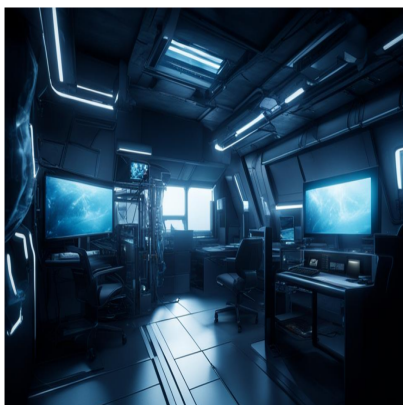




Контейнеры и виртуальные машины: объяснение различий

Описание

Вы только начинаете знакомиться с инструментами визуализации? Или пробиваете себе дорогу в технологическом разговоре о контейнерах и виртуальных машинах. Виртуализация – это процесс, в котором отдельные ресурсы, такие как оперативная память, процессор, сеть и диски, могут быть “виртуализированы” и рассматриваться как несколько ресурсов. Ключевое различие заключается в том, что контейнеры могут виртуализировать только программные слои, построенные выше уровня операционной системы. В отличие от них, виртуальные машины могут виртуализировать целые машины вплоть до аппаратных уровней. Несмотря на четкое различие, между контейнерами и виртуальными машинами есть целый список общих черт, в том числе то, как они повышают эффективность ИТ, обеспечивают переносимость приложений, улучшают DevOps и жизненный цикл разработки программного обеспечения (SDLC). Подробнее об этом позже. В этой статье мы разберем контейнеры и виртуальные машины, рассмотрим преимущества тех и других, а в завершение расскажем о том, как выбрать подходящий инструмент в зависимости от ваших потребностей. Также предполагается, что у вас есть необходимые знания о контейнерах и виртуальных машинах. Если же вы их не получили, это отличное место для начала.

История виртуализации

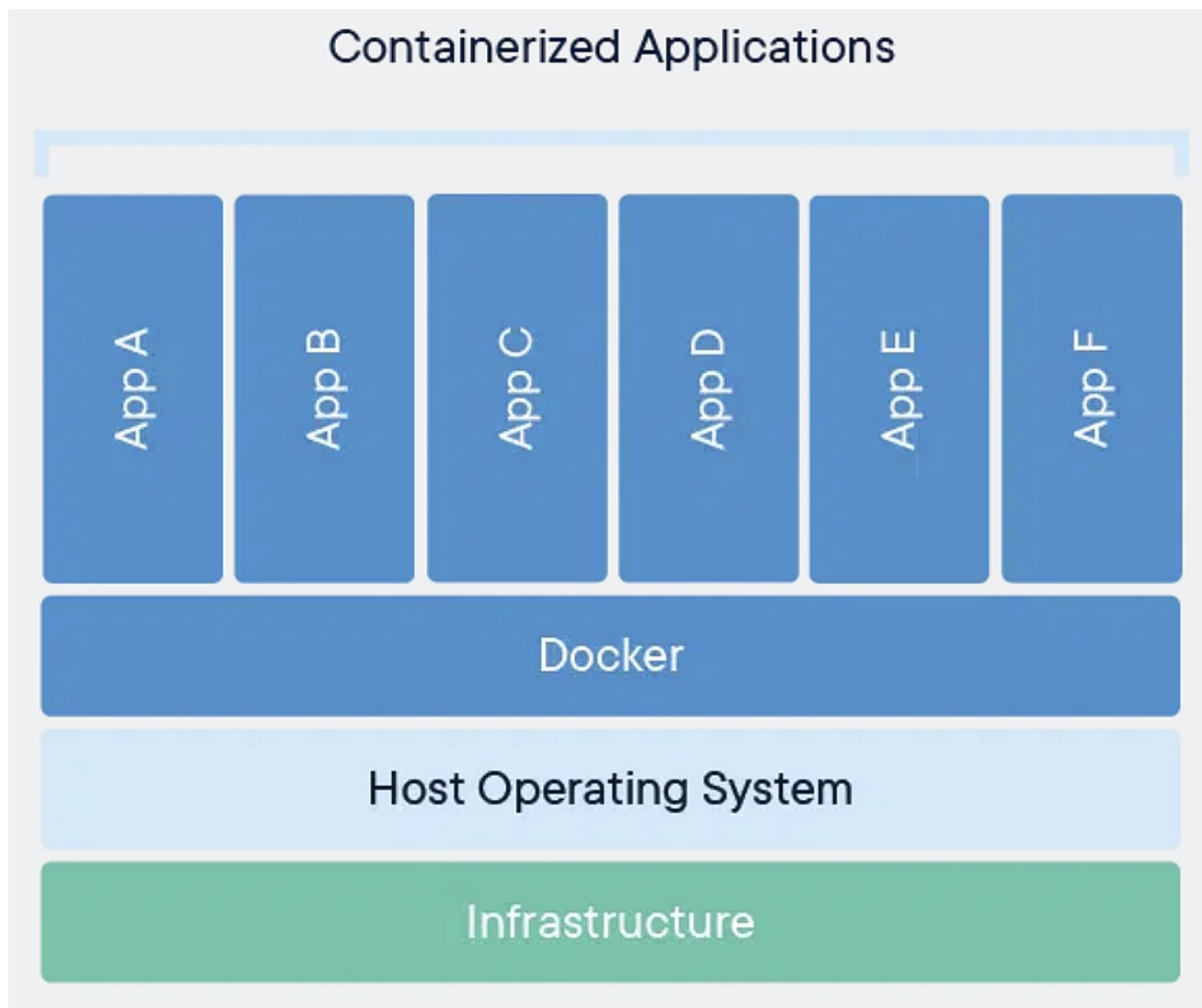
До появления контейнеров виртуальные машины были единственным решением для изоляции сред в рамках физической инфраструктуры. Но только в 2013 году компания Docker выпустила свое первое программное обеспечение для

