

АКТУАЛИЗИРОВАНО  
Решением Ученого совета ИЦТЭ КГЭУ  
Протокол №7 от 24.03.2026



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КГЭУ»)

УТВЕРЖДАЮ

Директор Института цифровых  
технологий и экономики

\_\_\_\_\_ Э.И. Беляев  
«30» мая 2023 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДЭ.01.01.07 DevOps

(Код и наименование дисциплины в соответствии с РУП)

Направление подготовки	<u>09.03.01 Информатика и вычислительная техника</u> (Код и наименование направления подготовки)
Направленность(и) (профиль(и))	<u>Технологии разработки программного обеспечения</u> (Наименование направленности (профиля) образовательной программы)
Квалификация	<u>Бакалавр</u> (Бакалавр / Магистр)

г. Казань, 2023

Программу разработал(и):

Наименование кафедры	Должность, уч.степень, уч.звание	ФИО разработчика
ИТИС	Доцент, к.т.н, доцент	Хамитов Р.М.

Согласование	Наименование подразделения	Дата	№ протокола	Подпись
Одобрена	кафедра ИТИС	27.04.2023	3	_____ Зав.каф.,д.п.н., доц. Торкунова Ю. В.
Согласована	кафедра ИТИС	27.04.2023	3	_____ Зав.каф.,д.п.н., доц. Торкунова Ю. В.
Согласована	Учебно-методический совет института ИЦТЭ	30.05.2023	7	_____ Директор, к.т.н., доц. Беляев Э.И.
Одобрена	Ученый совет института ИЦТЭ	30.05.2023	9	_____ Директор, к.т.н., доц. Беляев Э.И.

## 1. Цель, задачи и планируемые результаты обучения по дисциплине

*(Цель и задачи освоения дисциплины, соответствующие цели ОП)*

Целью освоения дисциплины DevOps является формирование у обучающихся компетенций в области методологии DevOps, для активного взаимодействия специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимной интеграции их рабочих процессов для обеспечения качества продукта, по использованию программных инструментов DevOps: Docker, Jenkins, Ansible, Kubernetes, Prometheus и т.п.

Задачами дисциплины являются: обучить студентов стратегиям объединения разработки программного обеспечения (Dev) и информационно-технологическое обслуживание (Ops) с целью сокращения жизненного цикла разработки систем и обеспечения непрерывной интеграции и поставки программного обеспечения.

Компетенции и индикаторы, формируемые у обучающихся:

Код и наименование компетенции	Код и наименование индикатора
ПК-1 Способен к определению требований, проектированию и разработке программного обеспечения с применением современных технологий	ПК-1.1 Проектирует, согласовывает с заинтересованными сторонами архитектуру программного обеспечения, техническую документацию и сроки выполнения поставленных задач
	ПК-1.2 Проектирует базы данных
	ПК-1.3 Разрабатывает программные интерфейсы

## 2. Место дисциплины в структуре ОП

Предшествующие дисциплины (модули), практики, НИР, др. Разработка приложений искусственного интеллекта, Архитектуры информационных систем, Цифровые системы автоматизации технологических процессов и интернет-вещей, Программное обеспечение ERP-систем, Проектный практикум

Последующие дисциплины (модули), практики, НИР, др. Производственная практика (проектная), Производственная практика (преддипломная).

### 3. Структура и содержание дисциплины

#### 3.1. Структура дисциплины

Для очной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Семестр(ы)
			8
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	3	108	108
КОНТАКТНАЯ РАБОТА*	-	40	40
АУДИТОРНАЯ РАБОТА	0,89	32	32
Лекции	0,28	10	10
Практические (семинарские) занятия	-	0	0
Лабораторные работы	0,611	22	22
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	2,11	76	76
Проработка учебного материала	2,11	76	76
Курсовой проект	-	0	0
Курсовая работа	-	0	0
Подготовка к промежуточной аттестации	-	0	0
Промежуточная аттестация:			3
			-

Для заочной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Курс/Сессия
			5/9
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	3	108	108
КОНТАКТНАЯ РАБОТА	-	31	31
АУДИТОРНАЯ РАБОТА	0,5	18	18
Лекции	0,17	6	6
Практические (семинарские) занятия	-	0	0
Лабораторные работы	0,33	12	12
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	2,5	90	90
Проработка учебного материала	2,38	86	86
Курсовой проект	-	0	0
Курсовая работа	-	0	0
Подготовка к промежуточной аттестации	0,11	4	4
Промежуточная аттестация:			3
			-

### 3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Всего часов	Распределение трудоемкости по видам учебной работы				Формы и вид контроля	Индексы индикаторов формируемых компетенций
		лекции	лаб. раб.	пр. зан.	сам. раб.		
Раздел 1 Основы методологии DevOps	36	4	8		24	ТК1	ПК-1.3, ПК-1.У, ПК-1.В
Раздел 2 Автоматизация администрирования DevOps	36	2	6		28	ТК2	ПК-1.3, ПК-1.У, ПК-1.В
Раздел 3 Инфраструктура современной разработки DevOps	36	4	8		24	ТК3	ПК-1.3, ПК-1.У, ПК-1.В
Зачет						<b>ОМ</b>	<b>ПК-1.3, ПК-1.У, ПК-1.В</b>
<b>Итого за 4 семестр</b>	<b>108</b>	<b>10</b>	<b>22</b>		<b>76</b>		
<b>ИТОГО</b>							

### 3.3. Содержание дисциплины

Раздел 1 Основы методологии DevOps.

Тема 1.1. Принципы, методы и средства реализации DevOps. Командные коммуникации и системы управления задачами. Отличие DevOps от других методик разработки программного обеспечения.

Тема 1.2. Основные этапы разработки, развертывания и обслуживания программного обеспечения. Системы контроля версий. Инфраструктура тестирования, сборки и доставки ПО.

Раздел 2 Автоматизация администрирования DevOps

Тема 2.1. Администрирование Linux

Тема 2.2. Использование Python для DevOps. Организация высокой доступности БД.

Раздел 3 Инфраструктура современной разработки DevOps

Тема 3.1. Системы виртуализации и контейнеризации (Docker). Системы непрерывной интеграции (Jenkins)

Тема 3.2. Системы управления конфигурацией. Инфраструктура как код (Ansible). Системы оркестрации (Kubernetes).

### 3.4. Тематический план практических занятий

*«Данный вид работы не предусмотрен учебным планом».*

### 3.5. Тематический план лабораторных работ

Раздел 1 Основы методологии DevOps.

Тема 1.1. Системы управления проектами (Jira) Проекты. Задачи. Дорожные карты.

Тема 1.2. Agile-разработка с помощью Jira. Scrum, Agile, Kanban доски.

Тема 1.3. Системы контроля версий Git. Сервис GitLab.

Тема 1.4. CI/CD технология непрерывной интеграции и доставки.

Раздел 2 Автоматизация администрирования DevOps

Тема 2.1. Базовое администрирование пользователей в Linux. Настройка сетевого подключения. Сетевые службы и сервисы Linux. Работа с файловой системой. Написание скриптов и управление процессами в Linux.

