

Андрей Маркелов

OPENSTACK

Практическое
знакомство с облачной
операционной системой



УДК 004.738.5:004.451.9OpenStack
ББК 32.972.53
M26

Маркелов А. А.

M26 OpenStack: практическое знакомство с облачной операционной системой. – 3-е изд., перераб. и доп. – М.: ДМК Пресс, 2017. – 268 с.: ил.

ISBN 978-5-97060-520-2

В книге описана работа с де-факто стандартом в области открытых продуктов построения облачной инфраструктуры типа «Инфраструктура как сервис» (IaaS) – OpenStack. Данный проект пользуется поддержкой более двух сотен компаний, включая практически всех лидеров ИТ-рынка.

Помимо теоретических знаний, книга содержит множество практических упражнений, благодаря которым читатель сможет развернуть собственный стенд для тестирования возможностей облака. Читатель познакомится с основными сервисами облачной операционной системы OpenStack. Рассмотрены такие компоненты, как сервис идентификации пользователей, сервисы построения объектного хранилища (Swift), блочного (Glance), хранилища образов. В главе, посвященной сетевым службам, рассматриваются концепции программно-определяемой сети, работа с OpenStack Neutron и Open vSwitch. Также рассматриваются сервис мониторинга Ceilometer, сервис оркестрации Heat и принципы обеспечения высокой доступности облака. Кроме того, отдельные главы рассказывают об интеграции OpenStack с контейнерами Docker и с хранилищем Ceph.

В третьем издании книга обновлена и дополнена в соответствии с изменениями в проекте OpenStack на начало 2017 года (версия Newton). Значительно переработаны главы, посвященные сервисам идентификации и телеметрии. Расширены главы, посвященные сетевой службе и работе с виртуальными машинами из командной строки.

Издание рассчитано на ИТ-специалистов (системных и сетевых администраторов, а также администраторов систем хранения данных), желающих изучить работу с OpenStack.

УДК 004.738.5:004.451.9OpenStack
ББК 32.972.53

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-97060-520-2

© Маркелов А. А., 2017
© Оформление, ДМК Пресс, 2017

Содержание

Предисловие	6
Глава 1. Введение в OpenStack	13
Что такое облачная инфраструктура?.....	13
Что такое облачные приложения?.....	15
История OpenStack.....	16
Архитектура OpenStack.....	18
Гипервизор KVM и эмулятор QEMU.....	21
Дистрибутивы OpenStack.....	23
Глава 2. Настройка лабораторного окружения OpenStack	27
Подготовка CentOS 7 к использованию дистрибутива OpenStack RDO.....	33
Отличия RDO от «Upstream».....	36
Как установить OpenStack RDO одной командой?.....	36
Как установить OpenStack одной командой из исходных кодов?.....	39
Как определить, какую версию OpenStack я использую?.....	41
Установка и настройка брокера сообщений.....	42
Установка и настройка базы данных.....	46
Переход на использование утилиты OpenStackClient.....	47
Глава 3. Сервис идентификации Keystone	49
Терминология Keystone.....	49
Установка и настройка Keystone.....	50
Работа с пользователями, ролями и проектами в Keystone.....	57
Глава 4. Сервис хранения образов Glance	65
Установка и настройка сервиса Glance.....	66
Подготовка образов виртуальных машин.....	71
Работаем с образами виртуальных машин.....	76
Глава 5. Сервис блочного хранилища Cinder	80
Архитектура Cinder.....	80
Настройка сервисов Cinder.....	82
Создание и удаление томов Cinder.....	87
Глава 6. Объектное хранилище Swift	90
Архитектура Swift.....	91
Подготовка дополнительных серверов лабораторного окружения.....	93
Установка сервиса Swift-проху.....	95
Установка узлов хранения Swift.....	95
Создание сервисных колец Swift.....	99

Завершение настройки.....	100
Работа с сервисом Swift	101
Настройка Swift в качестве хранилища для Glance	103
Рекомендации по поиску неисправностей в сервисах Swift	105
Глава 7. Контроллер и вычислительный узел Nova.....	106
Архитектура Nova	106
Установка контроллера Nova	107
Установка вычислительных узлов Nova	112
Глава 8. Службы сети Neutron	117
Архитектура Neutron	117
Работа Neutron при создании экземпляра виртуальной машины	120
Установка узла управления Neutron	121
Установка сетевого узла Neutron	128
Установка вычислительного узла Neutron	131
Глава 9. Работа с виртуальными машинами из командной строки	134
Сеть в OpenStack.....	134
Запускаем экземпляр виртуальной машины	139
Добавляем к экземпляру виртуальной машины сеть.....	149
Моментальные снимки и резервные копии.....	152
Шифрование томов Cinder.....	156
Квоты на ресурсы.....	157
Зоны доступности и агрегирование вычислительных узлов в Nova.....	159
Зоны доступности в Cinder	163
Живая миграция виртуальных машин	164
Настройка экземпляров виртуальных машин при помощи cloud-init.....	168
Глава 10. За фасадом Neutron.....	170
Виртуальный коммутатор Open vSwitch.....	170
Группы безопасности.....	176
Утилита для визуализации сети plotnetcfg.....	178
Зеркалирование трафика на Open vSwitch для мониторинга сети в OpenStack	179
Балансировщик нагрузки как сервис (LBaaS)	183
Глава 11. Веб-панель управления Horizon и работа пользователя из графического интерфейса.....	186
Установка веб-интерфейса.....	186
Работа с OpenStack в интерфейсе Horizon.....	188
Глава 12. Сервис сбора телеметрии	196
Установка служб Ceilometr управляющего узла.....	198

