

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**
990bea3aeec848c23bd8699e48288f8d1960c600 **«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

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

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Оркестрация контейнеров и платформы Kubernetes

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	3

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Оркестрация контейнеров и платформы Kubernetes».

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 облачные ресурсы на декларативных языках. Не знаком с инструментами IaC. Не может создать базовый манифест Terraform.	С помощью преподавателя создает простейшие манифесты на YAML для описания Pod и Service. Испытывает трудности при описании облачных ресурсов через Terraform	Самостоятельно описывает ресурсы Kubernetes на YAML. Уверенно работает с Terraform для описания базовой инфраструктуры (ноды, сети). Применяет модульную организацию кода, но допускает незначительные ошибки	Свободно описывает инфраструктуру Kubernetes (ноды, сети, хранилища) с помощью Terraform (провайдеры Yandex Cloud, VK Cloud). Активно применяет модульную организацию кода для повторного использования. Оптимизирует структуру манифестов.
ИД-2 ПК-2 Разрабатывает идемпотентные сценарии управления конфигурациями (Ansible, Puppet, Chef) для настройки операционных систем, прикладного ПО и бортового ПО автономных устройств	Не знаком с концепцией идемпотентности. Не может написать простейший Ansible-плейбук. Не понимает структуру инвентори	С помощью преподавателя создает простой плейбук для установки пакетов. Понимает принцип идемпотентности, но допускает ошибки при его реализации.	Самостоятельно разрабатывает идемпотентные плейбуки для настройки ОС и прикладного ПО. Корректно использует модули Ansible. Обеспечивает повторяемость конфигураций	Разрабатывает сложные идемпотентные Ansible-плейбуки для инициализации worker-ноды Kubernetes (установка containerd, kubelet, настройка sysctl). Отлаживает сценарии управления конфигурациями при многократном применении. Применяет best practices Ansible.
ИД-3 ПК-2 Применяет системы контроля версий (Git) для хранения, версионирования и совместной работы над кодом инфраструктуры и конфигураций	Не знает базовых команд Git. Не может создать репозиторий и выполнить commit. Не знаком с моделями ветвления.	Выполняет базовые операции Git (clone, add, commit, push). С помощью подсказок создает ветку и выполняет merge. Испытывает трудности при разрешении конфликтов.	Уверенно применяет Git для хранения и версионирования манифестов Kubernetes и Helm-чартов. Использует модели ветвления (Git Flow, GitHub Flow) для командной работы. Разрешает простые конфликты слияния.	Свободно применяет Git для хранения и версионирования кода инфраструктуры. Использует продвинутое техники (rebase, cherry-pick, stash). Эффективно разрешает сложные конфликты слияния конфигурационных файлов. Организует командную работу через Code Review.

