

Документ подписан простой электронной подписью

Информация о владельце: МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФИО: Максимов Алексей Борисович
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Должность: директор департамента по образовательной политике

Дата подписания: 30.09.2026 22:54:19

Уникальный программный ключ:

8db180d1a3f02ac9e60521a5672742735c18b1d6

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)**

Факультет Информационные технологии

УТВЕРЖДАЮ

Декан факультета

«Информационных технологий»

 / Д.Г. Демидов /
« 26 » февраль 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Технологии сборки и развертывания программного обеспечения»

Направление подготовки/специальность

09.03.01 Информатика и вычислительная техника

Профиль/специализация

«Веб-технологии и продуктовый дизайн»

Квалификация

бакалавр

Формы обучения

Очная, заочная

Год начала обучения – 2026 г.

Москва, 2026 г.

Разработчик(и):

К.Т.Н., доцент



/ Е.А. Пухова /

К.Т.Н., доцент



/ В.Ю. Верещагин /

Согласовано:

Заведующий кафедрой «Инфокогнитивные технологии»,



К.Т.Н., доцент

/ Е.А. Пухова /

Содержание

1	Цели, задачи и планируемые результаты обучения по дисциплине	4
2	Место дисциплины в структуре образовательной программы	5
3	Структура и содержание дисциплины	5
3.1	Виды учебной работы и трудоемкость для очной формы обучения	5
3.2	Тематический план изучения дисциплины для очной формы обучения	5
3.3	Содержание дисциплины	7
3.4	Тематика лабораторных занятий	8
4	Учебно-методическое и информационное обеспечение	8
4.1	Нормативные документы и ГОСТы	8
4.2	Основная литература	9
4.3	Дополнительная литература	9
4.4	Электронные образовательные ресурсы	9
4.5	Лицензионное и свободно распространяемое программное обеспечение	9
4.6	Современные профессиональные базы данных и информационные справочные системы	10
5	Материально-техническое обеспечение	10
6	Методические рекомендации	10
6.1	Методические рекомендации для преподавателя по организации обучения	10
6.2	Методические рекомендации для обучающихся по освоению дисциплины	10
7	Фонд оценочных средств	11
7.1	Методы контроля и оценивания результатов обучения	11
7.2	Шкала и критерии оценивания результатов обучения	12
7.3	Оценочные средства	13

1 Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины «Технологии сборки и развертывания программного обеспечения» относится: формирование у студентов компетенций в области автоматизации процессов сборки, тестирования, контейнеризации и развертывания веб-приложений в современных инфраструктурных средах.

К основным задачам дисциплины «Веб-аналитика» относятся:

- изучение жизненного цикла ПО (SDLC) и места процессов сборки и доставки в нем;
- освоение инструментов систем управления версиями (Git) для командной работы;
- изучение принципов контейнеризации приложений (Docker) и оркестрации (Kubernetes/Docker Compose);
- формирование навыков настройки конвейеров непрерывной интеграции и доставки (CI/CD);
- изучение методов управления конфигурациями и инфраструктурой как код (IaC);
- выполнение предоставленных практических заданий различных форм, как в процессе контактной, так и в ходе самостоятельной работы

Обучение по дисциплине «Технологии сборки и развертывания программного обеспечения» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-1. Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ИПК-1.2. Умеет: создавать, модифицировать и сопровождать информационные системы для решения задач бизнес-процессов и организационного управления; проводить сравнительный анализ и выбор ИКТ для решения прикладных задач и создания информационных систем; разрабатывать сайты, мобильные приложения, позволяющие автоматизировать конкретные бизнес-процессы для заданной организации.
ПК-4. Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ИПК-4.2. Умеет: разрабатывать алгоритмы для конкретных задач, учитывая различия в представлении информационных объектов современными браузерами; проводить работы по интеграции программных модулей с использованием языков программирования; разрабатывать программные модули для заданной организации, встраивать их в существующую систему и проверять работоспособность разработанных модуле и системы в целом; тестировать написанную программу на наличие ошибок, интегрировать различные модули в одну информационную систему

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.2 и междисциплинарно связана с поддерживающими дисциплинами: Рекрутинговые технологии, Разработка и анализ требований, Нейросетевые технологии, Проектирование пользовательских интерфейсов в веб и последующими дисциплинами: Современные методы разработки веб-приложений.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетных единицы, т.е. 72 академических часов.