контейнеризации. С тех пор вы, возможно, заметили растущий интерес к контейнерам и к тому, как они формируют ландшафт облачных вычислений. Многие разработчики проявляют повышенный интерес, особенно к преимуществам гибкой разработки, которую предлагают контейнеры. Однако следует учитывать, что контейнеры и виртуальные машины построены на оптимизации ресурсов существующей физической инфраструктуры.

Как работает виртуализация

Виртуализация подразумевает использование программного обеспечения для создания слоя абстракции над компьютерным оборудованием, позволяющего разделить аппаратные элементы, скажем, одного компьютера, на несколько компьютеров. Такое программное обеспечение называется гипервизором. Гипервизор позволяет различным операционным системам работать параллельно, совместно используя общие физические вычислительные ресурсы. При использовании на физических компьютерах или серверах в центрах обработки данных он позволяет отделить операционные системы (ОС) и приложения от аппаратного обеспечения. После этого он может разделить себя на несколько “виртуальных машин”.

Что такое контейнер?



Контейнеры – это легкий и гибкий метод работы с виртуализацией. Поскольку им не нужны гипервизоры, они позволяют быстрее предоставлять ресурсы и обеспечивать доступность новых приложений. Вы также можете рассматривать контейнеры как программные пакеты, содержащие зависимости, необходимые для выполнения управляемого приложения. Зависимости включают в себя системные библиотеки, пакеты внешнего (стороннего) кода, а также все остальные приложения уровня операционной системы. Все зависимости, включенные в контейнер, существуют на уровнях стека выше, чем операционные системы.

Плюсы

1. **Скорость итераций** – контейнеры легковесны и включают только программное обеспечение высокого уровня; они легко настраиваются, и их можно быстро итерировать. Контейнеры также обеспечивают быструю

доставку программного обеспечения благодаря разработке с учетом тестирования и доступности развертывания.

2. **Надежная экосистема** – В большинстве случаев системы выполнения контейнеров предоставляют уже размещенный публичный репозиторий готовых контейнеров. Репозиторий содержит различные популярные программные приложения, такие как базы данных и системы обмена сообщениями, которые можно загрузить и развернуть, что сокращает время для команд разработчиков.
3. **Масштабирование** – контейнеры предпочтительны благодаря своей способности обрабатывать большие объемы рабочей нагрузки и трафика. Специалисты по информационным технологиям (ИТ) могут использовать контейнеры для управления и предоставления облачной инфраструктуры в условиях экономии ресурсов и гибкого использования.
4. **Микросервисы** – контейнеры идеально подходят для развертывания нескольких микросервисов. Микросервисы – это небольшие компоненты, которые являются частью основного приложения или программного обеспечения. Когда вы упаковываете каждый микросервис в контейнер, вы получаете право на беспрепятственное развертывание, масштабирование и управление.

