

Документ подписан простой электронной подписью
Информация о владельце: **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**
ФИО: Порохня Андрей Алексеевич **РОССИЙСКОЙ ФЕДЕРАЦИИ**
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 14:50:30 **Федеральное государственное автономное образовательное учреждение**
Уникальный программный ключ: **высшего образования**
990bea3aeec48c23bd8699e48288f8d1960c600 **«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

УТВЕРЖДАЮ
И.о. директора института
перспективной инженерии канд. техн.
наук, доцент Порохня А.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Системная интеграция и управление конфигурациями

Направление подготовки	09.04.04 Программная инженерия
Направленность (профиль)	Разработка, мониторинг и управление жизненным циклом инженерных систем
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	1

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Системная интеграция и управление конфигурациями».

Разработчик Е.Н. Новикова, к.ф.-м.н., доцент, заведующий межинститутский базовой кафедрой департамента цифровых, робототехнических систем и электроники.

Проведена экспертиза ФОС.

Председатель Новикова Е.Н., межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Члены комиссии:

Азаров И.В., и.о. директора департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Николаев Е. И., доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Представитель организации-работодателя:

Руководитель группы разработки А.А. Шипулин, ООО «Стилсофт».

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-2 - Способен применять принципы "Инфраструктура как код" и "Конфигурация как код" для автоматизированного, идиоматичного, безопасного и версионизируемого управления облачными, гибридными средами и конфигурациями парков автономных устройств с использованием современных инструментов				
ИД-1 ПК-2. Описывает облачные ресурсы, сетевые компоненты и вычислительные мощности на декларативных языках конфигураций (HCL, YAML, JSON) с использованием инструментов класса IaC (Terraform, CloudFormation, Pulumi).	Не может описать облачные ресурсы декларативными языками; не знаком с инструментами IaC; не различает HCL, YAML, JSON; не может создать базовый Terraform-файл.	С помощью преподавателя создаёт простейшие декларативные описания (один ресурс, например, виртуальную машину) на одном из языков (HCL). Допускает синтаксические ошибки.	Самостоятельно описывает несколько взаимосвязанных облачных ресурсов (VPC, подсеть, виртуальную машину, диск) на HCL в Terraform. Использует переменные и outputs.	Уверенно описывает сложную облачную инфраструктуру (модули, data sources, remote state, provisioners) на Terraform или Pulumi. Выбирает оптимальные языки и инструменты под конкретную задачу.
ИД-2 ПК-2. Разрабатывает идиоматичные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем, прикладного ПО и бортового программного обеспечения автономных	Не понимает концепции идиоматичности; не знаком с синтаксисом Ansible или других CaC-инструментов; не может написать playbook.	Пишет простейший playbook для установки одного пакета (например, nginx) с помощью преподавателя. Идиоматичность не обеспечивает.	Пишет идиоматичный playbook для настройки веб-сервера с использованием переменных, шаблонов Jinja2 и handlers. Использует роли (roles).	Разрабатывает сложные, идиоматичные и хорошо структурированные сценарии управления конфигурациями для ОС, прикладного ПО и бортового ПО дронов. Использует Ansible Vault для секретов и динамический инвентарь.