Разделы дисциплины изучаются на четвертом курсе в седьмом семестре, форма промежуточной аттестации - зачет.

3.1 Виды учебной работы и трудоемкость для очной формы обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			7	
1	Аудиторные занятия	36	36	
	В том числе:			
1.1	Лекции	18	18	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	18	18	
2	Самостоятельная работа	36	36	
3	Промежуточная аттестация			
	Зачет/диф.зачет/экзамен		зачет	
	Итого:	72	72	

для заочной формы обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			8	
1	Аудиторные занятия	8	8	
	В том числе:			
1.1	Лекции	4	4	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	4	4	
2	Самостоятельная работа	64	64	
3	Промежуточная аттестация			
	Зачет/диф.зачет/экзамен		зачет	
	Итого:	72	72	

3.2 Тематический план изучения дисциплины для очной формы обучения

№ п/ п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
	Раздел 1. Введение в DevOps и управление версиями						
1	Тема 1.1 «Жизненный цикл ПО и методология DevOps»	8	2		2		4
2	Тема 1.2 «Продвинутое использование Git»	8	2		2		4
	Раздел 2. Контейнеризация приложений						
3	Тема 2.1 «Основы Docker»	8	2		2		4
4	Тема 2.2 «Оркестрация контейнеров»	8	2		2		4
	Раздел 3. Автоматизация сборки и доставки (CI/CD)						
5	Тема 3.1 «Настройка конвейеров непрерывной интеграции»	9	2		3		4
6	Тема 3.2 «Непрерывное развертывание (CD)»	9	2		3		4
	Раздел 4. Инфраструктура и мониторинг						
7	Тема 4.1 «Инфраструктура как код (IaC) и управление конфигурациями»	10	2		2		6
8	Тема 4.2 «Мониторинг и логирование»	12	4		2		6
Итого		72	18		18		36

для заочной формы обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
	Раздел 1. Введение в DevOps и управление версиями						
1	Тема 1.1 «Жизненный цикл ПО и методология DevOps»	9	0,5		0,5		8
2	Тема 1.2 «Продвинутое использование Git»	9	0,5		0,5		8
	Раздел 2. Контейнеризация приложений						

3	Тема 2.1 «Основы Docker»	9	0,5		0,5		8
4	Тема 2.2 «Оркестрация контейнеров»	9	0,5		0,5		8
	Раздел 3. Автоматизация сборки и доставки (CI/CD)						
5	Тема 3.1 «Настройка конвейеров непрерывной интеграции»	9	0,5		0,5		8
6	Тема 3.2 «Непрерывное развертывание (CD)»	9	0,5		0,5		8
	Раздел 4. Инфраструктура и мониторинг						
7	Тема 4.1 «Инфраструктура как код (IaC) и управление конфигурациями»	9	0,5		0,5		8
8	Тема 4.2 «Мониторинг и логирование»	9	0,5		0,5		8
Итого		72	4		4		64

3.3 Содержание дисциплины

Раздел 1. Введение в DevOps и управление версиями

- Тема 1.1. Жизненный цикл ПО и методология DevOps.
 - Понятие CI/CD (Continuous Integration, Continuous Delivery/Deployment).
 - Культура DevOps: взаимодействие разработки и эксплуатации.
 - Обзор инструментов: Git, Jenkins, GitLab CI, GitHub Actions.
- Тема 1.2. Продвинутое использование Git.
 - Стратегии ветвления (Git Flow, GitHub Flow, Trunk Based Development).
 - Code Review, Merge Requests, Hooks.
 - Разрешение конфликтов слияния.