Установка службы Aodh	203
Установка служб вычислительного узла для отправки сообщений телеметрии.....	204
Интеграция с сервисами Glance и Cinder.....	205
Работа со службой телеметрии	206
Глава 13. Сервис оркестрации Heat.....	211
Архитектура сервиса.....	212
Установка сервисов Heat	213
Запуск простого стека	216
Запуск стека, состоящего из нескольких виртуальных машин	220
Глава 14. Контейнеры Docker и OpenStack	222
Краткое знакомство с Docker.....	222
Совместное использование Docker и OpenStack.....	223
Настройка работы драйвера Docker для OpenStack Nova.....	225
Глава 15. Программно-определяемая система хранения данных Ceph.....	231
Архитектура Ceph.....	233
Быстрая установка кластера Ceph при помощи ceph-deploy.....	237
Установка кластера ceph вручную.....	241
Интеграция Ceph с сервисами OpenStack.....	246
Глава 16. Отказоустойчивость и производительность OpenStack	253
Обзор способов обеспечения высокой доступности сервисов облака	253
Выделение вычислительных ресурсов	255
Выделение оперативной памяти.....	256
Производительность сети	258
Определение аппаратных требований к оборудованию.....	259
Заключение	260
Приложение 1. Пример правил брандмауэра, реализующих группы безопасности на вычислительном узле.....	261
Приложение 2. Листинги шаблонов Heat	263
Приложение 3. Список основных используемых службами OpenStack сетевых портов	267

Глава 1

Введение в OpenStack

Что такое облачная инфраструктура?

Согласно Wikipedia, OpenStack – это свободная и открытая платформа для облачных вычислений. Для начала определимся с тем, что такое облачная платформа. Устоявшимся в индустрии определением является определение, данное National Institute of Standards and Technology (NIST):

Облачные вычисления – это модель предоставления широкодоступного, удобного доступа по сети к общему пулу настраиваемых вычислительных ресурсов по требованию (к таким, как сети, серверы, системы хранения данных, приложения и сервисы). Эти ресурсы оперативно выделяются и освобождаются при минимальных усилиях, затрачиваемых заказчиком на организацию управления и на взаимодействие с поставщиком услуг.

Этой модели присущи пять основных характеристик, три сервисные модели и четыре модели внедрения. В число характеристик входят: самообслуживание, универсальный доступ по сети, общий пул ресурсов, эластичность и учет потребления.

Сервисные модели различаются по границе контроля потребителем услуг предоставляемой инфраструктуры и включают в себя:

- инфраструктуру как сервис (IaaS) – собственно, этой сервисной модели и посвящена данная книга, поскольку OpenStack используют в основном именно для развертывания облаков этого типа. В данном случае пользователь получает контроль за всеми уровнями стека программного обеспечения, лежащими выше облачной платформы, а именно: виртуальными машинами, се-

тиями, выделенным пользователю объемом пространства на системе хранения данных (СХД). В этом случае пользователь выступает администратором операционной системы и всего, что работает поверх, вплоть до приложений. Примерами платформ, обеспечивающих подобную модель, помимо OpenStack, можно назвать Apache CloudStack, Eucalyptus и OpenNebula;

- программное обеспечение как сервис (SaaS) – в этом случае граница контроля пользователя – само приложение. Пользователь в данном случае может даже не знать, что такое виртуальная машина или операционная система, он просто работает с приложением. Примеры таких облачных продуктов: Google Docs, Office 365 или, например, Яндекс-почта.

Четыре модели внедрения облачной платформы включают в себя:

- платформу как сервис (PaaS) – облако, построенное по такой модели, вполне может располагаться «внутри» облака модели IaaS. В этом случае граница контроля пользователя лежит на уровне платформы построения приложений, например сервера приложения, библиотек, среды разработки или базы данных. Пользователь не контролирует и не администрирует виртуальные машины и операционные системы, установленные на них, СХД и сети. Примеры облачных платформ модели PaaS: Apache Stratos, Cloud Foundry, Deis и OpenShift Origin;
- частное облако – вся инфраструктура развернута в центре обработки данных (ЦОД) и служит подразделением одной компании или группы компаний;
- публичное облако – заказчиком облачных услуг может выступать любая компания или даже частное лицо. Это модель внедрения, на которой зарабатывают провайдеры облачных услуг;
- облако сообщества, или общественное облако. Модель, при которой потребителем является сообщество потребителей из организаций, имеющих общие задачи (например, миссии, требований безопасности, политики и соответствия различным требованиям);
- гибридное облако – это комбинация из двух или трех вышеописанных облаков, где разная нагрузка может располагаться как в частном, публичном или общественном облаке. Как правило, гибридное облако – это больше, чем просто сумма облаков, поскольку ему требуются механизмы и инструменты централизованного управления, распределения и миграции нагрузки между облачными инфраструктурами.

