

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 16:00:03
Уникальный программный ключ:
990bea3aeec548c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ
канд. физ.-мат. наук, доцент,
Новикова Е.Н.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Междисциплинарный проектный практикум-4

Направление подготовки
Направленность (профиль)

09.03.04 Программная инженерия
Разработка и сопровождение
программного обеспечения

Год начала обучения
Форма обучения
Реализуется в семестре

2026
очная
7

Разработано

Заведующая межинститутской базовой
кафедрой департамента цифровых,
робототехнических систем и электроники
(должность разработчика)
Новикова Елена Николаевна
Ф.И.О.

Ставрополь 2026 г.

1. Цель и задачи освоения дисциплины (модуля)

Сформировать у студентов комплексные практические компетенции по сквозной разработке, интеграции, тестированию, развертыванию и сопровождению промышленного высоконагруженного программного продукта с доказанными характеристиками надёжности, производительности и наблюдаемости (observability) в условиях командной работы с применением DevOps-практик, контейнеризации, оркестрации, CI/CD, мониторинга, а также интеграции компонентов искусственного интеллекта и обеспечения информационной безопасности

Задачи освоения дисциплины:

1. Организовать командную разработку по Agile (Scrum/Kanban) с Git Flow, Jira/Trello и оценкой трудоёмкости.
2. Спроектировать высоконагруженную распределённую систему (микросервисы/модульный монолит) с отказоустойчивостью (Circuit Breaker, Retry, Timeout, Fallback).
3. Реализовать backend (Java/Python/Go/C#) и frontend (React/Vue), асинхронное взаимодействие (Kafka/RabbitMQ), распределённое кэширование (Redis Cluster).
4. Провести unit, интеграционные, E2E, нагрузочные (p95/p99/RPS) и chaos-тесты (отказы узлов, задержки).
5. Настроить CI/CD (с SAST/DAST), контейнеризацию (Docker), оркестрацию (K8s с HPA), мониторинг (Prometheus+Grafana), логи (Loki), трейсинг (Jaeger), алерты (SLO/SLI).
6. Обеспечить горизонтальное масштабирование, репликацию БД, балансировку, автоматический failover и резервное копирование.
7. Интегрировать готовую ML-модель с асинхронным инференсом, батчированием и similarity-кэшированием.
8. Реализовать JWT + OAuth2, rate limiting, защиту от XSS/SQLi, пентест (OWASP ZAP), соблюдение Ф3-152.
9. Оформить архитектурные ADR, OpenAPI, руководство пользователя, презентацию и итоговый отчёт (рус./англ.).

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Управление жизненным циклом инженерных систем» относится к части, формируемой участниками образовательных отношений.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код и формулировка компетенции	Код индикатора и формулировка	Планируемый результат обучения по дисциплине
ПК-1 Способен осуществлять управление программными проектами: планировать, контролировать сроки и ресурсы, применять системы контроля версий и инструменты управления	ИД-3пк-1: Применяет инструменты управления задачами (Jira, Trello, YouTrack) для отслеживания прогресса (burndown chart), ведения бэклога и контроля исполнения	Способен организовать командную разработку по Agile (Scrum/Kanban) с использованием Git Flow, Jira/Trello/YouTrack, декомпозицией задач и

Код и формулировка компетенции	Код индикатора и формулировка	Планируемый результат обучения по дисциплине
задачами.		оценкой трудоёмкости в story points
ПК-2 Способен выполнять проектирование и разработку архитектуры программных систем среднего и крупного масштаба с применением современных паттернов и методов проектирования.	ИД-1пк-2: Выделяет и анализирует функциональные и нефункциональные требования (FURPS+), выбирает архитектурный стиль (монолит, микросервисы, EDA, Clean Architecture) на основе сценариев качества	Способен спроектировать высоконагруженную распределённую систему (микросервисную/модульную) в нотации C4, обосновать выбор паттернов (Saga, CQRS, Circuit Breaker) и рассчитать требуемые ресурсы
ПК-3 Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы, писать код на современных языках программирования, создавать веб- и desktop-приложения, работать с базами данных.	ИД-2пк-3: Создаёт клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	Способен реализовать backend на промышленном языке (Java/Python/Go/C#), frontend на React/Vue, асинхронное взаимодействие (Kafka/RabbitMQ), распределённое кэширование (Redis Cluster) и WebSocket-уведомления
ПК-4 Способен проводить тестирование, отладку и оценивать качество программного обеспечения, применяя инструменты автоматизированного	ИД-3пк-4: Проводит нагрузочное, стресс-тестирование и оценку надёжности (MTBF, MTTR) с применением chaos-инжиниринга, статического	Способен провести unit-, интеграционные, E2E-тесты, нагрузочное тестирование (k6/JMeter) и chaos-тестирование (отказы