Раздел 2. Контейнеризация приложений

- Тема 2.1. Основы Docker.
 - Архитектура Docker: образы, контейнеры, реестры.
 - Написание Dockerfile: инструкции, оптимизация слоев, multi-stage builds.
 - Управление сетями и томами в Docker.
- Тема 2.2. Оркестрация контейнеров.
 - Docker Compose для локальной разработки и тестирования.
 - Введение в Kubernetes: Pods, Deployments, Services.
 - Управление конфигурациями (ConfigMap, Secrets).

Раздел 3. Автоматизация сборки и доставки (CI/CD)

- Тема 3.1. Настройка конвейеров непрерывной интеграции.
 - Автоматизация сборки (Build automation): Maven, Gradle, npm, Webpack.
 - Автоматизация тестирования в пайплайне (Unit, Integration tests).
 - Статический анализ кода (Linting, SonarQube).
- Тема 3.2. Непрерывное развертывание (CD).
 - Стратегии деплоя: Blue-Green, Canary, Rolling Updates.
 - Деплой в тестовое и продуктивное окружение.
 - Rollback и управление версиями релизов.

Раздел 4. Инфраструктура и мониторинг

- Тема 4.1. Инфраструктура как код (IaC) и управление конфигурациями.
 - Основы Ansible: плейбуки, роли, инвентаризация.
 - Принципы идиоматичности.
- Тема 4.2. Мониторинг и логирование.
 - Сбор и анализ логов (ELK Stack, Loki).
 - Метрики и алертинг (Prometheus, Grafana).
 - Health checks и readiness probes.

3.4 Тематика лабораторных занятий

1. Настройка репозитория и стратегий ветвления. Создание структуры проекта, настройка Git hooks, реализация Git Flow.
2. Создание Docker-образа веб-приложения. Написание оптимизированного Dockerfile для frontend (React/Vue) и backend (Node.js/Python/Go) частей.
3. Оркестрация микросервисов с Docker Compose. Поднятие стека сервисов (приложение + БД + Cache + Nginx). Настройка сетей.
4. Настройка CI-пайплайна. Автоматизация запуска тестов и линтеров при каждом Push в репозиторий (GitLab CI или GitHub Actions).
5. Автоматическая сборка и публикация образов. Настройка шага сборки Docker-образа и отправки его в Registry (Docker Hub / GitLab Registry).
6. Развертывание приложения (CD). Настройка автоматического деплоя на тестовый сервер по тегу релиза.
7. Управление конфигурациями с Ansible. Написание плейбука для настройки веб-сервера и развертывания приложения без использования контейнеров (классический деплой).
8. Сквозная задача: Полный цикл DevOps. Реализация пайплайна от коммита кода до работающего приложения в изолированном окружении с мониторингом.

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);
2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденный Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального... Редакция с изменениями N 1456 от 26.11.2020
3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;
4. Порядок проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры, утвержденный приказом Минобрнауки России от 29 июня 2015 г. № 636;
5. Положение о практической подготовке обучающихся, утвержденное приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390;
6. Устав и локальные нормативные акты Московского политеха

Области профессиональной деятельности и сферы профессиональной деятельности, в которых выпускники, освоившие программу бакалавриата (далее - выпускники), могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом)

Выпускники могут осуществлять профессиональную деятельность в других областях и (или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника, предъявляемым соответствующими профессиональными стандартами.

4.2 Основная литература

Верещагин, В.Ю. Создание веб-страниц на стороне клиента: учебно- методическое пособие для студентов бакалавриата, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Веб-технологии» / В.Ю. Верещагин, М.В. Даньшина. – Москва: Московский Политех, 2023. – 1 CD-R. – Загл. с титул. экрана. – Текст: электронный. ISBN 978-5-2760-2786-9. — URL: <https://online.mospolytech.ru/mod/data/view.php?d=127&rid=5281&filter=1>

4.3 Дополнительная литература

1. Непрерывное развертывание контейнеризованных приложений с помощью GitLab https://yandex.cloud/ru/docs/tutorials/infrastructure-management/gitlab-containers?utm_referrer=https%3A%2F%2Fyandex.ru%2F

2. documentation.docker.com, kubernetes.io/docs, gitlab.com/docs (Официальная документация инструментов).

