Лекция 1. Облачные технологии

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: основные характеристики облачных вычислений, технологии предоставления облачных сервисов (программное обеспечение, платформа и инфраструктура как сервис), границы управляемости облачных сервисов и виды организации облачных сервисов.

Цель лекции:

Цель данной лекции – получить предварительные сведения о назначении облачных технологий, видах облачных сервисов и способах их организации.

Введение

На портале образовательных услуг http://wiki.vspu.ru/ облачные технологии определяются как «технологии обработки данных, в которых компьютерные ресурсы предоставляются Интернет-пользователю как онлайн-сервис» [1]. Такой подход к организации обработки данных существенно влияет на эффективность процессов создания, внедрения и эксплуатацию информационных систем. При использовании облачных технологий архитектура информационной системы трансформируется в набор слабосвязанных веб сервисов, обеспечивающих хранение, обработку и передачу информации. Веб сервисы развертываются на высокоэффективных серверных системах дата центров, работоспособность которых поддерживается современными технологиями виртуализации и квалифицированным персоналом.

При сервисном облачном подходе к организации инфраструктуры информационных систем облачные технологии предоставляют возможности повышения эффективности за счет минимизации затрат на предоставления ИТ-услуг. Это объясняется тем, что при внедрении информационной системы для предприятий и организаций отпадает необходимость создания собственной инфраструктуры системы, резко сокращается время развертывания веб сервисов в облаке, происходит сокращение штата обслуживающего персонала ИТ-службы.

Облачные технологии позволяют поднять процесс автоматизации деятельности компаний на новый уровень, а разработку и развертывание новых программных сервисов сократить до минимальных сроков, обеспечивая конкурентные преимущества, выход на новые рынки, расширение клиентской базы, количества заказчиков и т.п.

При исследовании вопроса реализации облачных технологий важным понятием является модель развертывания приложений (рис 1.1) [2]:

- в инфраструктуре предприятия;
- в стороннем хостинге;
- в облаке.

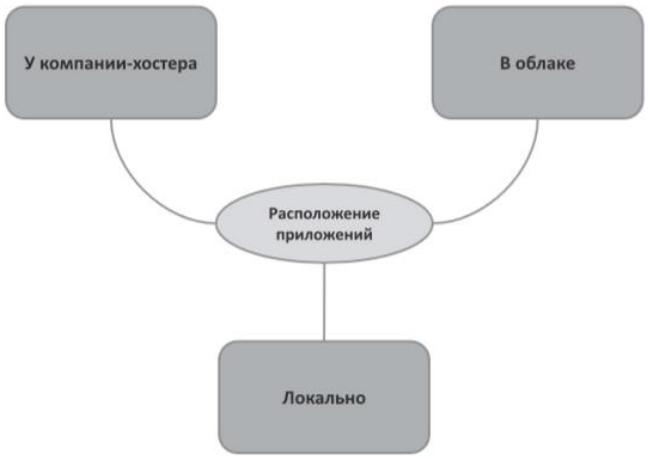

Рис. 1.1 – Варианты развертывания приложений

Развертывание в инфраструктуре предприятия. Эта модель развертывания приложений являлась общепринятой в течении последних десяти лет. Для её реализации предприятию необходимо создать собственную инфокоммуникационную инфраструктуру, приобрести лицензионное программное обеспечение, создать ИТ-служу для разработки собственного программного обеспечения и обслуживания инфраструктуры и программного обеспечения информационной системы. Такая модель предполагает инвестиции в аппаратное и программное обеспечение, сетевую инфраструктуру и высококвалифицированный персонал ИТ-службы.

Данная модель развертывания корпоративных приложений соответствовала исторически сформировавшемуся представлению менеджмента компаний о необходимости обеспечения полного контроля за инфраструктурой, аппаратным и программным обеспечением информационной системы предприятия.

Развертывание в стороннем хостинге. Данная модель развертывания приложений предполагает аренду таких инфраструктурных компонентов у хостера как серверное и программное обеспечение, которые обслуживаются централизовано хостером. При такой модели снижаются расходы предприятия на инфраструктурные компоненты информационной системы, а также обслуживающий персонал, поддерживающий безотказность и работоспособность серверов и инфраструктурного программного обеспечения хостера.

По сравнению с предыдущей моделью развертывания корпоративных приложений предприятие арендует и оплачивает фиксированные серверные, сетевые и программные ресурсы у хостера. При этом со стороны предприятия имеется меньший контроль за инфраструктурой, аппаратными и программными средствами. Оплата арендуемых ресурсов у хостера производится предприятием независимо от использования заявленных ресурсов.

Развертывание в облаке. Модель развертывания приложения в облаке предполагает его установку на сервере облачного дата центра. В отличие от развертывания приложения в стороннем хостинге, оплата арендуемых инфраструктурных, сетевых и программных ресурсов осуществляется по факту использования арендуемых ресурсов. Отличительной особенностью использования данной модели при создании корпоративной информационной системы предприятия является отсутствие контроля со стороны предприятия за инфраструктурой, аппаратным и сетевым обеспечением.

Облачные вычисления представляют собой модель сетевого доступа к вычислительным ресурсам, таким как сети передачи данных, серверы, устройства хранения данных, приложения и сервисы [3]. Доступ к вычислительным ресурсам предоставляется по запросу пользователя, при этом реализуется автоматическая процедура предоставления и освобождения ресурсов.

Облачные вычисления предоставляют пользователю возможности по оперативному управлению ресурсами при использовании требуемых вычислительных мощностей при изменении внешней нагрузки и особенностей решаемых задач.

Основные характеристики облачных вычислений

Масштабируемость. Облачные вычисления обеспечивают для информационных систем возможность поддержания требуемых уровней обслуживания (доступности, быстродействия, надежности) для различных нагрузок и объемов обрабатываемой информации. Масштабируемость обеспечивается за счет оперативного подключения или отключения одновременно запускаемых экземпляров приложений, предоставления необходимого количества серверов, систем хранения и передачи данных. Дата центры формируются на базе типовое оборудование, что снижает общую стоимость владения и упрощает сопровождение инфраструктуры [4].

Эластичность. Бизнесу необходимо адаптировать информационные системы для поддержания конкурентоспособности в современных быстро меняющихся условиях. Внедрение новых продуктов или услугу, быстрое проведение для этого полного цикла планирования, проектирования и разработки информационной системы предполагает использование гибких технологий и методологию DevOps. Эластичность облачных вычислений

позволяет быстро нарастить мощность информационной инфраструктуры предприятия, с минимальными начальными инвестициями в оборудование и программное обеспечение. Эластичность связана с масштабируемостью приложений, так как решает задачу моментального изменения количества вычислительных ресурсов, выделяемых для работы информационной системы.

Мультитенантность. Мультитенантность базируется на технологиях виртуализации и обеспечивает в рамках центра обработки данных надежную изоляцию большого количества виртуальных машин, которые могут использоваться разными организациями, требующими определенные уровни изоляции, предназначаться для различных групп пользователей, характеризующихся индивидуальными политиками безопасности, ориентироваться на разные категории потребителей с определенными настройками безопасности. Мультитенантность обеспечивает снижения расходов за счет максимального использования общих ресурсов для обслуживания различных групп пользователей, разных организаций, разных категорий потребителей.

Оплата облачных ресурсов. Использование облачной инфраструктуры для построения информационной системы предприятия позволяет перевести значительную часть капитальных затрат в операционные издержки. Предприятие имеет возможность заказывать на планируемое время необходимый объем вычислительных ресурсов, что обеспечивает оптимизацию затрат, связанных с работой информационных систем предприятия. Кроме того, мультитенантность облачных инфраструктур, позволяет распределять ресурсы между различными потребителями, что способствует дополнительному снижению расходов на информационную систему. Эластичность облачных инфраструктур обеспечивает динамическое изменение объемов потребляемых информационной системой ресурсов, как в сторону увеличения, так и уменьшения, что оптимизирует затраты предприятия на информационные технологии.

Самообслуживание. При использовании облачных вычислений, гибкого подхода и методологии DevOps задачи модификации функционала информационной системы для вывода на рынок нового продукта или услуги значительно упрощаются, сокращается время внесения изменений в функционирующую систему. Имеющийся в облачных платформах инструментарий планирования и развертывания вычислительной инфраструктуры и приложений позволяет значительно сократить время выхода на рынок новых товаров и услуг. Инструменты самообслуживания позволяют создавать скрипты для автоматического формирования инфраструктуры информационной системы (серверов, систем связи и хранения данных, операционных систем и прикладного программного обеспечения). При этом генерация требуемой инфраструктуры информационной системы выполняется за несколько десятков минут.

Следует отметить, что только сочетание нескольких характеристик облачных вычислений приводит к повышению эффективности информационных систем, способствуя увеличению доходов и сокращени. расходов. Так, оплата только использованных ресурсов максимально эффективна в сочетании с эластичностью инфраструктуры. Эластичность, в свою очередь, предполагает, что приложения масштабируются, в противном случае, быстрое выделение ресурсов не приведет к повышению производительности.

Облачные вычисления и предоставляемые ими сервисы

Реализация облачных вычислений и облачные сервисы предоставляются по запросам пользователей на условиях аренды вычислительных ресурсов (серверов, систем передачи и хранения данных, системного и прикладного программного обеспечения) и сервисов [5]. Потребности пользователей в процессе выполнения задач информационной системы предприятия могут увеличиваться или уменьшаться зависимости от повышения или понижения рабочих нагрузок. При этом пользователи оплачивают только за реальное использование арендованных вычислительных ресурсов и сервисов. Условия аренды предполагает предоставления облачных ресурсов в соответствии с требуемыми уровнями обслуживания по доступности, производительности и надежности. Существуют различные модели предоставления облачных ресурсов: программное обеспечение как сервис; платформа как сервис; инфраструктура как сервис.

Инфраструктура как сервис

Для формирования в облаке инфраструктуры информационной системы предприятия (серверов, систем хранения данных, сетевого оборудования) используют *модель предоставления инфраструктуры как сервиса* (Infrastructure as a Service, IaaS). В данной модели в обязанности поставщика сервиса входит управление и поддержание работоспособности всей облачной инфраструктуры в соответствии с заданными уровнями обслуживания, которые фиксируются в Соглашении о предоставлении сервисов (SLA). Потребитель (ИТ-персонал информационной системы предприятия) самостоятельно устанавливает операционные системы из стандартных образов ОС и прикладное программное обеспечение. В задачи ИТ-персонала входит поддержание программных средств в актуальном состоянии, их обновление и модификация. Ключевыми характеристиками сервисов в SLA для данной модели являются доступность виртуального сервера, время развертывания образа ОС. В данной модели оплата сервиса производится по фактическому использованию облачных ресурсов, пользователь имеет возможность увеличивать или уменьшать объем

используемой инфраструктуры через специальные порталы, предоставляемые поставщиками сервисов.

Платформа как сервис

Облачная платформа, предоставляемая как сервис, как правило, включает операционную систему и прикладные сервисы. Модель предоставления платформы как сервиса (Platform as a Service, PaaS) используется компаниями для организации процесса разработки программных систем, используя сервисы непрерывной интеграции (Continuous Integration – CI) и непрерывной доставки (Continuous delivery – CD). Данные сервисы обеспечивают постоянное и согласованное тестирование, а также создание кода и загрузку его в любой целевой объект. Платформа как сервис предполагает использование инфраструктуры как сервис. Примером платформы как сервис может служить Windows Azure. Ключевыми характеристиками сервисов в SLA для данной модели являются доступность среды выполнения приложений и ее производительность. Оплата облачной платформы рассчитывается исходя из объема использованных вычислительных ресурсов, таких как: время работы приложения; объем данных и количество операций с данными (транзакций); сетевой трафик.

Программное обеспечение как сервис

Для развертывания приложений заказчика в облаке применяют модель предоставления программного обеспечения как сервиса (Software as a Service, SaaS). Применение программного обеспечения как сервис предполагает использование платформы и инфраструктуры как сервис. Широко распространенными приложениями, предоставляемыми как облачный сервис является Azure DevOps, который включает ряд сервисов поддержки методологии разработки и развертывания программного обеспечения DevOps, система командной разработки проектов Bitbucket, система управления проектами Jira. При использовании программного обеспечения как сервис пользователь получает доступ к приложению через Интернет. Существует большое количество бесплатных облачных сервисов для индивидуального использования, а корпоративные пользователи осуществляют оплату по факту использования сервисов. Ключевыми характеристиками сервисов в SLA для данной модели являются доступность программных сервисов и их производительность. Для данной модели пользователь имеет очень ограниченные возможности по настройке приложения под свои бизнес-требования. Оплата конечного сервиса, как правило, производится ежемесячно и рассчитывается на основе количества пользователей приложения.

Границы управляемости облачных сервисов

При использовании различных моделей предоставления облачных сервисов ИТ-службам предприятий предоставляются различные возможности по управления инфраструктурой и программным обеспечением информационных систем, которые отличаются от возможностей управления собственной инфраструктурой предприятия. Если используется модель предоставления инфраструктуры как сервиса (IaaS), то ИТ-службы предприятия могут управлять системами управления базами данных, программной средой и приложениями, а также решать вопросы интеграции в системе и управление политиками безопасности. Если используется модель предоставления платформы как сервиса (PaaS), то разработчики программных систем, которые, в основном, используют такую модель, имеют возможность управлять только приложениями, которые они разрабатывают. Если используется модель предоставления программного обеспечения как сервиса (SaaS), то пользователям предоставляется возможность только использовать предоставляемые в облаке приложения. На рис. 1.2. иллюстрируются возможности пользователей по управлению ресурсами информационной системы в зависимости от используемой модели предоставления облачных сервисов [6].

IaaS	PaaS	SaaS
Данные	Данные	Данные
Приложения	Приложения	Приложения
Базы данных	Базы данных	Базы данных
Операционная система	Операционная система	Операционная система
Виртуализация	Виртуализация	Виртуализация
Физический сервер	Физический сервер	Физический сервер
Сети и хранилища	Сети и хранилища	Сети и хранилища
Дата-центр	Дата-центр	Дата-центр

Рис. 1.2. Границы управляемости

Как видно из рис. 1.2, в собственной инфраструктуре информационной системы предприятия ИТ-служба полностью управляет и контролирует всеми ресурсами и программным обеспечением. Модель предоставления облачных

сервисов IaaS позволяет управлять и контролировать среду исполнения кода, политику безопасности, способы и технологии интеграции, системы управления базами данных. Модель предоставления платформы как сервиса PaaS предполагает ограниченные возможности управления сервисами.

Типы организации облачных сервисов

В зависимости от организации различают следующие типы облачных сервисов: публичные, частные и гибридные облака [7].

Публичное облако представляет собой программно-аппаратную компьютерную систему, которая создается в центре обработки. Такая система предназначена для использования неограниченным числом пользователей.

Поставщик облачных услуг предоставляет клиентам доступ к ресурсам, обеспечивая заданный уровень качества. Пользователи арендуют аппаратные и программные ресурсы и производят оплату по факту их использования. Управление арендуемыми ресурсами осуществляется пользователями посредством облачных веб-порталов.

Частное облако в отличие от публичного создается предприятиями в корпоративном центре обработки данных и предназначено для сотрудников и клиентов предприятия. Частное облако предоставляет веб-сервисы, которые необходимы для обеспечения работы конкретного предприятия.

Гибридное облако представляет собой комбинацию публичного и частного облака. В гибридном облаке имеется возможность варьировать способы реализации задач информационной системы, используя публичное и частное облако. Такой подход позволяет выбирать наиболее подходящий вариант размещения рабочих нагрузок. Например, если нагрузка на веб-сайт колеблется в широких пределах, его можно разместить в публичном облаке и подключить к защищенной базе данных в корпоративном частном облаке.

Ключевые термины

Облачные вычисления — это такой подход к размещению, предоставлению и потреблению приложений и компьютерных ресурсов, при котором приложения и ресурсы становятся доступны через Интернет в виде сервисов.

Программное обеспечение как сервис – модель предоставления программного обеспечения как сервиса (Software as a Service, SaaS), которая обеспечивает возможность аренды приложений.

Платформа как сервис – модель предоставления платформы как сервиса (Platform as a Service, PaaS), которая предоставляет возможность аренды платформы, включающей операционную систему и прикладные сервисы.

Инфраструктура как сервис – модель предоставления инфраструктуры (аппаратных ресурсов) как сервиса (Infrastructure as a Service, IaaS), которая предоставляет возможность аренды таких инфраструктурных ресурсов, как серверы, устройства хранения данных и сетевое оборудование.

Публичное облако — это инфраструктура, которая создается в центре обработки данных провайдера и предназначена для свободного использования широкой публикой.

Частное облако — это инфраструктура, которая создается в собственном центре обработки данных компании, а пользователям предоставляются инструменты для самостоятельного использования ее ресурсов.

Гибридное облако — это инфраструктура, сочетающая в себе публичное и частное облако.

Вопросы для самопроверки

1. Основные модели расположения приложений.
2. Основные характеристики облачных вычислений.
3. Облачные вычисления и предоставляемые ими сервисы.
4. Облачные сервисы и границы управляемости.
5. Виды организации облачных сервисов.
6. Поясните назначение понятия Публичное облако.
7. Поясните назначение понятия Частное облако.
8. Поясните назначение понятия Гибридное облако.
9. Что определяет понятие Облачные вычисления?
10. Что определяет понятие Software as a Service?
11. Что определяет понятие Platform as a Service?
12. Что определяет понятие Infrastructure as a Service?

Литература

1. Облачные технологии. http://wiki.vspu.ru/playground/comp_8.
2. Облачные вычисления. http://studwood.ru/1840886/informatika/oblachnye_vychisleniya/.

3. Облачные вычисления.
   http://ru.wikipedia.org/wiki/Облачные_вычисления.
4. Понятия облачных вычислений. Основные характеристики облачных вычислений. http://learn-more.kz/konspekty-
   http://www.trinitygroup.ru/solution/infrastucture/virtualization/vdi/.
5. Что такое облачные вычисления? http://azure.microsoft.com/ru-ru/overview/what-is-cloud-computing/#benefits.
6. Федоров А., Мартынов Д. Windows Azure: облачная платформа Microsoft, 2010.
7. Модели облачных сервисов: разница между IaaS, SaaS, PaaS и примеры. http://www.sim-networks.com/ru/blog/cloud-computing-service-models.

Лекция 2. Веб-службы в облаке

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: платформа Windows Azure, используемые на платформе роли, виртуальные машины платформы, сервисы хранения данных, SQL Azure.

Цель лекции:

Цель данной лекции – получить предварительные сведения о возможностях платформы Windows Azure.

Введение

Платформа Windows Azure Platform предназначена для разработки и выполнения облачных сервисов. Для пользователей данная платформа предоставляется как сервис на основе модели PaaS. В состав платформы входят:

- Windows Azure;
- SQL Azure;
- Windows Azure AppFabric.

Windows Azure является облачной операционной системой, которая обладает такими свойствами как эластичность, масштабируемость и безопасность. С помощью данной операционной системе реализуются функции предоставления серверных ресурсов и систем передачи и хранения данных. Кроме того, Windows Azure предоставляет возможности управления веб сервисами.

SQL Azure является сервисом для поддержки работы с реляционными базами данных. Данный сервис обеспечивает хранение и манипулирование данными реляционной модели аналогично основным возможностям Microsoft SQL Server. Функции администрирования и сопровождения данных реализуются сервисом.

Windows Azure AppFabric предоставляет сервисы, которые используются для создания системы коммуникаций и обеспечения управления доступом к ресурсам. Данный сервис предназначен для поддержки интеграции облачных приложений между собой, а также приложений, которые развернуты в инфраструктуре клиента.

На рис. 2.1 представлены компоненты платформы Windows Azure.

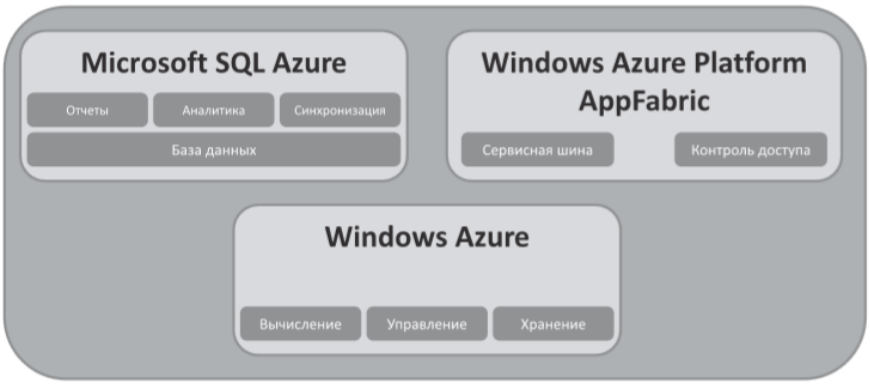

Рис. 2.1. Платформа Microsoft Windows Azure

Платформа Windows Azure расширяется, добавляются новые компоненты, улучшается функциональность существующих компонент. При этом использование сервисов Windows Azure можно выполняться различными способами: из приложений, работающих на этой платформе; из приложений, которые развернуты на инфраструктуре клиентов.

Платформа Windows Azure

Azure представляет собой облачную платформу, которая предоставляется пользователю как сервис [1]. На данной платформе пользователь может размещать приложения для реализации своих бизнес-задач. При увеличении нагрузки на приложение имеется возможность достаточно легко провести масштабирование ресурсов системы. В Azure входят службы, обеспечивающие как разработку, так и развертывание приложений пользователя. При развертывании приложений обеспечивается пользовательский контроль над их размещением.

Функциональность Windows Azure базируется на технологиях виртуализации с использованием контроллера структуры Fabric Controller. В задачи контроллера входят организация массива экземпляров виртуальных машин, автоматическое управление ресурсами, балансировка нагрузки, обеспечение устойчивости к сбоям, репликация в одном и/или географически удаленных центрах обработки данных. Кроме того, Fabric Controller обеспечивает пользователям и приложениям доступ к платформе Windows Azure. На рис. 2.2 приведены основные компоненты Windows Azure.

Платформа Windows Azure предоставляет набор сервисов для разработки приложений:

- вычислительные сервисы;
- сервисы хранения данных;
- коммуникационные сервисы;
- сервисы обеспечения безопасности;
- прикладные сервисы.

Рис. 2.2 – Компоненты Windows Azure

Вычислительные сервисы обеспечивают выполнение приложений на различных языках программирования (языки платформы .NET, Java, PHP, Python, Ruby on Rails и нативный код).

Сервисы хранения данных предоставляют возможности использовать распределенную систему хранения данных. Для облачных систем используются табличные структуры, бинарные объекты, очереди сообщений, а также традиционные файловые системы.

Для обмена сообщениями и брокера соединений с другими облачными сервисами или сервисами предоставляются коммуникационные сервисы.

Сервисы обеспечения безопасности управляют доступом на основе корпоративных политик, поддерживают механизмы федерации и внутренней идентификации.

Прикладные сервисы используются для разработки облачных приложений и прикладных сервисов.

Служба приложений ориентирована на создание веб-приложений для поддержки мобильных клиентов. Она позволяет использовать интерфейсы REST API. Платформа Azure обеспечивает аутентификацию посредством поставщиков социальных сетей, автомасштабирование на основе трафика, поддержку технологии разработки приложений DevOps для тестирования в рабочей среде и непрерывное развертывание.

При создании веб-приложений, виртуальных машин и облачных служб Azure используются веб-роли и рабочие роли.

Веб-роль в Azure обеспечивает поддержку протоколов HTTP и HTTPS и предоставляет выделенный веб-сервер служб IIS. Данная роль обеспечивает интерфейс веб-приложения.

Рабочая роль предназначена для реализации бизнес-логики приложения и может использоваться для выполнения различных асинхронных задач, для которых не требуется взаимодействие с пользователем.

На рис. 2.3 показаны основные роли в Windows Azure («БН» — средство балансировки нагрузки).

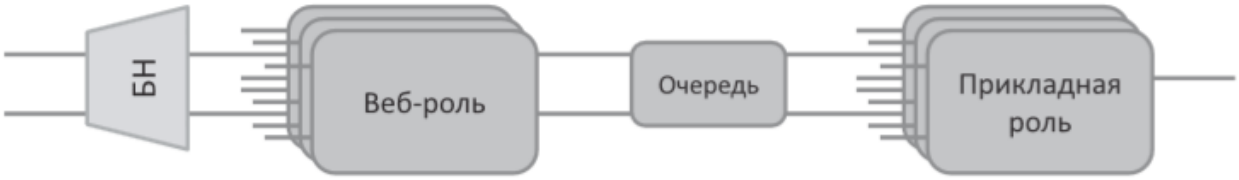

Рис. 2.3. Роли в Windows Azure

Виртуальные машины

Виртуальные машины Azure представляют собой один из типов вычислительных ресурсов, предоставляемых пользователю [2]. Эффективность использования виртуальных машин состоит в разнообразии их архитектурно-технических возможностей, использование на условиях аренды в соответствии с договором об уровне предоставления сервиса. Пользователи виртуальных машин должны осуществлять работы по установке и обновлению программного обеспечения, работающего на виртуальной машине.

Основными направлениями использования виртуальных машин Azure являются:

- разработка и тестирование;
- развертывание приложения в облаке.

При использовании виртуальных машин для организации процесса разработки и тестирования имеется возможность быстрого формирования и модификации компьютеров с определенными конфигурациями, которые необходимы для реализации бизнес-задач проекта.

Развертывание приложения в облаке оптимизирует затраты пользователей при наличии колебаний уровня спроса на приложение. При увеличении спроса создаются дополнительные виртуальные машины, если уровень спроса падает, то некоторые виртуальные машины отключаются. Пользователь может оперативно увеличить ресурсы виртуальных машин, используемых приложением, а также развернуть дополнительные виртуальные машины в соответствии с требованиями.

При создании виртуальных машин важными вопросами являются:

- размер виртуальной машины;
- максимальное число виртуальных машин, которые можно создать;
- операционная система, под управлением которой будет работать виртуальная машина.

Платформа Azure предоставляет широкий спектр конфигураций виртуальных машин, которые имеют следующие ориентации [3]:

- общего назначения;
- оптимизированные для вычислений;
- оптимизированные для памяти;

- оптимизированные для хранилища;
- для ресурсоемкой отрисовки изображений и редактирования видео;
- для высокопроизводительных вычислений.

Виртуальные машины общего назначения (Standart) серии Av2 можно развертывать с использованием оборудования и процессоров разных типов. Такие виртуальные машины лучше всего подходят для рабочих нагрузок начального уровня – проведение процессов разработки и тестирования, веб-серверы с низким уровнем трафика, базы данных малого и среднего размера, экспериментальные решения и репозитории кода. В табл. 2.1 приведены характеристики виртуальных машин серии Av2.

Таблица 2.1 – Характеристики виртуальных машин общего назначения

Размер	**Вир-туаль-ное ядро**	**Па-мять: ГиБ**	**Врем. Храни-лище (SSD): -ГиБ**	**Макс. проп. способ. вр. хран.: операции ввода-вывода в сек./скор. чтения (МБит/с)/скор. записи (МБит/с).**	**Максимальное число дисков данных/проп. способность: операции ввода-вывода в секунду.**	**Макси-мальное число сетевых адапте-ров**	**Ожидае-мая проп. Спосо-бность сети (Мбит/с)**
Standard_A1_v2	1	2	10	1000/20/10	2/2x500	2	250
Standard_A2_v2	2	4	20	2000/40/20	4/4x500	2	500
Standard_A4_v2	4	8	40	4000/80/40	8/8x500	4	1000
Standard_A8_v2	8	16	80	8000/160/80	16/16x500	8	2000
Standard_A2m_v2	2	16	20	2000/40/20	4/4x500	2	500
Standard_A4m_v2	4	32	40	4000/80/40	8/8x500	4	1000
Standard_A8m_v2	8	64	80	8000/160/80	16/16x500	8	2000

Службы хранения данных

Службы хранения данных Microsoft Azure представляют собой безопасное облачное хранилище с высоким уровнем масштабируемости для данных, приложений и рабочих нагрузок

В Microsoft Azure имеется несколько типов служб для работы с данными определенного типа [4]:

- таблицы – Azure Tables;
- бинарные объекты – Azure Blobs;

- очереди Azure;
- файлы Azure.

Таблицы Azure представляют собой структурированное хранилище нереляционного типа, в которых используется система хранения « ключ-значение». Таблицы являются хорошим решением для хранения большого объема неструктурированных или полуструктурированных данных. Таблицы Azure подходят для хранения данных без SQL, которые не могут храниться в системе реляционных баз данных, например SQL Server.

Табличное хранилище поддерживает гибкую схему данных, идеально подходит для веб-приложений, адресных книг и других пользовательских данных.

Бинарные объекты – это наиболее универсальное решение для хранения от Azure, в котором хранятся неструктурированные данные в виде объекта. Эти неструктурированные данные включают документы, изображения, видео, файлы журналов, а также диски виртуальных машин.

Хранилище BLOB-объектов может быть доступно с помощью HTTP/HTTPS. Кроме того, к нему также можно получить программный доступ с помощью Azure PowerShell, REST API, Azure CLI или клиентских библиотек (Python. PHP, Ruby и т.д.).

Тип BLOB-объекта может быть определен только во время создания. После создания большого двоичного объекта изменить его тип невозможно. Например, нельзя преобразовать большой двоичный объект блока в большой двоичный объект страницы или добавить большой двоичный объект.

Все большие двоичные объекты должны находиться в контейнере, который концептуально является каталогом. Контейнер может хранить неограниченное количество больших двоичных объектов.

Существует ряд решений, которые используются для переноса/копирования существующих данных в хранилище BLOB-объектов. Одним из распространенных инструментов является AzCopy.

Фабрика данных Azure (Azure Data Factory) поддерживает копирование данных в хранилище BLOB-объектов и из него.

Существуют следующие типы больших двоичных объектов Azure:

- блочные BLOB-объекты;
- BLOB-объект добавления;
- страничные BLOB-объекты.

Блочные Blob-объекты позволяют эффективно загружать большие файлы BLOB-объектов. Каждый блок имеет идентификатор блока. Он хранит текстовые и двоичные данные.

BLOB-объект добавления состоит из блоков и оптимизирован для операций добавления. При изменении добавляемого BLOB-объекта большие двоичные объекты добавляются в конец существующего BLOB-объекта, а изменение существующих блоков не поддерживается. Одним из вариантов

использования такого BLOB-объекта является ведение журнала данных виртуальных машин.

Страничные BLOB-объекты – это жесткие диски виртуальных машин. В то время как блочные BLOB-объекты состоят из блоков, страничные BLOB-объекты состоят из 512 байтовых страниц. Все диски виртуальных машин Azure используют страничные BLOB-объекты. Максимальный размер большого двоичного объекта страницы составляет 8 ТБ.

Очереди – это особый тип структуры данных, которые можно использовать для обмена сообщениями между компонентами. Эти компоненты могут отображаться как в облаке, так и в локальной среде.

Хранилище очереди позволяет помещать сообщения в очередь и асинхронно обрабатывать эти сообщения. Поэтому хранилище очередей является идеальным методом обработки событий, не требующих определенного порядка. Максимальный размер каждого сообщения составляет 64 КБ. Можно изменить содержимое сообщения на месте в очереди. Очередь сообщений может содержать неограниченное количество сообщений, если поддерживается емкость учетной записи хранения. Максимальное время, в течение которого сообщение может оставаться в очереди, составляет 7 дней.

Файлы Azure представляют собой решение для управляемого общего доступа к файлам в облачном приложении. Они действуют так же, как общий сетевой ресурс NAS (Network Attached Storage)), и они могут быть отформатированные под NTFS виртуальные диски (Virtual Hard Drives, VHDs) в страничных бинарных объектах. Их можно подключить к локальным виртуальным машинам и виртуальным машинам Azure с Windows, Linux и MAC OS.

Доступ к файлам Azure можно получить двумя способами:

- прямой облачный доступ;
- через синхронизацию файлов Azure.

Общие файловые ресурсы Azure можно кэшировать на серверах Windows с помощью синхронизации файлов Azure для быстрого доступа к месту использования данных.

Будучи управляемой службой, файловая папка Azure обеспечивает встроенную отказоустойчивость.

Для реализации программных решений в облаке можно сформировать архитектуру на базе Windows Azure — с ролями, сервисами хранения и вычислительными сервисами (рис. 2.4).

Следует отметить, что все внешние соединения происходят через средства балансировки нагрузки, тогда как внутренние коммуникации между ролями происходят без использования средств балансировки нагрузки, непосредственно через TCP-порты, хотя могут использоваться и очереди сообщений.

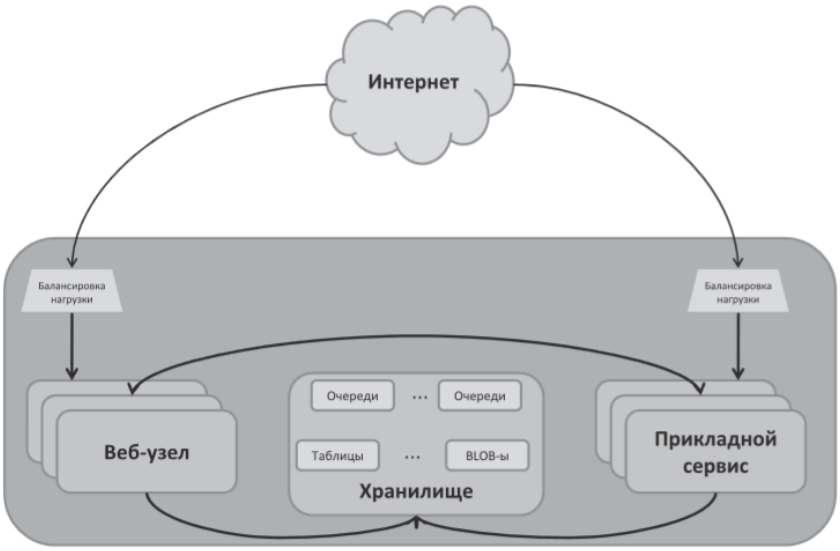

Рис. 2.4. Использование сервисов Windows Azure

SQL Azure

База данных SQL Azure — это полностью управляемая реляционная база данных со встроенными интеллектуальными функциями, поддерживающими функции самостоятельного управления, такие как настройка производительности и оповещения об угрозах. Корпорация Майкрософт выполняет все исправления и обновления базы кода, а также управляет базовой инфраструктурой. Благодаря высокой совместимости с SQL Server можно также перенести базы данных в управляемый экземпляр базы данных SQL без изменения приложений.

