

Системы управления виртуализацией

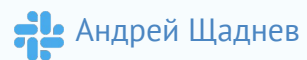


Андрей
Щаднев




Андрей Щаднев

Senior Engineer, Tele2



План занятия

1. [Системы управления виртуализацией](#)
2. [Гипервизор](#)
3. [Коммерческие продукты: VMWare и Hyper-V](#)
4. [Open source продукты: KVM и Xen](#)
5. [IaaS продукты: AWS и OpenStack](#)
6. [Итоги](#)
7. [Домашнее задание](#)



Системы управления виртуализацией

Системы управления виртуализацией

На данный момент на рынке доступен широкий спектр продуктов для управления / оркестрации виртуальных вычислительных систем.

Их можно классифицировать по многим параметрам:

- типу виртуализации;
- масштабу и области применения;
- дистрибуции;
- уровню абстракции;
- open source и коммерческие продукты.

Тип виртуализации

- **Виртуальные машины, включающие виртуальные накопители данных и сетевое оборудование** - Virtual box, Microsoft Hyper-V, KVM, Xen, VMWare, VMWare vSphere, AWS EC2, Azure Virtual Machines
- **Виртуальное окружение** - Vagrant
- **Контейнеры** - Docker, LXD, rkt
- **Виртуальный датацентр (приватное облако)** - Apache Mesos, Openstack

Масштаб и область применения

- **Пользовательская виртуализация**

Vagrant, Virtual box и VMWare являются примерами пользовательской виртуализации в рамках рабочей станции (ПК) и может покрывать задачи конкретного пользователя.

Hyper-V изначально был частью серверной версии Windows, сердцем Microsoft System Center и нацелен на продуктивные высоконагруженные системы. Windows 10 сделала его так же частью системы и реализаций пользовательской виртуализации.

Для примера Minikube или Kubernetes из поставки Docker for desktop запускается именно из-под Hyper-V. Docker, как технология, также является универсальным масштабируемым (сторонними средствами) решением и может применяться как окружение для локальной разработки и тестирования так и в качестве продуктивного окружения.

- **Продуктивная и тестовая инфраструктура**

Остальные системы из списка (VMware vSphere, Hyper-V, KVM, Xen) предусмотрены скорее не для пользовательской виртуализации, а для построение продуктивной и тестовой инфраструктуры проекта комбинируя визуальные машины, реализаций постоянного хранения данных и т.д.

Классификация системах управления виртуализацией

По типу дистрибуции - можно условно разделить доступные продукты на устанавливаемое программное обеспечение и сервисы публичного облака. AWS EC2, Azure Virtual machines являются примерами вышеуказанных сервисов публичного облака, в то время как остальные примеры так или иначе представляют из себя устанавливаемое программное обеспечение и/или часть операционной системы (Hyper-V)

По уровню абстракции - можно выделить гипервизоры - Hyper-v, Xen, KVM, vSphere, Virtual box и прикладные системы - все остальные представленные программные продукты.

Необходимо отметить, что решения для виртуализации могут выступать основой друг для друга - гипервизоры для прикладных систем и/или облачных сервисов, которые в свою очередь используют Xen, KVM и Hyper V. Для примера Vagrant в начале своего развития был привязан к Virtual box - т/е VirtualBox являлся дефакто гипервизором для Vagrant.

По типу лицензирования - Open source и коммерческие продукты - Virtual box, KVM, Xen, Docker; остальные системы из списка являются коммерческими продуктами.



Гипервизор



Понятие гипервизора

Гипервизор - программа или аппаратная схема, обеспечивающая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере.

Понятие гипервизора

- **Гипервизор обеспечивает** изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.
- **Гипервизор может предоставлять** работающим под его управлением на одном хост-компьютере ОС средства связи и взаимодействия между собой так, как если бы эти ОС выполнялись на разных физических компьютерах.
- **Гипервизор является минимальной операционной системой,** и управляет этими виртуальными машинами, выделением и освобождением ресурсов для них. Гипервизор позволяет независимое «включение», перезагрузку, «выключение» любой из виртуальных машин с той или иной ОС.

Типы гипервизоров

- Автономный гипервизор
- На основе базовой ОС
- Гибридный



Типы гипервизоров

Автономный гипервизор

- программа исполняемая непосредственно на аппаратном уровне компьютера и выполняющая функции эмуляции физического аппаратного обеспечения и управления аппаратными средствами и гостевыми ОС. То есть такой гипервизор сам по себе в некотором роде является минимальной операционной системой. Так как автономный гипервизор работает непосредственно на оборудовании, то он более производителен.