Тема 2.2. Введение в скрипты Python для DevOps. Сценарии Python: даты, классы и коллекции. Сценарии Python: файлы, наследование и базы данных. DevOps и автоматизация сборки с помощью Python.

Тема 2.3. Резервное копирование. Репликация. Кластерные технологии.

Раздел 3 Инфраструктура современной разработки DevOps

Тема 3.1. Системы виртуализации и контейнеризации (Docker). Системы непрерывной интеграции (Jenkins)

Тема 3.2. Системы управления конфигурацией. Инфраструктура как код (Ansible). Системы оркестрации (Kubernetes).

### 3.6. Курсовой проект /курсовая работа

*«Данный вид работы не предусмотрен учебным планом».*

## 4. Оценивание результатов обучения

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлетворительно	неудовлетворительно
			зачтено		не зачтено	
ПК-1	ПК-1.1	знать:  методологии управления инфраструктурой и разработки программных средств.	Уровень знаний в объеме, соответствует вующем программе подготовки, без ошибок	Уровень знаний в объеме, соответствует вующем программе, имеет место несколько негрубых ошибок	Минимально допустимый уровень знаний, имеет место много негрубых ошибок	Уровень знаний ниже минимальных требований, имеют место грубые ошибки

		уметь:				
		распределять роли в проектной команде; разрабатывать техническую документацию, необходимую для проектирования и разработки систем, построенных на основе облачных технологий ; применять современные информационно-коммуникационные технологии и программные средства на всех этапах жизненного цикла программных систем.	Сформированы умения	Частично сформированы умения	Сформированы базовые умения оценивать	Не сформированы базовые умения
		владеть:				
		Навыками планирования и управления жизненным циклом ИС; применения современных методологий организации разработки программного обеспечения при ведении проектов в области DevOps;	Продемонстрированы навыки	Сформированы базовые навыки	Имеется минимальный набор навыков	Не продемонстрированы базовые навыки, допущены значительные ошибки.
		знать:				
	ПК-1.2	порядок и особенности установки программного обеспечения; основы	Уровень знаний в объеме, соответствующем программ	Уровень знаний в объеме, соответствующем программ	Минимально допустимый уровень	Уровень знаний ниже минимальных требований

		современных систем управления базами данных	е подготовк и, без ошибок	е, имеет место несколько негрубых ошибок	знаний, имеет место много негрубых ошибок	ий, имеют место грубые ошибки
		уметь:				
		устанавливать программные средства на всех этапах жизненного цикла программных систем.	Сформир ован ы умения	Частично сформиро ваны умения	Сформир ован ы базовые умения оценивать	Не сформир ованы базовые умения
		владеть:				
		Навыками создания программного кода в соответствии с техническим заданием	Продемон стрирован ы навыки	Сформир ован ы базовые навыки	Имеется минималь ный набор навыков	Не продемон стрир ованы базовые навыки, допущен ы значител ьные ошибки.
	ПК-1.3	знать:				
		методологии и системы непрерывной интеграции системы виртуализации и контейнеризац ии , системы оркестрации	Уровень знаний в объеме, соответст вующем программ е подготовк и, без ошибок	Уровень знаний в объеме, соответст вующем программ е, имеет место несколько негрубых ошибок	Минимал ьно допустим ый уровень знаний, имеет место много негрубых ошибок	Уровень знаний ниже минимал ьных требован ий, имеют место грубые ошибки
		уметь:				
		управлять конфигурацией , применять знания для выполнения работ на этапе разработки программного обеспечения.	Сформир ован ы умения	Частично сформиро ваны умения	Сформир ован ы базовые умения оценивать	Не сформир ованы базовые умения
		владеть:				
		навыками проводить проверку	Продемон стрирован ы навыки	Сформир ован ы базовые	Имеется минималь ный	Не продемон стрир

		работоспособности и рефакторинг кода программного обеспечения		навыки	набор навыков	ованы базовые навыки, допущены значительные ошибки.
--	--	---	--	--------	---------------	---

Оценочные материалы для проведения текущего контроля и промежуточной аттестации приведены в Приложении к рабочей программе дисциплины.

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика.

## 5. Учебно-методическое и информационное обеспечение дисциплины

### 5.1. Учебно-методическое обеспечение

#### 5.1.1. Основная литература

1. Орлов С. А. Программная инженерия. Учебник для вузов. 5-е издание обновленное и дополненное. Стандарт третьего поколения. — (Серия «Учебник для вузов»). / С.А. Орлов. - Санкт-Петербург : Питер, 2021. - 640 с. - ISBN 978-5-4461-9590-9. - URL: <https://ibooks.ru/bookshelf/377401/reading> (дата обращения: 20.05.2023). - Текст: электронный.
2. Волк, В. К. Практическое введение в программную инженерию : учебное пособие для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 100 с. — ISBN 978-5-507-44920-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/249848>.
3. Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 196 с. — ISBN 978-5-8114-9323-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/189470>.
4. Орещенков, И. С. Инструментальные средства разработки программного обеспечения. Система Fossil / И. С. Орещенков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 284 с. — ISBN 978-5-507-44104-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/207560> (дата обращения: 22.05.2023). — Режим доступа: для авториз. пользователей.

#### 5.1.2. Дополнительная литература

1. Орлов С.А. Теория и практика языков программирования. Учебник для вузов. 2-е изд. Стандарт 3-го поколения / С.А. Орлов. - Санкт-Петербург : Питер, 2017. - 688 с. - ISBN 978-5-4461-0491-8. - URL: <https://ibooks.ru/bookshelf/355466/reading>. - Текст: электронный.
2. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. — (Серия «Бестселлеры O'Reilly»). - Санкт-Петербург : Питер, 2021. - 576 с. - ISBN 978-5-4461-0914-2. - URL: <https://ibooks.ru/bookshelf/376830/reading>. - Текст: электронный.
3. Кобылянский, В. Г. Операционные системы, среды и оболочки : учебное пособие для вузов / В. Г. Кобылянский. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 120 с. — ISBN 978-5-507-44969-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/254651>.

