


МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой  
ПиИТ

 /Махортов С.Д./  
03.05.2023

## РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

### Б1.В.13 Основы DevOps

- 1. Код и наименование направления подготовки/специальности:**  
09.03.04 Программная инженерия
- 2. Профиль подготовки/специализация:**  
Информационные системы и сетевые технологии
- 3. Квалификация выпускника:** бакалавр
- 4. Форма обучения:** очная
- 5. Кафедра, отвечающая за реализацию дисциплины:**  
Программирования и информационных технологий (ПиИТ)
- 6. Составители программы:** Махортов Сергей Дмитриевич, д.ф.-м.н., доцент
- 7. Рекомендована** НМС ФКН, протокол № 7 от 03.05.2023.

---

*отметки о продлении вносятся вручную)*

---

**8. Учебный год:** 2025 / 2026

**Семестр(ы):** 6

### 9. Цели и задачи учебной дисциплины:

- изложить основы современных подходов и методов DevOps, включая автоматизацию процессов разработки, тестирования, доставки и эксплуатации программного обеспечения;
- рассмотреть типовые задачи, связанные с контейнеризацией, оркестрацией, конфигурацией инфраструктуры и мониторингом систем;
- научить студентов профессионально использовать инструменты DevOps для построения и настройки процессов CI/CD, обеспечивая качество и стабильность программного продукта;
- выработать способности и мотивацию к внедрению DevOps практик, а также практические навыки их применения в реальных проектах с учетом требований бизнеса и современных технологий.

**10. Место учебной дисциплины в структуре ООП:** Часть, формируемая участниками образовательных отношений (вариативная) блока Б1. Требуется предварительное знание основ программирования, системного администрирования, сетевых технологий. Предшествует дисциплинам: производственная практика, научно-исследовательская работа.

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-1	Способен выполнять интеграцию программных компонент	ПК-1.1	Собирает программные компоненты в программный продукт.	<p>Знать: основные этапы жизненного цикла ПО и их связь с DevOps. Принципы работы систем контейнеризации (Docker) и оркестрации (Kubernetes).</p> <p>Уметь: создавать и управлять контейнерами с использованием Docker и Kubernetes.</p> <p>Владеть: подходами к интеграции DevOps в процесс разработки ПО. Методами развертывания, масштабирования и мониторинга приложений.</p>
ПК-3	Способен выполнять проектирование ПО	ПК-3.1 ПК-3.2 ПК-3.3	<p>ПК-3.1 Разрабатывает и согласовывает архитектуру ПО с системным аналитиком.</p> <p>ПК-3.2 Проектирует структуры данных и баз данных.</p> <p>ПК-3.3 Проектирует программные интерфейсы.</p>	<p>Знать: основные концепции и принципы методологии DevOps. Назначение и области применения инструментов DevOps. Основные понятия контейнеризации, оркестрации, систем управления конфигурацией и мониторинга. Цели, преимущества и инструменты непрерывной интеграции и доставки (CI/CD).</p> <p>Уметь: применять инструменты DevOps для разработки, тестирования, доставки и мониторинга ПО. Применять на практике инструменты CI/CD</p> <p>Владеть: навыками разработки и эксплуатации ПО. Подходами к построению инфраструктуры как кода (IaC). Инструментами мониторинга и анализа состояния систем.</p>

**12. Объем дисциплины в зачетных единицах/час.**(в соответствии с учебным планом) — 2 / 72.

**Форма промежуточной аттестации** (зачет/экзамен) зачет.

### 13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		№ 6	№ семестра	...
Аудиторные занятия	32	32		
в том числе:	лекции	16	16	
	практические	16	16	
	лабораторные	-	-	
Самостоятельная работа	40	40		
в том числе: курсовая работа (проект)	-	-		
Форма промежуточной аттестации (зачет с оценкой – 0 час.)	-	-		
Итого:	72	72		

#### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Лекции</b>		
1.1	Введение в DevOps и Docker.	Основы философии DevOps, роль DevOps-инженера в разработке ПО, жизненный цикл ПО. Основные принципы и практики DevOps. Введение в контейнеризацию и Docker. Основные понятия Docker: образы, контейнеры, тома, сети.
1.2	Системы виртуализации и контейнеризации.	Обзор систем виртуализации и контейнеризации. Docker образы: Dockerfile, слои, инструкции. Docker Compose: управление многоконтейнерными приложениями. Сети в Docker: bridge, host, none, overlay. Тома в Docker: использование для постоянного хранения данных.
1.3	Оркестрация. Системы оркестрации	Обзор оркестраторов: что такое оркестраторы, зачем они нужны, примеры использования. Основы оркестрации с использованием Kubernetes или других систем оркестрации. Обзор Kubernetes или других систем оркестрации: архитектура, основные компоненты.
1.4	Системы управления конфигурацией. Инфраструктура как код.	Что такое IaC и почему это важно. Обзор системы для описания и развертывания инфраструктуры через код (например, Terraform). Обзор систем управления конфигурацией. Основы управления конфигурацией с использованием Ansible. Основные встроенные модули Ansible.
1.5	Системы непрерывной интеграции и доставки	Обзор CI/CD: что такое CI/CD, зачем оно нужно, обзор основных инструментов CI/CD. Непрерывная интеграция: принципы, практики, инструменты. Непрерывная доставка и непрерывное развертывание: различия, принципы, практики, инструменты.
1.6	Мониторинг. Системы мониторинга	Введение в мониторинг, основы и принципы. Инструменты мониторинга.
<b>2. Практические занятия</b>		
2.1	Введение в DevOps и Docker	Изучение основных команд Docker CLI. создание и запуск Docker контейнеров на основе существующих образов, а также создание собственных Docker образов.
2.2	Системы виртуализации и контейнеризации	Создание Dockerfile для сборки собственных Docker образов, используют Docker Compose для управления многоконтейнерными приложениями, настраивают сети и тома в Docker для обеспечения связи между контейнерами и сохранения данных.
2.3	Оркестрация. Системы оркестрации	Применение Kubernetes для оркестрации контейнеров, управление и автоматизация развертывания и управление контейнерами с использованием основных инструментов Kubernetes. Интеграция оркестратор с другими инструментами DevOps.
2.4	Системы управления конфигурацией. Инфраструктура	Практическое задание с применением инструментов автоматизации конфигурации (Ansible). Управление конфигурацией сервера и приложения через код. Интеграция с другими инструментами DevOps.

	как код	
2.5	Системы непрерывной интеграции и доставки	Практические задания по настройке CI/CD с использованием Jenkins или другого инструмента. Создание пайплайнов CI/CD для автоматической сборки, тестирования и развертывания приложений.
2.6	Мониторинг. Системы мониторинга	Практическое применение инструментов мониторинга, таких как Grafana, Prometheus или других, для выявления проблем, отслеживания производительности.
<b>3. Лабораторные работы</b>		

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в DevOps и Docker	2	2		4	8
2	Системы виртуализации и контейнеризации	2	2		10	18
3	Оркестрация. Системы оркестрации	4	4		6	10
4	Системы управления конфигурацией. Инфраструктура как код	2	2		6	12
5	Системы непрерывной интеграции и доставки	4	4		10	16
6	Мониторинг. Системы мониторинга	2	2		4	8
	Итого:	16	16		40	72

### 14. Методические указания для обучающихся по освоению дисциплины

(рекомендации обучающимся по освоению дисциплины: работа с конспектами лекций, презентационным материалом, выполнение практических заданий, тестов, заданий текущей аттестации и т.д.)

Работа с конспектами лекций и презентационным материалом; выполнение практических заданий и тестов; выполнение лабораторных заданий; подготовка к заданиям текущей аттестации.

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Баланов, А. Н. DevOps: Интеграция и автоматизация : учебное пособие для вузов / Баланов А. Н. Санкт-Петербург : Лань, 2024. – 240 с.
2	Гэри, Грувер. Запуск и масштабирование DevOps на предприятии : учебно-методическое пособие / Гэри Грувер. Москва : ДМК-пресс, 2018 – 80 с.
3	Ленц, М. Python. Непрерывная интеграция и доставка : монография / Ленц М. Москва : ДМК-пресс, 2020. – 168 с.
4	Херинг, М. DevOps для современного предприятия : монография / Херинг М. Москва : ДМК-пресс, 2020. – 232 с.
5	Форсгрэн, Н. Ускоряйся! Наука DevOps. Как создавать и масштабировать высокопроизводительные цифровые организации : монография / Форсгрэн Н. ; Хамбл Д. ; Ким Д. Москва : Интеллектуальная литература, 2020. – 216 с.

б) дополнительная литература:

№ п/п	Источник
6	Дэвис Д.: Философия DevOps. Искусство управления. / Пер. с англ. – Питер, 2016 - 416 с.
7	Хамбл Д. Непрерывное развертывание / Д. Хамбл, Фарли. – М.: ООО, 2011, 432 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
-------	--------

--	--

\* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

**16. Перечень учебно-методического обеспечения для самостоятельной работы**  
(учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение)**

Для реализации учебного процесса применяются: современные инструментальные системы Visual Studio Code, JetBrains CLion, PyCharm Community, Microsoft Visual Studio Community, Jenkins, Ansible, Grafana, Docker, Kubernetes; ресурс «Электронный университет» (<https://edu.vsu.ru/>).