Что такое облачные приложения?

За время работы в компаниях, занимающихся как внедрением, так и разработкой облачных решений, автор столкнулся с тем, что потенциальные заказчики зачастую плохо представляют себе разницу в области применения облачной платформы, подобной OpenStack, и традиционных систем виртуализации. Очень часто, когда заказчик считал, что ему нужен OpenStack, на самом деле ему нужна была традиционная система промышленной виртуализации типа VMware vSphere или Microsoft Hyper-V, ориентированная не на облачные приложения, масштабируемые горизонтально, а на традиционные промышленные приложения, масштабируемые вертикально.

Проще всего объяснить различия между облачными и традиционными приложениями можно на примере аналогии.

С одной стороны, есть домашние животные – собака или кошка. Они долго живут с вами, вы знаете, как их зовут, и их характеры. Если домашнее животное болеет, вы ведете его к ветеринару. Смерть домашнего питомца будет для вас трагедией.

С другой стороны, есть стадо, например, коров. Вы знаете их общее число. Они не имеют индивидуальных имен и все взаимозаменяемы. Несчастный случай с одной коровой не означает того, что вы перестали быть ковбоем и потеряли стадо.

Традиционные приложения, например база данных или почтовый сервер, – это «домашние питомцы», они требуют функционала «системы промышленной виртуализации». Этот функционал включает в себя, но не ограничивается средствами обеспечения высокой доступности, живой миграции, резервного копирования, возможность добавлять в виртуальную машину ресурсы или забирать их. Жизненный цикл таких виртуальных машин – как правило, годы.

Современные облачные приложения – это аналог «стада» виртуальных машин. Они масштабируются горизонтально, добавлением виртуальных машин. Приложение пишется таким образом, что каждая виртуальная машина сама по себе не является критичной для функционирования всего приложения и не требует высокой доступности. Также вполне возможно, что виртуальные машины работают без сохранения состояния, что не требует их резервного копирования. Жизненный цикл подобных виртуальных машин – как правило, месяцы.

Именно для обеспечения работы облачных приложений в первую очередь и создавался OpenStack. В связи с этим не удивляйтесь, что

в составе функционала OpenStack отсутствует, например, высокая доступность виртуальных машин.

История OpenStack

Проект по разработке облачной операционной системы OpenStack появился в июне 2010 года как проект, объединивший разработку Национального космического агентства США (NASA) для создания виртуальных серверов Nova и программную систему хранения данных Swift от американского же хостинг-провайдера Rackspace. Первая версия под кодовым названием Austin вышла в октябре того же года.

По соглашению версии обозначаются именем, а порядковый номер в латинском алфавите первой буквы имени определяет номер версии Openstack: A – первая версия, B – вторая и т. д.

Уже в Vexar вдобавок к Nova и Swift появился третий сервис, предназначенный для хранения образов Glance. В Exssex появились веб-консоль управления Horizon и сервис идентификации Keystone. В Folsom – сервис сети, первоначально названный Quantum, но затем поменявший имя, так как оно совпадало с зарегистрированной торговой маркой, и сервис блочного хранения Cinder. В Havana добавился сервис оркестрации Heat и мониторинга Celiometer.

Важно понимать, что сам по себе OpenStack – это проект по разработке. Сайт проекта Openstack.org не предоставляет эталонного дистрибутива. Напротив, вендоры на основе кода проекта OpenStack создают свои дистрибутивы. Обзор некоторых из них приведен в разделе «Дистрибутивы и вендоры OpenStack».

Нужно отметить, что уже через год после выхода Austin к разработке OpenStack присоединились именитые ИТ-вендоры, как то: Dell, HP и Cisco.

В настоящий момент OpenStack реализуется под руководством OpenStack Foundation с числом индивидуальных членов около семи-десяти тысяч и корпоративных – более шести сотен. OpenStack поддерживают практически все ИТ-лидеры рынка. Бюджет OpenStack Foundation – более шестнадцати миллионов долларов в год.

Согласно отчету Linux Foundation, на настоящий момент OpenStack – это более 20 миллионов строк кода. Основной язык программирования – Python (71% от всего объема кода). Сам код распространяется под лицензией Apache 2.0.

К числу компаний, которые имеют самые большие инсталляции, относятся Bluehost, Canonical, CloudScaling, EasyStack, eNovance

(куплена Red Hat), HP, IBM, Metacloud, Mirantis, Oracle, Piston, Rackspace, Red Hat, SUSE, SwiftStack.

Если оценивать вклад компаний в разработку проекта OpenStack, то проще всего обратиться к сайту <http://stackalytics.com>. Этот сервис изначально был создан компанией Mirantis для сбора статистики и вклада своих инженеров в проект и его отдельные части, а затем стал использоваться большинством компаний, разрабатывающих OpenStack.