## 5.2. Информационное обеспечение

### 5.2.1. Электронные и интернет-ресурсы

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	Электронно-библиотечная система «Лань»	<a href="https://e.lanbook.com/">https://e.lanbook.com/</a>
2	Электронно-библиотечная система «ibooks.ru»	<a href="https://ibooks.ru/">https://ibooks.ru/</a>
3	Электронно-библиотечная система «book.ru»	<a href="https://www.book.ru/">https://www.book.ru/</a>
4	Портал «Открытое образование»	<a href="https://npoed.ru">https://npoed.ru</a>
5	Российская национальная библиотека	<a href="https://nlr.ru/">https://nlr.ru/</a>
6	КиберЛенинка	<a href="https://cyberleninka.ru">https://cyberleninka.ru</a>
7	Техническая библиотека	<a href="https://techlibrary.ru">https://techlibrary.ru</a>
8	Национальная электронная библиотека (НЭБ)	<a href="https://rusneb.ru/">https://rusneb.ru/</a>
9	Что такое DevOps? Описание DevOps   Microsoft Azure	<a href="https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-devops">https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-devops</a>
10	Agile-манифест разработки программного обеспечения	<a href="https://agilemanifesto.org/iso/ru/manifesto.html">https://agilemanifesto.org/iso/ru/manifesto.html</a>
11	Репозиторий образов Docker	<a href="https://www.docker.com/">https://www.docker.com/</a>
12	Репозиторий образов Kubernetes	<a href="https://kubernetes.io/ru/community/">https://kubernetes.io/ru/community/</a>
13	Предустановленные образы виртуальных машин	<a href="https://www.osboxes.org/">https://www.osboxes.org/</a>

### 5.2.2. Профессиональные базы данных / Информационно-справочные системы

№ п/п	Наименование профессиональных баз данных	Адрес	Режим доступа
1	Официальный интернет-портал правовой информации	<a href="http://pravo.gov.ru">http://pravo.gov.ru</a>	<a href="http://pravo.gov.ru">http://pravo.gov.ru</a>
2	Справочная правовая система «Консультант Плюс»	<a href="http://consultant.ru">http://consultant.ru</a>	<a href="http://consultant.ru">http://consultant.ru</a>
3	Справочно-правовая система по законодательству РФ	<a href="http://garant.ru">http://garant.ru</a>	<a href="http://garant.ru">http://garant.ru</a>

### 5.2.3. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Описание	Реквизиты подтверждающих документов
-------	---------------------------------------	----------	-------------------------------------

1	Операционная система Microsoft Windows 10	Пользовательская операционная система	Договор №133/2021 от 12.10.2021, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно
2	Microsoft Office 2019	Пакет офисных приложений	Договор №133/2021 от 12.10.2021, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно
3	LMS Moodle	ПО для эффективного онлайн- взаимодействия преподавателя и студента	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
4	Браузер Chrome	Система поиска информации в сети интернет	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.

## 6. Материально-техническое обеспечение дисциплины

Наименование вида учебной работы	Наименование учебной аудитории, специализированной лаборатории	Перечень необходимого оборудования и технических средств обучения
Лекции	Учебная аудитория для проведения занятий лекционного типа	Специализированная учебная мебель, технические средства обучения, служащие для представления учебной информации большой аудитории (мультимедийный проектор, компьютер (ноутбук), экран), демонстрационное оборудование, учебно-наглядные пособия
Лабораторные работы	Учебная лаборатория программной инженерии, ауд. В-608	Специализированное лабораторное оборудование по профилю лаборатории программной инженерии, специализированная учебная мебель на 50 посадочных мест, 24 компьютера с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические средства обучения (мультимедийный проектор, мультимедийная доска, моноблок), необходимое лицензионное программное обеспечение
	Компьютерный класс, ауд. В-610	Специализированная учебная мебель на 42 посадочных места, 17 компьютеров с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические

		средства обучения (мультимедийный проектор, экран для проектора, моноблок), необходимое лицензионное программное обеспечение
Учебная лаборатория информационной безопасности, ауд. В-615		Специализированное лабораторное оборудование по профилю лаборатории информационной безопасности, специализированная учебная мебель на 35 посадочных мест, 15 компьютеров с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические средства обучения (мультимедийный проектор, мультимедийная доска, моноблок), необходимое лицензионное программное обеспечение
Компьютерный класс, ауд. В-617		Специализированная учебная мебель на 24 посадочных места, 21 компьютер с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, необходимое лицензионное программное обеспечение
Компьютерный класс, ауд. В-619		Специализированная учебная мебель на 26 посадочных мест, 21 компьютер с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, необходимое лицензионное программное обеспечение
Компьютерный класс, ауд. В-621		Специализированная учебная мебель на 35 посадочных мест, 13 компьютеров с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические средства обучения (мультимедийный проектор, экран для проектора, моноблок), необходимое лицензионное программное обеспечение
Учебная лаборатория реинжиниринга и управления бизнес-процессами, ауд. В-623		Специализированное лабораторное оборудование по профилю лаборатории реинжиниринга и управления бизнес-процессами, специализированная учебная мебель на 34 посадочных места, 13 компьютеров с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические средства обучения (мультимедийный проектор, мультимедийная доска, моноблок), необходимое лицензионное программное обеспечение
Компьютерный класс, В-600		Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, компьютеров с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, технические средства обучения (мультимедийный проектор, ноутбук, экран), видеокamеры, необходимое лицензионное

		программное обеспечение
Самостоятельная работа	Компьютерный класс с возможностью выхода в Интернет и обеспечением доступа в ЭИОС В-600	Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран), видеокамеры, программное обеспечение
	Читальный зал библиотеки	Специализированная мебель, компьютерная техника с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, экран, мультимедийный проектор, программное обеспечение

## 7. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья и инвалидов

Лица с ограниченными возможностями здоровья (ОВЗ) и инвалиды имеют возможность беспрепятственно перемещаться из одного учебно-лабораторного корпуса в другой, подняться на все этажи учебно-лабораторных корпусов, заниматься в учебных и иных помещениях с учетом особенностей психофизического развития и состояния здоровья.

Для обучения лиц с ОВЗ и инвалидов, имеющих нарушения опорно-двигательного аппарата, обеспечены условия беспрепятственного доступа во все учебные помещения. Информация о специальных условиях, созданных для обучающихся с ОВЗ и инвалидов, размещена на сайте университета [www/kgeu.ru](http://www/kgeu.ru). Имеется возможность оказания технической помощи ассистентом, а также услуг сурдопереводчиков и тифлосурдопереводчиков.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушенным слухом справочного, учебного материала по дисциплине обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы оповещения о начале и конце занятия (слово «звонок» пишется на доске);
- внимание слабослышащего обучающегося привлекается педагогом жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);
- разговаривая с обучающимся, педагогический работник смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих обучающихся проводится путем:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;
- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;
- обеспечения возможности для обучающегося получить адресную консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушениями

зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой по выбранному направлению подготовки, обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;

- педагогический работник, его собеседник (при необходимости), присутствующие на занятии, представляются обучающимся, при этом каждый раз называется тот, к кому педагогический работник обращается;

- действия, жесты, перемещения педагогического работника коротко и ясно комментируются;

- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;

- обеспечивается необходимый уровень освещенности помещений;

- предоставляется возможность использовать компьютеры во время занятий и право записи объяснений на диктофон (по желанию обучающихся).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ и инвалидов определяется педагогическим работником в соответствии с учебным планом. При необходимости обучающемуся с ОВЗ, инвалиду с учетом их индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

## **8. Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися.**

Методическое обеспечение процесса воспитания обучающихся выступает одним из определяющих факторов высокого качества образования. Преподаватель вуза, демонстрируя высокий профессионализм, эрудицию, четкую гражданскую позицию, самодисциплину, творческий подход в решении профессиональных задач, в ходе образовательного процесса способствует формированию гармоничной личности.