SQL Azure использует сервис Cloud Fabric для управления экземплярами базы данных и обеспечения их развертывания, администрирования, обновления, мониторинга. Пользователи SQL Azure должны самостоятельно создать схему базы данных, провести оптимизацию запросов и поддерживать заданный уровень безопасности. Основные компоненты SQL Azure показаны на рис. 2.5.

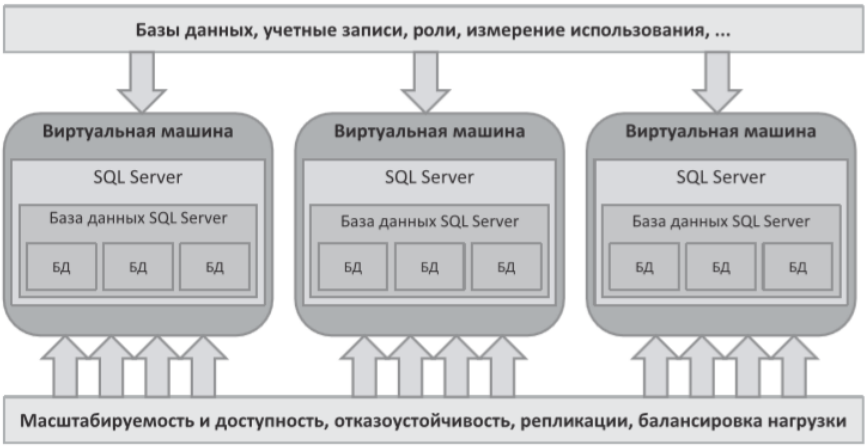

Рис. 2.5. Компоненты SQL Azure

Так как SQL Azure построена на основе SQL Server, пользователи получают знакомую реляционную модель данных, которая практически симметрична с серверами SQL Server, развернутыми у заказчиков. Поддерживаются многие возможности ядра SQL Server, хотя в текущей реализации облачной базы данных SQL Azure существует ряд ограничений, которые мы кратко перечислим ниже.

Ограничения на административном уровне

- в SQL Azure сервер баз данных не доступен на физическом уровне;
- предоставляется доступ к базе данных MASTER, но не к конструкциям уровня сервера, таким как sp_configure, командам DBCC, представлениям для управления данными (DMV) и системным представлениям.

Ключевые термины

Платформа Windows Azure — облачная платформа, которая предоставляется пользователю как сервис.

Виртуальная машина Azure – вычислительный ресурс, предоставляемый пользователю как сервис.

Службы хранения данных Microsoft Azure – облачное хранилище с высоким уровнем масштабируемости для данных, приложений и рабочих нагрузок.

Таблицы Azure – структурированное хранилище **нереляционного типа, использующее систему хранения «ключ-значение».**

Бинарные объекты – хранилище BLOB-объектов, в котором хранятся неструктурированные данные в виде двоичных объектов.

Очереди – тип структуры данных, которые используются для обмена сообщениями между компонентами.

Файлы Azure – решение для управляемого общего доступа к файлам в облачном приложении.

SQL Azure —управляемая реляционная база данных со встроенными интеллектуальными функциями.

Вопросы для самопроверки

1. Назначение платформа Windows Azure.
2. Основные компоненты платформы Windows Azure.
3. Назначение сервиса Windows Azure AppFabric.
4. Основной набор сервисов платформы Windows Azure для разработки приложений.
5. Назначение и направления использования виртуальных машин Azure.

6. Сколько виртуальных ядер может быть у виртуальной машины Azure общего назначения (Standart).
7. Назначение службы хранения данных Microsoft Azure.
8. Типы служб для работы с данными в Microsoft Azure.
9. Назначение и характеристики таблиц Azure**.**
10. Назначение и характеристики хранилища BLOB-объектов.
11. Назначение и характеристики очередей Azure**.**
12. Назначение и характеристики файлов Azure**.**
13. Назначение базы данных SQL Azure.

Литература

1. Руководство по началу работы для разработчиков Azure. https://docs.microsoft.com/ru-ru/azure/guides/developer/azure-developer-guide.
2. Виртуальные машины Windows в Azure. https://docs.microsoft.com/ru-ru/azure/virtual-machines/windows/overview.
3. Размеры виртуальных машин в Azure. https://docs.microsoft.com/ru-ru/azure/virtual-machines/sizes.
4. Azure Storage : Introduction. https://social.technet.microsoft.com/wiki/contents/articles/52312.azure-storage-introduction.aspx?wa=wsignin1.0
5. Azure SQL Database. https://www.azure.cn/en-us/pricing/details/sql-database/

Лекция 3. Windows Azure SDK

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: назначение и использование пакета Azure SDK в приложениях платформы .NET, проверка подлинности с помощью Azure SDK для приложений, создание и настройка журналов событий для приложений, интегрированных с Azure SDK.

Цель лекции:

Цель данной лекции – получить предварительные сведения о пакете Azure SDK, его возможностях и способах применения.

Введение

Пакет Azure SDK является инструментальным средством, которое обеспечивает взаимодействие приложений .NET со службами Azure [1]. С помощью инструментария Azure SDK можно из приложений .NET выполнять такие задачи как манипулирование данными хранилища BLOB-объектов, получение ключей из хранилища Azure Key Vault и другие операции. Пакет Azure SDK для .NET можно использовать в приложениях .NET Core и .NET Framework. На рис. 3.1 приведены основные возможности Azure SDK.

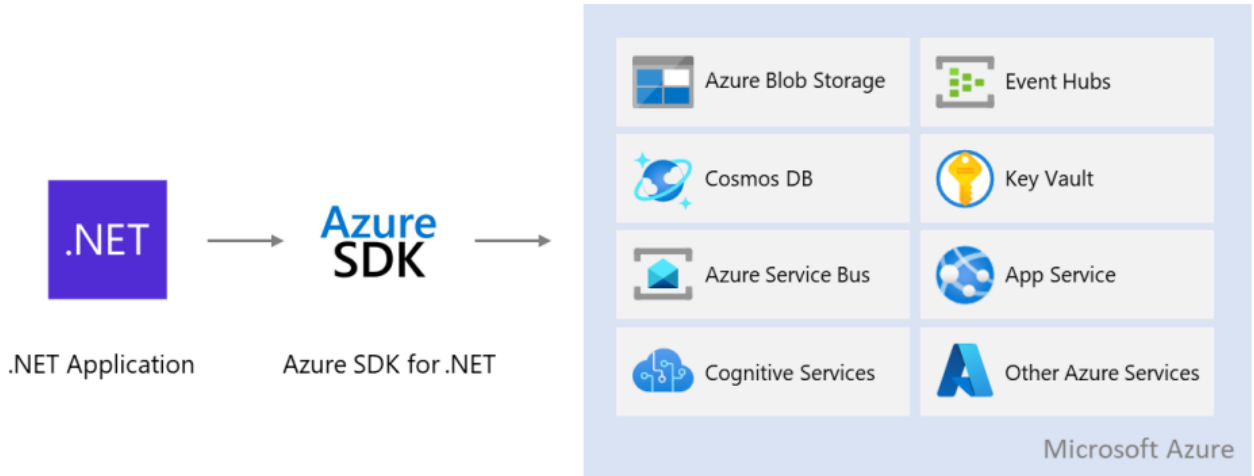

Рисунок 3.1 – Возможности Azure SDK

Разрабатываемое .NET приложение может через Azure SDK взаимодействовать со следующими облачными сервисами:

- Azure Blob Storage;
- Cosmos DB;

- Azure Service Bus;
- Cognitive Services;
- Event Hubs;
- Key Vault;
- App Servie.

Azure Blob Storage является сервисом для хранения больших двоичных объектов в облачной среде [2]. Данной сервис предназначен для хранения больших объемов неструктурированных данных.

Azure Cosmos DB представляет собой управляемую службу баз данных нереляционного типа для разработки различных приложений [3]. Отличительными особенностями данной службы является одинаковое время отклика, а также автоматическое и мгновенное масштабирование, что гарантирует быструю передачу при любых масштабах.

Azure Service Bus является сервисом для обмена сообщениями между приложениями и устройствами [4]. Данный сервис является полностью управляемым корпоративным брокером сообщений с поддержкой очередей сообщений и разделов для публикации и подписки.

Cognitive Services представляют собой облачные службы с REST API и пакетами SDK клиентской библиотеки, которые обеспечивают интеграцию когнитивных средств искусственного интеллекта в разрабатываемые приложения [5]. Azure Cognitive Services содержит различные службы искусственного интеллекта, которые позволяют создавать когнитивные решения, использующие функции просмотра и прослушивания данных, функцию речи, анализа данных и даже принятия решений.

Event Hubs Azure является сервисом для потоковой передачи больших данных и службу приема событий [6]. Данный сервис может получать и обрабатывать миллионы событий в секунду. Данные, отправляемые в концентратор событий, можно преобразовывать и сохранять с помощью любого поставщика аналитики в реальном времени, а также с помощью адаптеров пакетной обработки или хранения.

Azure Key Vault представляют собой облачную службу для безопасного хранения и получения доступа к критическим ресурсам [7]. Такими критическими ресурсами могут быть ключи API, пароли или криптографические ключи. Служба Key Vault поддерживает два типа контейнеров: хранилища и пулы управляемых аппаратных модулей безопасности (HSM). Хранилища обеспечивают хранение программного обеспечения и ключей, критических ресурсов и сертификатов с поддержкой HSM. Управляемые пулы HSM поддерживают только ключи с поддержкой HSM.

App Servie Azure является сервисом на базе HTTP для размещения веб-приложений, интерфейсов REST API и серверной части мобильных решений [8]. С его помощью можно выполнять разработку на следующих алгоритмических

языках: .NET, .NET Core, Java, Ruby, Node.js, PHP или Python. Приложения без затруднений работают и масштабируются в средах на основе операционных систем Windows и Linux.

Использование пакета Azure SDK для .NET в приложениях

Разработка приложений на платформе .NET совместно с Azure SDK предполагает выполнение следующих шагов.

Шаг 1. Разработчик должен определиться с использование определенного пакета SDK. Для этого необходимо изучить документацию [9] и выбрать требуемую службу Azure, которую необходимо интегрировать в .NET приложение. Как правило службы включают в свой состав клиентские пакеты для работы со службой и пакеты управления для создания экземпляров службы и управления ими. Для установки выбранного пакета в приложении используют менеджер пакетов NuGet.

Шаг 2. Для работы с приложением необходимо настроить проверку подлинности. Доступ к ресурсам Azure разрабатываемого приложения должен производиться с помощью соответствующих учетных данных и прав доступа, назначенных в Azure.

Шаг 3. Разработчик должен создать код приложения с использованием пакета SDK. Для работы со службами Azure необходимо создать объект клиента, через которого можно вызывать синхронные и асинхронные методы этого клиентского объекта для взаимодействия со службой.

Шаг 4. Необязательной функцией является настройка ведения журнала для пакета SDK. Требования к приложению могут содержать необходимость диагностики проблем интеграции приложения и служб Azure, что предполагает включение ведения журнала в пакете Azure SDK для .NET.

Проверка подлинности с помощью пакета Azure SDK для приложений .NET

Служба `Azure.Identity` используется сервисами Azure SDK для приложений .NET для проверки подлинности. Это служба является основной по сравнению с другими способами идентификации. Идентификаторы пакетов, поддерживающие учетные данные, предоставленные `Azure.Identity`, создаются поверх `Azure`.

При интеграции приложения со службами Azure требуется строка подключения или ключи для идентификации. Такой подход характерен при запросе ключа из Key Vault, сохранении BLOB-объекта в облачном хранилище.

Строки подключения службы Azure используются при подключении к базам данных SQL и Cosmos DB, для обеспечения доступа к защищенному выделенному кэшу Azure для Redis и Azure Service Bus. Разработчик может использовать различные подходы для получения строки подключения: с помощью портала Azure, CLI, PowerShell или воспользовавшись библиотекой управления Azure для .NET.

Управление ресурсами Azure

При разработке приложения .NET, интегрированного со службами Azure, необходимо предоставить права на чтение и создание ресурсов в подписке Azure.

Процесс формирования необходимых прав доступа предполагает создание экземпляра службы и настройку приложения для его выполнения с учетными данными экземпляра службы, которая обеспечивает заданные права доступа.

Экземпляр службы создает учетную запись, связанную с идентификатором пользователя. Этой учетной записи предоставляются только разрешения, необходимые для запуска приложения.

Для того чтобы убедиться, что используется подписка с заданными правами, необходимо войти в Azure Cloud Shell и выполнить следующую команду.

```
az account show
```

Сведения о подписке отображаются в следующем виде

```
{
  "environmentName": "AzureCloud",
  "id": "15dbcfa8-4b93-4c9a-881c-6189d39f04d4",
  "isDefault": true,
  "name": "my-subscription",
  "state": "Enabled",
  "tenantId": "43413cc1-5886-4711-9804-8cfea3d1c3ee",
  "user": {
    "cloudShellID": true,
    "name": "jane@contoso.com",
    "type": "user"
  }
}
```

Далее необходимо создать экземпляр службы с помощью следующей команды:

```
az ad sp create-for-rbac --sdk-auth
```

Сведения о службе будут отображены в виде JSON-файла.

```
{
  "clientId": "b52dd125-9272-4b21-9862-0be667bdf6dc",
  "clientSecret": "ebc6e170-72b2-4b6f-9de2-99410964d2d0",
  "subscriptionId": "ffa52f27-be12-4cad-b1ea-c2c241b6cceb",
  "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
  "activeDirectoryEndpointUrl": "https://login.microsoftonline.com",
  "resourceManagerEndpointUrl": "https://management.azure.com/",
  "activeDirectoryGraphResourceId": "https://graph.windows.net/",
  "sqlManagementEndpointUrl": "https://management.core.windows.net:8443/",
  "galleryEndpointUrl": "https://gallery.azure.com/",
  "managementEndpointUrl": "https://management.core.windows.net/"
}
```

Сформированные выходные данные JSON со сведениями о службе будут использоваться в дальнейшем в коде приложения.

Процесс аутентификации службы предполагает использование следующих пакетов:

- Microsoft.Azure.Management.Fluent;
- Microsoft.Azure.Management.ResourceManager.Fluent.

Добавление пакетов в проект приложения .NET осуществляется с помощью менеджера NuGet.

```
Install-Package Microsoft.Azure.Management.Fluent
Install-Package Microsoft.Azure.Management.ResourceManager.Fluent
```

После создания экземпляра службы можно выполнить аутентификацию службы для создания и администрирования ресурсов. Аутентификация службы может быть реализована следующими способами:

- с использованием учетных данных токена;
- на основе файла.

Аутентификация с использованием учетных данных токена

Данный способ предполагает создание токена объекта учетных данных в коде. Учетные данные следует безопасно хранить в файле конфигурации, реестре или Azure Key Vault.

```
var credentials = SdkContext.AzureCredentialsFactory
    .FromServicePrincipal(clientId,
        clientSecret,
        tenantId,
        AzureEnvironment.AzureGlobalCloud);
```

При создании экземпляра службы необходимо использовать значения clientId, clientSecret и tenantId из выходных данных JSON.

Затем следуем создать точку входа объекта Azure, чтобы приступить к работе с API:

```
var azure = Microsoft.Azure.Management.Fluent.Azure
    .Configure()
    .Authenticate(credentials)
    .WithDefaultSubscription();
```

В объекте `Azure` рекомендуется явно указать subscriptionId из выходных данных JSON:

```
var azure = Microsoft.Azure.Management.Fluent.Azure
    .Configure()
    .Authenticate(credentials)
    .WithSubscription(subscriptionId);
```

Аутентификация на основе файла

Аутентификация на основе файла позволяет поместить учетные данные субъекта-службы в текстовый файл и защитить его в файловой системе.

Для этого необходимо создать текстовый файл с именем azureauth.json. При создании субъекта-службы необходимо вставть выходные данные JSON.

Далее следует сохранить этот файл в безопасном расположении в проектируемой системе, доступном для кода. При помощи PowerShell следует установить переменную среды с именем AZURE_AUTH_LOCATION и указанием полного пути к файлу, например:

```
[Environment]::SetEnvironmentVariable("AZURE_AUTH_LOCATION", "C:\src\azureauth.json", "User")
```

Для начала работы с API необходимо прочитать содержимое файла и создать точку входа объекта `Azure`:

```
var azure = Microsoft.Azure.Management.Fluent.Azure
    .Configure()
    .Authenticate(credentials)
    .WithDefaultSubscription();
```

Включение ведения журнала с помощью встроенных методов

Класс EventSource используется для трассировки событий Windows (ETW). События регистрируются клиентской частью библиотек сервисов Azure SDK. В журналах формируется структурированное представление событий, при этом обеспечивается минимальное влияние на производительность приложения.

Журналы настраиваются на отслеживание определенных событий путем подписки на конкретные события в коде приложения.

В пакете Azure SDK имеется класс Azure.Core.Diagnostics.AzureEventSourceListener, с помощью которого реализуется формирование журнала событий для приложения .NET. Данный класс содержит два статических метода CreateConsoleLogger и CreateTraceLogger, которые предназначены для формирование логов и трассироваки. Для использования класса Azure.Core.Diagnostics.AzureEventSourceListener необходимо в проект приложения .NET подключить библиотеку Azure.Core с помощью менеджера NuGet.

Ведение журнала в окне консоли. Для упрощения анализа и визуализации журнала событий клиентских библиотек пакета Azure SDK для приложений .NET применяется метод CreateConsoleLogger. Он позволяет отправлять журналы в окно консоли с помощью одной строки кода:

```
using AzureEventSourceListener listener = AzureEventSourceListener.CreateConsoleLogger();
```

Ведение журнала с помощью трассировки диагностики. При подписки на события трассировки используют метод CreateTraceLogger для входа в стандартный механизм трассировки событий приложений .NET (System.Diagnostics.Tracing). В этом примере задается уровень детализации журнала:

```
using AzureEventSourceListener listener = AzureEventSourceListener.CreateTraceLogger(EventLevel.Verbose);
```

Настраиваемое ведение журнала

Для получения информации, необходимой для журнала событий из пакета SDK Azure для приложений .NET, необходимо подписаться на определенные событий. Разработчики могут самостоятельно настроить формирование журнала событий. Для это для этого используется экземпляр класса AzureEventSourceListener и создается собственная функция обратного вызова. Эта функция будет получать сообщения журнала, и их можно обрабатывать так,

как это требуется для мониторинга работы приложения .NET. Кроме того, при создании экземпляра можно указать включаемые уровни сообщения журнала.

В следующем примере создается делегат для отслеживания событий, который отправляет события журнала в консоль с использованием настраиваемого ведения журнала и фильтрует основные события Azure на уровне «Подробный».

```
using AzureEventSourceListener listener = new AzureEventSourceListener((e, message) =>
    {
        // Only log messages from Azure-Core event source
        if (e.EventSource.Name == "Azure-Core")
        {
            Console.WriteLine($"{DateTime.Now} {message}");
        }
    },
    level: EventLevel.Verbose);
```

Сопоставление с ведением журналов ASP.NET Core

Когда вызывается метод расширения AddAzureClients, служба AzureEventSourceLogForwarder регистрируется. Служба AzureEventSourceLogForwarder позволяет использовать стандартную конфигурацию ведения журнала ASP.NET Core.

В таблице 3.1 показано, как пакет Azure SDK для .NET EventLevel сопоставляется с ASP.NET Core LogLevel.

Таблица 3.1 – Сопоставление Azure SDK для .NET EventLevel и ASP.NET Core LogLevel

Пакет SDK Azure EventLevel	**ASP.NET Core LogLevel**
Critical	Critical
Error	Error
Informational	Information
Warning	Warning
Verbose	Debug
LogAlways	Information

В качестве примера приведен вызов AddAzureClients в методе Startup.ConfigureServices проекта ASP.NET Core. Метод AddAzureClients регистрирует клиент Служебной шины Azure и задает учетные данные по умолчанию, используемые для всех клиентов.

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddAzureClients(builder =>
    {
        builder.AddServiceBusClient(Configuration.GetConnectionString("ServiceBus"));
        builder.UseCredential(new DefaultAzureCredential());
    });

    // code omitted for brevity
}
```

В файле *appsettings.json* проекта ASP.NET Core можно изменить уровень ведения журнала по умолчанию для клиентской библиотеки служебной шины Azure. Например, переключите его в режим Debug, задав ключ Logging:LogLevel:Azure.Messaging.ServiceBus следующим образом:

