Microsoft

Введение в Windows Server 2016

Джон Мак-Кейб (John McCabe) и команда Windows Server

Издательство
Microsoft Press
Подразделение корпорации Майкрософт
One Microsoft Way
Редмонд, штат Вашингтон 98052-6399

ISBN: 978-0-7356-9774-4

Книги издательства Microsoft Press доступны в книжных магазинах и у торговых посредников во всем мире. Если вам нужна дополнительная поддержка в отношении этой книги, обратитесь в службу поддержки издательства Microsoft Press по адресу mspinput@microsoft.com. Поделитесь своим мнением об этой книге по адресу http://aka.ms/tellpress.

Книга предоставляется «как есть» и выражает точку зрения и мнение автора. В точки зрения, мнения и информацию, содержащиеся в этой книге, включая URL-адреса и другие ссылки на веб-сайты, могут быть внесены изменения без предварительного уведомления.

Некоторые рассмотренные примеры приведены только для справки и являются вымышленными. Любое сходство с реально существующими людьми или организациями является случайным.

Microsoft и товарные знаки, перечисленные по адресу http://www.microsoft.com на веб-странице «Товарные знаки», являются товарными знаками группы компаний Майкрософт. Все прочие товарные знаки являются собственностью соответствующих владельцев.

Рецензент издательства: Ким Спилкер (Kim Spilker)
Редактор по разработке: Боб Расселл (Bob Russell), Octal Publishing, Inc.
Издательская работа: Дайэнн Рассел (Dianne Russell), Octal Publishing, Inc.
Выпускающий редактор: Боб Расселл (Bob Russell)

Посетите наш веб-сайт по адресу

microsoftpressstore.com

- **Сотни наименований** — бумажные и электронные книги, сетевые ресурсы отраслевых экспертов
- **Бесплатная доставка в США**
- **Электронные книги в различных форматах** для чтения на компьютере, планшете, мобильных устройствах и устройствах для чтения электронных книг
- **Выгодные предложения бумажных и электронных книг**
- **Скидка недели на электронные книги** — скидка до 60 % на определенные наименования
- **Выпуски новостей и специальные предложения** — узнавайте первыми о новых выпусках, специальных предложениях и многом другом
- **Зарегистрируйте книгу** — получите дополнительные преимущества

Microsoft

Оглавление

Введение

Серверная операционная система Windows Server легла в основу ИТ-инфраструктуры целого поколения самых разных организаций, от небольших компаний до крупных корпораций. Вне зависимости от вашей роли в ИТ вы можете быть уверены в том, что работали с Windows Server в течение вашей карьеры или, по крайней мере, видели ее издалека! Эта книга познакомит вас с Windows Server 2016 — новой версией операционной системы Windows Server. После прочтения Вы узнаете обо всех главных и новых функциях Windows Server 2016.

Каждая глава написана экспертами или разработчиками, представившими самую актуальную информацию о каждом улучшении и новом функционале, включенных в эту версию Windows Server. Приведенная в книге информация, поможет вам подготовиться к переходу на Windows Server 2016 и предоставит возможности для планирования проекта внедрения Windows Server 2016 в условиях вашей ИТ-среды, чтобы вы смогли воспользоваться всеми преимуществами новой операционной системы. Все рекомендации, приведенные в этой книге, необходимо проверять, оценивать и испытывать в условиях тестовой среды, не следует сразу же внедрять их в рабочей среде.

Предполагается, что читатели знакомы с основными возможностями Windows Server (Microsoft Hyper-V, сети и системы хранения данных), а также с облачными технологиями, такими как Microsoft Azure. В этой книге рассматривается множество вопросов, связанных с технологиями, и описываются сценарии, ориентированные на заказчиков продукта. Вместе с тем данное руководство предоставляет только базовую информацию для ИТ-специалистов или ИТ-архитекторов. Для получение более детальных сведений о каждой главе, рекомендуется обращаться к официальным интернет-ресурсам Microsoft о Windows Server (часть из них представлена в этой книге). Используя их, вы можете получить последние рекомендации и лучшие практики для правильного внедрения или миграции на новую версию Windows Server.

Благодарность

Вклад в создание этой книги внесли многие авторы:

- Дэвид Холладей (David Holladay),
- Мич Туллок (Mitch Tulloch),
- Нед Пайл (Ned Pyle),
- Клаус Йоргенсен (Claus Joergensen),
- Мэтт Гэрсон (Matt Garson),
- Джон Марлин (John Marlin),
- Роберт Митчелл (Robert Mitchell),
- Дипак Шривастава (Deepak Srivastava),
- Рамниш Сингх (Ramnish Singh),
- Ритеш Моди (Ritesh Modi),

- Джейсон Андерсон (Jason M. Anderson),
- Шуманн Ге (Schumann Ge),
- Юрий Диогенес (Yuri Diogenes),
- Дэвид Брэнкам (David Branscome),
- Шаббир Ахмед (Shabbir Ahmed),
- Эндрю Мэйсон (Andrew Mason),
- Нил Питерсон (Neil Peterson),
- сотрудники издательства Microsoft Press.

Благодарим всех! А также спасибо за помощь и поддержку всем, кого мы не упомянули!

Над русским переводом книги работали:

- Роман Левченко: Microsoft MVP, MCSE, MCSA, MS, vExpert, VCP6-DCV https://rlevchenko.com

Microsoft Russia выражает благодарность Роману за глубокое вовлечение, энтузиазм и профессиональный подход в работе по переводу книги.

- Алексей Балтиков: MCT, MCSE, MCSA. https://www.facebook.com/alexey.baltikov
- Максим Фомин: MCT, MCSE Cloud, MCSE Mobility, MCSA Windows 10.
- Антон Фёдоров: MCT, MCSE Private Cloud, MCITP Enterprise Administrator, Database Administrator.
- Владислав Бородин: MCT, MCT Regional Lead Russia, MCSE, MCSA, MCPD. https://borntolearn.mslearn.net/members/vladboro
- Касачёв Константин: MCSE: Server Infrastructure. https://serverengineering.ru/

Бесплатные электронные книги издательства Microsoft Press

Бесплатные электронные книги издательства Microsoft Press охватывают различные предметные области — от технических обзоров до подробной информации по определенным темам. Эти книги доступны в форматах PDF, EPUB и Mobi (для электронных устройств Kindle). Их можно загрузить по адресу http://aka.ms/mspressfree.

Почаще заходите на этот веб-сайт, чтобы не пропустить новинки.

Ошибки, обновления и поддержка

Мы приложили все возможные усилия, для того чтобы обеспечить точность этой книги и вспомогательных материалов. Обновления книги (в виде списка обнаруженных ошибок и их исправлений) доступны по адресу https://aka.ms/IntroWinServ2016/errata.

Если обнаружите ошибку, которой нет в списке, отправьте, пожалуйста, ее описание на этой же странице.

Если вам потребуется помощь по другим вопросам, обратитесь в службу поддержки издательства Microsoft Press: mspinput@microsoft.com.

Обратите внимание на то, что по указанным выше адресам не предоставляется поддержка по программному обеспечению и оборудованию Microsoft. Чтобы получить помощь в отношении

программ и оборудования Microsoft, перейдите по адресу http://support.microsoft.com.

Нам важно ваше мнение

Для издательства Microsoft Press ваше мнение является главным приоритетом. Ваши отзывы — наш самый ценный ресурс. Поделитесь своим мнением об этой книге по адресу http://aka.ms/tellpress.

Ни один комментарий или предложение, которые вы оставите в ходе этого небольшого опроса, не останутся без нашего внимания. Заранее благодарим вас за участие!

Оставайтесь на связи

Мы всегда рады общению с вами! Наш Twitter: http://twitter.com/MicrosoftPress.

Введение в Microsoft Windows Server 2016

В любой организации, будь то небольшая компания, крупная корпорация или даже поставщик облачных услуг, требования к работе ИТ-подразделения быстро меняются. Заказчикам требуются доступ к различным приложениям и возможность выполнять повседневную работу безопасно и эффективно. Им не важно, как именно устроена ИТ-инфраструктура и с какими проблемами ежедневно приходится сталкиваться сотрудникам, которые ее поддерживают.

Введение

Как вы решаете перечисленные задачи, используя современную ИТ-среду? Справляются ли инфраструктура и приложения с предъявляемыми требованиями? Способны ли вы гибко реагировать на изменения и внедрять новые решения со скоростью, свойственной облачным технологиям? Количество сложных вопросов, касающихся локальных ИТ-инфраструктур, в этих и многих других областях постоянно растет.

Тем не менее не все готовы переходить на облачные решения, а во многих случаях такой переход может быть вовсе невозможен по ряду причин (например, если в условиях договора зафиксирован запрет на перемещение данных в облако).

Даже если вы не можете или не желаете перейти в облако сегодня, все равно важно приступить к модернизации инфраструктуры, чтобы воспользоваться всеми новыми наработками и возможностями Windows Server 2016, появившимися во многом благодаря накопленному Microsoft опыту применения облачных технологий.

Поддержка облачных технологий в Windows Server 2016

Windows Server 2016 — это, если в двух словах, готовая к использованию в облачной среде операционная система, обеспечивающая качественно новый уровень защищенности и новые возможности для повышения эффективности работы приложений и инфраструктуры, аналогичные тем, что имеются в Microsoft Azure.

При разработке этой версии корпорация Microsoft затратила довольно много времени на общение с заказчиками и сбор их мнений о том, какие возможности новой версии операционной системы оказались бы наиболее важными и как можно было бы удовлетворить требования к инфраструктуре, которые могут появиться в будущем. Все собранные мнения были разделены на три основных направления, которые представлены на рис. 1-1. Там также показаны проблемные области, о которых чаще всего упоминали клиенты. На основе их отзывов как раз и были разработаны новые возможности, реализованные в Windows Server 2016.

Безопасность	Программно-определяемый центр обработки данных	Платформа приложений
Рост количества атак Целью атак являются учетные данные Сложность защиты виртуальных сред	Недостаточная интеграция решений Сложность развертывания и эксплуатации Компактные серверы	Отсутствие интеграции между разработкой и эксплуатацией Быстрая и компактная операционная система Сложность планирования для общедоступного облака

Рис. 1-1. Категории отзывов, которые были использованы при разработке Windows Server 2016

Реагируя на отзывы заказчиков, разработчики Microsoft сосредоточили свои усилия на этих трех областях и выработали стратегию для каждой из них (рис. 1-2).

Безопасность	Программно-определяемый центр обработки данных	Платформа приложений
Блокирование атак, повышение безопасности виртуальных машин, приложений и данных за счет нескольких уровней безопасности, встроенных в операционную систему	Развитие центра обработки данных для достижения экономии и гибкости при помощи программно-определяемых технологий виртуализации вычислительных ресурсов, хранилищ и сетей, которые аналогичны возможностям Microsoft Azure	Ускорение внедрения новых решений на платформе приложений, оптимизированной как для традиционных, так и для облачных приложений

Рис. 1-2. Основные области и соответствующие стратегии развития возможностей Windows Server 2016

При проектировании новых возможностей мы опирались на опыт разработки и эксплуатации Azure: мы решили встроить аналогичные функции непосредственно в Windows Server 2016.

Все новые и усовершенствованные функции направлены на то, чтобы операционная система Windows Server 2016 стала предпочитаемой платформой с точки зрения безопасности, возможностей программно-определяемых центров обработки данных (эти возможности появились в Microsoft Azure, а теперь доступны и в локальной среде), а также в качестве платформы приложений, способной не только поддерживать работу традиционных приложений, но и предоставлять необходимую структуру для подготовки приложений к переносу в облако.

Следующие подразделы более подробно описывают направления, которые Microsoft обещает предоставить заказчикам, и, что самое главное, как их планируется реализовать.

Безопасность

Windows Server 2016 позволяет предотвращать атаки и выявлять подозрительные действия с помощью новых возможностей для управления привилегированным доступом, защиты виртуальных машин и повышения устойчивости платформы к новым угрозам. В Windows Server 2016 доступны следующие возможности:

- Предотвращение риска, связанного с утечкой учетных данных системных администраторов.

  Используя новые средства управления привилегированными удостоверениями, можно ограничить *область доступа* и *время доступа*. Кроме того, с помощью Credential Guard можно предотвратить кражу учетных данных администратора путем проведения атаки, основанной на получении хэша (*Pass-the-Hash*).

- Защита виртуальных машин от несанкционированных действий администраторов серверной структуры за счет использования экранированных виртуальных машин.

  Экранированная виртуальная машина — это виртуальная машина второго поколения, имеющая модуль Trusted Platform Module (TPM) и зашифрованная с помощью BitLocker. Такая виртуальная машина может быть запущена только на одобренных узлах серверной структуры.

- Снижение объема используемых ресурсов центра обработки данных и обеспечение высокой доступности за счет установки минимального набора необходимых компонентов операционной системы.

  Новый вариант развертывания, который называется Nano Server, занимает в 25 раз меньше дискового пространства, чем полноценный Windows Server с графической оболочкой. Использование Nano Server позволяет свести к минимуму возможности для атаки, повысить доступность и скорость запуска, а также снизить ресурсоемкость и время развертывания серверной операционной системы.

- Дополнительная защита всех развернутых экземпляров Windows Server 2016.

  При работе в облаке и локальной среде есть возможность использовать дополнительные средства безопасности, такие как проверка целостности кода (Core Integrity) и защита потока управления (Control Flow Guard). При этом возможен запуск только разрешенных двоичных файлов и обеспечивается защита от неизвестных уязвимостей.

- Обнаружение вредоносного поведения с помощью расширенного аудита безопасности, оптимизированного для выявления угроз.

  Новые категории аудита касаются членства в группах и PNP, они дают возможность выявлять события и получать о них более подробную информацию, благодаря чему администраторы могут полнее анализировать работу систем с целью обнаружения новых угроз.

- Защита от вредоносных программ с помощью встроенного антивирусного решения.

  В состав Windows Server 2016 входит Windows Defender, который оптимизирован для поддержки различных серверных ролей и интегрирован с Windows PowerShell с целью проверки наличия вредоносного ПО.

- Ограничение затрагиваемой области в случае вторжения.

  В случае нарушения системы безопасности система Windows Server 2016 может ограничить затрагиваемую область путем сегментации сети на основе рабочих нагрузок или потребностей бизнеса. При этом используются распределенный брандмауэр и группы сетевой безопасности. Можно применять сложные политики как внутри каждого сегмента, так и между сегментами.

- Использование контейнеров Hyper-V для получения уникального дополнительного уровня изоляции приложений в контейнерах без изменения содержимого самого контейнера.

  Контейнеры Hyper-V обеспечивают изоляцию на аппаратном уровне, при этом задействуются такие же методы изоляции, как при защите виртуализации на базе оборудования.

Программно-определяемый центр обработки данных

Выбрав Windows Server 2016, вы получаете гибкую и эффективную по затратам операционную систему для центра обработки данных, использующую программно-определяемые вычислительные ресурсы, системы хранения данных и функции виртуализации сети, аналогичные соответствующим возможностям Azure.

Программно-определяемые вычислительные решения

В следующем списке перечислены некоторые новые удивительные возможности, относящиеся к области программно-определяемых вычислительных решений в Windows Server 2016:

- Минимизация области атаки, повышение доступности и снижение использования ресурсов за счет установки только необходимых компонентов при развертывании Nano Server. Такой вариант развертывания занимает в 25 раз меньше дискового пространства, чем полная установка Windows Server
- Упрощенное перемещение в облако вашей рабочей нагрузки из Microsoft Hyper-V, того же самого гипервизора, который работает в Azure и Azure Stack.
- Развертывание приложений на множестве операционных систем, включая лучшую в своем классе поддержку операционной системы Linux на Hyper-V.
- Обновление кластеров инфраструктуры до Windows Server 2016 без простоев в работе приложений и рабочих нагрузок и без необходимости приобретать новое оборудование. Эти возможности поддерживаются благодаря поддержке работы кластеров в смешанном режиме (режим, при котором кластерные узлы могут выполняться как на Windows Server 2012 R2, так и на Windows Server 2016)
- Повышение доступности приложений за счет улучшенной устойчивости кластера к временным сбоям сетей и систем хранения данных.
- Дополнительная устойчивость кластеров при использовании Cloud Witness, использующий ресурсы Azure
- Автоматизация управления серверами с помощью встроенных средств, таких как Desired State Configuration и Windows PowerShell 5.0.
- Управление серверами с Windows из любого места с помощью средства управления серверами в виде службы Azure с веб-интерфейсом. Это особенно полезно для серверов Nano Server или Server Core.

Программно-определяемые системы хранения данных

В этом списке приведены некоторые возможности систем хранения данных корпоративного уровня, доступные в Windows Server 2016:

- Создание высокодоступных и масштабируемых программно-определяемых систем хранения данных при существенно меньших затратах, чем при использовании классических сетей

хранения данных (SAN) или сетевых хранилищ данных (NAS). С помощью Storage Spaces Direct можно создавать конвергентные и гиперконвергентные архитектуры хранения данных, используя локальные хранилища обычных серверов

- Синхронная и асинхронная репликация хранилища (Storage Replica) позволяет создавать доступные решения для обеспечения непрерывности бизнес-процессов и аварийного восстановления в центрах данных.
- Предоставление важнейшим бизнес-приложениям приоритетного доступа к ресурсам хранения данных путем использования возможностей по обеспечению требуемого уровня качества обслуживания систем хранения (QoS).

Программно-определяемые сети

Ниже перечислены некоторые новые возможности программно-определяемых сетей в Windows Server 2016:

- Развертывание сложных рабочих нагрузок с сотнями сетевых политик (изоляция, качество обслуживания, безопасность, балансировка нагрузки, коммутация, маршрутизация, шлюзы, DNS и т. д.) всего за несколько секунд с помощью масштабируемого сетевого контроллера — аналогично тому, как это делается в Azure.
- Динамическая сегментация сети на основе потребностей рабочих нагрузок с помощью распределенного брандмауэра и групп сетевой безопасности с возможностью применения комплексных политик как внутри отдельных сегментов, так и между сегментами. Маршрутизация или зеркалирование трафика в виртуальные устройства сторонних поставщиков с целью обеспечения дополнительного уровня безопасности.
- Более высокая доступность служб за счет устойчивого горизонтального и вертикального масштабирования программными средствами как инфраструктуры (узел, программная система балансировки нагрузки, шлюз, сетевой контроллер), так и рабочих нагрузок.
- Полный контроль над гибридными рабочими нагрузками, включая запуск их в контейнерах, перемещение между серверами, серверными стойками и облаками с использованием виртуальных сетей на основе VXLAN и NVGRE и мультитенантных гибридных шлюзов.
- Оптимизация затрат и производительности при объединении удаленного прямого доступа к памяти (RDMA) и клиентского трафика на одних и тех же объединенных сетевых адаптерах. Такой подход позволяет снизить расходы и при этом добиться требуемой гарантированной пропускной способности сети 40 Гбит/с и более.

Платформа приложений

В Windows Server 2016 поддерживаются новые способы развертывания и запуска приложений как в локальной среде, так и в Azure. Среди новых возможностей — контейнеры Windows и вариант сверхкомпактного развертывания Nano Server.

- Контейнеры в Windows Server 2016 обладают высокой гибкостью и плотностью на уровне, который требуется современным облачным приложениям. Поддержка контейнеров в Windows Server позволяет полноценно использовать контейнеры в экосистеме Windows и в некоторых случаях задействовать для работы важных приложений контейнеры Hyper-V с дополнительным уровнем изоляции, причем без написания дополнительного программного кода.
- Сверхкомпактный вариант развертывания Nano Server обеспечивает высокую гибкость, которая требуется разработчикам современных приложений. Это идеальный вариант для запуска приложений в контейнерах или с помощью микрослужб.
- При выполнении традиционных мощных приложений, таких как SQL Server 2016, обеспечиваются лучшие в своем классе показатели производительности, безопасности и доступности.

- Лицензии Azure Hybrid Use Benefit (HUB) позволяют экономить деньги при переносе имеющихся лицензионных продуктов на Windows Server в Azure: в этом случае оплачивается лишь базовый вычислительный тариф (требуется подписка SA).
- Переход к сервисной модели. Используя Nano Server, можно чаще обновлять операционную систему, получая больше новых возможностей в течение ее жизненного цикла, и предоставлять разработчикам инструменты для планомерного внедрения самых современных методик Agile и/или технологий безопасности, внедряемых корпорацией Microsoft.

В этой книге мы опишем все перечисленные элементы и предоставим ссылки на подробную информацию для всех упомянутых категорий, функций, возможностей и компонентов.

Microsoft любит Linux!

Не секрет, что корпорация Microsoft приложила немало усилий, для того чтобы платформа Linux получила в экосистеме Microsoft возможности корпоративного уровня. В частности, Microsoft внесла свой вклад в разработку ядра Linux и активно поддерживает службы интеграции с Linux (Linux Integration Services, LIS), чтобы обеспечить максимум возможностей при использовании Linux в Hyper-V.

В настоящее время в Hyper-V обеспечивается полная поддержка следующих дистрибутивов (в дальнейшем этот список будет расширен):

- Red Hat Linux,
- SUSE openSUSE,
- CentOS,
- Ubuntu,
- Debian,
- Oracle Linux.

В табл. 1-1 перечислены лишь некоторые новшества LIS

Таблица 1-1. Основные нововведения LIS

Область	Описание
Сети	Полная поддержка виртуального масштабирования размера приема (vRSS) для оптимизации производительности сетевых компонентов Linux. «Горячее» добавление и удаление виртуальных сетевых адаптеров.
Системы хранения данных	Поддержка «горячего» добавления дисков и изменения размера системы хранения.
Память	Поддержка изменения размера памяти без необходимости выключения машины
Управление	Упрощенное управление с помощью широко распространенных инструментов, таких как PowerShell DSC.
Производительность	Производительность Linux в Hyper-V сравнима с производительностью в других низкоуровневых оболочках.

