

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
(ФГАОУ ВО «ЮФУ»)
Инженерно-технологическая Академия
Институт компьютерных технологий и информационной безопасности
Кафедра Систем Автоматизированного Проектирования
им. В. М. Курейчика

РЕФЕРАТ №2

**на тему: «Виртуализация, виртуальные машины, виртуальные
контейнеры с примерами»**

по дисциплине «Операционные системы»

Выполнил
студент КТбo2-4

А. А. Воронов

Принял
профессор каф. САПР
им. В. М. Курейчика, к. т. н.

Е. В. Нужнов

Таганрог 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОСНОВЫ ВИРТУАЛИЗАЦИИ	4
1.1 Определение виртуализации и её типы	4
1.1.1 Определение виртуализации	4
1.1.2 Типы виртуализации	4
1.2 История развития виртуализации.....	5
1.3 Основные преимущества и недостатки виртуализации	5
1.3.1 Преимущества.....	6
1.3.2 Недостатки	6
1.4 Сферы применения виртуализации	6
2 ВИРТУАЛЬНЫЕ МАШИНЫ.....	8
2.1 Что такое виртуальная машина (ВМ)	8
2.2 Компоненты виртуальной машины.....	8
2.3 Типы виртуальных машин: системные и процессные	10
2.4 Примеры виртуальных машин	11
2.5 Преимущества и ограничения использования ВМ.....	11
2.5.1 Преимущества.....	12
2.5.2 Ограничения	13
2.6 Реальные примеры использования ВМ в компаниях и проектах	13
3 ВИРТУАЛЬНЫЕ КОНТЕЙНЕРЫ	15
3.1 Понятие виртуальных контейнеров.....	15
3.2 Отличия контейнеров от виртуальных машин	15
3.3 Популярные платформы для контейнеризации.....	16
3.3.1 Docker	16
3.3.2 Kubernetes.....	16
3.3.3 Podman.....	17
3.4 Преимущества контейнеризации: скорость, экономия ресурсов.....	18

3.5 Примеры использования контейнеров в разработке и эксплуатации ПО.....	18
4 СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВИРТУАЛЬНЫХ МАШИН И КОНТЕЙНЕРОВ.....	20
4.1 Сравнение архитектур и технологий	20
4.1.1 Структура	20
4.1.2 Использование ресурсов.....	20
4.1.3 Изоляция.....	20
4.1.4 Скорость и гибкость.....	20
4.2 Перспективы развития обеих технологий	21
4.2.1 Перспективы виртуальных машин	21
4.2.2 Перспективы контейнеров.....	21
ЗАКЛЮЧЕНИЕ	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24

ВВЕДЕНИЕ

Виртуализация является одной из ключевых технологий современной информационной среды, играя важную роль в повышении эффективности использования вычислительных ресурсов. Она предоставляет возможность создавать виртуальные экземпляры операционных систем, приложений и других компонентов, значительно упрощая управление и масштабирование ИТ-инфраструктуры.

Особое значение имеют виртуальные машины и виртуальные контейнеры, которые позволяют организациям адаптироваться к растущим требованиям бизнеса и ускорять внедрение инноваций. Виртуальные машины обеспечивают изоляцию и независимость работы разных операционных систем на одном физическом сервере, в то время как контейнеры предоставляют лёгкие и гибкие решения для разработки и развёртывания приложений.

Целью данной работы является исследование основ виртуализации, её применения в современных информационных системах, а также изучение различий между виртуальными машинами и контейнерами. В рамках реферата будет проведён анализ архитектуры этих технологий, их преимуществ и недостатков, а также рассмотрены практические примеры использования.

Задачи работы включают:

- Изучение теоретических основ виртуализации.
- Рассмотрение архитектур виртуальных машин и контейнеров.
- Проведение сравнительного анализа указанных технологий.
- Приведение примеров из практики.

Таким образом, реферат направлен на формирование целостного представления о роли и перспективах развития виртуализации в ИТ-сфере.