Минусы

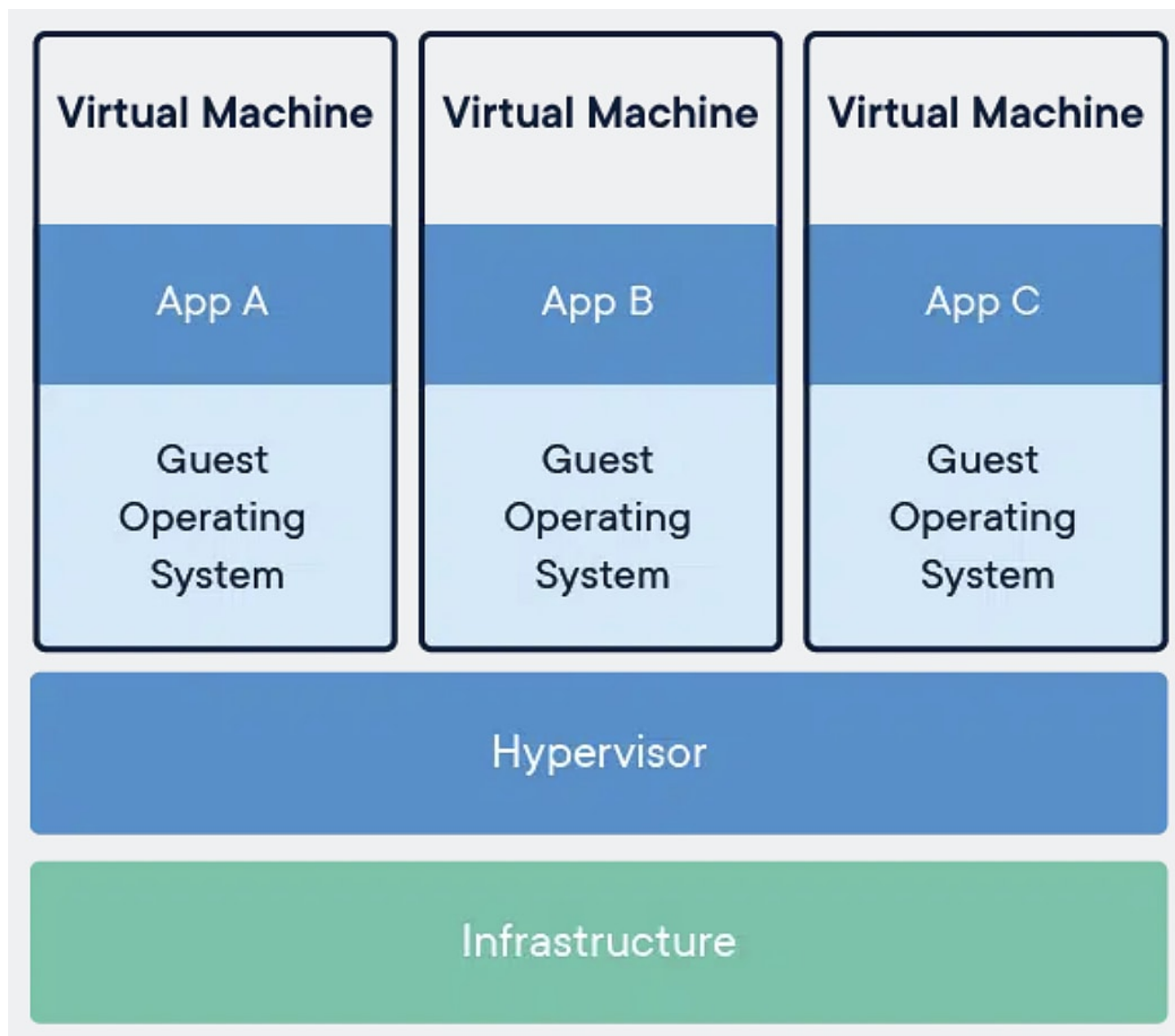
1. **Эксплойты для общего хоста** – поскольку контейнеры используют общую базовую аппаратную систему, построенную ниже уровня операционной системы, существует вероятность того, что эксплойт в одном контейнере может выйти из строя и распространиться, затронув базовое оборудование. Кроме того, тот факт, что большинство контейнеров имеют общедоступные репозитории предварительно собранных контейнеров, создает риск безопасности. Этот риск реализуется в тех случаях, когда скомпрометированный или эксплуатируемый публичный образ выступает в качестве уязвимости, чреватой атаками кибербезопасности.
2. **Работает только с одной операционной системой** – Если вы относитесь к тем, кто использует только одну операционную систему, контейнеры позволят вам запускать более ранние версии, что очень хорошо. Однако контейнеры – не лучший вариант, если вы ориентируетесь на разные операционные системы.