устройств.				
ИД-3 ПК-2. Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций.	Не знает основных команд Git; не использует систему контроля версий; не может создать коммит или клонировать репозиторий.	Выполняет базовые операции Git (init, add, commit, push, pull) с помощью подсказок. Не соблюдает модель ветвления.	Самостоятельно использует Git в командной работе: создаёт ветки, делает merge, разрешает простые конфликты. Соблюдает модель ветвления (Git Flow / GitHub Flow).	Уверенно работает с Git: использует rebase, cherry-pick, stash, submodules. Разрабатывает стратегию ветвления для команды. Проводит code review.
ИД-4 ПК-2. Интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки (CI/CD).	Не знаком с CI/CD концепциями; не может настроить пайплайн.	С помощью преподавателя создаёт простой CI/CD пайплайн (например, запуск Ansible playbook по триггеру из Git) в одной из платформ (GitLab CI / GitHub Actions).	Самостоятельно настраивает CI/CD пайплайн с несколькими стадиями (build, test, deploy) для автоматического развертывания инфраструктуры (Terraform) и применения конфигураций (Ansible).	Настраивает сложные CI/CD пайплайны с параллельным выполнением заданий, кэшированием зависимостей, динозавриком (динамическими средами), публикацией артефактов и автоматическим откатом при сбое тестов.
ИД-5 ПК-2. Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий.	Не понимает риски безопасности IaC; хранит секреты в открытом виде в коде; не использует инструменты сканирования	Использует Ansible Vault для шифрования паролей (с помощью преподавателя). Не сканирует код на уязвимости.	Самостоятельно настраивает сканирование Terraform-кода (checkov / tfsec) и Dockerfile (trivy) в CI/CD. Использует HashiCorp Vault для управления секретами (KV engine). Следует принципу минимальных привилегий в IAM политиках.	Внедряет полный DevSecOps цикл: статический анализ безопасности кода (SAST), сканирование уязвимостей контейнеров, управление динамическими секретами (Vault Database Engine), регулярную ротацию секретов и аудит доступа.
ИД-6 ПК-2. Реализует GitOps-подход для автоматической синхронизации состояния	Не знаком с GitOps-подходом; не понимает различия между push и pull моделями	С помощью преподавателя настраивает базовый GitOps цикл (например, ArgoCD для	Самостоятельно настраивает GitOps цикл с использованием ArgoCD (или Flux) для	Внедряет GitOps для всей инфраструктуры (многоокруженное развертывание, multi-cluster).

инфраструктуры с конфигурацией в Git-репозитории и устранения дрейфа конфигураций.	доставки.	одного приложения) в тестовой среде.	синхронизации состояния Kubernetes кластера с Git репозиторием. Понимает концепцию дрейфа конфигураций.	Настраивает автоматическое применение конфигураций, верификацию состояния (health checks) и автоматический откат. Использует механизмы устранения дрейфа.
ПК-10 - Способен проектировать, реализовывать и администрировать системы централизованного управления парками автономных аппаратов, включая диспетчеризацию задач, мониторинг состояния, управление конфигурациями, массовые обновления "по воздуху" и интеграцию со сторонними сервисами				
ИД-1 ПК-10. Проектирует архитектуру платформы централизованного управления парками автономных устройств с учетом требований масштабируемости, отказоустойчивости и безопасности.	Не может предложить архитектуру управления парком устройств; не учитывает требования к масштабируемости.	С помощью преподавателя рисует базовую архитектуру из 2-3 компонентов (например, шлюз приема телеметрии, БД, панель управления).	Проектирует распределённую архитектуру платформы управления парком устройств, выделяя ключевые компоненты (API Gateway, брокер сообщений, хранилище телеметрии, сервис команд, сервис конфигураций). Указывает протоколы взаимодействия.	Проектирует отказоустойчивую, горизонтально масштабируемую архитектуру для управления тысячами устройств с учётом сегментации сети, шифрования, многоуровневой безопасности и географического распределения.
ИД-2 ПК-10. Реализует механизмы диспетчеризации и планирования задач для групп устройств, включая оптимизацию маршрутов и распределение ресурсов.	Не понимает задачи диспетчеризации.	Реализует простейший планировщик задач (FIFO) без учёта состояния устройств (с помощью преподавателя).	Реализует механизм диспетчеризации задач с учётом приоритетов и состояния устройств (online/offline, уровень заряда). Использует простую стратегию распределения (round-robin или load-based).	Реализует сложный планировщик задач с оптимизацией маршрутов (для мобильных устройств), динамическим перераспределением ресурсов при сбоях, учётом географического положения и прогнозированием нагрузки.
ИД-3 ПК-10. Организует сбор, обработку, хранение и визуализацию	Не может организовать сбор телеметрии.	С помощью преподавателя настраивает сбор телеметрии через REST API раз в	Самостоятельно настраивает потоковый сбор телеметрии по протоколам MQTT	Организует высокопроизводительный пайплайн потоковой обработки