Пример: VMware vSphere



Типы гипервизоров

На основе базовой ОС

- специальный дополнительный программный слой, расположенный поверх основной хостовой ОС, который в основном выполняет функции управления гостевыми ОС, а эмуляцию и управление аппаратурой берет на себя хостовая ОС.

Пример: VMware Workstation, VirtualBox


Типы гипервизоров

Гибридный

- объединенный вариант первых двух, в котором функции управления аппаратными средствами выполняются тонким гипервизором и сервисной ОС, работающей под управлением тонкого гипервизора. Обычно гипервизор управляет напрямую процессором и памятью компьютера, а через сервисную ОС гостевые ОС работают с остальными аппаратными компонентами

Гибридный гипервизор состоит из двух частей: из тонкого гипервизора, контролирующего процессор и память, а также работающей под его управлением специальной сервисной ОС в кольце пониженного уровня. Через сервисную ОС гостевые ОС получают доступ к физическому оборудованию.

Пример: Xen, Microsoft Hyper-V



Коммерческие продукты: VMWare и Hyper-V



VMWare

VMWare - Монолитная архитектура



miro



VMWare

VMWare предлагает множество решений для виртуальных систем, в частности vSphere, который используется, как основа для продуктивных систем.

VMware's vSphere использует монолитный дизайн гипервизора, который требует, чтобы драйверы устройств были включены в слой Hypervisor перед использованием продукта виртуализации.

В слое гипервизора находятся компоненты продукта, которые включают в себя:

- планирование ресурсов;
- распределенная файловая система;
- сетевой стек - реализует сеть VMWare.
- стек хранения - позволяет компонентам слоя управления получить доступ к устройствам хранения.



Преимущества:

- нет ограничений по выбору операционной системы, необходимой для управления всеми компонентами;
- нет необходимости в патчах безопасности, для слоя управления;
- хорошая поддержка;
- высокий уровень безопасности.

Недостатки:

- не работает с оборудованием, которое не поддерживает VMWare;
- в зависимости от размера драйверов устройства требуется больше времени инициализации;
- высокие требования для обучения инженеров;
- бесплатная версия не предлагает полную функциональность.

Hyper-V



miro

Hyper-V

Hyper-V от Microsoft позволяет организациям виртуализировать окружения и инфраструктуру, масштабировать службы в публичном облаке, а также создавать частное облако. Hyper-V встроен непосредственно в Windows Server, хотя он также может быть установлен как Hyper-V Server, который является автономным.

Hyper-V использует микроядерную архитектуру, поэтому драйверы устройств работают независимо друг от друга в управляемом слое.

В слое управления также находятся следующие компоненты продукта:

- лайв миграция виртуальных машин;
- SMB и NTFS файловые системы;
- миграция накопителей данных;
- Hyper-V реплики.

Сетевой стек и стек хранения данных не являются частью компонентов продукта, а являются частью гипервизора.

Hyper-V



Преимущества:

- упрощенное управления драйверами устройств, широкий диапазон поддерживаемых устройств;
- простота установки новых ролей сервера;
- время инициализации сокращается благодаря небольшому размеру кода кода Microsoft Hypervisor;
- нет прерывания сервиса для обслуживания или обновления безопасности;
- сервисы могут быть масштабированы;
- более низкий порог вхождения для инженеров.

Недостатки:

- необходимость установки ОС для работы слоя гипервизора;
- ограниченная поддержка версий ОС.

Open source продукты: KVM и Xen

KVM

KVM является представителем аппаратной виртуализации, вы можете запустить практически любую операционную систему в качестве гостевой - BSD / Windows / Linux и с драйвером virtio вы получите близкую к нативной производительность.

KVM поддерживает установку из ISO образа, а также шаблоны установки, он поставляется с хорошим уровнем реализации безопасности, может иметь проблемы с I / O под тяжелой нагрузкой, которая влияет на гостевые и хостовую операционную системы.

Каждая гостевая машина работает, как процесс на хостовой машине, что хорошо для обнаружения источника проблемы, но также при проблемах с доступными ресурсами на хостовой машине все гостевые подвержены проблемам с производительностью.

KVM

KVM является нативным для большинства современных ядер Linux.