System Center 2016

Как уже было сказано, операционная система Windows Server 2016 полностью готова к использованию облачных технологий и содержит множество новых возможностей, аналогичных возможностям Azure. В частности, можно использовать Windows Server 2016 в качестве платформы для программно-определяемого центра обработки данных. Впрочем, возникает

необходимость управлять и облаками, как общедоступными, так и частными. Для этого предназначен продукт System Center 2016 — решение для управления центрами обработки данных. Microsoft инвестировала немало средств для его улучшения.

Обновленный System Center 2016 призван раскрыть все ключевые возможности Windows Server 2016, позволяющие создать и поддерживать полнофункциональный программно-определяемый центр обработки данных на базе Windows Server 2016.

Ниже перечислены лишь некоторые из новых возможностей System Center 2016:

- Управление устройствами.

  К этой группе функций относятся поддержка развертывания Windows 10, регистрация решения для управления мобильными устройствами (MDM) в Azure Active Directory и ограничение доступа на основе регистрации устройств и политики.

- Выделение ресурсов.

  Усовершенствования в этой области включают поддержку функций Hyper-V в Windows Server 2016, последовательный апгрейд кластеров (cluster rolling upgrade), упрощенное управление сетями, поддержку экранированных виртуальных машин, управление защищенными узлами и поддержку vCenter

- Мониторинг.

  Среди новшеств в области мониторинга — добавление поддержки Nano Server, хранилища на базе Windows, а также появившиеся в этой версии продукта поставщик SMI-S и каталог пакетов управления. Кроме того, обеспечиваются рост производительности, расширенная визуализация данных и поддержка партнерской программы SCOM.

- Автоматизация.

  Усовершенствования в этой области включают упрощенный перенос приложений в облако, пакеты интеграции SCO и модулей Runbook.

- Самообслуживание.

  Функции самообслуживания стали более удобными и производительными, а кроме того, появились портал самообслуживания на основе HTML5 и новый соединитель с Exchange.

- Защита данных.

  Усовершенствования в этой области включают поддержку Azure ExpressRoute, экранированных виртуальных машин и Storage Spaces Direct.

Все эти новые возможности пакета System Center станут мощным подспорьем для организаций, испытывающих потребность в облачных решениях нового поколения. Кроме того, System Center 2016 теперь обладает встроенными средствами интеграции с Microsoft Operations Management Suite.

Эта интеграция открывает перед вами новые возможности, дополняющие существующую широкую функциональность System Center. Администраторы при этом получают более полную информацию об ИТ-среде, а также средства защиты, управления и безопасности на уровне облаков. Применяя возможности Operations Management Suite для подготовки отчетности и встроенные механизмы интеграции с Microsoft Power BI, администраторы могут быстро и просто создавать эффективные и динамические отчеты с наглядным представлением данных.

На рис. 1-3 показан пример панели мониторинга, получающей информацию из пакетов сбора и анализа данных, которые по умолчанию входят в состав подписки Operations Management Suite. Здесь мы видим пример того, как можно работать с комплексной визуальной информацией, используя эти пакеты сбора и анализа данных, развернутые в ИТ-среде и соединенные с источниками данных.

Рис. 1-3. Панель мониторинга Operations Management Suite

Нажав на одну из плиток, можно получить более подробную информацию об интересующей области. По умолчанию каждый пакет сбора и анализа данных содержит собственный набор правил, при этом администраторы могут легко создавать наборы правил, исходя из своих потребностей, а также использовать средства визуализации полученных результатов и гибкие возможности работы с ними.

Пакет Operations Management Suite может дополнять уже развернутый System Center либо работать как автономная платформа, управляя системами, развернутыми в любых облачных и локальных средах.

Платформа Operations Management Suite разделяется на следующие функциональные области:

- **Наблюдения и аналитика.** Эта группа функций нацелена на получение данных из множества источников, установку взаимосвязей между процессами и предоставление механизмов, дающих возможность работать с полученными данными с использованием оповещений, а также производить поиск действий, которые нужно выполнить. Также поддерживается возможность сопоставления и анализа зависимостей рабочих нагрузок, использующих общие ресурсы.
- **Безопасность и соответствие требованиям.** Эта группа функций, основаннанных на технологиях Microsoft для сбора и анализа данных, имеющих отношение к информационной безопасности, поможет предотвращать и обнаруживать угрозы, а также реагировать на них гораздо эффективнее, чем прежде. Располагая более полными и точными данными о том, что происходит в среде, можно предотвращать опасные ситуации и применять политики безопасности, чтобы обеспечивать полный контроль над всей ИТ-экосистемой, в том числе над облаками.
- **Автоматизация и управление.** Здесь собраны функции, предоставляющие администраторам все необходимое для управления ИТ-средой. Предусмотрена возможность запуска Runbook на основе оповещений, полученных от средств сбора и анализа информации. Кроме того, использование различных средств автоматизации значительно повышает производительность работы.
- **Защита и восстановление.** В состав платформы включены простые и эффективные механизмы облачного резервного копирования и аварийного восстановления, отвечающие требованиям современных организаций. Они позволяют эффективно автоматизировать процессы аварийного восстановления, гарантируя успешный результат.

Перечень этих функциональных направлений важен с точки зрения более точного понимания областей использования Operations Management Suite и возможных вариантов его внедрения. Тем не менее здесь перечислены далеко не все возможности и не все пакеты решений, которые доступны уже сейчас или появятся в ближайшем будущем. На рис. 1-4 показаны существующие и перспективные пакеты решений, при помощи которых заказчики могут получить более полную и точную информацию о своей ИТ-среде для текущего и будущего состояний.

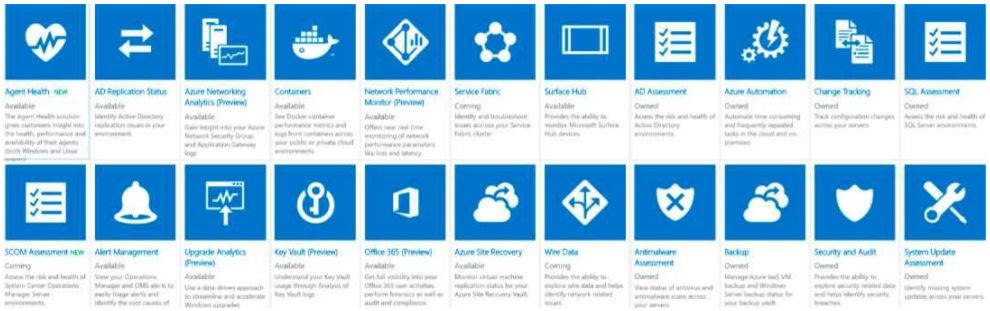


Рис. 1-4. Решения, доступные в Operations Management Suite в настоящее время, а также планируемые решения

Далее в этой книге мы подробнее поговорим о пакете Operations Management Suite и продемонстрируем на простых примерах, как это решение дополняет возможности Windows Server 2016.

ГЛАВА 2

Программно-определяемый центр обработки данных

В этой главе мы поговорим о новых и усовершенствованных возможностях Windows Server 2016 в области программно-определяемых центров обработки данных. Если ваша организация является поставщиком облачных услуг или стремится создать платформу для размещения приложений нового поколения, вы непременно оцените преимущества Windows Server 2016. Эта глава разделена на три части: вычисления, системы хранения данных и сети. Это три основных компонента любого программно-определяемого центра данных. Далее мы подробно рассмотрим каждую из этих областей.

Вычисления

Этот раздел посвящен серверным вычислениям. Значительное внимание в нем уделяется Hyper-V и новым возможностям Windows Server 2016. Мы разберем все функции, присущие программно-определяемым центрам обработки данных мирового уровня.

Hyper-V

Авторы: Роберт Митчелл (Robert Mitchell), Дипак Шривастава (Deepak Srivastava), Шаббир Ахмед (Shabbir Ahmed), Рамниш Сингх (Ramnish Singh)

В Windows Server 2016 технология виртуализации Microsoft Hyper-V получила ряд усовершенствований. В этом разделе описаны некоторые из них. Роберт Митчелл продемонстрирует новую функцию, которая называется «группы виртуальных машин», и опишет новые возможности перемещения виртуальных машин между серверами с разными версиями Windows Server. Дипак Шривастава расскажет о новой версии конфигурации виртуальной машины, новом формате файлов конфигурации и о поддержке использования контрольных точек в рабочей среде. Шаббир Ахмед и Рамниш Сингх опишут новую возможность «горячего» добавления и удаления оперативной памяти и сетевых адаптеров, которую теперь поддерживает роль Hyper-V.

Масштабирование

В Windows Server 2016 доступны новые, исключительно широкие возможности масштабирования для виртуализации любых без исключения рабочих нагрузок. В табл. 2-1 показано сравнение возможностей Windows Server 2012 и Windows Server 2012 R2 с новой версией.

Таблица 2-1. Сравнение возможностей Windows Server 2012 и 2012 R2 с версией Windows Server 2016

Описание	Windows Server 2012 и 2012 R2 (Standard и Datacenter)	Windows Server 2016 (Standard и Datacenter)
Объем памяти на физических серверах	До 4 ТБ на один физический сервер	До 24 ТБ на один физический сервер (в 6 раз больше)
Количество логических процессоров на физических серверах	До 320 логических процессоров	До 512 логических процессоров
Объем памяти в виртуальной машине	До 1 ТБ	До 16 ТБ (в 16 раз больше)
Количество виртуальных процессоров в виртуальной машине	До 64	До 240 (в 3,75 раза больше)

Вложенная виртуализация

Вложенная виртуализация позволяет использовать Hyper-V как гостевую виртуальную машину, работающую в среде Hyper-V! При этом виртуальная машина получает доступ к аппаратным расширениям виртуализации. Для использования этой возможности система должна отвечать определенным требованиям:

- операционная система Windows Server 2016 или Windows 10;
- не менее 4 ГБ оперативной памяти на физическом сервере;
- процессоры с поддержкой технологии Intel VT-x (на момент написания этой книги);
- поддержка EPT;
- динамическая память должна быть отключена для ВМ с вложенной виртуализацией.

Чтобы включить вложенную виртуализацию, необходимо сначала выполнить на сервере следующую команду Windows PowerShell для виртуальной машины, которую вы создали, но еще не включили:

```
Set-VMProcessor -VMName <имя_виртуальной_машины> -ExposeVirtualizationExtensions $true
```

Чтобы предоставить возможности подключения для гостевых виртуальных машин, которые будут размещены на вложенной виртуальной машине с Hyper-V, можно использовать два варианта. Первый — включить подмену MAC-адресов для гостевых виртуальных машин, после этого они смогут передавать трафик по сети. Чтобы включить подмену MAC-адреса на коммутаторе Hyper-V, используйте следующую команду:

```
Get-VMNetworkAdapter -VMName <имя_виртуальной_машины> | Set-VMNetworkAdapter -MacAddressSpoofing On
```

Второй вариант — использовать преобразование сетевых адресов (NAT). Включить NAT на вложенной виртуальной машине с Hyper-V можно с помощью следующих команд:

```
new-vmswitch -name VmNAT -SwitchType Internal

New-NetNat -Name LocalNAT -InternalIPInterfaceAddressPrefix "192.168.100.0/24"
```

После этого нужно назначить IP-адрес новому внутреннему адаптеру. Этот IP-адрес будет адресом шлюза для виртуальных машин, работающих на вложенной машине с Hyper-V. Вот соответствующая команда Windows PowerShell:

```
get-netadapter "vEthernet (VmNat)" | New-NetIPAddress -IPAddress 192.168.100.1 -AddressFamily IPv4
-PrefixLength 24
```

Для каждой вложенной гостевой виртуальной машины необходимо задать IP-адрес и шлюз следующим образом:

```
get-netadapter "Ethernet" | New-NetIPAddress -IPAddress 192.168.100.2 -DefaultGateway
192.168.100.1 -AddressFamily IPv4 -PrefixLength 24
```

> **Примечание.** Более подробная информация доступна по адресу https://msdn.microsoft.com/virtualization/hyperv_on_windows/user_guide/nesting.

Безопасная загрузка Linux

Виртуальные машины с Linux, созданные в виде виртуальных машин второго поколения, могут использовать безопасную загрузку. Для этого необходимо включить на виртуальной машине использование центра сертификации Microsoft UEFI, используя следующую команду Windows PowerShell:

```
Set-VMFirmware <имя_виртуальной_машины> -SecureBootTemplate MicrosoftUEFICertificateAuthority
```

Также можно включить безопасную загрузку с помощью диспетчера Hyper-V или диспетчера виртуальных машин. В настоящее время безопасная загрузка поддерживается только для следующих дистрибутивов:

- Ubuntu 14.04 и более поздние версии;
- SUSE Linux Enterprise Server 12 и более поздние версии;
- Red Hat Enterprise Linux 7.0 и более поздние версии;
- CentOS 7.0 и более поздние версии.

Службы интеграции

Обновления служб интеграции гостевых машин Windows распространяются посредством центра обновления Windows. Для поставщиков услуг и компаний по размещению частных облаков это означает, что контроль за применением обновлений переходит к потребителям выделенны услуг, которым принадлежат виртуальные машины. Теперь аони могут устанавливать на свои виртуальные машины Windows все обновления, включая службы интеграции, используя при этом один и тот же метод их доставки.

Усовершенствования диспетчера Hyper-V

Диспетчер Hyper-V получил ряд улучшенных возможностей:

- **Поддержка альтернативных учетных данных.** Теперь в диспетчере Hyper-V можно определить другой набор учетных данных при подключении к серверу Windows Server 2016 или к удаленному узлу Windows 10. Эти учетные данные можно сохранить для удобства повторного входа.
- **Управление более ранними версиями.** С помощью диспетчера Hyper-V в Windows Server 2016 и Windows 10 можно управлять компьютерами с Hyper-V и операционными системами Windows Server 2012, Windows 8, Windows Server 2012 R2 и Windows 8.1.
- **Обновленный протокол управления.** В диспетчер Hyper-V были внесены изменения, и теперь для связи с удаленными узлами Hyper-V используется протокол WS-MAN, который поддерживает проверку подлинности CredSSP, Kerberos и NTLM. При использовании CredSSP для подключения к удаленному узлу Hyper-V виртуальные машины можно динамически перемещать, не включая при этом ограниченное делегирование в Active Directory. В инфраструктуре на основе протокола WS-MAN также упрощается настройка узлов для удаленного управления. WS-MAN использует для подключения порт 80, который по умолчанию открыт.

Защита ресурсов узла

Одна из распространенных проблем виртуализации заключается в том, чтобы не позволить виртуальным машинам использовать больше ресурсов, чем им выделено. Чрезмерное

потребление ресурсов может привести к снижению производительности системы и гостевых виртуальных машин. По умолчанию мониторинг потребления ресурсов и защита ресурсов отключены. Чтобы их включить, выполните команду:

```
Set-VMProcessor -EnableHostResourceProtection $true
```

При этом будет запущен процесс, отслеживающий чрезмерное использование ресурсов и ограничивающий их для каждой виртуальной машины, чтобы избежать недопустимо высокой нагрузки.

Режим ожидания с подключением

Установив роль Hyper-V на компьютер, использующий режим электропитания «всегда включен, всегда подключен» (AOAC), можно использовать режим ожидания без прекращения сетевых подключений (connected standby)

Назначение устройств

С помощью этой функции можно предоставить виртуальной машине прямой монопольный доступ к некоторым устройствам с интерфейсом PCIe. При таком способе использования устройства обмен данными происходит в обход стека виртуализации Hyper-V, за счет чего обеспечивается ускоренный доступ.

> **Примечание.** Более подробная информация доступна по адресу http://blogs.technet.com/b/virtualization/archive/2015/11/19/discrete-device-assignment.aspx.

Windows PowerShell Direct

Windows PowerShell Direct дает возможность выполнять команды Windows PowerShell на виртуальной машине, запуская их на физическом сервере. Windows PowerShell Direct работает между хостом и виртуальной машиной. При этом не требуется сетевого подключения и настройки брандмауэра, этот функционал работает вне зависимости от настройки функций удаленного управления.

Windows PowerShell Direct работает аналогично Windows PowerShell, но без сетевого подключения.

Для подключения к виртуальной машине с узла используйте командлет Enter-PSSession:

```
Enter-PSSession -VMName <имя_виртуальной_машины>
```

Потребуется ввести учетные данные, после чего можно будет управлять этой виртуальной машиной из сеанса PSSession.

Командлет Invoke-Command обновлен и работает аналогичным образом. Например, можно запускать на узле сценарий для виртуальной машины:

```
Invoke-Command -VMName <имя_виртуальной_машины> -FilePath C:\Scripts\MyTestScript.psl
```

Удаленный доступ к памяти (RDMA)

В Windows Server 2016 можно включить удаленный доступ к памяти для сетевых адаптеров, не использующих объединение или без встроенного объединения на уровне коммутатора (SET). Более подробная информация приведена ниже в этой главе.

> **Примечание.** Более подробная информация об использовании RDMA доступна по адресу https://technet.microsoft.com/library/mt403349.aspx.

Группы виртуальных машин

Чтобы упростить управление несколькими виртуальными машинами, в Windows Server 2016 добавлена возможность объединения виртуальных машин в логические группы. Группы виртуальных машин представляют собой именно то, что следует из их названия, — логическую группировку виртуальных машин.

Имеются два типа групп:

- коллекции виртуальных машин,
- коллекции управления.

Коллекция виртуальных машин — это логический набор виртуальных машин. С помощью таких групп администраторы могут применять действия сразу ко всей группе, а не к каждой виртуальной машине по отдельности.

Коллекция управления — это логическая коллекция, состоящая из групп виртуальных машин. В группе этого типа администраторы могут размещать вложенные группы виртуальных машин.

В диспетчере Hyper-V можно выполнять действия с несколькими виртуальными машинами одновременно, выбирая несколько объектов, как показано на рис. 2-1.

Рис. 2-1. Доступные действия с виртуальными машинами

Можно выполнять эти действия и без использования групп виртуальных машин, но в этом случае список доступных действий ограничен. Группы виртуальных машин предоставляют значительно более широкие возможности. Особенно полезно использовать группы виртуальных машин в двух случаях: для резервного копирования и для репликации виртуальных машин. Разумеется, создание резервной копии или репликация виртуальной машины не представляют сложности, и соответствующие функции уже давно поддерживаются в Windows Server, но при этом необходимые действия приходится выполнять по отдельности для каждой виртуальной машины. В некоторых случаях в силу распределенной архитектуры приложений следует работать с несколькими виртуальными машинами как с единым целым. Это касается как резервного копирования, так и репликации.

Создание групп виртуальных машин

Чтобы облегчить создание сценариев, были введены следующие новые командлеты Windows PowerShell:

- New-VMGroup,
- Get-VMGroup,
- Remove-VMGroup,
- Add-VMGroupMember,
- Remove-VMGroupMember,
- Rename-VMGroup.

Чтобы объединить в группу три виртуальные машины (рис. 2-2), выполните следующие действия:

1. Создайте группу виртуальных машин.
2. Добавьте виртуальные машины в эту группу в качестве членов.

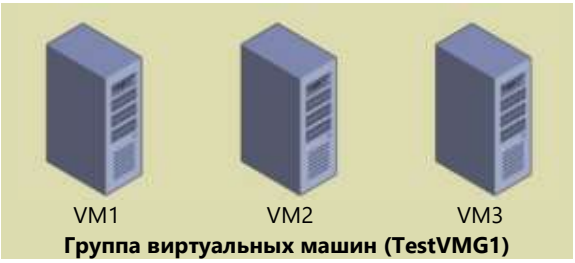


Рис. 2-2. Группы виртуальных машин

Ниже приведен код сценария Windows PowerShell для выполнения этой задачи. Помните, что мы создаем группу (или коллекцию) виртуальных машин. Помещать виртуальные машины напрямую можно только в группы виртуальных машин.

```
#Настройка переменных виртуальных машин
$VM1 = Get-VM -Name VM1
$VM2 = Get-VM -Name VM2
$VM3 = Get-VM -Name VM3

#Создание новой группы виртуальных машин
New-VMGroup -Name TestVMG1 -GroupType VMCollectionType

#Настройка переменной группы виртуальных машин
$TestVMG1 = Get-VMGroup -Name TestVMG1

#Добавление виртуальных машин в группу
Add-VMGroupMember -VMGroup $TestVMG1 -VM $VM1
Add-VMGroupMember -VMGroup $TestVMG1 -VM $VM2
Add-VMGroupMember -VMGroup $TestVMG1 -VM $VM3
```

В результате получится группа виртуальных машин, содержащая три виртуальные машины.

Проверить это можно с помощью средств управления, опросив как любую из виртуальных машин, так и группы виртуальных машин. В следующем примере показано, как это сделать с помощью командлетов Get-VM и Get-VMGroup.

```
PS C:\> Get-VM | ft Name, state, groups - AutoSize
Name         State       Groups
VM1          Running     {TestVMG1}
VM2          Running     {TestVMG1}
VM3          Running     {TestVMG1}

PS C:\> Get-VMGroup * | ft Name, vmmembers -AutoSize
Name         VMMembers
TestVMG1     {VM2, VM3, VM1}
```

Обновленный командлет Get-VM перечисляет группы (если они есть), членом которых является данная виртуальная машина. Одна и та же виртуальная машина может быть в составе нескольких групп. В этом случае командлет Get-VM возвратит список из нескольких групп.