потокowej телеметрии устройств в реальном времени с системой оповещения об аномалиях.		минуту и визуализацию в простой таблице.	/ WebSocket. Использует InfluxDB / ClickHouse для хранения временных рядов и Grafana для дашбордов. Настраивает простые алерты на превышение порогов.	телеметрии (например, Kafka Streams / Flink). Реализует обнаружение аномалий с помощью статистических методов или ML моделей. Настраивает систему алертинга (в том числе предиктивного).
ИД-4 ПК-10. Разрабатывает систему централизованного управления конфигурациями устройств с контролем версий и автоматическим применением настроек.	Не может предложить способ управления конфигурациями устройств.	С помощью преподавателя реализует статическую конфигурацию устройств (файл конфигурации).	Реализует централизованный сервис управления конфигурациями с версионированием (например, на основе Git). Устройства периодически опрашивают сервис на предмет изменений.	Реализует полноценную систему управления конфигурациями устройств с поддержкой канареечного развёртывания конфигураций, автоматическим откатом при ошибках и интеграцией с GitOps.
ИД-5 ПК-10. Внедряет механизмы массовых OTA-обновлений (по воздуху) бортового ПО, обеспечивающие надежность, целостность и безопасность доставки.	Не внедряет OTA обновления.	Реализует базовое OTA обновление (загрузка прошивки по HTTP) с проверкой целостности через хеш (MD5)	Внедряет OTA обновления с цифровой подписью прошивок (GPG / Cosign), стратегией канареечного развёртывания (5-10% устройств) и возможностью отката.	Внедряет комплексную систему OTA обновлений с A/B разделами, дельта-патчами, верификацией через TEE (аппаратный secure boot), автоматическим тестированием прошивки в симуляторе перед выкаткой и мониторингом успешности обновления.
ИД-6 ПК-10. Обеспечивает интеграцию платформы управления парком со сторонними сервисами	Не интегрирует платформу со сторонними сервисами.	С помощью преподавателя реализует простую интеграцию с одним внешним API (например, получение погоды) без учёта	Самостоятельно реализует защищённую интеграцию с 2-3 сторонними сервисами (например, картографический	Реализует отказоустойчивые, безопасные интеграции с любыми стандартизованными интерфейсами (REST, gRPC,

(картографические системы, API аналитики, облачные хранилища) через защищенные интерфейсы.		безопасности.	API, сервис уведомлений) с использованием API ключей или JWT, с обработкой ошибок (retry, fallback).	GraphQL) с аутентификацией OAuth2/mTLS, использованием паттернов Circuit Breaker и Rate Limiting, а также мониторингом состояния интеграций.
ИД-7 ПК-10. Администрирует инфраструктуру платформы управления, обеспечивая ее бесперебойную работу, масштабирование и защиту от несанкционированного доступа.	Не администрирует платформу; не обеспечивает её безопасность.	С помощью преподавателя запускает платформу в Docker Compose	Самостоятельно администрирует платформу: разворачивает в Kubernetes, настраивает горизонтальное автомасштабирование (HPA), базовую сетевую изоляцию (Network Policies), резервное копирование баз данных.	Полностью администрирует производственную платформу управления парками: многоокруженное развёртывание (dev/staging/prod), automatable (GitOps) восстановление после сбоев (DRP), защиту от DDoS, продвинутую сетевую безопасность (mTLS, WAF), аудит доступа и соответствие требованиям регуляторов.
ИД-8 ПК-10. Управляет жизненным циклом устройств в парке: регистрация, мониторинг состояния, обновление, вывод из эксплуатации.	Не может описать жизненный цикл устройства.	С помощью преподавателя регистрирует устройство вручную (например, через конфигурационный файл).	Самостоятельно реализует автоматическую регистрацию устройств (через API с верификацией), ведёт учёт состояния устройств (online/offline, версия прошивки), поддерживает возможность вывода устройства из эксплуатации (дерегистрация	Реализует полностью автоматизированный жизненный цикл устройств: саморегистрация с аттестацией (mTLS сертификаты), отслеживание состояния в реальном времени (Heartbeat, телеметрия), автоматическое OTA обновление с политиками, graceful вывод из эксплуатации (передача задач, безопасное стирание данных).

