
17. Опишите структуру файла docker-compose.yml.
18. Как задать зависимости между сервисами в Docker Compose?

К лабораторной работе №3:

1. Что такое Kubernetes и какие его основные компоненты?
2. Какие команды используются для инициализации кластера Kubernetes?
3. Как создать и применить манифест для развертывания приложения в Kubernetes?
4. Какие инструменты можно использовать для мониторинга кластера Kubernetes?
5. Объясните, как использовать Persistent Volumes и Persistent Volume Claims в Kubernetes
6. Как настроить сервис для доступа к приложению извне кластера?
7. Объясните разницу между kubectl apply и kubectl create
8. Чем отличаются livenessProbe, readinessProbe и startupProbe?
9. Что именно происходит внутри Pod'a при запуске команды kubectl port-forward?

К лабораторной работе №4:

1. Что такое CI/CD и какова его основная цель?
2. Назовите основные этапы CI/CD процесса
3. Какова роль Jenkins в процессе CI/CD?
4. Что такое "pipeline" в Jenkins и как его создать?
5. Опишите процесс сборки приложения в вашем конвейере.
6. Как осуществляется развертывание приложения в CI/CD?
7. Приведите архитектуру Jenkins
8. Опишите процесс сборки при использовании распределенной модели развертывания

К лабораторной работе №5:

1. Чем MLOps принципиально отличается от DevOps в части управления версиями?
2. Как организовать непрерывное обучение (online learning) в production?
3. Объясните конвейер CI/CD для ML-модели: какие этапы должны быть на CI (валидация кода, данных, модели), а какие на CD (канареечное развертывание, A/B тест, blue-green для модели)? Как "откатить" модель при ухудшении метрик?
4. Какие метрики мониторинга модели в production критически важны и почему одних accuracy/recall недостаточно?

Критерии оценивания.

В процессе защиты отчета по лабораторной работе, студенту задаются контрольные вопросы теоретического и практического характера, соответствующие теме работы. Для успешной защиты отчета студенту необходимо дать краткое изложение основных результатов, полученных в ходе выполнения лабораторной работы, устно ответить на

теоретические вопросы по теме лабораторной работы. Успешная защита отчета является необходимым условием для выставления оценки «зачтено» по соответствующей лабораторной работе

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПКС-3.3	Демонстрирует способность обосновано применять средства автоматизации развертывания и управления приложениями, систем оркестрации контейнеров и непрерывной разработки	Выполнение лабораторных работ и устное собеседование

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 7, Типовые оценочные средства для проведения дифференцированного зачета по дисциплине

6.2.2.1.1 Описание процедуры

Зачет ставится по результатам выполнения лабораторных работ, посещения лекций и ответов на вопросы из билета к зачету. Для допуска к зачету должны быть выполнены и защищены все лабораторные работы. На зачете студент должен устно ответить на вопросы билета. Студент вправе не отвечать на вопросы билета и получить зачет по дисциплине, если он активно работал в ходе лабораторного практикума и сдал его до зачетной недели. Студент вправе отвечать только на один из вопросов билета (по своему выбору) и получить зачет по дисциплине, если лабораторный практикум был сдан на зачетной неделе и было посещено более 70% занятий по дисциплине. Допускается письменный ответ на вопросы билета на зачете (по решению преподавателя).

Вопросы к зачёту:

1. Что такое Docker и какие основные преимущества его использования?
2. Какой командой можно проверить, что Docker запущен и работает корректно?
3. Что такое Docker-контейнер?
4. Каковы основные команды для управления контейнерами Docker? Приведите примеры.
5. Какая команда используется для запуска нового контейнера из образа?
6. Как можно остановить и удалить контейнер?
7. Что такое Docker-образ и как его создать?
8. Опишите процесс создания Docker-образа. Какие команды для этого используются?
9. Как посмотреть список всех существующих контейнеров и образов?
10. Объясните, что такое сеть в Docker и как ею управлять.
11. Как настраиваются тома (volumes) в Docker и зачем они нужны?
12. Что произойдёт с данными в контейнере, если контейнер будет удалён?
13. Как обеспечить безопасность контейнеров Docker?