1 ОСНОВЫ ВИРТУАЛИЗАЦИИ

1.1 Определение виртуализации и её типы

1.1.1 Определение виртуализации

Виртуализация — это технология, которая позволяет создавать изолированные программные среды на одном физическом устройстве. Она обеспечивает абстракцию вычислительных ресурсов и предоставляет возможность одновременного выполнения нескольких операционных систем или приложений на одном аппаратном устройстве [1].

1.1.2 Типы виртуализации

1. Программная. Гипервизор здесь будет устанавливаться на хост-систему. Он отвечает за формирование виртуальной среды, в которую устанавливается гостевая операционная система. Гипервизор будет играть роль своеобразного посредника между дополнительной и основной операционными системами.

2. Контейнерная. В качестве основной операционной системы изолироваться будет не другая ОС, а отдельно взятые процессы. Они будут выполняться (инкапсулироваться) в своем собственном окружении — с выделенной сетью, диском, а также файловой системой и так далее. С ее помощью не получится запустить полноценную операционную систему вроде Windows или Linux. Она используется для активации отдельных сервисов и программ, входящих в состав того или иного программного обеспечения.

3. Аппаратная виртуализация. Здесь гипервизор при исполнении команд гостевой операционной системы будет обращаться непосредственно к аппаратной части. Основная операционная система будет им игнорироваться. Гипервизор в соответствующем случае играть роль своеобразной мини-ОС, единственной функцией которой является запуск поверх себя других операционных систем, а также всевозможных процессов. Чтобы такой гипервизор нормально работал, у процессора и материнской платы оборудования должна быть поддержка аппаратной виртуализации [2].

1.2 История развития виртуализации.

Концепция виртуализации зародилась в 1960-х годах с развитием мейнфреймов IBM. В 1968 году IBM совместно с учёными из Кембриджа разработала систему CP/CMS, ставшую первой операционной системой с поддержкой виртуализации. В основе CP/CMS лежал гипервизор, запускавшийся на оборудовании и создававший несколько виртуальных машин, что позволяло эффективно использовать ресурсы и обеспечивало изоляцию процессов [3].

В 1970-х годах виртуализация памяти стала средством расширения размеров оперативной памяти на машинах второго поколения, что было обусловлено высокой стоимостью памяти на ферритовых сердечниках. Это позволило расширить возможности систем без значительных затрат на физическое оборудование.

В 1998 году была основана компания VMware, которая в 1999 году выпустила продукт VMware Workstation — одно из первых средств виртуализации для платформы x86. Изначально программа работала только на Windows, но позже была добавлена поддержка других операционных систем. Это событие ознаменовало новый этап в развитии виртуализации, сделав её доступной для широкого круга пользователей и организаций.

В последующие годы технологии виртуализации продолжали развиваться, становясь неотъемлемой частью современных ИТ-инфраструктур. Появление аппаратной поддержки виртуализации в процессорах, развитие контейнеризации и облачных технологий способствовали широкому распространению и внедрению виртуализации в различных сферах деятельности [4].

1.3 Основные преимущества и недостатки виртуализации

Виртуализация предоставляет ряд преимуществ, но также имеет определенные недостатки. Рассмотрим их в контексте виртуализации серверов, так как это является распространенным типом виртуализации.

1.3.1 Преимущества

Виртуализация позволяет использовать физический сервер с максимальной продуктивностью. Разберем преимущества технологии, скрывающиеся за характеристикой эффективности:

- Экономичность. Избавляет компанию от необходимости покупки дополнительных серверов, предназначенных для решения отдельных задач.
- Гибкость. При увеличении нагрузки на ВМ можно масштабировать ресурсы, чтобы избежать их исчерпания.
- Надежная изоляция. Сбои и заражения одной ВМ не отражаются на работоспособности других, расположенных на том же сервере.
- Управление версиями. Снимки виртуальной машины позволяют отменить неудачные изменения и быстро вернуться к первоначальному состоянию системы.
- Контроль над ресурсами. Виртуализация позволяет распределять вычислительные мощности между ВМ [5].

1.3.2 Недостатки

Использование технологии может быть связано со следующими минусами:

- Сбой оборудования приводит к выводу из строя всех VPS.
- Управление виртуальной средой требует специальных знаний.
- Некоторые решения могут быть связаны с расходами на лицензирование.
- При отдельных видах виртуализации без постоянного контроля возникает борьба за ресурсы [5].