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

№	Правильный ответ	Содержание вопроса	Компетенция
1	С	Какой подход предполагает описание и управление инфраструктурой (сервера, сети, БД) с помощью декларативных конфигурационных файлов, хранящихся в системе контроля версий? A) DevOps B) Agile C) Infrastructure as Code (IaC) D) Continuous Deployment	ПК-2 (ИД-1)
2	А	Какой язык используется в Terraform для декларативного описания облачных ресурсов и инфраструктуры? A) HCL (HashiCorp Configuration Language) B) YAML C) Python D) XML	ПК-2 (ИД-1)
3	В	Согласно методическим указаниям, какой инструмент из класса IaC используется для автоматизации развертывания и управления конфигурациями серверов (агентлесс архитектура)? A) Terraform B) Ansible C) Puppet D) Pulumi	ПК-2 (ИД-2)
4	Д	Какое ключевое свойство сценариев управления конфигурациями (playbook) в Ansible позволяет	ПК-2 (ИД-2)

№	Правильный ответ	Содержание вопроса	Компетенция
		применять их многократно без риска внесения нежелательных изменений в уже настроенную систему? А) Асинхронность В) Компилируемость С) Декларативность D) Идемпотентность	
5	А	В какой файловой структуре Ansible содержится логически сгруппированный набор tasks, handlers, templates и vars для переиспользования? А) Роль (Role) В) Инвентарь (Inventory) С) Плейбук (Playbook) D) Модуль (Module)	ПК-2 (ИД-2)
6	В	Для какой цели в Ansible используются Handlers? А) Для определения переменных окружения В) Для выполнения действий (например, перезапуска сервиса) только в случае, если задача сообщила об изменении состояния системы С) Для шифрования секретных данных D) Для описания структуры управляемых узлов	ПК-2 (ИД-2)
7	С	Какая команда Git используется для переноса изменений из одной ветки в другую, создавая отдельный коммит слияния? А) git rebase	ПК-2 (ИД-3)

№	Правильный ответ	Содержание вопроса	Компетенция
		B) git cherry-pick C) git merge D) git fetch	
8	A	Какой подход к автоматизации развертывания подразумевает, что Git-репозиторий является единственным источником истины (single source of truth) для состояния всей системы, а изменения применяются автоматически специальным оператором? A) GitOps B) ChatOps C) NoOps D) TestOps	ПК-2 (ИД-6)
9	D	Какой инструмент из описанных в методических указаниях используется для реализации GitOps-цикла и автоматической синхронизации состояния Kubernetes-кластера с конфигурацией в Git-репозитории? A) Jenkins B) GitLab Runner C) Terraform D) ArgoCD	ПК-2 (ИД-6)
10	B	Что из перечисленного НЕ является прямой обязанностью инструментов класса CI/CD (Continuous Integration/Continuous Delivery) согласно материалам курса? A) Автоматическая сборка приложения B) Написание кода бизнес-логики	ПК-2 (ИД-4)

№	Правильный ответ	Содержание вопроса	Компетенция
		С) Запуск интеграционных тестов Д) Развертывание артефактов в целевые среды	
11	А	Какой паттерн проектирования используется в Apache Camel для маршрутизации сообщений на основе их содержимого, позволяя направить сообщение в разные очереди в зависимости от типа заказа (STANDARD, EXPRESS)? А) Content-Based Router В) Aggregator С) Wire Tap Д) Dead Letter Channel	ПК-10 (ИД-6)
12	С	Какой HTTP метод является идемпотентным и используется в REST API для полного замещения ресурса? А) POST В) PATCH С) PUT Д) DELETE	ПК-10 (ИД-6)
13	В	Для чего в системной интеграции используется паттерн "Circuit Breaker"? А) Для увеличения скорости передачи данных В) Для предотвращения каскадных отказов при недоступности внешнего сервиса С) Для шифрования трафика Д) Для балансировки нагрузки между серверами	ПК-10 (ИД-6)

