

Содержание

Предисловие	6
Глава 1. Введение в OpenStack	12
Что такое облачная инфраструктура?	12
Что такое облачные приложения?	14
История OpenStack	15
Архитектура OpenStack	17
Гипервизор KVM и эмулятор QEMU	20
Дистрибутивы и сборки OpenStack.....	22
Глава 2. Настройка лабораторного окружения OpenStack	26
Подготовка CentOS 7 к использованию дистрибутива OpenStack RDO	31
Отличия RDO от «Upstream»	34
Как установить OpenStack RDO одной командой?.....	35
Как установить OpenStack одной командой из исходных кодов?	37
Как определить, какую версию OpenStack я использую?	38
Установка и настройка брокера сообщений.....	39
Переход на использование утилиты OpenStackClient.....	43
Глава 3. Сервис идентификации Keystone	45
Терминология Keystone	45
Установка и настройка Keystone.....	46
Настройка Keystone в OpenStack Mitaka при помощи Apache.....	49
Работа с пользователями, ролями и проектами в Keystone	51
Глава 4. Сервис хранения образов Glance	58
Установка и настройка сервиса Glance.....	59
Подготовка образов виртуальных машин	64
Работаем с образами виртуальных машин.....	68
Глава 5. Сервис блочного хранилища Cinder	72
Архитектура Cinder	72
Настройка сервисов Cinder	74
Глава 6. Объектное хранилище Swift	83
Архитектура Swift	84
Подготовка дополнительных серверов лабораторного окружения.....	86
Установка сервиса Swift-proxy	88
Установка узлов хранения Swift	88

Создание сервисных колец Swift.....	92
Завершение настройки.....	94
Работа с сервисом Swift	95
Настройка Swift в качестве хранилища для Glance.....	97
Рекомендации по поиску неисправностей в сервисах Swift.....	99
Глава 7. Контроллер и вычислительный узел Nova.....	100
Архитектура Nova.....	100
Установка контроллера Nova.....	101
Установка вычислительных узлов Nova.....	104
Глава 8. Службы сети Neutron	107
Архитектура Neutron	107
Установка узла управления Neutron	110
Установка сетевого узла Neutron.....	117
Установка вычислительного узла Neutron.....	120
Глава 9. Работа с виртуальными машинами из командной строки.....	123
Сеть в OpenStack.....	123
Запускаем экземпляр виртуальной машины	127
Зоны доступности и агрегирование вычислительных узлов.....	133
Добавляем к экземпляру виртуальной машины сеть.....	136
Моментальные снимки и резервные копии.....	139
Зашифрованные тома.....	144
Квоты на ресурсы.....	145
Живая миграция виртуальных машин.....	147
Глава 10. За фасадом Neutron.....	152
Виртуальный коммутатор Open vSwitch.....	152
Группы безопасности	158
Утилита для визуализации сети plotnetcfg.....	160
Зеркалирование трафика на Open vSwitch для мониторинга сети в OpenStack.....	160
Балансировщик нагрузки как сервис (LBaaS)	164
Глава 11. Веб-панель управления Horizon и работа пользователя из графического интерфейса	167
Установка веб-интерфейса.....	167
Работа с OpenStack в интерфейсе Horizon.....	169
Глава 12. Сервис сбора телеметрии Ceilometer	177
Установка служб управляющего узла.....	179

Установка служб вычислительного узла для отправки сообщений телеметрии.....	182
Работа со службой телеметрии	184
Глава 13. Сервис оркестрации Heat	190
Архитектура сервиса.....	191
Установка сервисов Heat	191
Запуск простого стека	194
Запуск стека, состоящего из нескольких виртуальных машин	198
Глава 14. Контейнеры Docker и OpenStack	201
Краткое знакомство с Docker.....	201
Совместное использование Docker и OpenStack.....	202
Настройка работы драйвера Docker для OpenStack Nova.....	204
Глава 15. Программно-определяемая система хранения данных Ceph.....	210
Архитектура Ceph.....	212
Быстрая установка кластера Ceph при помощи ceph-deploy.....	216
Установка кластера ceph вручную.....	220
Интеграция Ceph с сервисами OpenStack.....	225
Глава 16. Отказоустойчивость и производительность OpenStack	233
Обзор способов обеспечения высокой доступности сервисов облака	233
Выделение вычислительных ресурсов	235
Выделение оперативной памяти.....	236
Производительность сети	238
Определение аппаратных требований к оборудованию.....	239
Заключение	240
Приложение 1. Пример правил брандмауэра, реализующих группы безопасности на вычислительном узле	241
Приложение 2. Листинги шаблонов Heat.....	243
Запуск одной виртуальной машины – test-server.yml	243
Развертывание WordPress на двух виртуальных машинах	244
Приложение 3. Список используемых службами OpenStack сетевых портов	247