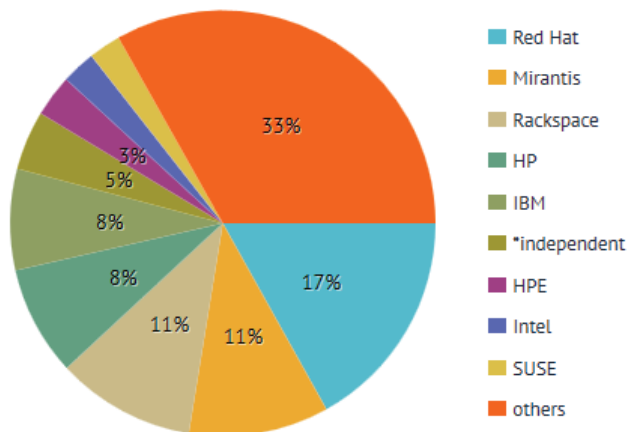


Рис. 1.1 ❖ Статистика в сервисе Stackalytics на начало 2017 года

В качестве примера на рисунке приведена диаграмма, показывающая вклад компаний в проект OpenStack за все время его существования (по состоянию на начало 2017 года). Как мы видим, в пятерку топ-контрибуторов входят Red Hat, Mirantis, Rackspace, IBM и HP/HPE.

Также интересным, с точки зрения трендов, в использовании OpenStack можно считать регулярные срезы с опросника пользователей с сайта проекта <https://www.openstack.org/user-survey>. На момент написания книги последним был опрос от апреля 2016 года. Скорее всего, после выхода книги будет доступна информация по следующему срезу статистики. Какую же интересную информацию оттуда можно получить?

- 71% принявших участие в опросе используют OpenStack для размещения производственной нагрузки.

- Подавляющее большинство инсталляций – в IT-компаниях (68%), академической среде (9%) и телекоммуникационных компаниях (14%).
- Европейские страны делят второе место с Азией в качестве региона проживания принявших участие в опросе (23% всех принявших участие). На первом месте – Северная Америка (46%).
- По типам нагрузки на первом месте – среды разработки и тестирования, затем инфраструктурные сервисы, веб-приложения, базы данных, хранение данных и «большие данные» с аналитикой.
- 93% всех внедрений (промышленных и тестовых) используют гипервизор KVM.
- Также интересной является статистика по средам развертывания и управления облака: 42% – Puppet, 42% – Ansible, 19% – Fuel, 12% – Chef.
- 39% в промышленной эксплуатации и 57% всего используют программную СХД Ceph.
- По драйверам сети суммарная статистика следующая: 60% – Open vSwitch, 31% – ML2, 21% – Linux Bridge.
- Дистрибутивы ОС распределились следующим образом: 74% – Ubuntu, 27% – CentOS, 20% – RHEL.

Согласно отчету Forrester Research (<http://www.openstack.org/assets/pdf-downloads/OpenStack-Is-Ready-Are-You.pdf>), OpenStack в настоящее время используют многие компании из списка Fortune 100, такие как BMW, Disney и Wal-Mart.

Ну и, наконец, перед тем как двигаться дальше, возможно, читателю будет интересно ознакомиться с порталом <https://www.openstack.org/enterprise/>, где приведены примеры промышленной эксплуатации OpenStack на предприятиях.

Архитектура OpenStack

Проект OpenStack, который также называют облачной операционной системой, состоит из ряда отдельных проектов, разрабатывающих отдельные подсистемы. Конкретная установка OpenStack может включать в себя лишь часть из них. Некоторые подсистемы могут использоваться вообще автономно или как часть других OpenSource-проектов. В этой книге рассматриваются лишь базовые сервисы. Их набор увеличивается от версии к версии проекта OpenStack как за счет появления новых, так и за счет разделения функционала существующих.

вующих. Например, сервис nova-volume выделился в отдельный проект Cinder.

Каждый из проектов имеет свой документированный набор REST API, утилит командной строки и «родные» интерфейсы Python, предоставляющие набор функций, аналогичных утилитам командной строки.