При реализации дисциплины преподаватель может использовать следующие методы воспитательной работы:

- методы формирования сознания личности (беседа, диспут, внушение, инструктаж, контроль, объяснение, пример, самоконтроль, рассказ, совет, убеждение и др.);

- методы организации деятельности и формирования опыта поведения (задание, общественное мнение, педагогическое требование, поручение, приучение, создание воспитывающих ситуаций, тренинг, упражнение, и др.);

- методы мотивации деятельности и поведения (одобрение, поощрение социальной активности, порицание, создание ситуаций успеха, создание ситуаций для эмоционально-нравственных переживаний, соревнование и др.)

При реализации дисциплины преподаватель должен учитывать следующие направления воспитательной деятельности:

### *Гражданское и патриотическое воспитание:*

- формирование у обучающихся целостного мировоззрения, российской идентичности, уважения к своей семье, обществу, государству, принятым в семье и обществе духовно-нравственным и социокультурным ценностям, к национальному, культурному и историческому наследию, формирование стремления к его сохранению и развитию;

- формирование у обучающихся активной гражданской позиции, основанной на традиционных культурных, духовных и нравственных ценностях российского общества, для повышения способности ответственно реализовывать свои конституционные права и обязанности;

- развитие правовой и политической культуры обучающихся, расширение конструктивного участия в принятии решений, затрагивающих их права и интересы, в том числе в различных формах самоорганизации, самоуправления, общественно-значимой деятельности;

- формирование мотивов, нравственных и смысловых установок личности, позволяющих противостоять экстремизму, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам, межэтнической и межконфессиональной нетерпимости, другим негативным социальным явлениям.

### *Духовно-нравственное воспитание:*

- воспитание чувства достоинства, чести и честности, совестливости, уважения к родителям, учителям, людям старшего поколения;

- формирование принципов коллективизма и солидарности, духа милосердия и сострадания, привычки заботиться о людях, находящихся в трудной жизненной ситуации;

- формирование солидарности и чувства социальной ответственности по отношению к людям с ограниченными возможностями здоровья, преодоление психологических барьеров по отношению к людям с ограниченными возможностями;

- формирование эмоционально насыщенного и духовно возвышенного отношения к миру, способности и умения передавать другим свой эстетический опыт.

### *Культурно-просветительское воспитание:*

- формирование эстетической картины мира;

- формирование уважения к культурным ценностям родного города, края, страны;

- повышение познавательной активности обучающихся.

### *Научно-образовательное воспитание:*

- формирование у обучающихся научного мировоззрения;

- формирование умения получать знания;

- формирование навыков анализа и синтеза информации, в том числе в профессиональной области.

**Вносимые изменения и утверждения на новый учебный год**

№ п/п	№ раздела внесения изменений	Дата внесения изменений	Содержание изменений	«Согласовано» Зав. каф. реализующей дисциплину	«Согласовано» председатель УМК института (факультета), в состав которого входит выпускающая
1	2	3	4	5	6
1					
2					
3					

*Приложение к рабочей  
программе дисциплины*



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КГУ»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
по дисциплине**

**Б1.В.ДЭ.01.01.07 DevOps**

*(Наименование дисциплины в соответствии с учебным планом)*

Направление подготовки 09.03.01 Информатика и вычислительная техника  
*(Код и наименование направления подготовки)*

Квалификация Бакалавр  
*(Бакалавр / Магистр)*

г. Казань, 2023



## 2. Оценочные материалы текущего контроля и промежуточной аттестации

### Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлетворительно	неудовлетворительно
			зачтено		не зачтено	
ПК-1	ПК-1.1	знать:				
		методологии управления инфраструктурой и разработки программных средств.	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок	Уровень знаний в объеме, соответствующем программе, имеет место несколько негрубых ошибок	Минимально допустимый уровень знаний, имеет место много негрубых ошибок	Уровень знаний ниже минимальных требований, имеют место грубые ошибки
		уметь:				
		распределять роли в проектной команде; разрабатывать техническую документацию, необходимую для проектирования и разработки систем, построенных на основе облачных технологий; применять современные информационно-коммуникационные технологии и программные средства на всех этапах жизненного	Сформированы умения	Частично сформированы умения	Сформированы базовые умения оценивать	Не сформированы базовые умения

		цикла программных систем.				
		владеть:				
		Навыками планирования и управления жизненным циклом ИС; применения современных методологий организации разработки программного обеспечения при ведении проектов в области DevOps;	Продемонстрированы навыки	Сформированы базовые навыки	Имеется минимальный набор навыков	Не продемонстрированы базовые навыки, допущены значительные ошибки.
	ПК-1.2	знать:				
		порядок и особенности установки программного обеспечения; основы современных систем управления базами данных	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок	Уровень знаний в объеме, соответствующем программе, имеет место несколько негрубых ошибок	Минимально допустимый уровень знаний, имеет место много негрубых ошибок	Уровень знаний ниже минимальных требований, имеют место грубые ошибки
		уметь:				
		устанавливать программные средства на всех этапах жизненного цикла программных систем.	Сформированы умения	Частично сформированы умения	Сформированы базовые умения оценивать	Не сформированы базовые умения
		владеть:				
	Навыками создания программного кода в соответствии с техническим заданием	Продемонстрированы навыки	Сформированы базовые навыки	Имеется минимальный набор навыков	Не продемонстрированы базовые навыки, допущены значительные ошибки.	

ПК-1.3	знать:				
	методологии и системы непрерывной интеграции системы виртуализации и контейнеризации, системы оркестрации	Уровень знаний в объеме, соответствует вующем программе подготовки, без ошибок	Уровень знаний в объеме, соответствует вующем программе, имеет место несколько негрубых ошибок	Минимально допустимый уровень знаний, имеет место много негрубых ошибок	Уровень знаний ниже минимальных требований, имеют место грубые ошибки
	уметь:				
	управлять конфигурацией, применять знания для выполнения работ на этапе разработки программного обеспечения.	Сформированы умения	Частично сформированы умения	Сформированы базовые умения оценивать	Не сформированы базовые умения
владеть:					
	навыками проводить проверку работоспособности и рефакторинг кода программного обеспечения	Продемонстрированы навыки	Сформированы базовые навыки	Имеется минимальный набор навыков	Не продемонстрированы базовые навыки, допущены значительные ошибки.