Это дает преимущество в производительности по сравнению с другими системами виртуализации. При этом данный продукт относительно новый и находится в активном развитии.

Большинство людей выбирают KVM для лучшей производительности и гибкости, хотя эта система не столь стабильна, как Xen.



Xen

Xen поставляется в двух вариациях, и может работать одновременно на том же физическом хосте в режиме **Xen PV** (паравиртуализация) и **HVM** (полная аппаратная виртуализация).

Гостевые машины в режиме Xen PV обычно основаны на шаблонах для быстрого развертывания и высокой производительности.

Вы можете запускать свое собственное ядро в Xen PV. По умолчанию вы можете запустить только Linux в этом режиме. Запуск BSD возможен с дополнительной конфигурацией.

Режим Xen HVM работает примерно так же, как KVM. Он имеет лучшие драйверы для Linux-дистрибутивов, однако в NetBSD и Windows, Xen HVM показывает себя хуже по сравнению с KVM, в то время как Xen Pv и Windows более совместимы, но реализация поддержки Windows в Xen в целом хуже, чем в KVM.

Xen

Xen довольно старый и очень зрелый продукт, большинство людей выбирают его для хорошей производительности и исключительной стабильностью.

Xen-хосты, как правило, предварительно выделяют ОЗУ и ядра для гипервизора, поэтому он имеет свои собственные выделенные ресурсы, на которые не могут повлиять гостевые ОС, что и дает стабильности в работе.

VMWare vs Hyper-V vs Xen vs KVM

- **VMware vSphere** является наиболее сбалансированным и универсальным продуктом для организаций с высокими требованиями к их виртуальной инфраструктуре.
- **Hyper-V** является выбором defacto для окружений с преобладанием технологий Microsoft. Богатый дополнительный функционал, высокое быстродействие и относительная простота в изучении может также стать решающими в выборе данной системы
- Если вы хотите сэкономить деньги и ваши окружения представляют собой исключительно Linux, то **KVM** может стать неплохим вариантом платформы виртуализации, предоставляя пользователям высокое быстродействие и поддержкой Windows, либо выбрать **Xen** с его большей надежностью и стабильностью.

Необходимо отметить, что оптимальный выбор системы очень зависит от конкретного сценария развертывания.