Предисловие

Благодарности

В первую очередь я хочу поблагодарить мою жену Лену за терпение и поддержку, проявленные во время написания этой книги и доработки текста для второго издания. Также хотелось бы поблагодарить рецензентов этой книги – Антона Арапова и Артемия Кропачева, благодаря ценным замечаниям которых данная книга стала значительно лучше и избавилась от многих неточностей и опечаток.

Автор приносит благодарность читателям первого издания, приславшим свои замечания и исправления: Александру Румянцеву и Владимиру Манькову.

Об авторе

Андрей Маркелов имеет более чем десятилетний опыт преподавания как авторских курсов, так и авторизованных курсов по ИТ-технологиям таких компаний, как Red Hat и Microsoft.

В настоящее время автор работает в качестве старшего менеджера по архитектуре компании Ericsson, специализируясь на облачных технологиях. До этого работал в качестве старшего системного архитектора в компании Red Hat, а также в крупных системных интеграторах России, получив более чем десятилетний опыт продаж, проектирования и внедрения сетевых и инфраструктурных решений.

Около шестидесяти публикаций в отечественных ИТ-журналах («Системный администратор», «Linux Format», «PC Week» и др.).

Андрей является сертифицированным архитектором Red Hat (RHCA) с 2009 года и имеет сертификаты Red Hat в таких технологиях, как OpenStack, OpenShift, RHEV, Cloud Storage, настройка производительности, безопасности Linux-систем и др.

Кроме того, автор имеет сертификаты Microsoft Certified System Engineer, Sun Certified System Administrator, Novell Certified Linux Professional, Mirantis Certified OpenStack Administrator и VMware Certified Professional.

Блог автора располагается по адресу: <http://markelov.blogspot.ru/>. Его twitter-аккаунт: @amarkelov.

Рецензенты

Антон Арапов – руководит командой, ответственной за развитие инфраструктурных проектов в компании Acision. Помогает клиентам полностью раскрыть потенциал мобильных каналов передачи данных, а также является ответственным за эволюционное развитие сервисов в сетях LTE 4G и проекты, способствующие ускорению возврата инвестиций. Сегодня технологии виртуализации являются основным средством для достижения поставленных целей. До Acision Антон руководил группой разработки виртуализации в ядре Linux в компании Red Hat. Антон является экспертом в технологиях виртуализации и их применении.

Артемий Кропачев – старший системный архитектор в группе консалтинга по инфраструктуре центров обработки данных в компании ICL Services. Артемий обладает сертификационными статусами RHCA, RHCSS, Oracle Certified Professional in Oracle Solaris. Основная область профессиональной деятельности – проектирование и внедрение инфраструктуры облачных решений, систем виртуализации, распределенных хранилищ данных и других сервисов на базе Linux/Solaris/IBM AIX.

Предполагаемая аудитория

Данная книга рассчитана на ИТ-специалистов (системных и сетевых администраторов, а также администраторов систем хранения данных), желающих познакомиться с де-факто стандартом в области открытых продуктов построения облачной инфраструктуры типа IaaS – OpenStack.

По аналогии с минимальными требованиями программного обеспечения к среде выполнения эта книга также предъявляет минимальные требования к читателю.

В первую очередь это желание разобраться с описываемым продуктом. Садясь за рукопись, я старался сделать книгу как можно более ориентированной на практическую работу. Лучше всего усваивается материал, который вы отработали собственными руками в лабораторном окружении. Единственной чисто теоретической главой в книге является первая, посвященная архитектуре продукта. Все остальные главы включают в себя обязательные практические упражнения.

Отсюда следует, что, прежде чем приступать к изучению продукта, написанного в первую очередь под Linux и для Linux, вам необходимы навыки работы с операционной системой. Умение работать в командной строке и знание базовых команд обязательны.