Оценка **«отлично»** выставляется за выполнение *лабораторных работ в семестре; тестовых заданий; понимание принципов и методов DevOps, умение определять требования, проектировать и разрабатывать программного обеспечения с применением современных технологий DevOps;*

Оценка **«хорошо»** выставляется за выполнение *лабораторных работ в семестре; тестовых заданий; понимание принципов и методов DevOps;*

Оценка **«удовлетворительно»** выставляется за выполнение *лабораторных работ в семестре и тестовых заданий;*

Оценка **«неудовлетворительно»** выставляется за слабое и неполное выполнение *лабораторных работ в семестре и тестовых заданий.*

### 3. Перечень оценочных средств

Краткая характеристика оценочных средств, используемых при текущем контроле успеваемости и промежуточной аттестации обучающегося по дисциплине:

Наименование оценочного средства	Краткая характеристика оценочного средства	Описание оценочного средства
Отчет по лабораторной работе (ОЛР)	Выполнение лабораторной работы, обработка результатов испытаний, измерений, эксперимента. Оформление отчета, защита результатов лабораторной работы по отчету	Перечень заданий и вопросов для защиты лабораторной работы, перечень требований к отчету
Тест (Тест)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Комплект тестовых заданий

### 4. Перечень контрольных заданий или иные материалы, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций в процессе освоения дисциплины

*Пример задания*

#### Для текущего контроля ТК1:

Проверяемая компетенция: ПК-1 Способен к определению требований, проектированию и разработке программного обеспечения с применением современных технологий

ПК-1.1 Проектирует, согласовывает с заинтересованными сторонами архитектуру программного обеспечения, техническую документацию и сроки выполнения поставленных задач

#### Отчет по лабораторной работе

Данный вид контроля за учебной деятельностью осуществляется в течение семестра. Посещение лабораторных занятий и выполнения заданий является допуском к зачету (промежуточной аттестации) по дисциплине. Обучающийся не допускается к зачету (промежуточной аттестации), если не сданы отчеты (в виде разработанных тестовых наборов или программ) по всем учебным модулям, а также в случае недобора баллов согласно бально-рейтинговой системы (менее 55). Для каждого раздела предусмотрено от 2 до 3 лабораторных заданий. Каждому студенту выдается индивидуальное задание.

Текущий контроль проводится перед началом каждого лабораторного занятия. Обучающиеся предоставляют отчет по лабораторной работе в электронном виде в виртуальной образовательной среде Moodle.

#### Лабораторная работа 1 Системы управления проектами Jira.

1. Создание и настройка проекта.
2. Создать задачу.

3. Создать подзадачи.
4. Пригласить участников команды.
5. Управление задачами

Лабораторная работа 2 Agile-разработка с помощью Jira.

1. Scrum разработка с помощью Jira Software
2. Создать kanban доску с помощью Jira Software
3. Создать эпики в Jira Software
4. Спринты в Jira Software
5. Работа с версиями с помощью Jira Software

Лабораторная работа 3 Сервис GitLab

1. Создание репозитория на GitLab. Создание SSH-ключа для подключения к репозиторию. Работа с удалённым репозиторием.
  1. Создать учетную запись и рабочую группу
  2. Создать репозиторий
  3. Загрузить файлы
  4. Добавить ключи авторизации.
2. Командная работа в GitLab. Работа с ветками. Сравнение версий и отмена изменений.
  1. Добавление файлов в репозиторий git
  2. Добавление пользователей в проект
  3. Отслеживание изменений в git
  4. Фиксация изменений
  5. Создание баг-репорта.

Лабораторная работа 4 CI/CD технология непрерывной интеграции и доставки

1. Знакомство с CD (Continuous delivery).
  1. Создаём .gitlab-ci.yml
  2. Задаём подготовительные команды
  3. Указываем этапы
  4. Описываем джобы и задаём команду
2. Continuous integration с использованием GitLab CI
  1. Запуск вручную
  2. Продолжение при провале
  3. Выполнение джобов по условию
  4. Запуск по расписанию
  5. Серия джобов

Критерием оценивания результатов является правильность выполнения задания согласно заявленным требованиям

При оценке выполненной лабораторной работы учитываются следующие критерии:

Отчет составлен верно, все тестовые программы проведены, даны подробные описания определений и понятий, верно приведены примеры – 5 баллов;

Отчет выполнен в целом верно, имеются незначительные ошибки при

оформлении тестовых отчетов – 4 балла;

Отчет выполнен со значительным количеством ошибок, не соответствует заявленному стандарту, часто демонстрируются ошибки при выполнении тестирования программ - 3 балла;

Отчет составлен неверно, большое количество ошибок при оформлении – 2 балла.

Максимальное количество баллов за каждое выполненное лабораторное задание – 5 баллов.

Максимальное количество баллов за все задания раздела – 20 баллов.

### **Для текущего контроля ТК2:**

Проверяемая компетенция: Проверяемая компетенция: ПК-1  
Способен к определению требований, проектированию и разработке программного обеспечения с применением современных технологий

ПК-1.2 Проектирует базы данных

ПК-1.3 Разрабатывает программные интерфейсы

Отчет по лабораторной работе

Данный вид контроля за учебной деятельностью осуществляется в течение семестра. Посещение лабораторных занятий и выполнения заданий является допуском к зачету (промежуточной аттестации) по дисциплине. Обучающийся не допускается к зачету (промежуточной аттестации), если не сданы отчеты (в виде разработанных тестовых наборов или программ) по всем учебным модулям, а также в случае недобора баллов согласно бально-рейтинговой системы (менее 55). Для каждого раздела предусмотрено от 2 до 3 лабораторных заданий. Каждому студенту выдается индивидуальное задание.

Текущий контроль проводится перед началом каждого лабораторного занятия. Обучающиеся предоставляют отчет по лабораторной работе в электронном виде в виртуальной образовательной среде Moodle.

Темы лабораторных работ:

Базовое администрирование пользователей в Linux.

Настройка сетевого подключения.

Сетевые службы и сервисы Linux.

Работа с файловой системой.

Написание скриптов и управление процессами в Linux.

Введение в скрипты Python для DevOps.

Сценарии Python: даты, классы и коллекции.

Сценарии Python: файлы, наследование и базы данных.

DevOps и автоматизация сборки с помощью Python.

Резервное копирование. Репликация. Кластерные технологии.

Лабораторная работа Базовое администрирование пользователей в Linux.

1. Контрольное задание

Скачайте образ операционной системы (Astra\_Linux\_2.1)

Первого пользователя в системе вы создали при установке (логин: student,

пароль: Asdf1234). В этом задании Вам нужно создать несколько новых пользователей, еще несколько групп в системе и добавить пользователей в группы.

Для создания пользователей и групп можно пользоваться любыми изученными средствами.

1. Создайте две группы: **analysts** с заданным "вручную" GID **444** и **managers** с паролем **qwerty123**.

2. Далее создайте трех обычных пользователей: **samuel** с корневым каталогом в **/etc/samuel**, **lily**, сразу при создании включив его в **managers** и **harry** с заданным "вручную" UID **1110**.

3. Добавьте пользователя **harry** в группу **managers** и **analysts**, а пользователя **samuel** в группу **managers**.

4. Назначьте пароль пользователю **samuel** пароль **my\_password**.

5. Назначьте пользователю **lily** права администратора (добавить в группу **sudo**).

6. Отправьте на проверку файлы **/etc/passwd** и **/etc/group**.

7. Удалите созданных пользователей.

2. Контрольное задание

1. Скачайте образ виртуальной машины ([AstraLinuxSE](#)).