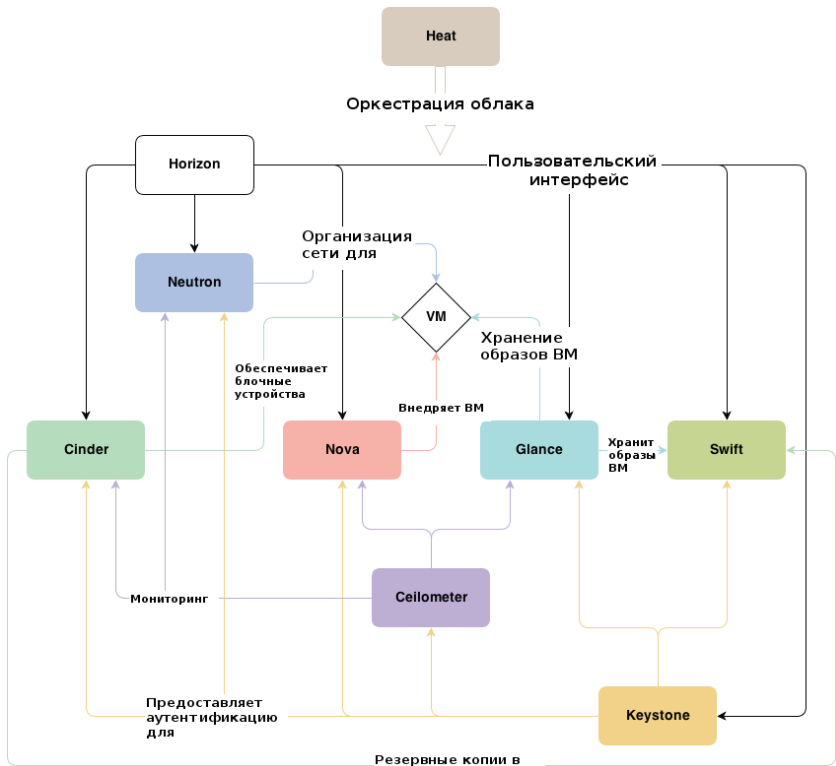


Рис. 1.2 ❖ Архитектура OpenStack

Одним из базовых сервисов является **OpenStack Compute (Nova)**. Этот сервис устанавливается на всех вычислительных узлах кластера. Он предоставляет уровень абстракции виртуального оборудования (процессоры, память, блочные устройства, сетевые адаптеры). Nova обеспечивает управление экземплярами виртуальных машин, обращаясь к гипервизору и отдавая такие команды, как, например, их запуск и остановку.

Важно отметить, что технологии OpenStack независимы от гипервизора. По адресу <https://wiki.openstack.org/wiki/HypervisorSupportMatrix> располагается матрица совместимости гипервизоров. Поддержка реализуется через соответствующие драйверы в проекте Nova. Разработка и тестирование OpenStack ведутся в первую очередь для KVM. Большинство внедрений также завязано на гипервизор KVM.

Следующий сервис под названием **OpenStack Networking (Neutron)** отвечает за сетевую связанность. Пользователи могут самостоятельно создавать виртуальные сети, маршрутизаторы, назначать IP-адреса. Один из механизмов, обеспечиваемых Neutron, называется «плавающие адреса». Благодаря этому механизму виртуальные машины могут получать внешние фиксированные IP-адреса. Через механизм подключаемых модулей можно реализовать такой функционал, как балансировщик сетевой нагрузки как сервис, брандмауэр как сервис и VPN как сервис.

Служба идентификации **OpenStack Keystone** представляет собой централизованный каталог пользователей и сервисов, к которым они имеют доступ. Keystone выступает в виде единой системы аутентификации облачной операционной системы. Keystone проверяет действительность учетных записей пользователей, проверяет сопоставление пользователей проектам OpenStack и ролям и в случае успеха выдает токен на доступ к другим сервисам. Также Keystone ведет каталог служб.

OpenStack Image Service (Glance) ведет каталог образов виртуальных машин, которые пользователи могут использовать как шаблоны для запуска экземпляров виртуальных машин в облаке. Также данный сервис предоставляет функционал резервного копирования и создания моментальных снимков. Glance поддерживает множество различных форматов, включая vhd, vmdk, vdi, iso, qcow2, ami и др.

Сервис **OpenStack Block Storage (Cinder)** управляет блочным хранилищем, которое могут использовать запущенные экземпляры виртуальных машин. Это постоянное хранилище информации для виртуальных машин. Можно использовать моментальные снимки для сохранения и восстановления информации или для целей клонирования. Чаще всего с cinder используют хранилище на основе Linux-серверов, однако имеются и подключаемые модули для аппаратных хранилищ.

Сервис **OpenStack Object Storage (Swift)**, помимо Nova, является одним из двух первых проектов, появившихся в OpenStack. Изначально он назывался Rackspace Cloud Files. Сервис представляет собой объектное хранилище, позволяющее пользователям хранить

файлы. Swift имеет распределенную архитектуру, обеспечивая горизонтальное масштабирование, а также избыточность и репликацию для целей отказоустойчивости. Swift ориентирован преимущественно на статические данные, такие как образы виртуальных машин, резервные копии и архивы.

Сервис **OpenStack Telemetry (Ceilometer)** представляет собой централизованный источник информации по метрикам облака и данным мониторинга. Этот компонент обеспечивает возможность биллинга для OpenStack.

OpenStack Orchestration (Heat) – сервис, задача которого – обеспечение жизненного цикла приложения в облачной инфраструктуре. При помощи шаблона формата AWS CloudFormation сервис управляет остальными сервисами OpenStack, позволяя создать большинство типов ресурсов (виртуальные машины, тома, плавающие IP, пользователи, группы безопасности и т. д.). Heat при помощи данных Ceilometer также может осуществлять автоматическое масштабирование приложения. Шаблоны описывают отношения между ресурсами, и это позволяет сервису Heat осуществлять вызовы API OpenStack в правильном порядке, например сначала создать сервер, а потом подключить к нему том.