Также важными будут умение самостоятельно решать задачи и навык решения проблем – один из основных навыков хорошего ИТ-специалиста – практика. Автор сознательно старался сделать изложение большей части материала никак не привязанной к конкретным версиям операционной системы и дистрибутива OpenStack. Но почти наверняка к тому моменту, как книга доберется до вас, выйдет следующая версия OpenStack, и вы, естественно, захотите воспроизвести упражнения на актуальной версии. Также небольшие детали могут отличаться в зависимости от вашего любимого дистрибутива GNU/Linux, версии и дистрибутива OpenStack. В первую очередь это расположение конфигурационных файлов, особенности организации репозитория с пакетами, названия пакетов и т. п.

Ну и, естественно, от ошибок и опечаток никто не застрахован, включая вашего покорного слугу. Как связаться с автором и прислать информацию об ошибках, исправлениях или задать вопрос, указано в заключении, на последних страницах книги.

О чем эта книга

Книга состоит из шестнадцати глав и знакомит читателя с основными сервисами облачной операционной системы OpenStack. Кроме того, рассмотрены вопросы интеграции OpenStack и системы управления контейнерами Docker, программно-определяемой системы хранения данных Ceph, настройки производительности и высокой доступности сервисов.

Глава 1. Введение в OpenStack. Первая глава вводит читателя в предметную область облачных вычислений. Дается представление об облачных, горизонтально масштабируемых приложениях, их отличиях от вертикально масштабируемых, традиционных приложений. Рассказывается об истории создания OpenStack, его основных компонентах и основных дистрибутивах, присутствующих в настоящий момент на рынке.

Глава 2. Настройка лабораторного окружения OpenStack. В данной главе рассказывается о подготовке виртуальных машин тестового окружения к установке компонентов OpenStack. Рассматриваются рекомендуемые настройки сети дистрибутива GNU/Linux CentOS 7,

подключение репозитория OpenStack RDO, установка и настройка требуемых для OpenStack компонентов.

Глава 3. Сервис идентификации Keystone. В этой главе обсуждаются концепции и терминология сервиса идентификации OpenStack, а также читатель знакомится с Keystone на практике. Читатель установит и настроит ключевой сервис облачной операционной системы и заложит фундамент для настройки остальных служб.

Глава 4. Сервис хранения образов Glance. Четвертая глава посвящена настройке каталога образов виртуальных машин. Читатель познакомится с концепциями и настройкой сервиса Glance. К концу отработки материала главы в лабораторном окружении должен появиться каталог образов с загруженным шаблоном для создания виртуальных машин.

Глава 5. Сервис блочного хранилища Cinder. В данной главе читатель познакомится с тем, как создать блочное хранилище, которое будут использовать запущенные экземпляры виртуальных машин. Настройка этого опционального сервиса позволит виртуальным машинам сохранять данные между перезагрузками.

Глава 6. Объектное хранилище Swift. В главе рассматриваются настройка и работа с одним из двух исторически первых сервисов OpenStack – объектным хранилищем Swift. Рассматриваются концепции и архитектура сервиса, а также его установка и настройка.

Глава 7. Контроллер и вычислительный узел Nova. В данной главе рассматриваются установка одного из самых важных сервисов OpenStack, непосредственно занимающимся управлением виртуальными машинами. В ходе практических упражнений мы добавим два вычислительных узла в наше лабораторное окружение.

Глава 8. Службы сети Neutron. В главе, посвященной сетевым службам, рассматриваются концепции программно-определяемой сети. Читатель познакомится с вариантами организации сетевого взаимодействия в облаке и настроит на сетевом и вычислительном узле сервис Neutron.

Глава 9. Работа с виртуальными машинами из командной строки. К девятой главе читатель построит свое минимальное лабораторное окружение OpenStack. В этой главе мы рассмотрим, как на практике работать с сетями и виртуальными машинами.

Глава 10. За фасадом Neutron. В этой главе кратко рассматриваются «внутренности» сетевого сервиса OpenStack, а также такие темы, как группы безопасности и виртуальный коммутатор Open vSwitch.