**18. Материально-техническое обеспечение дисциплины**

1. Компьютерный класс №5 (ауд. 295). ПК-Intel-Core2 14 шт., рабочее место преподавателя: проектор, видеокоммутатор, специализированная мебель: доска маркерная 1 шт., столы 16 шт., стулья 33 шт. В классе находится точка доступа беспроводной сети для доступа в Интернет и к учебно-методическим материалам, расположенным на внутренних серверах факультета.

2. Компьютерный класс №7 (ауд. 316п). ПК на базе IntelCore2Duo 2,8ГГц, ОЗУ 2ГБ, диск 160Gb – 30 шт. Специализированная мебель: доска маркерная 1 шт., доска интерактивная 1 шт., столы 32 шт., стулья 64 шт.; рабочее место преподавателя: проектор, видеокоммутатор. В классе находится точка доступа беспроводной сети для доступа в Интернет и к учебно-методическим материалам, расположенным на внутренних серверах факультета.

**19. Оценочные средства для проведения текущего контроля и промежуточной аттестаций**

**19.1. Текущий контроль**

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме письменных работ (контрольные, выполнение практико-ориентированных заданий). Критерии оценивания приведены ниже.

**19.1.1 Задания для текущего контроля успеваемости**

Приведённые ниже задания рекомендуется использовать при проведении диагностических работ для оценки остаточных знаний по дисциплине

**Тестовые задания – 1 балл**

1. Какой инструмент предназначен для работы с контейнерами??
  - 1) Jenkins
  - 2) Docker
  - 3) Ansible

2. Как называется файл, используемый для описания конфигурации контейнера Docker?
  - 1) Config.yml
  - 2) DockerConfig
  - 3) Dockerfile
  
3. Какая команда Docker используется для создания нового образа Docker?
  - 1) docker run
  - 2) docker build
  - 3) docker create
  
4. Какая команда служит для отображения запущенных контейнеров Docker?
  - 1) docker ps
  - 2) docker create
  - 3) docker build
  
5. Команда для запуска всех контейнеров, определенных в файле docker-compose.yml?
  - 1) docker-compose ps
  - 2) docker-compose up
  - 3) docker-compose start
  
6. Команда, позволяющая вывести список образов Docker Compose?
  - 1) docker-compose down
  - 2) docker-compose build
  - 3) docker-compose images
  
7. Какой из перечисленных инструментов применяется для оркестрации контейнеров??
  - 1) Kubernetes
  - 2) Docker
  - 3) Jenkins
  
8. Что означает принцип "инфраструктура как код"?
  - 1) Управление инфраструктурой через описания в виде кода
  - 2) Автоматизация процесса развертывания
  - 3) Создание серверов вручную
  
9. Как расшифровывается CI в CI/CD?
  - 1) Непрерывная поставка
  - 2) Управление версиями
  - 3) Непрерывная интеграция
  
10. Как расшифровывается CD в CI/CD?
  - 1) Непрерывная интеграция
  - 2) Непрерывная поставка
  - 3) Управление версиями
  
11. Что такое Pods в Kubernetes?
  - 1) Это группа контейнеров с общими разделами, запускаемых как единое целое.
  - 2) Это интерфейс командной строки.
  - 3) Это машина в кластере Kubernetes

12. Что является основной целью систем мониторинга?

- 1) Автоматизация процесса развертывания
- 2) Сбор и анализ данных о состоянии системы
- 3) Обеспечение безопасности системы

Номер вопроса	Ответ
1	2
2	3
3	2
4	1
5	2
6	3
7	1
8	1
9	3
10	2
11	1
12	2

**Задания с кратким ответом - 2 балла**

1. Назовите основное назначение Docker Compose.
2. В чем различие между непрерывной интеграцией (CI) и непрерывной доставкой (CD)??
3. Что такое Nodes в Kubernetes?
4. Что такое тома в Docker?
5. Какие сети поддерживаются в Docker?
6. Какую задачу решает оркестратор Kubernetes?
7. Что такое пайплайн?
8. Для чего служит автоматизация конфигурации?
9. Что такое Docker-образ?
10. Что такое контейнер?

