

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 26.06.2026 10:50:38	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Уникальный идентификатор средства для промежуточной аттестации по дисциплине "Современные методы DevOps" по направлению подготовки (специальности) "09.04.04 Программная инженерия" направленности (профилю) "Разработка программного обеспечения ФГБОУ ВО «ЧелГУ»" 04c19ed8b109b1506c07a48809a878888321529	стр. 1

**Фонд оценочных средств для промежуточной аттестации по дисциплине
Современные методы DevOps**

Направление подготовки (специальность)

09.04.04 Программная инженерия

Направленность (профиль)

Разработка программного обеспечения

Присваиваемая квалификация (степень)

магистр

Форма обучения

очная форма обучения

Год(ы) набора 2026

Челябинск 2026 г.

**09.04.04 Программная инженерия профиль Разработка программного обеспечения, дисциплина
Современные методы DevOps, 2026 год набора, очная форма обучения**

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.2026 А.А. Саламатов

Ученым советом института информационных технологий

Протокол заседания № 7 от 26.02.2026

Председатель Ученого совета
института информационных
технологий

согласовано

Ю.В. Петриченко

Заседанием кафедры информационных технологий и экономической информатики

Протокол заседания №7 от 26.02.2026

Заведующий кафедрой

согласовано

С.А. Скрипов

Автор (составитель)

С.А. Скрипов

**Структура фондов оценочных средств соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27
сентября 2022 № 573-1**



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	6
3.1. Виды оценочных средств	6
3.2. Содержание оценочных средств	7
4. Порядок проведения и критерии оценивания промежуточной аттестации	14
4.1. Порядок проведения промежуточной аттестации	14
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	14
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций	14



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Современные методы DevOps» по направлению подготовки 09.04.04 «Программная инженерия» направленности «Разработка программного обеспечения» ФГБОУ ВО «ЧелГУ»

стр. 3

1. Паспорт фонда оценочных средств

Направление подготовки: 09.04.04 Программная инженерия

Направленность: Разработка программного обеспечения

Дисциплина: Современные методы DevOps

Семестры: 3

Форма промежуточной аттестации: зачёт

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Современные методы DevOps» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-7	Способен применять при решении профессиональных задач методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях	ОПК-7.1. Знать архитектурные принципы клиент-серверных и распределённых систем, Модели хранения информации, протоколы передачи и кодирования сообщений, протоколы удаленного вызова процедур ОПК-7.2. Умеет проектировать схемы хранения данных под задачу, реализовывать интеграцию сервисов через API и выбирать модели хранения информации в целом ОПК-7.3. Имеет практический опыт анализа инструментов работы с данными, опыт взаимодействия со средствами хранения данных и технологиями сетевого взаимодействия.	Знать: принципы и инструменты DevOps Уметь: автоматизировать процессы интеграции и развертывания Владеть: навыками применения технологий DevOps
ПК-6	Способность к установке, администрированию программных систем и систем управления базами данных, оптимизации функционирования информационных систем и баз данных; способность проводить анализ системных проблем обработки информации, разрабатывать предложения по реализации технического сопровождения и перспективного развития информационных систем и баз данных	ПК-6.1. Демонстрирует знание архитектуры и администрирования информационных систем, систем управления базами данных, системного программного обеспечения, требований информационной безопасности ПК-6.2. Демонстрирует умения выбирать аппаратное и программное обеспечение исходя из требований к функционированию ИС и	Знать: Основы настройки инструментов DevOps Уметь: выбирать, устанавливать и настраивать инструменты DevOps Владеть: навыками использования инструментов DevOps



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю) «Современные методы DevOps» по направлению подготовки 09.04.04 «Программная инженерия» направленности «Разработка программного обеспечения» ФГБОУ ВО «ЧелГУ»

стр. 5

		баз данных, разрабатывать предложения по реализации сопровождения и развития информационных систем и ИТ-сервисов ПК-6.3. Имеет практический опыт установки, администрирования и интеграции программных систем и систем управления базами данных	
--	--	---	--