2. Разверните ее и войдите в систему (имя пользователя: **student**; пароль: **Asdf1234**)

3. Авторизуйтесь под пользователем **root**. Выполните команды.

4. `adduser admin1`

5. `usermod -G astra-admin admin1`

Авторизуйтесь под пользователем **admin1**. Добавьте трех новых пользователей с соответствующими домашними

директориями: **student7**, **student8**, **student9**. Задайте пароли для каждого из

них, используйте команду `adduser`.

6. Создайте группу **course** и добавьте в нее всех трех пользователей.

7. Для пользователя `student7` выставите ограничение: срок действия пароля 5 месяцев(150 дней) и предупреждение об окончании срока действия пароля 9 дней.

8. Заблокируйте пользователя `student8`, заблокировав его пароль. Проверьте, что блокировка подействовала.

9. Войдите в систему под пользователем `student9`, находясь в этом же терминале.

10. Войдите в систему под пользователем root. Разблокируйте пользователя `student8`. Проверьте, что блокировка снята.

11. Войдите в систему под пользователем admin Выполните команду `chage -l student7 > /home/admin1/student7`

12. Прикрепите файлы из домашней директории `admin1`: `.bash_history`, `student7`

Лабораторная работа Настройка сетевого подключения. Сетевые службы и сервисы Linux.

### 1. Контрольное задание

1. Скачайте образ виртуальной машины ([AstraLinuxSE](#)).

2. Разверните ее и войдите в систему (имя пользователя: student; пароль: Asdf1234)

3. Добавьте новый сетевой интерфейс в виртуальную машину как показано в видеолекциях, или измените существующий.

4. Затем проверьте вывод команды `ifconfig`, отредактируйте файлы конфигураций.

5. В файле `interfaces` назначьте данному интерфейсу адрес `192.168.1.5` с маской `255.255.255.0`.

6. Перезапустите службу и перезапустите интерфейс.

7. Проверьте доступность вашего сетевого интерфейса

8. Посмотрите текущую таблицу маршрутизации
9. Проверьте что получилось командой `ifconfig`:
10. Настройте виртуальный сетевой интерфейс, назначьте ему адрес `196.168.1.135`, маску `255.255.255.0`, broadcast `192.168.1.255`.
11. Перед тем, как добавить маршруты посмотрите таблицу маршрутизации.
12. Для локального интерфейса выполняем команду:
13. 

```
route add -net 127.0.0.0 netmask 255.0.0.0 lo
```
14. Снова посмотрите таблицу маршрутизации.
15. Выполните команды:
16. 

```
ifconfig > /home/task
```
17. 

```
route >> /home/task
```
18. Прикрепите файл `/home/student/.bash_history` и `/home/task`.

## 2. Контрольное задание

1. Скачайте образы виртуальных машин ([AstraLinuxSE](#), [AstraLinuxSE server](#)).
2. Для доступа в системы воспользуйтесь учетными данными (имя пользователя: student; пароль: Asdf1234)
3. Настройте сервер gw в соответствии с конфигурацией указанной выше. Номер Вашей сети обозначатся  $X = 5$ 
  - Ваша сеть: 192.168.5.0/24
  - Домен: example5.com
  - IP адрес для gw: 192.168.5.1
  - Маска сети: 255.255.255.0
  - Имя хоста для gw: gw.example5.com
4. По аналогии настройте server:
  - IP адрес для server: 192.168.5.2

- Маска сети: 255.255.255.0
  - Имя хоста для server: server.example5.com
5. На клиентской машине:
- Настроить получение адреса на интерфейсе по [dhcp](#)
  - Имя хоста для user: user.example5.com
6. При необходимости перезагрузите виртуальную машину и отключите `wicd`

7. Текущая система будет использована как gw.example5.com. Добавьте еще один сетевой интерфейс. Для этого на вашем ПК необходимо создать виртуальный коммутатор или сетевой мост для внутренней коммутации.

8. Разверните еще 2 виртуальные машины AstralinuxSE\_server и AstralinuxSE

9. На gw выполните команды:

- 10. `sudo ifconfig > gw.file`
- 11. `ping -c 3 192.168.5.2 >> gw.file`
- 12. `ping -c 3 ya.ru >> gw.file`
- 13. `hostname >> gw.file`

14. На server выполните команды:

- 15. `sudo ifconfig > server.file`
- 16. `ping -c 3 192.168.5.1 >> server.file`
- 17. `hostname >> server.file`

18. На клиентской машине user выполните:

- 19. `hostname > user.file`

20. Если нет доступа к Wincp ssh выполните:

- 21. `systemctl enable ssh.service`
- 22. `sudo init 6`

23. Прикрепите файлы gw.file, server.file и user.file

## Лабораторная работа Сценарии Python

Мы устроились на работу в компанию, где раньше уже был DevOps Engineer. Он написал скрипт, позволяющий узнать, какие файлы модифицированы в репозитории, относительно локальных изменений. Этим скриптом недоволено начальство, потому что в его выводе есть не все изменённые файлы, а также непонятен полный путь к директории, где они находятся. Как можно доработать скрипт ниже, чтобы он исполнял требования вашего руководителя?

```
```python
#!/usr/bin/env python3

import os

bash_command = ["cd ~/netology/sysadm-homeworks", "git status"]
result_os = os.popen(' && '.join(bash_command)).read()
is_change = False
for result in result_os.split('\n'):
    if result.find('modified') != -1:
        prepare_result = result.replace('\tmodified: ', '')
        print(prepare_result)
        break
```

#!/usr/bin/env python3
import os

basedir = "~/netology/sysadm-homeworks"
bash_command = [f"cd {basedir}", "git status "]
result_os = os.popen(' && '.join(bash_command)).read()
for result in result_os.split('\n'):
    if result.find('modified') != -1:
        prepare_result = result.replace('modified:', basedir)
        print(prepare_result)
% python3 test.py
~/netology/sysadm-homeworks README.md
```

## Лабораторная работа Сценарии Python

Доработать скрипт выше так, чтобы он мог проверять не только локальный репозиторий в текущей директории, а также умел воспринимать путь к репозиторию, который мы передаём как входной параметр. Мы точно знаем, что начальство коварное и будет проверять работу этого скрипта в директориях, которые не являются локальными репозиториями.

```
#!/usr/bin/env python3

import os
import sys

basedir = ""
try:
    basedir = sys.argv[1]
except:
```

```

print("Incorrect repository path")

if basedir != "":
    bash_command = [f"cd {basedir}", "git status "]
    result_os1 = os.listdir(basedir);

    if result_os1.__contains__(".git"):
        result_os = os.popen(' && '.join(bash_command)).read()
        for result in result_os.split('\n'):
            if result.find('modified') != -1:
                prepare_result = result.replace('modified:', basedir)
                print(prepare_result)
    else:
        print("There is no git repository on the entered path")
% python3 test2.py Documents/DevOps/git/devops-netology
Documents/DevOps/git/devops-netology README.md
% python3 test2.py Documents/DevOps/
There is no git repository on the entered path

```

### Лабораторная работа Логическое резервирование

1. На первом сервере создайте несколько баз данных. В них создайте различные объекты (например, таблицы, представления, индексы).
2. Сделайте копию только глобальных объектов кластера с помощью утилиты `pg_dumpall`.
3. Сделайте копии каждой базы данных кластера с помощью утилиты `pg_dump` в параллельном режиме.
4. Полностью восстановите кластер на другом сервере, используя созданные резервные копии.
5. Попробуйте подобрать такие данные и параметры команды `COPY`, чтобы созданную копию таблицы невозможно было загрузить.

