



Содержание

Благодарности	19
Об авторах	21
О техническом рецензенте	23
Предисловие к русскому изданию	24
Введение	25

Глава 1

Общее описание платформы Windows Azure	26
Введение в облачные службы	27
Отраслевая терминология	28
Типы облаков	30
Определение терминов	32
Поставщики облачных служб	32
Переход к облачной парадигме	33
Экосистема облачных служб	35
Поставщики служб	35
Производители ПО	36
Независимые поставщики ПО	36
Посредники	37
Предприятия	37
Облачная стратегия Microsoft	37
Обзор платформы Windows Azure	39
Архитектура вычислений на платформе Windows Azure ...	42
Windows Azure	44
Служба вычислений	48
Роль Web	49
Роль Worker	50

Роль VM	52
Windows Azure Connect	53
Служба хранения Windows Azure	55
Управление	58
SQL Azure	60
Windows Azure AppFabric	64
Ценообразование на платформе Windows Azure	67
Портал управления – займемся подготовкой	72
Windows Azure Marketplace DataMarket	74
Типичные сценарии использования платформы Windows Azure	76
Базовые сценарии	76
Корпоративные сценарии	78
Сценарии для независимых поставщиков ПО	79
Резюме	81
Библиография	81

Глава 2

Служба вычислений Windows Azure	82
Служба вычислений	82
Сетевые компоненты	83
Blade-серверы	84
Гипервизор Windows Azure	84
Структурный контроллер	84
Операционные системы	84
Среды исполнения приложений	85
Модель службы	85
Роли приложений	85
Домены обновления и неисправности	87
Домены в контексте приложения	89
Безопасность службы вычислений	90
Разработка служб для платформы Windows Azure	91
Структура Windows Azure API	92
Среда разработки	92
Windows Azure Tools for Visual Studio	92
Типы проектов в Visual Studio	93
Настройка роли	96
Что позволяет проект, созданный Visual Studio	102
Отладка в Visual Studio	104

Сборка пакета службы	107
Инструментальные средства Windows Azure SDK.....	111
Модели служб	112
Файл ServiceDefinition.csdef.....	112
Управление сертификатами	122
Локальное квазиоблако	126
Эмулятор хранилища	128
Диагностика	129
Протоколирование	132
Настройка прослушивателя трассировки.....	132
Задание местоположения хранилища.....	133
Запуск службы диагностики.....	134
Разработка служб Windows Azure с межролевой коммуникацией.....	137
Цели	137
Добавление диагностики и межролевой коммуникации ...	137
Модель службы.....	138
Роль Worker	141
Роль Web	145
Запуск службы HelloAzureCloud	146
Публикация в облаке Windows Azure	148
Просмотр журналов IntelliTrace	154
Подключение к удаленному рабочему столу	156
Географическая распределенность	157
Разрешение территориальной группировки	159
Управление службами Windows Azure.....	159
Структура API управления службами	161
Программирование с использованием API управления службами	162
Цикл разработки службы Windows Azure.....	165
Рекомендации по архитектурному проектированию	167
Резюме.....	168
Библиография	169

Глава 3

Хранилища Windows Azure, часть I – BLOB-объекты и диски	170
Классификация хранилищ	171
Архитектура службы хранения	174

Хранилище BLOB-объектов	176
Ограничения хранилища BLOB-объектов	177
Архитектура хранилища BLOB-объектов	178
Учетная запись в службе хранения Windows Azure.....	178
Контейнеры.....	179
BLOB-объекты.....	180
Типы BLOB-объектов.....	181
Страничные BLOB-объекты	181
Блочные BLOB-объекты	181
REST API	183
Запрос	183
Глагол HTTP	183
URI запроса	183
Параметры URI	184
Заголовки запроса	184
Тело запроса	187
Ответ	187
Код состояния	187
Заголовки ответа.....	187
Тело ответа.....	187
StorageClient API.....	188
StorageClient API для хранилища BLOB-объектов	189
Операции с учетной записью	196
Пример программы	200
Операции с контейнерами	203
Операция Create Container	206
Операция Set Container Metadata	208
Операция List Blobs	211
Операции с BLOB-объектами.....	218
Операция Put Blob.....	219
Операции Put Block и Put Block List	225
Операция Get Blob.....	228
Операция Copy Blob	231
Content Delivery Network (CDN).....	234
Диски Windows Azure.....	237
Общие сведения	238
Операции с дисками	239
Локальное создание диска	241
Загрузка диска	244
Монтирование диска	245

Создание и монтирование диска из экземпляра роли	245
Создание мгновенного снимка диска	246
Сценарии использования дисков Windows Azure.....	247
Сценарии использования хранилища BLOB-объектов.....	249
Загрузка больших объемов данных	250
Хранилище как облачная служба	250
Интеграция с корпоративными доменными учетными записями	251
Классификация хранилищ	252
Шифрование и дешифрирование	253
Корпоративная синхронизация файлов	253
Резюме	255
Библиография	255