Популярные поставщики контейнеров

Вот некоторые известные поставщики контейнеров:

1. **Docker** – Docker является наиболее распространенным в мире контейнерным решением. Docker Hub – это большой репозиторий для самых популярных контейнерных приложений. Вы можете загрузить контейнеры с Docker Hub и развернуть локальную среду выполнения Docker.
2. **Linux Containers (LXC)** – проект Linux Containers стоит за контейнерной средой выполнения Linux с открытым исходным кодом. LXC изолирует процессы операционной системы друг от друга. Если вы углубитесь в изучение Docker, то обнаружите, что Docker использует LXC в своей базовой архитектуре.

Что такое виртуальная машина?



Виртуальные машины (ВМ), с другой стороны, представляют собой большие (тяжелые) программные пакеты, обеспечивающие полную эмуляцию низкоуровневого аппаратного оборудования, такого как центральный процессор (ЦП), диск и сетевые устройства. ВМ позволяют запускать несколько машин на разных операционных системах, но при этом на одном компьютере. Как уже говорилось, гипервизоры – это способ взаимодействия виртуальных машин с физическими компьютерами. Гипервизоры отделяют виртуальные машины друг от друга и распределяют между ними ресурсы, такие как память и процессоры.

Плюсы

1. **Полная изоляция** – виртуальные машины работают в изоляции как полностью

автономные системы, не подверженные атакам и прерываниям со стороны других виртуальных машин. Если атака совершается на отдельную виртуальную машину, она изолируется, что делает невозможным заражение соседних виртуальных машин.

2. **Интерактивная разработка** – Часто контейнеры представляют собой статичные определения ожидаемых зависимостей и конфигураций, необходимых для запуска контейнера. Виртуальные машины динамичны, что позволяет вести интерактивную разработку. После определения аппаратных потребностей виртуальную машину можно рассматривать как “пустой” компьютер. Вы можете вручную установить программное обеспечение и использовать виртуальную машину для захвата конфигурации текущего состояния. Снимки виртуальной машины можно использовать для контроля версий и, при необходимости, для восстановления виртуальной машины до определенного времени или запуска дополнительных машин с нужной конфигурацией.

Минусы

1. **Скорость итераций** – Виртуальные машины являются полностековым программным обеспечением, поэтому на их создание требуется время. Внесение изменений в снимок виртуальной машины может потребовать времени на итерации и проверку того, что последние обновления ведут себя так, как ожидалось.
2. **Стоимость хранения** данных – исходя из того, что виртуальные машины – это опять же программное обеспечение полного стека, вы, вероятно, ожидаете, что они будут значительными. Да, они занимают много места в хранилище и быстро разрастаются до гигабайтов. Следствием этого является нехватка места либо на машине, на которой размещены ВМ.

Популярные поставщики виртуальных машин

Вот несколько популярных поставщиков виртуальных машин:

1. **Virtualbox** – Virtualbox – одна из наиболее известных платформ для виртуальных машин. Экосистема Virtualbox имеет дополнительные инструменты для разработки и распространения образов виртуальных машин. Это бесплатная система эмуляции архитектуры с открытым исходным кодом, принадлежащая компании Oracle.

2. **VMware** – VMware построена на аппаратном обеспечении архитектуры x86 и является публично торгуемой компанией. Это решение поставляется с гипервизором для развертывания и управления виртуальными машинами. Его предпочитают за надежный пользовательский интерфейс (UI) для управления виртуальными машинами и эффективный корпоративный инструмент, предлагающий функции поддержки.

Контейнеры против виртуальных машин

Хотя виртуальные машины существуют уже столько же лет, им на смену приходят контейнеры для аналогичных целей. Вы видели плюсы и минусы обоих технологических стеков. Вот основные различия.

Характеристика

Большая и менее переносимая, поскольку каждая виртуальная машина имеет свою операционную систему. Виртуальные машины занимают много места, измеряемого в гигабайтах. Виртуальные машины не переносятся.

Контейнеры

Не имеют собственной операционной системы.