1.4 Сферы применения виртуализации

Виртуализация находит широкое применение в различных областях ИТ-инфраструктуры. Эти применения способствуют повышению эффективности, гибкости и устойчивости ИТ-инфраструктуры в современных организациях:

- Виртуализация серверов. Позволяет запускать несколько виртуальных серверов на одном физическом. Это увеличивает эффективность использования оборудования.
- Виртуализация рабочих столов. Она обеспечивает доступ пользователей к своим рабочим средам с любого устройства, что упрощает управление и повышает безопасность.
- Виртуализация приложений. Такая виртуализация дает возможность запускать приложения в изолированных средах, предотвращая конфликты и облегчая их развёртывание.
- Виртуализация сетей. Создает программно-определяемые сети, чем упрощает управление и повышает гибкость сетевой инфраструктуры.
- Виртуализация хранения данных. Она объединяет различные устройства хранения в единую систему. Таким образом она облегчает управление данными и делает их доступнее [6].

2 ВИРТУАЛЬНЫЕ МАШИНЫ

2.1 Что такое виртуальная машина (ВМ)

Виртуальная машина (ВМ) — это программная или аппаратная система, которая эмулирует работу компьютера, предоставляя возможность запускать операционные системы и приложения в изолированной среде. ВМ создает иллюзию работы на отдельном физическом устройстве, используя ресурсы хоста — реального компьютера или сервера. Этот подход позволяет значительно повысить гибкость ИТ-инфраструктуры, снизить затраты и упростить администрирование.

2.2 Компоненты виртуальной машины

Архитектура виртуальной машины включает несколько ключевых компонентов, обеспечивающих её работу. Эти элементы работают вместе, создавая функциональную виртуальную среду для выполнения приложений и операционных систем.

1. Гипервизор (Hypervisor). Это специализированное программное обеспечение, которое позволяет создавать и управлять виртуальными машинами на физическом оборудовании. Гипервизор осуществляет разделение ресурсов между виртуальными машинами и физическими компонентами, таким образом обеспечивая виртуализацию вычислительных ресурсов. В своей работе гипервизор управляет процессами, оперативной памятью, хранилищем данных и сетевыми ресурсами, предоставляя каждому виртуальному компьютеру изолированную среду для работы. Гипервизор может быть как первым уровнем, который работает непосредственно на физическом оборудовании, так и вторичным, который запускается поверх операционной системы хоста.

2. Виртуальные машины (Virtual Machines). Это эмулированные или виртуализированные компьютеры, которые создаются гипервизором для работы в

виртуальной среде. Каждая виртуальная машина работает как независимый компьютер, полностью изолированный от других виртуальных машин и от физического хоста. Она имеет свою собственную операционную систему (гостевую ОС), приложения и ресурсы, которые гипервизор выделяет из общего пула ресурсов хоста. Виртуальные машины могут выполнять различные задачи, включая запуск разных операционных систем на одном физическом сервере, что повышает эффективность использования ресурсов.

3. Менеджер виртуальных машин (Virtual Machine Manager — VMM). Это программное обеспечение или интерфейс, который предоставляет пользователям средства для управления виртуальными машинами. VMM может быть представлен как графический интерфейс (GUI) или командная строка (CLI), через которые можно создавать, запускать, останавливать, удалять и мониторить виртуальные машины. Менеджер виртуальных машин также предоставляет возможность настройки параметров виртуальных машин, таких как количество выделенной памяти, процессоров и места для хранения данных, а также управление сетевыми интерфейсами виртуальных машин.

4. Ресурсы хоста (Host Resources). Это физические ресурсы машины, на которой работает система виртуализации. К ним относятся процессоры (CPU), оперативная память (RAM), устройства хранения данных (жесткие диски, SSD), сетевые ресурсы и другие аппаратные компоненты. В процессе виртуализации эти ресурсы распределяются между виртуальными машинами. Эффективное управление ресурсами хоста является критически важным для обеспечения хорошей производительности и стабильности работы виртуальных машин, а также для предотвращения перегрузки системы.