Глава 4

Хранилища Windows Azure, часть II – очереди	256
Ограничения очередей	257
Архитектура хранилища очередей	258
Учетная запись в службе хранения Windows Azure.....	259
Очереди	259
Сообщения	260
REST API	261
Запрос	261
Глагол HTTP	262
URI запроса	262
Параметры URI	262
Заголовки запроса	262
Тело запроса	264
Ответ	264
Код состояния	264
Заголовки ответа.....	264
Тело ответа.....	265
StorageClient API.....	266
StorageClient API для хранилища очередей	266
Операции с учетной записью	270
Операции с очередями	275
Операция Create Queue	276
Операция Set Queue Metadata	279

Операции с сообщениями	281
Операция Put Message	283
Операция Get Messages	286
Асинхронный API.....	291
Сценарии использования очередей.....	294
Сценарий 1: коммуникация между ролями Web и Worker	294
Сценарий 2: распределение нагрузки между ролями Worker.....	295
Сценарий 3: интероперабельная система обмена сообщениями.....	296
Сценарий 4: гарантированная обработка	297
Резюме.....	299
Библиография	299

Глава 5

Хранилища Windows Azure, часть III – таблицы	300
Архитектура хранилища таблиц.....	301
Учетная запись в службе хранения Windows Azure	301
Таблицы.....	301
Сущности	302
Свойства	302
REST API.....	307
Запрос.....	308
Ответ.....	311
Библиотека службы данных ADO.NET (и библиотека .NET Client).....	314
StorageClient API	315
Пример табличной модели	318
Операции с учетной записью.....	324
Операции с таблицами.....	324
Операция Create Table	325
Опрос таблицы	329
Операции с сущностями.....	333
Операция Query Entities	334
Операция Insert Entity	343
Операция Merge Entity	348
Средства аналитики хранилищ	352
Протоколирование	353
Метрики.....	355

Включение средств аналитики хранилищ	356
Хранилище таблиц и SQL Azure	357
Сценарии использования хранилища таблиц	358
Сценарий 1: чтение счетчиков производительности из хранилища таблиц	358
Сценарий 2: разбиение результирующего на страницы	362
Резюме	364
Библиография	365

Глава 6

Роль VM и Windows Azure Connect	366
Роль VM	366
Достоинства и недостатки роли VM	367
Сценарии	368
Создание образа виртуальной машины	369
Компоненты интеграции роли Windows Azure VM	370
Подготовка системы	370
Координация между образом и средой Windows Azure	371
Установка сертификатов	371
Создание локальных ресурсов хранения	371
Установка компонентов интеграции	372
Программа подготовки системы	373
Загрузка образа в Windows Azure	373
Просмотр образа на портале управления	374
Создание размещенной службы и определения службы	374
Создание пакета и разрешение удаленного доступа	376
Развертывание размещенной службы	377
Windows Azure Connect	378
Windows Azure Connect и шина служб	379
Подготовка Windows Azure Connect	380
Активированные конечные точки, группы и роли	381
Установка и активация конечной точки Azure на локальной машине	381
Протоколы и порты	383
Активация Windows Azure Connect для роли	384
Создание групп Connect	386

Резюме	387
Библиография	387

Глава 7

AppFabric: служба управления доступом	388
Что такое цифровое удостоверение?	388
Что такое утверждения?	390
Модель удостоверений на основе утверждений	393
Сценарии использования службы управления доступом ...	396
Сценарий 1: облачное корпоративное приложение	396
Сценарий 2: приложение, обслуживающее несколько предприятий	399
Сценарий 3: облачная служба независимого поставщика ПО	403
Получение маркеров от ACS	405
Портал управления службой ACS	405
Подготовка пространства имен службы ACS	406
Поставщики удостоверений	409
Active Directory Federation Services 2.0 (ADFS 2.0)	409
Глобальные поставщики удостоверений	412
Проверяющая сторона	417
Группы правил	419
Сертификаты и ключи	419
Подписание маркера	419
Создание собственного сертификата	421
Шифрование маркеров	422
Дешифрирование маркеров	423
Удостоверения служб	423
Администраторы портала	423
Служба управления	424
Интеграция с приложениями	425
Страницы входа	425
SDK и документация	428
Ссылки на конечные точки	428
Программирование приложений, интегрированных со службой ACS	429
Пассивная федерация с ACS	430
Веб-приложение: несколько поставщиков удостоверений, пользующихся ACS	430

Конфигурирование ACS с помощью портала администрирования	430
Добавление поставщиков удостоверений.....	431
Добавление проверяющего приложения	431
Создание групп правил и правил преобразования утверждений.....	431
Модификация проверяющего приложения для интеграции с ACS	432
Добавление поставщика, работающего по протоколу WS-Federation.....	434
Проектирование утверждений для проверяющей стороны	436
Проектирование правил ACS	436
Авторизация на основе утверждений.....	437
Резюме	441
Концепции и терминология.....	441
Поставщик удостоверений	442
Проверяющая сторона	442
Маркер безопасности (SAML-маркер).....	442
Служба маркеров безопасности (STS).....	442
Запрос на получение маркера безопасности (Request for Security Token – RST).....	443
Ответ на запрос на получение маркера безопасности (Request Security Token Response – RSTR)	443
Утверждение	443
Федерация удостоверений	444
Windows Identity Foundation (WIF)	444
Active Directory Federation Server (ADFS 2.0).....	445
Web Resource Authorization Protocol (WRAP) и Simple Web Token (SWT)	445
Библиография	446