Код и формулировка компетенции	Код индикатора и формулировка	Планируемый результат обучения по дисциплине
тестирования и методы обеспечения надежности и безопасности.	(SAST) и динамического (DAST) анализа безопасности	узлов, задержки) для проверки отказоустойчивости
ПК-5 Способен осуществлять сопровождение, настройку, развертывание и эксплуатацию программного обеспечения, в том числе в средах контейнеризации и облачных платформах.	ИД-3пк-5: Настраивает CI/CD пайплайны (GitLab CI, GitHub Actions) для автоматической сборки, тестирования и развёртывания приложений	Способен настроить CI/CD пайплайн с SAST/DAST, контейнеризацию (Docker), оркестрацию (K8s с HPA), автомасштабирование и резервное копирование
ПК-6 Владеет методами работы с данными и информационными системами: способен проектировать базы данных, выполнять анализ и обработку данных, интегрировать ПО с внешними системами.	ИД-2пк-6: Выполняет интеграцию с внешними API, брокерами сообщений (Kafka, RabbitMQ) и legacy-системами, обеспечивая идемпотентность и отказоустойчивость	Способен интегрировать внешние API, очереди сообщений, обеспечить репликацию БД и шардирование для масштабируемости
ПК-7 Способен применять современные информационные технологии, инструменты и платформы (включая отечественные), использовать DevOps-практики, облачные сервисы и средства автоматизации.	ИД-3пк-7: Применяет GitOps-подходы (ArgoCD), Helm и операторы Kubernetes для автоматизации развёртывания в кластере	Способен применить GitOps (ArgoCD), Helm, настроить мониторинг (Prometheus+Grafana+Loki) и распределённый трейсинг (Jaeger) для обеспечения наблюдаемости системы
ПК-8 Способен адаптировать и применять методы	ИД-3пк-8: Интегрирует предобученные модели и API	Способен интегрировать готовую ML-модель с

Код и формулировка компетенции	Код индикатора и формулировка	Планируемый результат обучения по дисциплине
искусственного интеллекта и машинного обучения для решения профессиональных задач, включая интеграцию готовых моделей в программные продукты.	ИИ (Hugging Face, OpenAI, YOLO) в программные продукты, разворачивает их как микросервисы и оптимизирует инференс (кэширование, батчинг)	асинхронным инференсом, батчированием и similarity-кэшированием (FAISS, косинусная близость)
ПК-9 Умеет оформлять техническую документацию, готовить презентации и отчеты, вести деловую коммуникацию на русском и иностранном языках в профессиональной среде.	ИД-1пк-9: Оформляет техническую документацию (ТЗ, пояснительная записка, руководство пользователя, API-спецификация) в форматах Markdown, OpenAPI, DocBook в соответствии с ГОСТ	Способен оформить архитектурные ADR, OpenAPI, руководство пользователя, итоговый отчет и презентацию (на русском и английском языке) для защиты проекта
ПК-10 Готов соблюдать правовые нормы, стандарты и требования информационной безопасности, участвовать в разработке нормативной и технической документации, учитывать законодательство в области ИТ.	ИД-2пк-10: Реализует меры защиты информации: аутентификацию (JWT, OAuth2), авторизацию (RBAC), шифрование, защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF, IDOR) с учётом требований регуляторов (ФСТЭК)	Способен реализовать JWT + OAuth2, rate limiting (Token Bucket/GCRA), защиту от XSS/SQLi, пентест (OWASP ZAP) и соблюдение требований ФЗ-152

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 3 з.е. 144 ак.ч.	ОФО, в ак. часах
Контактная работа:	54
Лекции/из них практическая подготовка	
Лабораторных работ/из них практическая подготовка	54/54
Практических занятий/из них практическая подготовка	

Самостоятельная работа	54
Формы контроля	
Экзамен	
Зачет	
Зачет с оценкой	
Расчетно-графические работы	
Курсовые работа	
Контрольные работы	

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием количества часов и видов занятий