Новый командлет Get-VMGroup перечисляет все виртуальные машины, входящие в состав указанной группы, либо, как в приведенном выше примере, где мы использовали подстановочный знак, в состав всех существующих групп. В этом примере мы опрашиваем все группы, поскольку нам заведомо известно, что существует только одна группа. Тем не менее можно добавить одну из виртуальных машин во вторую группу. Вот краткий сценарий Windows PowerShell для этого:

```
#Создание новой группы виртуальных машин
New-VMGroup -Name TestVMG2 -GroupType VMCollectionType

#Настройка переменной группы виртуальных машин
$TestVMG2 = Get-VMGroup -Name TestVMG2

#Добавление виртуальных машин в группу
Add-VMGroupMember -VMGroup $TestVMG2 -VM $VM1
```

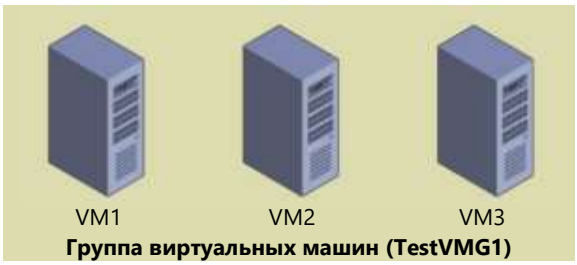
Теперь с помощью командлета Get-VM можно убедиться в том, что виртуальная машина VM1 входит в состав группы TestVMG1 и новой группы TestVMG2:

```
PS C:\> Get-VM | ft Name, state, groups - AutoSize

Name        State       Groups
VM1         Running     {TestVMG2, TestVMG1}
VM2         Running     {TestVMG1}
VM3         Running     {TestVMG1}
```

С помощью командлета Get-VMGroup перечислены обе группы. Кроме того, видно, что виртуальная машина VM1 входит в состав обеих групп:

```
PS C:\> Get-VMGroup * | ft Name, vmmembers -AutoSize

Name        VMMembers
TestVMG2    {VM1}
TestVMG1    {VM2, VM3, VM1}
```

Теперь у нас две группы виртуальных машин: одна с тремя виртуальными машинами, а другая с одной, как показано на рис. 2-3.



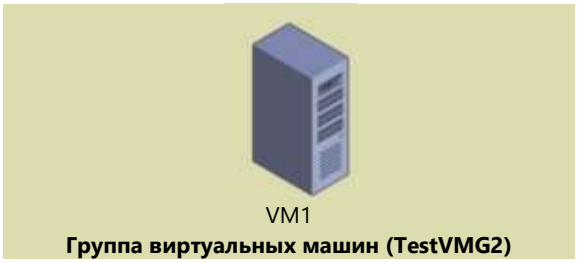


Рис. 2-3. Несколько групп виртуальных машин

При наличии двух групп виртуальных машин можно выполнять действия, которые необходимо осуществить с виртуальными машинами VM1, VM2 и VM3, используя группу TestVMG1. При этом действия, затрагивающие только VM1, можно выполнять с помощью группы TestVMG2.

Создание коллекций управления

Коллекции виртуальных машин устроены очень просто: каждая из них состоит из виртуальных машин. В отличие от них коллекции управления состоят из коллекций виртуальных машин. На рис. 2-4 показана группа управления, содержащая обе ранее созданные группы виртуальных машин. Эти группы виртуальных машин, в свою очередь, содержат фактические виртуальные машины. Обратите внимание на то, что виртуальные машины не могут входить непосредственно в состав коллекции управления.

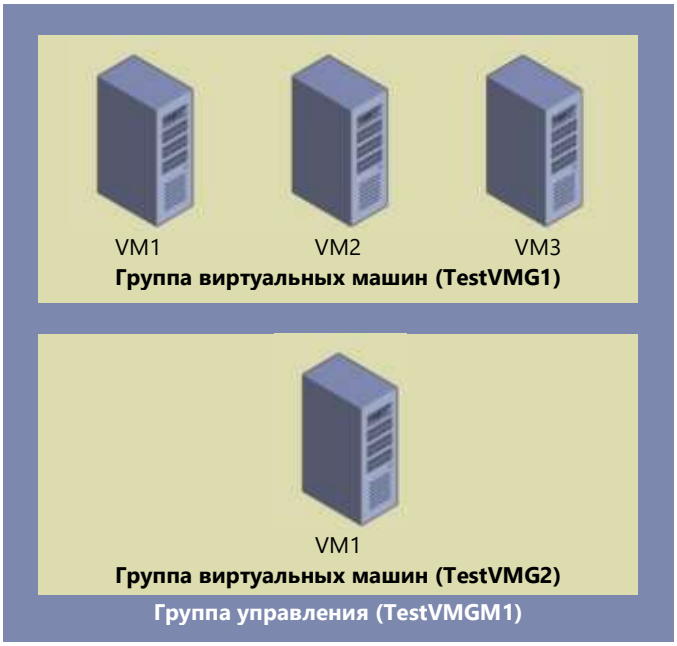


Рис. 2-4. Одна группа управления, содержащая несколько групп виртуальных машин

Создание групп управления почти идентично созданию групп виртуальных машин с помощью описанных ранее средств управления. Следующий сценарий Windows PowerShell создает новую группу управления и добавляет в нее обе существующие группы виртуальных машин:

```
#Создание новой группы управления
New-VMGroup -Name TestVMGM1 -GroupType ManagementCollectionType

#Настройка переменной группы управления
$TestVMGM1 = Get-VMGroup -Name TestVMGM1

#Добавление групп виртуальных машин в группу управления
Add-VMGroupMember -VMGroup $TestVMGM1 -VMGroupMember $TestVMG1
Add-VMGroupMember -VMGroup $TestVMGM1 -VMGroupMember $TestVMG2
```

Интересное различие между группами виртуальных машин и группами управления состоит в том, что в составе группы управления могут быть как группы виртуальных машин, так и другие группы управления. Проще говоря, могут существовать вложенные группы управления.

Следующий сценарий Windows PowerShell создает новую группу управления с именем Outside и добавляет в нее созданную ранее группу управления TestVMGM1:

```
#Создание новой группы управления
New-VMGroup -Name OutsideGroup -GroupType ManagementCollectionType

#Настройка переменной группы управления
$OutsideGroup = Get-VMGroup -Name OutsideGroup

#Добавление групп виртуальных машин в группу управления
Add-VMGroupMember -VMGroup $OutsideGroup -VMGroupMember $TestVMGM1
```

Группа управления (OutsideGroup) содержит другую группу управления (TestVMGM1), которая содержит две группы виртуальных машин (TestVMG1 и TestVMG2), которые, в свою очередь, содержат разные наборы трех виртуальных машин (VM1, VM2 и VM3), как показано на рис. 2-5.

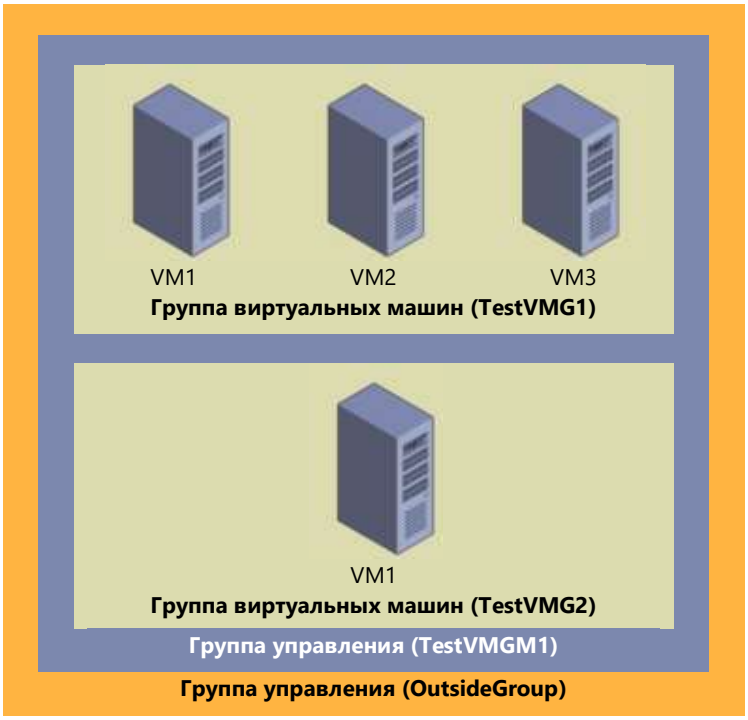


Рис. 2-5. Несколько уровней групп управления

Наконец, с помощью описанных выше средств управления можно определять, какие виртуальные машины и какие группы входят в состав других групп.

Разумеется, поддержка вложенности предоставляет новые возможности для организации структуры виртуальных машин: виртуальные машины становятся объектами, подобными объектам пользователей и компьютеров в Active Directory. Эти возможности станут для вас более понятными, когда вы начнете использовать вложенные группы, доступные в новой версии Virtual Machine Manager.

Улучшенная мобильность виртуальных машин

Возможность перемещения виртуальных машин с одного узла на другой поддерживалась с момента появления самой первой версии Hyper-V. В ранних версиях Hyper-V (при использовании Windows Server 2008) виртуальные машины можно было перемещать только в выключенном режиме (рис. 2-6): необходимо было выключить виртуальную машину, перенести ее, а затем снова включить, для этого использовались функции экспорта и импорта. В результате достигалась определенная мобильность, но простои виртуальных машин при этом были неизбежны.

Рис. 2-6. Перемещение в автономном режиме

С версии Windows Server 2008 R2 появилась возможность динамического перемещения — возможность перемещать работающую виртуальную машину. Впрочем, динамическое перемещение было возможно только между узлами кластера Hyper-V, в которых виртуальные машины находились в общем томе кластера (CSV), как показано на рис. 2-7.

Рис. 2-7. Динамическое перемещение

В Windows Server 2012 была достигнута совершенно новая степень свободы: возможность перемещать работающие виртуальные машины между любыми узлами Hyper-V одинаковой версии (рис. 2-8) вне зависимости от того, входят ли они в состав отказоустойчивого кластера.

Рис. 2-8. Перемещение между любыми узлами с одинаковой операционной системой

В Windows Server 2012 R2 возможности были вновь расширены — впервые стало возможно перемещать работающие виртуальные машины между разными версиями узлов. Теперь можно было переносить работающие виртуальные машины с любого узла Windows Server 2012 на любой узел Windows Server 2012 R2 вне зависимости от принадлежности к отказоустойчивому кластеру (рис. 2-9).

Рис. 2-9. Динамическое перемещение с узла версии 2012 на узел версии 2012 R2

В Windows Server 2016 преодолено еще одно ограничение: теперь можно перемещать виртуальные машины на узлы не только с более поздними, но и с более ранними версиями, благодаря чему администраторы получили настоящую свободу действий при управлении виртуальными машинами. Ранее динамическое перемещение было возможно либо в случае, если на обоих узлах была одинаковая версия, либо при перемещении на узел со следующей, более поздней версией Windows Server. В табл. 2-2 перечислены варианты перемещения виртуальных машин в Hyper-V для всех версий Windows Server.

Таблица 2-2. Варианты перемещения виртуальных машин

Операционная система узла	Варианты перемещения
Windows Server 2008	Перемещение в выключенном состоянии
Windows Server 2008 R2	Динамическое перемещение только между узлами кластера
Windows Server 2012	Динамическое перемещение в кластер или из него
Windows Server 2012 R2	Динамическое перемещение в кластер или из него, перемещение на более позднюю версию Windows Server
Windows Server 2016	Динамическое перемещение в кластер или из него, перемещение на более позднюю или более раннюю версию Windows Server

Windows Server 2016 — единственная версия, в которой доступна возможность динамически перемещать работающую виртуальную машину на узел с более ранней версией Windows Server (рис. 2-10).

Рис. 2-10. Перемещение виртуальной машины с узла Windows Server 2016 на узел с более ранней версией Windows Server

Для динамического перемещения работающих виртуальных машин с узла Windows Server 2016 на узлы с более ранними версиями Windows Server необходимо соблюдение следующих условий:

- оба узла должны быть членами одного и того же каталога Active Directory;
- на обоих узлах должна быть включена функция динамического перемещения.

Включение динамического перемещения производится так же, как и в прежних версиях: нужно открыть на узле окно «Параметры Hyper-V», установить флажок «Включить входящие и исходящие миграции» и затем выбрать источник, как показано на рис. 2-11.

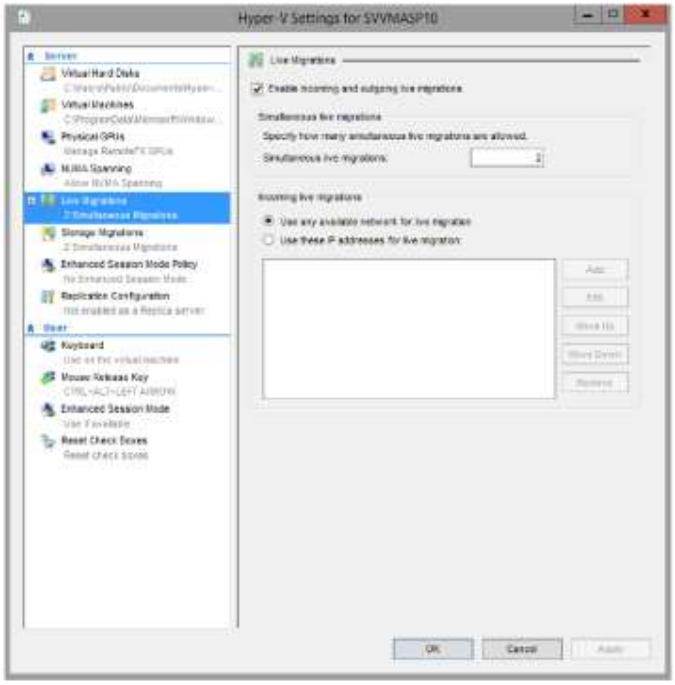


Рис. 2-11. Настройка динамического перемещения на узле

Динамическое перемещение виртуальных машин происходит так же, как в прежних версиях Windows Server. Возможны три способа. Можно использовать:

- диспетчер Hyper-V на узле;
- скрипт на Windows PowerShell;
- Virtual Machine Manager (не входит в состав Windows Server).

В диспетчере Hyper-V нажмите правую кнопку мыши на виртуальной машине, которую нужно переместить, и выберите в контекстном меню команду «Переместить», как показано на рис. 2-12.

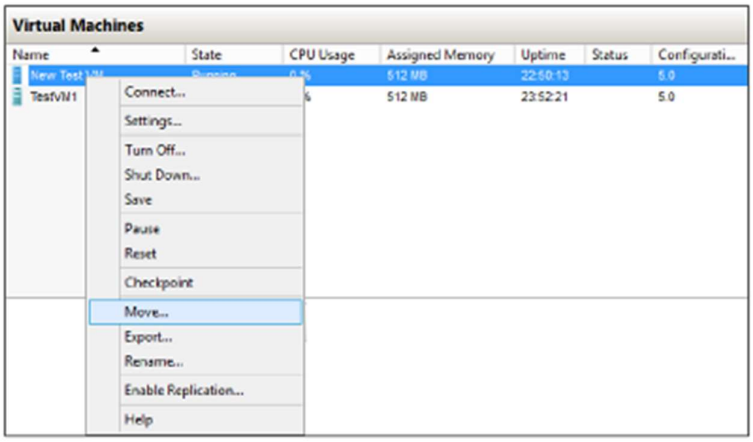


Рис. 2-12. Контекстное меню виртуальной машины

Эту же задачу можно выполнить с помощью Windows PowerShell, используя командлет Move-VM. Команда, показанная в следующем примере, перемещает виртуальную машину с именем New Test VM на сервер назначения с именем Hyper-Server:

```
PS C:\> Move-VM "New Test VM" Hyper-Server
```

Примечание. Указанный выше командлет перемещает виртуальную машину на узел Hyper-V в расположение по умолчанию

Обратите внимание: любую виртуальную машину можно динамически переместить с Windows Server 2012 на любой хост с более поздней версией Windows Server, но переносить с Windows Server 2016 на Windows Server 2012 R2 можно только виртуальные машины версии 5.0. Уточнить версию можно в диспетчере Hyper-V (рис. 2-13) или с помощью командлета Get-VM в Windows PowerShell.

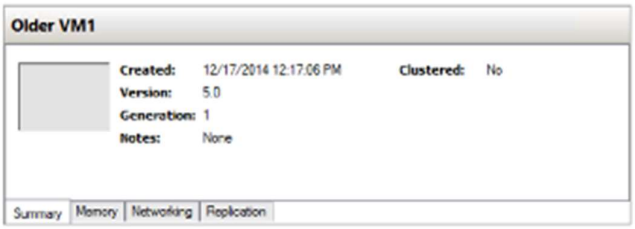


Рис. 2-13. Номер версии виртуальной машины

Примечание. Не следует путать версии и поколения. Виртуальные машины первого и второго поколения могут быть версии 5.0. Номер версии виртуальной машины относится к версии Windows Server, в которой были созданы виртуальные машины, а поколение относится к тому, какое оборудование, поддерживающее виртуализацию, доступно для виртуальных машин.

Также следует отметить, что, хотя и можно динамически переносить виртуальные машины из отказоустойчивого кластера, скорее всего, вы будете использовать новые возможности перемещения внутри такого кластера. Впервые после выпуска Windows Server 2003 отказоустойчивые кластеры поддерживают кластеры смешанного режима. Это означает, что можно обновить узлы кластера Windows Server 2012 R2 до Windows Server 2016, сохранив их членство в кластере. Благодаря улучшениям мобильности можно перемещать виртуальные машины между более старыми и более новыми узлами кластера в рамках общей стратегии его обновления.

Версия конфигурации виртуальной машины

В Windows Server 2016 изменен процесс обновления виртуальных машин. Ранее при переносе виртуальных машин на более новую версию Hyper-V они обновлялись автоматически, при этом не всегда можно было легко определить, какие именно виртуальные машины были импортированы из прежней версии Hyper-V, а какие созданы уже в новой версии. Это было обусловлено тем, что при обновлении хоста автоматически обновлялась и версия конфигурации виртуальных машин.

Основная проблема заключалась в отсутствии возможности отката виртуальных машин к прежней версии конфигурации. Версия виртуальной машины определяет, с какими версиями Hyper-V совместимы конфигурация виртуальной машины, сохраненное состояние и файлы моментальных снимков. В Windows Server 2016 обновление версии конфигурации виртуальных машин больше не производится автоматически. Благодаря этому можно перемещать виртуальные машины на сервер с более ранней версией Hyper-V — например, Windows Server 2012 R2. В этом случае у вас не будет доступа к новым возможностям виртуальных машин до тех пор, пока вы вручную не обновите версию их конфигурации.

Все возможности виртуальных машин остаются совместимыми, включая динамическое перемещение виртуальных машин, динамическое перемещение хранилища и динамическое выделение памяти. Таким образом, обновление виртуальных машин теперь выполняется вручную и отделено от обновления физического узла. Важно отметить, что после обновления версии конфигурации виртуальных машин вернуть прежнюю версию невозможно. Если вы используете виртуальные машины, которые были созданы в Windows Server 2012 R2, у вас не будет доступа к новым возможностям виртуальных машин до тех пор, пока вы не обновите версию их конфигурации вручную.

Виртуальные машины с версией конфигурации 5.0 совместимы с Windows Server 2012 R2, их можно использовать под управлением Windows Server 2012 R2 и Windows Server 2016. Виртуальные машины с версией конфигурации 6.0 совместимы с Windows Server 2016, но они не будут работать в Hyper-V под управлением Windows Server 2012 R2.

В табл. 2-3 перечислены поддерживаемые версии конфигурации в разных версиях Windows.

Таблица 2-3. Поддерживаемые версии конфигурации виртуальных машин в разных версиях Windows

Версия Windows на сервере Hyper-V	Поддерживаемые версии конфигурации виртуальных машин
Юбилейное обновление Windows 10	8.0, 7.1, 7.0, 6.2, 5.0
Windows Server 2016	8.0, 7.1, 7.0, 6.2, 5.0
Windows 10 (сборка 10565 или более поздняя)	7.0, 6.2, 5.0
Windows 10 (сборка до 10565)	6.2, 5.0
Windows Server 2012 R2	5.0
Windows 8.1	5.0

Обновление версии конфигурации

Чтобы обновить версию конфигурации, выключите виртуальную машину, запустите командную строку Windows PowerShell с правами администратора и выполните следующую команду:

```
Update-VmVersion vmname или vmobject.
```

Чтобы проверить версию конфигурации запущенной в Hyper-V виртуальной машины, выполните с правами администратора следующую команду в командной строке:

```
Get-VM * | Format-Table Name, Version
```

Для иллюстрации процесса обновления в следующем примере мы определим версию конфигурации виртуальной машины, импортированной с узла под управлением Windows Server 2012 R2, а затем покажем, как обновить версию конфигурации. В этом случае, как и следовало ожидать, в диспетчере Hyper-V отображается версия конфигурации 5.0 (рис. 2-14).

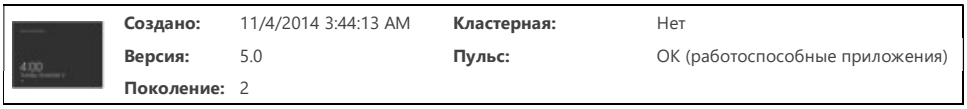


Рис. 2-14. Номер версии виртуальной машины

Для подтверждения можно использовать Windows PowerShell:

```
PS C:\Users\Administrator> Get-VM vm02 |Format-Table Name, Version
Name                                                    Version
vm02                                                    5.0
```

Как уже было отмечено, для обновления версии конфигурации необходимо выключить виртуальную машину и выполнить следующую команду Windows PowerShell:

```
PS C:\Users\Administrator> Update-VMVersion vm02
Confirm
Are you sure you want to perform this action?
Performing a configuration version update of "vm02" will prevent it from being migrated to or imported
on previous versions of Windows. This operation is not reversible.
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): Y
PS C:\Users\Administrator>
```

Проверим версию конфигурации в диспетчере Hyper-V: теперь отображается версия 6.0, как показано на рис. 2-15.