Глава 8

AppFabric: шина служб	447
Немного предварительной информации.....	447
Сервисная шина предприятия (ESB).....	449
Безопасность и управление доступом.....	450
Инфраструктура связи	450
Корпоративная схема именования	450

Контракты интерфейсов.....	451
Эволюция сервисной Интернет-шины (ISB)	451
Служба ретрансляции	454
Введение в шину служб AppFabric.....	455
Безопасность.....	456
Интеграция с ACS (аутентификация ретрансляции)	457
Защита сообщений.....	460
Служба именования	461
Реестр службы	464
Среда обмена сообщениями.....	466
Привязки шины служб AppFabric.....	468
Буфер сообщений	470
Очереди и темы	471
Программирование с использованием шины служб AppFabric	471
Пример: служба ProAzure Energy	473
Привязка NetOnewayRelayBinding.....	476
Контракт AppFabric	477
Реализация службы	479
Привязка службы.....	480
Защита ретрансляции	482
Защита сообщений.....	484
Конечные точки службы	488
Размещение службы.....	489
Проектирование клиента	490
Запуск приложения	492
Привязка NetEventRelayBinding	492
Контракт AppFabric	496
Реализация службы	496
Привязка службы.....	497
Защита ретрансляции	497
Защита сообщений.....	498
Конечные точки службы	498
Размещение службы.....	498
Проектирование клиента	499
Запуск приложения	500
Привязка NetTcpRelayBinding	501
Контракт AppFabric	505
Реализация службы	507
Привязка службы.....	508
Защита ретрансляции	508

Защита сообщений.....	509
Конечные точки службы	510
Размещение службы.....	510
Проектирование клиента	511
Запуск приложения	512
HTTP-привязки ретрансляции	513
Привязка WS2007HttpRelayBinding.....	515
Привязка WebHttpRelayBinding	519
Буфер сообщений.....	526
Программирование работы с буфером сообщений в приложениях	528
Создание политики буфера сообщений	528
Создание и удаление буфера сообщений	529
Отправка сообщений в буфер	530
Получение сообщений из буфера	530
Пример приложения, работающего с буфером сообщений	531
Обмен сообщениями на шине служб AppFabric: очереди и темы.....	532
Очереди шины служб AppFabric	533
Сравнение очереди шины служб AppFabric и хранилища очередей Azure	535
Темы шины служб AppFabric.....	535
Правила подписки.....	537
SQLFilterExpression	537
CorrelationFilterExpression	537
Программирование очередей и тем шины служб	537
Клиентский .NET API.....	537
Пространство имен Microsoft.ServiceBus	537
Пространство имен Microsoft.ServiceBus.Messaging	539
Основные компоненты среды обмена сообщениями	540
Создание и отправка сообщений	541
Обработка проблематичных сообщений и отказ от обработки.....	544
REST API.....	545
Защита REST-запросов средствами службы ACS.....	545
Очереди: команды для работы с сообщениями.....	547
Очереди: команды управления	548
Темы и подписки: команды для работы с сообщениями.....	551

Темы: команды управления.....	554
Правила: команды управления	559
Резюме	561
Библиография	561

Глава 9

AppFabric: кэширование 562

Сравнение кэширования AppFabric с другими технологиями кэширования	564
Подготовка кэша AppFabric	564
Клиенты кэша AppFabric	564
Ссылки на сборки	565
Настройка клиента кэша	565
Настройка клиента кэша с помощью конфигурационного файла приложения.....	565
Настройка кэша из программы	568
Программирование кэша AppFabric	570
Поставщик состояний сеансов ASP.NET	571
Включение поддержки состояния сеансов в кэше AppFabric	572
Включение кэширования вывода ASP.NET в кэше AppFabric	573
Резюме	574
Библиография	574

Глава 10

SQL Azure 575

Обзор SQL Azure	575
Архитектура SQL Azure	577
Инфраструктурный уровень	578
Платформенный уровень	578
Уровень служб.....	578
Клиентский уровень	579
Поддерживаемые функции и ограничения SQL Azure	579
Средства базы данных	580
Средства прикладного уровня.....	581

Средства администрирования	581
Доступ к данным в SQL Azure	581
Ближнее подключение	582
Дальнее подключение	583
Начинаем работу с SQL Azure	583
Создание сервера SQL Azure	585
Создание базы данных SQL Azure	587
Подключение к базе данных SQL Azure	589
Подключение из SQL Server Management Studio.....	590
Подключение с помощью программы Database Manager	597
Подключение с помощью SQLCMD	600
Подключение с помощью ADO.NET	602
Разработка служб Windows Azure с использованием SQL Azure	606
Описание службы	606
Бизнес-процессы в системе управления спросом.....	607
Техническая архитектура	609
Проектирование базы данных о шлюзах и тарифах	609
Создание базы данных proazuredemres.....	611
Создание таблиц базы данных	612
Создание хранимых процедур	613
Загрузка тестовых данных	617
Оптимизация запросов SELECT	619
Синхронизация таблицы тарифов	623
Проектирование шлюза	623
Запуск службы управления спросом Dem-Res	625
Стратегии переноса базы данных	626
Перенос определения данных	627
Перенос данных	628
Перенос бизнес-логики	629
Перенос приложений	630
Стратегии управления ростом базы данных	631
Отчеты в SQL Azure	633
Пример отчета	633
Создание отчетов.....	634
Определение источника данных	634
Определение запроса	635
Развертывание отчетов	635