№	Раздел (тема) дисциплины и краткое содержание	Формируемые компетенции и индикаторы	очная форма			
			Контактная работа обучающихся с преподавателем /из них в форме практической подготовки, часов			Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	
1	Тема: Настройка Kubernetes кластера и деплой первого микросервиса. Установка minikube/k3s, создание Docker-образа веб-приложения, написание манифестов Deployment (3 реплики) и Service типа ClusterIP, развертывание приложения, rolling update и откат версии, настройка Ingress (опционально). ЛАБОРАТОРНАЯ РАБОТА: Настройка Kubernetes кластера и деплой первого микросервиса	ИД-2пк-3 ИД-3пк-5			2/2	2
2	Тема: API Gateway + Service Discovery. Поднятие Service Discovery (Consul/Eureka), регистрация двух микросервисов (каталог, корзина), настройка API Gateway (Spring Cloud Gateway/Kong) с маршрутизацией /catalog/** и /cart/**, добавление фильтров (AddRequestHeader, логирование), балансировка нагрузки Round Robin/Weighted. ЛАБОРАТОРНАЯ РАБОТА: API Gateway + Service Discovery	ИД-1пк-2, ИД-2пк-6			2/2	2
3	Тема: Rate Limiting (Token Bucket / GCRA) в API Gateway. Развертывание API Gateway (Kong/Spring Cloud Gateway/Nginx), настройка rate limiting с Redis, реализация алгоритмов Token Bucket/GCRA с RPS и burst, возврат HTTP 429 при превышении лимита, разные лимиты для разных маршрутов (публичные 100 req/min, авторизованные 1000 req/min), нагрузочное тестирование (wrk/bombardier). ЛАБОРАТОРНАЯ РАБОТА: Rate Limiting (Token Bucket / GCRA) в API Gateway	ИД-2пк-10 ИД-3пк-4			2/2	2

4	Тема: Асинхронное взаимодействие через Kafka (producer/consumer). Развертывание Kafka (один брокер, KRaft), создание топика orders с 3 партициями, реализация producer на Spring Boot/Python/Go, consumer с логированием, настройка consumer group для параллельной обработки, обеспечение at-least-once (commit offset после обработки), запуск нескольких экземпляров consumer. ЛАБОРАТОРНАЯ РАБОТА: Асинхронное взаимодействие (Kafka, at-least-once)	ИД-2пк-3, ИД-2пк-6			2/2	2
5	Тема: Exactly-once доставка в Kafka. Включение enable.idempotence=true для producer, проверка отсутствия дубликатов, реализация транзакционной записи (initTransaction → send → commit/abort), настройка consumer с isolation.level=read_committed, демонстрация что сообщения из прерванной транзакции не читаются, транзакционное обновление consumer offset (sendOffsetsToTransaction). ЛАБОРАТОРНАЯ РАБОТА: Exactly-once доставка в Kafka	ИД-2пк-3, ИД-2пк-6			2/2	2
6	Тема: Circuit Breaker + Retry + Timeout (Resilience4j). Написание микросервиса, вызывающего нестабильный внешний API, включение Circuit Breaker (3 ошибки за 10 запросов → OPEN на 5 сек), добавление Retry (3 попытки, экспоненциальная задержка 1,2,4 сек), установка Timeout (500 ms), реализация fallback (статический ответ или кэш), тестирование поведения при сбоях, экспорт метрик в Prometheus ЛАБОРАТОРНАЯ РАБОТА: Circuit Breaker + Retry + Timeout (Resilience4j)	ИД-1пк-2, ИД-2пк-10			2/2	2
7	Тема: Bulkhead и RateLimiter на уровне кода. Написание сервиса, вызывающего медленный внешний API (имитация задержки 1-2 сек), ограничение параллельных вызовов через SemaphoreBulkhead (максимум 3), настройка RateLimiter (10 вызовов/сек), выбрасывание исключения с fallback при превышении, тестирование параллельных запросов (20 одновременных вызовов), добавление метрик в Prometheus. ЛАБОРАТОРНАЯ РАБОТА: Bulkhead и RateLimiter на уровне кода	ИД-2пк-10 ИД-3пк-8			2/2	2