И наконец, самый близкий к пользователю облака сервис **OpenStack Dashboard (Horizon)**, позволяющий управлять ресурсами облака через веб-консоль.

Также существует проект, чей код используется большинством остальных компонентов OpenStack, – это проект, объединяющий общие библиотеки компонентов облака **OpenStack Oslo**.

Гипервизор KVM и эмулятор QEMU

Как уже отмечалось, самым используемым гипервизором совместно с OpenStack является KVM (Kernel-based Virtual Machine). KVM – часть ядра Linux с 2007 года и требует аппаратной поддержки виртуализации на серверах стандартной архитектуры (AMD-V или Intel VT-x). В настоящее время KVM также адаптирован для ряда других платформ, например PowerPC. Для эмуляции устройств ввода/вывода в GNU/Linux используется эмулятор QEMU.

Обратите внимание, что аппаратная поддержка виртуализации может быть по умолчанию выключена в BIOS. Примеры включения опций в BIOS для ПК с процессорами AMD и Intel приведены на рис. 1.3, 1.4 и 1.5.

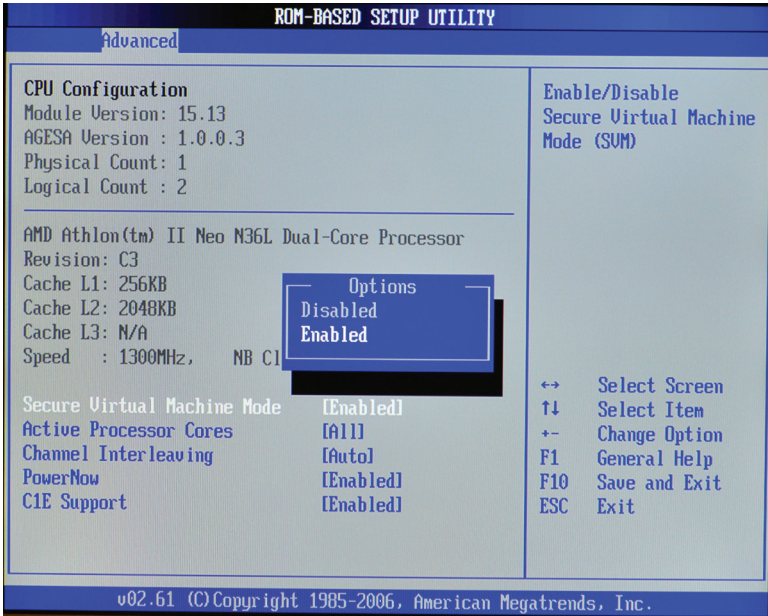


Рис. 1.3 ❖ Пример включения опции аппаратной поддержки виртуализации в BIOS (процессор AMD)

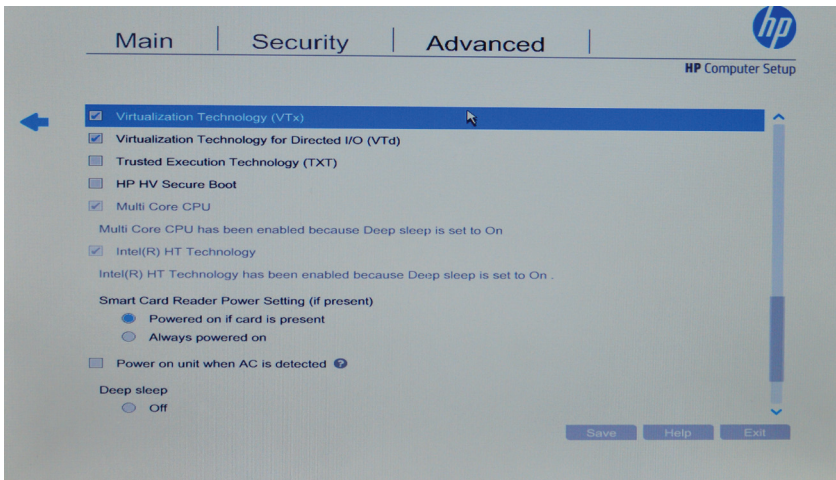


Рис. 1.4 ❖ Пример включения опции аппаратной поддержки виртуализации в BIOS ноутбука HP (процессор Intel)

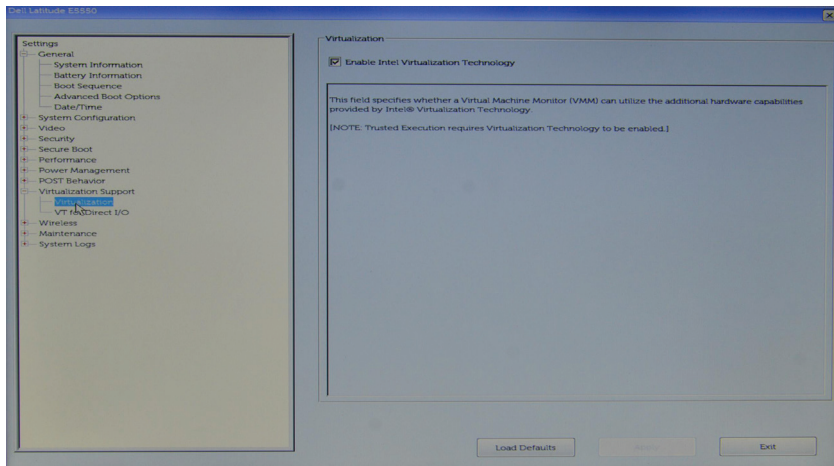


Рис. 1.5 ❖ Пример включения опции аппаратной поддержки виртуализации в BIOS ноутбука Dell (процессор Intel)