№	Правильный ответ	Содержание вопроса	Компетенция
14	A	Какой протокол чаще всего используется для обмена телеметрией в реальном времени с автономными устройствами (дроны, датчики) благодаря легковесности и работе поверх TCP/IP? A) MQTT B) AMQP C) SNMP D) ICMP	ПК-10 (ИД-3)
15	D	Согласно методическим указаниям, какая система используется для централизованного сбора метрик и хранения данных в виде временных рядов? A) Grafana B) Jaeger C) Loki D) Prometheus	ПК-10 (ИД-3)
16	B	Как называется процесс в Debezium, который отслеживает изменения в журнале транзакций (WAL) PostgreSQL и передает их в Kafka? A) Data Scrubbing B) Change Data Capture (CDC) C) Log Shipping D) Continuous Query	ПК-2 (ИД-1)
17	C	Какой из этих инструментов является API Gateway и используется для маршрутизации запросов, аутентификации и rate limiting? A) Apache Kafka	ПК-10 (ИД-6)

№	Правильный ответ	Содержание вопроса	Компетенция
		B) Ansible C) Kong D) Apache Airbyte	
18	A	Какая команда Git позволяет временно "спрятать" незакоммиченные изменения, чтобы переключиться на другую ветку? A) git stash B) git reset C) git archive D) git bundle	ПК-2 (ИД-3)
19	D	В чем основное преимущество gRPC перед REST/HTTP в контексте высоконагруженной интеграции микросервисов? A) Использование текстового формата JSON B) Встроенная поддержка HTML C) Отсутствие необходимости в версионировании D) Использование бинарного протокола Protocol Buffers и HTTP/2 (мультиплексирование)	ПК-10 (ИД-6)
20	B	Какой инструмент используется в лабораторных работах для распределенной трассировки запросов (distributed tracing) между микросервисами? A) Prometheus B) Jaeger C) Elasticsearch D) Logstash	ПК-10 (ИД-3)

№	Правильный ответ	Содержание вопроса	Компетенция
21	С	Что такое "Dead Letter Queue" (DLQ) в контексте очередей сообщений (RabbitMQ)? А) Очередь с самым высоким приоритетом В) Очередь для хранения резервных копий сообщений С) Очередь для сообщений, которые не удалось обработать после нескольких попыток D) Очередь, используемая только для системных логов	ПК-2 (ИД-2)
22	В	В какой тип обменника (exchange) RabbitMQ сообщение направляется во ВСЕ привязанные очереди, игнорируя routing key? А) Direct В) Fanout С) Topic D) Headers	ПК-2 (ИД-1)
23	Д	Какой инструмент ETL/ELT (Extract, Transform, Load) упоминается в МУ для интеграции данных между различными источниками и приемниками с веб-интерфейсом? А) Logstash В) dbt С) Spark D) Apache Airbyte	ПК-10 (ИД-6)
24	А	Какая команда Terraform применяется для вычисления того, какие действия необходимо выполнить для приведения инфраструктуры в соответствие с конфигурационными файлами (без фактического применения)? А) terraform plan	ПК-2 (ИД-4)

№	Правильный ответ	Содержание вопроса	Компетенция
		B) terraform apply C) terraform validate D) terraform state	
25	С	Какой механизм в адаптерах для внешних API подразумевает увеличение задержки (например, 1с, 2с, 4с) между повторными попытками вызова при сбое? A) Circuit Breaker B) Rate Limiting C) Exponential Backoff D) Bulkhead	ПК-10 (ИД-6)
26	В	Что такое OTA (Over-The-Air) обновления в контексте управления парками автономных устройств? A) Обновление антивирусных баз на сервере B) Удаленное обновление прошивки и программного обеспечения устройств по беспроводной сети C) Обновление с помощью USB-накопителя D) Обновление лицензионного соглашения	ПК-10 (ИД-5)
27	А	Какая модель ветвления в Git предполагает наличие двух "вечных" веток (main/master и develop) и краткосрочных веток для фич, релизов и хотфиксов? A) Git Flow B) GitHub Flow C) Trunk Based Development	ПК-2 (ИД-3)