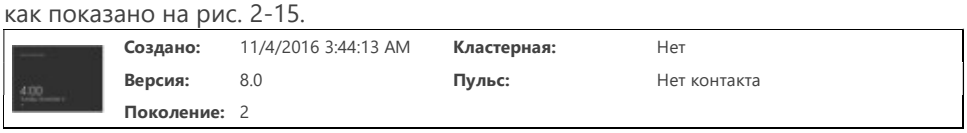


Рис. 2-15. Обновленный номер версии

Для подтверждения можно вновь использовать Windows PowerShell:

```
PS C:\Users\Administrator> Get-VM vm02 |Format-Table Name, Version
Name                    Version
vm02                    8.0
```

Если после обновления версии конфигурации виртуальных машин происходит сбой запуска, попробуйте включить безопасную загрузку, а затем выполните следующую команду Windows PowerShell:

```
Set-VMFirmware -VMName "имя_виртуальной_машины" -SecureBootTemplate MicrosoftWindows
```

Версия конфигурации виртуальной машины успешно обновлена. Теперь к этой виртуальной машине можно применять новые возможности, которые появились в Windows Server 2016.

Особенности процесса обновления

Перед обновлением версии конфигурации виртуальных машин необходимо учесть ряд особенностей:

- Перед тем как обновлять версии конфигурации, необходимо выключить виртуальную машину.
- Процесс обновления версии конфигурации односторонний: после обновления версии конфигурации виртуальных машин с 5.0 до 8.0 вернуть прежнюю версию невозможно. Обновленную виртуальную машину нельзя переместить на сервер под управлением Windows Server 2012 R2.
- Командлет Update-VMVersion блокируется в кластере Hyper-V, если функциональный уровень кластера — Windows Server 2012 R2. При этом по-прежнему можно перемещать виртуальные машины между всеми узлами кластера Hyper-V, если на одних узлах кластера используется Windows Server 2012 R2, а на других — Windows Server 2016.

Примечание. Более подробная информация о процессе обновления доступна по адресу https://technet.microsoft.com/en-us/windows-server-docs/compute/hyper-v/deploy/upgrade-virtual-machine-version-in-hyper-v-on-windows-or-windows-server.

Новый формат файлов конфигурации

После обновления версии конфигурации виртуальных машин, как описано в предыдущем разделе, для них будет использоваться новый формат файлов конфигурации. В нем применяется расширение VMCX для файлов данных конфигурации и VMRS — для файлов данных состояния выполнения. Новый формат файлов является двоичным, поэтому редактировать файлы вручную невозможно. В новом формате файлов повышена эффективность чтения и записи данных конфигурации виртуальных машин, снижена вероятность повреждения данных при отказе системы хранения данных, а кроме того, обеспечивается более высокая эффективность в целом.

На рис. 2-16 показан новый формат файлов конфигурации виртуальных машин: расширение VMCX для данных конфигурации и VMRS для данных состояния выполнения.

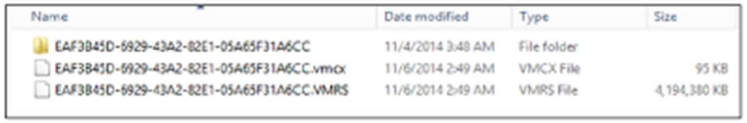


Рис. 2-16. Файлы конфигурации виртуальных машин

Вы можете определить расположение конфигурации виртуальных машин и связанную с ними информацию, отобразив свойства виртуальных машин с помощью Windows PowerShell:

```
PS C:\Users\Administrator> Get-VM -Name vm02 |Format-List *
VMName                                : vm02
VMId                                  : eaf3b45d-6929-43a2-82e1-05a65f31a6cc
Id                                    : eaf3b45d-6929-43a2-82e1-05a65f31a6cc
Name                                  : vm02
State                                 : Running
IntegrationServicesState              : Update required
```

```
OperationalStatus                     : {Ok}
PrimaryOperationalStatus              : Ok
SecondaryOperationalStatus            :
StatusDescriptions                    : {Operating normally}
PrimaryStatusDescription              : Operating normally
SecondaryStatusDescription            :
Status                                : Operating normally
Heartbeat                             : OkApplicationsHealthy
ReplicationState                      : Disabled
ReplicationHealth                     : NotApplicable
ReplicationMode                       : None
CPUUsage                              : 0
MemoryAssigned                        : 4294967296
MemoryDemand                          : 600834048
MemoryStatus                          :
SmartPagingFilelnUse                  : False
Uptime                                : 22:37:12
IntegrationServicesVersion            : 6.3.9600.16384
ResourceMeteringEnabled               : False
AutomaticCriticalErrorAction          : Pause
AutomaticCriticalErrorActionTimeout   : 30
ConfigurationLocation                 : c:\vmdata\vm02\vm02
SnapshotFileLocation                  : c:\vmdata\vm02\vm02
CheckpointType                        : Production
AutomaticStartAction                  : StartifRunning
AutomaticStopAction                   : Save
AutomaticStartDelay                   : 0
SmartPagingFilePath                   : c:\vmdata\vm02\vm02
NumaAligned                           : True
NumaNodesCount                        : 1
NumaSocketCount                       : 1
Key                                   : Microsoft.HyperV.PowerShell.VirtualMachineObjectKey
IsDeleted                             : False
ComputerName                          : SIGGPB04-T1
Version                               : 8.0
Notes                                 :
Generation                            : 2
Path                                  : c:\vmdata\vm02\vm02
CreationTime                          : 11/4/2016 3:44:13 AM
IsClustered                           : False
SizeOfSystemFiles                     : 97132
ParentSnapshotId                      :
ParentSnapshotName                    :
MemoryStartup                         : 4294967296
DynamicMemoryEnabled                  : False
MemoryMinimum                         : 536870912
MemoryMaximum                         : 1099511627776
ProcessorCount                        : 1
RemoteFxAdapter                       :
NetworkAdapters                       : {Network Adapter}
FibreChannelHostBusAdapters           : {}
ComPort1                              : Microsoft.HyperV.PowerShell.VMComPort
ComPort2                              : Microsoft.HyperV.PowerShell.VMComPort
FloppyDrive                           :
DVDDrives                             : {}
HardDrives                            : {Hard Drive on SCSI controller number 0 at location 0}
VMIntegrationService                  : {Time Synchronization, Heartbeat, Key-Value Pair Exchange,
Shutdown...}
```

Рабочие контрольные точки

В Windows Server 2016 реализован новый способ создания контрольных точек для виртуальных машин в рабочей среде — для этого используются рабочие контрольные точки. *Контрольная точка* — это запись состояния виртуальной машины в определенный момент времени, с помощью которой можно вернуть виртуальную машину в более раннее состояние. До появления Windows Server 2016 контрольные точки использовались при тестировании и разработке, но использовать их в рабочей среде не рекомендовалось.

В новой версии контрольные точки работают так же, как в Windows Server 2012 R2, но теперь они полностью поддерживаются и в рабочей среде. Основных причин для этого две:

- для создания контрольных точек теперь используется служба моментального снимка тома (VSS) вместо сохраненного состояния;

- восстановление контрольной точки выполняется аналогично восстановлению резервной копии системы.

Примечание. VSS используется для создания рабочих контрольных точек только в виртуальных машинах с операционной системой Windows. Виртуальные машины с Linux выполняют эту задачу путем записи системных буферов и создания контрольной точки, консистентной для файловой системы.

Если требуется создавать контрольные точки с помощью сохраненного состояния, можно, как и прежде, использовать стандартные контрольные точки для виртуальной машины. Однако при этом по умолчанию для новых виртуальных машин создаются рабочие контрольные точки с возможностью отката к стандартным контрольным точкам.

В определенных случаях администратору может потребоваться отключить контрольные точки для определенных виртуальных машин по эксплуатационным соображениям. В Windows Server 2016 такая возможность есть: можно включать и выключать контрольные точки на отдельных виртуальных машинах. Эта возможность существенно повышает гибкость работы и предоставляет администраторам Hyper-V возможность эффективного управления ресурсами и их оптимизации.

На рис. 2-17 показано изменение параметров виртуальной машины: включение и выключение контрольных точек, а также включение рабочих контрольных точек. По умолчанию флажок «Включить контрольные точки» установлен, создание рабочих контрольных точек разрешено, а если их невозможно создать, то будут создаваться стандартные контрольные точки.

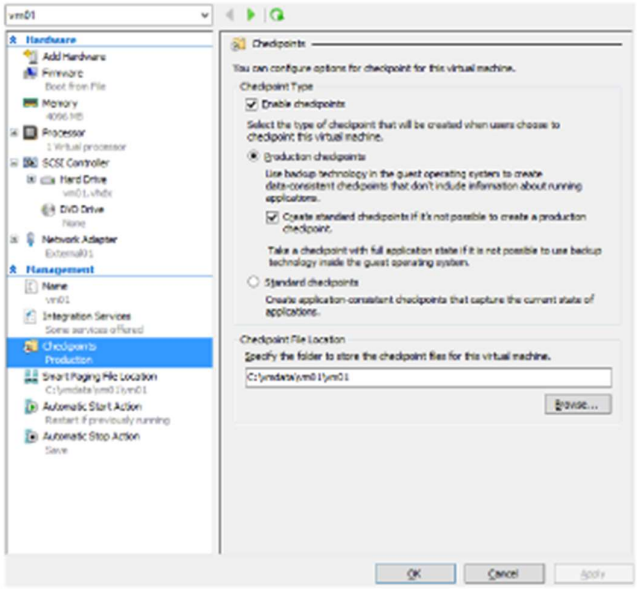


Рис. 2-17. Конфигурированиерабочих контрольных точек для виртуальной машины

Чтобы создать новую рабочую контрольную точку для виртуальной машины, необходимо их включить. Для этого нажмите правую кнопку мыши на этой виртуальной машине в диспетчере Hyper-V и выберите в меню команду Checkpoint, как показано на рис. 2-18.

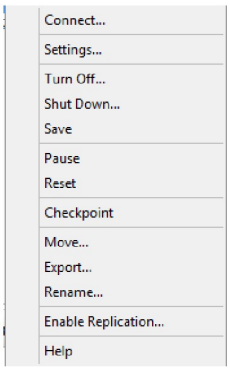


Рис. 2-18. Создание новой контрольной точки виртуальной машины с помощью меню.

> **Примечание.** Если использование рабочих контрольных точек выключено, команда Checkpoint не отображается в контекстном меню для этой виртуальной машины.

При создании рабочей контрольной точки появляется сообщение, показанное на рис. 2-19, которое подтверждает успешное её создание.

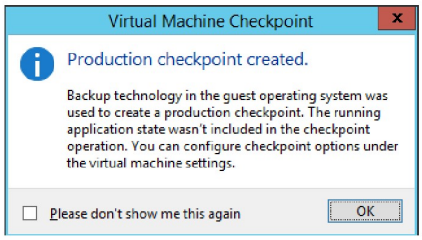


Рис. 2-19. Подтверждение успешного создания рабочей контрольной точки

Разумеется, все это можно сделать и с помощью Windows PowerShell.

«Горячие» добавление и удаление сетевых адаптеров и памяти

При использовании Windows Server 2016 больше не нужны плановые отключения, для того чтобы добавить или уменьшить оперативную память виртуальных машин, размещенных в Hyper-V. Добавление и удаление сетевых адаптеров также производятся без выключения виртуальных машин. Теперь поддерживаются «горячие» добавление и удаление сетевых адаптеров и памяти. Это важное усовершенствование, которое существенно упрощает работу администраторов Hyper-V. В физической среде установка дополнительной оперативной памяти или новой платы сетевого адаптера — достаточно длительный процесс, требующий планирования отключения и простоя систем. Новое усовершенствование дает возможность все сделать без простоев. И поставщики услуг, и компании могут увеличивать и уменьшать объем выделяемой для виртуальных машин памяти за считанные секунды, используя диспетчер Hyper-V или Windows PowerShell.

> **Примечание.** «Горячее» добавление памяти поддерживается для гостевых виртуальных машин первого и второго поколений, работающих под управлением Windows Server 2016. В Windows Server 2012 R2 и более ранних версиях данная фунциональность недоступна

«Горячие» добавление и удаление памяти

На рис. 2-20 показан диспетчер Hyper-V с двумя работающими на нем виртуальными машинами: VM1 и VM2. В диспетчере Hyper-V видно, что VM2 — виртуальная машина первого поколения, а в окне параметров показано, что этой виртуальной машине предоставлено 2 ГБ (2048 МБ) оперативной памяти.

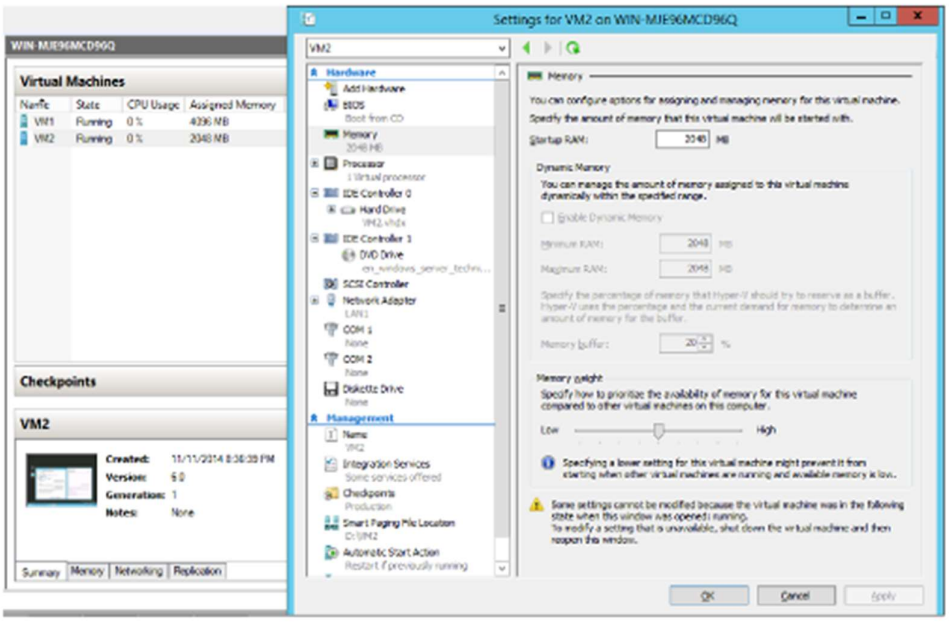


Рис. 2-20. Параметры памяти виртуальной машины первого поколения

При подключении к этой виртуальной машине видно, что на ее рабочем столе запущены два приложения: окно «Дата и время», в котором расположены часы с текущим временем, и диспетчер задач, отображающий использование памяти и показывающий, что доступно 2 ГБ (рис. 2-21).

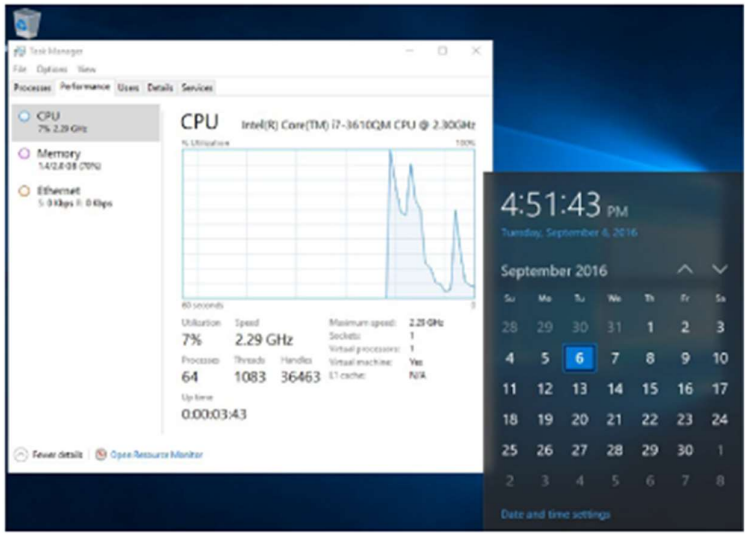


Рис. 2-21. Диспетчер задач виртуальной машины отображает использование памяти

В окне параметров VM2 измените объем оперативной памяти, используемый этой виртуальной машиной, с 2 до 4 ГБ и нажмите на кнопку «Применить», не останавливая работу виртуальной машины. Через несколько секунд в диспетчере Hyper-V станет видно, что виртуальная машина VM2 теперь использует 4 ГБ оперативной памяти, при этом перезагрузка не требуется (рис. 2-22).

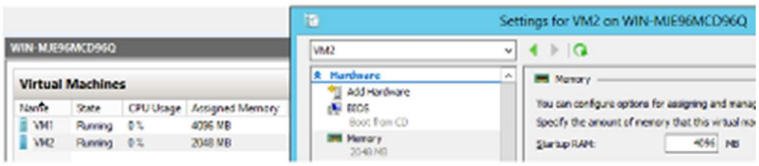


Рис. 2-22. Изменение объема памяти виртуальной машины в процессе ее работы

В окне подключения к виртуальной машине видно, что часы на VM2 по-прежнему работают, в диспетчере задач отображается 4 ГБ доступной памяти, которая была добавлена с помощью доступной в Windows Server 2016 функции «горячего» добавления, как показано на рис. 2-23.

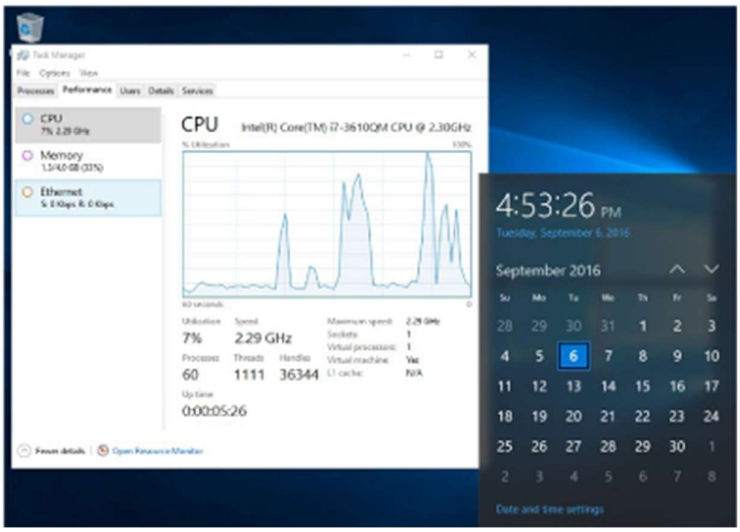


Рис. 2-23. Диспетчер задач отображает новый объем доступной памяти на виртуальной машине

«Горячие» подключение и удаление сетевых адаптеров

Добавление и удаление сетевых адаптеров работающей виртуальной машины без ее выключения поддерживаются только для виртуальных машин второго поколения с ОС Windows и Linux. В число поддерживаемых версий Windows входит Windows Server 2016.

В следующем примере при подключении к виртуальной машине второго поколения с именем VM1 и открытии папки «Сетевые подключения» видно, что у этой виртуальной машины есть только один сетевой адаптер с именем «Ethernet», как показано на рис. 2-24.

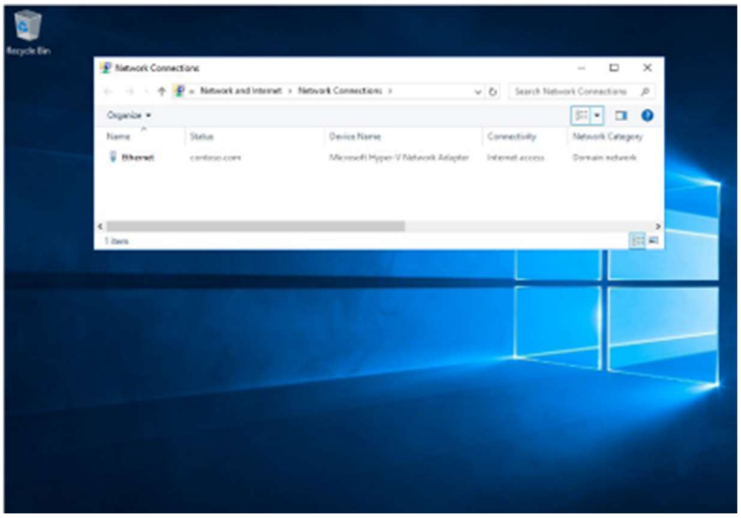


Рис. 2-24. Виртуальная машина с одним сетевым адаптером

Чтобы добавить еще один сетевой адаптер, в окне «Настройки» на странице «Добавление оборудования» выберите «Сетевой адаптер», как показано на рисунке 2-25.

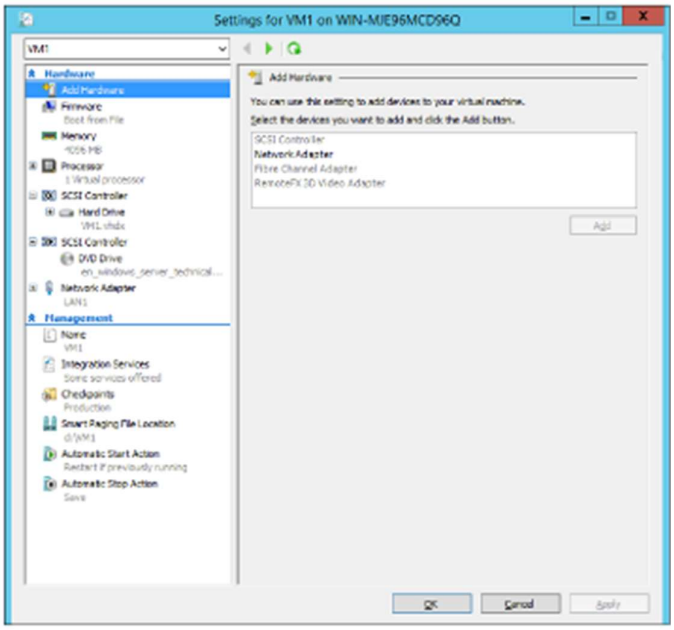


Рис. 2-25. Добавление сетевого адаптера в виртуальную машину

Примечание. Если это виртуальная машина первого поколения, то вариант «Сетевой адаптер» на странице «Добавление оборудования» окна «Настройки» будет недоступен: первое поколение не поддерживает «горячие» добавление и удаление сетевых адаптеров.