Сравнение с SSRS	636
Служба Data Sync.....	637
Дизайн Data Sync.....	638
Параметры синхронизации	638
Резюме.....	638
Библиография	639
Предметный указатель.....	640



Примечание к российскому изданию

Облачные технологии являются одной из наиболее активно развивающихся отраслей в современной ИТ-индустрии и информация по облакам является востребованной среди специалистов. Стремительное развитие технологий, однако, ставит очень непростую задачу перед авторами больших печатных изданий по этой теме: платформа Windows Azure развивается очень динамично и весьма вероятно, что с момента написания материала автором до того, как книга попадет к читателю, изменится очень многое. На мой взгляд Теджасви и Тони очень хорошо справились с этим – данная книга является прекрасным примером достаточно подробного описания технологических возможностей облачной платформы Windows Azure и спустя больше года с момента ее написания практически вся информация в ней не потеряла своей актуальности. Тем не менее за прошедшее время в Windows Azure появился целый ряд очень важных новых возможностей – таких как возможность использования IaaS, в том числе и запуск виртуальных машин под управлением ОС Linux, платформа для работы с медиаконтентом – Media Services, сервис по хостингу веб-сайтов – Web Sites, новый портал управления Windows Azure теперь использует технологию HTML5 и многое другое – новые возможности и обновления в Windows Azure появляются практически каждую неделю, облачная платформа является одним из наиболее динамично развивающихся продуктов Microsoft. Для читателей книги на русском языке я хотел бы дать ссылки на несколько важных ресурсов по Windows Azure:

- ❑ во-первых, это блог Windows Azure на русском языке http://blogs.msdn.com/b/windows_azure_in_russia/;
- ❑ сообщество Windows Azure разработчиков <https://www.facebook.com/groups/azurerus/>;
- ❑ твиттер-акаунт [@windowsazure_ru](https://twitter.com/windowsazure_ru) и контактный имейл для всех вопросов по облачной платформе azurerus@microsoft.com.



Введение

Облачные вычисления перестали быть рекламной шумихой. Технология вышла на такую стадию зрелости, когда различные компании начали переводить на эту платформу уже существующий бизнес и исследовать возможности для новых начинаний. Облако позволяет быстро претворить идею в жизнь, так как затраты на приобретение очень малы. Вы можете открыть учетную запись с помощью кредитной карты и уже через несколько минут приступить к использованию платформы в публичном облаке. Проработка архитектурных решений для Windows Azure – моя ежедневная работа, и в этой книге я хотел поделиться накопленным опытом. В книге изложена большая часть основополагающих концепций облачных вычислений на примере платформы Windows Azure, которая представляет собой полнофункциональное облако со средствами вычисления, хранения, управления и службами промежуточного слоя, позволяющими создавать распределенные приложения.

Активная работа над продуктом приведет к тому, что к моменту выхода книги из печати появится много новых служб. Но я обещаю рассказывать о них с помощью других каналов, в том числе в статьях в открытых источниках, в моей ленте в Twitter (@tejaswi_redkar), в «полевых заметках» (<http://www.microsoft.com/windowsazure/learn/real-worldguidance/>) и так далее. Так что ждите новостей.

Надеюсь, что, прочитав эту книгу, вы сможете самостоятельно создавать приложения для Windows Azure и начать пожинать плоды в виде новых возможностей, открываемых этой платформой. В каждой главе я описывал случаи из своей реальной практики. В книге достаточно исходного кода (за исключением главы 1), так что вам будет с чего начать. Весь исходный код можно загрузить с сайта издательства www.apress.com или со страницы книги на сайте CodePlex по адресу <http://azureplatformbook.codeplex.com>.

Искренне ваш,
Теджасви Редкар



Глава 1

Общее описание платформы Windows Azure

За последние два года идея облачных вычислений перевернула всю отрасль информационных технологий (ИТ). По масштабности ее воздействие можно сравнить с Интернетом и практикой частично-го перевода деятельности компаний в оффшорные зоны. В отчете Gartner Research говорится, что облачные вычисления – одна из «10 самых прорывных технологий за период с 2008 по 2012 год». При этом *прорывной* Gartner называет технологию, приведшую к кардинальному изменению общепринятого способа решения задач. С точки зрения разработчиков, архитекторов и эксплуатационников, облачные вычисления знаменуют совершенно новый подход к архитектуре, реализации, развертыванию и сопровождению программных служб.

Облачные вычисления демократизировали отрасль ИТ подобно тому, как Интернет демократизировал отрасли, производящие товары массового спроса. Интернет открыл потребителям доступ к безбрежному океану ресурсов – от бесплатного поиска по рекламным объявлениям до дистанционного банковского обслуживания. Нечто аналогичное облачные вычисления означают для предприятий – от малых до крупных. Чтобы оперативно реагировать на изменения, бизнес теперь может за умеренную плату развернуть свои приложения в принадлежащем сторонней компании центре обработки данных. При этом нет никаких затрат на оборудование, поскольку оно принадлежит поставщику облачных служб. Вроде бы похоже на привычное предоставление хостинга для размещения веб-сайтов, однако имеется принципиальное различие: полезная модель основана на прекрасно масштабируемых платформах ЦОД. Облачные вычисления – настолько эффективная технология, что даже корпорация Microsoft приступила к пересмотру своей бизнес-модели, намереваясь инвестировать средства во вновь открывающиеся возможности.

