

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
Уникальный программный ключ:
990bea3aeec48c23bd8699e48288f8d1960c600

Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Инфраструктура как код и платформенная инженерия

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	2

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Инфраструктура как код и платформенная инженерия».

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, идемпотентного, 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 способен описать инфраструктуру на декларативных языках. Допускает грубые ошибки в синтаксисе HCL/YAML. Не ориентируется в базовых конструкциях Terraform.	С помощью преподавателя описывает простые конфигурации. Знает базовый синтаксис, но допускает ошибки. Может создать одиночный ресурс, но не модуль.	Самостоятельно описывает инфраструктуру среднего уровня сложности. Использует переменные, outputs, data sources. Понимает модульную организацию, но допускает неточности.	Свободно описывает сложную инфраструктуру. Разрабатывает переиспользуемые модули. Оптимизирует конфигурации, применяет best practices. Аргументирует выбор инструментов.
ИД-2 ПК-2 Разрабатывает идемпотентные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем, прикладного ПО и бортового ПО автономных устройств	Не знает принципов идемпотентности. Не может написать простой плейбук Ansible. Допускает ошибки в синтаксисе YAML и модулях.	С помощью преподавателя пишет простые идемпотентные сценарии. Понимает базовые модули Ansible, но допускает ошибки в структуре playbooks.	Самостоятельно разрабатывает идемпотентные плейбуки и роли. Использует handlers, templates, variables. Понимает принципы idempotency применительно к конфигурациям.	Разрабатывает сложные идемпотентные сценарии с условными операторами, циклами и обработкой ошибок. Адаптирует Ansible для настройки бортового ПО автономных устройств. Оптимизирует производительность.
ИД-3 ПК-2 Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	Не владеет базовыми командами Git. Не понимает принципов ветвления и слияния. Допускает грубые ошибки при разрешении конфликтов.	Выполняет базовые операции: clone, add, commit, push. Понимает концепцию ветвления, но разрешает конфликты только с помощью преподавателя.	Уверенно работает с Git: создаёт ветки, сливает их, разрешает типовые конфликты. Использует модели ветвления (Git Flow, GitHub Flow). Применяет теги для версионирования.	Свободно применяет Git для командной работы над IaC-проектами. Разрабатывает стратегии ветвления. Автоматизирует проверки через pre-commit хуки. Обучает других членов команды.

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 запуск Terraform в CI/CD. Допускает ошибки в конфигурации пайплайнов.	С помощью преподавателя настраивает базовый CI/CD пайплайн для Terraform plan. Понимает этапы, но Quality Gates не настраивает.	Самостоятельно интегрирует Terraform и Ansible в CI/CD (GitLab CI/GitHub Actions). Настраивает автоматический запуск при изменениях в репозитории. Реализует статический анализ (tfsec, checkov).	Разрабатывает сложные пайплайны с параллельными этапами, кэшированием, условным выполнением. Настраивает Quality Gates с блокировкой при нарушениях. Интегрирует OPA/Conftest в CI/CD.
ИД-5 ПК-2 Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	Не знает инструментов SAST для IaC. Не понимает рисков, связанных с секретами в коде. Допускает хранение паролей в открытом виде.	С помощью преподавателя настраивает базовое сканирование уязвимостей. Знает о необходимости управления секретами, но использует переменные окружения.	Самостоятельно настраивает Checkov/tfsec в CI/CD. Использует HashiCorp Vault для управления секретами. Понимает принцип минимальных привилегий	Внедряет комплексную безопасность: SAST/SCA, подпись артефактов (Cosign), генерацию SBOM. Настраивает политики доступа через OPA. Реализует принцип zero-trust для сервисных аккаунтов.
ИД-6 ПК-2 Реализует GitOps-подход с использованием инструментов (ArgoCD, Flux) для автоматической синхронизации состояния среды с Git-репозиторием и устранения дрейфа конфигураций	Не знаком с GitOps концепцией. Не может объяснить отличие GitOps от CI/CD. Не настраивал ArgoCD/Flux	Понимает базовые принципы GitOps. С помощью преподавателя устанавливает ArgoCD и настраивает синхронизацию одного приложения.	Самостоятельно настраивает ArgoCD/Flux для синхронизации состояния кластера с Git-репозиторием. Понимает механизмы устранения дрейфа. Настраивает automatic sync.	Разрабатывает полноценную GitOps-стратегию для управления инфраструктурой и приложениями. Настраивает multi-tenant ArgoCD, sync windows, health checks. Интегрирует OPA для политик. Обучает команду.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