№	Правильный ответ	Содержание вопроса	Компетенция
		D) GitLab Flow	
28	C	Какой протокол используется для двунаправленной, полнодуплексной связи в реальном времени между браузером и сервером, часто применяемый в трейдинговых платформах или онлайн-чатах? A) FTP B) SMTP C) WebSocket D) SNMP	ПК-10 (ИД-3)
29	D	Согласно материалам, для чего используется HashiCorp Vault в контексте безопасности инфраструктуры? A) Для мониторинга нагрузки на сервер B) Для балансировки трафика C) Для хранения кода приложений D) Для централизованного хранения и управления секретами (пароли, токены, ключи)	ПК-2 (ИД-5)
30	B	Какой инструмент статического анализа кода инфраструктуры (IaC) упоминается для поиска уязвимостей в Terraform манифестах? A) SonarQube B) Checkov (или tfsec) C) ESLint D) Pylint	ПК-2 (ИД-5)

№	Правильный ответ	Содержание вопроса	Компетенция
31	C	Что такое "Consumer Group" в Apache Kafka? A) Группа топиков, объединенных для авторизации B) Группа брокеров, обслуживающих один топик C) Группа консьюмеров, которые совместно читают данные из топика, распределяя партиции между собой D) Группа продюсеров, отправляющих данные в один топик	ПК-10 (ИД-2)
32	A	Какой режим синхронизации в ETL-процессах подразумевает полную выгрузку всех данных из источника при каждом запуске? A) Full Refresh B) Incremental C) Append D) CDC	ПК-2 (ИД-1)
33	D	Какая функция API Gateway защищает бэкенд-сервисы от чрезмерного количества запросов от одного клиента? A) Аутентификация B) Маршрутизация C) Кэширование D) Rate Limiting	ПК-10 (ИД-6)
34	B	Какая стратегия развертывания обновлений подразумевает постепенное перенаправление небольшого процента пользователей на новую версию сервиса для минимизации рисков? A) Blue-Green Deployment	ПК-10 (ИД-5)

№	Правильный ответ	Содержание вопроса	Компетенция
		B) Canary Deployment C) Recreate D) A/B Testing	
35	С	Что из нижеперечисленного НЕ является типом данных (data type) в Protocol Buffers (protobuf), используемом в gRPC? A) int32 B) string C) array (используется ключевое слово repeated) D) enum	ПК-10 (ИД-6)
36	А	Какой тип брокера сообщений описан в МУ как распределенная платформа для потоковой передачи событий с высокой пропускной способностью, использующая концепцию логов? A) Apache Kafka B) RabbitMQ C) ActiveMQ D) Redis Pub/Sub	ПК-10 (ИД-2)
37	В	Какой формат данных чаще всего используется в REST API для обмена данными между клиентом и сервером? A) XML B) JSON C) YAML D) CSV	ПК-10 (ИД-6)

№	Правильный ответ	Содержание вопроса	Компетенция
38	D	Что делает директива <code>import</code> в Terraform? A) Загружает переменные окружения B) Выполняет bash скрипт C) Загружает модуль из реестра D) Добавляет существующую инфраструктуру в состояние Terraform	ПК-2 (ИД-4)
39	B	В чем заключается принцип "минимальных привилегий" (Least Privilege) при управлении доступом в облачных средах? A) Предоставление прав администратора всем пользователям для ускорения работы B) Предоставление пользователю или сервису ровно тех прав, которые необходимы для выполнения его функций, и не более C) Периодическая смена всех паролей доступа раз в месяц D) Использование только однофакторной аутентификации	ПК-2 (ИД-5)
40	A	Как называется компонент платформы управления парками автономных устройств, отвечающий за оптимизацию маршрутов и распределение задач между дронами? A) Диспетчеризация задач (Task Dispatching) B) Система телеметрии C) KPI Dashboard D) Data Lake	ПК-10 (ИД-2)
41	C	Для чего в CI/CD пайплайнах используются артефакты (artifacts)? A) Для хранения исходного кода B) Для управления версиями зависимостей	ПК-2 (ИД-4)