### Лабораторная работа Базовая резервная копия

1. В первом кластере создайте табличное пространство и базу данных с таблицей в этом пространстве.
2. Сделайте базовую резервную копию кластера с помощью `pg_basebackup` в формате `tar` со сжатием.
3. Разверните второй кластер из этой резервной копии. Табличное пространство разместите в другом каталоге, изменив файл `tablespace_mapping`.
4. Запустите второй сервер и проверьте его работоспособность.
5. Удалите базу данных и табличное пространство в обоих кластерах.

Критерием оценивания результатов является правильность выполнения задания согласно заявленным требованиям

При оценке выполненной лабораторной работы учитываются следующие критерии:

Отчет составлен верно, все тестовые программы проведены, даны подробные описания определений и понятий, верно приведены примеры – 5 баллов;

Отчет выполнен в целом верно, имеются незначительные ошибки при оформлении тестовых отчетов – 4 балла;

Отчет выполнен со значительным количеством ошибок, не соответствует заявленному стандарту, часто демонстрируются ошибки при выполнении тестирования программ - 3 балла;

Отчет составлен неверно, большое количество ошибок при оформлении – 2 балла.

Максимальное количество баллов за каждое выполненное лабораторное задание – 5 баллов.

Максимальное количество баллов за все задания раздела – 15 баллов.

#### **Для текущего контроля ТКЗ:**

Проверяемая компетенция: Проверяемая компетенция: ПК-1  
Способен к определению требований, проектированию и разработке программного обеспечения с применением современных технологий

ПК-1.2 Проектирует базы данных

ПК-1.3 Разрабатывает программные интерфейсы

#### **Отчет по лабораторной работе**

Данный вид контроля за учебной деятельностью осуществляется в течение семестра. Посещение лабораторных занятий и выполнения заданий является допуском к зачету (промежуточной аттестации) по дисциплине. Обучающийся не допускается к зачету (промежуточной аттестации), если не сданы отчеты (в виде разработанных тестовых наборов или программ) по всем учебным модулям, а также в случае недобора баллов согласно бально-рейтинговой системы (менее 55). Для каждого раздела предусмотрено от 2 до 3 лабораторных заданий. Каждому студенту выдается индивидуальное задание.

Текущий контроль проводится перед началом каждого лабораторного занятия. Обучающиеся предоставляют отчет по лабораторной работе в электронном виде в виртуальной образовательной среде Moodle.

Лабораторная работа 1 Системы виртуализации и контейнеризации (Docker)

Дайте письменные ответы на следующие вопросы:

- В чём отличие режимов работы сервисов в Docker Swarm кластере: replication и global?

В режиме replicated приложение запускается в том количестве экземпляров, какое укажет пользователь. При этом на отдельной ноде может быть как несколько экземпляров приложения, так и не быть совсем.

В режиме global приложение запускается обязательно на каждой ноде и в единственном экземпляре.

## • Какой алгоритм выбора лидера используется в Docker Swarm кластере?

Raft.

- Протокол решает проблему согласованности - чтобы все manager-ноды имели одинаковое представление о состоянии кластера
- Для отказоустойчивой работы должно быть не менее трёх manager-нод.
- Количество нод обязательно должно быть нечётным, но лучше не более 7 (рекомендация из документации Docker).
- Среди manager-нод выбирается лидер, его задача гарантировать согласованность.
- Лидер отправляет keepalive-пакеты с заданной периодичностью в пределах 150-300мс. Если пакеты не пришли, менеджеры начинают выборы нового лидера.
- Если кластер разбит, нечётное количество нод должно гарантировать, что кластер останется консистентным, т.к. факт изменения состояния считается совершенным, если его отразило большинство нод. Если разбить кластер пополам, нечётное число гарантирует, что в какой-то части кластера будет большинство нод.

## • Что такое Overlay Network?

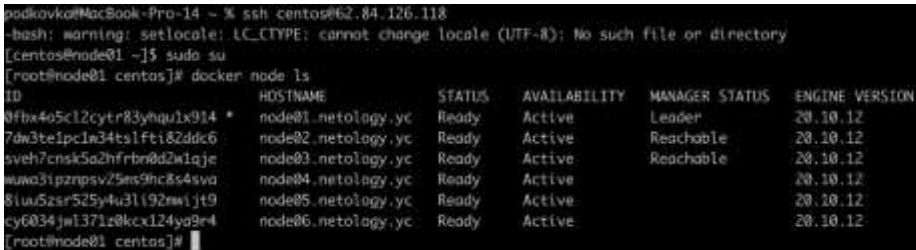
L2 VPN сеть для связи демонов Docker между собой. В основе используется технология vxlan

## Лабораторная работа 2

### Создать ваш первый Docker Swarm кластер в Яндекс.Облаке

Для получения зачета, вам необходимо предоставить скриншот из терминала (консоли), с выводом команды:

```
docker node ls
```



```
podkovka@MacBook-Pro-14 ~ % ssh centos@62.84.126.118
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[centos@node01 ~]$ sudo su
[root@node01 centos]# docker node ls
ID                HOSTNAME          STATUS    AVAILABILITY  MANAGER STATUS  ENGINE VERSION
8fbx465c12cytr83ynqu1x914 * node01.netology.yc Ready       Active         Leader           20.10.12
7dw3teipc1m34talfti82ddc6 node02.netology.yc Ready       Active         Reachable        20.10.12
sveh7cnsk5a2hfrbr8d2w1aje node03.netology.yc Ready       Active         Reachable        20.10.12
wawa31pzrpsv25as9nc8s4sya node04.netology.yc Ready       Active         Reachable        20.10.12
81uu5zsr525yfu31192mwi1t9 node05.netology.yc Ready       Active         Reachable        20.10.12
cy6034jwi1371z0kcx124ya9e4 node06.netology.yc Ready       Active         Reachable        20.10.12
[root@node01 centos]#
```

## Лабораторная работа 3

Создать ваш первый, готовый к боевой эксплуатации кластер мониторинга, состоящий из стека микросервисов.