ИД-4 ПК-2 Интегрирует процессы развертывания инфраструктуры и применения конфигураций в пайплайны непрерывной интеграции и доставки (CI/CD)	Не знаком с концепциями CI/CD. Не может настроить автоматический деплой. Не понимает структуру CI/CD конфигурационных файлов	С помощью преподавателя создает простой CI/CD пайплайн (например, для запуска тестов). Затрудняется в настройке деплоя в Kubernetes.	Самостоятельно интегрирует Helm-чарты и kubectl apply в CI/CD-пайплайн (SourceCraft/GitLab CI). Настраивает автоматическое развертывание на одно окружение (dev).	Свободно интегрирует процессы развертывания в CI/CD-пайплайн. Настраивает автоматическое развертывание на множественные окружения (dev/staging/prod) с использованием переменных. Реализует Quality Gates на основе результатов smoke-тестов. Применяет best practices безопасности в пайплайне.
ИД-5 ПК-2 Обеспечивает безопасность управления инфраструктурой через код: внедряет сканирование кода на уязвимости, управляет секретами и реализует принцип минимальных привилегий	Не знаком с инструментами сканирования уязвимостей. Не понимает принципы управления секретами. Не может настроить RBAC.	С помощью преподавателя запускает Trivy для сканирования образа. Знает о необходимости управления секретами, но не может настроить Vault.	Самостоятельно настраивает сканирование манифестов Kubernetes и Dockerfile на уязвимости (Trivy, Checkov). Управляет секретами через Kubernetes Secret. Настраивает базовый RBAC.	Обеспечивает сканирование манифестов и Dockerfile на уязвимости в CI/CD-пайплайне. Управляет секретами через External Secrets Operator с HashiCorp Vault. Реализует принцип минимальных привилегий для сервисных аккаунтов. Настраивает Network Policies и Pod Security Standards.
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 концепцией GitOps. Не может установить и настроить ArgoCD. Не понимает механизмы синхронизации.	С помощью преподавателя устанавливает ArgoCD и создает простейшее Application. Затрудняется в настройке автоматической синхронизации.	Самостоятельно реализует GitOps-цикл с использованием ArgoCD. Настраивает автоматическую синхронизацию состояния Kubernetes-кластера с Git-репозиторием Helm-чартов. Понимает механизмы обнаружения дрейфа.	Свободно реализует GitOps-цикл с использованием ArgoCD. Настраивает автоматическую синхронизацию и устранение дрейфа конфигураций (конфигурационный дрейтинг). Применяет App-of-Apps паттерн. Настраивает автоматический rollback при неудачной синхронизации.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
1	b	Какой компонент Kubernetes отвечает за хранение состояния кластера? a) kube-apiserver b) etcd c) kube-scheduler d) kube-controller-manager	ПК-2
2	a, d	Какие из перечисленных инструментов относятся к классу IaC (Infrastructure as Code)? a) Terraform b) Docker c) kubectl d) Pulumi	ПК-2
3	c	Какой тип Service в Kubernetes обеспечивает доступ к приложению извне кластера через статический порт на каждой ноде? a) ClusterIP b) LoadBalancer c) NodePort d) ExternalName	ПК-2
4	b, c	Какие типы probes используются в Kubernetes для проверки готовности приложения принимать трафик и его жизнеспособности?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		a) startupProbe b) livenessProbe c) readinessProbe d) healthProbe	
5	a	Какая команда Helm используется для установки релиза? a) helm install b) helm create c) helm repo add d) helm package	ПК-2
6	b	Какой инструмент является основным для реализации GitOps-подхода в Kubernetes? a) Jenkins b) ArgoCD c) GitLab CI d) SourceCraft	ПК-2
7	a, c, d	Какие компоненты входят в Control Plane Kubernetes? a) kube-apiserver b) kubelet c) etcd d) kube-scheduler	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
8	b	Какой тип метрик Prometheus представляет собой значение, которое может увеличиваться и уменьшаться? a) Counter b) Gauge c) Histogram d) Summary	ПК-2
9	a, b, d	Какие из перечисленных ресурсов являются российскими платформами для CI/CD и хранения кода? a) SourceCraft b) GitFlic c) GitHub d) GitVerse	ПК-2
10	c	Какой объект Kubernetes используется для хранения конфиденциальных данных (паролей, токенов, ключей)? a) ConfigMap b) Volume c) Secret d) ServiceAccount	ПК-2
11	a, d	Какие стратегии обновления поддерживаются в Kubernetes Deployment?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		a) RollingUpdate b) BlueGreen c) Canary d) Recreate	
12	b	Какой инструмент используется для сканирования уязвимостей Docker-образов? a) Checkov b) Trivy c) SonarQube d) Flake8	ПК-2
13	a, c	Какие форматы файлов используются для описания манифестов Kubernetes? a) YAML b) XML c) JSON d) TOML	ПК-2
14	b	Какой легковесный дистрибутив Kubernetes предназначен для Edge-вычислений? a) Minikube b) K3s c) Kind d) MicroK8s	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
15	a, c, d	Какие типы данных относятся к observability (наблюдаемости)? a) Metrics b) Alerts c) Logs d) Traces	ПК-2
16	c	Какой компонент KubeEdge отвечает за выполнение workloads на Edge-устройстве? a) CloudCore b) EdgeHub c) EdgeCore d) DeviceController	ПК-2