Нажмите на кнопку «Применить», чтобы изменения вступили в силу. Через несколько секунд новый сетевой адаптер будет установлен без выключения виртуальной машины. Новый сетевой адаптер появится в папке «Сетевые подключения» в окне подключения к виртуальной машине. На рис. 2-26 видно, что второй сетевой адаптер успешно добавлен.

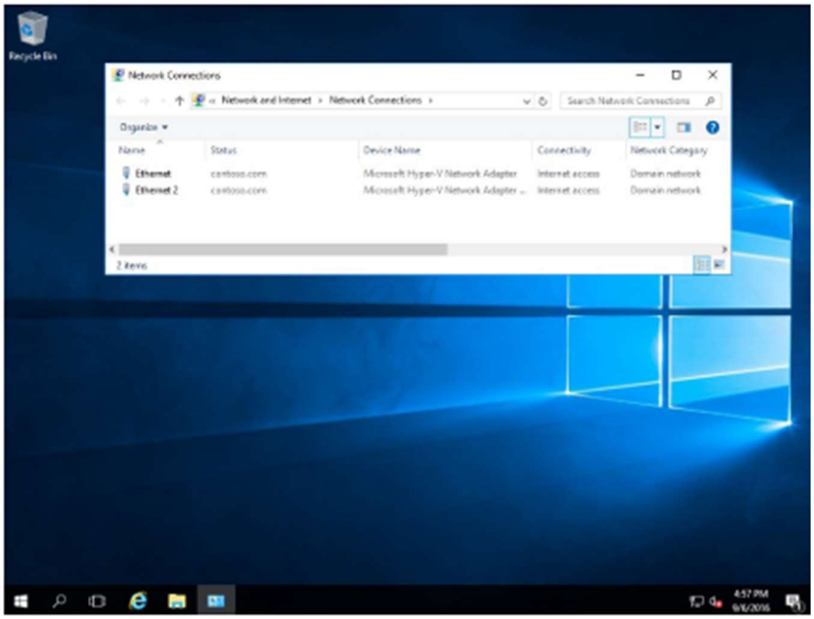


Рис. 2-26. Добавлен второй сетевой адаптер

Отказоустойчивый кластер

Авторы: Джон Марлин (John Marlin), Колин Робинсон (Colin Robinson)

В этом разделе описаны следующие усовершенствования отказоустойчивой кластеризации в Windows Server 2016:

- новый тип свидетеля (cloud witness) на базе облака Azure;
- усовершенствования общих VHDX-файлов;
- улучшенные журналы событий кластеров;
- активный дамп памяти;
- диагностика сетевых имен;
- последовательное обновление операционных систем в кластере;
- кластеры в рабочей группе и мультидоменной среде;
- SMB Multichannel и кластерные сети с использованием нескольких сетевых адаптеров;
- усовершенствования виртуальных машин.

Кроме того, Колин Робинсон проводит подробную демонстрацию последовательного обновления отказоустойчивого кластера.

Создание облака-свидетеля с помощью Azure

Начиная с Windows Server 2008 в каждой версии отказоустойчивой кластеризации внедрялся новый тип кворума в дополнение к уже существующим. Этот принцип не изменился: в Windows Server 2016 реализован тип кворума «облако-свидетель» (cloud witness), его можно создать в облаке с помощью Azure.

Этот тип кворума использует общедоступное облако Azure в качестве точки арбитража (свидетеля) для кластера. Такую конфигурацию можно построить без использования дополнительного сайта, она предназначена главным образом для многосайтовых кластеров. В этой конфигурации предоставляется кворум для следующих ситуаций:

- распределенные кластеры, не имеющие выделенного сайта для размещения файлового или иного ресурса в качестве дополнительного голоса для кворума;
- кластеры, в которых не используется общее хранилище;
- гостевые кластеры, размещенные в Azure;
- гостевые кластеры, размещенные в частных облаках;
- кластеры, использующие непосредственно подключенное хранилище (DAS).
- Кластеры в рабочей группе и разных доменах

Облако-свидетель работает аналогично файловому ресурсу-свидетелю и по такой же логике: оно не содержит копию базы данных кластера и выступает в качестве решающего голоса для предотвращения недоступности кластера, когда несколько узлов в одном и том же кластере не могут обмениваться данными друг с другом.

Для настройки облака-свидетеля требуется подписка Azure. Чтобы ее получить, выполните следующие действия:

1. Войдите на портал управления Azure (https://portal.azure.com) и создайте для этого свидетеля учетную запись хранения (если вы не знаете, как это сделать, перейдите по адресу https://azure.microsoft.com/documentation/articles/storage-create-storage-account/).

2. После создания учетной записи хранения выберите ее на портале, затем нажмите «Ключи доступа». Скопируйте основной ключ доступа, чтобы использовать его в дальнейшем.

3. В консоли диспетчера отказоустойчивости кластеров настройте кворум для облака-свидетеля. Для этого нажмите правую кнопку мыши на имени кластера, выберите в контекстном меню команду «Дополнительные действия», а затем — «Настроить параметры кворума кластера», как показано на рис. 2-27.

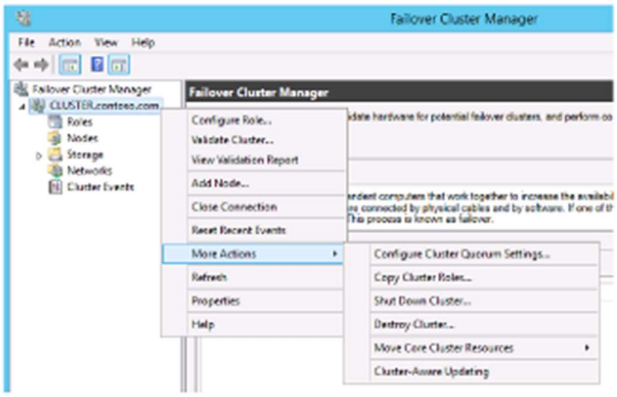


Рис. 2-27. Параметры кворума кластера

4. Откроется мастер настройки кворума кластера. На странице «Выбор параметра конфигурации кворума» нажмите «Выбрать свидетель кворума», как показано на рис. 2-28.

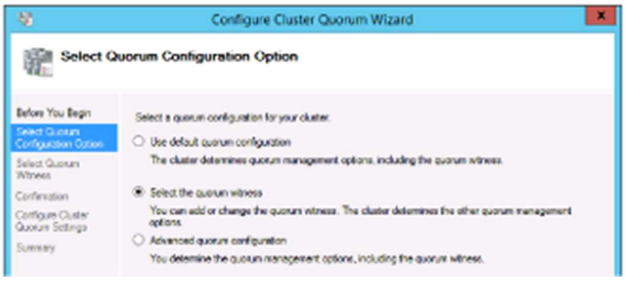


Рис. 2-28. Выбор типа кворума

5. На странице «Выбор свидетеля кворума» нажмите «Настроить облако-свидетель», как показано на рис. 2-29.

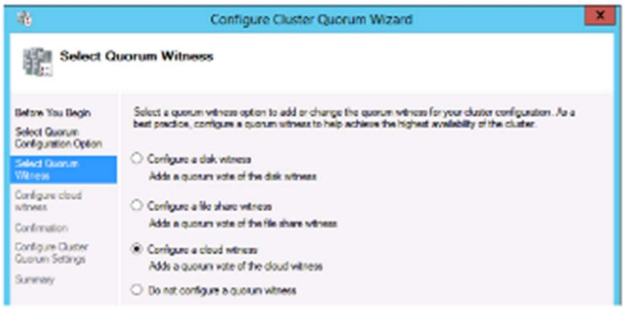


Рис. 2-29. Настройка облака-свидетеля

6. На странице «Настройка облачного следящего компонента» введите имя учетной записи хранения, созданной на портале управления, основной ключ учетной записи хранения Azure и конечную точку службы Azure, как показано на рис. 2-30.

Рис. 2-30. Ввод данных учетной записи хранения

Примечание. То же самое можно выполнить и в Windows PowerShell, запустив следующую команду:

```
Set-ClusterQuorum -CloudWitness -AccountName MyWitness -AccessKey <ключ учетной записи хранения> -Endpoint core.windows.net
```

Для использования облака-свидетеля требуется выполнение двух условий:

- необходима действующая подписка Azure;
- у всех узлов должен быть доступ к Интернету и к Azure.

Кроме того, как и в случае с файловым ресурсом-свидетелем, можно использовать одну и ту же учетную запись Azure или контейнер для нескольких кластеров.

Усовершенствования общих VHDX-файлов

В Hyper-V в Windows Server 2008 и более поздних версиях гостевые кластеры можно создавать в виде виртуальных машин. Тем не менее для создания общего хранилища любого типа требовалось использовать iSCSI. В Windows Server 2012 в качестве второго варианта общих хранилищ появилась поддержка виртуальных адаптеров Fibre Channel для виртуальных машин.

Впрочем, с точки зрения поставщика услуг, виртуальные адаптеры Fibre Channel не всегда были целесообразным вариантом. Такой адаптер открывает пользователю доступ к физической инфраструктуре хранилища (аналогично физическому интерфейсу iSCSI в системах хранения данных). Но если поставщик услуг создаст виртуальную машину и добавит поддержку iSCSI для того, чтобы клиент мог создавать общие диски, клиент вряд ли этому обрадуется: с него будет взиматься плата за дополнительную виртуальную машину.

Поэтому в качестве дополнительного варианта в Windows Server 2012 R2 была реализована поддержка общих VHDX-файлов. При использовании общих VHDX-файлов гостевые кластеры получают нужное им общее хранилище без доступа к физической инфраструктуре хранения данных. Таким образом, был добавлен еще один вариант использования общих дисков, но и в нем были определенные ограничения. Некоторые из этих ограничений были устранены в Windows Server 2016.

Предположим, у вас есть общий диск VHDX, он почти заполнен и нужно увеличить его размер. В Windows Server 2012 R2 это невозможно было сделать без простоя системы, поскольку, чтобы увеличить размер, требовалось выключать виртуальные машины. Для бизнеса, работающего непрерывно, это не вполне удобно. В Windows Server 2016 есть возможность изменения размера диска без выключения виртуальной машины.

Чтобы расширить диск, выполните следующие действия:

1. Откройте диспетчер отказоустойчивости кластеров, нажмите правую кнопку мыши на виртуальной машине и выберите «Параметры».
2. Выберите диск, который нужно увеличить, как показано на рис. 2-31.

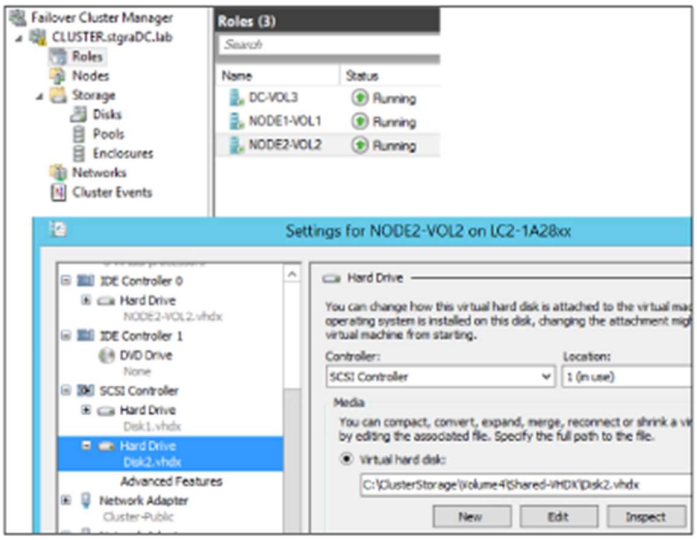


Рис. 2-31. Параметры жесткого диска

3. Выберите «Правка», чтобы запустить мастер изменения виртуального жесткого диска. Доступен только вариант «Расширить», поэтому он уже выбран, как показано на рис. 2-32.

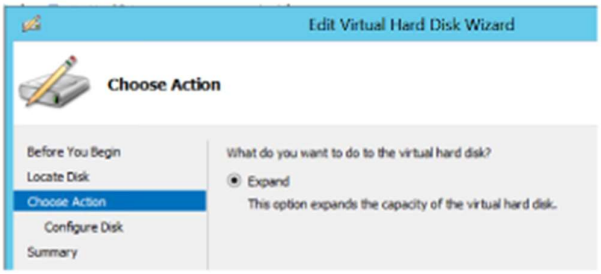


Рис. 2-32. Расширение жесткого диска

4. На странице «Настройка диска» введите нужный размер виртуального жесткого диска, как показано на рис. 2-33.

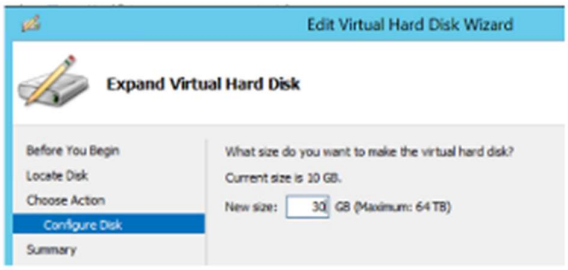


Рис. 2-33. Выбор нового размера диска

Примечание. То же самое можно выполнить в Windows PowerShell с помощью следующего командлета:
Resize-VHD -Path C:\ClusterStorage\Volume4\Shared-VHDX\Disk2.vhdx -SizeBytes 32212254720

5. По завершении работы мастера откройте виртуальную машину и расширьте том в диспетчере серверов, как показано на рис. 2-34.

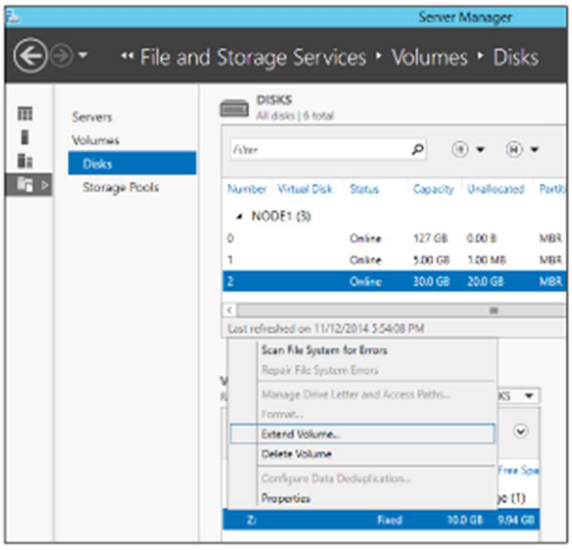


Рис. 2-34. Расширение тома в диспетчере серверов

Пока виртуальные машины работают в кластере Hyper-V, время от времени требуется создавать их резервную копию. В Windows Server 2012 R2 на физическом сервере нельзя создать резервную копию общего диска VHDX, присоединенного к виртуальной машине. Поскольку это общий диск, доступ к нему заблокирован, в том числе для резервного копирования. В Windows Server 2016 его можно указать в качестве виртуального жесткого диска (VHD) для создания резервной копии.

Виртуальные машины, включающие общие диски VHDX, теперь также могут входить в состав реплики Hyper-V. В прежних версиях Windows Server это было невозможно. Благодаря усовершенствованиям общих VHDX-дисков теперь можно не только реплицировать виртуальные машины, но и выбирать любые или все общие VHDX-диски для репликации, как показано на рис. 2-35.

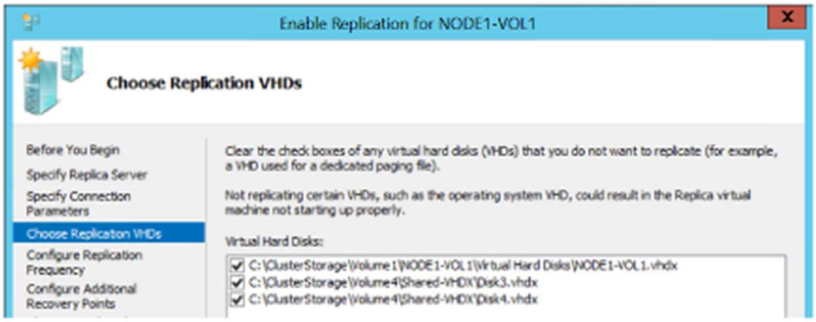


Рис. 2-35. Выбор VHD-файлов реплики

Улучшенные журналы событий кластеров

Быстро устранять проблемы в случае возникновения неполадок в работе отказоустойчивых кластеров помогает сбор как можно большего объема важной информации. Своевременное получение необходимых данных может оказаться критически важным для восстановления обслуживания. С учетом всех требований по уровню обслуживания правильная диагностика для многих компаний имеет первостепенное значение.

С точки зрения диагностики первым усовершенствованием в Windows Server 2016 стал журнал диагностики кластера. Журнал событий кластера всегда был полезным источником информации

для выявления ошибок, их причин, операций во время ошибки и т. п. Однако использовать журнал кластера, чтобы определить, к примеру, конфигурацию кластера, не так просто.

Например, можно выяснить какие ресурсы входят в состав кластера, поскольку эта информация включается в протокол запуска кластерной службы. Но собирать ее придется по частям из множества строк, как показано в следующем примере:

```
<Networks><vector len='4'>
<item><obj sig='NETW' id='1f509983-2478-4630-af8a-e13d2c486172' name='Cluster Network 2'/></item>
<item><obj sig='NETW' id='967df8d8-0f94-4d02-93d5-046fa1ce2369' name='Cluster Network 1'/></item>
<item><obj sig='NETW' id='99e6e621-0de5-4a1c-a468-a68057ee6278' name='Cluster Network 4'/></item>
<item><obj sig='NETW' id='a171b564-6b89-4c7d-91a5-f2dcbe450fbe' name='Cluster Network 3'/></item>
</vector>

<LIVE id='.Live' name='.Live'>
<NODE id='2' name='2012R2N1'>
<ITFC id='62edcfaa-fb25-4108-ac2b-236b3520af3f' name='2012R2N1 - WAN'/>
<ITFC id='884676e9-6df6-4a26-a423-c9b113a99056' name='2012R2N1 - iSCSI 2'/>
<ITFC id='65fa8c93-c242-45f0-88ab-9e26f0845c0b' name='2012R2N1 - Public'/>
<ITFC id='10962cca-3015-4380-8011-76e47ef575c4' name='2012R2N1 - iSCSI 1'/>
</NODE>
<NODE id='1' name='2012R2N2'>
<ITFC id='9e79c4c8-8d78-4d14-ab51-7a39d7191c4a' name='2012R2N2 - WAN'/>
<ITFC id='09569c99-262d-4f6b-95e4-f69185669f2b' name='2012R2N2 - iSCSI2'/>
<ITFC id='aa8f05e2-eeae-452c-9960-3905d0319fe1' name='2012R2N2 - Public'/>
<ITFC id='95d0074f-6ff6-4b5f-89a7-454fe2306332' name='2012R2N2 - iSCSI1'/>
</NODE>
</LIVE>
```

Для получения столь малого объема информации необходимо разобрать порядка 1000 строк в журналах, затратив на это немало времени и усилий. Для поиска другой информации о конфигурациях может потребоваться повторный анализ тех же самых, а также многих других записей в журналах.

Сведения такого типа можно получить из других журналов или из реестра, но в этом случае придется просматривать данные из нескольких файлов. Если вы не находитесь за компьютером лично, а поручили сбор журналов кому-то другому, то этот человек может собрать не все журналы, что, в свою очередь, приведет к дополнительным задержкам при получении нужной информации.

Преодолению этих сложностей было уделено серьезное внимание при работе над Windows Server 2016, поскольку они касаются отказоустойчивой кластеризации. Журнал диагностики кластера был переработан: в Windows Server 2016 в журнал при его создании добавляется дополнительная информация, сгруппированная по разделам, к которой можно быстро получить доступ, как продемонстрировано ниже:

```
[=== Cluster ===]
В этом разделе содержится информация о кластере: его версия, время работы, узел, с которого получен
журнал и т. п.

[=== Resources ===]
Список всех ресурсов (включая GUID), конфигурации и параметры этих ресурсов.

[=== Groups ===]
Список всех групп (включая GUID), конфигурации и параметры этих групп, узел-владелец и пр.

[=== Resource Types ===]
Список всех типов ресурсов (включая GUID), конфигурации и параметры этих типов ресурсов.

[=== Nodes ===]
В этом разделе содержится информация об узлах, включая версию, время работы, идентификаторы узлов
и пр.

[=== Networks ===]
В этом разделе содержится информация о сетях, включая роли, сетевые схемы, метрики, наличие поддержки
RSS
и пр.

[=== Network Interfaces ===]
В этом разделе содержится информация о сетях, включая имена, IP-адреса и т. п.

[=== System ===]
```

```
Все записи журнала событий системы, источником которых является отказоустойчивая кластеризация
(Failover Clustering)

[=== Microsoft-Windows-FailoverClustering/Operational logs ===]
Все события канала Microsoft-Windows-FailoverClustering/Operational, предоставляющие информацию
о формах кластера, присоединении к узлам, перемещениях групп и пр.

[=== Microsoft-Windows-ClusterAwareUpdating-Management/Admin logs ===]
Все события канала Microsoft-Windows-ClusterAwareUpdating-Management/Admin, предоставляющие
информацию
о кластерном обновлении

[=== Microsoft-Windows-ClusterAwareUpdating/Admin logs ===]
Все события канала Microsoft-Windows-ClusterAwareUpdating/Admin, предоставляющие информацию
о кластерном обновлении

[=== Microsoft-Windows-FailoverClustering/DiagnosticVerbose ===]
Это новый канал событий, он выдает данные журнала кластера, аналогичные уровню отладки 5, без
необходимости задавать этот уровень. Эту информацию можно использовать для более подробного изучения
вызовов и действий в кластере.

[=== Cluster Logs ===]
Это обычная информация журнала кластера.
```

Как видим, теперь этот журнал содержит очень много всевозможных данных. Использование одного-единственного журнала может помочь снизить время поиска нужных сведений и быстрее устранить неполадки. Вместо просмотра трех или более файлов, которые могут загружаться в трех разных приложениях в своих собственных форматах, теперь достаточно просмотреть только один файл.