5. Гостевые операционные системы (Guest Operating Systems). Это операционные системы, которые работают на виртуальных машинах. Гостевые ОС могут быть различными: например, на одной виртуальной машине может работать Windows, на другой — Linux, а на третьей — macOS. Важно, что эти операционные системы работают независимо друг от друга несмотря на то, что все они за-

пускаются на одном физическом хосте. Гостевые ОС получают ресурсы от гипервизора и могут быть настроены с учетом специфических требований, таких как использование определенного оборудования или приложений.

6. Интерфейсы управления и API. Современные гипервизоры обычно предоставляют API и интерфейсы для управления виртуальными машинами. Эти интерфейсы позволяют пользователям и разработчикам автоматизировать процессы создания, настройки и мониторинга виртуальных машин. С помощью API можно интегрировать управление виртуальными машинами с другими системами, а также разработать собственные приложения для администрирования и оптимизации работы виртуализированных сред. API и интерфейсы управления могут быть использованы для масштабируемости виртуальных сред, а также для выполнения сложных операций, таких как миграция виртуальных машин между хостами или автоматическое восстановление после сбоев [7].

2.3 Типы виртуальных машин: системные и процессные

Виртуальные машины (VM) делятся на два основных типа: системные и процессные.

- Системные виртуальные машины. Эмулируют полный набор аппаратных ресурсов компьютера и позволяют запускать полноценные операционные системы. Они создают изолированную среду, в которой можно выполнять различные ОС на одном физическом устройстве.

- Процессные виртуальные машины. Предназначены для выполнения отдельных процессов или приложений в изолированной среде. Они предоставляют виртуальную платформу для исполнения программ, обеспечивая независимость от базовой операционной системы и аппаратного обеспечения. Примером является виртуальная машина Java (JVM), которая позволяет запускать Java-приложения на различных платформах [8].

2.4 Примеры виртуальных машин

Рассмотрим три популярных решения для создания и управления виртуальными машинами:

- Oracle VM VirtualBox. Бесплатное программное обеспечение для виртуализации от компании Oracle. VirtualBox поддерживает широкий спектр гостевых операционных систем, включая различные версии Windows, Linux, macOS и другие. Программа предоставляет удобный интерфейс для создания и настройки ВМ, а также поддерживает функции, такие как общие папки, интеграция с хостовой системой и снимки состояния ВМ.

- VMware Workstation. Коммерческий продукт от компании VMware, предназначенный для профессиональной виртуализации на настольных системах. VMware Workstation обеспечивает высокую производительность и стабильность работы ВМ, поддерживает широкий спектр операционных систем и предоставляет расширенные возможности для разработчиков и ИТ-специалистов. Среди функций — поддержка 3D-графики, клонирование ВМ и интеграция с другими продуктами VMware.

- Microsoft Hyper-V. Гипервизор от компании Microsoft, встроенный в некоторые версии Windows, начиная с Windows 8 и Windows Server 2012. Hyper-V позволяет создавать и управлять виртуальными машинами на основе Windows и Linux, обеспечивая высокую производительность и глубокую интеграцию с операционной системой Windows. Hyper-V широко используется в корпоративных средах для серверной виртуализации и построения облачных инфраструктур.

2.5 Преимущества и ограничения использования ВМ

Далее рассмотрим некоторые преимущества и ограничения использования виртуальных машин (ВМ).

2.5.1 Преимущества

- ВМ универсальны. Их можно установить почти на любой компьютер или сервер. Это позволяет экономить средства на покупку дополнительного оборудования и время на его установку. Возможности ВМ не зависят от того, на какой хост-платформе они работают.
- ВМ обеспечивают высокую степень изоляции между различными операционными системами и приложениями. Это позволяет избежать взаимных влияний и конфликтов между ними.
- Несколько ВМ могут работать на одном физическом сервере, деля его ресурсы (процессорное время, память, дисковое пространство и др.). Это позволяет оптимально использовать вычислительные мощности и уменьшить затраты на аппаратное оборудование.
- ВМ можно легко создавать, клонировать и удалять. Это облегчает масштабирование инфраструктуры и развертывание новых сред без значительных затрат времени.
- ВМ позволяют создавать изолированные среды для тестирования и разработки, где можно проверять программное обеспечение без риска повреждения рабочей инфраструктуры.
- Виртуальные машины можно сохранять в виде образов или снимков состояния. Это позволяет быстро восстанавливать системы в случае сбоев или ошибок.
- ВМ можно архивировать и переносить с одного физического устройства на другое. Как правило, виртуальная машина представлена в виде единого файла, который можно сохранить на любой доступный носитель информации. Это упрощает балансировку нагрузки, обслуживание и обновление.
- На виртуальные машины можно устанавливать почти любые операционные системы, в том числе устаревшие. Это пригодится в том случае, когда для работы приходится использовать архаичное программное обеспечение. Вместо