```
{
  "ConnectionStrings": {
    "ServiceBus": "<connection_string>"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Error",
      "Azure.Messaging.ServiceBus": "Debug"
    }
  },
  "AllowedHosts": "*"
}
```

Так как ключ Logging:LogLevel:Azure.Messaging.ServiceBus имеет значение Debug, в журнале будут регистрироваться события клиента служебной шины вплоть до уровня EventLevel.Verbose.

Ключевые термины

Azure Cosmos DB – управляемая служба баз данных нереляционного типа.

Azure Service Bus – сервис для обмена сообщениями между приложениями и устройствами.

Cognitive Services – службы с REST API и пакетами SDK клиентской библиотеки, которые обеспечивают интеграцию когнитивных средств искусственного интеллекта в разрабатываемые приложения.

Event Hubs Azure – сервис для потоковой передачи больших данных и служба приема событий.

Azure Key Vault – служба для безопасного хранения и получения доступа к критическим ресурсам.

App Servie Azure – сервис на базе HTTP для размещения веб-приложений, интерфейсов REST API и серверной части мобильных решений.

`Azure.Identity` – служба, которая используется сервисами Azure SDK для приложений.NET для проверки подлинности.

EventSource – класс, который используется для трассировки событий Windows.

Вопросы для самопроверки

1. Назначение пакета Azure SDK.
2. Назначение сервиса Azure Cosmos DB.
3. Назначение сервиса Azure Service Bus.
4. Назначение сервиса Cognitive Services.
5. Назначение сервиса Event Hubs Azure.
6. Назначение сервиса Azure Key Vault.
7. Назначение сервиса App Servie Azure.
8. Приведите последовательность шагов разработки приложений на платформе .NET совместно с Azure SDK.
9. Способы проверки подлинности с помощью пакета Azure SDK для приложений .NET.
10. Как можно организовать ведение журнала событий при использовании пакета `Azure` SDK.

Литература

1. Azure SDK for .NET overview. https://docs.microsoft.com/ru-ru/dotnet/azure/sdk/azure-sdk-for-dotnet.
2. Introduction to Azure Blob storage. https://docs.microsoft.com/ru-ru/azure/storage/blobs/storage-blobs-introduction.
3. Azure Cosmos DB documentation. https://docs.microsoft.com/ru-ru/azure/cosmos-db/introduction.
4. Event Hubs — A big data streaming platform and event ingestion service https://docs.microsoft.com/ru-ru/azure/service-bus-messaging/service-bus-messaging-overview.
5. What are Azure Cognitive Services?. https://docs.microsoft.com/ru-ru/azure/cognitive-services/what-are-cognitive-services.
6. Azure Event Hubs — A big data streaming platform and event ingestion service.

7. Azure Key Vault basic concepts. https://docs.microsoft.com/ru-ru/azure/key-vault/general/basic-concepts.
8. App Service overview. https://docs.microsoft.com/en-us/azure/app-service/overview.
9. Azure SDK for .NET package index. https://docs.microsoft.com/ru-ru/dotnet/azure/sdk/packages.
10. Authenticate with the Azure SDK for .NET. https://docs.microsoft.com/en-us/dotnet/azure/sdk/authentication.

Лекция 4. Платформа Microsoft .Net Services

Краткая аннотация лекции

В рамках данной лекции будут рассмотрены следующие вопросы: виртуальные машины платформы Microsoft Azure, модели виртуальных машин, такие компоненты виртуальных машин как виртуальная сеть, IP-адрес, балансировщик нагрузки, сетевой адаптер, группы безопасности и доступности.

Цель лекции

Целью данной лекции является ознакомление с виртуальными машинами платформы Microsoft Azure

Введение

Платформа как услуга (PaaS) хорошо подходит для развертывания рабочих нагрузок определенного типа. Однако модель PaaS подходит не для всех решений, и это совершенно нормально. При использовании некоторых рабочих нагрузок необходимо контролировать практически все аспекты инфраструктуры: конфигурацию операционной системы, создание копии диска, возможность устанавливать и конфигурировать традиционное серверное программное обеспечение и т. д. Для решения этих задач используется подход «инфраструктура как услуга» (IaaS) и виртуальные машины Azure.

Виртуальные машины

Виртуальные машины Azure – одна из ключевых IaaS-возможностей Azure [1]. Виртуальные машины Azure можно устанавливать под управлением операционных систем Windows или Linux. Конфигурацией виртуальной машины полностью управляет администратор информационной системы. Установка, конфигурирование и обслуживание всего серверного программного обеспечения и исправлений операционной системы в данном случае являются задачей администратора.

Вычислительные возможности Azure PaaS и IaaS различаются в двух отношениях: сохраняемость и возможности управления. Управление такими компонентами PaaS, как облачные службы (т. е. веб-роли и рабочие роли) и службы приложений, практически полностью берет на себя платформа Azure, что позволяет уделять меньше внимания управлению серверной инфраструктурой в пользу разработки приложений. При работе с виртуальными машинами Azure настройка практически всех характеристик виртуальных машин является задачей администратора информационной системы.

В виртуальных машинах Azure поддерживаются два типа устойчивых (сохраняемых) дисков: диски ОС и диски с данными. Диск ОС требуется для работы виртуальной машины, диски с данными используются по необходимости. Устойчивость дисков обеспечивается хранилищем Azure. На диске ОС размещается операционная система (Windows или Linux), а на диск с данными можно поместить что-то еще – данные приложений, изображения и т. п. В облачных PaaS-службах Azure используется совершенно другой подход – несохраняемые диски, подключенные к физическому узлу, данные на которых могут быть утеряны в случае сбоя физического узла.

Доступные пользователю возможности управления и использования устойчивых дисков делают виртуальные машины идеальным вариантом для размещения множества серверных рабочих нагрузок, которые не соответствуют модели PaaS. Такой подход позволяет запускать серверы баз данных (SQL Server, Oracle, MongoDB и т. п.), Windows Server Active Directory, Microsoft SharePoint и многие другие нагрузки на платформе Microsoft Azure. При необходимости пользователи могут мигрировать такие рабочие нагрузки из локального центра обработки данных в один или несколько регионов Azure.

Модели виртуальных машин

Существует две модели работы со многими ресурсами Azure: диспетчер ресурсов (ДР) Azure и управление службами Azure [2]. Новое развертывание рекомендуется осуществлять с помощью диспетчера ресурсов. Классическая модель по-прежнему поддерживается, однако полный набор возможностей доступен только при использовании ДР Azure.

Модель с использованием ДР Azure предоставляет полные и тонкие возможности управления почти всеми характеристиками виртуальных машин Azure. Можно явным образом добавлять различные компоненты: сетевой адаптер, общедоступный IP-адрес, диски с данными, балансировщик нагрузки и многие другие. Для обеспечения доступа к ресурсам Azure и возможностей управления ими ДР использует различные поставщики ресурсов. При работе с виртуальными машинами Azure ключевую роль играют три поставщика ресурсов: Сеть, Служба хранилища и Вычисление.

Поставщик ресурсов «Сеть» (Microsoft.Network) управляет всеми аспектами сетевых соединений: IP-адресами, балансировщиками нагрузки, сетевыми адаптерами и т. д.

Поставщик ресурсов «Служба хранилища» (Microsoft.Storage) контролирует хранение дисков виртуальных машин (если мы говорим о виртуальных машинах Azure).

Поставщик ресурсов «Вычисление» (Microsoft.Compute) управляет характеристиками самих виртуальных машин: имена, параметры операционных систем и конфигурации (размер, количество дисков и т. д.).

Пользователю-администратору доступны не только прямые средства управления компонентами виртуальной машины, но и другие возможности ДР, например:

- развертывание логически связанных ресурсов и управление ими в составе групп ресурсов;
- теги для упорядочения и идентификации ресурсов;
- управление доступом на основе ролей (RBAC), позволяющее применять необходимые политики безопасности и контроля;
- декларативные файлы шаблонов;
- политики развертывания, обеспечивающие действие определенных правил организации;
- согласованный и централизованно управляемый (orchestrated) процесс развертывания.

Эта возможность позволяет тонко настроить среду под ваши конкретные потребности.

В рамках классической модели виртуальные машины всегда развертываются в контексте облачной службы Azure — контейнера виртуальных машин. Этот контейнер обеспечивает ряд важных возможностей: конечная точка DNS, сетевые подключения (в том числе, при необходимости, из общедоступного Интернета), безопасность, управление. Можно получить все эти возможности бесплатно (поскольку они унаследованы от модели облачных служб), но возможности управлять ими ограничены. Кроме того, в рамках классической модели недоступны дополнительные полезные функции ДР Azure (теги, файлы шаблонов и другие).

Компоненты виртуальных машин

Виртуальная машина, как и обычный компьютер, состоит из ряда компонентов, и ее также можно настроить множеством различных способов в соответствии с потребностями и желаниями владельца.

Виртуальную машину Azure иногда полезно представлять, как логическую сущность. Виртуальная машина характеризуется набором атрибутов: статус, параметры конфигурации (операционная система, процессорные ядра, память, диски, IP-адреса и т. п.) и состояние. В Azure может быть создан экземпляр этой логической сущности и выделены необходимые ресурсы, чтобы эта виртуальная машина заработала.

Данные виртуальных машин Azure хранятся на устойчивых подключенных дисках VHD (Virtual Hard Disc) [3]. Для виртуальных машин Azure доступно два типа VHD:

- *Образ* (*Image*) VHD этого типа является шаблоном для создания новой виртуальной машины, поэтому часть параметров (например, имя

машины, пользователь с правами администратора и т. п.) к таким дискам неприменима;

– *Диск* (*Disk*) VHD (возможно, загрузочный), который можно использовать в качестве подключаемого диска виртуальной машины. Диски делятся на два типа: диск ОС и диск с данными.

Для хранения устойчивых дисков (дисков ОС и дисков с данными) используются страничные BLOB-объекты в хранилище Azure. Поэтому все преимущества хранилища BLOB-объектов (высокая доступность, устойчивость, возможности обеспечения географической избыточности) относятся и к дискам. Хранилище BLOB-объектов обеспечивает механизм безопасного хранения данных, используемых виртуальной машиной. Диски можно подключать к виртуальной машине в качестве дисковых устройств. Платформа Azure использует постоянную аренду страничного BLOB-объекта, чтобы предотвратить случайное удаление страничного BLOB-объекта, который содержит VHD, соответствующий контейнер или учетную запись хранения.

Хранилища классов Standard и Premium. Для надежного хранения файлов дисков (файлов .vhd) можно использовать учетные записи хранения Azure классов Standard или Premium [4]. В хранилище Azure класса Premium используются твердотельные накопители (SSD), обеспечивающие высокую производительность и низкую задержку, что особенно важно для виртуальных машин, на которых запущены рабочие нагрузки, интенсивно считывающие или записывающие данные. Хранилище класса Standard доступно для виртуальных машин всех размеров; хранилище класса Premium доступно для виртуальных машин серий DS, DSv2, F и GS [5]. Хранилище класса Standard можно использовать также для виртуальных машин серий DS, DSv2, F и GS. В этом случае на твердотельном накопителе (SSD) размещается только локальный (несохраняемый) диск.

В общем случае для нагрузок в рабочей среде (особенно чувствительных к колебаниям производительности или активно выполняющим операции ввода-вывода) рекомендуется использовать хранилище Azure класса Premium. Рабочие нагрузки, предназначенные для разработки или тестирования, часто нечувствительны к колебаниям производительности и не осуществляют операции ввода-вывода слишком активно, поэтому для них рекомендуется использовать хранилище Azure класса Standard.

Диск ОС служит для размещения операционной системы. В случае виртуальной машины Windows диск ОС — это обычный диск C, на котором Windows размещает свои данные. Для виртуальной машины Linux это диск раздела /dev/sda1, в котором находится корневой каталог. Максимальный размер диска ОС в настоящее время составляет 1023 ГБ. Второй тип дисков, используемых в виртуальных машинах Azure, называется «диск с данными». Эти диски служат для хранения самых различных данных. Максимальный размер диска с данными также составляет 1023 ГБ. К виртуальной машине Azure можно

подключить несколько дисков с данными. Их максимальное количество зависит от размера виртуальной машины и обычно составляет два диска на процессор. Диски с данными часто используются для хранения данных приложений (в том числе созданных клиентами) или серверного программного обеспечения (например, Microsoft SQL Server), а также соответствующих данных и файлов журналов. Несколько дисков с данными можно преобразовать в дисковый массив с помощью инструмента «Дисковые пространства» (Storage Spaces) в Windows или утилиты mdadm в Linux.

Кроме того, в виртуальных машинах Azure используется временный диск на физическом узле, который не сохраняется в хранилище Azure. Этот временный диск представляет собой физический диск, размещенный в корпусе сервера. Временный жесткий диск может быть стандартным жестким диском или твердотельным накопителем (SSD), в зависимости от типа созданной виртуальной машины. Временный диск следует использовать только для размещения временных (или реплицированных) данных, потому что в случае сбоя физического узла или при остановке/освобождении виртуальной машины его содержимое будет удалено. На рис. 4.1 показаны различные типы дисков.

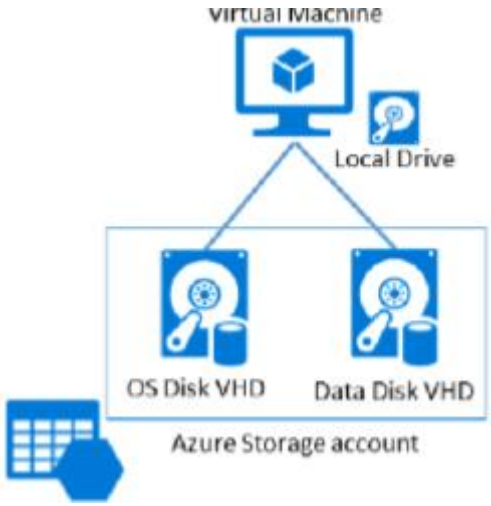

Рисунок 4.1 – Типы дисков в виртуальных машинах Azure.

Виртуальная сеть

В локальной физической инфраструктуре может содержаться множество компонентов, которые позволяют использовать масштабируемые и безопасные методы работы с виртуальными машинами. Вот некоторые из таких ресурсов: отдельные сетевые пространства для серверов, взаимодействующих с Интернетом, и для служебных серверов, балансировщики нагрузки, брандмауэры и многое другое. Многие из этих компонентов можно логически развернуть в виртуальной сети Azure (часто такие сети называют VNET) [6].

Виртуальная сеть Azure поддерживает многие аналогичные функции, например:

- *подсеть* (Subnet) – диапазон IP-адресов, относящихся к виртуальной сети. Виртуальную машину необходимо разместить в подсети, входящей в VNET. Виртуальные машины, размещенные в некоторой подсети VNET, могут свободно обмениваться данными с виртуальными машинами из другой подсети той же виртуальной сети. Вы можете управлять таким взаимодействием с помощью групп безопасности сети (NSG) и настраиваемых маршрутов;
- *IP-адрес* (IP-address), которые бывают общедоступными и частными. Общедоступный IP-адрес позволяет виртуальной машине принимать данные из Интернета. Такой адрес может выделяться динамически, то есть создаваться при запуске соответствующего ресурса (например, виртуальной машины или балансировщика нагрузки) и освобождаться при его остановке, либо статически, то есть назначаться немедленно и сохраняться до удаления ресурса. Частными IP-адресами называются адреса, не маршрутизируемые в сети Интернет. Они служат для обмена данными между виртуальными машинами и балансировщиками нагрузки в рамках одной сети VNET;
- *балансировщик нагрузки* (Load Balancer) Доступ к виртуальным машинам со стороны узлов Интернета или других виртуальных машин в составе сети VNET обеспечивается балансировщиками нагрузки Azure. Существует два типа балансировщиков нагрузки:
    - *внешний балансировщик нагрузки* (External Load Balancer), который используется для обеспечения высокой доступности нескольких виртуальных машин для узлов Интернета;
    - *внутренний балансировщик нагрузки* (Internal Load Balancer), который используется для обеспечения высокой доступности нескольких виртуальных машин для других виртуальных машин той же сети VNET.
- *группа безопасности сети* (Network Security Group), которая позволяет создавать правила, которые управляют входящим и исходящим сетевым трафиком (разрешают или отклоняют его) для сетевых адаптеров виртуальной машины или подсетей.

При создании виртуальной машины Azure с помощью ДР необходимо поместить ее в виртуальную сеть Azure (VNET). Администратор сам решает, использовать ли существующую сеть VNET или создать новую, в какой подсети разместить машину, нужен ли балансировщик нагрузки, а также выбирает IP-адрес, количество сетевых адаптеров и способ обеспечения безопасности сети (см. рис. 4.2). Может показаться, что это сильно осложняет развертывание виртуальной машины, однако все эти параметры очень важны для ее доступности и безопасности.

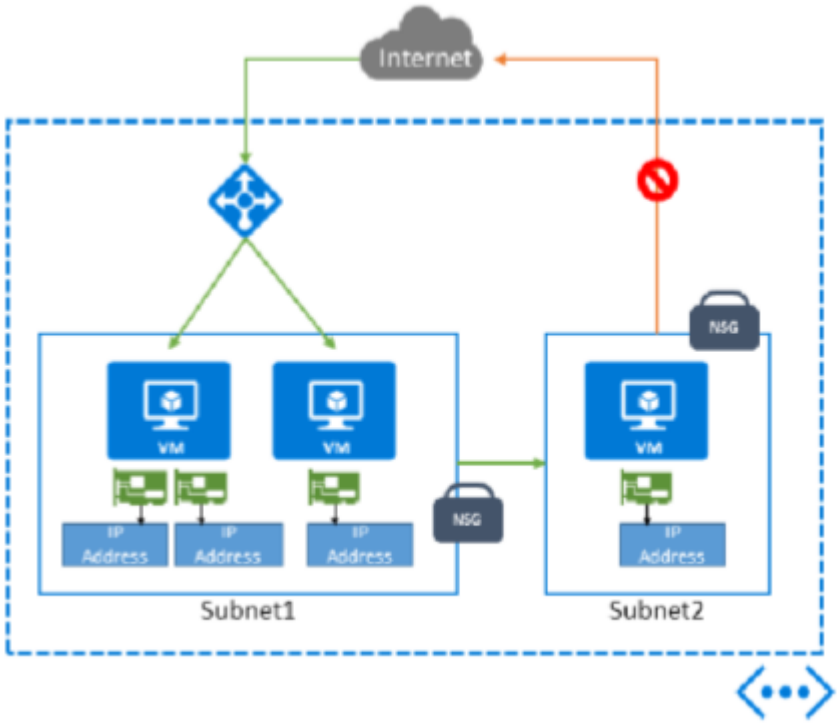

Рисунок 4.2 – Виртуальные машины в рамках модели с использованием ДР явным образом управляют соответствующими сетевыми компонентами.

Классические виртуальные машины также можно поместить в виртуальную сеть Azure, но это необязательное требование (тогда как для виртуальных машин в модели с использованием ДР — обязательное).

IP-адрес

В модели с использованием ДР у виртуальной машины по умолчанию нет IP-адреса. IP-адрес необходимо назначить виртуальной машине явным образом через подключенный к ней сетевой адаптер. Чтобы обмениваться данными с другими виртуальными машинами в виртуальной сети или с узлами общедоступного интернета, виртуальной машине нужен IP-адрес.

Каждому сетевому адаптеру соответствует частный адрес (его часто называют DIP или динамическим IP). Он служит для подключения к виртуальной сети и может быть сопоставлен с общедоступным IP-адресом, который делает возможным прямое подключение к общедоступному интернету. По умолчанию при остановке/освобождении виртуальной машины эти динамические IP-адреса сбрасываются, однако и виртуальную машину, и адрес можно сделать статическими, чтобы они сохранялись и после отключения/освобождения виртуальной машины. Это удобно в том случае, если виртуальной машине необходим постоянный DIP-адрес (примеры: виртуальные машины Microsoft SQL Server, виртуальные машины, используемые в качестве серверов DNS, и

постоянные общедоступные IP-адреса). Если требуется назначить виртуальной машине несколько DIP-адресов (например, чтобы разместить ее в нескольких подсетях), то к ней можно подключить несколько сетевых адаптеров с различными DIP-адресами.

В классической модели все примерно так же, с одним отличием: сетевые адаптеры и общедоступные IP-адреса не являются независимыми ресурсами — они могут существовать только в контексте виртуальной машины. Более того, в классической модели подключение к интернету обычно осуществляется не через общедоступный IP-адрес, а посредством балансировщика нагрузки Azure.

Балансировщик нагрузки Azure

Балансировщик нагрузки Azure используется для того, чтобы обеспечить примерно равное распределение трафика между несколькими виртуальными машинами (эти машины часто сконфигурированы похожим образом или связаны между собой логически). Балансировщик нагрузки позволяет обеспечить взаимодействие нескольких виртуальных машин — например, в коллекции веб серверов в среде веб-фермы. Запросы, поступающие для набора виртуальных машин с балансировкой нагрузки, не направляются одной конкретной виртуальной машине, а распределяются между доступными виртуальными машинами.

В Azure доступно два типа балансировщиков нагрузки: внешний и внутренний (рис. 4.3). Внешний балансировщик нагрузки служит для управления трафиком из Интернета (позволяя направить его одной виртуальной машине или распределить между несколькими). С его помощью можно обеспечить высокую доступность приложения и при необходимости быстро масштабировать среду.

Внутренний балансировщик нагрузки служит для распределения трафика, поступающего из виртуальной сети на набор виртуальных машин. Это может быть, например, трафик для веб-API или кластера баз данных, который должен быть доступен только для веб-серверов переднего плана, но не для всех узлов общедоступного Интернета.

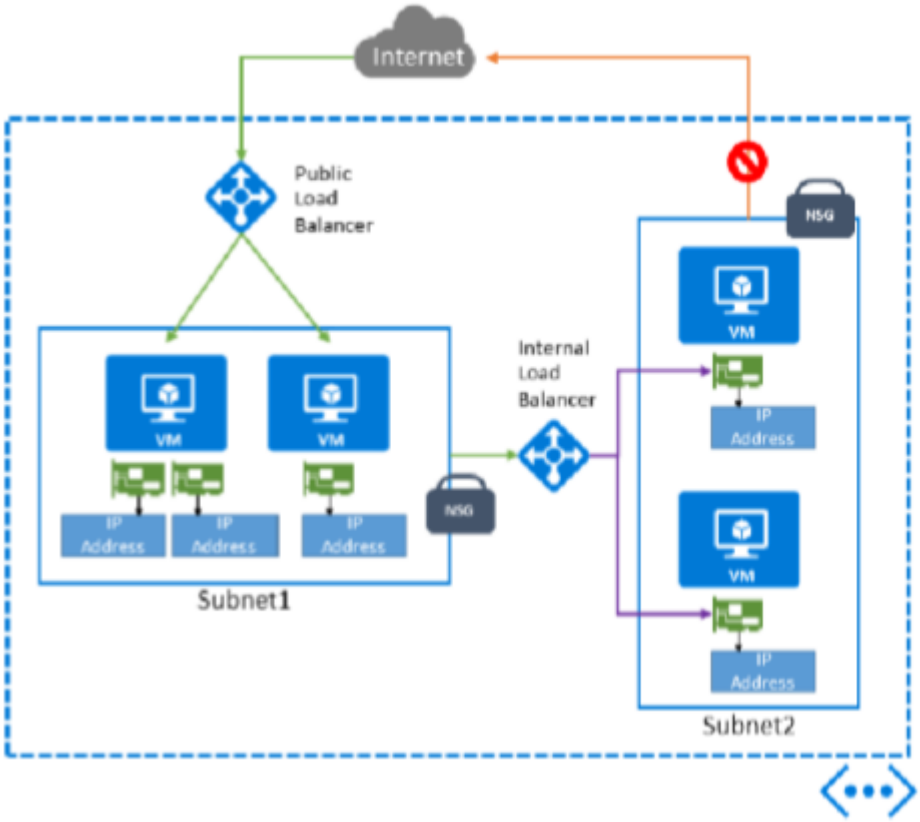

Рисунок 4.3 – Использование внешнего и внутреннего балансировщика нагрузки.

В модели развертывания с помощью Диспетчера Ресурсов Azure перед тем, как использовать балансировщик нагрузки, необходимо создать несколько дополнительных объектов:

- общедоступные IP-адреса для входящего сетевого трафика (в случае внешнего балансировщика нагрузки);
- пул служебных (частных) IP-адресов, назначенных сетевым адаптерам виртуальных машин;
- правила, определяющие соответствие между общедоступными портами балансировщика нагрузки и портами служебного пула;
- правила NAT для входящих соединений, определяющие соответствие между общедоступными портами балансировщика нагрузки и конкретными виртуальными машинами в пуле;
- проверки работоспособности, позволяющие контролировать функциональность виртуальных машин в составе пула.

В классической модели внешний балансировщик нагрузки предоставляется автоматически в рамках модели облачных служб. Все виртуальные машины, которые относятся к облачной службе и открывают конечную точку, доступную для подключения через Интернет, автоматически конфигурируются так, чтобы использовать балансировщик нагрузки.

Классические виртуальные машины также могут использовать внутренний балансировщик нагрузки.

Сетевой адаптер

Сетевой адаптер обеспечивает доступ к ресурсам в виртуальной сети Azure через сеть [8]. Сетевой адаптер является самостоятельным ресурсом, но для обеспечения доступа к сети его необходимо сопоставить с виртуальной машиной. Максимальное количество сетевых адаптеров, которые можно подключить к виртуальной машине, зависит от размера выбранной виртуальной машины.

При работе с сетевыми адаптерами и виртуальными машинами следует помнить несколько важных вещей:

- IP-адрес каждого сетевого адаптера виртуальной машины должен принадлежать подсети VNET, к которой относится виртуальная машина;
- если одной виртуальной машине назначено несколько сетевых адаптеров, то назначить общедоступный IP-адрес можно только основному адаптеру. Каждому сетевому адаптеру назначается частный IP-адрес (кроме ситуации, когда сетевой адаптер является основным и имеет общедоступный IP-адрес). Сетевые адаптеры могут относиться к различным подсетям в составе сети VNET;
- любой сетевой адаптер виртуальной машины можно добавить в группу безопасности сети (NSG).

При работе с классическими виртуальными машинами беспокоиться о конфигурации сетевых адаптеров не нужно, потому что она создается автоматически в рамках модели облачной службы и не может существовать вне контекста виртуальной машины.

Группы безопасности сети

Группы безопасности сети (NSG) позволяют явным образом задать детальные правила, контролирующие потоки входящего и исходящего сетевого трафика виртуальных машин и подсетей Azure [9]. Группы безопасности сети (NSG) позволяют управлять потоками сетевого трафика, входящего в вашу среду и исходящими из нее. Вы создаете правила, в которых указывается IP-адрес и порт отправителя и получателя. Правила групп безопасности сети (NSG) могут применяться к виртуальным машинам и (или) к подсетям. В случае с виртуальной машиной группа безопасности сети (NSG) ассоциируется с сетевым адаптером, подключенным к этой виртуальной машине.

Группы доступности

Виртуальные машины Azure размещаются на физических серверах, которые находятся в центрах обработки данных Microsoft Azure. Они, как и любые другие физические устройства, могут ломаться. При сбое физического сервера виртуальные машины Azure, размещенные на этом сервере, также перестанут работать. В случае сбоя платформа Azure перенесет виртуальные машины на работоспособный узел и запустит их. Восстановление работоспособности служб может занять несколько минут. В течение этого периода приложения, размещенные на этих виртуальных машинах, будут недоступны.

Помимо аппаратных сбоев на функционирование виртуальных машин также могут влиять периодические обновления, которые инициирует сама платформа Azure. Корпорация Microsoft периодически обновляет операционную систему узлов, на которых выполняются виртуальные машины (однако установка исправлений ОС для гостевых виртуальных машин, которые вы создаете, остается вашей задачей). В ходе этих обновлений виртуальные машины перезагружаются, и поэтому они недоступны некоторое время.

Для того чтобы в вашей инфраструктуре не присутствовала единая точка отказа, рекомендуется развернуть несколько экземпляров виртуальной машины. Более того, соглашение об уровне обслуживания (SLA) действует лишь в том случае, если в группе доступности [10] развернуто не менее двух виртуальных машин Azure. Это логическая функция, которая используется для того, чтобы гарантировать, что группа связанных между собой виртуальных машин развертывается таким образом, чтобы исключить их одновременный сбой или недоступность ввиду обновления операционной системы в центре обработки данных. Первые две виртуальные машины, развернутые в группе доступности, помещаются в два различных домена сбоя (fault domains). Благодаря этому сбой в каком-либо центре данных не затронет обе машины одновременно. Аналогичным образом первые пять виртуальных машин, развернутые в группе доступности, помещаются в пять различных доменов обновления (update domains), что минимизирует влияние обновления операционных систем узлов в Azure на работоспособность виртуальных машин (обновление выполняется в различных доменах поочередно). В одну группу доступности следует помещать виртуальные машины, выполняющие одинаковые функции.

Количество доменов сбоя и доменов обновления зависит от модели развертывания (с помощью ДР или классической модели). В модели с использованием ДР может использоваться до 3 доменов сбоя и до 20 доменов обновления. В классической модели может использоваться 2 домена сбоя и 5 доменов обновления.

Ключевые термины:

Служба «Виртуальные машины Azure» – служба, которая позволяет разворачивать виртуальные машины под управлением Windows или Linux в центре обработки данных Microsoft Azure.

Microsoft.Network – поставщик ресурсов, который управляет всеми аспектами сетевых соединений: IP-адресами, балансировщиками нагрузки, сетевыми адаптерами.

Microsoft.Storage – поставщик ресурсов, который контролирует хранение дисков виртуальных машин.

Microsoft.Compute – поставщик ресурсов, который управляет всеми характеристиками самих виртуальных машин: имена, параметры операционных систем и конфигурации.

VHD (Virtual Hard Disc) – диск, на котором хранятся данные виртуальных машин Azure.

Балансировщик нагрузки Azure – устройство для обеспечения равномерного распределения трафика между несколькими виртуальными машинами

Сетевой адаптер – ресурс, который обеспечивает доступ к другим ресурсам в виртуальной сети Azure через сеть.

Группы безопасности сети (NSG) – ресурс, который позволяют явным образом задать детальные правила, контролирующие потоки входящего и исходящего сетевого трафика виртуальных машин и подсетей Azure.

Группа доступности – это логическая группа виртуальных машин, которая позволяет Azure понять структуру вашего приложения, чтобы обеспечить избыточность и доступность.

Вопросы для самопроверки

1. Виртуальные машины Azure. Статус виртуальной машины
2. Модели и компоненты виртуальных машин.
3. Какие операционные системы можно установить на виртуальные машины Azure?
4. Назовите основные поставщики ресурсов для виртуальных машин Azure.
5. Какие существуют модели работы со многими ресурсами Azure?
6. Назначение виртуальной сети Azure.
7. Назначение балансировщика нагрузки Azure.
8. Назначение группы безопасности сети Azure.
9. Назначение группы доступности сети Azure.
10. Назначение поставщика ресурсов «Сеть» для виртуальных машин Azure.

11. Назначение поставщика ресурсов «*Служба хранилища*» для виртуальных машин Azure.
12. Назначение поставщика ресурсов «*Вычисление*» для виртуальных машин Azure.

Литература

1. Виртуальные машины Windows в Azure. https://docs.microsoft.com/ru-ru/azure/virtual-machines/windows/overview.
2. Azure Resource Manager. https://docs.microsoft.com/ru-ru/azure/azure-resource-manager/management/overview.
3. О виртуальных жестких дисках. https://docs.microsoft.com/ru-ru/windows/win32/vstor/about-vhd
4. Типы управляемых дисков Azure. https://docs.microsoft.com/ru-ru/azure/virtual-machines/disks-types#premium-ssd.
5. Размеры виртуальных машин в Azure. https://docs.microsoft.com/ru-ru/azure/virtual-machines/sizes.
6. Что такое виртуальная сеть Azure? https://docs.microsoft.com/ru-ru/azure/virtual-network/virtual-networks-overview
7. Что такое Azure Load Balancer? https://docs.microsoft.com/ru-ru/azure/load-balancer/load-balancer-overview.
8. Подключение отдельных серверов с помощью сетевого адаптера Azure. https://docs.microsoft.com/ru-ru/azure/architecture/hybrid/azure-network-adapter
9. Группы безопасности сети. https://docs.microsoft.com/ru-ru/azure/virtual-network/network-security-groups-overview
10. Обзор групп доступности. https://docs.microsoft.com/ru-ru/azure/virtual-machines/availability-set-overview

Лекция 5. Введение в SQL Azure

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: назначение служб Microsoft SQL Azure, описание сервисов База данных SQL Azure, Управляемый экземпляр SQL Azure, SQL Server на виртуальных машинах Azure, администрирование служб SQL Azure, основные параметры Соглашения об уровне обслуживания и создание отдельной базы данных в сервисе База данных SQL Azure.

Цель лекции:

Цель данной лекции – получить предварительные сведения о слyджбе Windows SQL Azure.

Введение

Служба Microsoft SQL Azure представляет собой пакет облачных продуктов, включающих [1]:

- базу данных SQL Azure;
- управляемый экземпляр SQL Azure;
- SQL Server на виртуальных машинах Azure.

База данных SQL Azure предназначена для применения в облачных приложениях с использованием интеллектуальной управляемой службы данных.

Управляемый экземпляр SQL Azure предоставляется в виде интеллектуального сервиса, возможности которого аналогичны возможностям ядра локальной СУБД SQL Server. Данный сервис рекомендуется использовать для миграции СУБД в облачную среду.

SQL Server на виртуальных машинах Azure поддерживает совместимость с СУБД SQL Server и обеспечивает доступ пользователей на уровне операционной системы.

Основные возможности СУБД SQL Server были использованы при создании сервиса SQL Azure. Данный подход определил преемственность в подходах к работе с языками и ресурсами системы управления базами данных.

База данных SQL Azure

Сервис «*База данных SQL Azure*» (БД SQL Azure) представляет собой облачный сервис, который предоставляется как услуга. БД SQL Azure является реляционной базой данных [2].

Она может применяться в современных облачных приложениях, которые предполагают использование стабильных возможностей Microsoft SQL Server. База данных имеет управляемое ядро СУБД SQL Server, функционал которого поддерживается в соответствии с последними обновлениями корпоративного выпуска SQL Server.

В SQL Azure, по сравнению с SQL Server, добавлены такие возможности как встроенная высокая доступность, аналитика и управление. База данных имеет широкие возможности масштабирования для повышения производительности без прерывания работы.

Для пользователя предоставляется возможность развертывания БД SQL Azure в двух вариантах:

- в первом варианте реализуется развертывание базы данных как отдельного экземпляра, который обладает собственным набором ресурсов. В этом случае управление реализуется с помощью логического сервера SQL. Развернутая таким способом база данных, может рассматриваться как автономная база данных в традиционном корпоративном SQL Server;
- второй вариант представляет собой коллекцию баз данных с общим набором ресурсов (*эластичный пул*), которые управляются с помощью логического сервера SQL. Эластичность базы данных проявляется в том, что отдельные базы данных можно добавлять в эластичный пул и удалять из него. Такой вариант развертывания SQL Azure целесообразно использовать для обеспечения мультитенантных возможностей приложений SaaS. Эластичные пулы позволяют оптимизировать процесс управления ресурсами информационных систем при наличии множества баз данных с разной динамикой использования.

Управляемый экземпляр SQL Azure

Сервис *«Управляемый экземпляр SQL Azure»* (УЭ SQL Azure) представляет собой коллекцию системных и пользовательских баз данных, которые имеют общий набор ресурсов, и предоставляются как облачный сервис по модели PaaS [3].

Данный сервис целесообразно использовать для разрабатываемых приложений, которые базируются на проверенных возможностях SQL Server, а перенос функциональности в облако производится с минимальными изменениями. Следует отметить, что сервис аналогичен экземпляру ядра СУБД Microsoft SQL Server. Это касается возможностей общих ресурсов для баз данных и дополнительные функции для экземпляра.

При использовании сервиса УЭ SQL Azure обеспечивается упрощение процесса переноса базы данных из локальной среды в облако с минимальными изменениями базы данных.

УЭ SQL Azure предоставляет все преимущества облачной модели PaaS в части управления ресурсами для базы данных SQL Azure. Кроме того, обеспечивает возможности, которые были доступны только на виртуальных машинах SQL Server: собственная виртуальная сеть; высокая совместимость с SQL Server локальной среды. Сервис предоставляет доступ к SQL Server и обеспечивает совместимость функций для миграции серверов SQL Server в Azure.

SQL Server на виртуальной машине Azure

Сервис «*SQL Server на виртуальной машине Azure*» (SQL Server VM Azure) обеспечивает развертывание SQL Server на полностью управляемой виртуальной машине в облаке [4]. Данный сервис предоставляется по модели IaaS и обеспечивает возможность развертывания SQL Server на виртуальных машинах Windows Server или Linux в облаке.

С помощью данного сервиса упрощается процесс миграции баз данных и приложений, требующий доступа на уровне ОС. Это определяется тем, что виртуальные машины SQL предоставляют полный административный контроль для экземпляра SQL Server и базовой ОС при миграции в облако.

Сервис SQL Server VM Azure отличается от сервисов БД SQL Azure и УЭ SQL Azure тем, что предоставляет полный контроль над ядром системы управления базами данных.

Администратор может определять время запуска для обслуживания и внесения исправлений, изменять модель восстановления на простую или с неполным протоколированием, приостанавливать или запускать службы при необходимости или настроить любые параметры ядра базы данных SQL Server. Администратор информационной системы может останавливать или возобновлять работу виртуальной машины при необходимости.

Предоставляемые администратору возможности контроля требуют выполнения работ по управлению виртуальными машинами.

При использовании сервиса SQL Server VM Azure у разработчика информационной системы появляется возможность быстрой разработки и тестирования базы данных без покупки оборудования для дополнительного локального сервера SQL Server.

Для сервиса SQL Server VM Azure виртуальные машины SQL устанавливаются в дата-центрах Майкрософт и вопросы их обслуживания возлагаются на персонал дата-центров.

Для виртуальных машин SQL можно использовать включенную в образ SQL Server лицензию с оплатой по мере использования или уже имеющуюся у вас лицензию.

Сервис SQL Server VM Azure обеспечивает упрощение процесса переноса имеющихся у пользователя приложений в облако или гибридного использования локальных и облачных приложений.

Сервис позволяет проводить разработку и тестирование традиционных приложений для СУБД SQL Server.

При использовании виртуальных машин SQL в процессе проектирования информационных систем, разработчик получает права администратора в выделенном экземпляре SQL Server и облачной виртуальной машине. Это позволяет персонализировать информационную систему с учетом бизнес-требований конкретного приложения к производительности и доступности.

В табл. 5.1 приведены дополнительные различия между БД SQL Azure, УЭ SQL Azure и SQL Server VM Azure. В тоже время сервисы БД SQL Azure, УЭ SQL Azure обеспечивают снижение общих затрат при подготовке нескольких баз данных и управление ими. Текущие затраты администрирования для информационной системы снижается, за счет того, отпадает необходимость в обслуживании виртуальных машин, операционной системы или программного обеспечения для базы данных.

В целом сервисы БД SQL Azure и УЭ SQL Azure могут повысить эффективность работы ИТ подразделения предприятия за счет сокращения затрат на администрирования баз данных. Эластичные пулы также поддерживают мультитенантные архитектуры для приложений SaaS, включая изоляцию клиентов и возможность масштабирования для сокращения затрат благодаря совместному использованию ресурсов в базах данных. Сервис УЭ SQL Azure обеспечивает возможности на уровне экземпляров баз данных для миграции существующих приложений, а также для совместного использования ресурсов в базах данных. Кроме того, сервис SQL Server VM Azure предоставляет администраторам возможности обслуживания баз данных, применяя навыки, полученные при работе с локальной средой.

Таблица 5.1 – Сравнение сервисов БД SQL Azure, УЭ SQL Azure и SQL Server VM Azure

База данных SQL Azure	**Управляемый экземпляр SQL Azure**	**SQL Server на виртуальной машине Azure**
Поддерживает большинство возможностей уровня базы данных в локальной среде. Доступны наиболее часто используемые функции SQL Server. Доступность гарантируется на уровне 99,995 %. Встроенное резервное копирование, исправления и восстановление. Последняя стабильная версия ядра СУБД. Возможность назначать необходимые ресурсы (ЦП/хранилище) для отдельных баз данных. Встроенные расширенные функции аналитики и безопасности. Изменение ресурсов в режиме онлайн (ЦП/хранилище).	Поддерживает почти все возможности уровня экземпляра и уровня базы данных в локальной среде. Высокая совместимость с SQL Server. Гарантия доступности 99,99 %. Встроенное резервное копирование, исправления и восстановление. Последняя стабильная версия ядра СУБД. Простой переход с SQL Server. Частный IP-адрес в виртуальной сети Azure. Встроенные расширенные функции аналитики и безопасности. Изменение ресурсов в режиме онлайн (ЦП/хранилище).	У вас есть полный контроль над системой SQL Server. Поддерживает все возможности в локальной среде. Доступность на уровне до 99,99 %. Полное равенство с соответствующей версии локального SQL Server. Исправленная и хорошо известная версия ядра СУБД. Простой переход с SQL Server. Частный IP-адрес в виртуальной сети Azure. У вас есть возможность развертывать приложения или службы на узле, где размещается SQL Server.
Переход с SQL Server может быть сложным. Некоторые функции SQL Server недоступны. Нет гарантированного точного времени обслуживания (но почти полная прозрачность). Совместимость с версией SQL Server может осуществляться только при использовании режима совместимости базы данных. Поддержка частных IP-адресов с использованием Приватного канала Azure.	Некоторое количество компонентов SQL Server еще не доступно. Нет гарантированного точного времени обслуживания (но почти полная прозрачность). Совместимость с версией SQL Server может осуществляться только при использовании режима совместимости базы данных.	Необходимо управлять резервным копированием и исправлениями. Необходимо реализовать собственное решение высокой доступности. Возникает простой при изменении ресурсов (ЦП/хранилище)
Базы данных размером до 100 ТБ.	До 16 ТБ.	Экземпляры SQL Server с хранилищем объемом до 256 ТБ. Экземпляр может поддерживать любое необходимое количество баз данных.
Локальное приложение получает доступ к данным в Базе данных SQL Azure.	Собственная реализованная виртуальная сеть и подключение к локальной среде с помощью Azure Express Route или VPN-шлюза.	ВМ SQL позволяют создавать приложения, которые частично работают в облаке и частично – на локальных ресурсах.

Администрирование

При переходе на использование сервисов SQL Azure, как правило, сокращаются затраты компаний на администрирования, а также сложность администрирования информационной системы в целом.

Модели предоставления облачных сервисов IaaS и PaaS в рамках платформы Azure, обеспечивают для пользователей системы управление базовой инфраструктурой, автоматическую репликацию данных, аварийное восстановление, настройку и обновление программного обеспечения базы данных, управление балансировкой нагрузки, а также прозрачную отработку отказа при сбое сервера в центре обработки данных.

Сервисы БД SQL Azure и УЭ SQL Azure предоставляют возможности управления базами данных, при этом отпадает необходимость в администрировании ядра СУБД, операционной системы и оборудования.

Функциями администраторов информационных систем являются вопросы управления базами данных и учетными записями, индексами, а также оптимизация запросов, аудит и безопасность. При развертывании баз данных в новом дата-центре настройка конфигурации и администрирование системы значительно упрощается.

Сервис SQL Server VM Azure обеспечивает контроль над операционной системой и конфигурацией экземпляра SQL Server. Использование в информационной системе виртуальных машин предполагает выполнение ИТ-подразделением предприятия задач обновления операционных систем и программного обеспечения баз данных. Сервис SQL Server VM Azure предоставляет дополнительные функции автоматизации процессов исправления программного обеспечения, резервного копирования и обеспечения высокой доступности. Администратор информационной системы может оперативно контролировать размер виртуальной машины, количество дисков и их конфигурации хранения. Сервис позволяет изменять размер виртуальной машины по мере необходимости.

Соглашение об уровне обслуживания

Соглашение об уровне обслуживания (Service Level Agreement – SLA) является основным документом, регламентирующим взаимоотношение ИТ-служб предприятия и поставщиков облачных сервисов [5]. Для ИТ-подразделений одним из важных показателей эффективности сервиса является время его непрерывной работы.

Для сервисов БД SQL Azure и УЭ SQL Azure корпорация Microsoft предоставляет соглашение об уровне обслуживания с уровнем доступности 99,99 %.

Для сервиса SQL Server VM Azure в отношении виртуальной машины гарантируется уровень доступности 99,95 %. Данное соглашение не распространяется на процессы (например, SQL Server), запущенные на виртуальной машине, и предусматривает наличие по крайней мере двух экземпляров виртуальных машин в каждой группе доступности.

Создание отдельной базы данных в сервисе БД SQL Azure

Создать отдельную базу данных в службе «База данных SQL Azure» можно с помощью портала Azure, скрипта PowerShell или Azure CLI [6]. Рассмотрим процесс создания базы данных в сервисе БД SQL Azure с помощью портала Azure.

Для этого необходимо перейти на портал Azure с учетной записью Azure (рис. 5.1).

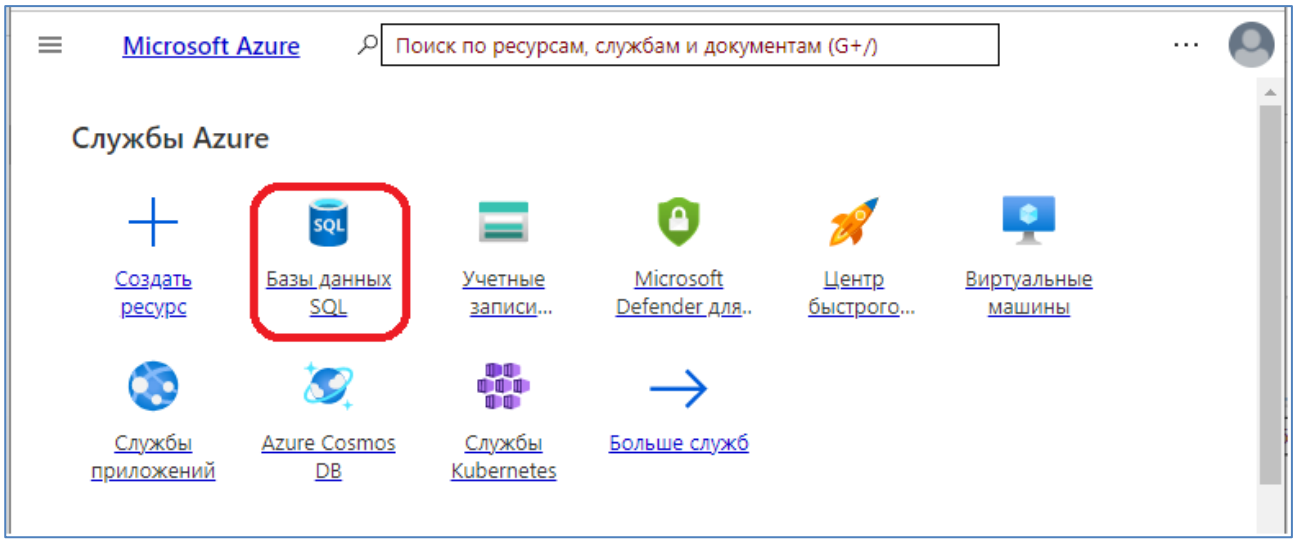

Рисунок 5.1 – Главная страница портала Azure

На главной странице необходимо выбрать пункт меню «Базы данных SQL» и затем пункт «Создать». В результате откроется диалоговое окно «Создать базу данных SQL» (рис. 5.2).

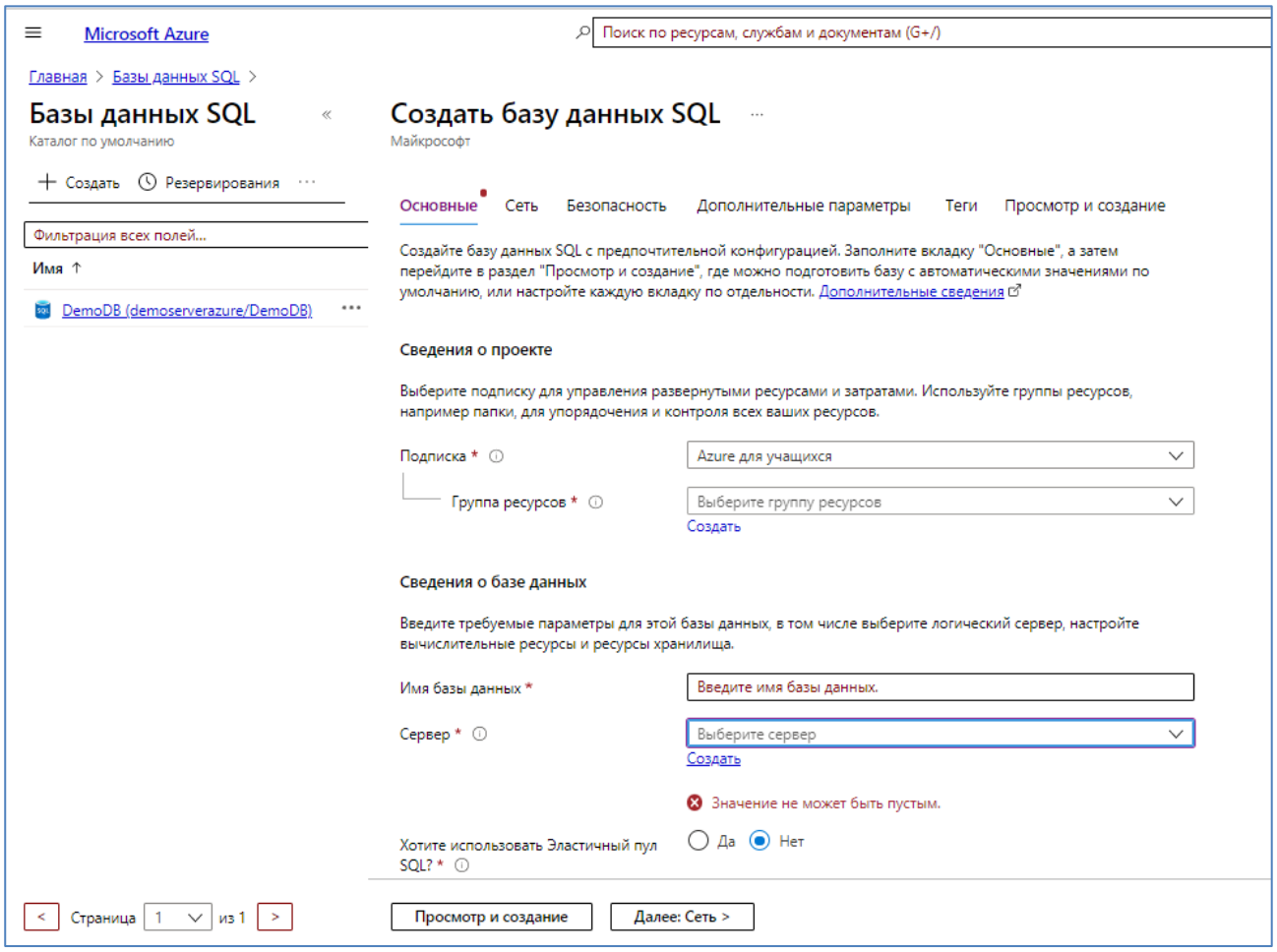

Рисунок 5.2 – Окно «Создать базу данных SQL»

Имя подписки на ресурсы Azure оставим без изменения и создадим новую группу ресурсов DBSQLAzure ()

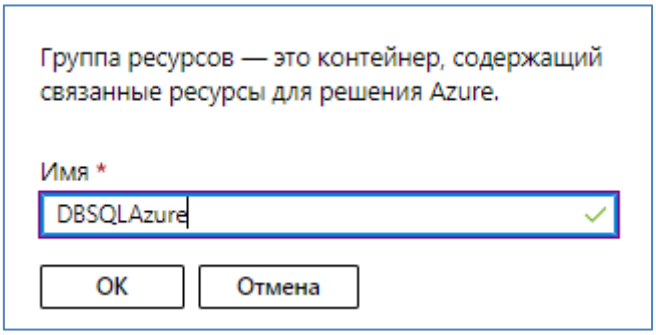

Рисунок 5.3 – Создание группы ресурсов

Введем имя базы данных DemoDBAzure и сформируем данные для сервера (рис. 5.4). Имя сервера – demorsue2022, регион расположения дата-центра – Западная Европа, способ проверки подлинности – использование аутентификации SQL, имя администратора сервера – dalex и пароль.

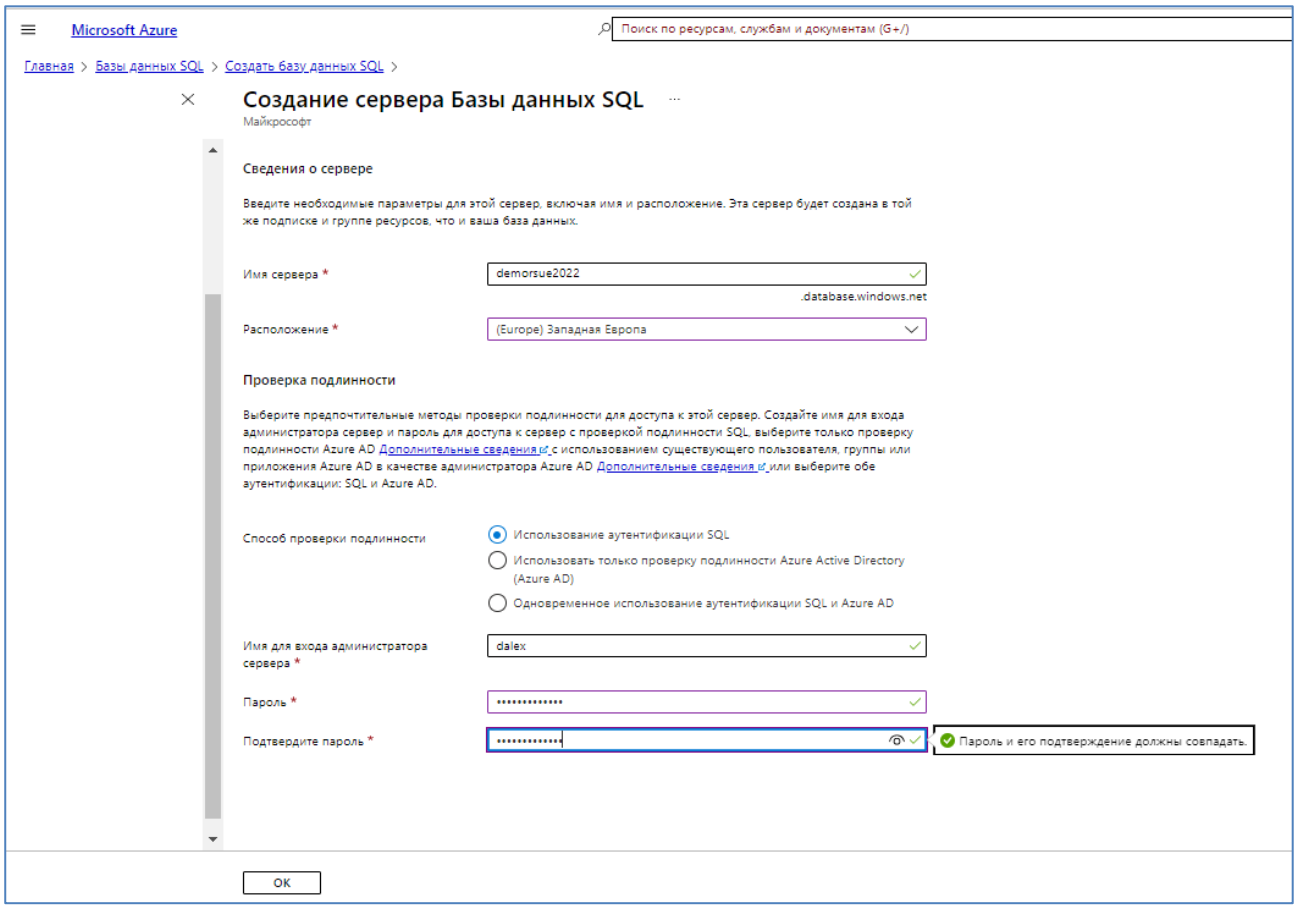

Рисунок 5.4 – Формирование данных для сервера

Для создаваемой базы данных не будем использовать Эластичный пул SQL, назначения вычислений и хранилища определим, как «Общего назначения» и для резервного хранилища определим режим «Локально избыточное хранилище резервных копий» (рис. 5.5).

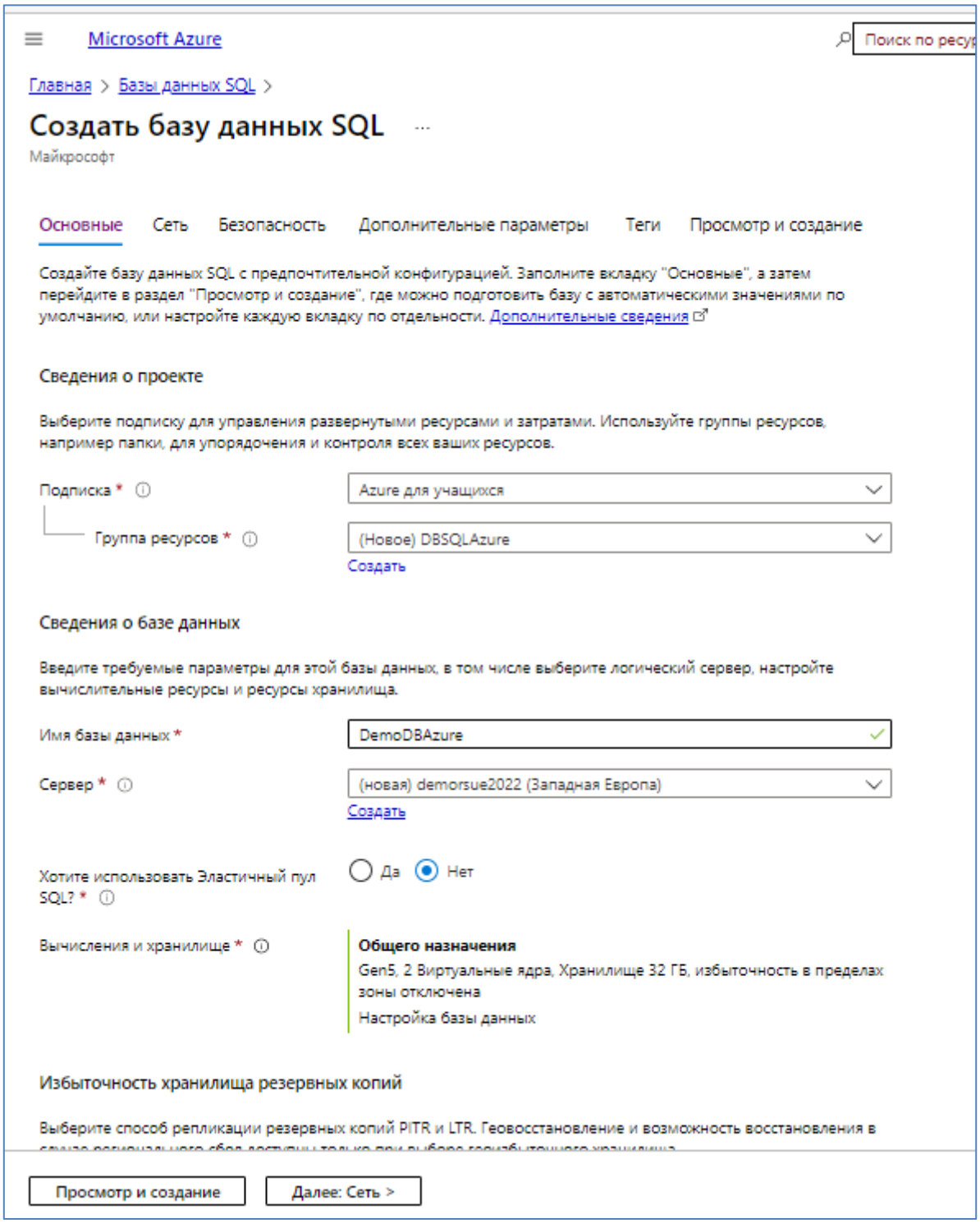

Рисунок 5.5 – Сформированное окно «Создать базу данных SQL»

Следующим этапом формирования базы данных SQL является переход на задание параметров сети (рис. 5.6).

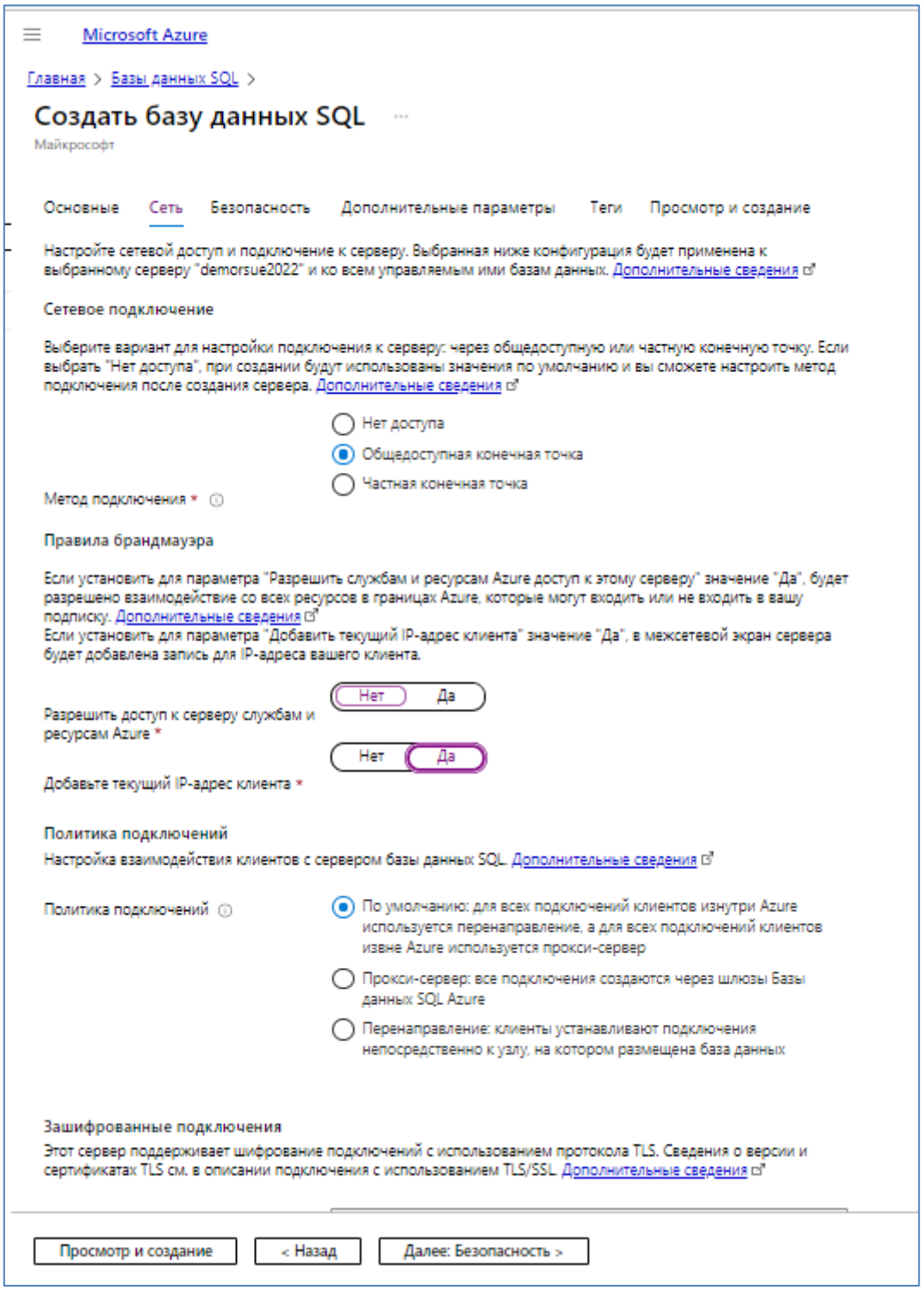

Рисунок 5.6 –Формирование параметров сети

Для параметра сетевое подключение выберем «Общедоступная конечная точка», для правил брандмауэра запретим доступ к серверу службам и ресурсам Azure и разрешим добавить текущий IP-адрес клиента.

Далее необходимо перейти на вкладку «Дополнительные параметры» (рис 5.7).

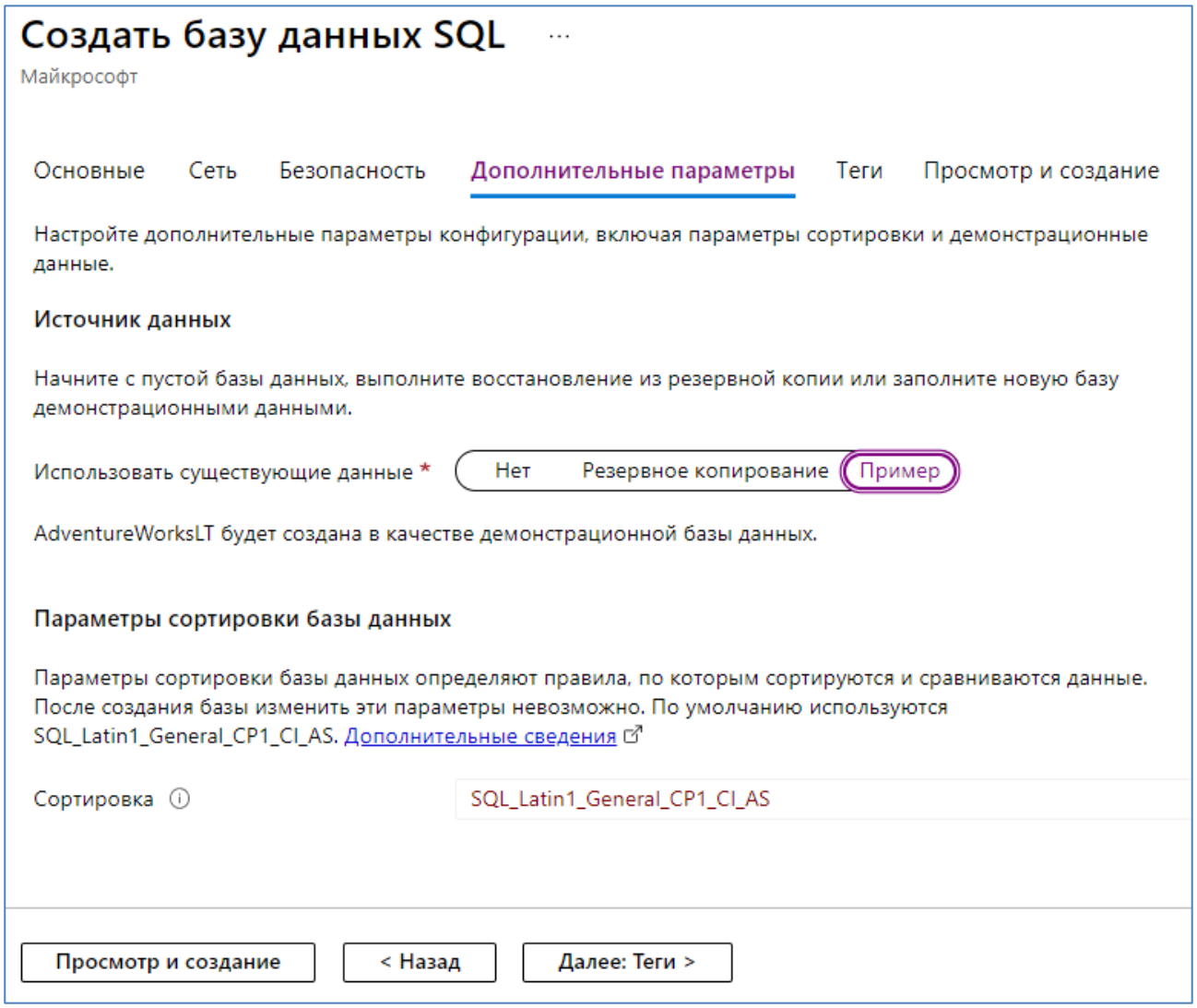

Рисунок 5.7 – Формирование дополнительных параметров

На вкладке «Дополнительные параметры» в разделе «Источник данных» для параметра «Использовать существующие данные» выберем значение «Пример». При этом будет создан образец базы данных AdventureWorksLT, где можно выполнять запросы к некоторым таблицам и данным и экспериментировать с ними в отличие от пустой базы данных.

После нажатия кнопки «Просмотр и создание» открывается окно в котором можно просмотреть параметры сформированной для создания базы данных SQL.

После нажатия кнопки «Создать» начинается процесс развертывания базы данных. При успешном развертывании базы данных на экран выводится информация, приведенная на рис. 5.8.

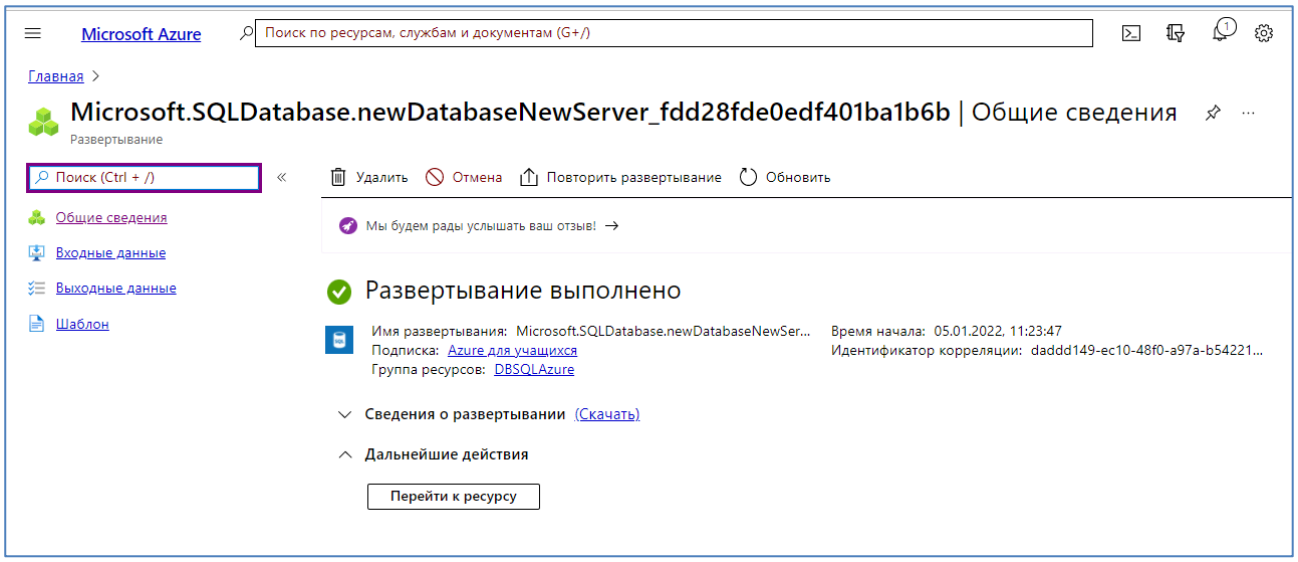

Рисунок 5.8 – Результат развертывания базы данных SQL

Для тестирования работоспособности созданной базы данных на портале Azure выберите элемент *Базы данных SQL* и окне баз данных выберите созданную ранее базу данных DemoDBAzure на сервере Demorsue2022 (рис. 5.9).

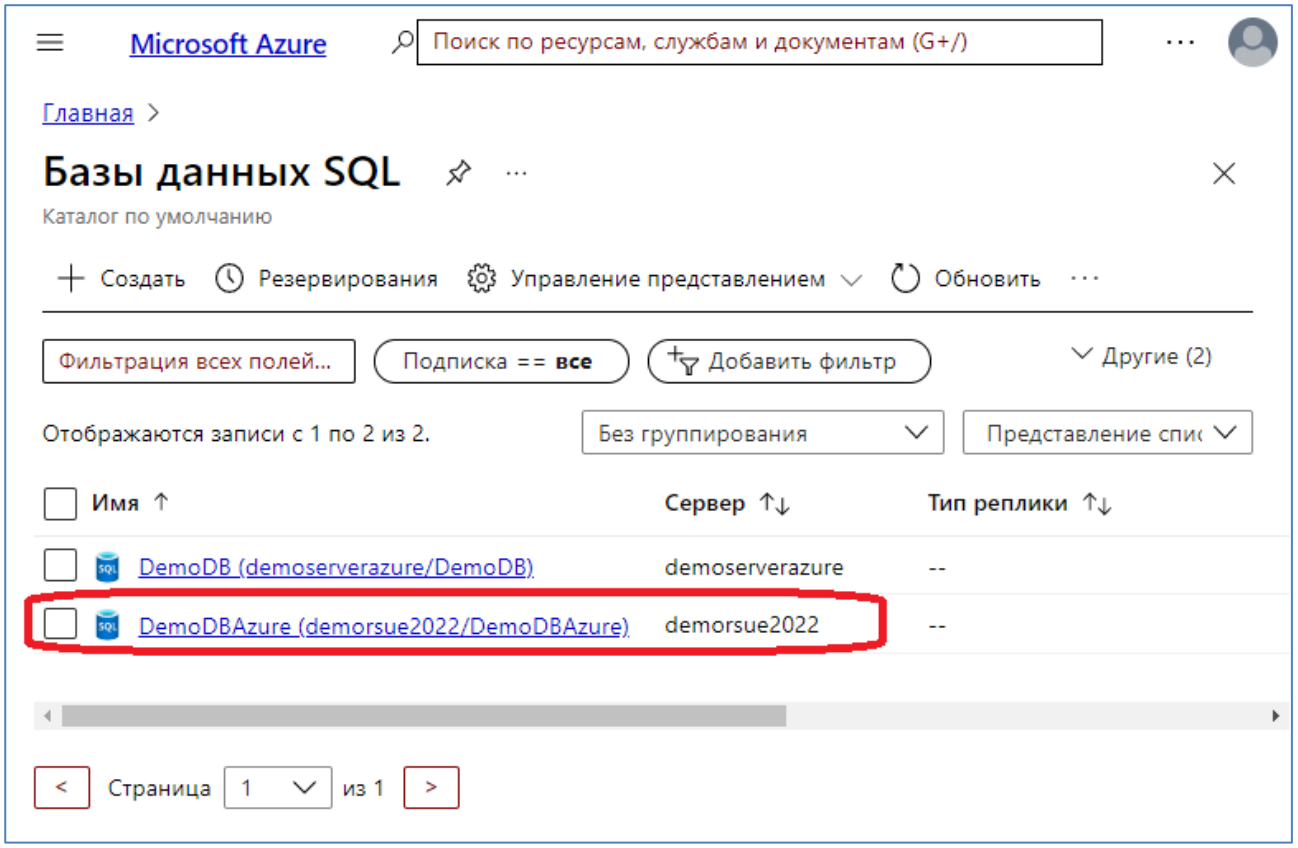

Рисунок 5.9 – Выбор базы данных SQL - DemoDBAzure

В окне базы данных DemoDBAzure выберите пункт меню «Редактор запросов» (рис. 5.10).

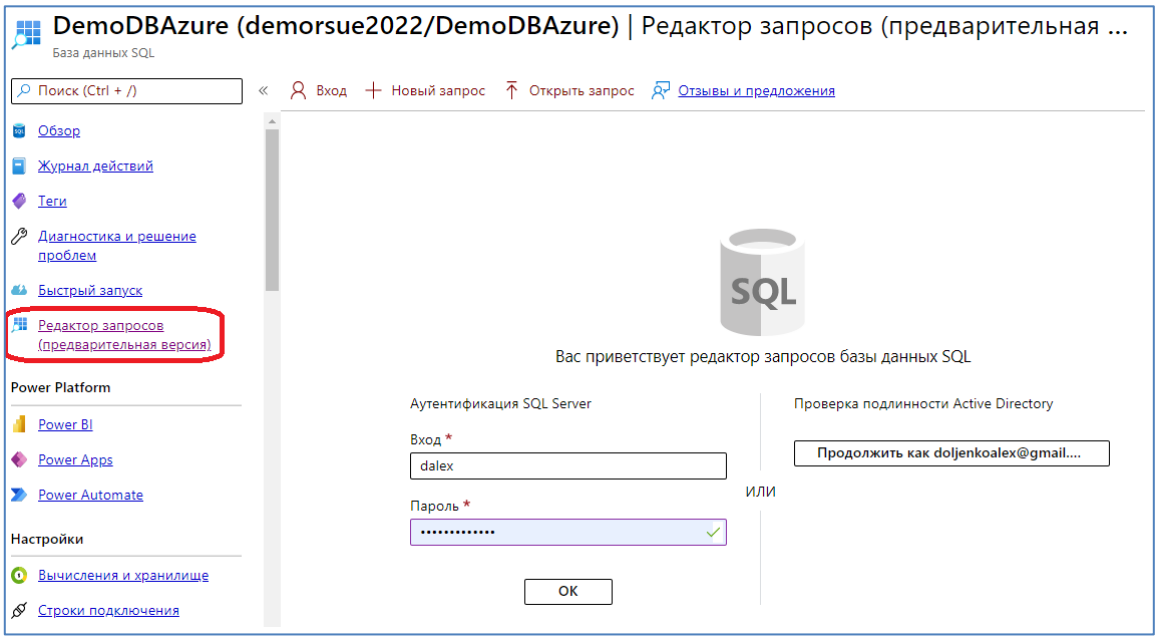

Рисунок 5.10 – База данных DemoDBAzure – Редактор запросов

После ввода логина и пароля администратора базы данных происходит переход на окно «Редактора запросов», в котором необходимо ввести следующий запрос

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
FROM SalesLT.ProductCategory pc
JOIN SalesLT.Product p
ON pc.productcategoryid = p.productcategoryid;
```