В этой главе я сначала расскажу об основных концепциях облачных служб, а затем перейду к обзору платформы Windows Azure от Microsoft. В предыдущем издании этой книги я знакомил читателей с моделями разработки, поддерживаемыми некоторыми присутствующими на рынке поставщиками облачных служб. Я остановился на таком подходе, потому что технология тогда была новой, и мне хотелось, чтобы читатели понимали различия между предлагаемыми решениями. В этом издании я тоже сравню поставщиков облачных служб, но не так подробно, как раньше. Сейчас достаточно литературы об этих платформах, поэтому в данной книге детали можно опустить.

Введение в облачные службы

В качестве вступления рассмотрим сценарий, типичный для современных средних и крупных предприятий. Предположим, какому-то структурному подразделению необходимо срочно развернуть узкоспециализированное интерактивное веб-приложение (микросайт), посвященное новому продукту, который планируется выпустить через пять месяцев. Приложение должно давать потребителю детальное представление о продукте, а также возможность разместить заказ прямо на сайте. У подразделения имеется бюджет, но нет ни времени, ни ресурсов для его освоения, а развернуть приложение необходимо в течение трех месяцев, чтобы к моменту выхода продукта на рынок все было готово.

Группа хостинга понимает требования, но для развертывания на ресурсах подразделения ИТ необходима координация между отделами, отвечающими за оборудование, программное обеспечение, эксплуатацию и поддержку. Не исключено, что только на заказ оборудования и подготовку операционной системы уйдет два месяца. Затем подразделение ИТ должно будет провести стандартную процедуру тестирования и приемки, чтобы убедиться в том, что идентифицированы все потребности служб эксплуатации и поддержки. Так что ввести систему в эксплуатацию получится никак не раньше, чем через шесть месяцев.

Владелец предприятия повышает приоритет задачи, но не может выйти за рамки принятых технологических процессов. В итоге проект передается внешнему поставщику, который сдает приложение через три месяца. Но хотя приложение внедрено, ему недостает качества эксплуатации и поддержки, отвечающего стандартам

предприятия. Так быть не должно – единственным ответом на все потребности бизнеса должно быть собственное ИТ-подразделение компании. Пусть даже аутсорсинг дает высокий доход на капиталовложения, в долгосрочной перспективе компания несет значительные потери из-за отсутствия инноваций. У ИТ-подразделения есть потенциал для внедрения инноваций, и ему надо предоставить такую возможность, а не заставлять преодолевать искусственные границы технологических процессов.

Я встречаюсь с такими случаями ежедневно и не вижу, как можно решить проблему без кардинальной перестройки процессов и структуры организации или перехода на технологии типа облачных вычислений.

Чем может помочь облако? Чтобы разобраться в этом, вспомним исходное бизнес-требование: развернуть приложение в течение трех месяцев. По существу, бизнес ждет от ИТ гибкости и оперативности, и если для разработки приложения нужен всего один месяц, то стоит ли тратить еще шесть на координацию и приобретение оборудования?

Облако предоставляет инфраструктуру для развертывания приложений тогда, когда это необходимо. Процедура предоставления оборудования, операционной системы и программного обеспечения полностью автоматизирована и управляется поставщиком облачных служб.

Отраслевая терминология

Для стандартизации терминологии, относящейся к облачным вычислениям, в отрасли определены следующие три основные категории облачных служб: «инфраструктура как услуга» (IaaS), «платформа как услуга» (PaaS) и «программное обеспечение как услуга» (SaaS).

IaaS – это служба, предоставляющая оборудование и виртуализированные операционные системы, работающие в массивно масштабируемых центрах обработки данных (ЦОД), принадлежащих поставщику облачных служб. Эту инфраструктуру можно взять в аренду, развернуть в ней свое программное обеспечение и управлять жизненным циклом не только собственных программ, но и операционной системы, в которой они работают. В IaaS вы сами несете ответственность за модернизацию, применение исправлений и обслуживание операционной системы и программ, работающих на

арендованном оборудовании. Поэтому потенциальными потребителями IaaS являются системные администраторы и инженеры-эксплуатационники. Короче говоря, IaaS абстрагирует оборудование и инфраструктуру виртуализации.

PaaS – это служба, предоставляющая оборудование, операционные системы и среду исполнения для запуска приложений в массивно масштабируемых ЦОД, принадлежащих поставщику облачных служб. PaaS осуществляет за вас обслуживание операционных систем и оборудования, но за собственные приложения и данные целиком отвечаете вы. Следовательно, типичными потребителями PaaS являются разработчики. Хотя окончательное развертывание и сопровождение будет осуществляться силами отдела эксплуатации, платформа позволяет разработчику принимать определенные решения о развертывании путем настройки параметров. Короче говоря, PaaS абстрагирует инфраструктуру и операционную систему.