Проверить, что поддержка включена и процессор поддерживает одну из технологий, можно командой

```
$ grep -E 'svm | vmx' /proc/cpuinfo
```

Вы должны среди поддерживаемых процессором флагов увидеть `svm` или `vmx`. Также, если вы дадите команду:

```
$ lsmod | grep kvm
kvm_intel      143187  3
kvm            455843  1 kvm_intel
```

вы должны увидеть два загруженных в оперативную память модуля ядра. `kvm` – это не зависимый от производителя процессора модуль, а второй, `kvm_intel` или `kvm_amd`, реализует функционал VT-x или AMD-V соответственно.

Дистрибутивы OpenStack

Как уже отмечалось выше, OpenStack – это проект по созданию облачной инфраструктуры, но не продукт. Однако множество компаний, участвующих в создании OpenStack, создают на основе его кода свои продукты или дистрибутивы, зачастую используя свои проприетарные компоненты. Тут ситуация примерно аналогична созданию дистрибутивов GNU/Linux. Исходные коды в виде пакетов `tar.gz` до-

ступны по адресу <http://tarballs.openstack.org/>. На настоящий момент это более 350 отдельных проектов.

Автор попытался дать очень краткий обзор дистрибутивов OpenStack. Обзор не претендует на всеобъемлющий охват. Также информация, приведенная ниже, актуальна на момент написания книги. Достаточно полный список основных дистрибутивов приведен в разделе Marketplace официального сайта OpenStack – <https://www.openstack.org/marketplace/distros/>.

Книга описывает использование дистрибутива **RDO** (<https://www.rdoproject.org/>), с него и начнем наш краткий обзор. RDO – спонсируемый Red Hat проект по созданию открытого дистрибутива OpenStack. В противоположность коммерческому дистрибутиву Red Hat Enterprise OpenStack Platform (RHOSP), для RDO нельзя купить поддержку. Взаимосвязь между RHOSP и RDO примерно такая же, как между Red Hat Enterprise Linux (RHEL) и Fedora. RDO призван создать сообщество вокруг разработок Red Hat. В качестве установщика изначально предлагалось использовать скрипт packstack для тестов и Foreman для промышленного развертывания. В последних версиях вместо Foreman предлагается RDO Manager, основанный на проектах OpenStack Ironic и OpenStack TripleO. RDO можно развернуть на RHEL и его производных (CentOS, Oracle Linux и других).

Второй по популярности коммерческий вендор GNU/Linux также имеет свой собственный дистрибутив OpenStack под названием **SUSE OpenStack Cloud**. В качестве дистрибутива операционной системы используется на выбор SUSE Linux Enterprise Server 11 или SUSE Linux Enterprise Server 12. В качестве системы хранения данных поддерживается SUSE Enterprise Storage – собственный вариант сборки программно-определяемого хранилища данных Ceph. В качестве инструмента установки используются проект Crowbar (<http://crowbar.github.io>) и Chef – один из лидирующих инструментов управления конфигурациями в мире OpenSource.

Следующий дистрибутив – **Mirantis OpenStack (MOS)**. Как и RDO, в нем отсутствуют проприетарные компоненты, а отличительной особенностью Mirantis позиционирует систему установки Fuel, которая значительно упрощает масштабные развертывания. Также нужно отметить поддержку OpenStack Community Application Catalog, основанного на каталоге приложений Murano. В качестве дистрибутива GNU/Linux MOS требует на выбор Ubuntu или CentOS. С целью упрощения развертывания демостендов или изучения OpenStack имеются скрипты для быстрого развертывания на VirtualBox.

Сама компания Mirantis имеет в России и Украине офисы разработки, а также регулярно проводит мероприятия, посвященные OpenStack. Автор считает, что, как и RDO, дистрибутив от Mirantis может стать хорошей отправной точкой для знакомства с OpenStack.

VMware Integrated OpenStack (VIO) стоит несколько особняком от остальных рассмотренных в этом разделе, поскольку для своей работы требует развернутой проприетарной инфраструктуры VMware, включая гипервизор ESXi, платформу виртуализации сети NSX и систему управления vCenter. Все это делает VIO с учетом соответствующих лицензий относительно дорогим решением. Все компоненты OpenStack развертываются внутри виртуальных машин на ESXi в заранее созданном кластере VMware. Поскольку в основе лежит инфраструктура VMware vSphere, то отличительным преимуществом дистрибутива будет являться «родной» функционал VMware как корпоративной системы виртуализации: HA, DRS, vMotion и т. д. Это несколько переориентирует OpenStack от облачной к традиционной нагрузке. Также VIO снижает порог вхождения для существующих инфраструктур и обученных администраторов VMware. С другой стороны, помимо необходимости приобретать лицензии на проприетарное программное обеспечение, VIO привязывает пользователя этого дистрибутива к вендору решения.

