

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

---

УТВЕРЖДЕНО  
решением Ученого совета ННГУ  
протокол № 15 от 24.12.2025 г.

**Рабочая программа дисциплины**

Системы виртуализации и контейнеризации

---

Уровень высшего образования  
Бакалавриат

---

Направление подготовки / специальность  
02.03.02 - Фундаментальная информатика и информационные технологии

---

Направленность образовательной программы  
Сопряженная разработка программного и аппаратного обеспечения

---

Форма обучения  
очная

---

г. Нижний Новгород

2026 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.12.03 Системы виртуализации и контейнеризации относится к части, формируемой участниками образовательных отношений образовательной программы.

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-4: Способен проектировать программное обеспечение	<p>ПК-4.1: Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</p> <p>ПК-4.2: Знает методы и средства проектирования программного обеспечения</p> <p>ПК-4.3: Знает методы и средства проектирования баз данных</p> <p>ПК-4.4: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения</p> <p>ПК-4.5: Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных</p>	<p>ПК-4.1: Имеет представление о виртуализации и уровнях ее реализации</p> <p>ПК-4.2: Знает технологии аппаратной виртуализации</p> <p>ПК-4.3: Знает технологии аппаратной виртуализации и на уровне операционной системы (контейнеры)</p> <p>ПК-4.4: Умеет разворачивать виртуальные вычислительные системы различных уровней.</p> <p>ПК-4.5: Умеет конфигурировать и администрировать виртуальные вычислительные системы различных уровней</p>	Тест Практическое задание	Зачёт: Контрольные вопросы Практическое задание

## 3. Структура и содержание дисциплины

### 3.1 Трудоемкость дисциплины

	очная
--	-------

<b>Общая трудоемкость, з.е.</b>	<b>2</b>
<b>Часов по учебному плану</b>	<b>72</b>
в том числе	
<b>аудиторные занятия (контактная работа):</b>	
- занятия лекционного типа	<b>12</b>
- занятия семинарского типа (практические занятия / лабораторные работы)	<b>12</b>
- КСР	<b>1</b>
<b>самостоятельная работа</b>	<b>47</b>
<b>Промежуточная аттестация</b>	<b>0</b> <b>Зачёт</b>

### 3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
Ф	Ф	Ф	Ф	Ф	
Тема 1. Основы виртуализации	12	2	2	4	8
Тема 2. Управление виртуальными машинами	12	2	2	4	8
Тема 3. Контейнеризация и Docker	12	2	2	4	8
Тема 4. Оркестрация контейнеров	12	2	2	4	8
Тема 5. Безопасность виртуализированных сред	12	2	2	4	8
Тема 6. Автоматизация и Infrastructure as Code	11	2	2	4	7
Аттестация	0				
КСР	1			1	
<b>Итого</b>	<b>72</b>	<b>12</b>	<b>12</b>	<b>25</b>	<b>47</b>

### Содержание разделов и тем дисциплины

#### 1. Основы виртуализации

##### Тема 1.1. Введение в виртуализацию

- История и эволюция виртуализации
- Типы виртуализации: полная, паравиртуализация, аппаратная
- Преимущества и недостатки виртуализации
- Сценарии применения в современной ИТ-инфраструктуре

##### Тема 1.2. Гипервизоры Type 1 (Bare-metal)

- Архитектура гипервизоров первого типа
- VMware ESXi: возможности и особенности
- Xen и его применение
- Hyper-V от Microsoft
- Практика: установка и базовая настройка гипервизора

#### Тема 1.3. Гипервизоры Type 2 (Hosted)

- Архитектура гипервизоров второго типа
- VirtualBox: установка и конфигурация
- VMware Workstation/Fusion
- Сравнение производительности Type 1 vs Type 2
- Практика: создание и настройка виртуальной машины (2 ак. ч.)

#### Тема 1.4. QEMU/KVM

- Архитектура KVM в Linux kernel
- QEMU как эмулятор и виртуализатор
- libvirt для управления виртуализацией
- Практика: развертывание VM через KVM и virt-manager (2 ак. ч.)