8	Тема: Распределённое кэширование с Redis Cluster. Запуск Redis Cluster (3 мастера) в Docker Compose, реализация сервиса с паттерном Cache-Aside (сначала кэш, при промахе — БД и заполнение кэша), настройка TTL (например, 5 минут), инвалидация кэша при обновлении данных, измерение времени ответа с кэшем и без кэша, (опционально) реализация Write-Through. ЛАБОРАТОРНАЯ РАБОТА: Распределённое кэширование с Redis Cluster	ИД-2пк-3 ИД-2пк-6			2/2	2
9	Тема: Сессии и кэширование пользовательских данных. Реализация аутентификации (логин/пароль), генерация session ID и сохранение в Redis (user-id, срок действия, данные пользователя), отправка session ID через cookie или заголовок, middleware проверки сессии, реализация logout (удаление сессии), запуск двух экземпляров с балансировщиком, сравнение с JWT. ЛАБОРАТОРНАЯ РАБОТА: Сессии и кэширование пользовательских данных	ИД-2пк-3 ИД-2пк-10			2/2	2
10	Тема: Нагрузочное тестирование с k6 / JMeter. Подготовка тестового эндпоинта (GET /products, POST /orders), написание скрипта нагрузки на k6 с параметризацией, проведение теста с постепенным нарастанием нагрузки (ramp-up), замер RPS, p95, p99 latency, количества ошибок (HTTP 5xx), анализ отчёта, повтор теста после	ИД-3пк-4 ИД-3пк-5			2/2	2
	Оптимизации ЛАБОРАТОРНАЯ РАБОТА: Нагрузочное тестирование с k6 / JMeter					
11	Тема: Профилирование и оптимизация (async profiler). Написание CPU-интенсивной задачи (сортировка массива, шифрование) и I/O-bound задачи (много вызовов БД/HTTP), проведение профилирования (async-profiler/py-spy), построение флейм-графа, нахождение горячих точек, оптимизация кода (улучшение алгоритма, добавление кэша, асинхронность), сравнение результатов. ЛАБОРАТОРНАЯ РАБОТА: Профилирование и оптимизация (async profiler)	ИД-3пк-4, ИД-3пк-8			2/2	2

12	Тема: Мониторинг (Prometheus + Grafana). Добавление в приложение метрик (RPS, ошибки, гистограмма времени ответа), экспорт метрик в формате Prometheus (/actuator/prometheus), настройка Prometheus (scrape config), построение дашборда в Grafana (RPS, p95 latency, ошибки), настройка алерта (ошибки >1% за минуту), проведение нагрузочного теста и наблюдение за метриками. ЛАБОРАТОРНАЯ РАБОТА: Мониторинг (Prometheus + Grafana)	ИД-3пк-7 ИД-3пк-4			2/2	2
13	Тема: Централизованное логирование (Loki + Promtail). Поднятие Loki и Promtail (Docker/K8s), настройка Promtail для сбора логов из stdout и файлов, добавление структурирования логов (JSON) с полями timestamp, level, service, message, correlation_id, выполнение запросов для генерации логов, поиск логов по correlation ID в Grafana, настройка алерта (при появлении ERROR). ЛАБОРАТОРНАЯ РАБОТА: Централизованное логирование (Loki + Promtail)	ИД-3пк-7 ИД-3пк-4			2/2	2
14	Тема: Распределённый трейсинг (Jaeger). Добавление в приложение OpenTelemetry SDK, настройка экспорта трейсов в Jaeger (HTTP/gRPC), генерация запросов через 2-3 сервиса, анализ трассировки в интерфейсе Jaeger (спаны, задержки), определение узкого места, добавление атрибутов в спаны (http.method, http.status, user.id). ЛАБОРАТОРНАЯ РАБОТА: Распределённый трейсинг (Jaeger)	ИД-3пк-7 ИД-3пк-4			2/2	2
15	Тема: CI/CD пайплайн с SAST/DAST (GitHub Actions / GitLab CI). Создание репозитория с веб-приложением, настройка CI пайплайна при каждом push (сборка, модульные тесты, SAST (SonarQube/Semgrep), DAST (OWASP ZAP) против тестового стенда), установка порогов качества (покрытие >80%, отсутствие критических уязвимостей), настройка автоматического деплоя на staging, тестирование пайплайна с внесением уязвимости (SQL инъекция). ЛАБОРАТОРНАЯ РАБОТА: CI/CD с SAST и DAST (GitHub Actions)	ИД-3пк-5 ИД-3пк-4 ИД-2пк-10			2/2	2
16	Тема: SAST-анализ кода с SonarQube. Запуск SonarQube (Docker/локально), подключение проекта (Java/Python/Go/JS), настройка Quality Gate (покрытие >80%, дублирование <3%, критических уязвимостей = 0), анализ кода и	ИД-3пк-4 ИД-3пк-5 ИД-2пк-10			2/2	2