Номер вопроса	Ответ
1	Управления несколькими контейнерами, входящими в состав приложения
2	CI автоматизирует сборку и тестирование кода, а CD автоматизирует процесс развертывания обновлений
3	Это машина в кластере Kubernetes
4	Хранилище информации, используемое в контейнерах
5	Bridge, Host, None, Overlay
6	Управление развертыванием, масштабированием и жизненным циклом контейнеризированных приложений.
7	Это последовательность действий или процессов, которые выполняются для достижения заданной цели
8	Позволяет управлять конфигурацией серверов и

	приложений через код
9	Шаблон для создания Docker-контейнеров.
10	Единица программного обеспечения, которая упаковывает код и все его зависимости

### Задания с развернутым ответом – 3 балла

1. Опишите основные принципы и практики DevOps.
2. Какова роль DevOps-инженера в разработке ПО?
3. Что такое контейнеризация и зачем она нужна?
4. Что такое инфраструктура как код (IaC)?
5. Преимущества методологии инфраструктура как код?
6. Что представляет собой жизненный цикл ПО в контексте DevOps?
7. Что такое мониторинг системы?
8. Что такое CI/CD и какие преимущества оно предоставляет 15 командам разработки?
9. Какая разница между непрерывной интеграцией, непрерывной доставкой и непрерывным развертыванием?
10. Какую роль выполняет оркестрация?
11. Основные принципы мониторинга?
12. Преимущества Continuous Delivery (CD) в разработке программного обеспечения?
13. Преимущества Continuous Integration (CI) в разработке программного обеспечения?

<i>Критерии оценивания</i>	<i>Шкала оценок (в баллах)</i>
Обучающийся отвечает на все вопросы правильно.	3
Обучающийся отвечает на все вопросы. Допускаются незначительные неточности.	2
Обучающийся отвечает не на все вопросы. Ответ не содержит грубых ошибок.	1
Обучающийся отвечает не на все вопросы. Присутствуют грубые ошибки.	0

### 19.2. Промежуточная аттестация

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний и практические задания, позволяющие оценить степень сформированности умений и навыков.

Для оценивания результатов обучения на зачете с оценкой используются следующие показатели: владение теоретическими основами дисциплины, способность иллюстрировать ответ примерами, применять теоретические знания для решения практических задач.

Для оценивания результатов обучения на зачете применяется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Используются следующие показатели и их соотношения:

- уверенное владение теоретическими основами дисциплины, способность применять теоретические знания для решения практических задач, результаты выполнения всех заданий лабораторного практикума – «отлично»;



- хорошее владение теоретическими основами дисциплины, способность применять теоретические знания для решения практических задач, результаты выполнения большинства заданий лабораторного практикума – «хорошо»;
- неполное владение теоретическими основами дисциплины, затруднения в применении теоретических знаний для решения практических задач, результаты выполнения не менее 30% заданий лабораторного практикума – «удовлетворительно»;
- слабое владение теоретическими основами дисциплины, неспособность применять теоретические знания для решения практических задач, результаты выполнения менее 30% заданий лабораторного практикума – «неудовлетворительно».

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: собеседование по зачетным билетам (КИМ). Перечень вопросов к зачету:

1. Введение
  - 1.1. Основы философии DevOps.
  - 1.2. Жизненный цикл программного обеспечения (ПО).
  - 1.3. Особенности методологии DevOps.
  - 1.4. Основные инструменты DevOps и их области применения.
2. Системы виртуализации и контейнеризации
  - 2.1. Определение контейнеризации.
  - 2.2. Основные понятия Docker: образы, контейнеры, Dockerfile.
  - 2.3. Сети в Docker.
  - 2.4. Преимущества использования Docker.
3. Системы оркестрации контейнеров
  - 3.1. Оркестрация: определение и предназначение.
  - 3.2. Архитектура Kubernetes.
  - 3.3. Основные понятия Kubernetes: кластер, поды, сервисы, репликация, мастер-нода.
  - 3.4. Цели использования Kubernetes.
  - 3.5. Преимущества Kubernetes.
4. Системы управления конфигурацией. Инфраструктура как код (IaC)
  - 4.1. Что такое инфраструктура как код (IaC).
  - 4.2. Преимущества IaC: повторяемость процессов, легкость масштабирования, автоматическая документируемость, управление версиями конфигураций.
  - 4.3. Проблемы и решения при работе с IaC:
    - 4.3.1 Версионирование и контроль исходного кода.
    - 4.3.2 Тестирование инфраструктурного кода.
    - 4.3.3 Развертывание и масштабирование.
  - 4.4. Инструменты управления инфраструктурой как кодом.
  - 4.5. Инструменты автоматизации конфигурации.
5. Системы непрерывной интеграции и доставки
  - 5.1. Что такое Continuous Integration (CI).
  - 5.2. Цели и принципы CI.
  - 5.3. Преимущества CI.
  - 5.4. Что такое Continuous Delivery (CD).
  - 5.5. Цели и принципы CD.
  - 5.6. Преимущества CD.
  - 5.7. Основные инструменты CI/CD.

### ### 6. Мониторинг и системы мониторинга

- 6.1. Что такое мониторинг.
- 6.2. Основные принципы мониторинга.
- 6.3. Метрики мониторинга.
- 6.4. Инструменты мониторинга.
- 6.5. Дашборды: что они включают.