Глава 11. Веб-панель управления Horizon и работа пользователя из графического интерфейса. Одиннадцатая глава, вероятно, самая дру-

жественная к читателю. В ней мы, наконец, встанем на место пользователя облачного сервиса и познакомимся с тем, как в графическом интерфейсе веб-консоли создать проект и запустить виртуальную машину.

Глава 12. Сервис мониторинга Ceilometer. Сервис Ceilometer представляет собой централизованный источник информации по метрикам облака и данным мониторинга. Этот компонент обеспечивает возможность биллинга для OpenStack. В данной главе читатель познакомится с настройкой Ceilometer и тем, как снимать и использовать данные телеметрии OpenStack.

Глава 13. Сервис оркестрации Heat. В этой главе рассказывается о сервисе Heat – «кольце всевластия», призванном связать все компоненты облака OpenStack воедино. Читатель познакомится с шаблоном формата HOT, при помощи которого Heat может создавать большинство типов ресурсов (виртуальные машины, тома, плавающие IP, пользователи, группы безопасности и т. д.) как единое целое, обеспечивая управление жизненным циклом приложения в облачной инфраструктуре.

Глава 14. Контейнеры Docker и OpenStack. В четырнадцатой главе мы кратко познакомимся с технологией контейнеров Docker и тем, как она используется совместно с OpenStack.

Глава 15. Программно-определяемая система хранения данных Ceph. Рассматриваются установка и интеграция программной СХД Ceph, которая становится для OpenStack «выбором по умолчанию» во многих новых инсталляциях.

Глава 16. Отказоустойчивость и производительность OpenStack. В этой главе автор дает обзорный материал о том, какими методами обеспечивается высокая доступность сервисов OpenStack, и рассматривает вопросы настройки производительности.

Что нового во втором издании?

Текст книги обновлен, чтобы соответствовать актуальным версиям рассматриваемых проектов и компонент OpenStack по состоянию на начало 2016 года (версия Liberty), а объем книги увеличен более чем на треть. Некоторые изменения второго издания:

- исправлен ряд неточностей и опечаток первого издания, вышедшего в июне 2015 года;
- добавлена глава, посвященная программно-определяемой системе хранения данных Ceph и использованию Ceph совместно с OpenStack;

- расширен и переработан материал по работе с виртуальными машинами и сетью (агрегация узлов, зоны доступности, живая миграция, создание образов виртуальных машин, работа с сетью и многое другое). С целью лучшей структуризации материал разбит на две отдельные главы;
- при описании настройки тестового окружения разделены управляющий, сетевой и вычислительный узлы, что позволяет нагляднее познакомиться с типичными ролями серверов при развертывании OpenStack;
- значительно переработан материал по работе сети в OpenStack;
- вопросы производительности и отказоустойчивости сервисов OpenStack выделены в отдельную главу;
- добавлено более пятнадцати новых иллюстраций и снимков с экрана.

Что вам необходимо помимо книги

Поскольку книга называется «Практическое введение», автор подразумевает, что читатель будет не просто читать книгу, а следуя за изложением материала, воспроизводить представленные примеры в лабораторном окружении. Помимо самой книги, вам понадобится персональный компьютер с минимальным объемом оперативной памяти 8 Гб (желательно 16 Гб) и системой виртуализации по вашему выбору. Автор во время написания книги и тестирования примеров использовал Ubuntu Linux с гипервизором KVM, но эта информация приведена лишь в качестве примера. Любая современная система виртуализации поддерживает GNU/Linux – фундамент облака OpenStack. Конечно, никто не мешает использовать и физическое «железо».

Вам также потребуется доступ в Интернет для обращения к репозиториям с обновлениями и пакетами OpenStack, а также скаченный ISO-образ дистрибутива CentOS 7.

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 широко доступного, 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, сетями, 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;

- 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;
- 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:

- 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 восемнадцати тысяч и корпоративных – более пяти 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 2,3 млн 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.