SaaS – это служба предоставления программного обеспечения как услуги от начала до конца. На вашу долю остается только управление данными, которые хранятся и обрабатываются этой службой. Оборудованием, операционной системой и программным обеспечением управляет поставщик SaaS, действующий в ваших интересах. Следовательно, типичным потребителем SaaS является владелец предприятия, который может зайти на сайт SaaS, оформить заказ на услугу и начать ей пользоваться.

Таким образом, SaaS естественно строится на основе PaaS, а PaaS – на основе IaaS, и, значит, PaaS предоставляет все средства IaaS, на которой базируется. Предлагать ли IaaS как отдельную услугу – по большей части стратегическое решение. На рис. 1.1 представлены типичные административные границы между IaaS, PaaS и SaaS.

Способ управления удостоверениями пользователей зависит от конкретного сценария. Иногда предприятие раскрывает свою службу поставки удостоверений в виде федеративной службы, а иногда удостоверения, действующие в SaaS и на территории предприятия, различны. В последние два года появились еще некоторые термины, например: «данные как услуга» (Data as a Service – DaaS), «ИТ как услуга» (IT as a Service), «безопасность как услуга» (Security as a Service) и т. д. Но для простоты я разбил все службы на три категории: IaaS, PaaS и SaaS.

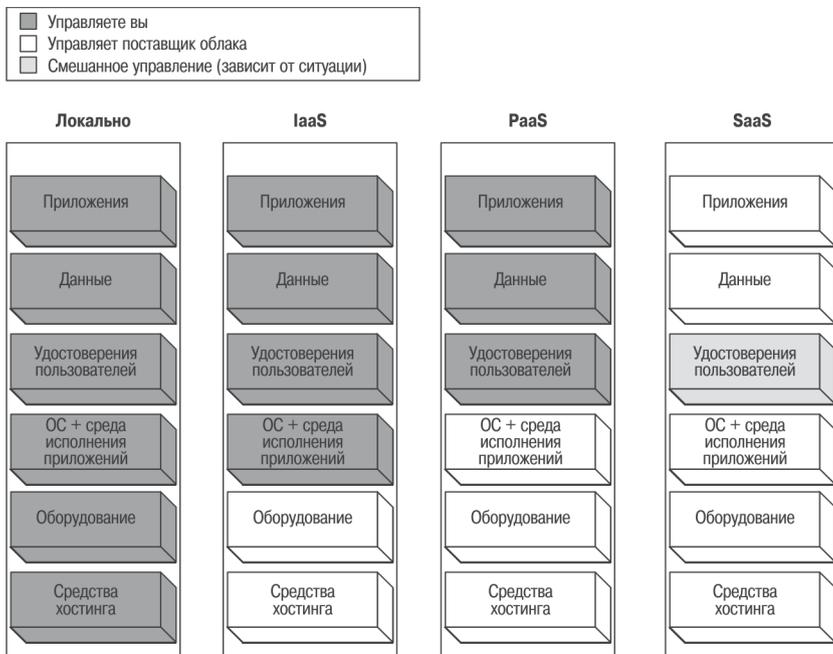


Рис. 1.1. Административные границы между IaaS, PaaS и SaaS

Типы облаков

Помимо типов облачных служб, в отрасли часто говорят о типах представленных на рынке облаков. Под облаком понимается архитектура ЦОД, который стоит за облачными службами. Тогда в чем же различие между поставщиком хостинга и поставщиком облачных служб? Хороший вопрос...

Основываясь на собственном опыте и знаниях, я бы говорил об облаке, только если архитектура ЦОД предоставляет следующие услуги:

- ❑ **Услуга «Оплата по факту»** – модель, при которой плата начисляется только за использованные ресурсы или по числу пользователей, обращающихся к службе. Цена снижается или возрастает в зависимости от объема потребленных ресурсов.
- ❑ **Портал самообслуживания** – облако предоставляет средства для получения и освобождения ресурсов вручную или программно.

- ❑ **Абстракция серверного оборудования** – облако снимает с вас заботу о получении и (или) обслуживании аппаратных ресурсов, необходимых для работы приложения.
- ❑ **Абстракция сетевого оборудования** – облако снимает с вас заботу о получении и (или) обслуживании сетевых ресурсов, требуемых приложению.
- ❑ **Динамическая масштабируемость** – облако предоставляет средства ручного и (или) программного динамического масштабирования в соответствии со спросом на ресурсы.
- ❑ **Соглашение об уровне обслуживания (СУО), обеспечивающем высокую доступность** – облако четко определяет СУО, гарантированно обеспечивающее доступность платформы.

Местоположение облака определяет его тип: частное или публичное. Чтобы не усложнять изложение, я определю только эти два типа облаков.

Публичное облако – это ЦОД, расположенный в открытом домене и доступный через Интернет. Публичное облако управляется поставщиком облачных служб. Некоторые публичные облачные платформы интегрируются с Интранет-службами компании за счет федерации и виртуальных частных сетей или аналогичных средств обеспечения связности. Основное приложение и относящиеся к нему данные тем не менее находятся в центре обработки данных, принадлежащем поставщику облачных служб.