Результат выполнения запроса приведен на рис. 5.11.

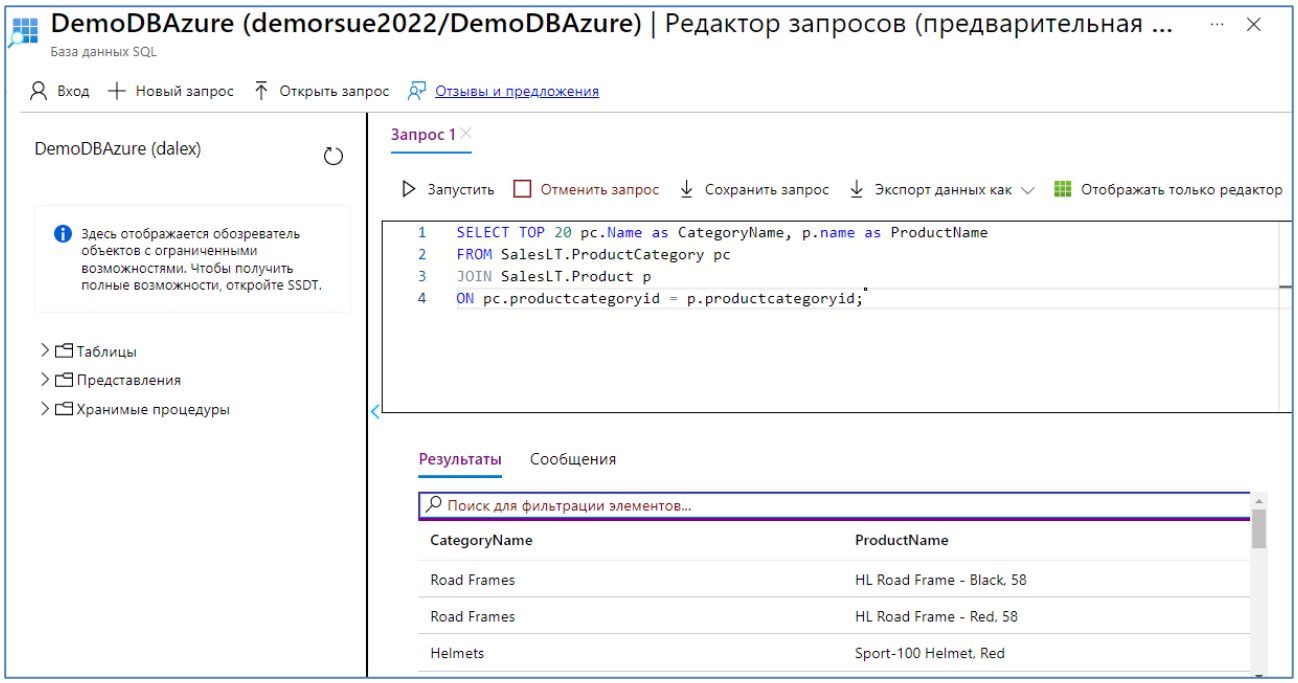

Рисунок 5.11 –Запрос к базе данных DemoDBAzure

Результаты тестирования показывают корректность созданной базы данных DemoDBAzure в облачном сервисе *Базы данных SQL*.

Ключевые термины

База данных SQL Azure – облачный сервис, который предоставляется как услуга и реляционной базой данных.

Управляемый экземпляр SQL Azure – коллекция системных и пользовательских баз данных, которые имеют общий набор ресурсов, и предоставляются как облачный сервис по модели PaaS.

SQL Server на виртуальной машине Azure – облачный сервис, который обеспечивает развертывание SQL Server на полностью управляемой виртуальной машине в облаке.

Соглашение об уровне обслуживания – основной документ, регламентирующим взаимоотношение ИТ-служб предприятия и поставщиков облачных сервисов.

Вопросы для самопроверки

1. Назначение службы Microsoft SQL Azure.
2. Назначение сервиса База данных SQL Azure.
3. Какие возможности развертывания имеются для База данных SQL Azure
4. Какие преимущества имеет сервис База данных SQL Azure?
5. Назначение сервиса Управляемый экземпляр SQL Azure.
6. Какие преимущества имеет сервис Управляемый экземпляр SQL Azure?
7. Назначение сервиса SQL Server на виртуальной машине Azure.
8. Какие преимущества имеет сервис SQL Server на виртуальной машине Azure?
9. Приведите особенности администрирования сервисов SQL Azure.
10. Приведите сравнительные характеристики сервисов База данных SQL Azure и Управляемый экземпляр SQL Azure.
11. Приведите сравнительные характеристики сервисов Управляемый экземпляр SQL Azure и SQL Server на виртуальной машине Azure.
12. Какие уровни доступности декларирует корпорация Microsoft для сервисов База данных SQL Azure и Управляемый экземпляр SQL Azure.

Литература

1. What is Azure SQL? https://docs.microsoft.com/en-us/azure/azure-sql/azure-sql-iaas-vs-paas-what-is-overview
2. База данных SQL Azure. https:// azure.microsoft.com/ru-ru/products/azure-sql/database/#overview

3. Управляемый экземпляр SQL Azure. https://azure.microsoft.com/ru-ru/products/azure-sql/managed-instance/
4. What is SQL Server on Windows Azure Virtual Machines? https://docs.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/sql-server-on-azure-vm-iaas-what-is-overview/
5. Define service-level agreements. https://docs.microsoft.com/en-us/dynamics365/customer-service/define-service-level-agreements
6. Quickstart: Create an Azure SQL Database single database. https://docs.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?tabs=azure-portal

Лекция 6. Windows Azure AppFabric

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: архитектура Azure Service Fabric, управление жизненным циклом приложений Service Fabric, микрослужбы Service Fabric, поддерживаемые модели программирования в Service Fabric, тестирование приложений и служб, кластеры.