Еще одна полезная особенность этого журнала в том, что, если при его создании не указать никаких параметров, будет приведено все содержимое каналов событий (например, System). Тем не менее можно получить историю событий, относящуюся к определенной категории. В случае если есть возможность воспроизвести условия ошибки и не требуется полная история, можно, например, сгенерировать журнал за последние пять минут (TimeSpan=5). Таким образом, файл журнала будет значительно меньше, что значительно упростит диагностику.

Активный дамп памяти

Еще одна новая функция, связанная с диагностикой, — возможность записывать дамп памяти. Предположим, вы используете большой кластер Hyper-V, в котором у каждого узла имеется по 512 ГБ памяти. Если в работе этого узла возникнут неполадки и вы создадите дамп, содержащий память пользовательского режима и память режима ядра, размер этого дампа будет превышать 512 ГБ. Работать с файлом такого размера чрезвычайно сложно. Вначале необходимо будет постараться найти диск с достаточным запасом свободного места, чтобы туда уместился этот файл. Кроме того, потребуется потратить несколько часов на то, чтобы сжать этот файл, передать и распаковать, прежде чем можно будет его открыть. Если неполадка связана с хостом, на котором запущены виртуальные машины (использующие 500 ГБ памяти), этот огромный объем информации вам попросту не нужен, поскольку он не относится к хосту.

По этой причине в Windows Server 2016 появился новый параметр дампа — активный дамп памяти. В этом случае в дамп записывается только та память, которую хост фактически использует. Если используется только 5 ГБ, будет создан файл дампа размером 5 ГБ. За счет меньшего размера проанализировать этот файл будет значительно проще, чем файл размером 512 ГБ в описаном выше примере.

Параметр «Активный дамп памяти» находится и устанавливается там же, где и настройки обычного дампа: в разделе «Загрузка и восстановление» окна «Свойства системы», как показано на рис. 2-36.

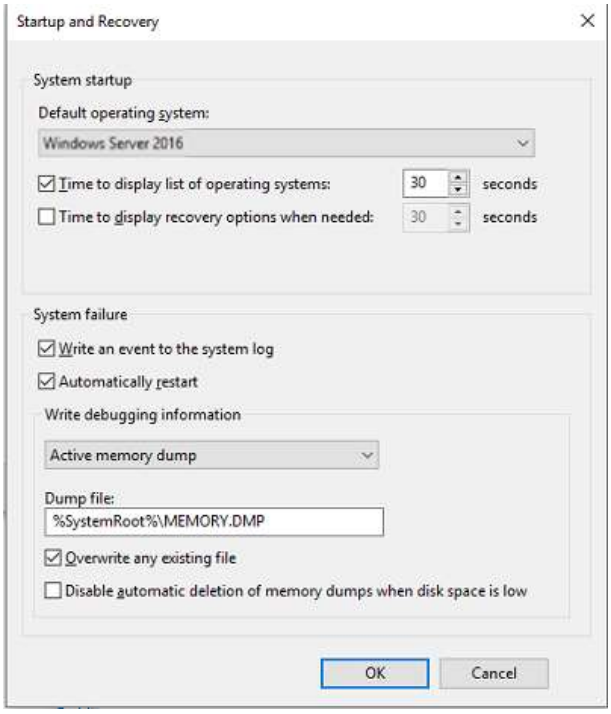


Рис. 2-36. Настройка активного дампа памяти

Примечание. Если говорить о дампах памяти в целом, стоит отметить, что отказоустойчивая кластеризация интегрирована с динамическими дампами, чтобы записывать дампы в случаях, когда истекло время ожидания. Эти дампы можно анализировать для выявления причин неполадок. Также реализована интеграция с отчетами об ошибках Windows (Windows Error Reporting, WER), чтобы можно было сохранять и добавлять другие журналы.

В зависимости от настроек времени ожидания ресурсов и служб кластера, динамический дамп может вызвать критическую ошибку («синий экран») и инициировать аварийный сбой сервера, чтобы создать дамп памяти. Хотя дамп памяти полезен для определения причин неполадок, запись его в ходе перезагрузки сервера значительно увеличивает время простоя. Благодаря интеграции динамического дампа процесс создания дампа работает в фоновом режиме, в то время как сам сервер работает в обычном режиме, не прерывая работы.

Цель этого компонента — сбор достаточного объема журналов и дампов, для того чтобы служба технической поддержки Microsoft могла успешно устранять различные неполадки, возникающие у пользователей при работе с кластерами. Объем информации должен быть достаточным для того, чтобы специалисты поддержки могли решить проблему пользователя и при этом не просили воспроизвести ее снова.

Диагностика сетевых имен

В Windows Server 2016 улучшены функции диагностики неполадок, касающихся сетевых имен. Иногда события, связанные с подобными неполадками, не попадают в журналы или содержат малопонятную информацию. Например, ранее при проблемах с обновлением DNS возникала стандартная ошибка, в сообщении о которой указывалось, что не удалось обновить DNS. При этом в описании ошибки не содержалось ни слова о том, по какой причине DNS-запись не смогла обновиться. Причины могут быть разными:

- служба DNS не принимает динамические обновления;
- используется защищенный DNS-сервер, а у кластера нет необходимых прав доступа;

- превышено время ожидания при подключении к DNS-серверу.

На устранение подобных ситуаций тратилось немало времени, поскольку требовалось постепенно сужать область поисков до тех пор, пока не удастся выявить проблему. Теперь в Windows Server 2016 события содержат более точную причину ошибок. Таким образом, если проблема возникает по одной из перечисленных выше причин, вы узнаете об этом из сообщения об ошибке и сможете сразу определить причину — это значительно ускорит устранение неполадок.

Кроме того, добавлены дополнительные проверки сетевых имен, чтобы предотвратить потенциальные проблемы, которые могут возникнуть не прямо сейчас, а через несколько дней или даже через несколько недель. При каждом включении/отключении ресурса и каждый час, пока сетевое имя подключено к сети, Windows Server 2016 проверяет следующее:

- доступен ли контроллер домена;
- синхронизирован ли пароль объекта имени кластера (CNO);
- включен ли CNO в службе каталогов Active Directory;
- существуют ли объект CNO и объект виртуального компьютера (VCO) в Active Directory.

В Windows Server 2016 также добавлено несколько дополнительных тестов при проверке кластерных настроек в части сетевых имен:

- имена CNO и VCO содержат более 15 символов;
- проверьте, что у объекта имени кластера есть разрешения на создание учетных записей компьютеров в подразделении (OU) Active Directory, в состав которого он входит;
- проверьте, что объект CNO и соответствующий объект VCO доступны для входа;
- удостоверьтесь, что в локальной группе «Users» на узлах кластера есть участники «CLISUR» и «NT AUTHORITY\Authenticated Users».

Данные проверки были добавлены, поскольку приведенные выше примеры неправильных настроек часто становятся причинами неполадок с сетевыми именами.

Последовательное обновление операционных систем в кластере

В Windows Server 2016 появился замечательный новый способ обновления операционной системы серверных кластеров, требующий значительно меньших простоев и трудозатрат, — *последовательное обновление операционных систем в кластере*. Такой способ обновления в Windows Server 2016 рекомендуется для кластеров Hyper-V и для кластеров масштабируемых файловых серверов (SOFS), но может использоваться так же и для других кластеров, к примеру SQL, с некоторой недоступностью, необходимой для завершения перехода.

До сих пор администраторам, чтобы обновить операционную систему в кластере, требовалось разрабатывать подробный план предстоящей процедуры. Зачастую администраторы ждали закупки нового оборудования для кластеров, приобретаемого в рамках обновления систем. Это означало, что кластеры могли работать без получения новых возможностей в течение нескольких лет, а для перемещения служб со старых кластеров на новые требовались плановые отключения.

Для последовательного обновления операционной системы кластера не требуется приобретать никакое дополнительное оборудование, все обновления выполняются на существующих узлах. Сам кластер при этом не нужно ни останавливать, ни перезапускать. Обновление происходит на уровне узлов кластера, причем в процессе последовательного обновления все службы продолжают работать. В отличие от типичных сценариев обновления кластеров, создавать новый кластер не требуется. Объекты существующего кластера, включая его имя и IP-адреса, остаются неизменными и во время обновления сохраняют работоспособность. Более того, этот процесс

полностью обратим до тех пор, пока не будет изменен атрибут «Функциональный уровень кластера» (подробнее об этом ниже).

На рис. 2-37 показан кластер Hyper-V с именем ContosoPVTCloud, состоящий из трех узлов.

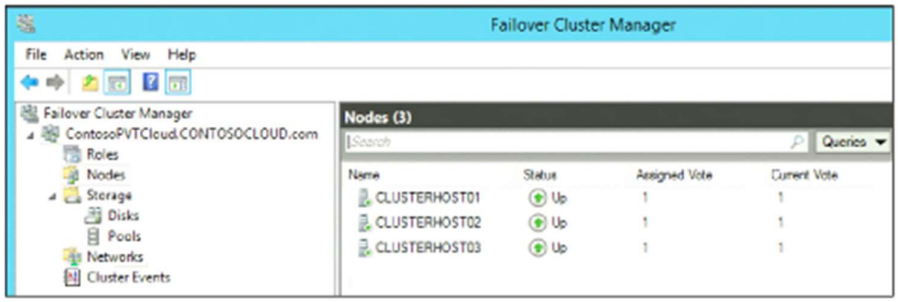


Рис. 2-37. Консоль диспетчера отказоустойчивости кластеров с кластером из трех узлов

На рис. 2-38 и 2-39 показаны общие тома кластера и работающие виртуальные машины.

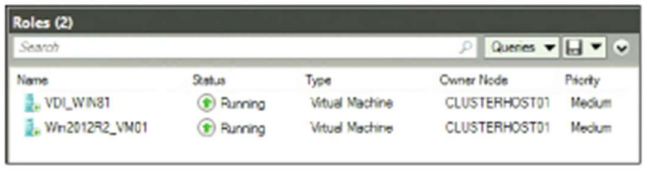


Рис. 2-38. Консоль диспетчера отказоустойчивости кластеров с двумя работающими виртуальными машинами

Рис. 2-39. Консоль диспетчера отказоустойчивости кластеров с тремя общими томами

Как можно заметить, кластер находится в полностью рабочем состоянии, узлы обновлены и все ресурсы выполняются без ошибок. Перед тем как начать обновление, рекомендуется так же выполнить резервное копирование кластера на случай, если потребуется восстановить его до первоначального состояния.

На следующем этапе можно приступить к последовательному обновлению операционной системы кластерных узлов. Убедитесь в том, что в используемой среде достаточно кластерных ресурсов, чтобы можно было обновлять узлы по одному, в то время как все кластерные роли будут обслуживаться оставшимися узлами кластера.

Чтобы приступить к последовательному обновлению, нужно исключить из кластера один из узлов. Приостановить и исключить узел можно в диспетчере отказоустойчивости кластеров или с помощью Windows PowerShell, используя командлет Suspend-ClusterNode, а затем Remote-ClusterNode. Чтобы начать последовательное обновление, можно выбрать любой из узлов кластера. В этом примере обновление начинается с узла ClusterHost01 кластера ContosoPVTCloud. Нажмите правую кнопку мыши на этом узле кластера и выберите команду «Приостановить» - «Очистить роли», как показано на рис. 2-40.

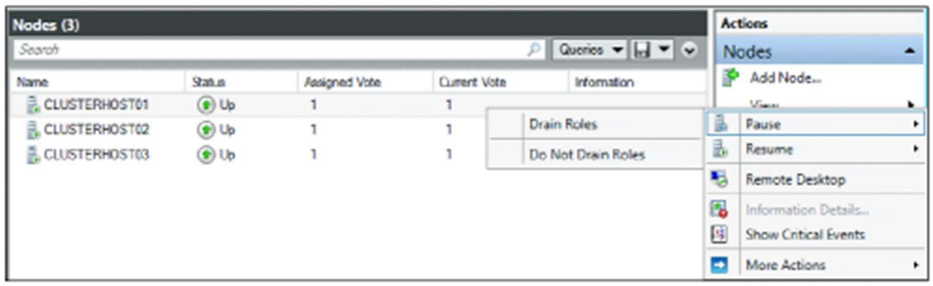


Рис. 2-40. Приостановка узла кластера

Затем нажмите правую кнопку мыши на узле кластера и выберите команду «Исключить», как показано на рис. 2-41.

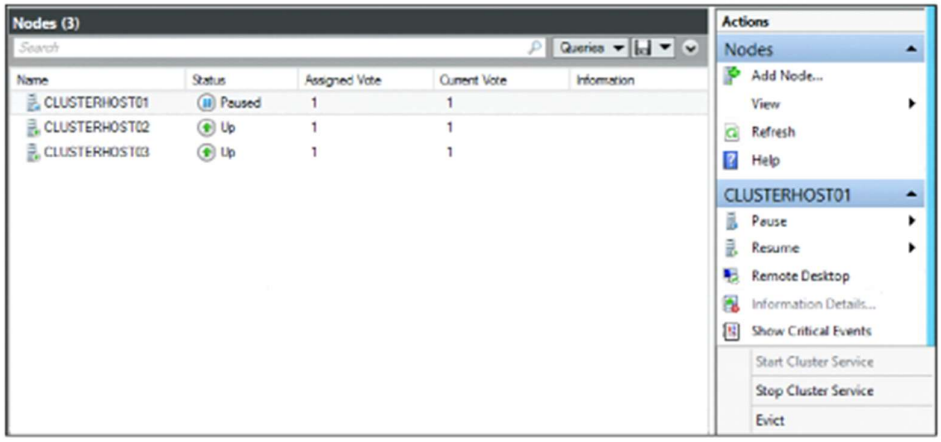


Рис. 2-41. Исключение узла кластера

Теперь можно начать установку на этот узел новой операционной системы.

> **Примечание.** Рекомендуется полная установка ОС с форматированием системных разделов. Обновление ОС на месте (in-place) крайне нежелательно, несмотря на то, что при сохранении стандартных драйверов работа узлов после обновления происходит в штатном режиме. Следует внимательно изучить все журналы событий для минимизации рисков при обновлении.

После установки Windows Server 2016 на узел ClusterHost01 добавьте роль Hyper-V, кластеризации (Failover Clustering), а также (при необходимости) многопутевой ввод-вывод (Multipath I/O). Настройте сеть и систему хранения данных так, как они были настроены до переустановки. На этом этапе желательно проверить, доступны ли какие-либо обновления для этой версии Windows Server 2016.

Присоедините узел ClusterHost01 к домену ContosoCloud, в котором находится обновляемый кластер.

Войдите на узел ClusterHost01 с правами администратора домена ContosoCloud или в качестве другого пользователя, обладающего правами доступа к существующему кластеру ContosoPVTCloud. Запустите диспетчер отказоустойчивости кластеров и нажмите «Добавить узел» или запустите в Windows PowerShell командлет Add-ClusterNode. Это необходимо сделать на сервере с Windows Server 2016, а не на узле кластера с Windows Server 2012 R2.

В консоли диспетчера отказоустойчивости кластеров на узле ClusterHost01 будет показано, что этот узел успешно присоединен к кластеру (рис. 2-42).

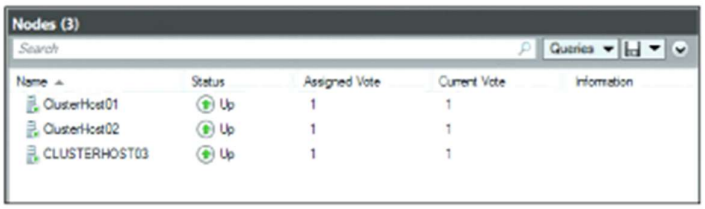


Рис. 2-42. Узлы кластера успешно присоединены

Обе роли виртуальных машин перемещены на узел ClusterHost01, как показано на рис. 2-43.

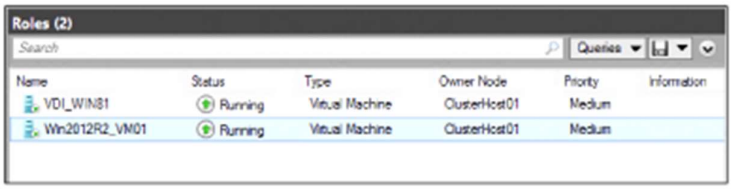


Рис. 2-43. Роли виртуальных машин перемещены на узел ClusterHost01

Один из общих томов кластера и диск-свидетель (disk witness) кворума также перемещены на узел ClusterHost01, как показано на рис. 2-44.

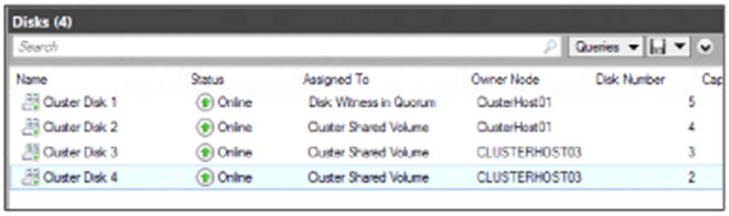


Рис. 2-44. Диски перемещены на узел ClusterHost01

Любые службы и роли можно перемещать между любыми узлами кластера. Это не однонаправленное перемещение в новый кластер. Все узлы кластера работают в обычном режиме. В ходе последовательного обновления, когда на узлах кластера используются разные наборы операционных систем, в кластере можно разместить любую роль или ресурс.

Пока кластер работает в смешанном режиме с использованием операционных систем Windows Server 2012 R2 и Windows Server 2016, можно устанавливать исправления и обслуживать все узлы обычным образом вплоть до завершения последовательного обновления. Также можно выполнять резервное копирование, кроме узла, обновляемого в данный момент.

Продолжите последовательное обновление операционной системы кластера, повторив описанные выше действия для узлов ClusterHost02 и ClusterHost03: приостанавливайте и исключайте узлы по одному, устанавливайте на исключенный узел систему Windows Server 2016, а затем снова присоединяйте узлы к домену и кластеру.

Эту же задачу можно выполнить с помощью Windows PowerShell, используя следующие командлеты:

1. Приостановить один из узлов и очистить его роли:

```
PS C:\> Suspend-ClusterNode –Drain –TargetNode 2012R2-NODE4
```

2. Исключить узел из кластера:

```
PS C:\> Remove-ClusterNode –Name 2012R2-NODE4
```

3. Выполнить чистую установку Windows Server на исключенный узел.

4. Добавить роль отказоустойчивого кластера:

```
PS C:\> Install-WindowsFeature –ComputerName 2012R2-NODE4 –Name Failover-Clustering –
IncludeManagementTools-IncludeAllSubFeature
```

5. Добавить узел с Windows Server в кластер Windows Server 2012 R2:

```
PS C:\> Add-ClusterNode –Cluster Cluster –Name Preview-Node5

PS C:\> Get-ClusterNode
Name              ID          State
2012R2-NODE3      1           Up
PREVIEW-NODE5     2           Up
```

6. Переустановить роли, компоненты и программное обеспечение, используемые в кластере (например, Hyper-V, SQL и т. п.).

7. Проверить перемещение ресурсов (failover).

8. Если все правильно работает, повторить шаги 2–7 для остальных узлов.

Когда на одних узлах используется Windows Server 2012 R2, а на других — Windows Server 2016, работает режим смешанных операционных систем. С этим режимом связан функциональный уровень кластера. И пока кластер работает в режиме смешанных операционных систем, следующая команда Windows PowerShell, запущенная на узле с ОС Windows Server 2016, возвратит значение 8 для узлов с Windows Server 2012 R2 и значение 9 для узлов с Windows Server 2016:

```
Get-ClusterNode | ft -auto NodeName, MajorVersion, MinorVersion, BuildNumber, NodeHighestVersion,
@{Expression={$_.NodeHighestVersion -shr 16}; Label="NHV.Cluster Functional Level";width=21},

@{Expression={$_.NodeHighestVersion –band 0xffff};Label=”NHV.Cluster Upgrade Version”;width=24},

NodeLowestVersion,

@{Expression={$_.NodeLowestVersion -shr 16}; Label="NLV.Cluster Functional Level";width=21},
@{Expression={$_.NodeLowestVersion -band 0xffff};Label="NLV.Cluster Upgrade Version";width=24}
```

На рис. 2-45 показаны выходные данные Windows PowerShell, когда осталось обновить только один узел.

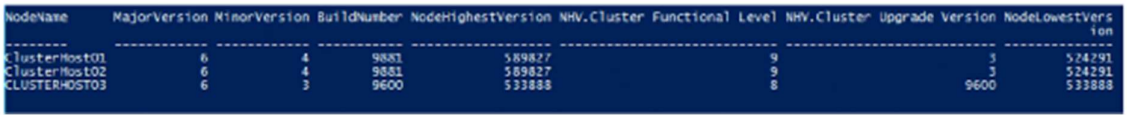


Рис. 2-45. Узлы, доступные для обновления

Выходные данные содержат значения NodeHighestVersion (высшая версия узла) и NodeLowestVersion (низшая версия узла). Когда эти значения будут совпадать для всех узлов, можно обновить функциональный уровень кластера, запустив командлет Update-ClusterFunctionalLevel, как показано на рис. 2-46.