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	ОПК-7.1. Знать архитектурные принципы клиент-серверных и распределённых систем, Модели хранения информации, протоколы передачи и кодирования сообщений, протоколы удаленного вызова процедур Знать: принципы и инструменты DevOps	Основные принципы и инструменты DevOps Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО	Практическая работа	Задания теста № 1-49
2	ОПК-7.2. Умеет проектировать схемы хранения данных под задачу, реализовывать интеграцию сервисов через API и выбирать модели хранения информации в целом Уметь: автоматизировать процессы интеграции и развертывания	Основные принципы и инструменты DevOps Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО	Практическая работа	Задания теста № 1-49
3	ОПК-7.3. Имеет практический опыт анализа инструментов работы с данными, опыт взаимодействия со средствами хранения данных и технологиями сетевого взаимодействия. Владеть: навыками применения технологий DevOps	Основные принципы и инструменты DevOps Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО	Практическая работа	Задания теста № 1-49
4	ПК-6.1. Демонстрирует знание архитектуры и администрирования информационных систем, систем управления базами данных, системного программного обеспечения, требований информационной безопасности Знать: Основы настройки инструментов DevOps	Основные принципы и инструменты DevOps Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО	Практическая работа	Задания теста № 1-49
5	ПК-6.2. Демонстрирует умения выбирать аппаратное	Основные принципы и инструменты DevOps	Практическая работа	Задания теста № 1-49



	и программное обеспечение исходя из требований к функционированию ИС и баз данных, разрабатывать предложения по реализации сопровождения и развития информационных систем и ИТ-сервисов Уметь: выбирать, устанавливать и настраивать инструменты DevOps	Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО		
6	ПК-6.3. Имеет практический опыт установки, администрирования и интеграции программных систем и систем управления базами данных Владеть: навыками использования инструментов DevOps	Основные принципы и инструменты DevOps Основные принципы и инструменты MLOps Методы DevOps в жизненном цикле разработки ПО	Практическая работа	Задания теста № 1-49

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

Примеры тестовых вопросов

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
1.	Механизм ядра Linux для ограничения ресурсов, приоритизации, это:	a. namespaces b. cgroup c. runtime-spec d. image-spec
2.	Выберите пример "Container runtime"	a. CRI-O b. docker daemon c. OverlayFS d. cgroupfs
3.	спецификация образов контейнеров, это:	a. namespaces b. cgroup c. runtime-spec d. image-spec
4.	Механизм в Linux, позволяющий разным процессам иметь различные точки монтирования, сетевые адаптеры, списки процессов, это:	a. namespaces b. cgroup c. runtime-spec d. image-spec
5.	Если в Union File System верхний и нижний слои имеют файлы с одинаковым именем, после запуска контейнера:	a. Будет использован файл верхнего слоя



		<p>b. Будет использован файл нижнего слоя c. Произойдёт ошибка, контейнер не будет работать d. Произойдет объединение файлов</p>
6.	Система, эмулирующая аппаратное обеспечение компьютера, это:	<p>a. Виртуальная машина b. Контейнер c. Операционная система d. Файловая система</p>
7.	Экземпляр образа, работающий в изолированном пространстве имён это:	<p>a. Виртуальная машина b. Контейнер c. Операционная система d. Файловая система</p>
8.	Вы создали образ контейнера на основе другого существующего образа. Выберите правильный вариант:	<p>a. Оба образа будут использовать общие слои с совпадающими данными, отличающиеся данные нового образа будут добавлены как новый слой b. Новый образ будет иметь один слой, содержащий все данные, в том числе копию данных из предыдущего образа c. Новый образ будет хранить только команды, с помощью которых он был создан. При создании контейнера новый слой будет создаваться путём повторного выполнения этих команд</p>
9.	Вы создали образ контейнера на основе другого существующего образа. В новом образе будет удалён файл. Выберите правильный вариант:	<p>a. Новый слой будет содержать специальный файл, обозначающий удаление файла b. Будет создана копия слоя, содержавшего удаленный файл, но без этого файла. c. Произойдет ошибка, удалять файлы базового образа нельзя</p>
10.	Выберите пример файловой системы контейнера	<p>a. CRI-O b. docker daemon c. OverlayFS d. cgroupfs</p>
11.	Команда[linux]# docker pull ubuntu:22.04	<p>a. Загрузит в локальное хранилище образ ubuntu с тегом 22:04</p>