3. Ресурсы Habr, Medium по тегам CI/CD, Docker, Kubernetes.

4. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2023. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519714>.

4.4 Электронные образовательные ресурсы

1. ЭОР еще не разработан

4.5 Лицензионное и свободно распространяемое программное обеспечение

- ОС: Linux (Ubuntu/CentOS) или macOS (возможно использование WSL2 на Windows).
- Системы контроля версий: Git.
- Контейнеризация: Docker, Docker Compose.
- CI/CD системы: GitLab CI (Self-hosted или SaaS), GitHub Actions.
- Конфигурация: Ansible.
- Виртуализация: VirtualBox / VMware или облачные провайдеры (Yandex Cloud, Selectel)

4.6 Современные профессиональные базы данных и информационные справочные системы

1. <https://doka.guide/>
2. <https://developer.mozilla.org/ru/>
3. <https://roadmap.sh/frontend>

5 Материально-техническое обеспечение

Для проведения лабораторных работ и самостоятельной работы студентов подходят аудитории, оснащенные компьютерами с программным обеспечением в соответствии со списком в пункте 4.5 и подключенные к интернету.

Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

Рабочее место преподавателя должно быть оснащено компьютером с подключенным к нему проектором или иным аналогичным по функциональному назначению оборудованием.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические рекомендации для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются лекции и самостоятельная работа.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;

- сформированность компетенций;
- оформление материала в соответствии с требованиями.

Приветствуется обсуждение самих заданий с другими студентами: можно как давать, так и получать советы по общей стратегии выполнения и изучения материала, давать и получать помощь в отладке. Однако писать код студент должен самостоятельно. Делиться кодом или писать его совместно запрещено.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Приведенные ниже правила выставления оценок и опозданий могут быть изменены, если преподаватель сочтет это необходимым. Важно, чтобы студенты регулярно просматривали план курса, выложенный в СДО, на предмет его обновления или изменения.

Достижение компетенций оценивается с помощью лабораторных работ и рубежных контролей.

В соответствии с планом на дисциплины студентам выдаются задания на лабораторные работы. Помимо требований и описания функционала в работе указан крайний срок сдачи. Для сдачи лабораторной работы студенту необходимо прислать ссылку на репозиторий в GitHub Classroom и на хостинг, где размещен результат работы с реализованным функционалом, описанным в задании. Работа считается сданной если в ней реализовано 80% и более требований и функционала, описанного в задании.

Каждый студент имеет право на 6 дней опоздания, которые могут быть потрачены на любые задания в течение семестра. Опоздания предназначены для решения особых ситуаций, таких как болезнь или чрезвычайные семейные обстоятельства.

Когда использованы все дни опоздания за каждый день просрочки начисляется штраф в размере 25% от максимального результата за задание. Задания, присланные позже, чем 4 дня, не будут оцениваться. В связи с зависимостью между работами студентам может потребоваться все равно выполнить предыдущие работы, даже если они не оцениваются.

После сдачи лабораторной работы студент должен ее защитить. Во время защиты лабораторной работы преподаватель проверяет репозиторий, хостинг и выполнение критериев и требований задания, а студент отвечает на вопросы преподавателя по его коду, а также теоретических вопросов, приведенных после текста задания лабораторной работы. Если студент отказывается отвечать на вопросы, или дает полностью неверные ответы, или ответы не по теме, то работа может считаться сданной, но при этом она не оценивается.

Работа должна быть выполнена студентом самостоятельно: в репозитории в системе контроля версий студента содержатся коммиты только за его авторством, по этим коммитам можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной и не оценивается.

Рубежные контроли пишутся в аудитории индивидуально по варианту задания, выданному преподавателем в назначенные дни. При отсутствии студента в день написания контрольной работы ему дается еще один шанс ее написать на последнем занятии в семестре, но обязательно очно.