№	Правильный ответ	Содержание вопроса	Компетенция
		С) Для передачи результатов сборки (jar, exe, отчеты) между этапами пайплайна D) Для авторизации пользователей	
42	D	Какая технология используется для симуляции работы внешних зависимостей (БД, очередей) внутри Docker-контейнеров при написании интеграционных тестов? A) Mockito B) Postman C) Selenium D) Testcontainers	ПК-2 (ИД-4)
43	B	Какой паттерн интеграции (EIP) описан в ЛР №5 и используется для разделения составного сообщения на несколько частей для параллельной обработки? A) Aggregator B) Splitter C) Router D) Filter	ПК-10 (ИД-6)
44	A	В лабораторной работе №8 для реализации real-time уведомлений и чата используется библиотека <u>Socket.io</u> на Node.js. Какой протокол лежит в её основе? A) WebSocket B) HTTP 1.1 C) FTP D) SMTP	ПК-10 (ИД-3)

№	Правильный ответ	Содержание вопроса	Компетенция
45	В	Какой статус HTTP (код ответа) возвращает сервер при успешном создании нового ресурса (например, при POST-запросе) согласно спецификации REST? A) 200 OK B) 201 Created C) 202 Accepted D) 204 No Content	ПК-10 (ИД-6)
46	С	Как называется файл, в котором в GitLab CI описываются этапы пайплайна (stages), задания (jobs) и скрипты их выполнения? A) Jenkinsfile B) Makefile C) .gitlab-ci.yml D) docker-compose.yml	ПК-2 (ИД-4)
47	Д	Какие метрики относятся к группе "Золотых сигналов" (Google SRE Book) для мониторинга системы? A) Температура CPU, загрузка диска B) Количество строк кода, число коммитов C) Версия ОС, тип процессора D) Латентность, трафик, ошибки, насыщение (Latency, Traffic, Errors, Saturation)	ПК-10 (ИД-3)
48	А	Какой из следующих сценариев управления конфигурациями является идемпотентным? A) Скрипт, который проверяет, установлен ли пакет, и устанавливает его только при отсутствии B) Скрипт, который всегда перезаписывает конфигурационный файл, даже если он не изменился	ПК-2 (ИД-2)

№	Правильный ответ	Содержание вопроса	Компетенция
		С) Скрипт, который перезагружает сервер после каждого запуска Д) Скрипт, который добавляет строку в конец файла при каждом запуске	
49	С	Какой тип данных в MQTT используется для определения уникальности источника данных (например, <code>/drone/123/temperature</code>)? A) Payload B) Client ID C) Topic D) Session	ПК-10 (ИД-3)
50	В	Что подразумевается под термином "Observability" (Наблюдаемость) в контексте современных интеграционных платформ? A) Способность системы включаться и выключаться без помощи администратора B) Способность системы предоставлять достаточно данных (метрики, логи, трассы), чтобы понять её внутреннее состояние, даже если проблема возникла впервые C) Наличие веб-интерфейса у системы D) Автоматическое резервное копирование данных каждую ночь	ПК-10 (ИД-3)

Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала, оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

Критерии оценивания компетенций

Оценка «отлично» (5)

выставляется студенту, если он:

1. Полно и без ошибок демонстрирует знание теоретических основ системной интеграции, IaC (Terraform, Pulumi), CaC (Ansible), принципов CI/CD, GitOps, архитектуры платформ управления парками автономных устройств, протоколов интеграции (REST, gRPC, WebSocket, MQTT) и паттернов отказоустойчивости (Circuit Breaker, Retry).
2. Уверенно и самостоятельно разрабатывает декларативные описания инфраструктуры (HCL, YAML) и идемпотентные сценарии управления конфигурациями (Ansible playbooks/roles) для настройки ОС, прикладного ПО и бортового ПО автономных устройств, обеспечивая повторяемость и безопасность.
3. В полном объёме выполняет лабораторные работы (уровень сложности 3) и сквозной проект, демонстрируя:
4. работоспособность интеграционной платформы (REST/gRPC/GraphQL API, очереди сообщений, API Gateway, ETL/CDC, адаптеры внешних API);
5. настроенную систему мониторинга и трассировки (Prometheus + Grafana + Jaeger);
6. реализованные механизмы безопасности (JWT, OAuth2, mTLS) и управления секретами (HashiCorp Vault);
7. работающий CI/CD пайплайн с интеграционными тестами и автоматическим развёртыванием.
8. Грамотно и без ошибок отвечает на контрольные вопросы, использует профессиональную терминологию и свободно ориентируется в технологическом стеке (Python/Go/Java, Kong/Nginx, RabbitMQ/Kafka, PostgreSQL/ClickHouse, Git, Docker, Kubernetes).
9. Предлагает и обосновывает оптимальные архитектурные решения, выходящие за рамки базового задания, демонстрирует системное мышление и способность к анализу и устранению проблем интеграции.