		b. Загрузит в локальное хранилище образ ubuntu, созданный не ранее, чем 22 апреля c. Создаст контейнер с тегом 22:04 на основе ubuntu
12.	При использовании этого типа хранилища в Docker, файлы могут храниться в любом месте хост-системы. Не-Docker процессы могут изменять их в любое время	a. Volumes b. Bind mounts c. tmpfs mount
13.	Вы запускаете контейнер: <code>docker run -d --name peach -v /html:/usr/share/nginx/html -p 8888:80 nginx:1.25.2</code> Как получить доступ по сети к содержимому <code>/html</code> ?	a. http://HOST_IP:8888/ b. <code>http://HOST_IP:80/</code> c. Никак, контейнер не будет взаимодействовать с сетью
14.	При использовании этого типа хранилища в Docker, файлы хранятся только в памяти хост-системы и никогда не записываются в файловую систему	a. Volumes b. Bind mounts c. tmpfs mount
15.	При использовании этого типа хранилища в Docker, файлы хранятся в части файловой системы хоста, которой управляет Docker. Не-Docker процессы не должны изменять эту часть файловой системы	a. Volumes b. Bind mounts c. tmpfs mount
16.	Если вашему приложению в контейнере Docker требуется высокопроизводительный ввод-вывод, лучше использовать:	a. Volumes b. Bind mounts
17.	Если вашему приложению в Docker требуется нативное поведение файловой системы, лучше использовать:	a. Volumes b. Bind mounts
18.	Вы хотите запустить контейнер Docker в интерактивном режиме. Для этого необходимо использовать опцию(опции):	a. <code>-d</code> b. <code>-e</code> c. <code>-it</code>
19.	Вы хотите запустить контейнер Docker в фоновом режиме. Для этого необходимо использовать опцию(опции):	a. <code>-d</code> b. <code>-e</code> c. <code>-it</code>
20.	Вы хотите запустить контейнер Docker и передать ему переменные окружения. Для этого необходимо использовать опцию(опции):	a. <code>-d</code> b. <code>-e</code> c. <code>-it</code>
21.	Если вы монтируете непустой том в каталог в контейнере, в котором существуют некоторые файлы или каталоги:	a. эти файлы или каталоги будут скрыты монтированием b. эти файлы или каталоги будут удалены c. произойдет ошибка монтирования тома
22.	Команда CMD в Dockerfile:	a. Выполняет команду сборки внутри базового образа



		<p>b. Указывает номер порта, который будет прослушивать служба в контейнере</p> <p>c. Устанавливает команду по умолчанию, которая запускается, когда пользователь запускает контейнер на основе этого образа. Либо определяет параметры по умолчанию для ENTRYPOINT</p>
23.	Команда в Dockerfile COPY hello.py /Сделает следующее:	<p>a. Скопирует локальный файл hello.py в собираемый образ</p> <p>b. Скопирует локальный файл hello.py внутри файловой системы хоста</p> <p>c. Скопирует файл hello.py из собираемого образа в локальную файловую систему</p>
24.	Команда docker build для сборки образа использует файл:	<p>a. Dockerfile</p> <p>b. image.yaml</p> <p>c. Imagefile</p>
25.	Команда RUN в Dockerfile:	<p>a. Выполняет команду сборки внутри базового образа</p> <p>b. Указывает номер порта, который будет прослушивать служба в контейнере</p> <p>c. Устанавливает команду, которая запускается, когда пользователь запускает контейнер на основе этого образа</p>
26.	Зачем в Dockerfile объединяют команды, например: RUN apt-get update && apt-get install -y python3 python3-pip	<p>a. Каждая строка Dockerfile - это отдельный слой образа. Команды объединяются для уменьшения количества слоёв</p> <p>b. Каждая строка Dockerfile - это отдельный процесс в контейнере, созданном на основе этого образа. Команды объединяются для уменьшения количества процессов</p> <p>c. Для лучшей читаемости</p>



		Dockerfile
27.	Docker Compose предназначен для:	a. Для запуска приложений из нескольких контейнеров b. Для сборки новых образов c. Для запуска контейнеров в кластере
28.	Как обычно называется файл, который использует Docker Compose для создания приложения?	a. docker-compose.yml b. compose.sh c. app.json
29.	Для создания и запуска контейнеров в Docker Compose используется команда:	a. up b. start c. exec
30.	Откуда Docker Compose берёт имя приложения?	a. Имя приложения совпадает с именем текущего каталога, где находится файл docker-compose.yml b. Имя приложения находится в файле docker-compose.yml c. Имя приложения присваивается случайным образом
31.	Сервисы в приложении Docker Compose:	a. Могут взаимодействовать по сети, можно использовать имя сервиса как сетевое имя b. Не могут взаимодействовать по сети, для взаимодействия необходимо использовать API Docker Compose c. Имеют одинаковый IP адрес 127.0.0.1 но разные порты
32.	Вы создали Deployment в K8s а затем вручную удалили все pod этого Deployment что произойдёт?	a. Все pod из Deployment будут автоматически запущены в соответствии с количеством реплик b. Deployment будет автоматически удалён c. Вы не сможете вручную удалить эти pod
33.	Утилита, которая автоматизирует установку и настройку компонентов Kubernetes	a. kubeadm b. kubectl c. kubelet