Частное облако – это облачная инфраструктура, развернутая в вашем собственном ЦОД. Поскольку определение облака дает широкие возможности для интерпретации, каждая компания выработала свое определение частного облака. Я буду считать, что описанные выше средства составляют минимальные требования, позволяющие рассматривать облако как публичное или частное. Если какие-то из этих услуг не предоставляются частным облаком, значит, это просто оптимизированный ЦОД. И это вовсе не обязательно плохое решение – в некоторых случаях оптимизированный ЦОД лучше облака. Основное различие между частным и публичным облаком – объем капитальных затрат на создание и поддержание инфраструктуры. В случае публичного облака таких затрат попросту нет.

Примечание. В этой книге я в зависимости от контекста буду употреблять термины «облачные службы» и «облачные приложения» как синонимы. Вообще говоря, в некоторых случаях облачную службу можно

представлять себе как набор облачных приложений, но в данной книге мы будем понимать под «службой» и «приложением» одно и то же.

Определение терминов

Прежде чем приступить к углубленному изучению облачных служб, давайте познакомимся с терминологией, используемой в этой книге. Я специально отвел этот раздел для определения терминов (см. табл. 1.1), чтобы впоследствии не было разночтений.

Таблица 1.1. Терминология, применяемая в книге

Термин	Определение
Azure или Windows Azure	Платформа Microsoft Windows Azure
Облачное приложение	Приложение, развернутое на платформе облачных служб и обычно являющееся частью более крупной облачной службы
Облачная платформа	Платформа PaaS, предлагаемая поставщиком для развертывания облачных служб (например, платформа Windows Azure, предлагаемая Microsoft)
Локальное (on-premise)	Относится к приложениям или службам, развернутым в собственных ЦОД предприятия и управляемых им же
Внешнее (off-premise)	Относится к приложениям или службам, развернутым в облаке
Решение	Если слово используется изолированно, то означает набор приложений и (или) облачных служб, предназначенных для решения конкретной задачи бизнеса (например, решение по расчету зарплаты состоит из трех облачных служб и четырех локальных приложений)

Поставщики облачных служб

В последние два года несколько крупных компаний по разработке ПО и Интернет-платформ начали предлагать облачные службы. Для таких компаний, как Amazon, Google и Microsoft, которые уже давно присутствуют в Интернете, это стало естественным развитием. А VMware строит соответствующие системы благодаря приобретению компаний Springsource и Zimbra. Предложения на рынке облачных служб фрагментированы, и иногда бывает трудно разобраться даже в том, что предлагает один-единственный поставщик. В табл. 1.2 перечислены несколько поставщиков, имеющих достаточно проработанные предложения облачных служб. Эту матрицу возможностей можно применить к любым поставщикам облачных служб – ныне существующим и будущим.

Таблица 1.2. Возможности облачных служб

Возможность	IaaS	PaaS	SaaS
Публичное	Amazon EC2 Windows Rackspace.com	платформа Windows Azure AppFabric	Office 365 Salesforce.com Google Apps
Частное	VMWare vSphere Hyper-V	Windows Azure Appliance (пока не предлагается)	SharePoint как ИТ-услуга

Таблица 1.2 позволяет найти поставщика облачных служб, отвечающего вашим требованиям. Как правило, не удастся найти одного поставщика, который удовлетворяет всем пожеланиям, даже если речь идет о локальном облаке, развернутом на территории предприятия.

Примечание. С июня 2012 года Windows Azure также предоставляет функционал IaaS, в том числе использование виртуальных машин на базе ОС Windows Server или Linux.

Переход к облачной парадигме

В предыдущем разделе мы видели, что при оценке имеющихся предложений вы можете оказаться перед непростым выбором и, скорее всего, решите протестировать по меньшей мере две облачные службы, прежде чем остановитесь на какой-то конкретной. Переход от традиционной модели локального развертывания к модели внешнего облака – это принципиальная для бизнеса смена парадигмы. Обычно бизнес комфортно себя чувствует, когда подразделение ИТ само управляет внутренними ресурсами. Но с переходом к облачной модели, несмотря на очевидную экономию затрат, бизнес сталкивается с необходимостью принять трудное решение: выйти из комфортного уголка и начать пользоваться облачными службами, чтобы сохранить конкурентоспособность. Такой переход не происходит за одну ночь; требуется несколько месяцев скрупулезного анализа, планирования и внедрения. В зависимости от соотношения затрат и прибылей, рисков и угроз безопасности предприятие может либо сохранить локальную модель, либо полностью перевести операции в облако, либо выбрать смешанное решение, которое, с одной стороны, позволяет сэкономить, а, с другой, оставить основные операции внутри предприятия. На рис. 1.2 показано, кто владеет ключевыми