Для получения зачета, вам необходимо предоставить скриншот из терминала (консоли), с выводом команды:

```
docker service ls
```

```
[root@node03 centos]# docker service ls
ID                NAME                                MODE                REPLICAS  IMAGE                                PORTS
3z55r7vcy>86     swarm_monitoring_alertmanager     replicated          1/2        stefanprodan/swarmrise-alertmanager:v0.14.0
3jpeM3b3joc      swarm_monitoring_caddy             replicated          1/2        stefanprodan/caddy:latest
1zccwqgqpw       swarm_monitoring_caddyhttp         global             6/6        google/caddyhttp:latest
135y9d8taca      swarm_monitoring_dockers-exporter  global             6/6        stefanprodan/caddy:latest
4y3vfy9tkaa7     swarm_monitoring_grafana           replicated          1/2        stefanprodan/swarmrise-grafana:5.3.4
3ygnf1i3tipg     swarm_monitoring_node-exporter     global             6/6        stefanprodan/swarmrise-node-exporter:v0.10.0
3zcvfy9tkaa7     swarm_monitoring_prometheus        replicated          1/2        stefanprodan/swarmrise-prometheus:v2.5.0
41w1j1j043       swarm_monitoring_unkee              replicated          1/2        cloudflare/unkee:v0.8.0
[root@node03 centos]#
```

## Лабораторная работа 4

Выполнить на лидере Docker Swarm кластера команду (указанную ниже) и дать письменное описание её функционала, что она делает и зачем она нужна: # см. документацию: [https://docs.docker.com/engine/swarm/swarm\\_manager\\_locking/](https://docs.docker.com/engine/swarm/swarm_manager_locking/)  
`docker swarm update --autolock=true`

```
[root@node03 centos]# service docker restart
Redirecting to /bin/systemctl restart docker.service
[root@node03 centos]# docker node ls
Error response from daemon: Swarm is encrypted and needs to be unlocked before it can be used. Please use "docker swarm unlock" to unlock it.
[root@node03 centos]# docker swarm unlock
Please enter unlock key:
[root@node03 centos]# docker node ls
ID                HOSTNAME                STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
1zs279vooalxonzk1vsum8ia  node01.netology.yc     Ready  Active         Leader           20.10.12
nd26qddkqkfx3lgrv4f0qshje  node02.netology.yc     Ready  Active         Reachable        20.10.12
1zhu0y9e0xz44natkyfbc1o9 *  node03.netology.yc     Ready  Active         Reachable        20.10.12
1arbfw14mj130j1hns8r8bqa  node04.netology.yc     Ready  Active         Reachable        20.10.12
1ahdxa7dfhj4bjsybfyn7hzu  node05.netology.yc     Ready  Active         Reachable        20.10.12
jag5gq8yfa6h38roc81qa272l  node06.netology.yc     Ready  Active         Reachable        20.10.12
[root@node03 centos]#
```

`--autolock=true` обязывает вводить ключ разблокировки на ноде, чтобы она могла заново присоединиться к кластеру, если была перезапущена. Ввод ключа позволит расшифровать лог Raft и загрузить все "секреты" в память ноды (логины, пароли, TLS ключи, SSH ключи и т.д.)  
 Для защиты кластера от несанкционированного доступа к файлам ноды. Например, получив жесткий диск сервера или образ диска VM с нодой, не получить доступ к кластеру и нодам без ключа.

## Лабораторная работа 1 Системы непрерывной интеграции (Jenkins)

### 1. Установить jenkins по любой из инструкций

version: '3.8'

services:

jenkins:

image: jenkins/jenkins:latest-jdk11

privileged: true

user: root

ports:

- 8080:8080

- 50000:50000

container\_name: jenkins

volumes:

- \$HOME/jenkins\_compose/jenkins\_configuration:/var/jenkins\_home

- /var/run/docker.sock:/var/run/docker.sock

### 2. Запустить и проверить работоспособность

\$ docker-compose up -d

\$ docker-compose ps

| NAME    | COMMAND                  | SERVICE | STATUS  |
|---------|--------------------------|---------|---------|
| jenkins | "/sbin/tini -- /usr/..." | jenkins | running |

0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::8080->8080/tcp, :::50000->50000/tcp

\$ docker exec jenkins cat /var/jenkins\_home/secrets/initialAdminPassword

3. Сделать первоначальную настройку
4. Настроить под свои нужды
5. Поднять отдельный cloud

Критерием оценивания результатов является правильность выполнения задания согласно заявленным требованиям

При оценке выполненной лабораторной работы учитываются следующие критерии:

Отчет составлен верно, все тестовые программы проведены, даны подробные описания определений и понятий, верно приведены примеры – 5 баллов;

Отчет выполнен в целом верно, имеются незначительные ошибки при оформлении тестовых отчетов – 4 балла;

Отчет выполнен со значительным количеством ошибок, не соответствует заявленному стандарту, часто демонстрируются ошибки при выполнении тестирования программ - 3 балла;

Отчет составлен неверно, большое количество ошибок при оформлении – 2 балла.

Максимальное количество баллов за каждое выполненное лабораторное задание – 5 баллов.

Максимальное количество баллов за все задания раздела – 20 баллов.

## Для промежуточной аттестации:

### Тест

| Вопрос  | Варианты ответа      |
|---|----------------------|
| Какой тип систем контроля версий сейчас используется чаще всего?  | Системные            |
|   | Централизованные     |
|   | Локальные            |
|   | Распределенные       |
| Какая команда git используется для клонирования удаленного репозитория на локальный компьютер?  | git download         |
|   | git checkout         |
|   | git clone            |
|   | git copy             |
| Какая команда git используется для получения изменений с удаленного репозитория на локальный компьютер?   | git push             |
|   | git pull             |
|   | git sync             |
|   | git clone            |
| Какая команда позволяет просмотреть справочную информацию или руководства?  | man                  |
|   | which                |
|   | what                 |
|   | mal                  |
| Какой командой следует запустить скрипт script1.sh, находящийся в каталоге /etc/dir1/, если работа пользователя в данный момент происходит в /home/Liza/dir2? | ./script1.sh         |
|   | /etc/dir1/script1.sh |
|   | .././../script1.sh   |
|   | run script1.sh       |
| 1 Какой командой можно повысить привилегии своего пользователя?   | chown                |
|   | chmod                |
|   | sudo                 |
|   | useradd              |
| 3. Какой командой посмотреть запущенные процессы в системе Linux?   | ls                   |
|   | ps                   |
|   | pwd                  |
|   | find                 |
| Какой командой скопировать файл?  | file                 |
|   | rm                   |
|   | cp                   |
|   | locate               |
| Как собрать образ Docker?   | docker run           |
|   | docker build         |
|   | docker ps            |
|   | docker commit        |
| Как называется инструкция сборки образа Docker?   | Docker Image         |
|   | Docker Registry      |
|   | Dockerfile           |
|   | Docker Daemon        |
| Какой командой можно отправить образ Docker в Registry?   | docker push          |
|   | docker pull          |
|   | docker run           |

|   |                         |
|---|-------------------------|
|   | <i>docker exec</i>      |
| <i>Какой командой можно<br/>посмотреть все контейнеры<br/>Docker?</i> | <i>docker ps -a</i>     |
|   | <i>docker ps</i>        |
|   | <i>docker images</i>    |
|   | <i>docker images -a</i> |