34.	Утилита для управления кластером Kubernetes, позволяющая например создать pod, deployment и т.п., это:	a. kubectl b. kubectl c. kubelet
35.	В командекubectl run nginx --image=nginx --port=80 -n test1 "test1" - это:	a. Пространство имён b. Метка c. Название pod d. Название node
36.	Агент, который работает на каждом node Kubernetes, это:	a. kubectl b. kubectl c. kubelet
37.	Kubernetes предназначен для:	a. Автоматизации развёртывания, масштабирования и координации контейнеризированных приложений в условиях кластера b. Сборки образов, хранения образов контейнеров c. Запуска приложений, состоящих из нескольких контейнеров, на одном узле
38.	Отдельная физическая или виртуальная машина, на которой развёрнуты и выполняются контейнеры приложений в K8s, это:	a. Узел (node) b. Под (pod) c. Том (volume) d. Сервис(service)
39.	Базовая единица для запуска и управления приложениями: один или несколько контейнеров, которым гарантирован запуск на одном узле в K8s, это:	a. Узел (node) b. Под (pod) c. Том (volume) d. Сервис(service)
40.	Вы создали Deployment в K8s и установилиreplicas:3Что это значит?	a. Будет поддерживаться 3 экземпляра pod b. В случае ошибки будет 3 попытки перезапуска c. различные pod этого Deployment будут развернуты на 3 обязательно разных node
41.	Общий ресурс хранения для совместного использования из контейнеров, развёрнутых в пределах одного пода в K8s, это:	a. Узел (node) b. Под (pod) c. Том (volume) d. Сервис(service)
42.	Для автоматизации настройки и развёртывания программного обеспечения лучше использовать:	a. Ansible b. Prometheus c. cadvisor d. Grafana



43.	Для запуска задач на хостах в Ansible:	a. Нет необходимости устанавливать Ansible на все хосты b. Необходимо установить Ansible на все эти хосты c. Хосты должны быть созданы на основе одного из дистрибутивов, имеющих поддержку Ansible в ядре
44.	Для запуска задач на выбранных хостах в Ansible можно использовать команду:	a. ansible-playbook b. ansible-start c. ansible-run
45.	Выберите систему для визуализации данных	a. Ansible b. Prometheus c. Grafana d. docker
46.	Программа, для снятия метрик с системы (процессор, память, и т.д.) и дальнейшей передачи Prometheus:	a. Pushgateway b. Node exporter c. Alertmanager
47.	Выберите систему для сбора и анализа данных о работоспособности IT-оборудования	a. Ansible b. Prometheus c. Grafana d. docker
48.	Для визуализации данных Prometheus хорошо подходит:	a. Grafana b. cadvisor c. Pushgateway
49.	Для того, чтобы существующие короткое время пакетные задания могли предоставлять свои метрики Prometheus, можно использовать:	a. Pushgateway b. Node exporter c. Alertmanager



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Зачёт проводится в виде тестирования. Студент должен ответить на вопросы закрытого типа, которые предполагают выбор вариантов ответа. Всего 20 тестовых вопросов. Продолжительность теста – 35 минут.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Тест формируется в системе электронного обучения MOODLE.
Максимальный балл за тест — 100 баллов.

Оценка	Зачтено	Незачтено
Баллы	100-60 баллов	59-0 баллов
Уровень освоения проверяемых компетенций	высокий	низкий

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Для получения оценки "зачтено" обучающийся должен выполнить все практические работы и получить не менее, чем 60 баллов за итоговый тест.

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке зачтено:
 - предполагает формирование компетенций на высоком уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки удовлетворительно;
 - студент умеет применять на практике знания, полученные в рамках изучения дисциплины
 - формируются навыки использования теоретических и практических разделов дисциплины для решения задач профессиональной деятельности;
2. Низкий уровень соответствует оценке незачтено.