№	Правильный ответ	Содержание вопроса	Компетенция
1	В	Что означает аббревиатура IaC в контексте управления инфраструктурой? A. Internet as Code B. Infrastructure as Code C. Integration as Code D. Interface as Code	ПК-2
2	A, B, D	Какие из перечисленных языков используются для декларативного описания инфраструктуры? A. HCL B. YAML C. Python D. JSON	ПК-2
3	С	Какой инструмент является основным для реализации Infrastructure as Code в экосистеме HashiCorp? A. Ansible B. Pulumi C. Terraform D. Packer	ПК-2
4	В	Что такое Terraform State? A. Файл с переменными окружения B. Файл, содержащий отображение между конфигурацией и реальной инфраструктурой C. Лог выполнения Terraform команд	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		D. Конфигурация провайдера	
5	A, C	Какие команды Terraform необходимо выполнить перед apply? (выберите два) A. terraform init B. terraform destroy C. terraform plan D. terraform output	ПК-2
6	A	Для чего используется команда <code>terraform fmt</code>? A. Для форматирования конфигурационных файлов в стандартном стиле B. Для форматирования state-файла C. Для форматирования логов выполнения D. Для создания нового модуля	ПК-2
7	B	Что такое идемпотентность в контексте управления конфигурациями? A. Возможность выполнять операции только один раз B. Свойство операции давать одинаковый результат при многократном применении C. Возможность отката изменений D. Свойство выполнять операции параллельно	ПК-2
8	B	Какой инструмент преимущественно используется для идемпотентного управления конфигурациями в открытом ПО? A. Terraform B. Ansible	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		C. Docker D. Kubernetes	
9	C	Что такое Ansible playbook? A. Файл инвентаризации хостов B. Конфигурационный файл Ansible C. Файл, описывающий набор операций для выполнения на хостах D. Лог выполнения Ansible	ПК-2
10	A	Каким принципом руководствуется Ansible для обеспечения повторяемости конфигураций? A. Идемпотентностью B. Декларативностью C. Императивностью D. Асинхронностью	ПК-2
11	D	Какая команда Git используется для создания новой ветки? A. git branch --new B. git new branch C. git checkout new D. git branch <имя_ветки>	ПК-2
12	C	Как разрешить конфликт слияния в Git? A. Использовать git reset --hard B. Использовать git rebase --skip	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		С. Вручную исправить конфликтующие строки и выполнить git add + git commit D. Использовать git merge --abort	
13	В	Что такое Git Flow? A. Способ хранения Git-репозитория B. Модель ветвления для управления релизами C. Утилита для оптимизации Git D. Протокол передачи данных	ПК-2
14	А	Для чего используются Git-теги? A. Для маркировки определённых версий кода (например, релизов) B. Для создания временных веток C. Для комментирования коммитов D. Для игнорирования файлов	ПК-2
15	В	Какой инструмент позволяет организовать GitOps-процесс в Kubernetes? A. Jenkins B. ArgoCD C. Prometheus D. Grafana	ПК-2
16	А	Что является единым источником истины (single source of truth) в GitOps? A. Git-репозиторий B. State-файл Terraform	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		С. База данных Kubernetes D. Конфигурационный файл Ansible	
17	В	Что такое дрейф конфигураций (configuration drift)? A. Автоматическое обновление конфигураций B. Отклонение реального состояния инфраструктуры от описанного в коде C. Оптимизация конфигурационных файлов D. Версионирование конфигураций	ПК-2
18	С	Как GitOps решает проблему дрейфа конфигураций? A. Автоматическим удалением отклоняющихся ресурсов B. Ручной синхронизацией C. Автоматической синхронизацией состояния кластера с Git-репозиторием D. Периодическими отчетами	ПК-2
19	В	Что такое Terragrunt? A. Инструмент для тестирования Terraform B. Обёртка над Terraform для создания DRY-конфигураций C. Альтернатива Terraform от Microsoft D. Хранилище Terraform state	ПК-2
20	С	Какой принцип реализует Terragrunt? A. KISS (Keep It Simple, Stupid) B. YAGNI (You Aren't Gonna Need It)	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		C. DRY (Don't Repeat Yourself) D. SOLID	
21	A	Для чего в Terragrunt используется блок <code>dependency</code>? A. Для объявления зависимости outputs одного модуля от другого B. Для добавления внешних зависимостей C. Для указания версии Terraform D. Для настройки провайдеров	ПК-2
22	B	Что такое Policy as Code? A. Хранение политик безопасности в виде PDF B. Подход, при котором правила и политики описываются кодом и автоматически проверяются C. Шифрование политик D. Политики, написанные на Python	ПК-2
23	A	Какой инструмент является основным для реализации Policy as Code в Kubernetes? A. OPA Gatekeeper B. Prometheus C. Ingress Controller D. Helm	ПК-2
24	C	Что такое ConstraintTemplate в OPA Gatekeeper? A. Конфигурация Kubernetes B. Инструмент для валидации	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		C. Шаблон, описывающий логику политики на Rego D. Метрика Gatekeeper	
25	A	Какой режим enforcementAction в Gatekeeper используется для тестирования без блокировки? A. dryrun B. deny C. warn D. audit	ПК-2 ПК-2 ПК-2 ПК-2