14. Как можно управлять сетевыми настройками контейнеров?
15. Как настроить перенаправление портов для контейнера?
16. Как можно запустить контейнер и обеспечить доступ к приложению из внешней сети?
17. Что такое Docker Compose и для чего он используется?
18. Что такое многослойные образы в Docker и как они работают?
19. Опишите структуру файла docker-compose.yml.
20. Какие команды используются для работы с Docker Compose? Приведите примеры
21. Как задать зависимости между сервисами в Docker Compose?
22. Как выполнить многоконтейнерное приложение с помощью Docker Compose? Приведите пример.
23. Как обновить сервис в Docker Compose после изменения исходного кода?
24. Как можно масштабировать сервисы в Docker Compose?
25. Что такое Kubernetes и какие его основные компоненты?
26. Какие команды используются для инициализации кластера Kubernetes?
27. Как создать и применить манифест для развертывания приложения в Kubernetes?
28. Какие инструменты можно использовать для мониторинга кластера Kubernetes?
29. Объясните, как использовать Persistent Volumes и Persistent Volume Claims в Kubernetes
30. Как настроить сервис для доступа к приложению извне кластера?
31. Что такое Jenkins и для чего он используется?
32. Опишите процесс установки Jenkins на операционную систему Windows/Linux
33. Какие шаги необходимо выполнить для первого запуска Jenkins?
34. Как создать новый проект (job) в Jenkins?
35. Объясните, как настроить систему сборки проекта на основе Git.
36. Как провести резервное копирование конфигурации Jenkins?
37. Что такое непрерывная интеграция?
38. Что такое непрерывное развертывание?
39. Что такое CI/CD и какова его основная цель?
40. Назовите основные этапы CI/CD процесса
41. Какова роль Jenkins в процессе CI/CD?
42. Что такое "pipeline" в Jenkins и как его создать?
43. Как осуществляется развертывание приложения в CI/CD?

Пример задания:

Типовой билет к зачёту:

Вопрос 1:

Что такое CI/CD и какова его основная цель?

Вопрос 2:

Опишите структуру файла docker-compose.yml.

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительный	Неудовлетворительно
Глубокое понимание теоретических основ изученного материала, уверенное	Хорошее понимание теоретических основ изученного материала, владеет	Знание основных понятий из разделов дисциплины, владеет изученными программно-инструментальными	Наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неуверенность и

владение понятийным аппаратом и терминологией, способен применять изученные программно-инструментальные средства для решения задач в области профессиональной деятельности, грамотное и логически стройное изложение материала при ответах на все вопросы из билета, знание дополнительно рекомендованной литературы	изученными программно-инструментальными средствами и способен применять их для решения задач в области профессиональной деятельности, незначительные ошибки при ответах на заданные вопросы	средствами, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов	неточность ответов на дополнительные и наводящие вопросы, отсутствие навыков владения программно-инструментальными средствами из лабораторного практикума
---	--	---	--

7 Основная учебная литература

1. Баланов А. Н. DevOps: интеграция и автоматизация : учебное пособие для вузов / А. Н. Баланов, 2024. - 240.

[Сайт] – URL: <https://e.lanbook.com/book/401129>

2. Орещенков И. С. Операционные системы. Bodhi Linux 6.0: установка, настройка, эксплуатация : учебное пособие для вузов / И. С. Орещенков, 2024. - 364.

[Сайт] – URL: <https://e.lanbook.com/book/403370>

3. Baier J. The Complete Kubernetes Guide. Become an Expert in Container Management with the Power of Kubernetes / J. Baier, G. Sayfan, J. White, 2019. - 616.

[Сайт] – URL: <http://elib.istu.edu/viewer/view.php?file=/files3/er-37672.pdf>

8 Дополнительная учебная литература и справочная

1. Ляхов Д. А. Linux для начинающих / Дмитрий Ляхов, 2003. - 254.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Лицензионное программное обеспечение Системное программное обеспечение
2. Лицензионное программное обеспечение Пакет прикладных офисных программ
3. Лицензионное программное обеспечение Интернет-браузер
4. Свободно распространяемое программное обеспечение Свободно распространяемая система контроля версий Git
5. Свободно распространяемое программное обеспечение Свободно распространяемая программная система с открытым исходным кодом Jenkins
6. Свободно распространяемое программное обеспечение Свободно распространяемый контейнеризатор приложений Docker
7. Свободно распространяемое программное обеспечение Открытое программное обеспечение для оркестровки контейнеризированных приложений Kubernetes

12 Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения лекционных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, доска. Мультимедийное оборудование (в том числе переносное): мультимедийный проектор, экран, акустическая система, компьютер с выходом в интернет.
2. Учебная аудитория для проведения лабораторных/практических (семинарских) занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащение: комплект учебной мебели, рабочее место преподавателя, компьютеры "i5-4440(3.1)/8Gb/500Gb/VGA/23"" или аналогичные по техническим характеристикам в количестве не менее 16 шт.