Оценка «хорошо» (4)

выставляется студенту, если он:

1. Демонстрирует хорошие знания теоретического материала, но допускает незначительные неточности или неуверенность в отдельных вопросах.
2. В целом успешно разрабатывает декларативные описания инфраструктуры и сценарии управления конфигурациями (уровень сложности 2), обеспечивая идемпотентность и версионирование, но может испытывать трудности с оптимизацией или отладкой сложных сценариев.
3. Выполняет лабораторные работы и сквозной проект в полном объёме для уровня сложности 2, демонстрируя:
4. работающее интеграционное решение (2–3 сервиса, очередь, API Gateway);
5. базовый мониторинг (Prometheus + Grafana) без трассировки;

6. базовую безопасность (JWT или API Key) без mTLS/OAuth2;
7. CI/CD пайплайн с интеграционными тестами и публикацией отчётов.
8. Допускает отдельные ошибки в ответах на контрольные вопросы или при демонстрации работы, но исправляет их самостоятельно или с помощью наводящих вопросов.
9. Умеет использовать систему контроля версий Git для совместной работы, но может испытывать затруднения с разрешением сложных конфликтов или GitOps-подходом.

Оценка «удовлетворительно» (3)

выставляется студенту, если он:

1. Демонстрирует минимально достаточные знания теории для выполнения базовых заданий, допускает существенные пробелы в понимании ключевых концепций (например, не понимает разницу между синхронной и асинхронной интеграцией, не различает IaC и CaC).
2. С помощью преподавателя разрабатывает простейшие декларативные описания и сценарии конфигурации (уровень сложности 1), но не способен обеспечить полную идемпотентность или безопасность.
3. Выполняет лабораторные работы и сквозной проект с ошибками, исправляемыми только после наводящих вопросов преподавателя, при этом:
4. интеграционное решение работает с перебоями или не в полном объёме;
5. мониторинг и трассировка отсутствуют или настроены некорректно;
6. безопасность не реализована или реализована с грубыми нарушениями;
7. CI/CD пайплайн отсутствует или нестабилен.
8. Допускает грубые ошибки в ответах на контрольные вопросы, не владеет профессиональной терминологией, не ориентируется в изученных инструментах.
9. Не умеет эффективно использовать Git (не соблюдает модель ветвления, допускает конфликты, не умеет их разрешать).

Оценка «неудовлетворительно» (2)

выставляется студенту, если он:

1. Не демонстрирует знание основных теоретических разделов дисциплины (не может объяснить, что такое системная интеграция, принципы REST API, цели управления конфигурациями).
2. Не разработал ни одного работоспособного сценария управления конфигурациями или описания инфраструктуры, не понимает синтаксиса HCL/YAML и назначения инструментов IaC/CaC.
3. Не выполнил лабораторные работы в минимально необходимом объёме (уровень сложности 1) или представил неработоспособные решения, которые не могут быть исправлены наводящими вопросами преподавателя.
4. Допускает принципиальные ошибки при ответах на контрольные вопросы, не может объяснить назначение изученных протоколов, инструментов и паттернов интеграции.
5. Не умеет применять Git для хранения и версионирования кода, не выполнил командную работу в репозитории, не представил отчёт по проекту.

Оценка	Уровень сложности 1 (макс. баллов)	Уровень сложности 2 (макс. баллов)	Уровень сложности 3 (макс. баллов)
5 (отлично)	90–100%	90–100%	90–100%
4 (хорошо)	70–89%	70–89%	70–89%
3	50–69%	50–69%	50–69%

Оценка	Уровень сложности 1 (макс. баллов)	Уровень сложности 2 (макс. баллов)	Уровень сложности 3 (макс. баллов)
(удовлетворительно)			
2 (неудовлетворительно)	<50%	<50%	<50%