Цель лекции:

Цель данной лекции – получить предварительные сведения о платформе распределенных систем Azure Service Fabric.

Введение

Azure Service Fabric представляет собой платформу распределенных систем, которая обеспечивает упаковку и развертывание масштабируемых микрослужб и контейнеров, а также их управление [1]. Применение Service Fabric при разработке облачных приложений способствует снижению сложности процесса разработки и повышению эффективности управления ими.

Service Fabric ориентирован на создание служб с отслеживанием состояния. Это позволяет вести разработку на любом языке программирования, используя модель программирования Service Fabric или запускать контейнерные службы с отслеживанием состояния. Кластеры Service Fabric являются универсальным решением, что позволяет их создавать не только в Azure, но и в других средах, включая локальные среды и другие общедоступные облака Windows Server и Linux.

Архитектура Service Fabric

На рис. 6.1 приведена архитектура Service Fabric. На базе Service Fabric работают многие службы Майкрософт, в том числе база данных SQL Azure, Azure Cosmos DB, Cortana, Microsoft Power BI, Microsoft Intune, Центры событий Azure, Центр Интернета вещей Azure, Dynamics 365, Skype для бизнеса, а также многие основные службы Azure.

Управление жизненным циклом приложения (Lifecycly Menegment). Service Fabric предоставляет поддержку полного жизненного цикла приложений и CI/CD для облачных приложений и контейнеров: от разработки, развертывания, ежедневного мониторинга, управления и технического обслуживания до вывода приложения из эксплуатации [2].

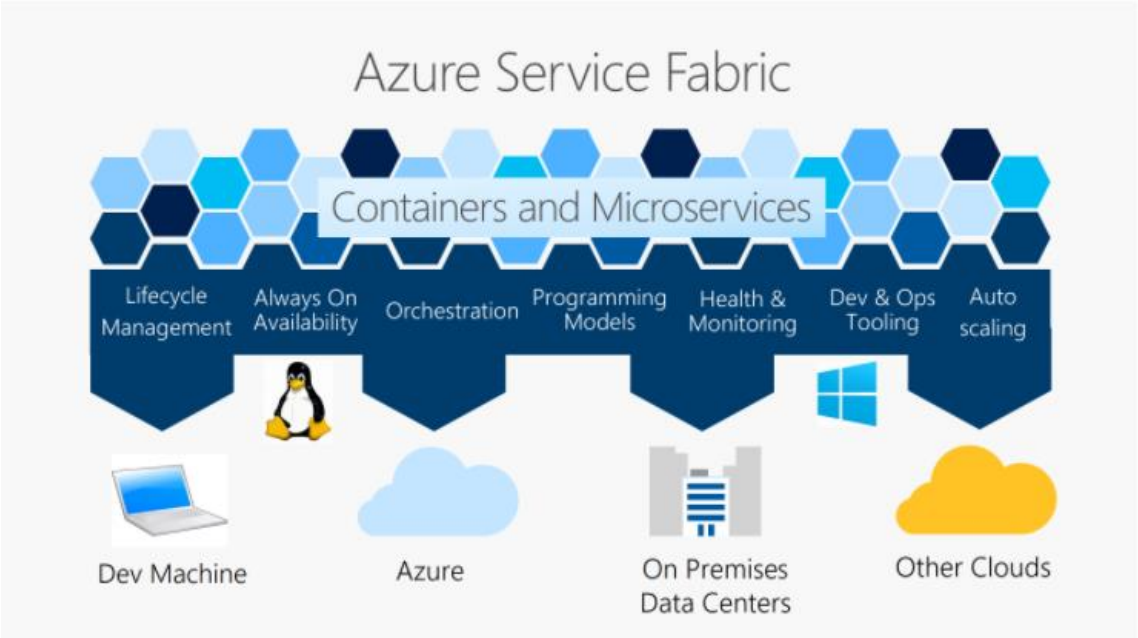

Рис.6.1 – Архитектура Azure Service Fabric

В Service Fabric интегрированы инструменты CI/CD, такие как Azure Pipelines, Jenkins и Octopus Deploy. Эту службу также можно использовать с любым другим популярным инструментом CI/CD.

Зоны доступности (Always on availability) –это предложение высокой доступности, которое защищает приложения и данные от сбоев центра обработки данных [3]. Зона доступности — это уникальное физическое расположение, оснащенное независимым питанием, охлаждением и сетью в регионе Azure.

Оркестрация контейнеров (Orchestration) – этот сервис предназначен для развертывания и управления микрослужбами в кластере [4]. Service Fabric производит быстрое развертывание приложения с высокой. С помощью Service Fabric можно объединять в одном приложении службы с процессами и контейнерами.

Модели программирования (Programming Models) – это различные модели программирования, которые поддерживаются Azure Service Fabric: Reliable Services, Reliable Actors, Containers, ASP.NET Core, Containers [5].

В модели программирования микрослужбы с отслеживанием и без отслеживания состояния, для которых обеспечивается надежная поддержка при создании служб с отслеживанием состояния на основе встроенных моделей программирования или контейнерных служб с отслеживанием состояния.

Кластеры Service Fabric можно создавать на базе операционных систем Windows и Linuz, в облаке или локально.

Управление жизненным циклом приложения

При разработке приложений в Azure Service Fabric необходимо выполнить следующие этапы: проектирование, разработка, тестирование, развертывание, обновление, техническое обслуживание и удаление [2]. Service Fabric предоставляет инструментарий для поддержки жизненного цикла приложений в облаке. В процессе проектирования облачных приложений используются несколько ролей служб:

- разработчик службы;
- разработчик приложения;
- администратор приложения;
- оператор.

Разработку модульных и универсальных служб выполняет *Разработчик службы*. Такие службы предполагают повторное использование и предназначены для применения в различных приложениях. Примером может служить служба очередей, которую можно применить для создания приложения для обработки обращений или приложения для электронной.

Разработчик приложения разрабатывает код в соответствии с техническими и бизнес-требованиями, используя библиотеки, созданных ранее служб. Так для веб-сайта электронной коммерции может потребоваться интеграция интерфейсный службы JSON без отслеживания состояния службы, аукцион с отслеживанием состояния службы и служба очереди с отслеживанием состояния службы для создания аукционного приложения.

Администратор приложения отвечает за разработку конфигурации приложения, процедур развертывания, а также поддержание заданного качества обслуживания. В задачи администратора приложения входит формирование региональных языковых настроек приложения, используемых в зависимости от региона.

В задачи *Оператора* входим развертывание приложения в соответствии с заданной конфигурацией и требованиями, определенных администратором приложения. Оператор может провести развертывание приложения и поддерживать его работу в облаке. В задачи оператора входим мониторинг работоспособности и производительности приложений и поддержка заданной физической инфраструктуры.

В процессе разработки приложения *Разработчик службы* разрабатывает различные типы служб, используя модель программирования Reliable Actors или Reliable Services. При этом декларативно описывают типы создаваемых служб в файле манифеста служб, конфигурации и пакеты данных. Далее *Разработчик приложений* проводит кодирование приложения, используя для этого службы различных типов. При этом *Разработчик приложений* декларативно описывает тип приложения в манифесте приложения путем ссылок на манифесты составляющих его служб и применяет переопределение и назначение параметров

различных конфигураций и настроек развертывания служб, из которых состоит приложение.

При выполнении процесса развертывания приложения *Администратор приложения* изменяет приложение определенного типа для конкретного применения. Приложение развертывается в кластере Service Fabric, при этом задаются параметры элемента ApplicationType в манифесте. Загрузку пакета приложения в хранилище образов кластера выполняет *Оператор*. При этом загружаются манифест приложения и коллекция пакетов служб. Структура служб выполняет развертывание приложений из пакета приложений, размещенного в хранилище образов. Далее *Оператор* задает тип приложения в целевом кластере из загруженного пакета приложения. После этого осуществляется запуск приложения. Для развернутого приложения *Оператор* создает экземпляры служб.

Для тестирования приложение развертывается в локальном кластере разработки или в тестовом кластере. *Разработчик службы* использует специальный тестовый сценарий для проверки переключения на резервный ресурс. При тестировании приложения в сценарии предусматривают выполнение наиболее критичных ситуаций в работе приложения, с целью выполнения допустимых показателей доступности и работоспособности. После выполнения специального сценария *Разработчик службы* выполняет тестирование со случайно генерируемым сценарием. В данном тесте в случайном порядке вызывает множественные ошибки на уровне узла, пакета кода и реплики в кластере. На заключительном этапе тестирования *Разработчик службы* проверяет корректность обмена данными между службами, создавая сценарии проверки для перемещения первичных реплик в кластере.

На этапе обновления приложения *Разработчик службы* выполняет обновление новыми версиями служб экземпляра приложения, при выявлении ошибок кода устраняет их, а также формирует новую версию манифеста служб.

В функции *Разработчика приложения* на этом этапе жизненного цикла приложения входим переопределение и параметризация настройки файлов конфигурации и развертывания, а также и формирование новой версии манифеста приложения. Затем *Разработчик приложения* актуализирует версии манифестов служб в приложение, формирует и собирает новую версию приложения в обновленном пакете приложения.

На *Администратора приложения* возлагается обязанность актуализировать новую версию приложения в целевое приложение посредством обновления необходимых параметров.

Загрузку обновленного пакета приложения в хранилище образов кластера выполняет *Оператор*. Обновленный пакет включает манифест приложения и коллекцию пакетов служб. В функции *Оператор* входим предоставление новой версии приложения для целевого кластера, обновление целевого приложения до новой версии, мониторинг хода обновления. Если возникают проблемы, то

Оператор может изменить и повторно применить параметры текущего обновления приложения, а также инициировать откат текущего обновления приложения.

Цель *технического обслуживания* является периодическое обновление и исправление операционной системы структуры служб, которые взаимодействуют с инфраструктурой Azure для того, чтобы обеспечивать требуемые параметры по доступности всех приложений в кластере.

Для обновлений и исправлений в платформе Service Fabric процесс обновления самой службы Service Fabric выполняется без потери доступности любых приложений, запущенных в кластере.

В функции *Администратора приложения* входит утверждение модификации узлов в кластере на основе анализа и мониторинга лог-файлов об использовании мощностей и прогнозируемой потребности в мощностях в будущем.

На основании рекомендаций *Администратора приложения* добавление и удаление узлов приложения выполняет *Оператор*.

При добавлении новых или удалении существующих узлов из кластера структура службы автоматически балансирует нагрузку запущенных приложений на всех узлах в кластере, чтобы достичь оптимальной производительности.

На этапе Удаления по указанию *Администратора приложения* может реализовываться процесс удаления определенных экземпляров запущенной службы в кластере без удаления всего приложения. Эту работу выполняет *Оператор*. Кроме того, *Оператор* может также удалить экземпляр приложения и все его службы. При остановке приложения и служб *Оператор* имеет возможность отменить выделение мощностей для приложения.

Микрослужбы Service Fabric

С помощью службы Service Fabric можно проектировать приложения на базе микрослужб или контейнеров. Микрослужбы могут быть без отслеживания состояния и с отслеживанием состояния. Микрослужбы без отслеживания состояния (протоколы шлюзов, веб-прокси и т.д.). Такие микрослужбы требуют обработку запроса службой для поддержки изменяемого состояния. К службам можно отнести рабочие роли в облачных службах Azure.

Микрослужбы с отслеживанием состояния (учетные записи пользователей, базы данных, устройства, корзины интернет-магазинов, очереди и т. д.) поддерживают изменяемые достоверные состояния без обработки запроса службой. Современные веб-приложения могут одновременно содержать микрослужбы с отслеживанием состояния и без него.

Можно создать службы оперативной обработки транзакций (OLTP) с высокой пропускной способностью, низкой задержкой и хорошей

отказоустойчивостью, размещая программы и данные рядом на одной виртуальной машине. В качестве примеров таких служб можно привести онлайн-магазины, службы поиска, системы Интернета вещей, торговые системы, системы обработки кредитных карт и обнаружения мошенничества, а также службы управления персональными данными.

Можно создать микрослужбы с отслеживанием состояния, которые устраняют необходимость в дополнительных очередях и кэшах и обычно требуются для обеспечения доступности и минимизации задержек в приложениях без отслеживания состояния. Для служб с отслеживанием состояния изначально характерны высокая доступность и минимальные задержки, поэтому приложением в целом будет проще управлять.

Поддерживаемые модели программирования

С помощью Service Fabric можно использовать несколько способов регистрации и управления создаваемыми службами. API-интерфейсы платформы Service Fabric позволяют эффективно использовать компоненты платформы приложений. При этом программу, написанную на любом языке можно представить, как службу, размещенную в кластере Service Fabric.

Контейнеры. Основным способом развертывания и активации служб в Service Fabric являются процессы. В тоже время Service Fabric поддерживает развертывание служб в контейнерах [6]. Допускаются комбинированное развертывание и активация служб, когда в одном случае используется приложении службы с процессами, а в другом – с контейнерами.

Служба Service Fabric предоставляет разработчику возможность использовать контейнеры Linux или Windows на Windows Server 2016.

В контейнерах можно развернуть имеющиеся приложения, службы без отслеживания состояния или службы с отслеживанием состояния.

Надежные службы (Reliable Services) представляются собой облегченную платформу для регистрации служб, которые интегрируются с платформой Service Fabric [7]. Эти службы обеспечивают поддержку всех х функций платформы. Службы Reliable Services могут быть как без отслеживания состояния, так и с отслеживанием состояния. Если состояние службой не отслеживается, состояние сохраняется во внешнем решении, таком как база данных Azure или хранилище таблиц Azure. Если состояние отслеживается, оно сохраняется прямо в службе с использованием Reliable Collections.

Состояние становится высокодоступным за счет репликации и распределения путем секционирования, которыми автоматически управляет Service Fabric.

Надежные субьекты (Reliable Actor) представляют собой платформу приложений, реализующую модель Virtual Actor на основе шаблона проектирования субъектов, которая реализуется на базе Reliable Services [8].

Reliable Actor, называемые субъектами, обеспечивают применение независимых единиц вычислений и состояний с однопоточным выполнением. Платформа Reliable Actor обеспечивает встроенное взаимодействие для субъектов, а также предустановленное сохранение состояния и масштабируемые конфигурации.

ASP.NET Core является кроссплатформенным фреймворком с открытым исходным кодом, который интегрируется с Service Fabric [9]. В Service Fabric фреймворк *ASP.NET Core* рассматривается в качестве модели программирования первого класса для создания веб-приложений и приложений API. Существуют два способа применения ASP.NET Core в Service Fabric:

В первом способе *ASP.NET Core* следует разместить в виде гостевого исполняемого файла. В основном это используется для запуска существующих приложений ASP.NET Core в Service Fabric без изменения кода.

Во втором способе *ASP.NET Core* следует выполнить в службе Reliable Service. Это обеспечивает более эффективную интеграцию со средой выполнения Service Fabric и позволяет использовать службы ASP.NET Core с отслеживанием состояния.

Гостевой исполняемый файл представляет собой это произвольный существующий исполняемый файл, который может быть написан на любом языке программирования [10]. Данный файл должен быть размещен в кластере Service Fabric среди других служб.

Следует отметить, что данные файлы не интегрируются с интерфейсами API Service Fabric напрямую. Тем не менее они поддерживают преимущества предлагаемых платформой Service Fabric функций, таких как настраиваемые отчеты о работоспособности и загрузке, а также возможности обнаружения службы путем вызова REST API.

Тестирование приложений и служб

При разработке облачных приложений и служб на платформе Service Fabric неотъемлемым этапом является тестирование в части устойчивости к реальным сбоям. Для этого используется *Служба анализа сбоев* [11]. Данная служба использует тестовые сценарии для выявления значимых ошибок облачных приложений и служб. Тестовые сценарии моделируют работоспособность приложений и служб в контролируемых, безопасных и согласованных условиях для разных состояний и переходов, происходящие со службой в течение ее жизненного цикла.

В тестовых сценариях выполняются в службах действия для тестирования с использованием отдельных критических ситуаций [12]. Отдельные сценарии могут использоваться в качестве стандартных блоков для создания более сложных сценариев. Ниже приведены примеры моделирования критических ситуаций:

- перезапустите узел для моделирования любого количества ситуаций, в которых выполняется перезагрузка компьютера или виртуальной машины;
- переместите реплики службы с отслеживанием состояния для имитации балансировки нагрузки, отработки отказа или обновления приложения;
- вызовите потерю кворума в службе с отслеживанием состояния, чтобы создать ситуацию, в которой операции записи невозможны, так как отсутствуют «резервные» или «вторичные» реплики, необходимые для приема новых данных;
- вызовите потерю данных в службе с отслеживанием состояния, чтобы создать ситуацию, в которой все данные о состоянии в памяти полностью уничтожаются.

Кластеры

Кластер Service Fabric представляют собой набор виртуальных машин или физических компьютеров, объединенный в сети. На вычислительных ресурсах кластера развертывают микрослужбы и реализуется управление ими. Кластеры поддерживают широкие возможности по масштабированию до нескольких тысяч машин. В качестве узла кластера может быть компьютер или виртуальная машина, которым присваивается имя. Каждый узел кластера описывается определенными характеристиками, в частности свойствами размещения.

Кластеры Service Fabric можно создать на виртуальных или физических компьютерах под управлением Windows Server или Linux [13].

Кластеры в Azure реализуются в среде Service Fabric. Это предопределяет упрощение процессов интеграции с различными другими функциями и службами Azure. Кластер находится под управлением Azure Resource Manager, что упрощает управление им. Для кластера может использоваться система диагностики Azure и ведение журналов с помощью Azure Monitor, а функция автомасштабирования является встроенной.

Service Fabric можно реализовывать для Linux и Windows, что обеспечивает возможность создания, развертывания высокодоступных кластеров с высокой масштабируемостью.

Изолированные кластеры можно создавать в локальной среде и у любого поставщика облачных служб [14]. Для данных могут существовать ограничения на расположение хранения и хранить их локально, то можно создать собственный кластер и приложения. Приложения Service Fabric можно развертывать и выполнять в различных операционных средах без изменений, так что информация о создании приложений сохраняют свою актуальность при переходе из одной среды размещения в другую.

Безопасность кластера является важным вопросом функционирования приложений Service Fabric. Кластеры должны иметь защиту от

несанкционированного подключения к ним неавторизованных пользователей [15]. Для обеспечения безопасности кластера необходимо обеспечить безопасность обмена данными между узлами, безопасность обмена данными между клиентами и узлами, а также реализовать управление доступом на основе ролей Service Fabric.

Масштабирование. Если вы добавите новые узлы в кластер, реплики секции подвергнутся повторной балансировке Service Fabric с учетом этих узлов. Общая производительность приложения улучшится, а конфликт доступа к памяти уменьшится. При неэффективном использовании узлов в кластере вы можете уменьшить их количество. Service Fabric снова перераспределит реплики и экземпляры секции по меньшему количеству узлов, чтобы эффективно использовать оборудование на каждом узле. Кластеры в Azure можно масштабировать вручную [16] или программным способом [17]. Автономные кластеры можно масштабировать вручную.

Обновления кластера. Периодически выпускаются новые версии среды выполнения Service Fabric [18]. При запуске среды выполнения или Service Fabric кластер обновляется, так что вы всегда используете поддерживаемую версию. Кроме обновлений Service Fabric вы можете также обновить конфигурацию кластера, например сертификаты или порты приложения.

Кластер Service Fabric представляет собой ресурс, который принадлежит вам, но частично управляется корпорацией Майкрософт. Корпорация Майкрософт отвечает за исправления базовой операционной системы и обновления Service Fabric в кластере. Вы можете настроить для кластера автоматическое обновление Service Fabric по мере выпуска новых версий корпорацией Майкрософт или же выбрать нужную версию в списке поддерживаемых. Обновления Service Fabric и конфигурации можно настроить с помощью портала Azure или Resource Manager.

Автономный кластер является ресурсом, который полностью принадлежит вам. Вы отвечаете за исправления базовой операционной системы и запуск обновлений Service Fabric. Если кластер может подключиться к странице https://www.microsoft.com/download, вы можете настроить автоматическое скачивание и подготовку нового пакета среды выполнения Service Fabric. Затем можно инициировать само обновление. Если же кластер не может получить доступ к странице https://www.microsoft.com/download, вы можете вручную скачать новый пакет среды выполнения, используя компьютер с доступом к Интернету, а затем запустить обновление.

Ключевые термины

Azure Service Fabric — платформа распределенных систем, которая обеспечивает упаковку и развертывание масштабируемых микрослужб и контейнеров, а также их управление.

Lifecycly Menegment Service Fabric –поддержка полного жизненного цикла приложений и CI/CD для облачных приложений и контейнеров.

Зоны доступности (Always on availability) – предложение высокой доступности, которое защищает приложения и данные от сбоев центра обработки данных.

Оркестрация контейнеров (Orchestration) – сервис предназначен для развертывания и управления микрослужбами в кластере.

Модели программирования (Programming Models) – различные модели программирования, которые поддерживаются Azure Service Fabric.

Разработчик службы – роль, которая выполняет разработку модульных и универсальных служб Azure Service Fabric.

Разработчик приложения – роль, которая разрабатывает код в соответствии с техническими и бизнес-требованиями, используя библиотеки, созданных ранее служб.

Администратор приложения – роль, которая отвечает за разработку конфигурации приложения, процедур развертывания, а также поддержание заданного качества обслуживания.

Оператор – роль, которая отвечает за развертывание приложения в соответствии с заданной конфигурацией и требованиями, определенных администратором приложения.

Reliable Services –облегченная платформа для регистрации служб, которые интегрируются с платформой Service Fabric.

Reliable Actor – платформа приложений, реализующая модель Virtual Actor на основе шаблона проектирования субъектов, которая реализуется на базе Reliable Services.

Кластер Service Fabric – набор виртуальных машин или физических компьютеров, объединенный в сети.

Вопросы для самопроверки

1. Назначение Azure Service Fabric.
2. Архитектура Service Fabric.
3. Назначение Зон доступности.
4. Назначение оркестрации контейнеров.
5. Назначение Моделей программирования Service Fabric.
6. Возможности управления жизненным циклом приложения в Service Fabric.
7. Роли служб в Service Fabric.
8. Назначение микрослужб в Service Fabric.
9. Назначение контейнеров в Service Fabric.
10. Службы Reliable Services в Service Fabric.
11. Модель Virtual Actor в Service Fabric.
12. Назначение кластеров в Service Fabric.

Литература

1. Overview of Azure Service Fabric. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview
2. Service Fabric application lifecycle. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-application-lifecycle
3. Regions and availability zones. https:// docs.microsoft.com/en-us/azure/availability-zones/az-overview
4. jhrtcnhfwbz
5. Service Fabric programming model overview. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-choose-framework
6. Service Fabric and containers. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-containers-overview
7. Reliable Services overview. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction
8. Introduction to Service Fabric Reliable Actors. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction
9. ASP.NET Core in Azure Service Fabric Reliable Services. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-communication-aspnetcore
10. Deploy an existing executable to Service Fabric. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-guest-executables-introduction
11. Introduction to the Fault Analysis Service. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-testability-overview
12. Testability actions. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-testability-actions
13. Comparing Azure and standalone Service Fabric clusters on Windows Server and Linux. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-deploy-anywhere
14. Create a standalone cluster running on Windows Server. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-creation-for-windows-server
15. Service Fabric cluster security scenarios. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-security
16. Scale a cluster in or out. https:// docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-scale-in-out
17. Scale a Service Fabric cluster programmatically. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-programmatic-scaling
18. Upgrading and updating Azure Service Fabric clusters. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-cluster-upgrade

19. Azure Service Fabric support options. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-support

Лекция 7. Сервисы хранения данных в Windows Azure

Краткая аннотация лекции:

В рамках данной лекции будут рассмотрены следующие вопросы: службы хранения данных – BLOB-объекты, таблицы, очереди; безопасность и защита данных, управление доступом к данным.

Цель лекции:

Цель данной лекции – получить предварительные сведения о службах хранения данных Azure.

Введение

Хранилище Microsoft Azure — управляемая корпорацией Microsoft служба, которая обеспечивает надежные, масштабируемые и резервируемые возможности хранения [1]. В рамках одной подписки Azure можно создать до 100 учетных записей хранения, каждую из которых можно использовать для размещения 500 ТБ данных.

В состав хранилища Azure входит четыре службы для работы с данными:

- хранилище BLOB-объектов (Blob storage);
- хранилище файлов (File storage);
- хранилище таблиц (Table storage);
- хранилище очередей (Queue storage).

Доступно два класса службы хранилища BLOB-объектов — Standard и Premium. В классе Premium для хранения данных используются только твердотельные накопители (SSD), обеспечивающие высочайшую производительность.

Учетные записи хранения

В таблице 7.1 перечислены различные типы учетных записей хранения и объекты, которые поддерживает каждая из них.

Чтобы просмотреть свои объекты данных, можно воспользоваться любым из множества обозревателей хранилища, которые различаются функциональностью [2]. Некоторые данные можно просматривать и обновлять при помощи портала Azure, но в этом случае поддерживаются не все доступные функции.

Например, при помощи портала нельзя загрузить BLOB-объекты или просмотреть сообщения в очереди.

Существует два типа учетных записей хранения общего назначения: Standard и Premium.

Таблица 7.1 – Типы учетных записей хранения

Тип учетной записи хранения	*Учетная запись общего назначения класса Standard*	*Учетная запись общего назначения класса Premium*	*Учетная запись хранилища BLOBобъектов, «горячий» и «холодный» уровни доступа*
Поддерживаемые службы	*Службы BLOB-объектов, файлов, таблиц, очередей*	*Служба BLOB-объектов*	*Служба BLOB-объектов*
Типы поддерживаемых BLOB-объектов	*Блочные BLOB-объекты, страничные BLOBобъекты, BLOB-объекты с добавлением данных*	*Страничные BLOBобъекты*	*Блочные BLOB-объекты и BLOB-объекты с добавлением данных*

Учетные записи хранения класса Standard используются чаще всего. Они подходят для размещения данных всех четырех типов – BLOB-объектов, файлов, таблиц и очередей. Класс Standard подразумевает хранение данных на магнитных накопителях.

Учетные записи хранения класса Premium – это высокопроизводительное хранилище для страничных BLOB-объектов, в том числе виртуальных жестких дисков (VHD). Класс Premium подразумевает хранение данных на твердотельных накопителях (SSD). Рекомендуется использовать хранилище класса Premium для всех виртуальных машин.

Учетная запись хранения BLOB-объектов – специализированная учетная запись для размещения блочных BLOB-объектов и BLOB-объектов с добавлением данных. Хранить страничные BLOB-объекты (а значит, и файлы VHD) в таких учетных записях невозможно. Для этих учетных записей можно выбрать уровень («холодное» или «горячее» хранение) и изменить его в любое время. В «горячих» хранилищах размещаются данные, к которым обращаются часто. Стоимость хранения BLOB-объектов в горячих хранилищах выше, однако цена обращения к ним гораздо ниже. В «холодных» хранилищах размещаются данные, которыми пользуются редко. Стоимость обращения к BLOB-объектам в «холодном» хранилище выше, однако цена их хранения намного ниже.

Службы хранения данных

Хранилище Azure поддерживает размещение объектов четырех типов: BLOB-объектов, файлов (в файловых ресурсах общего доступа), таблиц и очередей. Рассмотрим каждый из этих типов подробнее.

Хранилище BLOB-объектов

Сокращение BLOB означает «binary large object», то есть большой двоичный объект. По сути, BLOB-объект представляет собой обычный файл, подобный тем, которые хранятся на накопителях компьютеров (планшетов, мобильных устройств и т. п.) [3]. Это может быть изображение, файл Microsoft Excel, файл HTML, виртуальный жесткий диск (VHD) — все, что угодно. Служба BLOB-объектов Azure позволяет размещать файлы и обращаться к ним из любой точки мира по URL-адресу посредством интерфейса REST или какой-либо клиентской библиотеки Azure SDK для работы с хранилищем. Клиентские библиотеки для работы с хранилищем доступны на нескольких языках, в числе которых .NET, Node.js, Java, PHP, Ruby и Python. Для работы со службой BLOB-объектов требуется создать учетную запись хранения. После этого вы сможете создавать контейнеры (аналоги папок) и помещать в них BLOB-объекты. Количество контейнеров в учетной записи хранения и количество BLOB-объектов в каждом контейнере не ограничено, однако максимальный объем используемого пространства в учетной записи хранения составляет 500 ТБ. Иерархия контейнеров в службе BLOB-объектов может быть только одноуровневой. Это значит, что помещать одни контейнеры в другие нельзя. Хранилище Azure поддерживает BLOB-объекты трех типов: блочные, страничные и BLOB-объекты с добавлением данных.

Блочные BLOB-объекты используются для хранения обычных файлов размером до 195 ГБ (4 МБ × 50 000 блоков). Чаще всего блочные BLOB-объекты используются для хранения файлов, которые считываются с начала до конца, — например, мультимедийных файлов или изображений для вебсайтов. Такие BLOB-объекты называются блочными, потому что файлы размером более 64 МБ передаются в виде маленьких блоков, которые после этого объединяются в финальный BLOB-объект.

Страничные BLOB-объекты используются для размещения файлов с произвольным доступом размером до 1 ТБ. Страничные BLOB-объекты чаще всего используются как резервное хранилище VHD, то есть для обеспечения устойчивости дисков виртуальных машин Azure (компонента IaaS, который относится к службе вычислений Azure). Такие BLOB-объекты называют страничными, потому что они предоставляют возможность совершать произвольные операции чтения и записи над страницами размером 512 байт.

BLOB-объекты с добавлением информации состоят из блоков, как блочные BLOB-объекты, но при этом оптимизированы для операций дозаписи. Такие объекты часто используют для ведения журналов на основе данных, поступающих в BLOB-объект от одного или нескольких источников. Например, можно вести все журналы трассировки приложения, которое выполняется на нескольких виртуальных машинах, в одном BLOB-объекте с добавлением данных. Максимальный размер BLOB-объекта с добавлением данных составляет 195 ГБ.

Для адресации BLOB-объектов используются URL-адреса следующего формата:

https://[имя учетной записи хранилища].blob.core.windows.net/[контейнер]/[имя BLOB-объекта]

Служба BLOB-объектов поддерживает только один физический уровень контейнеров. Однако она позволяет имитировать файловую систему с папками внутри контейнеров, поскольку имена BLOB-объектов могут содержать символ «/». Клиентские API правильно интерпретируют такую имитацию файловой системы.

Если открыть список BLOB-объектов в обозревателе хранилища, то будет отображена либо иерархическая структура каталогов, либо одноуровневый список.

Также можно назначить учетной записи хранения пользовательский домен, чтобы изменить корневую часть URL-адреса. Тогда адрес может выглядеть примерно так:

http://[storage.companyname.com]/[контейнер]/[имя BLOB-объекта]

В этом случае при обращении к файлам из BLOB-хранилища с веб-сайта для всех ресурсов можно использовать один и тот же домен компании. Кроме того, хранилище BLOB-объектов поддерживает общий доступ к ресурсам независимо от источника, что облегчает работу с ресурсами, размещенными в различных местах.

Хранилище файлов. Служба файлов Azure позволяет создавать сетевые файловые ресурсы общего доступа с высоким уровнем доступности, к которым можно подключаться по стандартному протоколу SMB. Так можно организовать доступ к одним и тем же файлам с возможностями чтения и записи для нескольких виртуальных машин. Также с файлами можно работать посредством интерфейса REST или клиентских библиотек хранилища. Благодаря службе файлов не требуется самостоятельно размещать файловые ресурсы общего доступа в виртуальных машинах Azure и конфигурировать их так, чтобы обеспечить высокую доступность (а это непростая задача). Важное отличие файловых ресурсов общего доступа Azure и локальных файловых ресурсов общего доступа заключается в том, что Azure позволяет обращаться к файлам из

любой точки мира по URL-адресу. Чтобы воспользоваться этой возможностью, необходимо создать подписанный URL-адрес.

Такие файловые ресурсы используются во многих локальных приложениях, что упрощает миграцию в Azure решений, использующих общий доступ к данным. Если подключить файловый ресурс общего доступа к той же букве диска, которая использовалась в локальном приложении, то та часть приложения, которая обращалась к файловому ресурсу общего доступа, должна функционировать без каких-либо изменений.

Если разместить файлы конфигурации в файловом ресурсе общего доступа, то ими могут пользоваться несколько виртуальных машин.

В файловом ресурсе общего доступа можно сохранять журналы диагностики, данные о метриках, аварийные дампы и многое другое.

Инструменты и служебные программы, которыми пользуется группа из нескольких разработчиков, также можно разместить в файловом ресурсе общего доступа, чтобы обеспечить соответствие версий и удобную возможность загрузки.

Чтобы файловый ресурс общего доступа был видим для виртуальной машины, его достаточно просто подключить, после чего к нему можно обращаться по сетевому URL-адресу или через назначенную букву диска. Формат сетевого URL-адреса:

```
\\[имя учетной записи хранения].file.core.windows.net\[имя ресурса общего доступа].
```

После подключения ресурса общего доступа с ним можно работать посредством стандартных API файловой системы: добавлять, изменять, удалять и считывать каталоги и файлы.

Чтобы создать или просмотреть файловый ресурс общего доступа, отправить в него файлы или загрузить их извне Azure, можно воспользоваться порталом Azure, PowerShell, интерфейсом командной строки (Azure CLI), интерфейсами REST, клиентской библиотекой хранилища либо AzCopy, программой командной строки Microsoft.