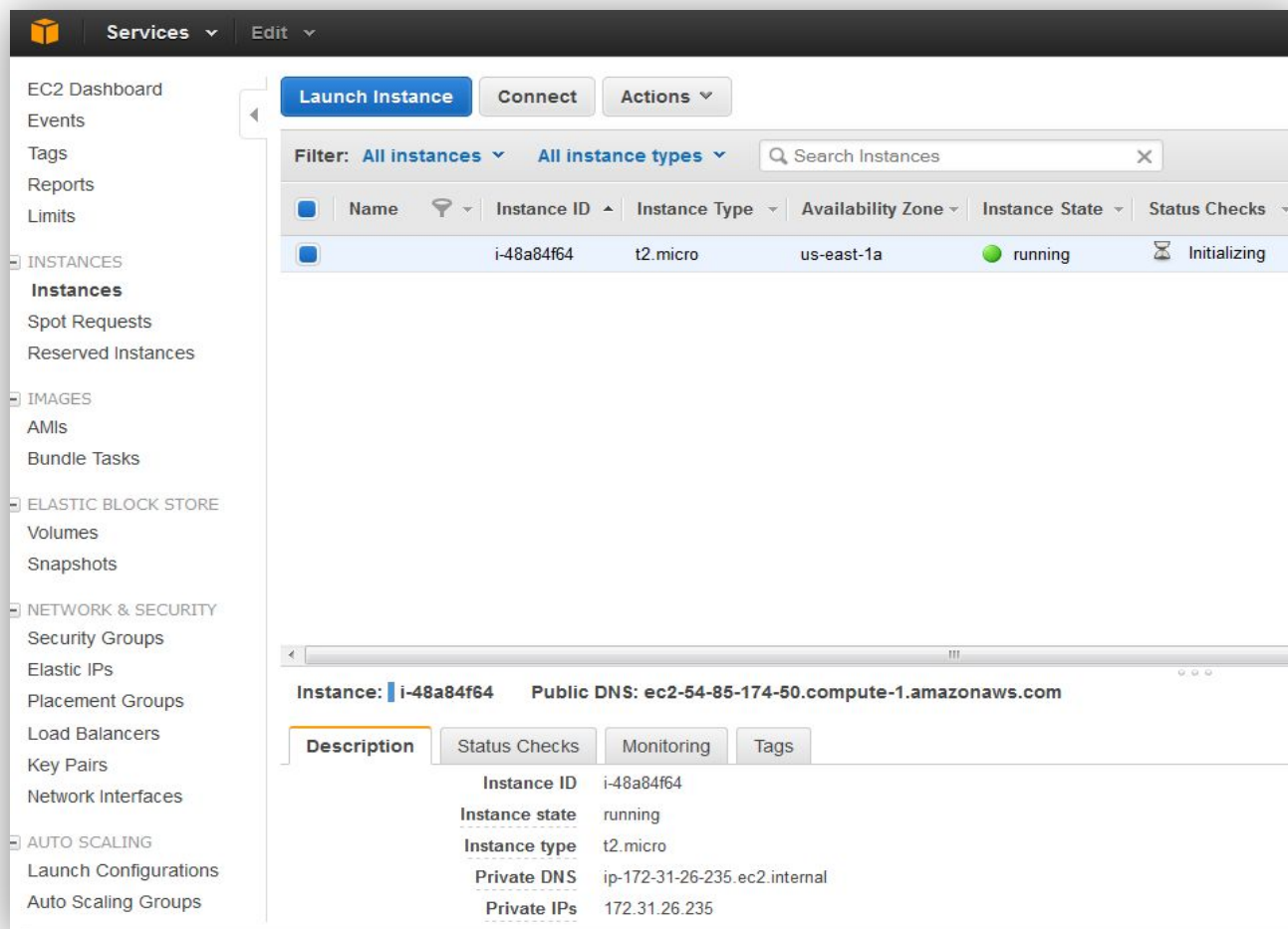
IaaS продукты: AWS и OpenStack

AWS EC2

Сервисы облачных провайдеров - самая современная реализация систем управления виртуализацией. Гипервизоры и прочая реализация системы скрыта от пользователя, при этом есть гарантированные показатели доступности и отказоустойчивости.

Пользователь может создавать инфраструктуру продуктивных окружений и использовать уникальные преимущества такие, как установка в нескольких локациях, построение цепочки с другими сервисами облака, программное API для автоматизации создания и изменения ресурсов

AWS EC2



Services **Edit**

EC2 Dashboard
Events
Tags
Reports
Limits

INSTANCES
Instances
Spot Requests
Reserved Instances

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces

AUTO SCALING
Launch Configurations
Auto Scaling Groups

Launch Instance **Connect** **Actions**

Filter: **All instances** **All instance types** Search Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-48a84f64	t2.micro	us-east-1a	running	Initializing

Instance: **i-48a84f64** Public DNS: **ec2-54-85-174-50.compute-1.amazonaws.com**

Description **Status Checks** **Monitoring** **Tags**

Instance ID	i-48a84f64
Instance state	running
Instance type	t2.micro
Private DNS	ip-172-31-26-235.ec2.internal
Private IPs	172.31.26.235