того, чтобы содержать устаревшие компьютеры и серверы и вкладываться в дорогостоящую техническую поддержку снятого с производства оборудования, достаточно разместить такое ПО в виртуальной машине.

2.5.2 Ограничения

- Производительность. Несмотря на значительный прогресс технологий виртуализации, производительность ВМ всё ещё может быть ниже по сравнению с физическими системами. Это связано с накладными расходами гипервизора.
- Сложность управления. Администрирование большого количества ВМ требует специальных навыков и инструментов. Например, компании с десятками или сотнями ВМ должны внедрять сложные системы управления для обеспечения мониторинга и резервного копирования.
- Зависимость от хостовой системы. Сбой в работе физического хоста может привести к выходу из строя всех ВМ, размещённых на нём. Это требует высокой надёжности серверов и дополнительных мер для обеспечения отказоустойчивости.

2.6 Реальные примеры использования ВМ в компаниях и проектах

Виртуальные машины нашли применение в самых разных сферах бизнеса и технологий, предоставляя мощные инструменты для повышения гибкости и эффективности.

- Тестирование программного обеспечения. Компании-разработчики создают виртуальные среды для тестирования своих продуктов. Например, Microsoft активно использует ВМ для тестирования своих обновлений Windows на различных конфигурациях железа и версий ОС. Это позволяет минимизировать ошибки перед выпуском.
- Образовательные учреждения. Примером внедрения является Уральская инженерная школа, которая использовала проект виртуальных рабочих мест

(VDI) для автоматизации учебного процесса. Это решение позволило централизованно обновлять программное обеспечение на всех виртуальных рабочих столах студентов.

- Облачные сервисы. Провайдеры облачных услуг, такие как Amazon Web Services (AWS) и Google Cloud Platform (GCP), активно используют виртуализацию для предоставления клиентам мощностей в формате виртуальных серверов. Это даёт клиентам возможность масштабировать инфраструктуру в зависимости от текущих нужд.

- Разработка кроссплатформенных приложений. Разработчики, такие как Google, применяют ВМ для эмуляции устройств с различными ОС. Например, для разработки Android-приложений инженеры Google используют виртуальные машины, чтобы тестировать приложения на различных версиях Android.

- Стартапы и малый бизнес. Для оптимизации бюджета многие стартапы используют ВМ вместо физических серверов. Например, небольшие компании разрабатывают и тестируют свои продукты на локальных ВМ перед выпуском на рынок, экономя на аренде серверного оборудования.

3 ВИРТУАЛЬНЫЕ КОНТЕЙНЕРЫ

3.1 Понятие виртуальных контейнеров

Виртуальные контейнеры представляют собой легковесные, изолированные среды, предназначенные для запуска приложений и их зависимостей. В отличие от виртуальных машин, контейнеры используют общее ядро операционной системы хоста, что позволяет им быть более эффективными с точки зрения использования ресурсов. Каждый контейнер содержит всё необходимое для работы приложения: код, библиотеки, зависимости и конфигурационные файлы. Это обеспечивает переносимость приложений при их запуске в различных средах.

3.2 Отличия контейнеров от виртуальных машин

Основные различия между контейнерами и виртуальными машинами заключаются в архитектуре, использовании ресурсов и уровне изоляции:

- Архитектура. Виртуальные машины эмулируют полный набор аппаратных ресурсов и включают собственную операционную систему поверх гипервизора. Контейнеры, напротив, работают на общем ядре операционной системы хоста, что делает их более легковесными.