Виртуальные машины

Имеют собственные операционные системы и, таким образом, выполняют дополнительные задачи, включая запуск программ, несовместимых с основной ОС, нескольких программ на разных операционных системах и выполнение приложений, которые не могут совместно использовать ресурсы и функциональные возможности операционной системы.

Размер и портативность	Маленькие и портативные платы для операционной системы совместного использования. Контейнеры занимают место, измеряемое в мегабайтах. Контейнеры можно перемещать по разным компьютерам. ВМ работают медленно, поскольку сначала им приходится загружать собственные операционные системы. Загрузка занимает несколько минут.	Контейнеры выполняются быстрее, чем виртуальные машины, благодаря уже запущенной операционной системе. Запуск занимает несколько секунд.
Скорость загрузки	ВМ работают медленно, поскольку сначала им приходится загружать собственные операционные системы. Загрузка занимает несколько минут.	Стоимость виртуальных машин относительно выше – высокая потребность в ресурсах, сложность уровней и высокая цена.
Доступ к ресурсам	Контейнеры имеют доступ ко всем ресурсам хост-машины.	Виртуальные машины используют определенные ресурсы, выделенные им гипервизором.
Безопасность	Менее безопасно. Контейнеры, работающие на общей операционной системе хоста, уязвимы в случае взлома ОС хоста.	Сложное развертывание; уровень сложности виртуальных машин указывает на то, что им требуется больше времени на разработку. Развертывание ничем не отличается от этого.

Развертывание	Простота развертывания; это связано с тем, что по своей природе он “самодостаточен”, и его легко развернуть, переместить или обновить.	Цена виртуальных машин относительно выше. Высокая потребность в ресурсах и уровни сложности, а вместе с ними и высокая стоимость.
Стоимость	Стоимость контейнеров относительно ниже, поскольку к ним предъявляется меньше требований, чем к виртуальным машинам.	Цена виртуальных машин относительно выше – высокая потребность в ресурсах и сложность уровней, а вместе с этим и высокая стоимость.

Как использовать контейнеры и виртуальные машины одновременно

Если вы задавались вопросом, можно ли использовать контейнеры и виртуальные машины вместе, то ответ – да. Хотя практические возможности ограничены. Вы можете создать виртуальную машину для эмуляции определенных аппаратных конфигураций и установки операционной системы. Как только ваша ВМ загрузит операционную систему и станет полностью функциональной, вы получите эмулированную вычислительную систему с определенным оборудованием, на которое можно устанавливать контейнеры. Хороший пример, иллюстрирующий эту конфигурацию, – эксперименты с системой для развертывания чипов. Некоторые популярные методы для вычислительных устройств на чипе, такие как платы разработки BeagleBone и Raspberry Pi, можно эмулировать как виртуальные машины, чтобы поэкспериментировать с рабочими (запущенными) контейнерами перед тестированием на реальном оборудовании. Еще одно преимущество использования союза контейнеров и виртуальных машин – повышение безопасности. Например, вы можете развернуть контейнеры на виртуальных машинах. Рассмотрим пример, в котором десять контейнеров развернуты на одном компьютере, чтобы продемонстрировать, насколько это полезно. Если компьютер

будет взломан, вы рискуете повредить все десять контейнеров. Решение достигается за счет распределения десяти контейнеров между десятью виртуальными машинами. Если одна виртуальная машина будет скомпрометирована, остальные части системного приложения, как правило, продолжают работать.

Заключительные слова

Если у вас есть точные спецификации аппаратного обеспечения для вашего проекта или вы разрабатываете на одном оборудовании, но ориентируетесь на другое, например Windows против MacOS, лучше всего использовать виртуальные машины. В других случаях, когда требования предъявляются к “только программному обеспечению”, лучшим вариантом будет использование контейнеров. В большинстве случаев ваши потребности будут удовлетворены любым из вариантов. Чтобы выбрать лучший из них, поймите свои потребности в ресурсах и связанные с ними компромиссы. Поэтому выбор подходящего инструмента зависит от конкретного проекта. Если вам нужно быстрое и эффективное развертывание, контейнеры – лучший выбор. Если организации необходимо виртуализировать полную операционную систему на аппаратном обеспечении, лучше всего подойдут виртуальные машины. А если вам нужна максимальная безопасность, то лучше всего использовать комбинацию обоих вариантов, однако это связано с определенными затратами.

Дата Создания

18.03.2024