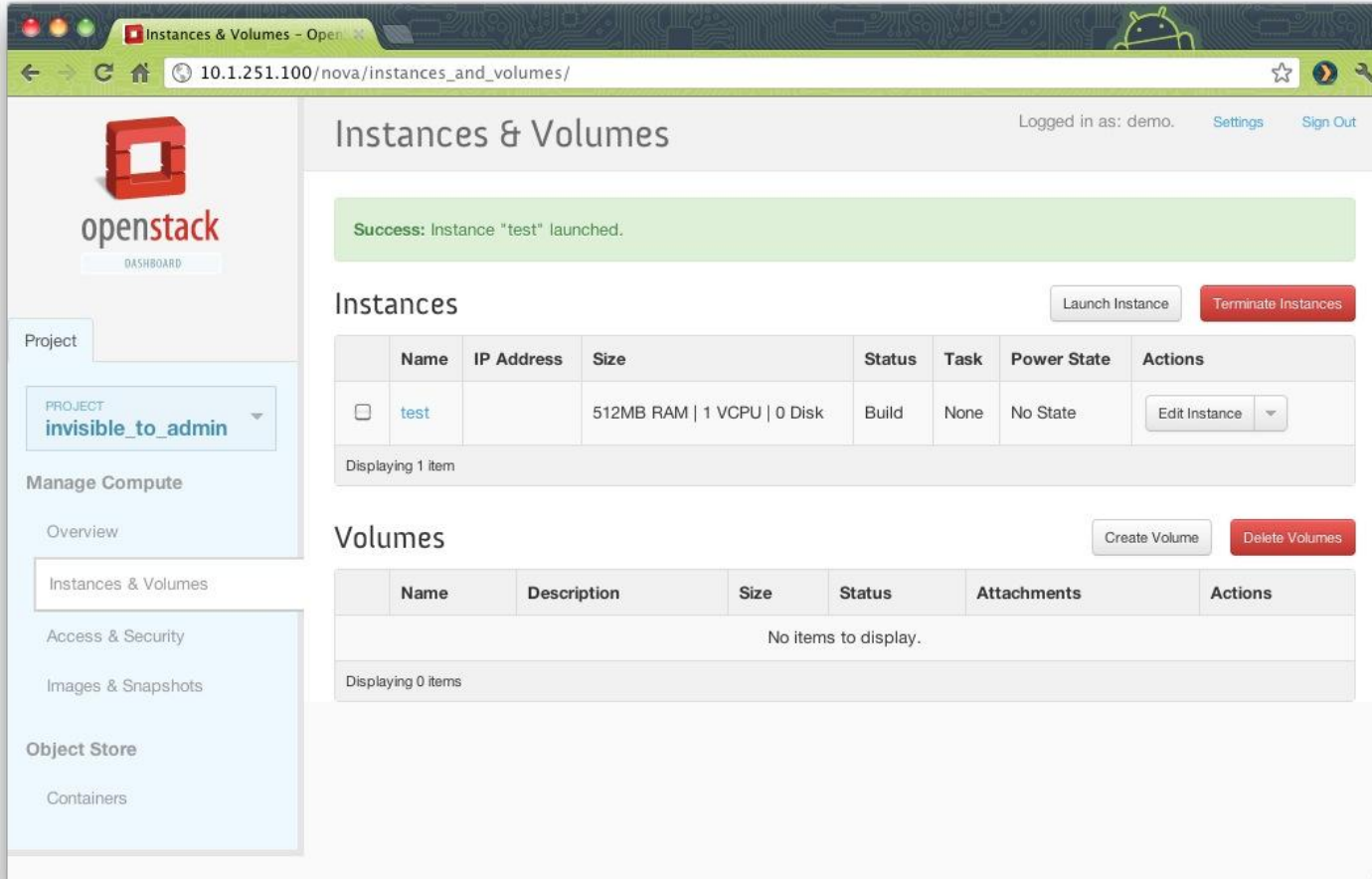
OpenStack

Приватные облака являются программной надстройкой для личного ЦОД или ко-локейшн, и дают часть преимуществ публичного облака, такие как:

- построение продуктивных цепочек с другими компонентами инфраструктуры;
- программное API, являющееся самым важным преимуществом.

Приватные облака не могут похвастаться гарантированными уровнями доступности и отказоустойчивости, а также многими локациями для установки, но имеют более высокий уровень приватности, необходимый для конкретных проектов.

OpenStack




The screenshot shows the OpenStack web interface for managing instances and volumes. The browser address bar shows the URL `10.1.251.100/nova/instances_and_volumes/`. The user is logged in as 'demo'. A success message indicates that an instance named 'test' has been launched.

Instances & Volumes

Logged in as: demo. [Settings](#) [Sign Out](#)

Success: Instance "test" launched.

Instances [Launch Instance](#) [Terminate Instances](#)

	Name	IP Address	Size	Status	Task	Power State	Actions
	test		512MB RAM 1 VCPU 0 Disk	Build	None	No State	Edit Instance ▼

Displaying 1 item

Volumes [Create Volume](#) [Delete Volumes](#)

Name	Description	Size	Status	Attachments	Actions
No items to display.					

Displaying 0 items

Left Sidebar:

- Project
 - PROJECT: invisible_to_admin ▼
- Manage Compute
 - Overview
 - Instances & Volumes
 - Access & Security
 - Images & Snapshots
- Object Store
 - Containers



Итоги

Итоги

- Мы познакомились с основными системами управления виртуализацией, их классификацией, преимуществами и недостатками каждой из них, а также понятием гипервизора и его типами
- Также узнали о более комплексном подходе для виртуализации таким, как инфраструктура как сервис и его реализациях в виде публичных и частных облаков



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и
пишите отзыв о лекции!**

Андрей Щаднев