Также можно воспользоваться любым обозревателем хранилища. При работе с компонентом «Файлы Azure» важно иметь в виду следующее:

- при использовании протокола SMB 2.1 ресурс общего доступа доступен только для тех виртуальных машин, которые относятся к тому же региону, что и учетная запись хранения. Причина заключается в том, что протокол SMB 2.1 не поддерживает шифрование;
- при использовании SMB 3.0 ресурс общего доступа можно подключить к виртуальной машине из любого региона (и даже к настольному компьютеру). Обратите внимание: для подключения файлового ресурса общего доступа Azure должен быть открыт порт 445 (SMB);

- при необходимости обсудите этот вопрос с ответственными сотрудниками вашей компании. Многие поставщики услуг Интернета и ИТ-отделы компаний блокируют этот порт;
- при использовании виртуальных машин Linux подключать можно только файловые ресурсы общего доступа, доступные в регионе, к которому относится учетная запись хранилища;
- подключать файловые ресурсы общего доступа Azure на компьютерах Mac нельзя, так как операционная система Mac не поддерживает шифрование в SMB 3.0;
- интерфейсы REST API позволяют обращаться к данным из любой точки мира;
- эмулятор хранилища не поддерживает файлы Azure;
- максимальный размер файлового ресурса общего доступа составляет 5 ТБ;
- пропускная способность — до 60 МБ/с на ресурс общего доступа;
- максимальный размер файла, который можно поместить в ресурс общего доступа, составляет 1 ТБ;
- максимальное количество операций ввода-вывода в секунду на один ресурс общего доступа составляет 1000 (блоками по 8 КБ);
- аутентификация с использованием Active Directory и списков управления доступом (ACL) на текущий момент не поддерживается, но эти возможности планируется добавить в будущем;
- если к некоторым файлам часто обращаются многократно, вы можете распределить этот набор файлов между несколькими ресурсами общего доступа, чтобы обеспечить максимальную производительность.

Хранилище таблиц

Хранилище таблиц Azure – масштабируемое хранилище данных NoSQL, позволяющее хранить большие объемы частично структурированных нереляционных данных [4]. Оно не поддерживает сложные операции объединения, использование внешних ключей и выполнение хранимых процедур. Каждая таблица содержит один кластеризованный индекс, который можно использовать для быстрой обработки запросов к данным. Альтернативный способ обращения к данным — запросы LINQ и Odata с помощью библиотек WCF Data Service .NET. Хранилище таблиц обычно используется для ведения журналов диагностики. Для работы с хранилищем таблиц необходимо создать учетную запись хранения. После этого можно будет создавать таблицы и вносить в них данные. В таблице хранятся сущности (строки), каждая из которых содержит набор пар ключ-значение. У каждой сущности есть три системных параметра: ключ раздела, ключ строки и метка времени. Сочетание ключа раздела и ключа строки должно быть уникальным, потому что вместе они образуют первичный ключ таблицы. Свойство

PartitionKey (ключ раздела) используется для распределения сущностей между различными узлами хранилища для балансировки нагрузки между ними. Все сущности с одинаковым значением свойства PartitionKey хранятся в одном узле хранилища. Свойство RowKey используется для обеспечения уникальности в рамках одного раздела. Чтобы получить максимальную производительность, необходимо тщательно продумать значения PrimaryKey и RowKey, а также способы получения данных. Не рекомендуется хранить все данные на одном разделе либо помещать каждую сущность в отдельный раздел. Служба таблиц Azure поддерживает цели масштабирования как для учетной записи хранения, так и для разделов. Свойством Timestamp (метка времени) управляет платформа Azure. Это значение соответствует дате и времени последнего изменения сущности. Служба таблиц Azure использует это значение для реализации оптимистичного параллелизма с помощью меток Etag.

Каждая сущность содержит не только системные параметры, но и набор пар «ключ-значение», которые называются свойствами. Какая-либо схема для них отсутствует, поэтому пары «ключ-значение» каждой сущности могут содержать значения различных свойств.

Хранилище очередей

Служба очередей Azure используется для хранения и получения сообщений [5]. Максимальный размер сообщения в очереди составляет 64 КБ, а количество сообщений в очереди ограничено лишь допустимым объемом занятого места в учетной записи хранения. Чаще всего очереди используются для организации списков сообщений для асинхронной обработки. В службе очередей поддерживаются очереди, которые стремятся максимально соответствовать принципу FIFO (но не гарантируют его реализации). Рассмотрим в качестве примера фоновый процесс (например, рабочую роль или веб-задание Azure), который непрерывно проверяет, не появились ли в очереди сообщения. При обнаружении сообщения процесс выполняет его обработку и удаление из очереди. Один из наиболее распространенных примеров такой системы – система обработки изображений или видео. Рассмотрим, например, веб-приложение, которое позволяет клиенту загружать изображения в контейнер в хранилище BLOB-объектов. Для каждого изображения требуется создать эскиз. Не нужно заставлять клиента ждать, пока изображение будет обработано, – файл можно поместить в очередь, указав идентификатор клиента и имя контейнера. Получение сообщения и извлечение идентификатора клиента и имени контейнера выполняет фоновый процесс. После этого он извлекает изображение, создает эскиз и сохраняет его в том же контейнере хранилища BLOB-объектов, в котором находится исходное изображение. После обработки всех изображений фоновый процесс удаляет сообщение из очереди. Что, если требуются сообщения размером больше 64 КБ? В этом случае файл с данными можно

записать в BLOB-объект в соответствующем хранилище, а в сообщение очереди добавить его URL-адрес. Фоновый процесс сможет принимать сообщения из очереди, извлекать URL-адрес и считывать файл из хранилища BLOB-объектов для последующей обработки. Очереди Azure устроены так, что каждое сообщение может быть считано один или несколько раз. Поэтому обработка сообщения должна быть полностью идемпотентной, то есть результат обработки не должен зависеть от количества повторений этой операции. Когда вы получаете сообщение из очереди, оно не удаляется из нее автоматически: сообщение необходимо удалить явным образом, закончив обработку. После считывания сообщения из очереди оно становится невидимым. Параметр «таймаут невидимости» соответствует допустимому интервалу времени для обработки сообщения. Если в течение этого времени сообщение не удалено из очереди, оно вновь становится видимым для обработчиков. В общем случае рекомендуется присваивать этому параметру значение, которое соответствует максимально допустимому времени обработки сообщения. Так вы сможете избежать ситуации, когда сообщение принимает для обработки определенный экземпляр или рабочая роль, а другой исполнитель обнаруживает это сообщение видимым в очереди и запускает его параллельную обработку. Считывать сообщение, удалять его из очереди и только потом начинать обработку не рекомендуется. Если процесс-получатель сообщения завершится с ошибкой, то оно вообще не будет обработано. Сохранение сообщения в очереди (в невидимом состоянии) позволяет справиться с ошибками принимающего процесса. Через некоторое время сообщение вновь станет видимым и будет обработано другим экземпляром получателя.

Можно формировать рабочий процесс, используя различные очереди для различных этапов. Процесс может принимать сообщение из очереди, обрабатывать его и удалять из очереди по завершении, а затем помещать другое сообщение в другую очередь, где с ним будет работать процесс следующего этапа. Также можно управлять приоритетом сообщений, используя очереди и назначая сообщениям в них различные приоритеты для обработчиков.

Служба очередей обрабатывает подозрительные сообщения (poison messages), используя счетчик вывода из очереди. Проблема заключается в том, что неверно составленное сообщение может привести к аварийному завершению приложения-обработчика, и тогда сообщение снова станет видимым в очереди и снова вызовет аварийный сбой приложения при следующей обработке. Такие сообщения называются подозрительными. Чтобы избежать такой ситуации, вы можете проверять значение параметра «счетчик вывода из очереди» для сообщения (dequeue count for the message). Если он превосходит некоторое значение, то следует прекратить обработку сообщения, удалить его из очереди и поместить его копию в отдельную очередь подозрительных сообщений для последующего анализа. Такие сущности можно обрабатывать периодически: настроить отправку электронного письма при каждом добавлении сущности в

такую очередь или просто накапливать их и проверять содержимое очереди вручную.

Кроме того, поддерживается пакетная обработка сообщений очереди: можно принять несколько (до 32) сообщений одним вызовом и обработать их по отдельности. Обратите внимание: при приеме пакета сообщений платформа устанавливает для каждого из них общее значение таймаута. Это значит, что все эти сообщения нужно будет обработать за выделенное время.

Избыточность

Что произойдет, если узел хранилища, на котором размещены ваши BLOB-объекты, аварийно завершит работу? Что, если случится сбой всей стойки, в которой находится узел хранения? Azure поддерживает избыточность. Параметр избыточности может принимать четыре значения и устанавливается при создании учетной записи хранения. Режим избыточности можно изменить после его начальной установки, если не было выбрано хранилище, избыточное в пределах зоны.

Локально избыточное хранилище (Locally Redundant Storage, LRS) Для обеспечения высокой доступности хранилище Azure считает операцию записи успешной только после синхронного создания трех копий данных. Эти копии хранятся на устройствах в одном регионе и в одном помещении. Реплики размещаются в различных доменах сбоя и доменах обновления. Так обеспечивается доступность данных даже в том случае, если на узле хранилища с данными произойдет сбой или отключение для обновления. Когда вы отправляете запрос на обновление хранилища, Azure направляет его всем трем копиям, ждет их ответа об успешной операции и только потом передает ответ вам. Это значит, что копии, размещенные в основном регионе, всегда синхронизированы. LRS дешевле, чем GRS, и обеспечивает более высокую пропускную способность. Если приложение хранит данные, которые можно легко восстановить, LRS будет оптимальным вариантом.

Геоизбыточное хранилище (Geo-Redundant Storage, GRS) В режиме GRS создается три синхронные копии данных в основном регионе для обеспечения высокой доступности, а затем асинхронно создается три реплики в связанном регионе для возможности аварийного восстановления. Каждому региону Azure соответствует определенный связанный регион GRS, который относится к той же геополитической зоне. Такую пару образуют, например, западная часть США и восточная часть США. Это мало влияет на целевые показатели масштабируемости учетной записи хранения. Копии GRS в связанном регионе для вас недоступны. Аварийное восстановление данных из GRS — задача корпорации Microsoft, а не ваша. В случае масштабного сбоя в основном регионе Microsoft откроет доступ к репликам GRS, но такого еще ни разу не случалось.

Геоизбыточное хранилище с доступом на чтение (Read-Access Geo-Redundant Storage, RA-GRS) Это хранилище GRS с дополнительной возможностью считывать данные из вторичного региона, благодаря которой оно подходит для частичного аварийного восстановления клиентских данных. В случае проблем в основном регионе вы сможете настроить приложение так, чтобы оно считывало данные из связанного региона. Возможность переключения на резервный регион реализована в компоненте Microsoft.WindowsAzure.Storage.RetryPolicies.LocationMode клиентской библиотеки хранилища. Она позволяет считывать данные из вторичной копии в случае, если основная копия недоступна. Эта возможность реализована и готова к использованию. При сбое ваши клиенты, возможно, не смогут обновлять данные, однако смогут их считывать, составлять отчеты на их основе и т. п. Такое хранилище также подходит для приложений, в которых много пользователей, которые считывают данные, и лишь несколько пользователей, которые их записывают. Вы можете настроить приложение, записывающее данные, так, чтобы оно взаимодействовало с основным регионом, а систему для чтения данных — на работу со связанным регионом. Это хороший способ распределить обращения к учетной записи хранения.

Хранилище, избыточное в пределах зоны (Zone-Redundant Storage, ZRS) Этот тип хранилища поддерживается только для блочных BLOB-объектов в стандартной учетной записи хранения. При использовании такого хранилища данные реплицируются между двумя или тремя физическими средами, которые находятся в одном или в двух регионах. Это обеспечивает более высокую устойчивость, чем при использовании LRS, однако учетные записи ZRS не поддерживают ни метрики, ни ведение журналов.

Безопасность и хранилище Azure

Хранилище Azure поддерживает ряд функций обеспечения безопасности, которые помогают разработчикам создавать защищенные приложения [6]. Чтобы обезопасить свою учетную запись хранения, можно воспользоваться управлением доступом на основе ролей (RBAC) и Microsoft Azure Active Directory (Azure AD). Доступно несколько механизмов защиты данных при передаче: шифрование на стороне клиента, HTTPS и SMB 3.0. Если включить шифрование службы хранилища, то служба хранилища Azure будет шифровать данные, которые помещаются в учетную запись хранения. Шифрование службы хранилища теперь можно включить в том числе для дисков ОС и дисков с данными виртуальных машин. Для защиты доступа к объектам плоскости данных (data plane; например, к BLOB-объектам) можно воспользоваться подписанным URL-адресом (SAS). Рассмотрим каждую из этих возможностей отдельно.

Обеспечение защиты учетной записи хранения

В первую очередь необходимо обеспечить безопасность учетной записи хранения [7].

Ключи учетной записи хранения. Каждой учетной записи хранения соответствует два ключа проверки подлинности — основной и вторичный. Любой из них позволяет выполнять все допустимые действия. Ключа создается два, чтобы между ними можно было переключаться для повышения безопасности. Крайне важно обеспечить надежное хранение этих ключей. Наличие любого из них и знание имени учетной записи открывает полный доступ ко всем данным учетной записи хранения. Предположим, вы используете ключ 1 учетной записи хранения в нескольких приложениях. Вы можете запустить повторную генерацию ключа 2, изменить все приложения так, чтобы в них использовался ключ 2, и развернуть их в рабочей среде. Если после этого вы запустите повторную генерацию ключа 1, то все пользователи и приложения, которые используют его прежнюю версию, лишатся доступа. Рассмотрим ситуацию, в которой эта возможность оказывается полезной. Допустим, ваш отдел использует обозреватель хранилища, который сохраняет ключи учетной записи. В один день один из сотрудников уходит из отдела или вообще из компании. Вам требуется лишить его доступа к данным. Такая необходимость часто возникает неожиданно, поэтому следует заранее подготовить процедуру, которая позволит понять, какие именно приложения нужно изменить, а затем попрактиковаться в проведении ротации ключей, чтобы это не вызвало сложностей, когда такая необходимость действительно возникнет.

Управление доступом к учетным записям хранения Диспетчера Ресурсов с помощью RBAC, Azure AD и Azure Key Vault

RBAC и Azure AD. Механизм RBAC Диспетчера Ресурсов позволяет назначать роли пользователям, группам и приложениям [8]. Каждой роли соответствуют наборы разрешенных и запрещенных действий. При предоставлении доступа к учетной записи хранения посредством RBAC затрагиваются только операции управления этой учетной записью. Предоставлять доступ к объектам плоскости данных (например, к отдельным контейнерам или файловым ресурсам общего доступа) с помощью RBAC нельзя. Однако посредством RBAC можно открыть доступ к ключам учетной записи хранения, а с их помощью — считывать объекты данных.

Например, вы можете предоставить пользователю роль «Владелец» (Owner) в отношении учетной записи хранения. У такого пользователя будет доступ к ключам, а значит, он сможет работать с объектами данных, создавать учетные записи хранения и вообще выполнять практически любые действия.

Также предусмотрена роль, которая называется «Читатель» (Reader). Она позволяет получать информацию об учетной записи хранения. Пользователь с такой ролью сможет получать данные о ресурсах и их группах, однако не будет иметь доступа к ключам учетной записи хранения, а значит, и к объектам данных.

Если пользователю необходимо создавать виртуальные машины, ему следует назначить роль «Участник виртуальных машин» (Virtual Machine Contributor). Она позволяет получать ключи учетной записи хранения, но не позволяет создавать новые учетные записи хранения. Ключи потребуются такому пользователю для создания файлов VHD, на основе которых работают диски виртуальных машин.

Azure Key Vault. Azure Key Vault помогает обеспечить безопасность криптографических ключей и секретных данных, которые используются в службах и приложениях Azure. Ключи учетной записи хранения также можно поместить в Azure Key Vault. Для чего это нужно? С помощью Active Directory управлять доступом непосредственно к объектам данных нельзя, а к Azure Key Vault — можно. Таким образом, вы можете поместить ключи учетной записи хранения в Azure Key Vault и предоставить доступ к ним определенному пользователю, группе или приложению.

В качестве примера рассмотрим веб-приложение, которое загружает файлы в учетную запись хранения. Необходимо надежно защитить эти файлы от несанкционированного доступа. Вы вносите это приложение в Azure Active Directory и предоставляете ему доступ к Azure Key Vault, в котором размещены ключи этой учетной записи хранения. После этого доступ к ключам будет только у этого приложения. Это гораздо более безопасный подход, нежели простое размещение ключей в файле web.config, где хакер может до них добраться.

Обеспечение защиты доступа к данным

Существует два способа защитить доступ к объектам данных. Первый (управление доступом к ключам учетной записи хранения) мы только что рассмотрели. Второй способ — использование подписанных URL-адресов и хранимых политик доступа [9]. SAS представляет собой строку, содержащую маркер безопасности. Эту строку можно дописать в конец URI-кода ресурса, который поддерживает делегирование доступа к конкретным хранимым объектам, и задавать ограничения, — например, разрешения и интервал времени, в течение которого доступ открыт.

Поддерживается предоставление доступа к BLOB-объектам, контейнерам, сообщениям очереди, файлам и таблицам. В рамках таблиц можно предоставлять доступ к конкретным ключам разделов. Если, например, в качестве ключа раздела вы используете название города, то сможете предоставить кому-либо доступ только к информации по Новосибирску.

Такой механизм позволяет предоставлять строго тот уровень доступа, который необходим для решения конкретных задач. Например, вы можете предоставить веб-приложению возможность записывать сообщения в очередь, запретив считывать и удалять их. Рабочей роли или веб-задаче Azure можно назначить права на чтение сообщений, их обработку и удаление. Каждому компоненту назначается максимально узкий набор прав, достаточный для выполнения его работы.

Существуют SAS уровня учетной записи и уровня службы. SAS уровня учетной записи позволяют получать списки контейнеров, создавать контейнеры, удалять файловые ресурсы общего доступа и т. п. SAS на уровне службы позволяют лишь получить доступ к объектам данных. С их помощью можно, например, загрузить BLOB-объект в контейнер.

Также можно создавать хранимые политики доступа для объектов с функциональностью контейнеров, например, для контейнеров BLOB-объектов или файловых ресурсов общего доступа. Это позволит задать значения параметров запроса, которые будут действовать по умолчанию. После этого можно будет создавать SAS-адреса, в которых указывается политика и параметры запроса, которые отличаются от параметров политики. Например, можно создать политику, которая предоставляет доступ на чтение к некоторому контейнеру. Когда кому-нибудь понадобится доступ к этому контейнеру, вы сможете создать SAS на основе политики и воспользоваться им.

У хранимых политик доступа есть два преимущества. Во-первых, они позволяют скрыть параметры, которые определены в политике. Если вы настроите политику так, чтобы она открывала доступ на 30 минут, то это значение не будет содержаться в URL-адресе — в нем будет указано только имя политики. Это безопаснее, чем передавать все параметры в открытом виде.

Вторая причина использовать хранимые политики доступа заключается в том, что их можно отозвать. Вы можете изменить дату окончания срока действия политики на более раннюю относительно текущего времени или просто удалить политику. Так можно отменить ошибочно предоставленный доступ к объекту. При использовании URL-адресов SAS вам потребовалось бы удалять ресурс или менять ключи учетной записи хранения, чтобы отозвать доступ.

Подписанные URL-адреса (SAS) и хранимые политики доступа — два самых надежных способа предоставления доступа к объектам данных.

Обеспечение защиты данных при передаче

При размещении данных в хранилище Azure важно обеспечить безопасность данных в процессе их передачи между службой хранения и приложениями. Первая рекомендация: всегда используйте протокол HTTPS. Он обеспечивает безопасность передачи данных через общедоступные Интернет-

узлы. Механизм SAS позволяет добавить в запрос параметр, при наличии которого URL-адрес будет доступен только по протоколу HTTPS.

Что касается файловых ресурсов общего доступа Azure, протокол SMB 3.0 в Windows шифрует данные, передаваемые через общедоступный Интернет. Когда разработчики Apple и Linux добавят поддержку безопасности в SMB 3.0, появится возможность подключать файловые ресурсы общего доступа к компьютерам под управлением этих систем, и данные при передаче также будут шифроваться.

Можно воспользоваться шифрованием на стороне клиента, которое реализовано в клиентских библиотеках хранилища .NET и Java. Эта функция позволяет зашифровать данные перед тем, как передавать их по сети. Данные будут расшифровываться при получении. Эта технология реализована в клиентских библиотеках хранилища на языках .NET и Java. Она также считается способом шифрования данных при хранении, потому что хранимые данные также зашифрованы.

Ключевые термины:

Хранилище Microsoft Azure - служба, которая обеспечивает надежные, масштабируемые и резервируемые возможности хранения данных.

Учетная запись хранения класса Standard – учетная запись для размещения данных всех четырех типов – BLOB-объектов, файлов, таблиц и очередей.

Учетная запись хранения класса Premium – это высокопроизводительное хранилище для страничных BLOB-объектов, в том числе виртуальных жестких дисков.

Учетная запись хранения BLOB-объектов – специализированная учетная запись для размещения блочных BLOB-объектов и BLOB-объектов с добавлением данных.

Хранилище BLOB-объектов – представляет собой хранилище больших двоичных файлов.

Хранилище таблиц Azure – масштабируемое хранилище данных NoSQL, позволяющее хранить большие объемы частично структурированных нереляционных данных.

Хранилище очередей – служба Azure, которая используется для хранения и получения сообщений.

Локально избыточное хранилище – хранилище, которое используется для обеспечения высокой доступности, и считает операцию записи успешной только после синхронного создания трех копий данных.

Геоизбыточное хранилище – хранилище, которое используется для обеспечения высокой доступности, в котором создаются три синхронные копии

данных в основном регионе, а затем асинхронно создается три реплики в связанном регионе для возможности аварийного восстановления.

Геоизбыточное хранилище с доступом на чтение – хранилище, которое используется для обеспечения высокой доступности, в котором имеется дополнительная возможность считывать данные из вторичного региона, благодаря которой оно подходит для частичного аварийного восстановления клиентских данных.

Хранилище, избыточное в пределах зоны чтение – хранилище, которое используется для обеспечения высокой доступности, в которое поддерживается только для блочных BLOB-объектов в стандартной учетной записи хранения.

RBAC Azure – это система авторизации на основе Azure Resource Manager, которая обеспечивает широкие возможности управления доступом к ресурсам Azure.

Подписанные URL-адреса – это механизм авторизации на основе утверждений, использующий простые маркеры.

Вопросы для самопроверки

1. Назначение хранилища Microsoft Azure.
2. Учетные записи хранения.
3. Назначение учетной записи хранения класса Standard.
4. Назначение учетной записи хранения класса Premium.
5. Назначение учетной записи хранения BLOB-объектов.
6. Хранилище BLOB-объектов.
7. Служба файлов Azure.
8. Хранилище таблиц Azure.
9. Избыточность хранилищ Azure.
10. Безопасность для хранилища Azure.
11. Управление доступом к учетным записям хранения.
12. Обеспечение защиты доступа к данным.

Литература

1. Общие сведения об основных службах хранилища Azure. https://docs.microsoft.com/ru-ru/azure/storage/common/storage-introduction.
2. Клиентские инструменты Майкрософт для работы со службой хранилища Azure. https://azure.microsoft.com/documentation/articles/storage-explorers/.
3. Общие сведения о хранилище BLOB-объектов Azure. https://docs.microsoft.com/ru-ru/azure/storage/blobs/storage-blobs-introduction.

4. Что собой представляет табличное хранилище Azure? https://docs.microsoft.com/ru-ru/azure/storage/tables/table-storage-overview
5. Что такое Хранилище очередей Azure? https://docs.microsoft.com/ru-ru/azure/storage/queues/storage-queues-introduction.
6. Рекомендации по обеспечению безопасности для хранилища BLOB-объектов. https://docs.microsoft.com/ru-ru/azure/storage/blobs/security-recommendations
7. Общие сведения об учетной записи хранения. https://docs.microsoft.com/ru-ru/azure/storage/common/storage-account-overview.
8. Что такое управление доступом на основе ролей в Azure (RBAC)? https://docs.microsoft.com/ru-ru/azure/role-based-access-control/overview
9. Управление доступом к служебной шине с помощью подписанных URL-адресов. https://docs.microsoft.com/ru-ru/azure/service-bus-messaging/service-bus-sas.

Лекция 8. Частное облако: идеология построения частного облака, базовые типы сервисов

Краткая аннотация лекции.

В данной лекции рассматривается вопросы конфигурации на портале веб-приложений и возможности масштабирования этого веб-приложения

Цель лекции.

Целью данной лекции является ознакомление с конфигурированием, масштабированием и мониторингом веб-приложений

Введение

Платформа как услуга (PaaS) хорошо подходит для развертывания рабочих нагрузок определенного типа. Однако модель PaaS подходит не для всех решений, и это совершенно нормально. При использовании некоторых рабочих нагрузок необходимо контролировать практически все аспекты инфраструктуры: конфигурацию операционной системы, сохраняемость диска, возможность устанавливать и конфигурировать традиционное серверное программное обеспечение и т. д. Для решения этих задач используется подход «инфраструктура как услуга» (IaaS) и виртуальные машины Azure.

Виртуальные машины

Виртуальные машины Azure — одна из ключевых IaaS-возможностей Azure (вторая, столь же важная — виртуальные сети Azure) [1]. Служба «Виртуальные машины Azure» (Azure Virtual Machines) позволяет разворачивать виртуальные машины под управлением Windows или Linux в центре обработки данных Microsoft Azure. Конфигурацией виртуальной машины полностью управляет администратор информационной системы. Установка, конфигурирование и обслуживание всего серверного программного обеспечения и исправлений операционной системы в данном случае являются задачей администратора.

Вычислительные возможности Azure PaaS и IaaS различаются в двух отношениях: сохраняемость и возможности управления. Управление такими компонентами PaaS, как облачные службы (т. е. веб-роли и рабочие роли) и службы приложений, практически полностью берет на себя платформа Azure, что позволяет уделять меньше внимания управлению серверной инфраструктурой в пользу разработки приложений. При работе с виртуальными

машинами Azure настройка практически всех характеристик виртуальных машин является задачей администратора информационной системы.

В виртуальных машинах Azure поддерживаются два типа устойчивых (сохраняемых) дисков: диски ОС и диски с данными. Диск ОС требуется для работы виртуальной машины, диски с данными используются по необходимости. Устойчивость дисков обеспечивается хранилищем Azure. На диске ОС размещается операционная система (Windows или Linux), а на диск с данными можно поместить что-то еще — данные приложений, изображения и т. п. В облачных PaaS-службах Azure используется совершенно другой подход: несохраняемые диски, подключенные к физическому узлу, данные на которых могут быть утеряны в случае сбоя физического узла.

Доступные пользователю возможности управления и использования устойчивых дисков делают виртуальные машины идеальным вариантом для размещения множества серверных рабочих нагрузок, которые не соответствуют модели PaaS. Такой подход позволяет запускать серверы баз данных (SQL Server, Oracle, MongoDB и т. п.), Windows Server Active Directory, Microsoft SharePoint и многие другие нагрузки на платформе Microsoft Azure. При необходимости пользователи могут мигрировать такие рабочие нагрузки из локального центра обработки данных в один или несколько регионов Azure. Такой процесс часто называют переносом (lift and shift).

Выставление счетов. Стоимость виртуальных машин Azure указывается за час использования, однако для формирования счетов используется поминутная тарификация. Например, если вы развернете виртуальную машину только на 23 минуты, то и оплатить вам потребуется 23 минуты. В стоимость виртуальной машины входит наценка за использование операционной системы Windows. Экземпляры с Linux немного дешевле, потому что они не требуют лицензионных отчислений. Приобретение используемого программного обеспечения и его лицензирование являются вашими обязанностями. В стоимость некоторых образов виртуальных машин, приобретаемых в Azure Marketplace (например, Microsoft SQL Server), может входить дополнительная лицензия (помимо базовой стоимости виртуальной машины).

Статус виртуальной машины непосредственно влияет на ее стоимость для пользователя.

Выполняется (Running) Виртуальная машина включена и работает нормально (а значит, подлежит оплате).

Остановлено (Stopped) Виртуальная машина остановлена, но по-прежнему развернута на физическом узле (подлежит оплате)

Остановлено (освобождено) (Stopped (Deallocated)) Виртуальная машина не развернута на физическом узле (не подлежит оплате). Стоимость устойчивого хранилища, которым пользуется виртуальная машина, взимается с пользователя отдельно. Статус виртуальной машины никак не связан со стоимостью хранилища для пользователя; даже если виртуальная машина

остановлена/освобождена и оплата за нее не взимается, необходимо оплачивать пространство, занятое дисками. По умолчанию при остановке виртуальной машины на портале Azure она переходит в состояние «Остановлено (Освобождено)» (Stopped (Deallocated)). Чтобы остановить виртуальную машину, но сохранить ее выделенной, воспользуйтесь командлетом PowerShell или интерфейсом командной строки (CLI) Azure.

Остановка виртуальной машины Azure. Чтобы остановить виртуальную машину, но оставить ее готовой к работе, воспользуйтесь командлетом PowerShell Stop-AzureRmVM. Пример команды:
Stop-AzureRmVM -Name "AzEssentialDev3" -ResourceGroup "AzureEssentials" -StayProvisioned

Для остановки классической виртуальной машины воспользуйтесь аналогичным командлетом, StopAzureVM. В Azure CLI поддерживаются две команды управления остановленным состоянием виртуальной машины: azure vm stop и azure vm deallocate.

Если выключить виртуальную машину средствами операционной системы, которая в ней выполняется, эта виртуальная машина будет остановлена, но не освобождена.

На момент написания этой лекции Microsoft предлагает соглашение об уровне обслуживания (SLA), гарантирующее возможность подключения на уровне 99,95 % для виртуальных машин из нескольких экземпляров, развернутых в группе доступности. Это означает следующее: для того чтобы соглашение об уровне обслуживания (SLA) вступило в силу, необходимо развернуть не менее двух экземпляров виртуальной машины в группе доступности. Далее в этой главе мы обсудим группы доступности виртуальных машин Azure подробнее.

Модели виртуальных машин

Существует две модели работы со многими ресурсами Azure: ДР Azure и управление службами Azure (второй подход часто называют классической моделью или ASM) [2]. Новое развертывание рекомендуется осуществлять с помощью Диспетчера Ресурсов (ДР) [3]. Классическая модель по-прежнему поддерживается, однако полный набор возможностей доступен только при использовании ДР Azure. В этой лекции будут рассмотрены обе модели, однако ориентироваться будем в основном на ДР Azure. Способы работы с виртуальными машинами Azure в рамках этих двух моделей кардинально различаются.

Модель с использованием ДР Azure предоставляет полные и тонкие возможности управления почти всеми характеристиками виртуальных машин Azure. Можно явным образом добавлять различные компоненты: сетевой адаптер, общедоступный IP-адрес, диски с данными, балансировщик нагрузки и

многие другие. Для обеспечения доступа к ресурсам Azure и возможностей управления ими ДР использует различные поставщики ресурсов. При работе с виртуальными машинами Azure ключевую роль играют три поставщика ресурсов: Сеть, Служба хранилища и Вычисление.

Поставщик ресурсов Сеть (Microsoft.Network) управляет всеми аспектами сетевых соединений: IP-адресами, балансировщиками нагрузки, сетевыми адаптерами и т. д. [4].

Поставщик ресурсов Служба хранилища (Microsoft.Storage) контролирует хранение дисков виртуальных машин Azure [5].

Поставщик ресурсов Вычисление (Microsoft.Compute) управляет характеристиками самих виртуальных машин: имена, параметры операционных систем и конфигурации (размер, количество дисков и т. д.).

Пользователю-администратору доступны не только прямые средства управления компонентами виртуальной машины, но и другие возможности ДР, например:

- развертывание логически связанных ресурсов и управление ими в составе групп ресурсов;
- теги для упорядочения и идентификации ресурсов;
- управление доступом на основе ролей (RBAC), позволяющее применять необходимые политики безопасности и контроля;
- декларативные файлы шаблонов;
- политики развертывания, обеспечивающие действие определенных правил организации;
- согласованный и централизованно управляемый (orchestrated) процесс развертывания.

Эта возможность позволяет тонко настроить среду под ваши конкретные потребности.

В рамках классической модели виртуальные машины всегда развертываются в контексте облачной службы Azure — контейнера виртуальных машин. Этот контейнер обеспечивает ряд важных возможностей: конечная точка DNS, сетевые подключения (в том числе, при необходимости, из общедоступного Интернета), безопасность, управление. Можно получить все эти возможности бесплатно (поскольку они унаследованы от модели облачных служб), но возможности управлять ими ограничены. Кроме того, в рамках классической модели недоступны дополнительные полезные функции ДР Azure (теги, файлы шаблонов и другие).

Компоненты виртуальных машин

Виртуальная машина, как и обычный компьютер, состоит из ряда компонентов, и ее также можно настроить множеством различных способов в соответствии с потребностями и желаниями владельца.

Виртуальную машину Azure иногда полезно представлять, как логическую сущность. Виртуальная машина характеризуется набором атрибутов: статус, параметры конфигурации (операционная система, процессорные ядра, память, диски, IP-адреса и т. п.) и состояние. В Azure может быть создан экземпляр этой логической сущности и выделены необходимые ресурсы, чтобы эта виртуальная машина заработала.

Данные виртуальных машин Azure хранятся на устойчивых подключенных дисках VHD. Для виртуальных машин Azure доступно два типа VHD:

- **Образ** (Image) VHD этого типа является шаблоном для создания новой виртуальной машины, поэтому часть параметров (например, имя машины, пользователь с правами администратора и т. п.) к таким дискам неприменима;
- **Диск** (Disk) VHD (возможно, загрузочный), который можно использовать в качестве подключаемого диска виртуальной машины. Диски делятся на два типа: диск ОС и диск с данными.

Для хранения устойчивых дисков (дисков ОС и дисков с данными) используются страничные BLOB-объекты в хранилище Azure. Поэтому все преимущества хранилища BLOB-объектов (высокая доступность, устойчивость, возможности обеспечения географической избыточности) относятся и к дискам. Хранилище BLOB-объектов обеспечивает механизм безопасного хранения данных, используемых виртуальной машиной. Диски можно подключать к виртуальной машине в качестве дисковых устройств. Платформа Azure использует постоянную аренду страничного BLOB-объекта, чтобы предотвратить случайное удаление страничного BLOB-объекта, который содержит VHD, соответствующий контейнер или учетную запись хранения.

Хранилища классов Standard и Premium. Для надежного хранения файлов дисков (файлов .vhd) можно использовать учетные записи хранения Azure классов Standard или Premium. В хранилище Azure класса Premium используются твердотельные накопители (SSD), обеспечивающие высокую производительность и низкую задержку, что особенно важно для виртуальных машин, на которых запущены рабочие нагрузки, интенсивно считывающие или записывающие данные. Хранилище класса Standard доступно для виртуальных машин всех размеров; хранилище класса Premium доступно для виртуальных машин серий DS, DSv2, F и GS. Хранилище класса Standard можно использовать также для виртуальных машин серий DS, DSv2, F и GS. В этом случае на

твердотельном накопителе (SSD) размещается только локальный (несохраняемый) диск.