	исправление проблем, интеграция SonarQube в CI (GitHub Actions/GitLab CI). ЛАБОРАТОРНАЯ РАБОТА: SAST-анализ кода с SonarQube					
17	Тема: DAST-тестирование с OWASP ZAP. Развертывание тестового веб-приложения с уязвимостями (SQL injection, XSS, CSRF), настройка OWASP ZAP, проведение базового сканирования (Baseline Scan), активное сканирование (Active Scan) с подстановкой параметров, анализ отчёта, исправление уязвимостей (параметризованные запросы, экранирование, CSRF-токен), повторное сканирование. ЛАБОРАТОРНАЯ РАБОТА: DAST-тестирование с OWASP ZAP	ИД-3пк-4 ИД-2пк-10			2/2	2
18	Тема: Автомасштабирование в Kubernetes (HPA). Развертывание приложения с известной нагрузкой в K8s, настройка Metrics Server для сбора метрик CPU, создание HPA с target CPU = 50%, запуск нагрузочного теста (k6/JMeter), наблюдение за увеличением числа подов, (дополнительно) настройка HPA по кастомной метрике RPS через Prometheus adapter. ЛАБОРАТОРНАЯ РАБОТА: Автомасштабирование в Kubernetes (HPA)	ИД-3пк-5 ИД-3пк-7 ИД-3пк-8			2/2	2
19	Тема: Event-driven автомасштабирование (KEDA). Установка KEDA в кластер, развертывание Kafka/RabbitMQ и consumer, написание ScaledObject для отслеживания lag в Kafka (max threshold = 50), масштабирование consumer от 1 до 10 подов, генерация нагрузки (много сообщений), наблюдение за масштабированием, уменьшение подов при снижении нагрузки. ЛАБОРАТОРНАЯ РАБОТА: Event-driven автомасштабирование (KEDA)	ИД-3пк-5 ИД-3пк-7 ИД-3пк-8			2/2	2
20	Тема: GitOps с ArgoCD. Установка ArgoCD (в minikube/k3s), создание Git-репозитория с манифестами (Deployment, Service, ConfigMap), регистрация приложения в ArgoCD с указанием репозитория и пути, настройка автоматической синхронизации (sync policy: automated), изменение манифеста (новый тег образа) в Git и проверка применения изменений, включение self-heal. ЛАБОРАТОРНАЯ РАБОТА: GitOps с ArgoCD	ИД-3пк-1 ИД-3пк-5 ИД-3пк-7			2/2	2

21	Тема: Резервное копирование и восстановление PostgreSQL. Развертывание PostgreSQL (Docker/K8s), настройка бэкапа через pg_dump и cron (ежедневно), включение WAL-архивации (в S3 или локальную папку), имитация сбоя (удаление таблицы), восстановление базы на момент до сбоя (PITR), настройка автоматической ротации бэкапов. ЛАБОРАТОРНАЯ РАБОТА: Резервное копирование и восстановление PostgreSQL	ИД-2пк-6 ИД-3пк-5			2/2	2
22	Тема: Полный Highload-стек в Docker Compose. Создание docker-compose.yml для: веб-приложения (бэкенд), PostgreSQL/ClickHouse, Redis (кэш), Kafka + Zookeeper/RabbitMQ, Nginx (reverse proxy), настройка сетей (bridge) и volumes для персистенции данных, добавление healthcheck для каждого сервиса, нагрузочный тест через k6. ЛАБОРАТОРНАЯ РАБОТА: Полный Highload-стек в Docker Compose	ИД-1пк-2 ИД-2пк-3 ИД-2пк-6 ИД-3пк-5			2/2	2
23	Тема: Развёртывание в облаке (Yandex Cloud / VK Cloud). Создание managed-кластера K8s (2-3 узла), настройка kubectl для доступа, развертывание приложения с помощью манифестов или Helm, настройка Ingress-контроллера (Nginx Ingress) и получение SSL-сертификата (Let's Encrypt) через cert-manager, создание Service типа LoadBalancer для публичного доступа, проверка доступности по HTTPS. ЛАБОРАТОРНАЯ РАБОТА: Деплой в облачный managed-кластер Kubernetes	ИД-3пк-5 ИД-3пк-7 ИД-2пк-10			2/2	2
24	Тема: Комплексный проект: Highload-маркетплейс (часть 1 – архитектура и инфраструктура) Проектирование микросервисной системы (каталог, корзина, заказы, оплата, нотификации), выбор технологического стека (Java/Spring Boot/Python/Go), создание Git-репозитория, настройка CI/CD пайплайна (GitHub Actions/GitLab CI) для сборки и публикации Docker-образов, развертывание инфраструктуры (K8s, БД, Redis, Kafka), написание базовых манифестов K8s. ЛАБОРАТОРНАЯ РАБОТА: Комплексный проект: Highload-маркетплейс (часть 1 – проектирование)	ИД-1пк-2 ИД-2пк-3 ИД-3пк-1 ИД-3пк-5 ИД-3пк-7 ИД-2пк-6 ИД-2пк-10 ИД-3пк-8 ИД-1пк-9			2/2	2