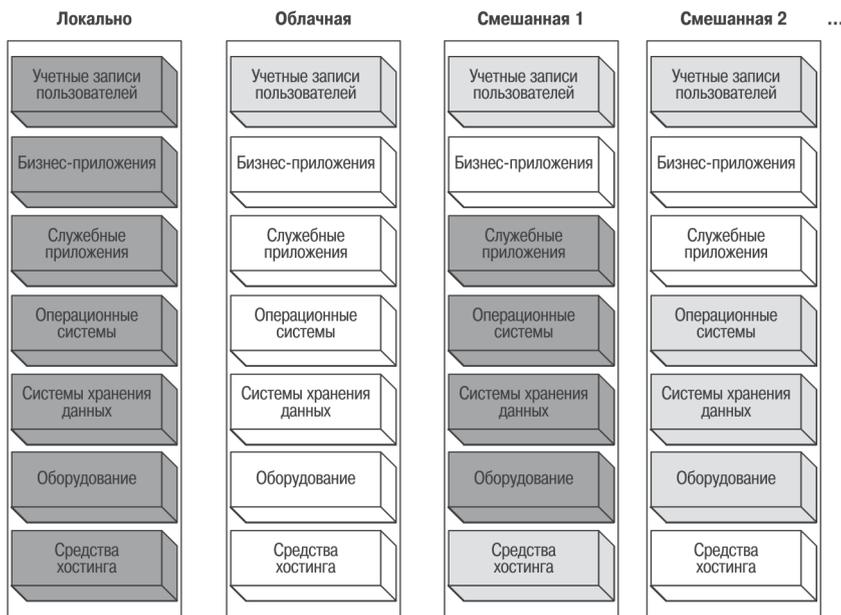
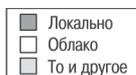


Рис. 1.2. Локальная, облачная и смешанная модель

ресурсами предприятия в случае локальной, облачной и смешанной модели.

Рекомендуется осуществлять переход постепенно, перенося не более одного приложения за раз. Когда в 2000 году стала популярной оффшорная модель разработки ПО, бизнес столкнулся с той же проблемой при оценке аутсорсинга. В настоящее время многие предприятия вложили значительные средства в оффшорную разработку и убедились, что это окупается. Но для смены парадигмы потребовалось время и изучение чужого опыта. Чтобы облачные службы добились успеха, бизнес должен сменить парадигму еще раз.

С локальной и облачной моделью все просто – ресурсы целиком находятся либо на территории предприятия, либо в облаке. Учетные записи пользователей обычно должны находиться в обоих местах из-за требований к единой точке входа в локальную и облачную систему. В смешанных моделях предприятие и поставщик служб долж-

ны договориться о том, где лучше размещать конкретные ресурсы: локально, в облаке или и там и тут. В смешанной модели 1 учетные записи пользователей и средства хостинга находятся в обоих местах, бизнес-приложения – в облаке, а служебные приложения, операционные системы, системы хранения данных и оборудование – внутри предприятия. В смешанной модели 2 учетные записи пользователей, операционные системы, системы хранения данных и оборудование находятся в обоих местах, тогда как бизнес-приложения, служебные приложения и средства хостинга – в облаке. Большинство компаний как правило выбирают наиболее подходящую для себя смешанную модель.

Экосистема облачных служб

В экосистеме облачных служб выделяют пять основных ролей, изображенных на рис. 1.3.



Рис. 1.3. Экосистема облачных служб

Поставщики служб

Поставщики служб – это компании, предоставляющие облачные службы предприятиям и потребителям. Они организуют гигантские центры обработки данных, в которых эксплуатируются виртуализированные и многократно резервированные программно-аппаратные системы. В эту категорию попадают такие поставщики, как Amazon, со своей службой EC2, и Microsoft, с платформой Windows Azure. Такие компании обладают опытом не только в сфере эксплуатации

ЦОД, но и в области управления масштабируемым программным обеспечением. Поставщики служб могут предлагать услуги предприятиям, потребителям или независимым поставщикам ПО.

Производители ПО

Программное обеспечение, предназначенное для работы внутри предприятия, сильно отличается от ПО, ориентированного на облако. Хотя функциональность, предоставляемая конечным пользователям, одинакова, архитектура может быть совершенно иной. Облачные службы должны проектироваться с учетом множественной аренды (multi-tenancy), масштабируемости, надежности и производительности в гораздо более широких масштабах, чем локальные приложения. Облачные службы работают в ЦОД, организуемых поставщиками служб. В некоторых случаях имеется существенное перекрытие между функциями поставщиков служб и производителей ПО. Например, платформа Microsoft Windows Azure, Microsoft Office 365 и Google Apps – это облачные системы, работающие в собственных ЦОД производителя. Эти производители ПО сочли экономически оправданным объединить оборудование и программное обеспечение в ЦОД, чтобы оптимизировать предложение своих продуктов через облако.

Независимые поставщики ПО

Независимые поставщики ПО (ISV) начинают играть важную роль в облачных службах благодаря опыту разработки вертикальных бизнес-приложений. Как правило, независимые поставщики создают вертикальные приложения на уже существующей платформе. Независимый поставщик выявляет спрос на конкретное решение на вертикальном рынке и стремится предложить его на различных имеющихся платформах. Облако – это прекрасная платформа для создания вертикальных решений. Например, независимый поставщик может разработать облачную систему для выставления счетов на медицинские услуги и предлагать ее как услугу многим врачам и больницам. Инфраструктура, необходимая для создания масштабируемого ПО с множественной арендой, уже имеется у поставщиков служб, так что независимый поставщик может сосредоточиться на разработке бизнес-решения и за счет этого с ошеломляющей быстротой выходить на новые рынки.