- Использование ресурсов. Контейнеры занимают меньше места и потребляют меньше ресурсов по сравнению с виртуальными машинами, поскольку не требуют отдельной операционной системы для каждого экземпляра. Это позволяет запускать большее количество контейнеров на одном хосте по сравнению с виртуальными машинами.

- Уровень изоляции. Виртуальные машины обеспечивают более высокий уровень изоляции, так как каждая ВМ имеет собственное ядро операционной системы. Контейнеры, используя общее ядро, предлагают изоляцию на уровне пользовательского пространства, что может быть менее безопасным в случае уязвимостей ядра.

Скорость запуска. Контейнеры запускаются быстрее, так как не требуют загрузки полноценной операционной системы, в отличие от виртуальных машин, которым необходимо время для инициализации ОС.

3.3 Популярные платформы для контейнеризации

Контейнеризация стала основой современной разработки и развертывания программного обеспечения благодаря своей эффективности и гибкости. Рассмотрим три наиболее популярных платформы: Docker, Kubernetes и Podman.

3.3.1 Docker

Docker является одной из самых популярных и широко используемых платформ для контейнеризации.

Ключевые особенности:

- Простота использования: Docker предоставляет удобные инструменты для создания, управления и развертывания контейнеров.
- Эффективное использование ресурсов: Контейнеры Docker легковесны и не требуют запуска отдельной операционной системы.
- Портативность: Контейнеры, созданные с помощью Docker, могут запускаться на любом устройстве с установленным Docker Engine.
- Экосистема Docker Hub: Платформа предоставляет обширный репозиторий готовых контейнерных образов для широкого спектра приложений.

Пример использования Docker — автоматизация развертывания микросервисов в облачных средах. Он используется компаниями, такими как PayPal и Spotify, для управления сложной инфраструктурой.

3.3.2 Kubernetes

Kubernetes, или K8s, — это система оркестрации контейнеров, которая управляет их развертыванием, масштабированием и работой.

Ключевые особенности:

- Автоматизация. Kubernetes автоматизирует управление контейнерами, включая балансировку нагрузки и восстановление после сбоев.
- Масштабируемость. Система легко адаптируется к росту нагрузки и позволяет масштабировать приложения как в рамках одного сервера, так и в облаке.
- Управление инфраструктурой. Kubernetes поддерживает развертывание контейнеров в гибридных и многокластерных средах.

Компании, такие как Google, активно используют Kubernetes для управления своей облачной инфраструктурой. Он стал стандартом де-факто для масштабируемых приложений.

3.3.3 Podman

Podman — это современная альтернатива Docker, предназначенная для управления контейнерами без необходимости использования привилегированного демона (демон, или daemon, — это процесс или служба, которая выполняется на хосте и управляет контейнерами).

Ключевые особенности:

- Безопасность. Подход «без демона» снижает потенциальные уязвимости системы.
- Совместимость. Podman поддерживает Docker-образы, что облегчает переход для пользователей Docker.
- Гибкость. Он позволяет запускать контейнеры как непривилегированные пользователи, что делает его идеальным для сред с повышенными требованиями безопасности.

Podman активно используется в организациях, где безопасность является приоритетом, таких как банковский сектор и правительственные учреждения.

3.4 Преимущества контейнеризации: скорость, экономия ресурсов

Контейнеризация предоставляет значительные преимущества в разработке и эксплуатации программного обеспечения, особенно в аспектах скорости и экономии ресурсов [10].

- Скорость запуска. Контейнеры обладают легковесной структурой, что обеспечивает почти мгновенный запуск приложений. В отличие от виртуальных машин, которые требуют времени для загрузки операционной системы, контейнеры используют общее ядро ОС хоста, позволяя приложениям стартовать практически моментально.

- Экономия ресурсов. Благодаря совместному использованию ядра операционной системы, контейнеры потребляют меньше ресурсов по сравнению с виртуальными машинами. Это позволяет запускать большее количество приложений на одном сервере, повышая эффективность использования вычислительных мощностей и снижая затраты на инфраструктуру.

- Переносимость. Контейнеры обеспечивают стабильную работу приложений на любой платформе или в облаке, что помогает не привязываться к одной физической машине.