25	Тема: Комплексный проект: Highload-маркетплейс (часть 2 – бизнес-логика). Реализация сервиса «Каталог» (GET /products, GET /products/{id}), сервиса «Корзина» (POST /cart, GET /cart, DELETE /cart), сервиса «Заказы» (POST /orders, GET /orders), интеграция через Kafka (событие «заказ создан» → сервис оплаты), добавление JWT-аутентификации, написание docker-compose для локального запуска всех сервисов. ЛАБОРАТОРНАЯ РАБОТА: Комплексный проект: Highload-маркетплейс (часть 2 – бизнес-логика)	ИД-1пк-2, ИД-2пк-3, ИД-2пк-6, ИД-2пк-10, ИД-3пк-8, ИД-1пк-9			2/2	2
26	Тема: Комплексный проект: Highload-маркетплейс (часть 3 – нагрузочное тестирование и наблюдаемость). Написание сценария нагрузочного тестирования (k6) для эндпоинтов (каталог, корзина, заказы), проведение теста (ramp-up до 200 VU), замер p95/p99 latency, сбор метрик приложений (Micrometer/prometheus_client), построение дашбордов в Grafana, настройка централизованного сбора логов (Loki) и трейсинга (Jaeger), создание алертов (p99 latency >500ms, ошибки >1%), повторный тест после оптимизации. ЛАБОРАТОРНАЯ РАБОТА: Комплексный проект: Highload-маркетплейс (часть 3 – нагрузочное тестирование и наблюдаемость)	ИД-3пк-4 ИД-3пк-7 ИД-1пк-9			2/2	2
27	Тема: Комплексный проект: защита и документация. Оформление архитектурных решений (ADR – минимум 3), спецификация API (OpenAPI/Swagger) для каждого микросервиса, документация по развёртыванию (локальный docker-compose и облачный K8s), подготовка презентации (7-10 слайдов): архитектура, технологии,	ИД-1пк-9 ИД-3пк-1 ИД-1пк-2 ИД-3пк-4 ИД-3пк-7 ИД-2пк-10			2/2	2
	нагрузочное тестирование, результаты оптимизации, защита проекта перед комиссией (демонстрация работающего приложения и дашбордов мониторинга).					
	ЛАБОРАТОРНАЯ РАБОТА: Комплексный проект: защита и документация					
	Зачет с оценкой					
	ИТОГО за 7 семестр				54/54	54
	ИТОГО				54/54	54

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств (ФОС) по дисциплине (модулю) базируется на перечне осваиваемых компетенций с указанием индикаторов. ФОС обеспечивает объективный контроль достижения запланированных результатов обучения. ФОС включает в себя:

- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
 - методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций (включаются в методические указания по тем видам работ, которые предусмотрены учебным планом и предусматривают оценку сформированности компетенций);
 - типовые оценочные средства, необходимые для оценки знаний, умений и уровня сформированности компетенций.
- ФОС является приложением к данной программе дисциплины.

7. Методические указания для обучающихся по освоению дисциплины

Приступая к работе, каждый студент должен принимать во внимание следующие положения.

Дисциплина построена по тематическому принципу, каждая тема представляет собой логически завершённый раздел.

Практические занятия проводятся с целью закрепления усвоенной информации, приобретения навыков ее применения при решении практических задач в соответствующей предметной области.

Самостоятельная работа студентов направлена на самостоятельное изучение дополнительного материала, подготовку к лабораторным занятиям, а также выполнения всех видов самостоятельной работы.

Для успешного освоения дисциплины, необходимо выполнить все виды самостоятельной работы, используя рекомендуемые источники информации.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

8.1.1. Перечень основной литературы:

1. Ньюмен С. Проектирование микросервисной архитектуры. – СПб.: Питер, 2022. – Доступно в ЭБС «Лань», «Юрайт».
2. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. – М.: ДМК Пресс, 2021. – Доступно в ЭБС «ZNANIUM».
3. Клеппманн М. Высоконагруженные приложения. – СПб.: Питер, 2022. – Доступно в ЭБС «Лань».
4. Кьюб Д. Kubernetes в действии. – М.: ДМК Пресс, 2020. – Доступно в ЭБС «Айбукс».
5. Кохлс К. Docker на практике. – СПб.: Питер, 2021. – Доступно в ЭБС «Юрайт».