26	B	Что такое SLSA? A. Инструмент для логирования B. Фреймворк для обеспечения безопасности цепочки поставки программного обеспечения C. Язык программирования D. База данных	ПК-2
27	C	Для чего используется Cosign из экосистемы Sigstore? A. Для шифрования данных B. Для аутентификации пользователей C. Для подписи контейнерных образов D. Для сбора метрик	ПК-2
28	A	Что такое SBOM?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		A. Software Bill of Materials — список компонентов программного продукта B. Security Bill of Materials — список мер безопасности C. System Bill of Materials — список системных требований D. Service Bill of Materials — список сервисов	
29	В	Какой инструмент позволяет оценить стоимость инфраструктуры до деплоя? A. Terraform cost B. Infracost C. Cloud cost D. FinOps CLI	ПК-2
30	А	Что такое FinOps? A. Подход к управлению облачными затратами B. Финансовое программирование C. Безопасность финансовых операций D. Финансовый менеджмент в IT	ПК-2
31	С	Что такое ОТА-обновления? A. Обновления через USB B. Обновления по Bluetooth C. Обновления «по воздуху» для удалённых устройств D. Обновления в офисе	ПК-2
32	В	Какая стратегия ОТА-обновления предполагает сначала обновление 1% устройств?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		A. Rolling update B. Canary update C. Blue-green update D. A/B update	
33	А, С	Какие требования предъявляются к edge-устройствам при сборе телеметрии? (выберите два) A. Низкое энергопотребление B. Высокая вычислительная мощность C. Буферизация при потере связи D. Постоянное интернет-подключение	ПК-2
34	В	Для чего используется OpenTelemetry Collector на edge-устройствах? A. Для выполнения бизнес-логики B. Для сбора и отправки метрик, логов и трасс C. Для хранения данных D. Для управления устройством	ПК-2
35	Д	Какой протокол чаще всего используется для передачи телеметрии с устройств? A. FTP B. SMTP C. SNMP D. MQTT	ПК-2
36	С	Что такое API Gateway?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		А. База данных для API В. Инструмент для разработки API С. Единая точка входа для управления API-запросами к микросервисам и внешним сервисам D. Протокол API	
37	А	Какая платформа используется в качестве API Gateway в лабораторных работах? А. Kong В. Terraform С. Ansible D. ArgoCD	ПК-2
38	В	Что такое rate limiting? А. Ограничение скорости сети В. Ограничение количества запросов к API в единицу времени С. Ограничение размера данных D. Ограничение времени выполнения запроса	ПК-2
39	А	Какая команда Terraform используется для просмотра списка ресурсов в состоянии? А. terraform state list В. terraform list С. terraform show resources D. terraform resources	ПК-2
40	В	Что означает переменная <code>CI_COMMIT_REF_SLUG</code> в GitLab CI?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		А. Идентификатор коммита В. Уникальный идентификатор ветки или тега, приведённый к безопасному формату С. Номер пайплайна D. Ссылка на репозиторий	
41	С	Какой инструмент используется для статического анализа безопасности Terraform кода? А. Terraform validate В. Terraform lint С. Checkov / tfsec D. Terraform plan	ПК-2
42	А, С	Какие из перечисленных инструментов относятся к GitOps? (выберите два) А. ArgoCD В. Jenkins С. Flux D. Terraform	ПК-2
43	В	Что такое Karpenter? А. Инструмент для мониторинга Kubernetes В. Инструмент для автоматического провижининга узлов в Kubernetes с оптимизацией затрат С. Инструмент для деплоя приложений D. Система логирования	ПК-2
44	D	Какой backend Terraform поддерживает блокировки состояния?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		A. local B. http C. consul D. s3 (с DynamoDB)	
45	В	Для чего в Terraform используется <code>terraform taint</code> ? A. Для удаления ресурса B. Для пометки ресурса как проблемного, чтобы он был пересоздан при следующем apply C. Для добавления меток к ресурсу D. Для отката изменений	ПК-2
46	А	Что означает ключевое слово <code>locals</code> в Terraform? A. Для объявления локальных переменных внутри модуля B. Для указания региона ресурса C. Для импорта внешних данных D. Для настройки провайдера	ПК-2
47	С	Какой инструмент используется для централизованного управления секретами? A. Git Secrets B. Kubernetes Secrets C. HashiCorp Vault D. AWS KMS	ПК-2
48	В	Что такое mTLS?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
		А. Однонаправленная аутентификация В. Взаимная TLS-аутентификация, при которой и клиент, и сервер подтверждают свою подлинность С. Протокол шифрования данных Д. Сертификат безопасности	
49	А	Какая команда Terragrunt выполняет Terraform apply во всех модулях текущей директории? А. terragrunt run-all apply В. terragrunt apply-all С. terragrunt apply Д. terragrunt all apply	ПК-2
50	С	Что означает принцип «shift-left» в контексте безопасности? А. Сдвиг ответственности вправо В. Перенос тестирования в production С. Перенос проверок безопасности на ранние этапы разработки Д. Увеличение числа тестировщиков	ПК-2