### 2. Управление виртуальными машинами

#### Тема 2.1. Жизненный цикл виртуальных машин

- Создание VM: выбор параметров, ресурсов
- Снапшоты и клонирование
- Миграция VM (холодная и живая миграция)
- Операции с VM в различных гипервизорах

#### Тема 2.2. Виртуальные сети и хранилища

- Типы виртуальных сетей: NAT, Bridge, Host-only
- VLAN и сегментация сети
- Виртуальные диски: форматы (VMDK, VHD, QCOW2)
- Storage: NFS, iSCSI, SAN для виртуальных сред
- Настройка сетевой инфраструктуры и подключение хранилищ

#### Тема 2.3. Оптимизация и производительность VM

- Распределение ресурсов: CPU, RAM, I/O
- Оверкоммит ресурсов
- Балансировка нагрузки
- Мониторинг и оптимизация работы VM

### 3. Контейнеризация и Docker

#### Тема 3.1. Введение в контейнеризацию

- Отличия контейнеров от виртуальных машин
- История контейнеризации: chroot, LXC, Docker
- Архитектура Docker: daemon, client, registry
- Практика: установка Docker, первые контейнеры (2 ак.ч.)

#### Тема 3.2. Работа с Docker-образами

- Структура Docker-образов и слои
- Dockerfile: инструкции и best practices
- Docker Hub и приватные registry
- Мультистейдж сборки для оптимизации размера
- Создание собственных образов для различных приложений

#### Тема 3.3. Docker Compose и многоконтейнерные приложения

- Описание сервисов в docker-compose.yml
- Сети и volumes в Docker Compose
- Управление зависимостями между контейнерами
- Развертывание многоуровневого приложения (frontend, backend, database)

#### Тема 3.4. Изоляция ресурсов в Linux

- Namespaces: PID, NET, MNT, UTS, IPC, USER
- Cgroups: ограничение CPU, памяти, I/O
- Capabilities и дропнинг привилегий
- Исследование изоляции контейнеров, настройка ограничений ресурсов

### 4. Оркестрация контейнеров

#### Тема 4.1. Введение в Kubernetes

- Архитектура Kubernetes: master и worker nodes
- Основные объекты: Pod, Service, Deployment, ConfigMap, Secret
- kubectl: основные команды
- Установка Minikube/Kind, развертывание первого приложения

#### Тема 4.2. Управление приложениями в Kubernetes

- Deployment strategies: rolling update, blue-green, canary
- StatefulSet для stateful-приложений
- DaemonSet и Job
- Развертывание и обновление приложений с различными стратегиями (3 ак.ч.)

#### Тема 4.3. Сетевые возможности Kubernetes

- CNI плагины (Calico, Flannel)
- Ingress controllers для маршрутизации трафика
- Network policies для безопасности
- Настройка внешнего доступа к сервисам

#### Тема 4.4. Helm - пакетный менеджер для Kubernetes

- Концепция Helm charts
- Создание и кастомизация charts
- Helm repositories
- Установка популярных приложений через Helm, создание собственного chart

### 5. Безопасность виртуализированных сред

#### Тема 5.1. Безопасность контейнеров

- Сканирование образов на уязвимости (Trivy, Clair)
- Подписывание образов
- Runtime security (Falco)

- Анализ и устранение уязвимостей в образах

#### Тема 5.2. SELinux и AppArmor

- Mandatory Access Control (MAC)
- Политики SELinux для контейнеров
- Профили AppArmor
- Настройка политик безопасности

#### Тема 5.3. RBAC и управление доступом

- Role-Based Access Control в Kubernetes
- ServiceAccounts и токены
- Pod Security Standards
- Настройка ролей и прав доступа

### 6. Автоматизация и Infrastructure as Code

#### Тема 6.1. Terraform для управления инфраструктурой

- Декларативное описание инфраструктуры
- Providers и resources
- State management
- Развертывание виртуальной инфраструктуры с помощью Terraform

#### Тема 6.2. Ansible для конфигурирования

- Playbooks и роли
- Inventory и переменные
- Интеграция с Docker и Kubernetes
- Автоматизация настройки серверов и развертывания приложений

#### Тема 6.3. Мониторинг виртуализированных сред

- Prometheus и Grafana для Kubernetes
- Логирование: ELK/EFK stack
- Метрики контейнеров и VM
- Настройка базового мониторинга

## **4. Учебно-методическое обеспечение самостоятельной работы обучающихся**

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Самостоятельная работа студентов при изучении дисциплины проводится с целью углубления знаний по дисциплине и предусматривает:

- повторение пройденного учебного материала, чтение рекомендованной литературы;
- подготовку к практическим занятиям;
- выполнение общих и индивидуальных домашних заданий;
- работу с электронными источниками;
- подготовку к сдаче формы промежуточной аттестации.