8.1.2. Перечень дополнительной литературы:

1. Грубер М. SQL. Полное руководство. – М.: Вильямс, 2021. – Доступно в ЭБС «Лань».
2. Штайнер Й. Безопасность веб-приложений. – СПб.: Питер, 2020. – Доступно в ЭБС «Юрайт».
3. Кноберг А. SCRUM. Революционный метод управления проектами. – М.: Манн, Иванов и Фербер, 2020. – Печатное издание.
4. Вайпан В.А. (под ред.) Актуальные вопросы информационного и цифрового права. – М.: Юстицинформ, 2026. – Печатное издание.
5. Официальная документация Docker. – Режим доступа: docs.docker.com. – Свободный доступ.

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

1. Методические указания по выполнению лабораторных работ (ЛР №1–27). – Размещены в ЭИОС университета (PDF формат).
2. Методические указания к самостоятельной работе по дисциплине. – Размещены в ЭИОС университета (PDF формат).

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Национальная электронная библиотека (НЭБ): <https://нэб.рф>
2. ЭБС «Лань»: <https://e.lanbook.com>
3. ЭБС «Юрайт»: <https://urait.ru>
4. ЭБС «ZNANIUM»: <https://znanium.com>
5. Официальная документация PostgreSQL (рус.): <https://postgrespro.ru/docs>
6. Официальная документация Docker (англ.): <https://docs.docker.com>
7. Официальная документация Kubernetes (рус.): <https://kubernetes.io/ru/docs>
8. Портал «Отечественный софт» (реестр российского ПО): <https://reestr.digital.gov.ru>

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При чтении лекций используется компьютерная техника, демонстрации презентационных мультимедийных материалов. На семинарских и практических занятиях студенты представляют презентации, подготовленные ими в часы самостоятельной работы.

Информационные справочные системы:

Информационно-справочные и информационно-правовые системы, используемые при изучении дисциплины:

Программное обеспечение:

Операционные системы

1. Astra Linux (ОСНОВА) – российская ОС, включена в реестр, бесплатно для образования.
2. РЕД ОС – российская ОС, бесплатно для образования.
3. Альт Образование – российская ОС, бесплатно для образования.
4. Ubuntu / Debian Linux – свободное ПО (Open Source), полностью доступно.

5. Windows 10/11 – платное ПО (лицензия ОС), доступно, обновления ограничены.
- Средства разработки (IDE)
6. OpenIDE – российская IDE (форк IntelliJ IDEA), бесплатно (AGPLv3).
 7. VS Code – свободное ПО (Open Source), полностью доступно.
 8. PyCharm Community Edition – свободное ПО (Open Source), полностью доступно.
 9. IntelliJ IDEA Community Edition – свободное ПО (Open Source), полностью доступно.
 10. Eclipse – свободное ПО (Open Source), полностью доступно.
- Системы контроля версий и CI/CD
11. Git – свободное ПО (GPL), полностью доступно.
 12. GitLab (self-managed) – свободное ПО, можно установить на свои сервера.
 13. Gitea – свободное ПО (Open Source), полностью доступно.
 14. Jenkins – свободное ПО (Open Source), полностью доступно.
 15. GitHub Actions – бесплатно (ограничения), данные уходят за рубеж, используется с осторожностью.
- Базы данных и очереди сообщений
16. PostgreSQL – свободное ПО (PostgreSQL License), полностью доступен.
 17. Redis – свободное ПО (BSD), полностью доступен.
 18. Apache Kafka – свободное ПО (Apache 2.0), полностью доступен.
 19. RabbitMQ – свободное ПО (MPL 2.0), полностью доступен.
 20. ClickHouse – свободное ПО (Apache 2.0), российская разработка.
 21. MySQL – свободное ПО (GPL), полностью доступен.
- Контейнеризация и оркестрация
22. Docker Engine – свободное ПО, полностью доступен.
 23. Podman – свободное ПО (Apache 2.0), полностью доступен.
 24. Deckhouse (Флант) – российская K8s-платформа, бесплатно (Community Edition).
 25. Платформа «Боцман» (Группа Астра) – российская K8s-платформа, бесплатно для образования.
 26. Minikube – свободное ПО (Apache 2.0), полностью доступен.
- Инструменты управления проектами и задачами
27. SimpleOne SDLC – российская система управления жизненным циклом разработки, платная / демо-доступ.
 28. YouTrack – бесплатно до 5 пользователей, разработка JetBrains, доступен.
 29. Jira – платное ПО, зарубежное, продажи ограничены.
 30. Trello – бесплатно (ограничения), зарубежное ПО.
- Мониторинг и наблюдаемость
31. Prometheus – свободное ПО (Apache 2.0), полностью доступен.
 32. Grafana – свободное ПО (AGPLv3), полностью доступен.
 33. Loki – свободное ПО (AGPLv3), полностью доступен.
 34. Jaeger – свободное ПО (Apache 2.0), полностью доступен.
- Инструменты нагрузочного тестирования
35. k6 – свободное ПО (AGPLv3), полностью доступен.
 36. JMeter – свободное ПО (Apache 2.0), полностью доступен.
 37. Безопасность (SAST/DAST)
 38. SonarQube (Community Edition) – свободное ПО (LGPLv3), полностью доступен.
 39. OWASP ZAP – свободное ПО (Apache 2.0), полностью доступен.
 40. Semgrep – свободное ПО (LGPLv2), полностью доступен.
- Офисное ПО и работа с документацией
41. МойОфис – российское ПО, бесплатно для образования.