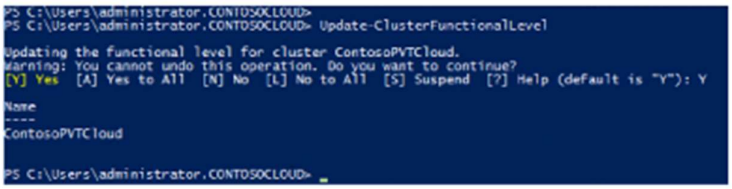


Рис. 2-46. Обновление функционального уровня кластера

После обновления всех узлов кластера до Windows Server 2016 снова запустите команду Windows PowerShell. На этот раз вы получите значение 9. Это говорит о том, что все узлы обновлены (рис. 2-47).

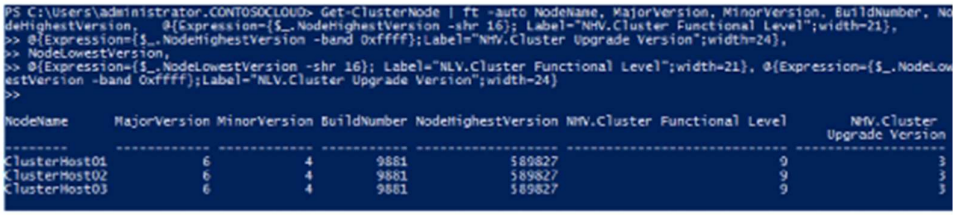


Рис. 2-47. Все узлы обновлены

Итак, процесс завершен: операционная система кластера обновлена на всех узлах и полностью работоспособна для поддержки кластерных ролей. В ходе этого процесса удалось избежать простоев и все службы оставались доступными. Поскольку в кластере есть виртуальные машины и файловые ресурсы, которые были настроены с помощью функций Windows Server 2012 R2, необходимо выполнить еще ряд действий, чтобы получить полную функциональность Windows Server 2016.

Если посмотреть на виртуальные машины в диспетчере Hyper-V, то станет видно, что их номер версии — 5.0 (рис. 2-48).

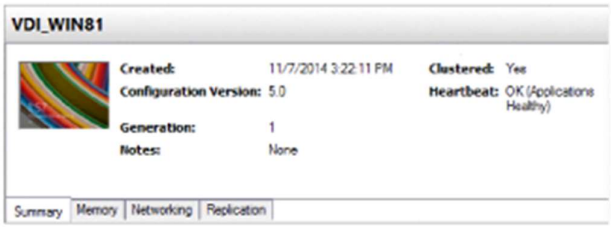


Рис. 2-48. Номер версии виртуальной машины

Это означает, что конфигурация операционной машины соответствует операционной системе Windows Server 2012 R2. Чтобы при следующем запуске обслуживания обновить версию конфигурации Hyper-V каждой виртуальной машины, можно запустить командлет Get-VM, а затем — командлет Update-VMVersion, как показано на рис. 2-49.

```
PS C:\Windows\system32> Update-VMVersion

cmdlet Update-VMVersion at command pipeline position 1
Supply values for the following parameters:
Name[0]: 1C01
Name[1]:

Confirm
Are you sure you want to perform this action?
Performing a configuration version update of "1C01" will prevent it from being migrated to or imported on previous
versions of Windows. This operation is not reversible.

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
```

Рис. 2-49. Обновление конфигурации виртуальной машины

При этом версия конфигурации виртуальных машин будет изменена до 8.0 и все новые возможности, доступные для Hyper-V в Windows Server 2016, будут включены, как показано на вкладке «Сводка» (рис. 2-50).

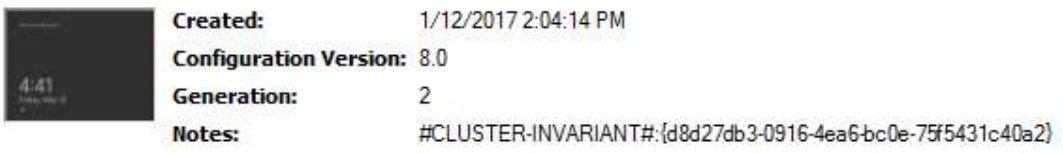


Рис. 2-50. Версия конфигурации виртуальной машины

Для серверов SOFS, на которых выполнен процесс последовательного обновления операционной системы кластера, также потребуется обновить функциональность пулов носителей. Их функциональность будет ограничена уровнем Windows Server 2012 R2 до тех пор, пока не будет выполнен командлет Update-StoragePool. Укажите имя пула хранилища, который нужно обновить.

После того как будет завершено последовательное обновление операционной системы кластера на всех узлах, а также обновление функционального уровня кластера, версии конфигурации виртуальных машин и пула хранилища до уровня Windows Server 2016, все работы по обновлению кластера будут закончены.

Как видим, необходимость создавать новый кластер, а также заново создавать или перемещать рабочие нагрузки осталась в прошлом. Последовательное обновление операционной системы

кластера упрощает управление кластером в долгосрочной перспективе и устраняет имевшиеся ранее трудности.

Кластеры в рабочей группе и мультидоменной среде

В Windows Server 2012 R2 и более ранних версиях можно было создавать кластеры только из узлов, присоединенных к одному и тому же домену. В Windows Server 2016 это ограничение снято — добавлена возможность создавать отказоустойчивые кластеры без какой-либо зависимости от Active Directory.

Теперь можно создавать отказоустойчивые кластеры со следующими конфигурациями:

- **однодоменные кластеры** — кластеры, все узлы которых присоединены к одному и тому же домену;
- **мультидоменные кластеры** — кластеры, узлы которых присоединены к разным доменам;
- **кластеры в среде рабочих групп** — кластеры, узлы которых являются рядовыми серверами рабочей группы (не присоединены к домену).

Примечание. Более подробная информация о кластерах в рабочих группах и многодоменных кластерах доступна по адресу https://blogs.msdn.microsoft.com/clustering/2015/08/17/workgroup-and-multi-domain-clusters-in-windows-server-2016/.

SMB Multichannel и кластерные сети с использованием нескольких сетевых адаптеров

Сети отказоустойчивых кластеров больше не связаны ограничением в один сетевой адаптер на подсеть или сеть. Благодаря упрощению многоканального протокола SMB Multichannel и поддержке кластерных сетей с несколькими сетевыми адаптерами, настройка сетей производится автоматически, а каждый сетевой адаптер в подсети можно использовать для трафика кластера и рабочих нагрузок. Это усовершенствование позволяет повысить пропускную способность сети для Hyper-V, экземпляров отказоустойчивого кластера SQL Server и других рабочих нагрузок SMB.

Примечание. Более подробная информация об SMB Multichannel и кластерных сетях с несколькими сетевыми адаптерами доступна по адресу https://technetmicrosoft.com/windows-server-docs/compute/failover-clustering/simplified-smb-multichannel-and-multi-nic-cluster-networks.

Усовершенствования виртуальных машин

Работа виртуальных машин в составе отказоустойчивых кластеров была усовершенствована в трех областях.

- Равнодоступность узлов виртуальных машин (VM fairness)

  Равнодоступность узлов виртуальных машин упрощает оптимальную балансировку нагрузки между узлами кластера. Перенагруженные узлы определяются по показателям использования памяти и загрузки процессоров на узле. Затем виртуальные машины переносятся (динамически, Live migraton) с перенагруженного узла на узлы с более доступными ресурсами. Есть возможность настроить агрессивность балансировки, чтобы добиться оптимальной производительности и загруженности кластера. Равнодоступность узлов отключается, если включена динамическая оптимизация SCVMM. В противном случае она включена по умолчанию.

- Порядок запуска виртуальных машин

  От порядка запуска виртуальных машин зависит работа виртуальных машин в составе кластера. Их теперь можно группировать в уровни (tiers) и определять зависимости порядка запуска между

разными уровнями. Благодаря этому самые важные виртуальные машины запускаются в первую очередь. Виртуальные машины зависимого уровня не запускаются до тех пор, пока не будут запущены виртуальные машины того уровня, от которого зависят эти виртуальные машины.

- Устойчивость виртуальных машин

Примечание. Более подробная информация доступна по адресу https://blogs.msdn.microsoft.com/clustering/2015/06/03/virtual-machine-compute-resiliency-in-windows-server-2016/.

Хранение данных

Системы хранения данных на современных предприятиях, как правило, представляют собой традиционные дисковые массивы Fiber Channel или iSCSI с большим количеством накопителей. Усовершенствованные, а также новые технологии хранения данных, реализованные в Windows Server 2016, предоставляют организациям абсолютно новые возможности программно-определяемых хранилищ.

Реплика хранилища

Реплика хранилища — новая функция Windows Server 2016, с помощью которой можно настроить независимую от конкретной системы хранения данных синхронную или асинхронную репликацию на уровне блоков между кластерами или серверами для аварийного восстановления. Эту функцию также можно использовать для расширения отказоустойчивого кластера на несколько сайтов с целью обеспечения высокой доступности. Синхронная репликация дает возможность создавать зеркальные копии данных на физических сайтах с устойчивыми к сбоям томами, гарантируя нулевые потери данных на уровне файловой системы. Асинхронная репликация дает возможность расширять сайты за пределы границ одного города, правда, при этом есть некоторая вероятность потери части данных. Реплика хранилища работает не на файловом уровне, как репликация распределенной файловой системы (DFS), а на уровне блоков данных, как репликация блоков, поэтому работоспособность сохраняется вне зависимости от файловых блокировок, открытых дескрипторов и т. п.

Примечание. Реплика хранилища не является развитием существовавших ранее технологий, это новая возможность Windows Server 2016 и программно-определяемого центра обработки данных. Более подробная информация доступна по адресу http://aka.ms/sroverview.

Синхронная репликация

При синхронной репликации гарантируется одновременная запись данных по крайней мере в двух местах, прежде чем завершится операция записи. Такой тип репликации лучше всего подходит для важных данных, поскольку требует инвестиций в сети и системы хранения данных. Кроме того, при использовании синхронной репликации возникает угроза снижения производительности приложений. Синхронная репликация пригодна для решений высокой доступности (HA) и аварийного восстановления (DR).

Как показано на рис. 2-51, когда приложение записывает исходную копию данных (1), первоначальное хранилище не сразу подтверждает операцию ввода-вывода. Вместо этого изменения данных реплицируются в удаленную копию назначения (2), записываются данные журнала (3), затем удаленный сайт возвращает подтверждение (4), и только после этого приложение получает подтверждение ввода-вывода (5). За счет этого обеспечивается постоянная синхронизация удаленного сайта с исходным сайтом; операции ввода-вывода хранилища данных распространяются по сети. В случае отказа исходного сайта приложение может отработать отказ, переключившись на удаленный сайт, и возобновить работу. При этом гарантируется отсутствие потерь данных.

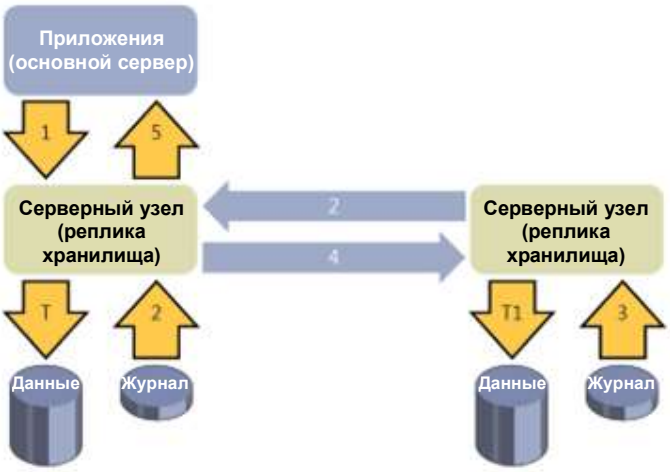


Рис. 2-51. Синхронная репликация при использовании реплики хранилища

> **Примечание.** На рис. 2-51 «T» означает данные, записываемые на томе, находящемся на исходном сайте, а «T1» — данные, записываемые на томе, который находится на удаленном сайте. Во всех случаях используется сквозная запись журнала (write-through)

Асинхронная репликация

В отличие от синхронной репликации, при асинхронной репликации данные, записываемые приложением, передаются на удаленный сайт без обязательного немедленного подтверждения (рис. 2-52). В этом режиме снижается время отклика для приложения и ускоряется работа географически распределенного решения для аварийного восстановления. Поскольку значение целевой точки восстановления (RPO) больше нуля, асинхронная репликация менее пригодна для решений высокой доступности, таких как отказоустойчивые кластеры, поскольку такие решения предназначены для непрерывной работы с дублированием и без потери данных.

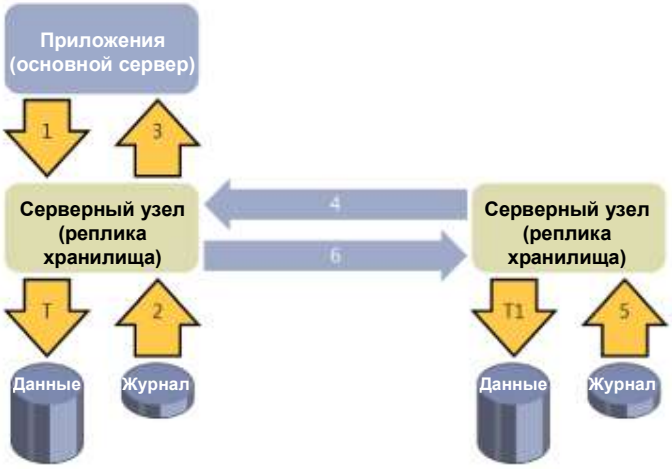


Рис. 2-52. Аинхронная репликация при использовании реплики хранилища

Как показано на рис. 2-52, когда приложение записывает данные (1), подсистема репликации регистрирует запись в журнале (2) и сразу же выдает подтверждение приложению (3). После этого данные реплицируются в удаленное хранилище (4), удаленный узел обрабатывает копию данных, делает запись в журнале (5), а затем (возможно, не сразу) отправляет подтверждение в исходную копию (6). Производительность репликации больше не влияет на работу конвейера ввода-вывода приложения. Скорость реагирования удаленного сайта и расстояние до него не столь важны. Правда, существует риск потери данных, если исходные данные будут утрачены,

пока копия данных, предназначенная для удаленной записи, будет находиться в буфере и не покинет исходное расположение.

Особенности реализации

Реплика хранилища использует SMB 3.1.1. в качестве надежного и высокоскоростного транспортного протокола репликации. При этом доступны все преимущества SMB 3.0, в том числе многоканальные подключения, удаленный доступ к памяти (RDMA), шифрование, подписи и безопасность на основе Kerberos. Для использования реплики хранилища не требуется вносить изменения в доменные службы Active Directory и в какие-либо разрешения администрирования. В табл. 2-4 перечислены особенности реализации реплики хранилища.

Таблица 2-4. Реализация реплики хранилища

Возможность	Поддержка
Тип	На основе узлов
Синхронная	Да
Асинхронная	Да
Не зависит от оборудования хранилища	Да
Единица репликации	Том (раздел)
Создание распределенного кластера Windows Server	Да
Репликация с сервера на сервер	Да
Транспортный протокол	SMB 3.1.1
Сеть	TCP/IP или RDMA
RDMA	iWARP, InfiniBand
Требования к портам брандмауэра в сети репликации	Один порт IANA (TCP 445 или 5445)
Многоканальная передача данных	Да (SMB3)
Поддержка Kerberos	Да (SMB3)
Шифрование и подпись данных при передаче	Да (SMB3)
Допускается отработка отказов для каждого тома	Да
Пользовательский интерфейс управления	Windows PowerShell, диспетчер отказоустойчивых кластеров

Примечание. Реплика хранилища в Windows Server 2016 не поддерживает так называемую транзитивную репликацию, которая также называется топологией A-B-C (синхронная репликация с сервера A на сервер B, затем асинхронная репликация с сервера B на сервер C). Реплика хранилища не поддерживает схему репликации «один отправитель — несколько получателей». Есть возможность (только для рабочих нагрузок в виртуальной среде) использовать реплику Hyper-V в качестве дополнительного механизма асинхронной репликации. В этом случае подразумеваются использование реплики Hyper-V на исходном томе A и репликация на сервер, отличный от B, по топологии «с сервера A на сервер B + с сервера A на сервер C».

Требования

Для использования реплики хранилища существуют следующие требования:

- Windows Server 2016 Datacenter.
- Не менее 2 ГБ оперативной памяти на каждом физическом сервере и по крайней мере два процессорных ядра.
- Доменные службы Active Directory — AD DS (для использования Kerberos в SMB). Обновления схем, объектов AD DS, функциональных уровней AD DS и т. п. не требуются.
- Сеть:
  - пропускная способность сети между серверами не менее 1 Гбит/с;
  - открытые порты брандмауэра: SMB, WS-MAN.
- Хранение данных:

    - один том с файловой системой NTFS/ReFS, выделенный для репликации, на каждый сервер/узел кластера (не менее 8 ГБ свободного пространства);
    - таблица разделов GPT (не MBR);
- JBOD, iSCSI, DAS, SCSI или SATA, Storage Spaces Direct (между кластерами), общий VHDX или сеть хранения данных (SAN);
    - одинаковый размер секторов для томов с данными и томов с журналами;
    - на реплицируемых томах и томах журналов не должно быть папки %SystemRoot% и файлов подкачки.

Рекомендации

Для использования реплики хранилища действуют следующие рекомендации:

- Сеть:
    - пропускная способность: не менее 10 Гбит/с между серверами;
    - задержки: в среднем не более 5 мс на путь туда и обратно для синхронной репликации.
- Хранение данных:
    - твердотельные накопители (SSD) для томов журнала;
    - не менее 8 ГБ свободного места.

Примечание. Чтобы убедиться в соответствии требованиям и получить подсказки по настройке рекомендуемой конфигурации для файлов журналов, можно воспользоваться командлетом Test-SRTopology в Windows PowerShell.

Сценарии

Реплика хранилища разработана для двух следующих сценариев:

- расширение отказоустойчивого кластера для высокой доступности;
- репликация между серверами для аварийного восстановления.

Репликация при расширении кластера

Растянутый кластер (его также называют *мультисайтовым кластером*) использует реплику хранилища для подключения двух наборов асимметричных общих хранилищ в пределах одного отказоустойчивого кластера. В качестве хранилища можно использовать набор накопителей SCSI JBOD (это сокращение означает «просто несколько накопителей» — just a bunch of drives), iSCSI или сеть хранения данных (SAN). Узлы кластера подключаются к каждому из двух наборов хранилища, которые (по возможности) должны находиться в разных местах — например, в разных зданиях в пределах одного комплекса или же в разных центрах обработки данных, географически удаленных один от другого. Реплицируемое хранилище может представлять собой общие тома кластера (CSV) или назначенные какой-либо роли ресурсы физического диска (PDR).

На рис. 2-53 показана типовая архитектура репликации при внедрении растянутого кластера с репликой хранилища. С левой стороны — сайт в Редмонде, содержащий два сервера (SR-SRV-01 и SR-SRV-02) и общее хранилище (SAN, JBOD или iSCSI). С правой стороны — сайт в Беллвью, содержащий еще два сервера (SR-SRV-03 и SR-SRV-04) и другое хранилище. С помощью реплики хранилища можно объединить серверы и общие хранилища на этих двух сайтах в один растянутый кластер, используя асимметричную репликацию хранилища с одного сайта на другой.

Рис. 2-53. Типовая архитектура репликации при растягивании кластера

В такой конфигурации кластер становится устойчивым к отказам не только отдельных узлов, но и целых сайтов. В случае отказа одного из узлов сайта другой узел этого же сайта становится новым источником репликации. При отказе всех узлов сайта новым источником репликации становится узел другого сайта. Все это происходит автоматически, как на обычном нерастянутом кластере. Для использования растянутого кластера требуется не менее двух узлов. Кластер может содержать до 64 узлов.

В Windows Server 2016 рекомендуется использовать репликацию для двух кластерных ролей: Hyper-V и файлового сервера общего назначения. Не следует настраивать в качестве растянутых кластеров масштабируемые файловые серверы (SOFS), поскольку отказоустойчивые кластеры Windows Server не имеют встроенной поддержки сайтов. Приложения будут подключаться к узлам на обоих сайтах, а затем перенаправляться на главный узел, где происходит запись ввода-вывода. Это может привести к снижению производительности приложений. Гостевая кластеризация виртуальных машин в данном сценарии поддерживается только в ознакомительных целях.

Для управления таким кластером можно использовать диспетчер отказоустойчивости кластеров (cluadmin.msc) с простым интерфейсом на основе мастера. Для создания растянутого кластера достаточно создать общий том кластера (CSV) и настроить роль «Файловый сервер общего назначения» или «Виртуальная машина Hyper-V». Нажмите правую кнопку мыши на исходном хранилище (на рис. 2-54 это «Cluster Disk 3» в кластере np-sr-cluster.com), выберите команду «Репликация», а затем — «Включить».