Цель самостоятельной работы - формирование навыков непрерывного самообразования и профессионального совершенствования.

Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.

Основные виды самостоятельной работы студентов:

- работа с основной и дополнительной литературой;
- изучение категориального аппарата дисциплины;
- самостоятельное изучение тем дисциплины;
- подготовка к зачёту;
- работа в библиотеке;
- изучение сайтов по темам дисциплины в сети Интернет.

Для повышения эффективности самостоятельной работы студентов преподавателю целесообразно использовать следующие виды деятельности:

- консультации,
- выдача заданий на самостоятельную работу,
- информационное обеспечение обучения,
- контроль качества самостоятельной работы студентов.

## **5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)**

### **5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:**

#### **5.1.1 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-4:**

**1.** Какой тип гипервизора работает непосредственно на аппаратном обеспечении без промежуточной ОС?

- A) Type 2 (hosted)
- B) Type 1 (bare-metal)
- C) Программная виртуализация
- D) Паравиртуализация

**2.** Какое расширение процессоров Intel отвечает за аппаратную виртуализацию ввода-вывода (прямое назначение устройств)?

- A) VT-x
- B) EPT
- C) VT-d
- D) VMX

**3.** Что из перечисленного является механизмом ядра Linux, ограничивающим ресурсы (CPU, память) для группы процессов?

- A) Namespaces
- B) Cgroups

- C) OverlayFS
- D) Seccomp

**4.** Какой гипервизор изначально разрабатывался с упором на паравиртуализацию и используется в AWS (до перехода на Nitro)?

- A) KVM
- B) Hyper-V
- C) Xen
- D) ESXi

**5.** Какой формат/стандарт описывает контейнерный образ и рантайм и поддерживается Docker, containerd, podman?

- A) OpenStack
- B) OCI (Open Container Initiative)
- C) CNI
- D) CRI

**6.** В Kubernetes минимальная единица развертывания, которая может содержать один или несколько контейнеров, – это:

- A) Deployment
- B) Pod
- C) Service
- D) ReplicaSet

**7.** Какая файловая система позволяет организовать слоистую структуру образов Docker (слои только для чтения + верхний записываемый)?

- A) ext4
- B) XFS
- C) OverlayFS
- D) ZFS

**8.** Что такое «гипервызов» (hypercall) в контексте паравиртуализации?

- A) Обращение к гипервизору из гостевой ОС
- B) Аппаратное прерывание таймера
- C) Команда процессора для включения виртуализации
- D) Системный вызов в ядре хоста

**9.** Какая технология позволяет запускать контейнеры внутри изолированной легковесной виртуальной машины для усиления безопасности?

- A) Kata Containers
- B) Docker Swarm
- C) KubeVirt
- D) Open vSwitch

**10.** Какой объект Kubernetes обеспечивает стабильный IP-адрес и балансировку трафика между подами?

- A) Ingress
- B) ConfigMap
- C) Service
- D) PersistentVolume

**11.** Какая команда Docker позволяет запустить интерактивную оболочку bash внутри работающего контейнера?

- A) `docker run -it bash`
- B) `docker exec -it <container> bash`
- C) `docker attach bash`
- D) `docker start -i bash`

**12.** Что такое «бинарная трансляция» в программной виртуализации?

- A) Преобразование двоичного кода гостевой ОС в код хоста в реальном времени
- B) Сжатие образа виртуальной машины
- C) Перевод сетевых пакетов между виртуальными машинами
- D) Шифрование дискового трафика

**13.** Какой компонент Kubernetes отвечает за фактический запуск подов на узлах кластера?

- A) kube-apiserver
- B) etcd
- C) kube-scheduler
- D) kubelet

**14.** Для чего используется `docker commit`?

- A) Сохраняет изменения в работающем контейнере как новый образ
- B) Фиксирует версию Compose-файла
- C) Отправляет образ в реестр
- D) Останавливает и удаляет контейнер

**15.** Какая платформа управления одновременно поддерживает виртуальные машины (KVM) и контейнеры (LXC) из коробки?

- A) OpenShift
- B) Proxmox VE
- C) vSphere
- D) Rancher

**16.** При использовании аппаратной виртуализации (VT-x/AMD-V) гостевая ОС может напрямую выполнять привилегированные инструкции без вмешательства гипервизора. (Верно / Неверно)