17	a, b	Какие типы Volume в Kubernetes могут использоваться для временного хранения данных? a) emptyDir b) hostPath c) PersistentVolume d) ConfigMap	ПК-2
18	b	Какой протокол используется для сбора метрик в Prometheus? a) push b) pull c) stream	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		d) batch	
19	a, c	Какие аннотации необходимо добавить в Service для автоматического обнаружения Prometheus? a) <u>prometheus.io/scrape</u>: "true" b) <u>monitoring.io/enabled</u>: "true" c) <u>prometheus.io/port</u>: "8080" d) metrics.scrape: "yes"	ПК-2
20	d	Какой язык запросов используется в Grafana Loki? a) PromQL b) SQL c) GraphQL d) LogQL	ПК-2
21	a, b	Какие параметры необходимо указать в PersistentVolume? a) capacity b) accessModes c) storageClass d) reclaimPolicy (необязательно, но тоже верно; вопрос с несколькими ответами)	ПК-2
22	c	Какой Admission Controller в Kubernetes отвечает за применение Pod Security Standards?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		a) ValidatingWebhook b) MutatingWebhook c) PodSecurity d) ResourceQuota	
23	b, c, d	Какие инструменты используются для distributed tracing? a) Prometheus b) Jaeger c) Grafana Tempo d) OpenTelemetry	ПК-2
24	a	Какой параметр в HorizontalPodAutoscaler определяет минимальное количество реплик? a) minReplicas b) maxReplicas c) targetCPUUtilizationPercentage d) scaleTargetRef	ПК-2
25	a, b	Какие российские облачные платформы предоставляют Managed Kubernetes? a) Yandex Cloud b) VK Cloud c) AWS d) Google Cloud	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
26	с	Какой объект Kubernetes обеспечивает стабильный сетевой идентификатор для StatefulSet? a) Service (ClusterIP) b) Ingress c) Headless Service d) Endpoints	ПК-2
27	а, с	Какие типы Secret поддерживаются в Kubernetes? a) Opaque b) BasicAuth c) <u>kubernetes.io/tls</u> d) docker-registry	ПК-2
28	б	Какой инструмент используется для статического анализа IaC-кода Terraform? a) Trivy b) Checkov c) Semgrep d) Bandit	ПК-2
29	а, б, d	Какие способы аутентификации в HashiCorp Vault поддерживаются? a) Token b) Kubernetes c) OAuth2	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		d) LDAP	
30	c	Какой тип Service используется для внутренней балансировки нагрузки между подами внутри кластера? a) NodePort b) LoadBalancer c) ClusterIP d) ExternalName	ПК-2
31	a, d	Какие ресурсы можно использовать для управления конфигурацией приложения в Kubernetes? a) ConfigMap b) PersistentVolume c) Deployment d) Secret	ПК-2
32	b	Какая команда kubectl используется для просмотра логов контейнера? a) kubectl describe b) kubectl logs c) kubectl exec d) kubectl get	ПК-2
33	a, b, c	Какие компоненты входят в состав KubeEdge Cloud Core?	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		a) CloudHub b) EdgeController c) DeviceController d) EdgeHub	
34	d	Какой режим работы External Secrets Operator синхронизирует секреты из Vault в Kubernetes? a) Sync b) Reconcile c) Watch d) SecretStore + ExternalSecret	ПК-2
35	a, c	Какие политики восстановления (reclaimPolicy) поддерживаются в PersistentVolume? a) Retain b) Recycle c) Delete d) Preserve	ПК-2
36	b	Какой инструмент используется для развертывания локального Kubernetes кластера в Docker? a) Minikube b) Kind c) K3s d) Kubeadm	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
37	a, b, d	Какие метрики DORA используются для оценки эффективности DevOps? a) Deployment Frequency b) Lead Time for Changes c) Uptime Percentage d) Change Failure Rate	ПК-2
38	c	Какой параметр в ресурсах контейнера (resources) задает максимальное количество CPU? a) requests.cpu b) limits.memory c) limits.cpu d) requests.memory	ПК-2
39	a, c	Какие типы Ingress контроллеров поддерживаются в Kubernetes? a) NGINX Ingress Controller b) Apache Ingress c) Traefik d) HAProxy	ПК-2
40	b	Какой компонент отвечает за принятие решения о том, на какую ноду разместить под? a) kube-controller-manager b) kube-scheduler c) kubelet	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		d) kube-proxy	
41	a, b, c	Какие типы обновлений поддерживаются в ArgoCD? a) Auto-sync b) Manual sync c) Automated sync with prune d) Scheduled sync	ПК-2
42	d	Какой инструмент используется для сбора логов в Kubernetes с архитектурой, оптимизированной для Prometheus? a) Elasticsearch b) Fluentd c) Logstash d) Loki	ПК-2
43	a, b	Какие аннотации необходимо добавить в Pod для автоматического обнаружения метрик Prometheus? a) <code>prometheus.io/scrape: "true"</code> b) <code>prometheus.io/port: "8080"</code> c) <code>metrics.prometheus.io/enabled: "yes"</code> d) <code>scrape.metrics: "on"</code>	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
44	с	Какой объект Kubernetes используется для управления внешним доступом к сервисам на уровне L7? a) Service b) LoadBalancer c) Ingress d) Gateway	ПК-2
45	a, c, d	Какие режимы работы доступны для Vertical Pod Autoscaler (VPA)? a) Auto b) Scale c) Initial d) Off	ПК-2
46	b	Какой протокол используется для передачи трассировочных данных в OpenTelemetry Collector? a) HTTP b) OTLP (gRPC/HTTP) c) TCP d) UDP	ПК-2
47	a, b, c	Какие методы аутентификации поддерживаются в ArgoCD? a) Local users b) OIDC (Keycloak, Dex)	ПК-2