42. LibreOffice – свободное ПО (LGPLv3), полностью доступен.

43. ONLYOFFICE – свободное ПО (AGPLv3), доступен.

Языки программирования и среды выполнения

44. Java (OpenJDK, Axiom JDK) – свободное ПО, Axiom JDK – российская сборка.

45. Python – свободное ПО (PSF License), полностью доступен.

46. Go – свободное ПО (BSD-3), полностью доступен.

47. Node.js – свободное ПО (MIT), полностью доступен.

48. C# (.NET SDK) – свободное ПО (MIT), полностью доступен.

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Лабораторные работы	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Самостоятельная работа	Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и возможностью доступа к электронной информационно-образовательной среде университета

11. Особенности освоения дисциплины (модуля) лицами с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины (модуля) обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),

- письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

- специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

- индивидуальное равномерное освещение не менее 300 люкс,

- при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочесть и оформить задание, в том числе, записывая под диктовку),
- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;
- обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата (в том числе с тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
- по желанию студента задания могут выполняться в устной форме.

12. Особенности реализации дисциплины с применением дистанционных образовательных технологий и электронного обучения

Согласно части 1 статьи 16 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» под *электронным обучением* понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. Под *дистанционными образовательными технологиями* понимаются образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Реализация дисциплины может быть осуществлена с применением дистанционных образовательных технологий и электронного обучения полностью или частично. Компоненты УМК дисциплины (рабочая программа дисциплины, оценочные и методические материалы, формы аттестации), реализуемой с применением дистанционных образовательных технологий и электронного обучения, содержат указание на их использование.

При организации образовательной деятельности с применением дистанционных образовательных технологий и электронного обучения могут предусматриваться асинхронный и синхронный способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет».

При применении дистанционных образовательных технологий и электронного обучения в расписании по дисциплине указываются: способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет» (ВКС-видеоконференцсвязь, ЭТ – электронное тестирование); ссылки на электронную информационно-образовательную среду СКФУ, на образовательные платформы и ресурсы иных организаций, к которым предоставляется открытый доступ через информационно-телекоммуникационную сеть «Интернет»; для синхронного обучения - время проведения онлайн-занятий и преподаватели; для асинхронного обучения - авторы онлайн-курсов.

При организации промежуточной аттестации с применением дистанционных образовательных технологий и электронного обучения используются Методические рекомендации по применению технических средств, обеспечивающих объективность результатов при проведении промежуточной и государственной итоговой аттестации по

образовательным программам высшего образования - программам бакалавриата, программам специалитета и программам магистратуры с применением дистанционных образовательных технологий (Письмо Минобрнауки России от 07.12.2020 г. № МН-19/1573-АН "О направлении методических рекомендаций").

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется с использованием электронной информационно-образовательной среды СКФУ, к которой обеспечен доступ обучающихся через информационно-телекоммуникационную сеть «Интернет», или с использованием ресурсов иных организаций, в том числе платформ, предоставляющих сервисы для проведения видеоконференций, онлайн-встреч и дистанционного обучения (МТС-Линк), а также с использованием возможностей социальных сетей для осуществления коммуникации обучающихся и преподавателей.

Учебно-методическое обеспечение дисциплины, реализуемой с применением электронного обучения и дистанционных образовательных технологий, включает представленные в электронном виде рабочую программу, учебно-методические пособия или курс лекций, методические указания к выполнению различных видов учебной деятельности обучающихся, предусмотренных дисциплиной, и прочие учебно-методические материалы, размещенные в информационно-образовательной среде СКФУ.