В общем случае для нагрузок в рабочей среде (особенно чувствительных к колебаниям производительности или активно выполняющим операции ввода-вывода) рекомендуется использовать хранилище Azure класса Premium. Рабочие нагрузки, предназначенные для разработки или тестирования, часто нечувствительны к колебаниям производительности и не осуществляют операции ввода-вывода слишком активно, поэтому для них рекомендуется использовать хранилище Azure класса Standard. Подробная информация о хранилище Azure класса Premium и о том, как его использование влияет на виртуальные машины Azure, приводится в документации https://azure.microsoft.com/documentation/articles/storage-premium-storage/.

Диск ОС служит для размещения операционной системы. В случае виртуальной машины Windows диск ОС — это обычный диск C, на котором Windows размещает свои данные. Для виртуальной машины Linux это диск раздела /dev/sda1, в котором находится корневой каталог. Максимальный размер диска ОС в настоящее время составляет 1023 ГБ. Второй тип дисков, используемых в виртуальных машинах Azure, называется «диск с данными». Эти диски служат для хранения самых различных данных. Максимальный размер диска с данными также составляет 1023 ГБ. К виртуальной машине Azure можно подключить несколько дисков с данными. Их максимальное количество зависит от размера виртуальной машины и обычно составляет два диска на процессор. Диски с данными часто используются для хранения данных приложений (в том числе созданных клиентами) или серверного программного обеспечения (например, Microsoft SQL Server), а также соответствующих данных и файлов журналов. Несколько дисков с данными можно преобразовать в дисковый массив с помощью инструмента «Дисковые пространства» (Storage Spaces) в Windows или утилиты mdadm в Linux.

Кроме того, в виртуальных машинах Azure используется временный диск на физическом узле, который не сохраняется в хранилище Azure. Этот временный диск представляет собой физический диск, размещенный в корпусе сервера. Временный жесткий диск может быть стандартным жестким диском или твердотельным накопителем (SSD), в зависимости от типа созданной виртуальной машины. Временный диск следует использовать только для размещения временных (или реплицированных) данных, потому что в случае сбоя физического узла или при остановке/освобождении виртуальной машины его содержимое будет удалено. На рис. 16.1 показаны различные типы дисков.

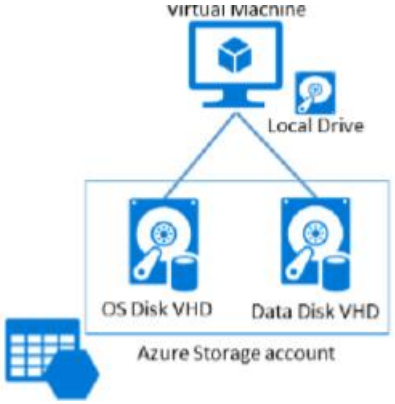

Рисунок 8.1 – Типы дисков в виртуальных машинах Azure.

Виртуальная сеть

В локальной физической инфраструктуре может содержаться множество компонентов, которые позволяют использовать масштабируемые и безопасные методы работы с виртуальными машинами. Вот некоторые из таких ресурсов: отдельные сетевые пространства для серверов, взаимодействующих с Интернетом, и для служебных серверов, балансировщики нагрузки, брандмауэры и многое другое. Многие из этих компонентов можно логически развернуть в виртуальной сети Azure (часто такие сети называют VNET).

Виртуальная сеть Azure поддерживает многие аналогичные функции, например:

- **Подсеть** (Subnet) Подсетью называется диапазон IP-адресов, относящихся к виртуальной сети. Виртуальную машину необходимо разместить в подсети, входящей в VNET. Виртуальные машины, размещенные в некоторой подсети VNET, могут свободно обмениваться данными с виртуальными машинами из другой подсети той же виртуальной сети. Вы можете управлять таким взаимодействием с помощью групп безопасности сети (NSG) и настраиваемых маршрутов;
- **IP-адрес** (IP-address) IP-адреса бывают общедоступными и частными. Общедоступный IP-адрес позволяет виртуальной машине принимать данные из Интернета. Такой адрес может выделяться динамически, то есть создаваться при запуске соответствующего ресурса (например, виртуальной машины или балансировщика нагрузки) и освобождаться при его остановке, либо статически, то есть назначаться немедленно и сохраняться до удаления ресурса. Частными IP-адресами называются адреса, не маршрутизируемые в сети Интернет. Они служат для обмена данными между виртуальными машинами и балансировщиками нагрузки в рамках одной сети VNET;

- **Балансировщик нагрузки** (Load Balancer) Доступ к виртуальным машинам со стороны узлов Интернета или других виртуальных машин в составе сети VNET обеспечивается балансировщиками нагрузки Azure. Существует два типа балансировщиков нагрузки:
  - **Внешний балансировщик нагрузки** (External Load Balancer) Используется для обеспечения высокой доступности нескольких виртуальных машин для узлов Интернета;
  - **Внутренний балансировщик нагрузки** (Internal Load Balancer) Используется для обеспечения высокой доступности нескольких виртуальных машин для других виртуальных машин той же сети VNET;
- **Группа безопасности сети** (Network Security Group) Группа безопасности сети (NSG) позволяет создавать правила, которые управляют входящим и исходящим сетевым трафиком (разрешают или отклоняют его) для сетевых адаптеров виртуальной машины или подсетей.

При создании виртуальной машины Azure с помощью ДР необходимо поместить ее в виртуальную сеть Azure (VNET). Администратор сам решает, использовать ли существующую сеть VNET или создать новую, в какой подсети разместить машину, нужен ли балансировщик нагрузки, а также выбирает IP-адрес, количество сетевых адаптеров и способ обеспечения безопасности сети (см. рис. 8.2). Может показаться, что это сильно осложняет развертывание виртуальной машины, однако все эти параметры очень важны для ее доступности и безопасности.

Классические виртуальные машины также можно поместить в виртуальную сеть Azure, но это необязательное требование (тогда как для виртуальных машин в модели с использованием ДР — обязательное).

IP-адрес

В модели с использованием ДР у виртуальной машины по умолчанию нет IP-адреса. IP-адрес необходимо назначить виртуальной машине явным образом через подключенный к ней сетевой адаптер. Чтобы обмениваться данными с другими виртуальными машинами в виртуальной сети или с узлами общедоступного Интернета, виртуальной машине нужен IP-адрес.

Каждому сетевому адаптеру соответствует частный адрес (его часто называют DIP или динамическим IP). Он служит для подключения к виртуальной сети и может быть сопоставлен с общедоступным IP-адресом, который делает возможным прямое подключение к общедоступному Интернету.

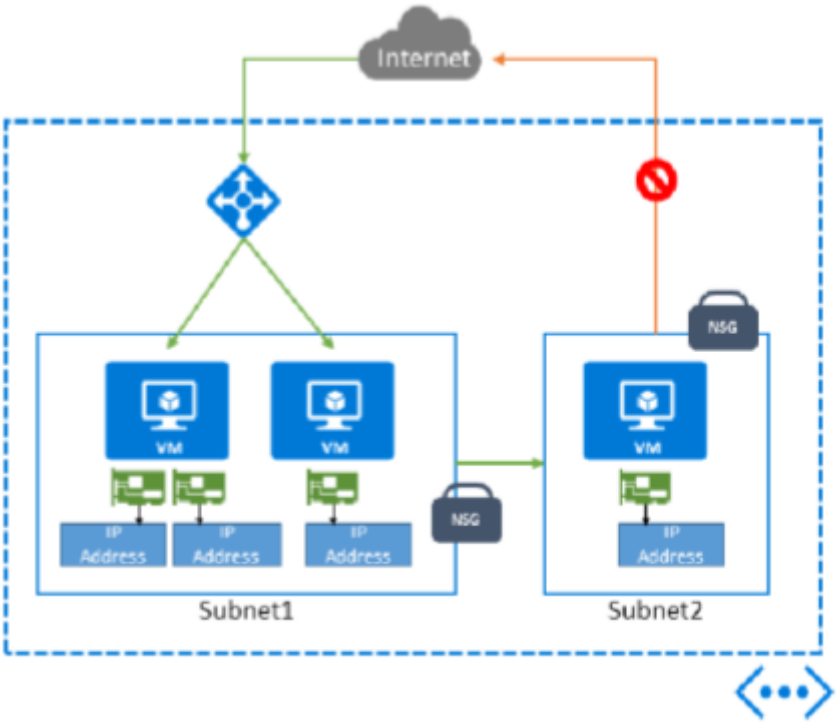

Рисунок 8.2 – Виртуальные машины в рамках модели с использованием ДР явным образом управляют соответствующими сетевыми компонентами.

По умолчанию при остановке/освобождении виртуальной машины эти динамические IP-адреса сбрасываются, однако и виртуальную машину, и адрес можно сделать статическими, чтобы они сохранялись и после отключения/освобождения виртуальной машины. Это удобно в том случае, если виртуальной машине необходим постоянный DIP-адрес (примеры: виртуальные машины Microsoft SQL Server, виртуальные машины, используемые в качестве серверов DNS, и постоянные общедоступные IP-адреса). Если требуется назначить виртуальной машине несколько DIP-адресов (например, чтобы разместить ее в нескольких подсетях), то к ней можно подключить несколько сетевых адаптеров с различными DIP-адресами.

В классической модели все примерно так же, с одним отличием: сетевые адаптеры и общедоступные IP-адреса не являются независимыми ресурсами — они могут существовать только в контексте виртуальной машины. Более того, в классической модели подключение к Интернету обычно осуществляется не через общедоступный IP-адрес, а посредством балансировщика нагрузки Azure.

Балансировщик нагрузки Azure

Балансировщик нагрузки Azure используется для того, чтобы обеспечить примерно равное распределение трафика между несколькими виртуальными машинами (эти машины часто сконфигурированы похожим образом или связаны

между собой логически). Балансировщик нагрузки позволяет обеспечить взаимодействие нескольких виртуальных машин — например, в коллекции веб серверов в среде веб-фермы. Запросы, поступающие для набора виртуальных машин с балансировкой нагрузки, не направляются одной конкретной виртуальной машине, а распределяются между доступными виртуальными машинами.

В Azure доступно два типа балансировщиков нагрузки: внешний и внутренний (см. рисунок 8.3). Внешний балансировщик нагрузки служит для управления трафиком из Интернета (позволяя направить его одной виртуальной машине или распределить между несколькими). С его помощью можно обеспечить высокую доступность приложения и при необходимости быстро масштабировать среду.

Внутренний балансировщик нагрузки служит для распределения трафика, поступающего из виртуальной сети на набор виртуальных машин. Это может быть, например, трафик для веб-API или кластера баз данных, который должен быть доступен только для веб-серверов переднего плана, но не для всех узлов общедоступного Интернета.

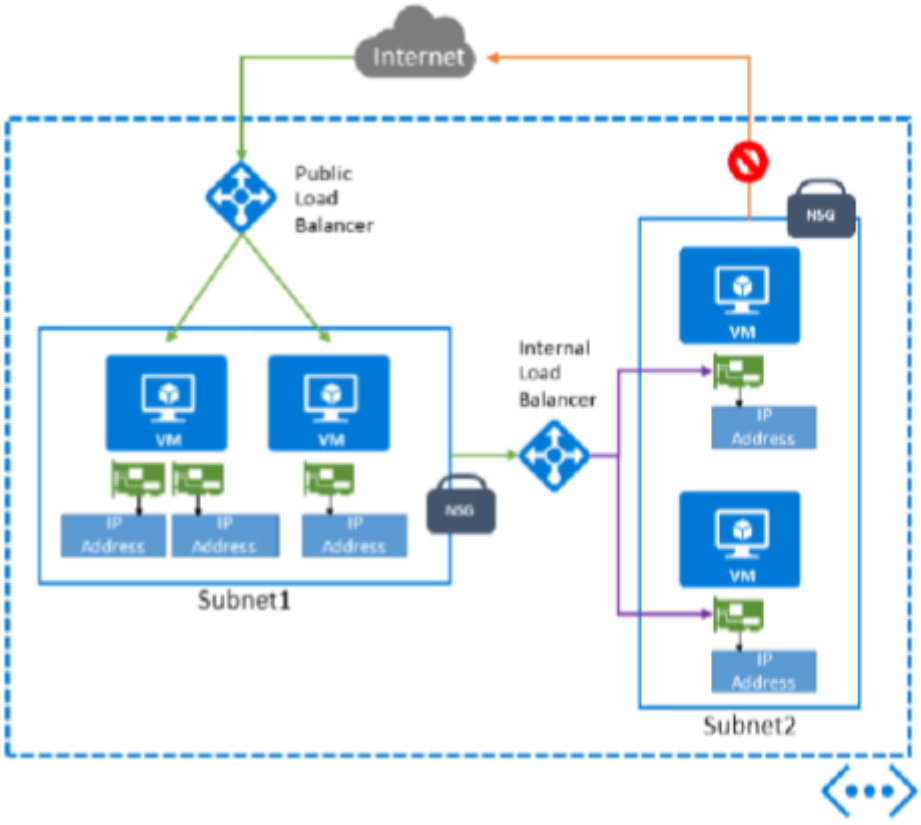

Рисунок 8.3 – Использование внешнего и внутреннего балансировщика нагрузки.

В модели развертывания с помощью Диспетчера Ресурсов Azure перед тем, как использовать балансировщик нагрузки, необходимо создать несколько дополнительных объектов:

- общедоступные IP-адреса для входящего сетевого трафика (в случае внешнего балансировщика нагрузки);
- пул служебных (частных) IP-адресов, назначенных сетевым адаптерам виртуальных машин;
- правила, определяющие соответствие между общедоступными портами балансировщика нагрузки и портами служебного пула;
- правила NAT для входящих соединений, определяющие соответствие между общедоступными портами балансировщика нагрузки и конкретными виртуальными машинами в пуле;
- проверки работоспособности, позволяющие контролировать функциональность виртуальных машин в составе пула.

В классической модели внешний балансировщик нагрузки предоставляется автоматически в рамках модели облачных служб. Все виртуальные машины, которые относятся к облачной службе и открывают конечную точку, доступную для подключения через Интернет, автоматически конфигурируются так, чтобы использовать балансировщик нагрузки. Классические виртуальные машины также могут использовать внутренний балансировщик нагрузки.

Сетевой адаптер

Сетевой адаптер обеспечивает доступ к ресурсам в виртуальной сети Azure через сеть. Сетевой адаптер является самостоятельным ресурсом, но для обеспечения доступа к сети его необходимо сопоставить с виртуальной машиной. Максимальное количество сетевых адаптеров, которые можно подключить к виртуальной машине, зависит от размера выбранной виртуальной машины.

При работе с сетевыми адаптерами и виртуальными машинами следует помнить несколько важных вещей:

- IP-адрес каждого сетевого адаптера виртуальной машины должен принадлежать подсети VNET, к которой относится виртуальная машина;
- если одной виртуальной машине назначено несколько сетевых адаптеров, то назначить общедоступный IP-адрес можно только основному адаптеру. Каждому сетевому адаптеру назначается частный IP-адрес (кроме ситуации, когда сетевой адаптер является основным и имеет общедоступный IP-адрес). Сетевые адаптеры могут относиться к различным подсетям в составе сети VNET;
- любой сетевой адаптер виртуальной машины можно добавить в группу безопасности сети (NSG).

При работе с классическими виртуальными машинами беспокоиться о конфигурации сетевых адаптеров не нужно, потому что она создается автоматически в рамках модели облачной службы и не может существовать вне контекста виртуальной машины.

Группы безопасности сети

Группы безопасности сети (NSG) позволяют явным образом задать детальные правила, контролирующие потоки входящего и исходящего сетевого трафика виртуальных машин и подсетей Azure. Группы безопасности сети (NSG) позволяют управлять потоками сетевого трафика, входящего в вашу среду и исходящими из нее. Вы создаете правила, в которых указывается IP-адрес и порт отправителя и получателя. Правила групп безопасности сети (NSG) могут применяться к виртуальным машинам и (или) к подсетям. В случае с виртуальной машиной группа безопасности сети (NSG) ассоциируется с сетевым адаптером, подключенным к этой виртуальной машине.

Группы доступности

Виртуальные машины Azure размещаются на физических серверах, которые находятся в центрах обработки данных Microsoft Azure. Они, как и любые другие физические устройства, могут ломаться. При сбое физического сервера виртуальные машины Azure, размещенные на этом сервере, также перестанут работать. В случае сбоя платформа Azure перенесет виртуальные машины на работоспособный узел и запустит их. Восстановление работоспособности служб может занять несколько минут. В течение этого периода приложения, размещенные на этих виртуальных машинах, будут недоступны.

Помимо аппаратных сбоев на функционирование виртуальных машин также могут влиять периодические обновления, которые инициирует сама платформа Azure. Корпорация Microsoft периодически обновляет операционную систему узлов, на которых выполняются виртуальные машины (однако установка исправлений ОС для гостевых виртуальных машин, которые вы создаете, остается вашей задачей). В ходе этих обновлений виртуальные машины перезагружаются, и поэтому они недоступны некоторое время.

Для того чтобы в вашей инфраструктуре не присутствовала единая точка отказа, рекомендуется развернуть несколько экземпляров виртуальной машины. Более того, соглашение об уровне обслуживания (SLA) действует лишь в том случае, если в группе доступности развернуто не менее двух виртуальных машин Azure. Это логическая функция, которая используется для того, чтобы гарантировать, что группа связанных между собой виртуальных машин развертывается таким образом, чтобы исключить их одновременный сбой или

недоступность ввиду обновления операционной системы в центре обработки данных. Первые две виртуальные машины, развернутые в группе доступности, помещаются в два различных домена сбоя (fault domains). Благодаря этому сбой в каком-либо центре данных не затронет обе машины одновременно. Аналогичным образом первые пять виртуальных машин, развернутые в группе доступности, помещаются в пять различных доменов обновления (update domains), что минимизирует влияние обновления операционных систем узлов в Azure на работоспособность виртуальных машин (обновление выполняется в различных доменах поочередно). В одну группу доступности следует помещать виртуальные машины, выполняющие одинаковые функции.

Количество доменов сбоя и доменов обновления зависит от модели развертывания (с помощью Диспетчера Ресурсов или классической модели). В модели с использованием Диспетчера Ресурсов может использоваться до 3 доменов сбоя и до 20 доменов обновления. В классической модели может использоваться 2 домена сбоя и 5 доменов обновления.

Ключевые термины:

Служба «Виртуальные машины Azure» – служба, которая позволяет разворачивать виртуальные машины под управлением Windows или Linux в центре обработки данных Microsoft Azure.

Несохраняемые диски – диски, подключенные к физическому узлу, данные на которых могут быть утеряны в случае сбоя физического узла.

Поставщик ресурсов Сеть – ресурс, который управляет всеми аспектами сетевых соединений.

Поставщик ресурсов Служба хранилища – ресурс, который контролирует хранение дисков виртуальных машин

Поставщик ресурсов Вычисление – ресурс, который управляет характеристиками самих виртуальных машин: имена, параметры операционных систем и конфигурации

Вопросы для самопроверки

1. Какие существуют модели виртуальных машин Azure?
2. Какие существуют поставщики ресурсов для виртуальных машин Azure?
3. Как производится хранение виртуальных машин Azure?
4. Поясните назначение Хранилища классов Standard и Premium.
5. Поясните назначение виртуальной сети Azure.
6. Поясните назначение Балансировщик нагрузки для виртуальной сети Azure.

7. Для чего применяют Группы безопасности сети виртуальной сети Azure.
8. Поясните назначение Сетевой адаптер в Azure.
9. Поясните назначение Групп доступности для виртуальной сети Azure.
10. Для чего применяется подсеть в Виртуальной сети Azure?

Литература

1. Windows virtual machines in Azure. https://docs.microsoft.com/en-us/azure/virtual-machines/windows/overview.
2. Серия виртуальных машин. https://azure.microsoft.com/ru-ru/pricing/details/virtual-machines/series/
3. What is Azure Resource Manager? https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview
4. Microsoft global network. https://docs.microsoft.com/en-us/azure/networking/microsoft-global-network
5. Introduction to the core Azure Storage services. https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction

Лекция 9. Частное облако: архитектура и средства управления частным облаком, миграция приложений в облако

Краткая аннотация лекции.

В данной лекции рассматривается служба приложений Azure, которая включает в себя веб-приложения, приложения Logic Apps, мобильные приложения, приложения API и приложения-функции. Особое внимание уделяется веб-приложениям и их взаимодействию со службой приложений.

Цель лекции.

Целью данной лекции является ознакомление со службой приложений Azure.

Служба приложений и планы службы приложений

Служба приложений — это служба, предназначенная для размещения приложений пяти типов [1]:

- веб-приложения;
- мобильные приложения;
- Logic Apps;
- приложения API;
- приложения-функции.

Каждое приложение функционирует на базе отдельной службы приложений. Если вам потребуется найти запись вашего веб-сайта на портале Azure, то вам нужно будет искать службу приложений, в которой он выполняется. Имя этой службы соответствует имени размещенного приложения, что довольно удобно.

План службы приложений

План службы приложений определяет набор и объем ресурсов, доступных для одной или нескольких служб приложений, которые сопоставлены с этим планом [2]. При создании плана службы приложений вы можете задать ряд параметров. Некоторые из них перечислены ниже.

- расположение (например, западная часть США);
- количество экземпляров;
- ценовая категория (например, Free, Standard или Premium) — от нее зависят различные параметры производительности и обслуживания:

- количество ядер или размер экземпляра;
- объем оперативной памяти;
- емкость хранилища;
- максимальное количество экземпляров;
- параметры автоматического масштабирования (в зависимости от уровня — автоматически, вручную или отключено)

При первом развертывании службы приложений выбирается план службы приложений, который будет использоваться. В ходе развертывания можно выбрать ранее созданный план службы приложений либо подготовить новый.

В рамках подхода «инфраструктура как услуга» (IaaS) вы можете создавать виртуальные машины, развертывать в них приложения, настраивать IIS и пулы приложений, а также выполнять многие другие действия. В этом случае при каждом изменении приложения вам потребуется развертывать его на всех виртуальных машинах заново. Если вы увеличите масштаб до четырех или восьми виртуальных машин, эта задача уже становится трудоемкой. В рамках подхода IaaS обслуживанием службы и ее управлением вы занимаетесь самостоятельно. Именно здесь приходят на помощь планы службы приложений. Они позволяют выполнять несколько приложений на одном наборе виртуальных машин, даже если каждое приложение развертывается отдельно.

Предположим, вы хотите разместить пять веб-сайтов и три мобильных приложения. Вы можете запустить каждое из этих решений на отдельной виртуальной машине — потребуется восемь виртуальных машин. Для организации резервирования (что рекомендуется сделать) понадобится 16 виртуальных машин. Даже если вы выберете маленькие экземпляры, суммарная стоимость вырастет очень быстро. Кроме того, каждый набор виртуальных машин потребуется масштабировать отдельно.

Есть другой вариант, гораздо более экономически эффективный и удобный в управлении: выполнять все эти приложения на одном наборе из двух виртуальных машин. Такие наборы очень просто организовать с помощью служб приложений Azure. Вы подготавливаете план службы приложений, указывая размер виртуальной машины, количество экземпляров и другие параметры. Затем вы развертываете восемь приложений, указывая для каждого из них один и тот же план службы приложений. В результате все восемь приложений будут выполняться на одном наборе из двух виртуальных машин. Вы сможете развертывать и обновлять каждое приложение по необходимости: обновлять их все одновременно не потребуется.

При создании плана службы приложений платформа выделяет вам запрошенные ресурсы. При развертывании приложения в этот план службы приложений оно просто запускается на базе этих выделенных ресурсов.

Если вы решите увеличить количество виртуальных машин с двух до четырех, для этого достаточно будет войти на портал Azure и изменить количество экземпляров в плане службы приложений с 2 на 4. Платформа создаст еще две виртуальные машины и развернет на них ваши приложения. Если вы хотите перейти от небольших виртуальных машин к средним, просто измените ценовую категорию в плане службы приложений.

Когда веб-приложения выполняются в службе приложений с использованием плана службы приложений, все операции управления выполняются автоматически, и вы можете быстро изменить масштаб среды, просто скорректировав параметры плана службы приложений.

Создание плана службы приложений на портале Azure

Сейчас мы научимся создавать план службы приложений на портале Azure [3]. Затем мы узнаем, как создать веб-приложение и развернуть его в службе приложений с использованием этого плана службы приложений.

1. Войдите на портал Azure.
2. Нажмите «Создать» (New), затем «Показать все» (See all), как показано на рис. 14.1.

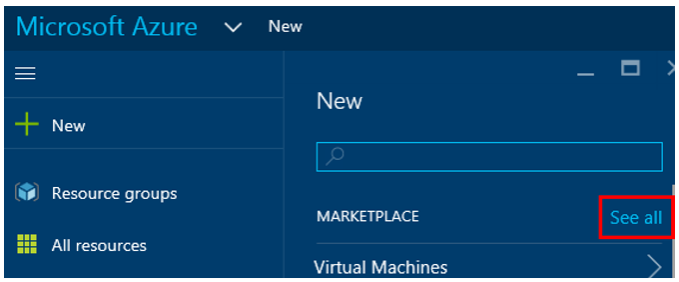

Рисунок 9.1 – Перейдите в Marketplace и найдите ресурс, который требуется добавить

3. Отобразится экран поиска по Marketplace (рис. 9.2). Введите в поле поиска план службы приложений (app service plan) и нажмите клавишу «Ввод» (Enter).

Рисунок 9.2 – Marketplace, поле ввода поискового запроса

4. В результатах поиска выберите пункт «План службы приложений» (App Service Plan), как показано на рис. 9.3.

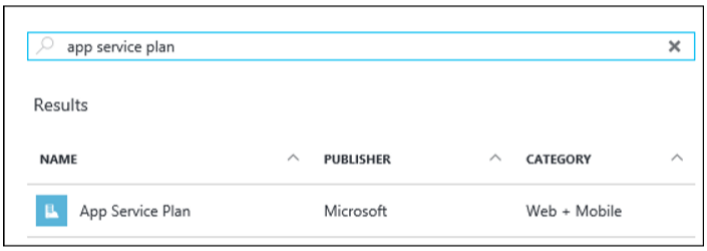

Рисунок 9.3 – Результат поиска по запросу «план службы приложений» (app service plan)

5. Нажмите «Создать» (Create) в колонке «План службы приложений» (App Service Plan). Она показана на рис. 9.4.

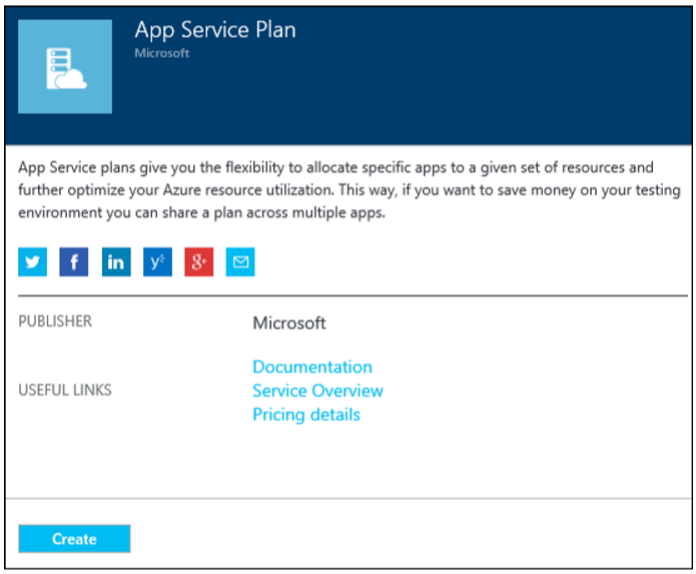

Рисунок 9.4 – Нажмите «Создать» (Create), чтобы создать новый план службы приложений

6. После этого отобразится колонка «План службы приложений» (App Service plan) (ее примерный вид показан на рис. 9.5). В ней вы можете задать параметры плана службы приложений.

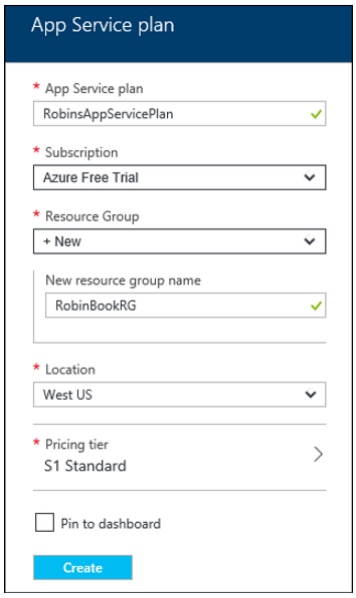

Рисунок 9.5 – Поля, которые потребуется заполнить для создания плана службы приложений

- **План службы приложений** (App Service Plan) Введите в этом поле название плана службы приложений. Рекомендуется выбрать имя, по которому вы сможете быстро понять назначение плана при дальнейшем использовании.
- **Подписка** (Subscription) Если ваша учетная запись позволяет администрировать несколько подписок, то в этой части колонки отображается раскрывающийся список, в котором можно выбрать требуемую подписку.
- **Группа ресурсов** (Resource Group) Группы ресурсов представляют собой логические контейнеры для ресурсов, связанных между собой. Например, все ресурсы, которые вы создадите при работе над этой книгой, можно поместить в одну группу ресурсов. Когда вы закончите освоение материала, вы сможете удалить группу ресурсов, и тогда отмена распределения и удаление всех ресурсов в ней будут выполнены автоматически. Создадим новую группу ресурсов для нашего плана службы приложений. Далее в ходе этой главы мы создадим веб-приложение и сопоставим его с нашим планом службы приложений. Оставьте значение по умолчанию и укажите имя новой группы ресурсов. Для группы ресурсов рекомендуется выбрать имя, которое отражало бы ее назначение.

– **Расположение** (Location) Регион Azure, в котором будет размещена группа ресурсов, в том числе метаданные (например, журналы аудита) о том, где находится каждый ресурс группы. Эти характеристики зависят от типа ресурсов. Они важны в тех случаях, когда имеет значение, где именно размещаются данные, — например, если вы работаете в странах, где действуют законы о локальном хранении данных. Кроме того, операции диспетчера ресурсов реализуются через этот регион, поэтому рекомендуется выбирать здесь тот регион, в котором размещена большая часть ресурсов группы. В рамках этого примера выберите ближайший регион.
– **Ценовая категория** (Pricing tier) Нажмите в этом поле, чтобы открыть список доступных вариантов. Отобразится новая колонка (рисунок 2-6), в которой перечислены рекомендуемые ценовые планы (из всех доступных ценовых категорий). Чтобы просмотреть все планы, нажмите «Показать все» (View all) в этой колонке. Ценовой план позволяет задать объем хранилища, масштабируемость, настройки резервного копирования и другие параметры.

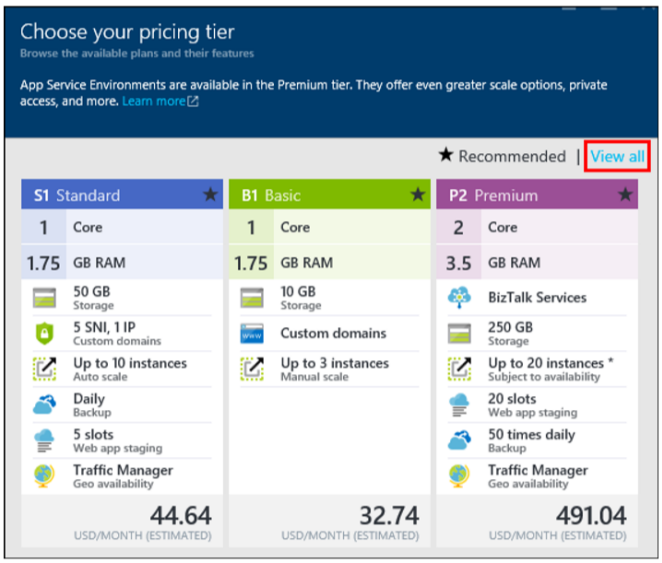

Рисунок 9.6 – Колонка «Ценовая категория» (Pricing Tier)

Выберите вариант «S1, ценовой план Standard», затем нажмите «Выбрать» (Select) в нижней части колонки. Теперь в столбце «План службы приложений» (App Service Plan) должен отображаться выбранный ценовой план.

7. Установите в нижней части колонки «План службы приложений» (App Service Plan) флажок «Закрепить на панели мониторинга» (Pin To Dashboard). Тогда ваш план службы приложений будет отображаться на панели мониторинга, и с ним будет удобно работать. Нажмите «Создать» (Create). Платформа создаст план и добавит плитку на вашу панель мониторинга.
8. После того как план службы приложений будет создан, вы сможете открыть его, выбрав плитку на панели мониторинга, и изменить его. Там же показано, какие приложения используют этот план. Теперь мы создадим и развернем приложение, а потом покажем, как можно масштабировать приложения посредством масштабирования плана службы приложений.

После этого вы сможете создать одну или несколько служб приложений (например, веб-приложений) и сопоставить их с этим планом службы приложений. Все эти службы будут выполняться на одних и тех же виртуальных машинах.

Создание и развертывание веб-приложений

Итак, мы узнали, что такое службы приложений и планы служб приложений. Сейчас мы расскажем вам, что такое веб-приложение, обсудим некоторые его характеристики и поговорим о доступных возможностях при их создании. Затем мы покажем, как ими можно воспользоваться, чтобы создать и развернуть веб-приложение [4].

Веб-приложение

Веб-приложение (в терминологии Azure) — это веб-приложение (в общем понимании), размещенное в службе приложений. Служба приложений — это управляемая служба Azure, позволяющая быстро развернуть веб-приложения и сделать их доступными для пользователей через Интернет. Как мы уже говорили, вам не нужно непосредственно администрировать виртуальные машины, на которых выполняется ваше веб-приложение; платформа обслуживает их за вас. Более того, у вас вообще нет доступа к этим виртуальным машинам.

В Azure поддерживаются следующие языки программирования: .NET, Java, PHP, Node.js и Python. Вам не обязательно создавать собственное веб-приложение: можно использовать в качестве отправной точки одно из подготовленных веб-приложений (в числе поддерживаемых — WordPress, Umbraco, Joomla! и Drupal).

Вы можете организовать непрерывное развертывание, используя Team Foundation Server (TFS), GitHub, TeamCity, Jenkins или BitBucket, чтобы каждый раз при фиксации изменения развертывалась новая версия веб-приложения.

Масштабирование веб-приложения осуществляется путем масштабирования плана службы приложений, к которому оно относится. Количество экземпляров можно увеличивать и уменьшать по необходимости. Вы можете задать параметры автоматического масштабирования, руководствуясь параметрами производительности (например, загрузкой процессора). Также вы можете опубликовать свой веб-сайт в нескольких регионах и с помощью диспетчера трафика Azure управлять маршрутизацией трафика между ресурсом и расположением, ближайшим к вашему клиенту.