- Изоляция. Контейнеры обеспечивают изоляцию приложений и их зависимостей, что упрощает управление и повышает безопасность.

Эти преимущества делают контейнеризацию привлекательным решением для современных ИТ-инфраструктур, способствуя ускорению разработки, тестирования и развертывания приложений.

3.5 Примеры использования контейнеров в разработке и эксплуатации ПО

Контейнеры широко применяются в различных аспектах разработки и эксплуатации программного обеспечения, обеспечивая гибкость и эффективность процессов.

- Разработка и тестирование. Контейнеры позволяют разработчикам создавать однородные вычислительные среды, что упрощает развертывание и управление приложениями.
- Микросервисная архитектура. Контейнеризация идеально подходит для микросервисного подхода, где каждое приложение разделено на небольшие, независимые сервисы. Это облегчает обновление, масштабирование и обслуживание отдельных компонентов без влияния на всю систему.
- Облачные вычисления. Многие облачные провайдеры предлагают услуги контейнеризации, позволяя компаниям легко переносить свои приложения между различными облачными платформами и локальными серверами, обеспечивая гибкость и отказоустойчивость.
- Непрерывная интеграция и доставка (CI/CD). Контейнеры упрощают автоматизацию процессов сборки, тестирования и развертывания приложений, способствуя более быстрому выпуску новых версий и улучшению качества ПО.

4 СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВИРТУАЛЬНЫХ МАШИН И КОНТЕЙНЕРОВ

4.1 Сравнение архитектур и технологий

Сравнение архитектур виртуальных машин (ВМ) и контейнеров позволяет выявить ключевые отличия, которые влияют на их использование в различных сценариях.

4.1.1 Структура

- Виртуальные машины включают гипервизор, изолированную операционную систему и приложения. Такая архитектура обеспечивает высокий уровень изоляции, но требует больше ресурсов.
- Контейнеры используют общее ядро операционной системы хоста, что делает их легковесными и ускоряет запуск.

4.1.2 Использование ресурсов

- ВМ требуют значительных ресурсов для запуска отдельной ОС, что увеличивает нагрузку на сервер.
- Контейнеры экономят ресурсы за счет совместного использования ОС, что позволяет запускать большее количество приложений на одной инфраструктуре.

4.1.3 Изоляция

- ВМ обеспечивают полную изоляцию, включая ядро и приложения, что повышает безопасность.
- Контейнеры предоставляют изоляцию на уровне процессов, что может быть менее безопасным, если есть уязвимости в ядре ОС.

4.1.4 Скорость и гибкость

- Виртуальные машины требуют времени для загрузки ОС, что замедляет запуск.

– Контейнеры запускаются мгновенно, что ускоряет разработку, тестирование и развертывание.

4.2 Перспективы развития обеих технологий

Технологии виртуализации и контейнеризации продолжают активно развиваться, находя новые применения и совершенствуясь в ответ на растущие потребности бизнеса. Перспективы развития обеих технологий связаны с их интеграцией, повышением эффективности и внедрением новых подходов к управлению вычислительными ресурсами.

4.2.1 Перспективы виртуальных машин

1. Интеграция с облачными технологиями. Виртуальные машины остаются основой для частных и гибридных облаков. Развитие технологий, таких как VMware Cloud Foundation, направлено на оптимизацию использования ресурсов и упрощение развертывания масштабируемых систем.

2. Повышение производительности. Новые поколения гипервизоров и аппаратных решений (например, процессоров с встроенной поддержкой виртуализации) увеличивают производительность ВМ, уменьшая их накладные расходы.

3. Углубленная изоляция. Спрос на высокую безопасность стимулирует разработку ВМ с улучшенной изоляцией, особенно в индустриях, связанных с конфиденциальной информацией, таких как медицина и финансы.

4. Гибридные модели. Разработка решений, объединяющих возможности ВМ и контейнеров, таких как Hyper-V Containers, позволяет использовать преимущества обеих технологий.

4.2.2 Перспективы контейнеров

1. Масштабируемость в облачных платформах. Контейнеризация станет еще более популярной благодаря интеграции с облачными сервисами (например, AWS Fargate или Google Kubernetes Engine), что упростит развертывание приложений.