Oracle OpenStack for Oracle Linux выделяется заметно недорогой, по сравнению с конкурентами, технической поддержкой в случае коммерческого использования. Он бесплатен при наличии премиальной поддержки. Из отличительных особенностей можно отметить поддержку Oracle ZFS. Также в качестве виртуальных машин поддерживается Solaris x86. Как и в случае других производителей аппаратного обеспечения, например IBM и HP, Oracle поддерживает коммерческое использование своей сборки только на своем «железе». Для инсталляции используются проект OpenStack Kolla и контейнеры Docker.

Ericsson Cloud Execution Environment – дистрибутив, созданный с учетом требований приложений виртуализации сетевых функций (NFV) и специфики операторов связи. В первую очередь это означает повышенную производительность сетевой подсистемы и ориентацию на приложения с требованием операций реального времени. По сравнению с дистрибутивами традиционных ИТ-компаний, ориентированный на операторов дистрибутив Ericsson Cloud Execution Environment отличается гарантированным компанией Ericsson SLA. В качестве примера функционала можно привести поддержку VLAN

Trunking, ускоренный при помощи библиотеки Intel DPDK виртуальный коммутатор (Ericsson Virtual Switch), мониторинг, высокую доступность виртуальных машин и многое другое. Также из отличий можно назвать свой собственный веб-интерфейс на базе Horizon. Дистрибутив создан на базе Mirantis OpenStack и в качестве опции в дальнейшем будет предлагать поддержку дистрибутива Red Hat.

Hewlett Packard Enterprise (HPE) Helion OpenStack – сборка от компании HPE, которая в последнее время активно вкладывает ресурсы в развитие проекта. На момент написания книги компания являлась самым большим работодателем для специалистов в OpenStack с сотнями вакансий по всему миру. Кроме того, компания является вендором номер один по вкладу в разработку OpenStack. Дистрибутив HPE Helion OpenStack построен на основе Ubuntu Linux. В качестве системы развертывания используется проект TripleO – OpenStack on OpenStack (<https://wiki.openstack.org/wiki/TripleO>). TripleO позволяет установить облако OpenStack при помощи самого OpenStack. Для этого создается специальное управляющее облако в виртуальной машине, которое в дальнейшем используется только для администрирования основного облака. Проект TripleO входит в число базовых проектов OpenStack. С этой точки зрения знакомство с развертыванием HPE Helion OpenStack было бы также интересным.

Говоря о дистрибутивах OpenStack, также необходимо упомянуть **проект OPNFV** (<https://www.opnfv.org>). OPNFV – это проект по построению открытой стандартной платформы для виртуализации сетевых функций (NFV). OPNFV объединяет ряд проектов, включая OpenStack, OpenDaylight, Ceph Storage, KVM, Open vSwitch и GNU/Linux. В проекте принимают участие крупнейшие телекоммуникационные компании и вендоры (AT&T, Cisco, EMC, Ericsson, HP, Huawei, IBM, Intel, NEC, Nokia, Vodafone, ZTE и многие другие). В настоящий момент проект предоставляет несколько сборок в виде ISO с различными установщиками.

Глава 2

Настройка лабораторного окружения OpenStack

На момент написания третьего издания книги последней версией OpenStack была четырнадцатая с момента появления проекта – Newton. В предыдущих двух изданиях рассматривались версии Juno и Liberty. Поскольку эта книга является введением в OpenStack, различия между релизами с точки зрения изучения OpenStack не очень велики. В книге по шагам, без использования средств автоматизации рассмотрены установка и настройка базового лабораторного окружения с использованием CentOS 7 и дистрибутива RDO.

В отличие от первого издания книги, где автор рассматривал установку большинства сервисов на одну виртуальную машину, мы будем разделять управляющий, сетевой и вычислительный узлы, что позволит нагляднее познакомиться с типичными ролями серверов при развёртывании OpenStack.

С первой по пятую главу вам понадобится только машина controller минимум с двумя виртуальными процессорами и не менее чем с 3 Гб оперативной памяти. В шестой главе вам дополнительно понадобятся три виртуальные машины sw1, sw2 и sw3 для объектного хранилища Swift, каждая с 1 Гб оперативной памяти. Если ресурсы вашего стенда ограничены, то в последующих главах эти виртуальные машины можно не использовать. В седьмой главе вам необходимо будет добавить виртуальную машину compute в качестве гипервизора. Чем больше оперативной памяти вы выделите для compute, тем лучше. Для большинства примеров в книге с запуском виртуальных машин Cirros Linux вам хватит 2 Гб. Если у вас достаточно ресурсов, то вы также можете создать второй опциональный вычислительный узел compute-opt. Он