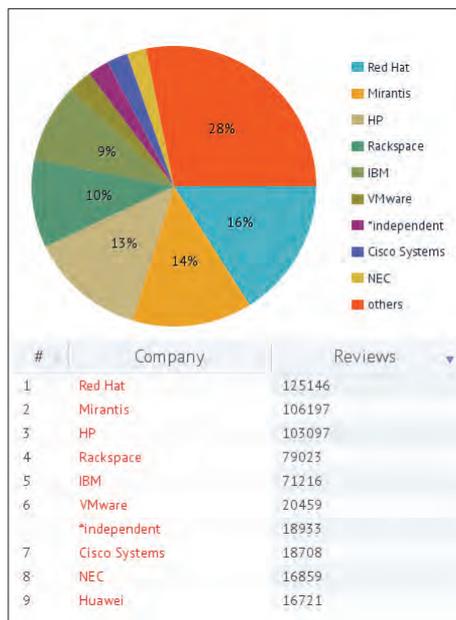


Рис. 1.1 ❖ Статистика в сервисе Stackalytics

В качестве примера на рисунке приведена диаграмма, показывающая вклад компаний в проект OpenStack за все время его существования (по состоянию на январь 2016 года). Как мы видим, в пятерку топ-контрибуторов входят Red Hat, HP, Mirantis, Rackspace и IBM.

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 октября 2015 года. Скорее всего, после выхода книги будет доступна информация по следующему

шему срезу статистики. Какую же интересную информацию оттуда можно получить?

- 60% принявших участие в опросе используют OpenStack для размещения производственной нагрузки;
- подавляющее большинство инсталляций – в IT-компаниях (64%), академической среде (10%) и телекоммуникационных компаниях (12%);
- Европейские страны занимают третье место в качестве страны проживания принявших участие в опросе (22% всех принявших участие). На первом месте – Северная Америка (44%), а затем – Азия (28%);
- по типам нагрузки на первом месте среды разработки и тестирования, затем инфраструктурные сервисы, веб-приложения, базы данных, хранение данных и «большие данные» с аналитикой;
- 95% всех внедрений (промышленных и тестовых) используют гипервизор KVM;
- также интересной является статистика по средам развертывания и управления облака: 53% – Puppet, 39% – Ansible и 19% – Chef;
- 38% в промышленной эксплуатации и 62% всего используют программную СХД Ceph;
- по драйверам сети суммарная статистика следующая: 62% – Open vSwitch, 28% – ML2, 21% – Linux Bridge. Самыми распространенными драйверами для проприетарных сетевых подсистем названы Juniper и VMware NSX с семью и пятью процентами;
- дистрибутивы ОС распределены следующим образом: 51% – Ubuntu, 28% – CentOS, 12% – 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. 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.

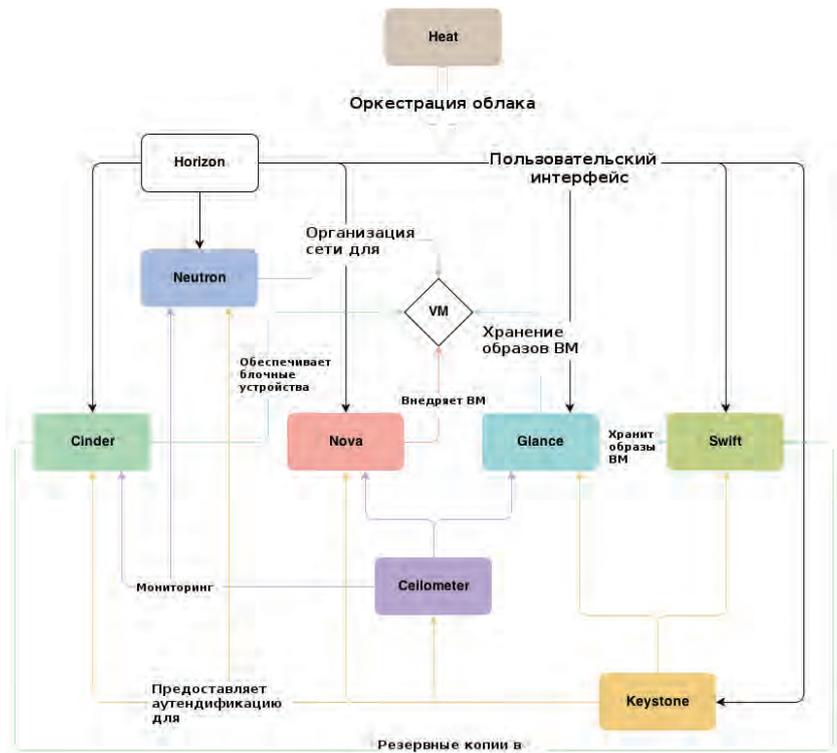


Рис. 1.2 ❖ Архитектура OpenStack

Одним из базовых сервисов является **OpenStack Compute (Nova)**. Этот сервис устанавливается на всех вычислительных узлах класте-

ра. 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 остановка.