2. Улучшение инструментов оркестрации. Kubernetes и другие платформы будут развиваться, чтобы лучше поддерживать сложные сценарии развертывания, включая управление несколькими кластерами и автономные системы.

3. Расширение стандартов. Создание стандартов для контейнеров (OCI — Open Container Initiative) способствует их унификации и совместимости между различными платформами.

4. Повышение безопасности. Разработка решений, таких как Secure Containers, направлена на усиление изоляции и защиту контейнеров от угроз.

ЗАКЛЮЧЕНИЕ

В ходе работы было рассмотрено понятие виртуализации, её историческое развитие и значимость в современных условиях. Изучены особенности виртуальных машин и контейнеров, их архитектурные различия, а также примеры успешного применения в различных сферах.

Анализ показал, что виртуальные машины и контейнеры являются взаимодополняющими технологиями, каждая из которых обладает своими преимуществами. Виртуальные машины обеспечивают высокий уровень изоляции и универсальности, что делает их идеальными для работы с разнородными операционными системами. Контейнеры, в свою очередь, выделяются лёгкостью, скоростью развертывания и экономией ресурсов, что особенно важно для современных DevOps-подходов.

Сравнительный анализ выявил критерии выбора между этими технологиями, среди которых ключевыми являются требуемый уровень изоляции, ресурсозатраты и специфические требования проекта.

Подводя итоги, можно утверждать, что виртуализация — это не только эффективный инструмент для оптимизации ИТ-ресурсов, но и основа для развития таких концепций, как облачные вычисления и микросервисная архитектура. Перспективы развития технологий виртуализации связаны с дальнейшей интеграцией искусственного интеллекта, автоматизацией управления ресурсами и повышением уровня безопасности.

Таким образом, виртуализация и её инструменты остаются важнейшими составляющими современной ИТ-инфраструктуры, обеспечивая её гибкость, надёжность и устойчивость к растущим вызовам цифровой эпохи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Виртуализация. — Текст : электронный // Wikipedia : [сайт]. — URL: ru.wikipedia.org/wiki/Виртуализация (дата обращения: 24.12.2024).
2. Виртуализация: типы, принципы работы, особенности. — Текст : электронный // Otus : [сайт]. — URL: <https://otus.ru/journal/virtualizaciya-tipy-principy-raboty-osobennosti/> (дата обращения: 24.12.2024).
3. Краткая история виртуализации, или Зачем вообще что-то делить. — Текст : электронный // ispsystem : [сайт]. — URL: www.ispsystem.ru/news/brief-history-of-virtualization (дата обращения: 24.12.2024).
4. Виртуализация: история развития и технологии аппаратной поддержки. — Текст : электронный // LVEE : [сайт]. — URL: lvee.org/en/reports/LVEE_2011_20 (дата обращения: 24.12.2024).
5. Преимущества и недостатки различных технологий виртуализации. — Текст : электронный // RU WEB : [сайт]. — URL: <https://ruweb.net/articles/preimushchestva-i-nedostatki-razlichnykh-tekhnologij-virtualizacii> (дата обращения: 24.12.2024).
6. Виртуализация — что это и для чего она нужна. — Текст : электронный // Vstack : [сайт]. — URL: ru.vstack.com/blog/virtualizacziya/virtualizacziya/ (дата обращения: 24.12.2024).
7. Что такое виртуальная машина и гипервизор и зачем они нужны?. — Текст : электронный // beseller : [сайт]. — URL: <https://beseller.by/blog/virtualnyye-mashiny-gipervizory/> (дата обращения: 24.12.2024).
8. Знакомство с виртуальными машинами. — Текст : электронный // Habr : [сайт]. — URL: <https://habr.com/ru/companies/timeweb/articles/665786/> (дата обращения: 24.12.2024).
9. VM или Docker?. — Текст : электронный // Habr : [сайт]. — URL: <https://habr.com/ru/articles/474068/> (дата обращения: 24.12.2024).

10. Что такое контейнеризация. — Текст : электронный // Yandex Cloud : [сайт]. — URL: <https://yandex.cloud/ru/docs/glossary/containerization> (дата обращения: 24.12.2024).