Для целей диагностики можно собирать статистику о производительности, а также вести журналы приложений, веб-серверов и IIS (в том числе журналы неудачных запросов к IIS). Если вы используете Microsoft Visual Studio, то сможете выполнять удаленную отладку работающего в облаке приложения.

Как видите, Azure поддерживает множество средств удобного развертывания, управления веб приложением и устранения его неполадок.

Способы создания веб-приложений

В Azure поддерживается множество способов создания веб-приложений и развертывания содержимого в службе приложений. Мы рассмотрим некоторые из них, в том числе следующие.

- **Azure Marketplace**. На этом портале доступны все ресурсы, которые можно развернуть в Azure. Мы покажем, как с помощью Marketplace создать веб-приложение на основе готового шаблона (например, WordPress).
- **Visual Studio Code.** Это бесплатный кроссплатформенный редактор программного кода (с открытым исходным кодом), в котором поддерживаются возможности отладки.
- **Visual Studio.** Это полнофункциональная интегрированная среда разработки, разработанная корпорацией Microsoft.

Marketplace. В Azure Marketplace доступно множество готовых веб-сайтов и шаблонов. Чтобы открыть список всех доступных вариантов, войдите на портал Azure и нажмите «Создать» > «Интернет + мобильные устройства» > «Показать все» (New > Web + Mobile > See All). Отобразится колонка Marketplace со списком приложений категории «Интернет + мобильные устройства» (Web + Mobile), как показано на рис. 9.7.

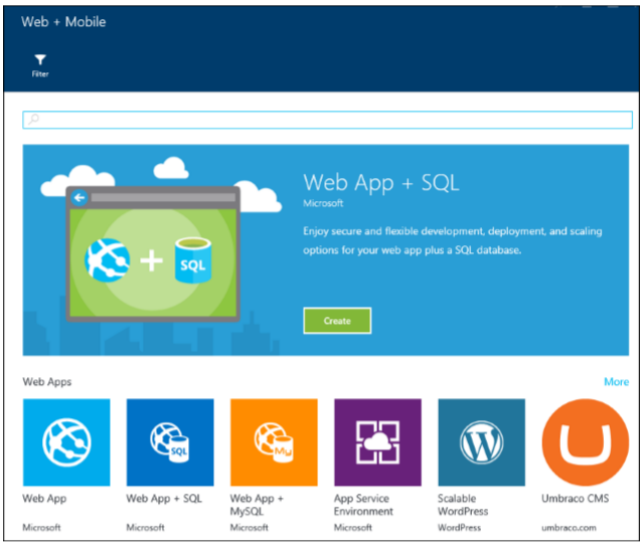

Рисунок 9.7 – Доступные в Azure Marketplace приложения категории «Интернет + мобильные устройства» (Web + Mobile)

Прокрутите страницу вниз, чтобы открыть список категорий. Выберите ссылку «Еще» (More) в конце любой строки, чтобы просмотреть другие предложения в соответствующей категории. Вот лишь некоторые из доступных вариантов:

- **Веб-приложения** (Web Apps) Веб-приложение, Веб-приложение и SQL, Веб-приложение и MySQL, WordPress и Umbraco CMS
- **Блоги и CMS** (Blogs + CMSs) Joomla!, Drupal, DNN, Orchard CMS, Umbraco CMS и MonoX
- **Starter web apps** ASP.NET, HTML5, Node.js, PHP, Apache Tomcat и ряд готовых приложенийпримеров, например, веб-приложения Bakery и Java Coffee Shop

Visual Studio Code. Visual Studio Code (VS Code) — бесплатный редактор программного кода (с открытым исходным кодом), поддерживающий набор инструментов разработки, в числе которых — отладка, выполнение задач и управление версиями. Доступны версии для Windows, OS X и Linux.

VS Code упрощает отладку программ благодаря таким функциям, как автоматическое дополнение программного кода IntelliSense и удобные средства рефакторинга кода. Он поддерживает интеграцию с Git, с диспетчерами пакетов, репозиториями и различными средствами сборки.

VS Code изначально поддерживает Node.js, JavaScript и TypeScript. Установив требуемые расширения, вы сможете использовать VS Code для отладки программ на других языках, например, C#, C++, Python, Ruby и PowerShell. В его составе также доступны инструменты для работы с веб-технологиями, в том числе HTML, CSS, JSON и Markdown.

На портале Azure вы можете настроить свое веб-приложение так, чтобы оно получало исходный код из OneDrive, Dropbox или локального репозитория, например, GitHub или Visual Studio Team Service. Если включить для веб-приложения непрерывное развертывание, то при каждом изменении исходного кода в вашем репозитории обновления будут публиковаться автоматически.

Visual Studio. Visual Studio — это полнофункциональная среда разработки, позволяющая создавать самые различные приложения, в том числе приложения ASP.NET MVC, клиентские приложения .NET, службы Windows Communication Foundation (WCF), Web API и облачные службы на различных языках, например, C#, C++, VB, F# и XAML. Кроме того, с помощью Visual Studio можно создавать веб-приложения и публиковать их в службе приложений Azure. Мы покажем это на следующем примере.

Пример: создание веб-приложения с помощью Azure Marketplace

Давайте посмотрим, как можно создать веб-приложение на основе одного из шаблонов, доступных в Azure Marketplace.

1. Войдите на портал Azure. Нажмите «Создать» (New) в левой части страницы, затем нажмите «Показать все» (See all), как показано на рис. 9.8.

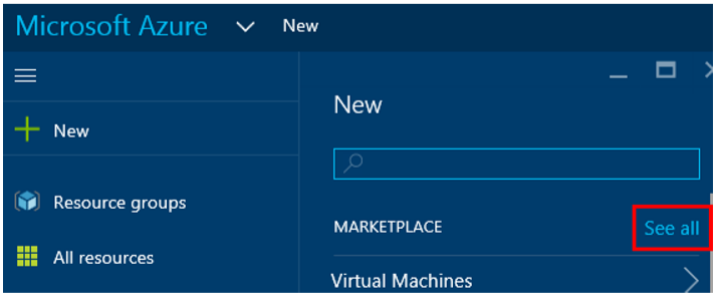

Рисунок 9.8 – Переход к колонке «Поиск» (Search) в Marketplace

2. Отобразится поле поиска по Marketplace. В Marketplace опубликованы все ресурсы, которые можно развернуть в Azure: виртуальные машины, виртуальные сети, учетные записи хранения, веб приложения и многое другое. Введите WordPress и нажмите клавишу «Ввод» (Enter), чтобы выполнить поиск (см. рис. 9.9).

Рисунок 9.9 – Поиск по запросу «WordPress».

3. Отобразится список совпадений, как показано на рис. 9.10.

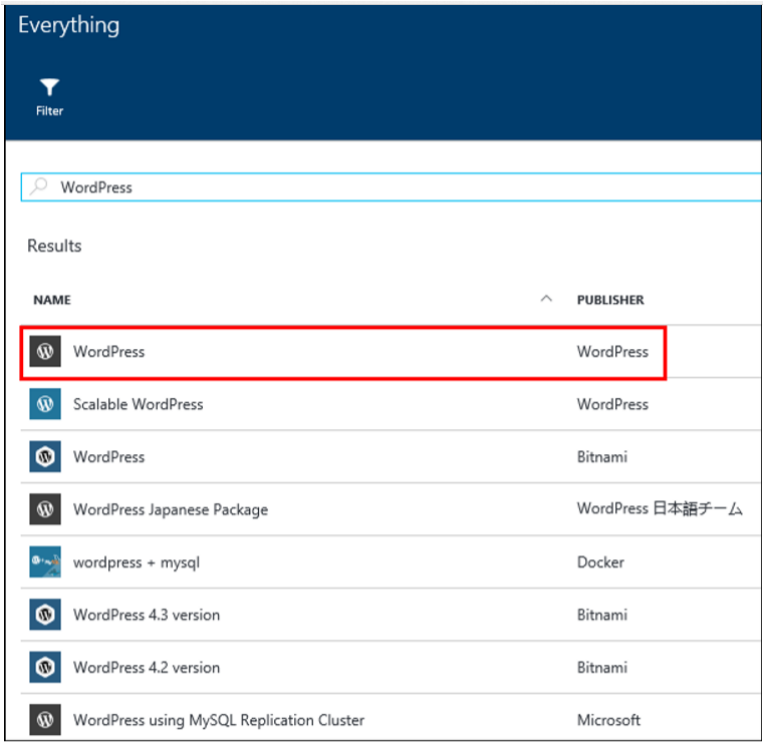

Рисунок 9.10 – Результаты поиска по запросу «WordPress»

4. Выберите строку, в которой в столбцах «Имя» (Name) и «Издатель» (Publisher) указано «WordPress». Откроется колонка WordPress. Нажмите кнопку «Создать» (Create) в ее нижней части, чтобы создать сайт WordPress. Откроется колонка, в которой вы можете изменить конфигурацию сайта WordPress. Она показана на рис. 9.11.

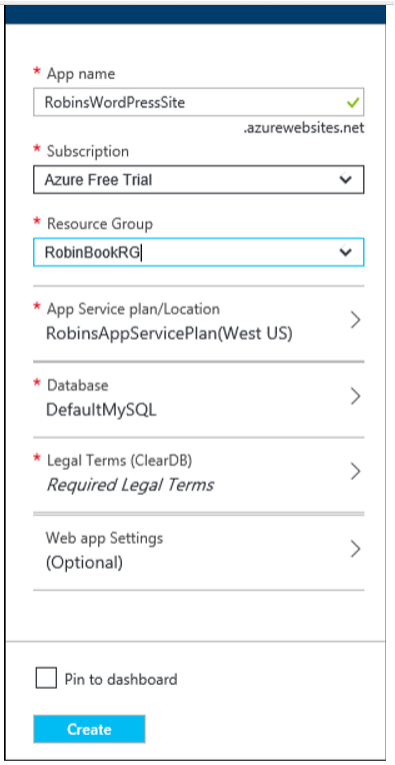

Рисунок 9.11 – Создание веб-сайта WordPress

5. Заполним поля в этой колонке:

– **Имя приложения** (App name) Оно будет использоваться в URL-адресе для доступа к вашему веб-приложению.

– **Подписка** (Subscription) Если учетная запись, которую вы используете, позволяет управлять несколькими подписками, то в этом поле вы сможете выбрать требуемую подписку.

– **Группа ресурсов** (Resource Group) Группа ресурсов служит для объединения нескольких связанных между собой ресурсов, например, веб-приложения и базы данных. Выберите группу ресурсов, которую вы использовали для ранее созданного плана службы приложений.

– **План службы приложений** (App Service Plan) Выберите план службы приложений, который вы создали ранее в ходе этой главы. Нажмите «База данных» (Database), чтобы просмотреть параметры базы данных, как показано на рис. 9.12. По умолчанию для WordPress используется СУБД MySQL. В соответствующих полях укажите имя и тип базы данных (общая или выделенная). В поле «Расположение» (Location) выберите регион, в котором будет выполняться приложение.

Нажмите «Ценовая категория» (Pricing Tier) и выберите самый дешевый вариант из доступных (на момент написания книги это Mercury). Нажмите OK, чтобы сохранить параметры базы данных

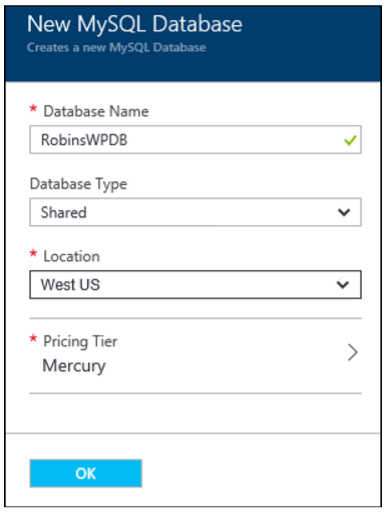

Рисунок 9.12 – Параметры базы данных

– вернитесь к колонке «Настройки WordPress» (WordPress Settings) для вашего нового веб-сайта и выберите пункт «Условия» (Legal Terms). Если вы согласны с условиями использования, нажмите OK в нижней части экрана, чтобы подтвердить это;
– чтобы изменить настройки WordPress, можно выбрать пункт «Настройки веб-приложения (необязательные)» (Web App Settings (Optional), как показано на рис. 9.13. Это необязательный этап;

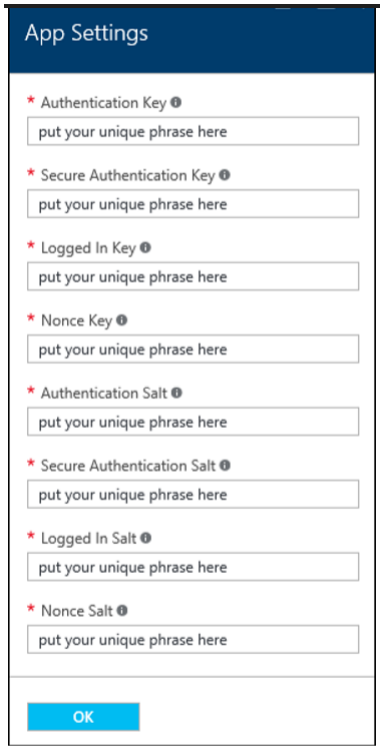

Рисунок 9.13 – Задание настроек приложения (необязательный этап)

– вернитесь в колонку WordPress, установите флажок, чтобы закрепить новое веб-приложение на панели мониторинга, и нажмите «Создать» (Create). В Azure будет создан для вас сайт WordPress.

6. После того как завершится публикация веб-приложения, нажмите на плитку на панели мониторинга, чтобы открыть ее свойства, как показано на рис. 9.9. Чтобы открыть сайт, нажмите на его URLадрес. Платформа запросит у вас сведения, необходимые для создания сайта WordPress, такие как язык, заголовок сайта, имя пользователя, пароль и адрес электронной почты. Заполните все поля и нажмите кнопку «Установить WordPress» (Install WordPress). Дождитесь завершения установки WordPress.

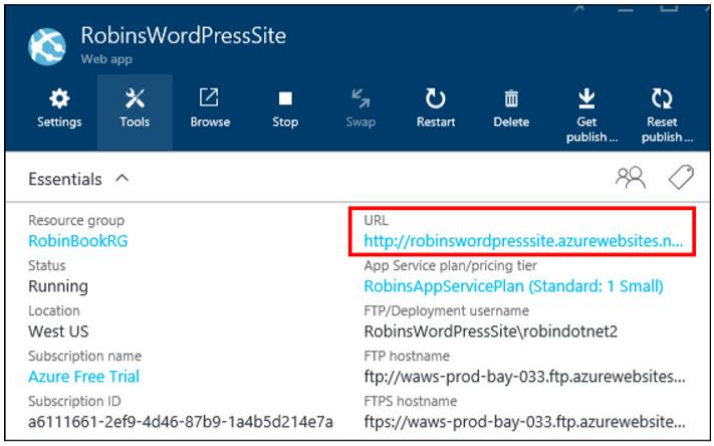

Рисунок 9.14 – Нажмите на URL-адрес созданного веб-сайта WordPress, чтобы открыть его

Экземпляры Application Insights выделены в нем прямоугольниками. Обратите внимание: их значок отличается от значка веб-приложений. Просто выберите эти ресурсы Application Insights и удалите их (рис. 9.15). (При выборе этого ресурса откроется несколько колонок. Закрывайте их, пока не вернетесь к первой, а в ней выберите «Удалить» (Delete).)

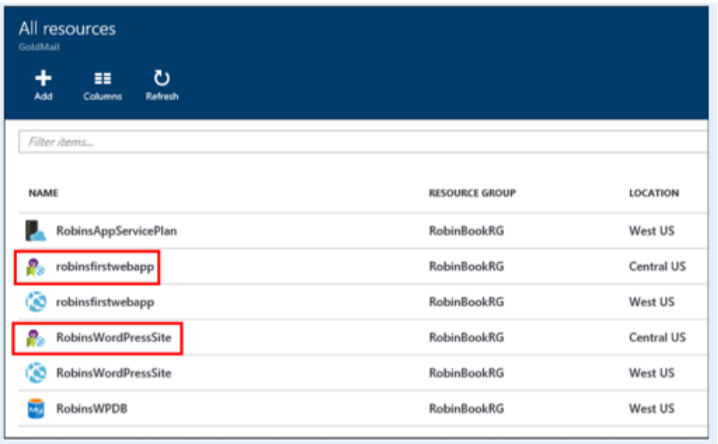

Рисунок 9.15 – При создании веб-приложения автоматически создаются экземпляры Application Insights

Пример: создание веб-сайта ASP.NET в Visual Studio и развертывание его в виде веб-приложения

Для воспроизведения этого примера вам потребуется установить Visual Studio 2017 или Visual Studio 2019, а также актуальную версию инструментов Azure и пакета SDK [5]. Чтобы создать новое веб-приложение в Visual Studio, выполните следующие действия.

1. Запустите Visual Studio. Выберите «Файл» > «Создать» > «Проект» (File > New > Project).

2. Выберите «Веб-приложение ASP.NET»; откроется диалоговое окно создания проекта, показанное на рис. 9.16. Снимите флажок «Добавить Application Insights к проекту» (Add Application Insights To Project) в правой части окна, чтобы не создавать для этого веб-приложения отдельный экземпляр Application Insights.

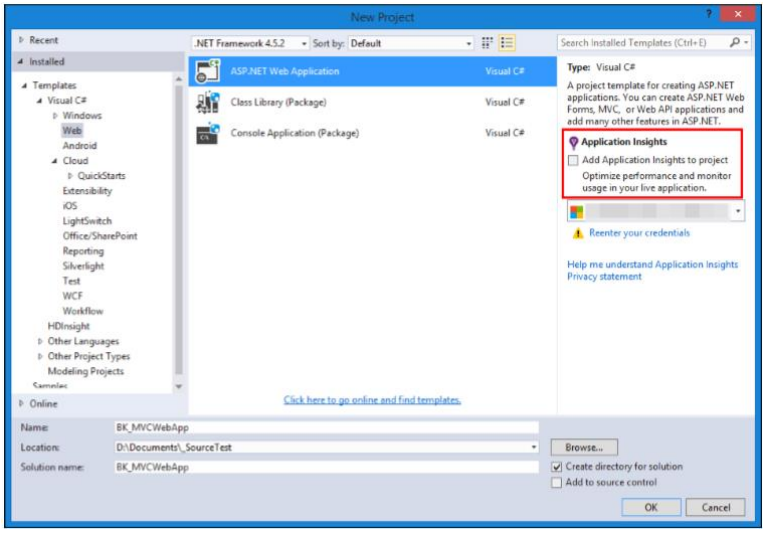

Рисунок 9.16 – Создайте проект «Веб-приложение ASP.NET» и снимите флажок Application Insights

3. Укажите имя и расположение приложения в соответствующих полях. Нажмите OK.
4. Когда откроется окно с запросом указать тип приложения ASP.NET, которое требуется создать, выберите MVC в списке «Шаблоны ASP.NET», как показано на рисунке 2-17. Снимите флажок «Разместить в облаке» (Host In The Cloud). Соответствующие параметры мы настроим отдельно. Нажмите OK, чтобы продолжить.

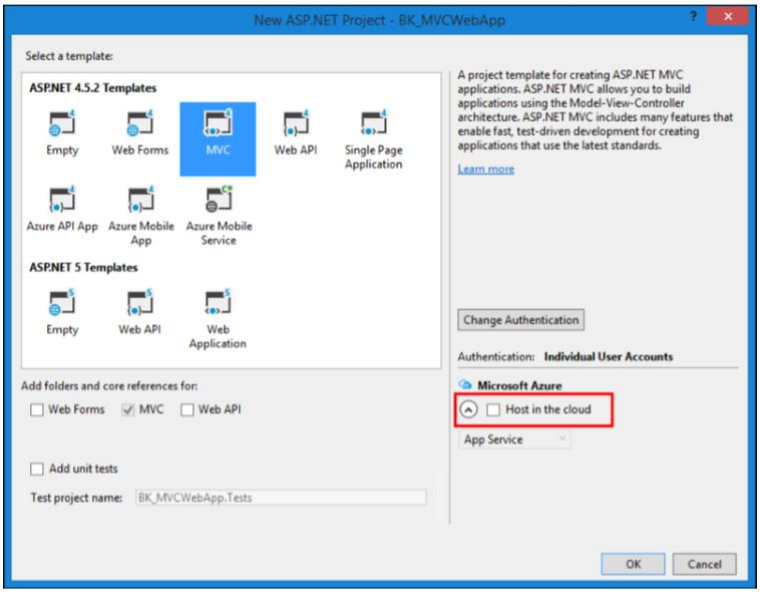

Рисунок 9.17 – Выберите шаблон «Приложение MVC» (MVC application) и снимите флажок «Разместить в облаке» (Host In The Cloud)

5. Visual Studio создаст элементарное приложение ASP.NET MVC, которое уже можно будет запустить. Вы можете добавить в него необходимые функции позже.
6. Теперь опубликуем это приложение в службе приложений Azure и сопоставим его с планом службы приложений, который мы создали ранее в ходе этой главы. Вы создадите службу приложений при первой публикации веб-приложения. Нажмите на веб-сайте правую кнопку мыши и выберите пункт «Опубликовать» (Publish) (рис. 9.18).

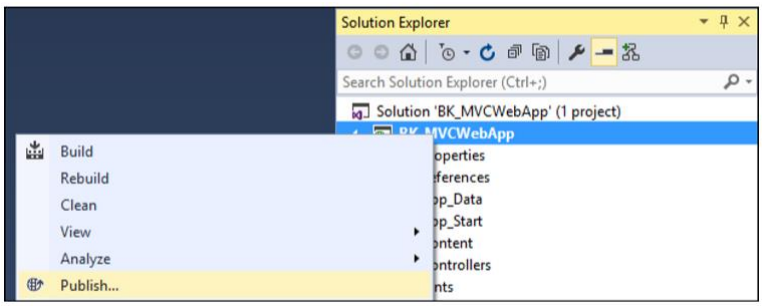

Рисунок 9.18 – Первый этап публикации веб-приложения

7. Откроется окно «Опубликовать веб-узел» (Publish Web). Выберите пункт «Служба приложений Microsoft Azure» (Microsoft Azure App Service) (рис. 9.19).

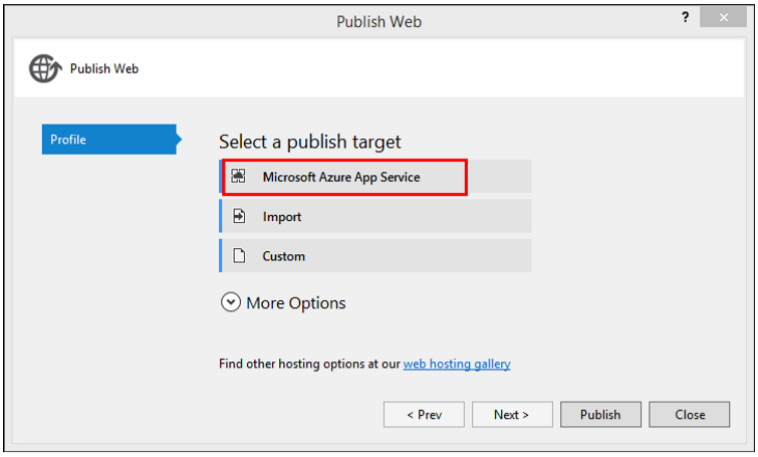

Рисунок 9.19 – В качестве цели для публикации выберите службу приложений Microsoft Azure

8. Необходимо указать имя подписки. После этого может потребоваться ввести учетные данные подписки Azure. Если отображается не та учетная запись, нажмите на нее, чтобы появился раскрывающийся список, и при необходимости добавьте учетную запись. Выберите учетную запись, затем подписку (Subscription). В поле «Представление» (View) выберите «Группа ресурсов» (Resource Group). Откройте группу ресурсов. В ней содержатся созданные ранее ресурсы. На рис. 9.20 показаны веб-приложения, которые были созданы у нас предварительно. Нажмите «Создать» (New), чтобы опубликовать это приложение в виде нового веб-приложения.

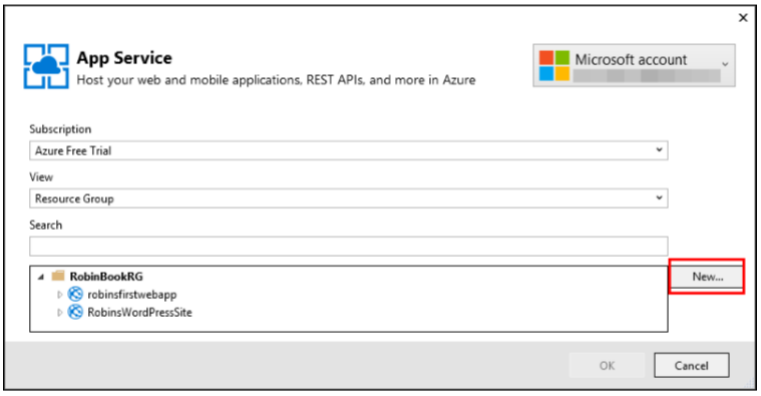

Рисунок 9.20 – Убедитесь, что выбрана требуемая учетная запись и подписка; выберите отображение ресурсов по группам

9. Далее откроется окно «Новая служба приложений» (Create App Service) (рис. 9.21). Помните, что служба приложений — это просто среда для веб-приложений, мобильных приложений, Logic App, приложений API и приложений-функций. Здесь вы создадите новую службу приложений, в которой будет размещаться веб-приложение MVC.

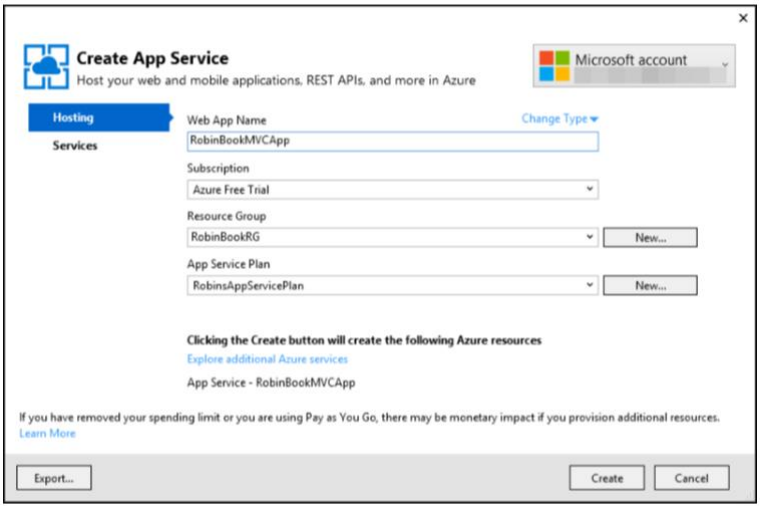

Рисунок 9.21 – Создайте службу приложений, в которой будет размещаться приложение MVC

- Заполните поле «Имя веб-приложения» (Web App Name). Это важный параметр: он будет использован для формирования URL-адреса веб-приложения.
- Выберите требуемую подписку (Subscription).
- Выберите группу ресурсов (Resource Group). Если вы используете группу, которую создали в начале этой главы, то после завершения работы сможете удалить эту группу ресурсов, чтобы удалить все ресурсы в ней.
- И наконец, выберите план службы приложений (App Service plan), который создали ранее в ходе этой главы. Это приложение будет размещаться в той же виртуальной машине, что и все прочие веб-приложения, которые вы поместили в этот план.

Нажмите «Создать» (Create), чтобы создать службу приложений.

Если сейчас вы откроете портал Azure, то увидите там созданную службу приложений.

10. Теперь выполним веб-развертывание, чтобы опубликовать веб-приложение в службе приложений. После создания службы

приложений откроется окно «Опубликовать веб-узел» (Publish Web) (рис. 9.22). Нам подойдут значения по умолчанию.

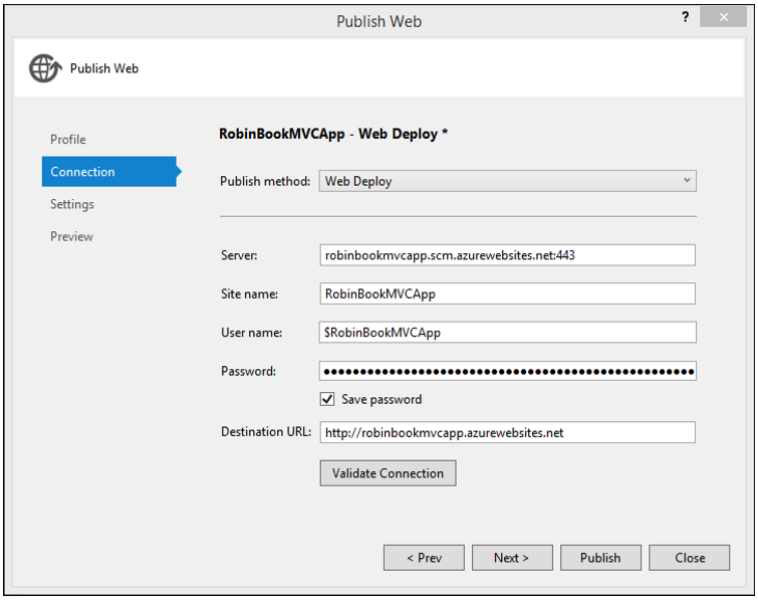

Рисунок 9.22 – Параметры публикации приложения MVC

11. Нажмите «Проверить подключение» (Validate Connection), чтобы убедиться в том, что введенная информация верна. После проверки нажмите «Далее» (Next). Откроется следующее окно (рис. 9.23).

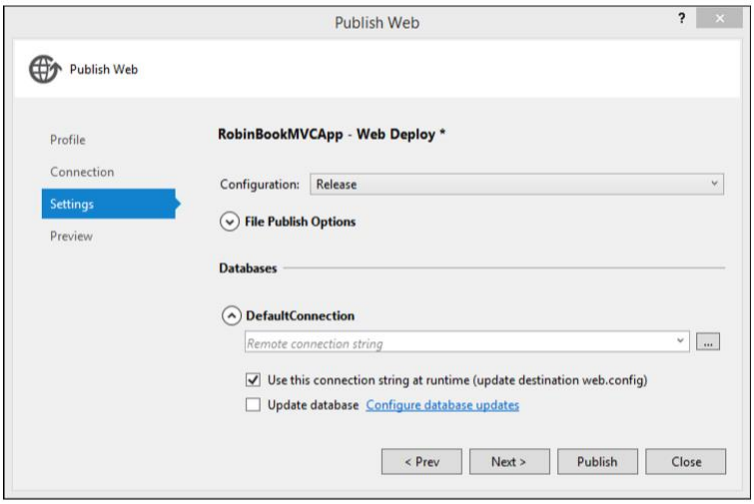

Рисунок 9.23 – Параметры, используемые при публикации приложения MVC

12. В этом окне можно выбрать конфигурацию (Debug или Release) и, при необходимости, указать строку подключения к базе данных. Обратите внимание: если потребуется отлаживать веб-приложение удаленно,

следует выбрать конфигурацию Debug. Нажмите «Далее» (Next). Откроется последняя страница (рис. 9.24).

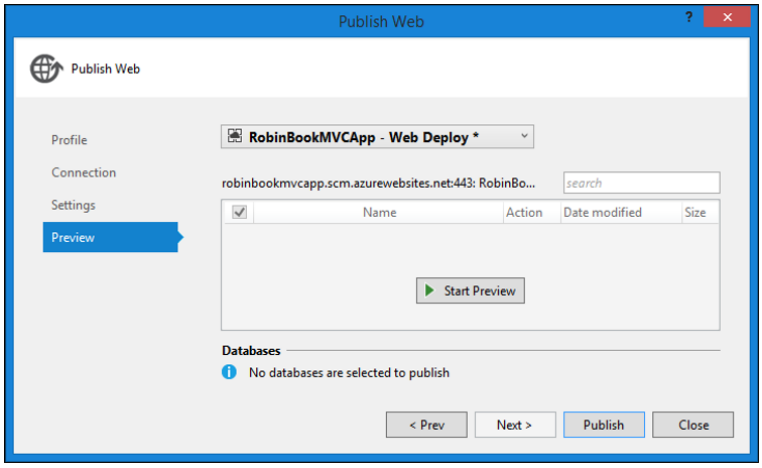

Рисунок 9.24 – Публикация приложения MVC

13.Здесь доступен предварительный просмотр созданного сайта. Теперь нажмите кнопку «Опубликовать» (Publish), чтобы развернуть веб-приложение в службе приложений.

После публикации веб-приложение откроется в браузере, установленном по умолчанию. После внесения изменений в веб-сайт для его повторной публикации можно выполнить те же действия. Обратите внимание: в этом случае будут опубликованы только добавленные и измененные файлы.

Ключевые термины:

Служба приложений —служба, предназначенная для размещения приложений пяти типов: веб-приложения; мобильные приложения; Logic Apps; приложения API; приложения-функции.

План службы приложений – план, который определяет набор и объем ресурсов, доступных для одной или нескольких служб приложений.

Подписка – учетная запись, которая позволяет администрировать несколько подписок.

Группа ресурсов – логические контейнеры для ресурсов, связанных между собой.

Location – регион Azure, в котором будет размещена группа ресурсов.

Веб-приложение Azure - веб-приложение, размещенное в службе приложений Azure.

Вопросы для самопроверки

1. Назначение службы приложений Azure.
2. Назначение план службы приложений Azure.
3. Поясните процесс создания плана службы приложений на портале Azure.
4. Назначение Группы ресурсов Azure.
5. Назначение веб-приложения Azure.
6. Какие языки программирования поддерживает Azure.
7. Поясните назначение Подписки Azure.
8. Как увеличить/уменьшить количество виртуальных машин для приложения?
9. Что определяет параметр Location для виртуальной машины?
10. Что определяют ценовые планы для службы приложений Azure.

Литература

1. Служба приложений. https://azure.microsoft.com/ru-ru/services/app-service/
2. Azure App Service plan overview. https://docs.microsoft.com/en-us/azure/app-service/overview-hosting-plans
3. Manage an App Service plan in Azure. https://docs.microsoft.com/en-us/azure/app-service/app-service-plan-manage
4. Deploy an Azure Web App. https://docs.microsoft.com/en-us/Azure/Devops/pipelines/targets/webapp?tabs=windows%2Cclassic&view=azure-devops-2019
5. Quickstart: Deploy an ASP.NET web app. https://docs.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore?tabs=net60&pivots=development-environment-vs