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 и 1.4.

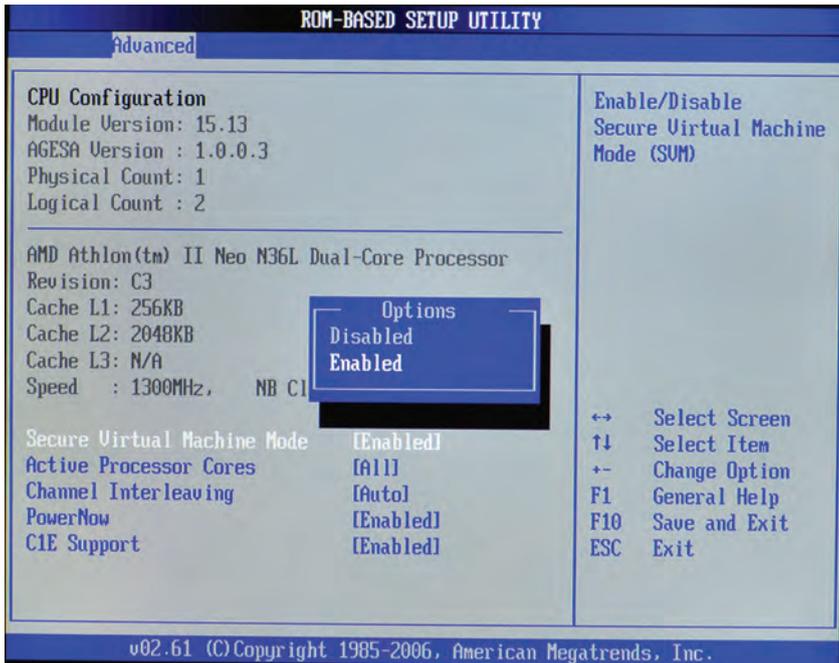


Рис. 1.3 ❖ Пример включения опции аппаратной поддержки виртуализации в BIOS (процессор AMD)

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 вендорнезависимый модуль, а второй – `kvm_intel` или `kvm_amd` – реализует функционал VT-x или AMD-V соответственно.

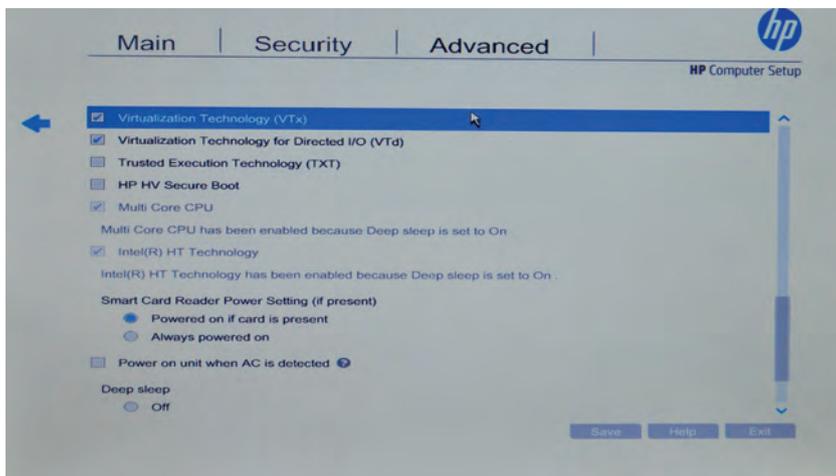


Рис. 1.4 ❖ Пример включения опции аппаратной поддержки виртуализации в BIOS (процессор Intel)

Дистрибутивы и сборки 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.

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

Nat Enterprise Linux OpenStack (RHEL OSP), для RDO нельзя купить поддержку. Взаимосвязь между RHEL OSP и RDO примерно такая же, как между 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 Forman 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 Cowbar (<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 за-

ранее 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. Компания Mirantis является партнером Ericsson.

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 устанавливать 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 две сборки на ISO: для установки при помощи Fuel и для установки при помощи RDO Manager.