**17.** В Docker все контейнеры на одном хосте имеют полностью изолированные ядра ОС. (Верно / Неверно)

**18.** Kubernetes Ingress работает только на 4-м уровне модели OSI (TCP/UDP). (Верно / Неверно)

**19.** Паравиртуализация требует модификации гостевой ОС. (Верно / Неверно)

**20.** Live-миграция виртуальной машины возможна без остановки сервисов, но требует общей сети хранения данных (SAN) или распределенного хранилища. (Верно / Неверно)

### Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
превосходно	96-100% правильных ответов
отлично	81-95% правильных ответов
очень хорошо	76-80% правильных ответов
хорошо	61-75% правильных ответов
удовлетворительно	46-60% правильных ответов
неудовлетворительно	31-45% правильных ответов
плохо	30% и меньше правильных ответов

### 5.1.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-4:

#### Задание 1. Виртуальная машина через CLI (libvirt/Virsh)

Создайте и запустите VM с Ubuntu Server 22.04, используя `virt-install` (или через `virsh`).

- Объем ОЗУ: 1 ГБ, диск 10 ГБ (`qcow2`), сеть `bridge`.
- Установите внутри VM пакет `nginx`.
- Проверьте доступность веб-сервера с хостовой системы (используйте IP-адрес VM).
- Выполните `snapshot` виртуальной машины с именем `snap1`, а затем откатите состояние к `snapshot`.  
(Результат: предоставить команды и подтверждение работы.)

#### Задание 2. Базовые контейнеры Docker

1. Напишите `Dockerfile`, который собирает образ на основе `python:3.9-slim`, копирует скрипт `app.py` (выводящий "Hello, Virtualization!") и запускает его.

2. Соберите образ с тегом `hello-python:v1`.
3. Запустите контейнер из этого образа и убедитесь, что строка выводится.
4. Остановите контейнер, удалите его. Запустите повторно с ограничением CPU (0.5 ядра) и памяти (256 МБ). Проверьте ограничения через `docker stats`.

### Задание 3. Работа с томами и сетью

- Создайте контейнер `nginx:alpine`, смонтировав хост-каталог `./html` в `/usr/share/nginx/html`.
- Поместите в `./html` файл `index.html` с произвольным содержимым.
- Пробросьте порт 8080 хоста на порт 80 контейнера.
- Проверьте отображение страницы через `curl http://localhost:8080`.

### Задание 4 (сравнительное)

На одном хосте запустите 3 виртуальные машины (через `libvirt`) с Alpine Linux (минимальная ОЗУ 256 МБ) и 10 контейнеров `busybox:latest` с бесконечным циклом `while true; do sleep 1; done`. Сравните потребление памяти и CPU (утилитами `top`, `vmstat`, `docker stats`). Напишите короткое заключение (2-3 предложения) о плотности размещения.

### Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
превосходно	Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного и дополнительного материала
отлично	Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного материала
очень хорошо	Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание материала, допущено не более 2 неточностей не принципиального характера
хорошо	Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, допущены неточности не принципиального характера, но обучающийся показывает систему знаний по теме своими ответами на поставленные вопросы
удовлетворительно	Задание выполнено не в полном объеме (решено более 50% поставленных задач), но обучающийся допускает ошибки, нарушена последовательность ответа, но в целом раскрывает содержание основного материала
неудовлетворительно	Задание выполнено не в полном объеме (решено менее 50% поставленных задач), обучающийся дает неверную информацию при ответе на поставленные задачи, допускает грубые ошибки при толковании материала, демонстрирует

Оценка	Критерии оценивания
	незнание основных терминов и понятий.
плохо	Задание не выполнено, обучающийся демонстрирует полное незнание материала

## 5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

### Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с	Продемонстрированы базовые навыки при решении стандартных задач с некоторым	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и	Продемонстрированы навыки при решении нестандартных задач без	Продемонстрирован творческий подход к решению нестандартных задач

	обучающегося от ответа		некоторым и недочетами	и недочетами	недочетов	ошибок и недочетов	
--	------------------------	--	------------------------	--------------	-----------	--------------------	--

### Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

### 5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

#### 5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Дайте определение виртуализации. Перечислите основные цели, которые достигаются с ее помощью.
2. Назовите и кратко охарактеризуйте уровни реализации виртуализации (от уровня инструкций до уровня приложений).
3. В чем разница между программной и аппаратной виртуализацией? Приведите примеры технологий для каждого подхода.
4. Какие расширения процессоров поддерживают аппаратную виртуализацию (Intel и AMD)? Для чего нужны VT-x/AMD-V и EPT/NPT?
5. Опишите метод бинарной трансляции в программной виртуализации. В каких случаях он применяется?
6. Что такое гипервизор? Чем отличаются гипервизоры типа 1 (bare-metal) от типа 2 (hosted)? Приведите примеры.