Студенты должны заранее сообщать о том, что у них могут возникнуть трудности со своевременной сдачей задания или проекта. При наличии реальных причин задержки студентам следует как можно скорее связаться с преподавателем и обсудить возможные условия.

7.2 Шкала и критерии оценивания результатов обучения

Лабораторная работа оценивается в процентах степени выполнения следующих критериев и для выставления оценки суммируются проценты за каждый из четырех критериев:

1. Полнота выполнения практического задания (30%): соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.

2. Качество и структура кода (10%): качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.

3. Творчество и инновации (10%): творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.

4. Ответы на вопросы по коду студента и теории (50%):

Дает краткий ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неправильно (10% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает неверно (20% из 50%)

Дает развернутый ответ, содержащий ошибки или неточности. На наводящие вопросы отвечает правильно (30% из 50%)

Дает правильные и развернутые ответы на вопросы (50% из 50%).

R лабораторные рассчитывается как среднее результатов за все лабораторные работы. За полное и безошибочное выполнение всех лабораторных работ в срок и их защиту можно получить максимум 100 баллов (R лабораторные).

Рубежный контроль оценивается по следующим критериям:

Полнота выполнения практического задания: соответствует ли функциональность заданным требованиям и целям, насколько точно и без ошибок код выполняет поставленные задачи.

Качество и структура кода: качество, читаемость и организация кода, рациональность выполнения задания, последовательность именования и соблюдение лучших практик.

Творчество и инновации: творческий подход студентов к выполнению заданий, насколько студенты вышли за рамки основных требований и реализовали дополнительные возможности или использовали уникальные решения.

Пользовательский опыт: отзывчивость, доступность, насколько эффективно задание отвечает требованиям целевой аудитории и обеспечивает приятное восприятие.

Самостоятельность решения: в репозитории студента есть коммиты только за его авторством, по коммитам в репозитории можно проследить как велась работа, студент может объяснить свой код и ход выполнения работы, если эти правила не соблюдаются, то работа не считается сданной.

Более подробное описание критериев дается в тексте задания рубежного контроля.

За полностью выполненные рубежные контроли также можно получить 100 баллов (R контроль).

Также имеется коэффициент сданных работ K сданные, который равен 1 если все работы сданы и 0 если хотя бы одна работа не сдана.

Итоговый балл рассчитывается по формуле: $R_{\text{сем}} = R_{\text{лабораторные}} * K_{\text{сданные}}$.

Итоговый балл пересчитывается по шкале ниже и на основании полученной оценки фиксируется результат промежуточной аттестации.

Соответствие баллов в 100 балльной рейтинговой системе оценке по 4-балльной шкале:

- **«Отлично» (85-100 баллов):** Все работы выполнены в срок, код оптимизирован, пайплайны работают без ошибок, студент глубоко понимает принципы работы инструментов.
- **«Хорошо» (70-84 балла):** Работы выполнены с незначительными замечаниями, пайплайн работает, но есть недочеты в конфигурации или безопасности.
- **«Удовлетворительно» (55-69 баллов):** Работы выполнены с помощью преподавателя, базовый функционал работает, но отсутствуют продвинутые практики (например, нет тестов в CI).
- **«Неудовлетворительно» (<55 баллов):** Критические ошибки в конфигурации, непонимание принципов контейнеризации или отсутствие работ.

7.3 Оценочные средства

7.3.1 Промежуточная аттестация

Примерный список вопросов:

- В чем разница между Continuous Integration, Continuous Delivery и Continuous Deployment?
- Опишите структуру Dockerfile. В чем преимущество Multi-stage builds?
- Чем контейнер отличается от виртуальной машины?
- Как разрешать конфликты слияния в Git?
- Что такое Artifact и где он хранится в процессе CI/CD?
- Опишите стратегии безопасного развертывания (Blue-Green, Canary).
- Для чего нужны Health Checks в Kubernetes/Docker?
- Что такое Infrastructure as Code и какие проблемы это решает?
- Как организовать хранение секретов (паролей, ключей) в пайплайне?
- Принципы работы Ansible: Inventory, Playbook, Module, Role.