№	Правильный ответ	Содержание вопроса	Компетенция
3 семестр			
		c) LDAP d) SAML	
48	d	Какой инструмент является российской альтернативой Grafana для визуализации данных? a) Kibana b) Tableau c) Power BI d) Графиня (Grafanya)	ПК-2
49	a, b	Какие параметры нужно указать при создании ExternalSecret для синхронизации с Vault? a) secretStoreRef b) target c) vaultConnection d) authMethod	ПК-2
50	a, b, d	Какие стратегии развертывания могут быть реализованы с помощью Argo Rollouts? a) Blue-Green b) Canary c) Rolling Update d) A/B Testing	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 уровня сформированности компетенций производится по результатам выполнения всех видов учебной работы: посещения лекций, выполнения лабораторных работ, самостоятельной работы, сдачи зачета с оценкой. Итоговая оценка по дисциплине выставляется на основе комплексной оценки сформированности компетенции **ПК-2** и ее индикаторов.

Критерии оценивания по уровням

Уровень	Оценка	Характеристика
Высокий	Отлично (5 баллов)	Компетенция сформирована в полном объеме. Студент демонстрирует системные знания, умения и навыки, способен самостоятельно решать профессиональные задачи повышенной сложности.
Средний	Хорошо (4 балла)	Компетенция сформирована в достаточном объеме. Студент демонстрирует устойчивые знания и умения, но допускает незначительные ошибки, которые самостоятельно исправляет.
Минимальный	Удовлетворительно (3 балла)	Компетенция сформирована на базовом уровне. Студент демонстрирует знания и умения, достаточные для выполнения типовых задач, но испытывает затруднения при решении нестандартных заданий.
Минимальный уровень не достигнут	Неудовлетворительно (2 балла)	Компетенция не сформирована. Студент не владеет необходимыми знаниями, умениями и навыками, не способен выполнять профессиональные задачи.