7. Назовите основные гипервизоры, используемые в современных ЦОД. Укажите их ключевые особенности (VMware ESXi, KVM, Hyper-V, Xen).
8. Дайте определение паравиртуализации. Чем она отличается от полной виртуализации? Приведите пример интерфейса (API), используемого в паравиртуализации.
9. В чем преимущества и недостатки паравиртуализации по сравнению с полной виртуализацией на базе аппаратной поддержки?
10. Что такое серверная виртуализация? Перечислите основные задачи, которые она решает (консолидация, live-миграция, отказоустойчивость).
11. Какие платформы (системы управления) виртуализацией вы знаете? Назовите проприетарные и открытые решения (vSphere, OpenStack, Proxmox VE).
12. Сравните виртуализацию на уровне ОС (контейнеры) и аппаратную виртуализацию (VM) по критериям изоляции, производительности и плотности размещения.
13. Объясните, как работает прямая передача устройств (PCI passthrough, VT-d). В каких сценариях она необходима?
14. Что такое «чувствительные инструкции» в контексте виртуализации x86? Как эта проблема решалась до появления аппаратной поддержки?
15. Опишите архитектуру KVM. Почему KVM считается гипервизором типа 1, хотя работает внутри ядра Linux?
16. Что такое контейнеризация? В чем принципиальное отличие контейнера от виртуальной машины?
17. Назовите и опишите механизмы ядра Linux, обеспечивающие работу контейнеров (namespaces и cgroups). Приведите примеры пространств имен (pid, net, mnt).
18. Что такое UnionFS (OverlayFS)? Как она используется при формировании образов контейнеров?
19. Какие существуют форматы и стандарты для контейнерных образов (OCI)? Чем образ отличается от контейнера?
20. Перечислите основные компоненты архитектуры Docker (клиент, демон, containerd, runc). Какова роль каждого?
21. Для чего нужен Docker Compose? Приведите пример сценария использования.
22. Какие проблемы решает оркестрация контейнеров? Почему одного Docker недостаточно для работы в production-окружении?
23. Опишите базовые объекты Kubernetes: Pod, Deployment, Service, Ingress. Назначение каждого.
24. Как в Kubernetes реализовано масштабирование приложений (Horizontal Pod Autoscaler) и обновления без простоя (rolling update)?
25. Сравните системы управления виртуализацией и системы оркестрации контейнеров. Приведите примеры унифицированных платформ (OpenShift, Rancher, KubeVirt).
26. Какие существуют способы повышения безопасности контейнеров (drop capabilities, seccomp, SELinux/AppArmor)?
27. Что такое gVisor и Kata Containers? В чем их отличие от классических контейнеров (runc)?
28. Как организовано сетевое взаимодействие между контейнерами на одном хосте (bridge) и между разными хостами (overlay)?
29. Опишите типовой процесс CI/CD с использованием Docker и Kubernetes (сборка образа, публикация в реестре, развертывание).
30. Назовите основные реестры контейнерных образов (публичные и приватные). Чем Docker Hub отличается от Harbor?

## **Критерии оценивания (оценочное средство - Контрольные вопросы)**

Оценка	Критерии оценивания
зачтено	вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно»
не зачтено	хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно»

### 5.3.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-4

#### Задание 1. Развертывание локального кластера Kubernetes

Используя `minikube` или `kind`, создайте кластер с одной нодой.

- Установите подающую сеть (Calico или Flannel).
- Разверните Deployment из образа `nginx:alpine` с 3 репликами.
- Создайте Service типа ClusterIP для доступа к этим репликам.
- Выполните `port-forward` и убедитесь, что `nginx` отвечает.

#### Задание 2. Обновление приложения без простоя (rolling update)

- Создайте Deployment с образом `your_app:v1` (например, простой веб-сервер на Flask с версией в ответе).
- Выполните обновление до `v2` с изменением содержимого ответа.
- В процессе обновления запустите непрерывную отправку запросов (например, `while true; do curl ...; sleep 0.2; done`) и запишите, были ли ошибки.
- Продемонстрируйте стратегию RollingUpdate (параметры `maxSurge`, `maxUnavailable`).