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» выставляется студенту, если он:

1. полностью освоил компетенцию ПК-2 (ИД-1 – ИД-6);
2. демонстрирует глубокие теоретические знания принципов IaC, Policy as Code, GitOps и Platform Engineering;
3. самостоятельно выполняет все лабораторные работы (ЛР1-ЛР18) на высоком уровне, с соблюдением best practices;
4. свободно ориентируется в Terraform, Terragrunt, Ansible, OPA, Vault, Infracost, Karpenter, Cosign, ArgoCD/Flux, OpenTelemetry и API Gateway;
5. настраивает автоматизированные CI/CD пайплайны с Quality Gates и верификацией политик;
6. реализует GitOps-цикл и управление конфигурациями edge-устройств (ОТА, телеметрия, валидация);
7. защищает работы с аргументацией решений, отвечает на контрольные вопросы;
8. оформляет отчеты в соответствии с требованиями;
9. набирает не менее 90 баллов из 100 по итогам семестра.

2. Оценка «хорошо»

Оценка «хорошо» выставляется студенту, если он:

1. освоил компетенцию ПК-2 в основном, но допускает отдельные неточности, не носящие принципиального характера;
2. демонстрирует хорошие теоретические знания, но испытывает небольшие затруднения при объяснении сложных концепций;
3. выполнил не менее 14 лабораторных работ (из 18) в полном объеме;
4. уверенно работает с основными инструментами дисциплины, но допускает отдельные ошибки, которые исправляет самостоятельно при указании на них;
5. настраивает CI/CD пайплайны с Quality Gates, но возможны недочеты в конфигурации (например, отсутствие кэширования, неполное покрытие политиками);

6. реализует базовый GitOps-цикл (ArgoCD/Flux), но испытывает трудности при настройке автоматической синхронизации или устранении дрейфа;
7. выполняет задания по управлению устройствами (ОТА, телеметрия), но допускает незначительные ошибки в конфигурациях;
8. защищает работы, отвечает на большинство контрольных вопросов, но некоторые ответы требуют уточнений;
9. оформляет отчеты с незначительными нарушениями требований;
10. набирает от 75 до 89 баллов по итогам семестра.

3. Оценка «удовлетворительно»

Оценка «удовлетворительно» выставляется студенту, если он:

1. освоил компетенцию ПК-2 частично, на минимально достаточном уровне;
2. демонстрирует базовые теоретические знания, но испытывает значительные трудности при объяснении материала;
3. выполнил не менее 10 лабораторных работ (из 18) в минимальном объеме;
4. работает с основными инструментами только при наличии методических указаний и помощи преподавателя;
5. настраивает простые CI/CD пайплайны, но Quality Gates и статический анализ отсутствуют или настроены с ошибками;
6. имеет общее представление о GitOps, но не может самостоятельно настроить ArgoCD/Flux;
7. выполняет базовые задания по управлению устройствами, но допускает ошибки в ОРА-политиках или ОТА-конфигурациях;
8. защищает работы с большим количеством ошибок, отвечает менее чем на половину контрольных вопросов;
9. отчеты оформлены с нарушениями, отсутствуют отдельные обязательные разделы;
10. набирает от 60 до 74 баллов по итогам семестра.

4. Оценка «неудовлетворительно»

Оценка «неудовлетворительно» выставляется студенту, если он:

1. не освоил компетенцию ПК-2 (ИД-1 – ИД-6);
2. не демонстрирует теоретических знаний основных принципов и инструментов дисциплины;
3. выполнил менее 10 лабораторных работ (из 18);
4. не может самостоятельно работать с Terraform, Ansible, ОРА, GitOps и другими

инструментами;

5. не настроил CI/CD пайплайн или пайплайн не работает, Quality Gates отсутствуют;
6. не понимает концепции GitOps, не может объяснить отличие от классического CI/CD;
7. не выполнил задания по управлению устройствами (OTA, телеметрия, валидация) или они не работают;
8. не защитил лабораторные работы, не отвечает на контрольные вопросы;
9. отчеты отсутствуют или не соответствуют требованиям;
10. набрал менее 60 баллов по итогам семестра.