Оценка «**отлично**» выставляется студенту, если он:

По теоретической подготовке:

1. **Полно и системно** излагает теоретический материал по архитектуре Kubernetes, принципам работы Control Plane и Worker Nodes, основным объектам (Pod, Deployment, Service, Ingress, ConfigMap, Secret, PV, PVC, StatefulSet).
2. **Демонстрирует глубокое понимание** концепций «Инфраструктура как код», «Конфигурация как код» и GitOps, способен объяснить их преимущества и области применения.
3. **Свободно ориентируется** в инструментах оркестрации (Helm, ArgoCD, K3s, KubeEdge), понимает их архитектуру и сценарии использования.
4. **Владеет современными методами** обеспечения observability (Prometheus, Grafana, Loki, Jaeger), понимает взаимосвязь метрик, логов и трасс.
5. **Знает и применяет** принципы безопасности Kubernetes: RBAC, Network Policies, Pod Security Standards, управление секретами через HashiCorp Vault.

По практической подготовке:

6. **Самостоятельно разворачивает** и настраивает кластер Kubernetes с использованием инструментов Kind, Minikube или K3s.
7. **Разрабатывает и отлаживает** YAML-манифесты и Helm-чарты любой сложности, использует шаблонизацию и best practices.
8. **Настраивает полноценный CI/CD пайплайн** в SourceCraft или GitLab CI, включающий этапы linting, тестирования, сборки, сканирования уязвимостей (Trivy) и деплоя.
9. **Реализует GitOps-цикл** с использованием ArgoCD, настраивает автоматическую синхронизацию и устранение дрейфа конфигураций.
10. **Разворачивает стек observability** (Prometheus + Grafana + Loki + Jaeger), настраивает ServiceMonitor, создает информативные дашборды и алерты.
11. **Управляет секретами** с помощью HashiCorp Vault и External Secrets Operator, реализует принцип минимальных привилегий.
12. **Выполняет все лабораторные работы** в полном объеме, строго по своему варианту, сдаёт отчеты в срок и успешно защищает их.

По результатам аттестации:

13. **Уверенно отвечает** на контрольные вопросы, демонстрирует понимание материала, а не механическое запоминание.
14. **Выполняет сквозной проект** (лабораторная работа №18) на высоком уровне, с использованием всех изученных инструментов и практик.
15. **Проявляет самостоятельность** при решении нетиповых задач, умеет находить информацию в документации и применять её на практике.

Оценка «хорошо» выставляется студенту, если он:

По теоретической подготовке:

1. **Излагает теоретический материал** в достаточном объеме, но допускает отдельные неточности, которые исправляет при помощи преподавателя.
2. **Понимает основные концепции** IaC, CaC и GitOps, но может испытывать затруднения при объяснении деталей.
3. **Ориентируется** в инструментах оркестрации, но недостаточно свободно владеет их архитектурой.
4. **Знает основные методы** observability, но может путать назначение различных инструментов.

5. **Владеет базовыми принципами** безопасности Kubernetes, но испытывает трудности при настройке сложных политик.

По практической подготовке:

6. **Развертывает кластер Kubernetes** с помощью инструментов, но может нуждаться в незначительной помощи при настройке.
7. **Разрабатывает YAML-манифесты и Helm-чарты**, но допускает незначительные ошибки, которые самостоятельно исправляет.
8. **Настраивает CI/CD пайплайн** в SourceCraft или GitLab CI, но пайплайн может содержать не все этапы или иметь незначительные недочеты.
9. **Реализует GitOps-цикл** с ArgoCD, но испытывает трудности при настройке автоматической синхронизации или устранении дрейфа.
10. **Разворачивает стек observability**, но может допускать ошибки при настройке ServiceMonitor или создании дашбордов.
11. **Выполняет большинство лабораторных работ** в полном объеме, но отдельные работы сданы с задержкой или имеют незначительные недочеты.

По результатам аттестации:

12. **Отвечает на контрольные вопросы**, но ответы могут быть недостаточно полными или развернутыми.
13. **Выполняет сквозной проект** на хорошем уровне, но проект может содержать не все заявленные компоненты или иметь незначительные недочеты.