#### Задание 3. Использование ConfigMap и Secret

- Создайте ConfigMap с параметрами приложения (например, `APP_COLOR=blue`, `LOG_LEVEL=info`).
- Создайте Secret с паролем БД (закодируйте `mysqlpass`).
- Настройте Deployment так, чтобы переменные окружения из ConfigMap и Secret были переданы в контейнер.
- Проверьте внутри контейнера (через `env`), что переменные доступны.

#### Задание 4. Гибридная задача: KubeVirt

*(Если есть возможность установить KubeVirt в minikube/kind)*

- Установите KubeVirt на кластер.
- Определите VirtualMachine (YAML) на базе образа `cirros` (минимальная Linux VM).
- Запустите эту VM внутри Kubernetes.

- Подключитесь к VM через `virtctl console` и выполните команду `uname -a`.
- Удалите VM и отключите KubeVirt.

## Задание 5. Сетевая политика (NetworkPolicy)

- В кластере K8s создайте два пространства имен: `frontend` и `backend`.
- Разверните в каждом pod с метками (`app: front`, `app: back`).
- Создайте NetworkPolicy, которая разрешает доступ к `backend` только из подов `frontend` (на порт 80).
- Проверьте доступность с помощью `kubectl exec` и `wget`.

### Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	вся компетенция (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно»
не зачтено	хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно»

## 6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Русакова М. С. Моделирование физических процессов и визуализация данных средствами Python: практикум / Русакова М. С., Рогачева Е. В. - Самара : Самарский университет, 2024. - 76 с. - Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С. П. Королева» в качестве практикума для обучающихся по основной образовательной программе высшего образования по направлению подготовки 03.03.02 Физика. - Книга из коллекции Самарский университет - Информатика. - ISBN 978-5-7883-2099-1. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=973902&idb=0>.
2. Инновационные подходы к визуализации и разработке с применением унифицированного языка моделирования (UML) : учебное пособие для студентов направлений 09.03.02 "информационные системы и технологии", 38.03.05 "бизнес-информатика" / Шлаев Д. В., Сорокин А. А., Аникуев С. В., Орел Ю. В. - Ставрополь : СтГАУ, 2024. - 72 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции СтГАУ - Информатика. - СЭБ. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1005211&idb=0>.
3. Кабаков Р.И. R в действии. Анализ и визуализация данных на языке R : монография / Кабаков Р.И. - Москва : ДМК-пресс, 2023. - 590 с. - ISBN 978-5-89818-347-9. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=878908&idb=0>.

## Дополнительная литература:

1. Введение в технологии контейнеризации : учебное пособие, направления подготовки: 09.03.02, 09.04.01, 2.2.15, 2.3.8 / Галицкий М. В., Докучаев В. А., Маклачкова В. В., Неронов Ф. А., Плотников П. С. - Москва : МТУСИ, 2024. - 112 с. - Книга из коллекции МТУСИ - Информатика. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=928714&idb=0>.
2. Никулин Е. А. Компьютерная графика. Оптическая визуализация : учебное пособие для вузов / Никулин Е. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2023. - 200 с. - Рекомендовано Ученым советом Нижегородского государственного технического университета им. Р. Е. Алексеева в качестве учебного пособия для студентов направления подготовки «Информатика и вычислительная техника». - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-47029-7. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=863288&idb=0>.
3. Жорняк А. Г. Обработка больших массивов данных и визуализация результатов научных исследований с использованием библиотек NumPy и Matplotlib языка программирования Python : учебное пособие / Жорняк А. Г., Морозова Т. А. - Москва : МАИ, 2025. - 92 с. - Библиогр.: доступна в карточке книги, на сайте ЭБС Лань. - Книга из коллекции МАИ - Информатика. - СЭБ. - ISBN 978-5-00246-342-8. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=1003073&idb=0>.

## Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционная система Microsoft Windows
2. Пакет прикладных программ Microsoft Office
3. Правовая система «Консультант плюс»
4. Правовая система «Гарант».
5. Интернет браузеры (Mozilla Firefox, Google Chrome)
6. VirtualBox,
7. Docker,
8. Kubernetes

## **7. Материально-техническое обеспечение дисциплины (модуля)**

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Мееров Иосиф Борисович, кандидат технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 17.12.2025, протокол № протокол №6.