Оценка «удовлетворительно» выставляется студенту, если он:

По теоретической подготовке:

1. **Излагает теоретический материал** фрагментарно, с существенными пробелами, но воспроизводит основные определения.
2. **Понимает базовые концепции** IaC, CaC и GitOps, но не способен объяснить их детали или применить к конкретным задачам.
3. **Имеет общее представление** об инструментах оркестрации, но не владеет их архитектурой.
4. **Знает отдельные методы** observability, но не понимает их взаимосвязи.
5. **Владеет отдельными принципами** безопасности Kubernetes, но не способен самостоятельно настроить политики.

По практической подготовке:

6. **С помощью преподавателя** развертывает кластер Kubernetes, но испытывает значительные затруднения.
7. **С помощью преподавателя** разрабатывает простейшие YAML-манифесты, но не владеет Helm или использует его на минимальном уровне.
8. **С помощью преподавателя** настраивает базовый CI/CD пайплайн, но пайплайн работает с ошибками или не включает этапы безопасности.
9. **Имеет общее представление** о GitOps и ArgoCD, но не способен самостоятельно настроить синхронизацию.
10. **С помощью преподавателя** разворачивает отдельные компоненты observability, но не может их настроить.
11. **Выполняет не менее 60% лабораторных работ**, но работы сданы с большими задержками или имеют существенные недостатки.

По результатам аттестации:

12. **Отвечает на контрольные вопросы** минимально, допускает существенные ошибки.
13. **Выполняет сквозной проект** на минимальном уровне, проект содержит только базовые компоненты.

Оценка «неудовлетворительно» (2 балла)

Оценка «**неудовлетворительно**» выставляется студенту, если он:

По теоретической подготовке:

1. **Не способен изложить** теоретический материал даже на базовом уровне, не знает основных определений.
2. **Не понимает** концепции IaC, CaC и GitOps.
3. **Не знаком** с инструментами оркестрации контейнеров.
4. **Не владеет** методами observability.
5. **Не знает** принципов безопасности Kubernetes.

По практической подготовке:

6. **Не способен** развернуть кластер Kubernetes даже с помощью преподавателя.
7. **Не может** разработать YAML-манифесты или Helm-чарты.
8. **Не имеет навыков** работы с CI/CD пайплайнами.
9. **Не знаком** с GitOps и ArgoCD.
10. **Не может** развернуть и настроить инструменты observability.
11. **Выполнил менее 50% лабораторных работ**, работы не сданы или не защищены.

По результатам аттестации:

12. **Не отвечает** на контрольные вопросы или ответы полностью неверны.
13. **Не выполнил** сквозной проект или проект не работает.

Сводная таблица критериев

Компонент оценки	Отлично (5)	Хорошо (4)	Удовлетворите льно (3)	Неудовлетворите льно (2)
Теоретичес кие знания	Глубокие, системные	Хорошие, с незначительн ыми пробелами	Базовые, фрагментарные	Отсутствуют
Практичес кие навыки	Самостоятель ная работа любой сложности	Самостоятель ная работа типовых задач	Выполнение задач с помощью	Не способен выполнять

Компонент оценки	Отлично (5)	Хорошо (4)	Удовлетворительно (3)	Неудовлетворительно (2)
Лабораторные работы	Выполнены 100% в срок	Выполнены >80%	Выполнены >60%	Выполнены <50%
Сквозной проект	Полноценное решение	Основной функционал	Минимальный функционал	Не выполнен
Ответы на вопросы	Полные, развернутые	Полные, с незначительными неточностями	Неполные, с ошибками	Отсутствуют или неверные

Шкала перевода баллов в оценку (накопленная)

Накопленный балл (из 100 возможных)	Оценка
85 – 100	Отлично
70 – 84	Хорошо
50 – 69	Удовлетворительно
0 – 49	Неудовлетворительно

Накопленный балл формируется как:

- 60% — результаты выполнения и защиты лабораторных работ (18 работ)
- 20% — результаты выполнения сквозного проекта (лабораторная работа №18)
- 10% — результаты собеседований по теоретическому материалу
- 10% — результаты тестирования (50 тестовых вопросов)