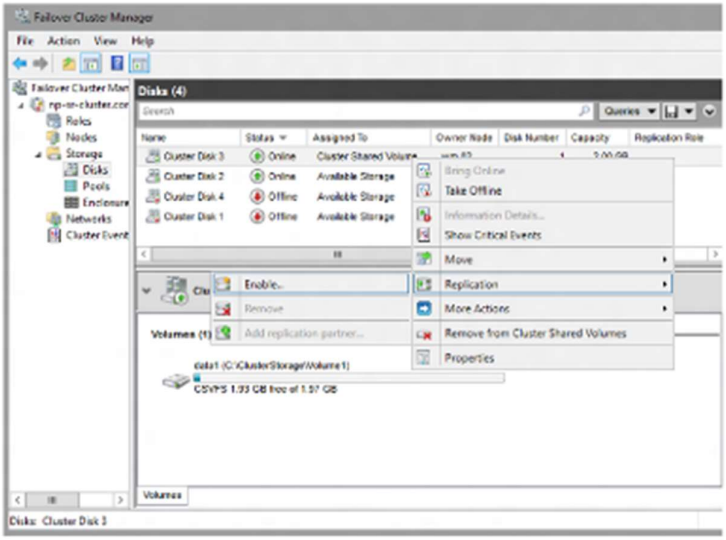


Рис. 2-54. Включение репликации диска в диспетчере отказоустойчивости кластеров

Будет запущен мастер настройки реплики хранилища. В этом мастере на странице «Выбор конечного диска с данными» выберите в списке доступных накопителей диск назначения для репликации исходного диска. На рис. 2-55 в качестве конечного диска с данными выбран «Cluster Disk 1».

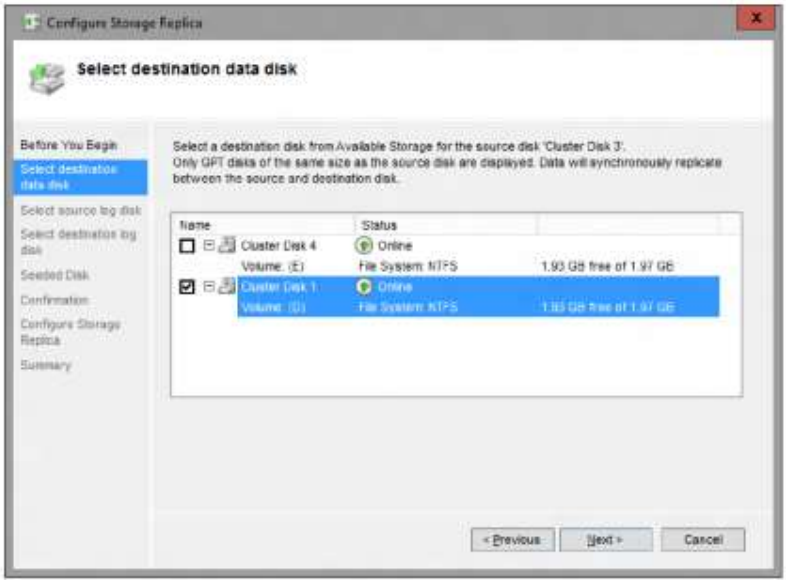


Рис. 2-55. Выбор диска назначения для реплики

Чтобы завершить настройку растянутого кластера, выполните дальнейшие инструкции мастера. После завершения конфигурирования хранилище будет синхронно реплицироваться между исходным хранилищем и хранилищем назначения в кластере. После завершения репликация образует растянутый кластер, а реплика хранилища защищает данные на исходном и конечном накопителях. Например, на рис. 2-56 показано направление репликации после отработки отказа. В этом случае накопитель Cluster Disk 1 является источником репликации, а накопитель Cluster Disk 3 — пунктом назначения.

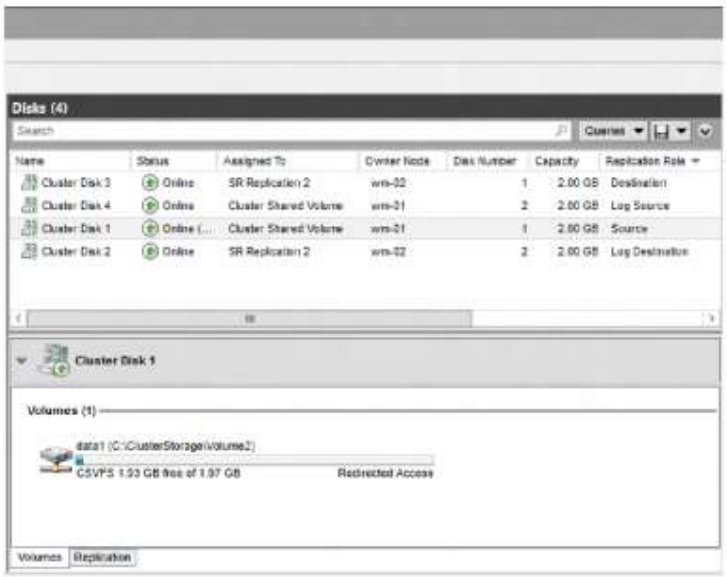


Рис. 2-56. Информация о дисках после создания реплики

Примечание. Для создания растянутого кластера также можно использовать модули Failover-Clustering и StorageReplica в Windows PowerShell.

Репликация между серверами и между кластерами

Реплика хранилища может соединять два отдельных сервера (такая топология иногда называется автономной репликацией) и их тома. В качестве хранилища можно использовать набор накопителей JBOD SCSI, iSCSI, SAN или даже локальное непосредственно подключенное хранилище — например, диски SCSI, подключенные к локальному контроллеру RAID. Кроме того, реплика хранилища поддерживает репликацию между двумя кластерами таким же образом, как если бы это были серверы, с использованием любых поддерживаемых кластерами общих хранилищ. Репликация происходит между двумя физическими местоположениями — например, разными зданиями в пределах одного комплекса или разными центрами данных. Реплицируемые хранилища должны представлять собой тома NTFS или ReFS.

На рис. 2-57 показана типовая архитектура репликации между серверами при использовании реплики хранилища. С левой стороны — сайт в Редмонде, содержащий сервер SR-SRV-01 и хранилище (SAN, JBOD или iSCSI). С правой стороны — сайт в Беллвью, содержащий сервер SR-SRV-02 и другое хранилище. С помощью реплики хранилища можно объединить серверы и хранилища на этих двух сайтах в партнерство, используя асимметричную репликацию хранилища с одного сайта на другой.

Рис. 2-57. Типовая архитектура репликации между серверами

Инструментов с графическим интерфейсом или средств автоматического управления отработкой отказов для работы со сценариями репликации между серверами или между кластерами нет, администрирование осуществляется только вручную с помощью модуля StorageReplica в Windows PowerShell. Как правило, вся настройка Storage Replica в таких сценариях может сводиться к одной лишь команде в PowerShell

Модуль Windows PowerShell StorageReplica содержит следующие команды в Windows Server 2016:

```
PS C:\Windows\system32> Get-Command -Module StorageReplica | FT –A

CommandType Name                       Version Source
----------- ----                       ------- ------
Function    Clear-SRMetadata           1.0     StorageReplica
Function    Export-SRConfiguration     1.0     StorageReplica
Function    Get-SRAccess               1.0     StorageReplica
Function    Get-SRDelegation           1.0     StorageReplica
Function    Get-SRGroup                1.0     StorageReplica
Function    Get-SRNetworkConstraint    1.0     StorageReplica
Function    Get-SRPartnership          1.0     StorageReplica
Function    Grant-SRAccess             1.0     StorageReplica
Function    Grant-SRDelegation         1.0     StorageReplica
Function    New-SRGroup                1.0     StorageReplica
Function    New-SRPartnership          1.0     StorageReplica
Function    Remove-SRGroup             1.0     StorageReplica
```

```
Function    Remove-SRNetworkConstraint 1.0    StorageReplica
Function    Remove-SRPartnership       1.0    StorageReplica
Function    Revoke-SRAccess            1.0    StorageReplica
Function    Revoke-SRDelegation        1.0    StorageReplica
Function    Set-SRGroup                1.0    StorageReplica
Function    Set-SRNetworkConstraint    1.0    StorageReplica
Function    Set-SRPartnership          1.0    StorageReplica
Function    Suspend-SRGroup            1.0    StorageReplica
Function    Sync-SRGroup               1.0    StorageReplica
Cmdlet      Test-SRTopology            1.0    StorageReplica
```

Для настройки репликации достаточно указать следующую информацию:

```
New-SRPartnership -SourceComputerName np-sr-srv05 -SourceRGName rg01 -SourceVolumeName
g: -SourceLogVolumeName
h: -DestinationComputerName np-sr-srv06 -DestinationRGName rg02 –DestinationVolumeName
g: -DestinationLogVolumeName
h: -LogSizeInBytes 16GB
```

Командлет New-SRPartnership поддерживает множество параметров, в том числе создание асинхронной репликации. Также можно создавать репликацию с более точными параметрами: сначала нужно запустить New-SRGroup на каждом сервере, а затем, чтобы объединить их, использовать New-SRPartnership. Добавить дополнительные тома в группу репликации можно с помощью командлета Set-SRGroup. На одном сервере могут работать одновременно несколько групп репликации.

Реплика хранилища в Windows Server 2016

Ниже приводятся некоторые важные особенности работы реплики хранилища в Windows Server 2016.

- **Снижается пропускная способность сети и возникают задержки даже при использовании быстрых хранилищ.** У синхронной репликации есть физические ограничения. Реплика хранилища применяет механизм фильтрации ввода-вывода с использованием журналов, для нее требуется передача данных по сети туда и обратно, поэтому при синхронной репликации запись данных в приложении будет замедлена. Можно свести к минимуму снижение производительности, используя сети с низкими задержками и высокой пропускной способностью, а также подсистемы накопителей с высокой пропускной способностью для хранения журналов.
- **Конечный том недоступен при репликации.** При настройке репликации конечный том будет отключен. Для обычных программ с графическим пользовательским интерфейсом и для операций записи пользователей он будет недоступен до тех пор, пока репликация не будет отменена или пока этот том не станет источником при отработке отказа.

  Технологии репликации на уровне блоков несовместимы с предоставлением доступа к файловой системе целевого тома. Файловые системы NTFS и ReFS не поддерживают запись пользовательскими приложениями данных на томе одновременно с изменением блоков в томе.
- **Реализации асинхронной репликации различаются.** Реализация асинхронной репликации корпорации Microsoft отличается от большинства отраслевых реализаций, в которых используются моментальные снимки (snapshots) и периодическая передача разностных данных на другие узлы с последующим слиянием. Асинхронная репликация в реплике хранилища, напротив, работает точно так же, как синхронная, за исключением того, что в асинхронной репликации не требуется последовательное синхронное подтверждение от хранилища назначения. Это означает, что теоретически у реплики хранилища будет более низкое значение RPO, поскольку репликация производится постоянно. Впрочем, из этого также следует, что асинхронная репликация Microsoft опирается на гарантию внутренней однородности приложения, а не на использование моментальных снимков для

принудительного достижения однородности в файлах приложений. Реплика хранилища гарантирует устойчивость к отказам во всех режимах репликации.

- **Реплика хранилища не является репликацией распределенной файловой системы (DFSR).** Репликация хранилищ на уровне блоков малопригодна для использования в офисах филиалов. Для сетей офисов филиалов характерны значительные задержки, высокая степень использования и малая пропускная способность, из-за чего использование синхронной репликации становится нецелесообразным. В офисах филиалов часто используется репликация из одного источника в несколько мест назначения, которые доступны только для чтения (например, для распространения программного обеспечения), а первый выпуск реплики хранилища не поддерживает такую топологию. При репликации данных из офиса филиала в главный офис реплика хранилища отключает том назначения, чтобы предотвратить прямой доступ к нему.

  Тем не менее важно отметить, что многие заказчики все же используют репликацию распределенной файловой системы (DSFR) в качестве решения для аварийного восстановления, хотя для этого сценария такой тип репликации не слишком пригоден: DSFR не может реплицировать открытые файлы, этот тип репликации оптимизирован для снижения нагрузки на сеть в ущерб производительности, что приводит к значительной разнице между точками восстановления. Использование реплики хранилища дает возможность отказаться от использования DSFR для некоторых сценариев аварийного восстановления.

- **Реплика хранилища не является резервным копированием.** В некоторых ИТ-средах системы репликации развертываются в качестве решений для резервного копирования, поскольку при использовании репликации (в отличие от ежедневного создания резервных копий) гарантируется отсутствие потерь данных. Реплика хранилища сохраняет все изменения всех блоков данных тома вне зависимости от типа изменений. Например, если пользователь удалит из тома все данные, реплика хранилища мгновенно скопирует удаление на другой том, вследствие чего данные будут удалены с обоих серверов без возможности восстановления.

- **Реплика хранилища не является репликой Hyper-V или SQL AlwaysOn.** Реплика хранилища — это подсистема общего назначения, не зависящая от используемого оборудования. По определению ее поведение невозможно настроить настолько же безупречно, как при репликации на уровне приложений. Из-за этого может обнаружиться нехватка некоторых возможностей, для получения которых потребуется сохранить или развернуть технологии репликации уровня приложений.

Локальные дисковые пространства

С помощью локальных дисковых пространств (storage spaces direct) поставщики услуг и крупные организации могут создавать высокодоступные и масштабируемые программно-определяемые хранилища, используя стандартные серверы с внутренними накопителями. Использование серверов с внутренними накопителями дает возможность снизить сложность, повысить масштабируемость и использовать устройства хранения, которые ранее не поддерживались, — например, твердотельные накопители с интерфейсом SATA (в целях снижения стоимости флэш-памяти), или твердотельные накопители NVMe для наибольшей производительности.

Локальные дисковые пространства устраняют необходимость использования инфраструктуры общих дисков SAS, упрощают развертывание и настройку. В качестве структуры хранения данных используется сеть. В качестве высокоскоростного протокола доступа к хранилищам с низкими задержками и с низкой процессорной нагрузкой используются SMB3 и SMB Direct (RDMA). Для масштабирования достаточно просто увеличить число серверов: будут увеличены и емкость

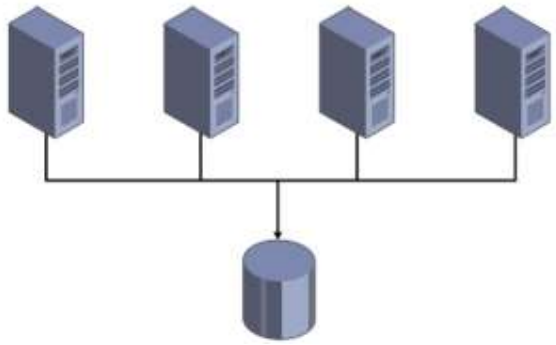
хранилища данных, и производительность ввода-вывода. Ниже описываются некоторые возможности и особенности локальных дисковых пространств.

- **Хранилище для Hyper-V и Microsoft Azure Stack.** Локальные дисковые пространства используются главным образом в качестве хранилищ для виртуальных машин Hyper-V или Azure Stack.
- **Оборудование.** Локальные дисковые пространства дают возможность создавать высокодоступные и масштабируемые хранилища, используя современное оборудование хранения данных, в том числе твердотельные накопители (SSD) с интерфейсом SATA для снижения затрат и накопители SSD с интерфейсом NVMe для повышения производительности и снижения процессорной нагрузки. Также можно использовать сетевую инфраструктуру с поддержкой RDMA, чтобы обращаться к хранилищам с меньшими задержками, чем при использовании сетей Ethernet, и с пониженной процессорной нагрузкой. Снижение процессорной нагрузки дает возможность повысить плотность рабочих нагрузок.
- **Проверенные конфигурации.** Корпорация Microsoft сотрудничает с партнерами, занимающимися производством оборудования, с целью совместной подготовки и проверки конфигураций серверов, рекомендуемых для локальных дисковых пространств. При использовании данных аппаратных конфигураций достигаются наибольшая эффективность локальных дисковых пространств и наивысшая производительность, а также обеспечивается доступ ко всем возможностям.
- **Конфигурации хранилищ.** Локальные дисковые пространства можно использовать с различными конфигурациями хранилищ данных. Вот наиболее распространенные конфигурации:
  - твердотельные накопители (SSD) и традиционные жесткие диски, при этом SSD используются в качестве кеша чтения и записи для повышения производительности ввода-вывода;
  - конфигурация с использованием только твердотельных накопителей на основе флэш-памяти: SSD с интерфейсом NVMe и SATA обеспечивают исключительно высокую производительность ввода-вывода;
  - трехуровневое физическое хранилище: твердотельные накопители NVMe, твердотельные накопители SATA и традиционные жесткие диски.
- **Тип развертывания.** При использовании локальных дисковых пространств заказчики могут выбрать нужный тип развертывания: конвергентную либо гиперконвергентную инфраструктуру. В гиперконвергентной инфраструктуре вычислительные ресурсы и хранилища данных предоставляются одними и теми же серверами, благодаря чему упрощаются масштабирование и управление. В конвергентной инфраструктуре вычислительные ресурсы отделены от хранилищ данных, что дает возможность гибко масштабировать вычислительные ресурсы и хранилища независимо друг от друга.
- **Отказоустойчивость.** Локальные дисковые пространства устойчивы к отказам накопителей. При отказе накопителя поврежденные данные автоматически восстанавливаются на оставшихся накопителях. Локальные дисковые пространства поддерживают три типа доменов сбоя: а) отдельный сервер, б) корпус (шасси), в) стойка. Размещение данных, восстановление данных и повторная балансировка производятся в соответствии с доменом сбоя.
- **Ускоренная работа «кода избыточности» (erase coding).** В локальных дисковых пространствах появились гибридные тома, дополняющие существующие типы томов (зеркальные и с кодом избыточности). В гибридных томах преимущества зеркальных томов (высокая производительность) и томов с избыточным кодированием (эффективность)

объединяются в одном томе с автоматическим выбором уровня хранилища в реальном времени.

- **Эффективные контрольные точки (checkpoints) виртуальной машины.** Локальные дисковые пространства используют новую файловую систему ReFS версии 2, которая (при использовании вместе с Hyper-V) позволяет очень быстро и эффективно создавать контрольные точки виртуальных машин практически с минимальной добавочной нагрузкой на подсистему ввода-вывода хранилища.
- **Масштабируемость.** Локальные дисковые пространства можно масштабировать в пределах от 2 до 16 серверов. Добавлять серверы можно по мере необходимости, при этом данные будут перераспределены таким образом, чтобы как можно эффективнее задействовать дополнительные ресурсы. На конференции IDF 2015 корпорации Microsoft и Intel продемонстрировали хранилище Storage Spaces Direct из 16 серверов, использующее только твердотельные накопители NVMe.
- **Служба проверки состояния.** Локальные дисковые пространства включают встроенную подсистему диагностики, с помощью которой администраторы (даже с ограниченной квалификацией) могут отслеживать состояние системы в повседневной деятельности. К функциям службы проверки относятся:
  - отслеживание работы кластера, оборудования хранения данных и технологий программно-определяемого хранилища, выявление неполадок и выдача оповещений, содержащих понятные инструкции по устранению неполадок;
  - объединение сведений о производительности и емкости в целостную картину, которая дает общее представление о состоянии доступных ресурсов;
  - автоматизация часто встречающихся задач (например, замена накопителей и обновление их микрокода), чтобы снизить нагрузку на администраторов.

Чтобы лучше познакомиться с локальными дисковыми пространствами, сначала рассмотрим дисковые пространства в высокодоступных системах хранения данных в Windows Server 2012 R2. В Windows Server 2012 R2 для работы высокодоступной системы, использующей дисковые пространства, накопители должны быть физически подключены ко всем узлам хранилища. Для этого накопители необходимо разместить во внешнем корпусе JBOD, физически подключив каждый узел к этому корпусу. Кроме того, поскольку к каждому накопителю будет подключено несколько узлов хранилища, необходимо использовать интерфейс SAS: протокол SAS допускает такой совместный доступ, тогда как интерфейс SATA не разрешает наличие нескольких инициаторов. В силу этих требований подобная топология развертывания называется «*дисковые пространства с общим JBOD*» (в отличие от локальных дисковых пространств). На рис. 2-58 показана схема дисковых пространств с общим JBOD.



Общий JBOD с дисками SAS.

Рис. 2-58. Пример развертывания дисковых пространств с общим JBOD

Такая топология развертывания обладает множеством преимуществ по сравнению с прежними высокодоступными системами хранения данных. Тем не менее необходимость физического подключения накопителей к каждому узлу налагает жесткие ограничения на тип допустимых устройств и может привести к образованию сложных конфигураций инфраструктуры SAS, особенно при масштабировании.

Локальные дисковые пространства в Windows Server 2016 позволяют создавать высокодоступные системы хранения данных, используя только локальные накопители. Это могут быть либо внутренние накопители каждого узла хранилища, либо накопители в устройствах JBOD, причем каждый JBOD достаточно подключить только к одному узлу. При этом можно полностью отказаться от инфраструктуры SAS с присущей ей сложностью, а также использовать накопители с интерфейсом SATA, что позволит добиться снижения затрат и повышения производительности. На рис. 2-59 показано развертывание локального дискового пространства.

Рис. 2-59. Пример развертывания Storage Spaces Direct

Важно понимать, что Storage Spaces Direct является развитием прежних версий Storage Spaces и расширением программно-определяемых хранилищ в Windows Server. Еще одна важная особенность Storage Spaces Direct — использование протокола SMB3 для передачи данных между узлами (такой обмен информацией также называется *east-west*). Используются все современные возможности SMB3, в том числе SMB Direct (на сетевых контроллерах, поддерживающих RDMA) для передачи данных с высокой пропускной способностью и малыми задержками, а также многоканальные возможности SMB Multichannel для объединения пропускной способности и реализации отказоустойчивости сети.

Подробная информация о внедрении

Локальные дисковые пространства легко интегрируются с уже существующими компонентами программно-определяемых хранилищ Windows Server: масштабируемыми файловыми серверами SOFS (SMB3), файловой системой общих томов кластера (CSVFS), дисковыми пространствами, отказоустойчивой кластеризацией. На рис. 2-60 показаны компоненты локальных дисковых пространств.

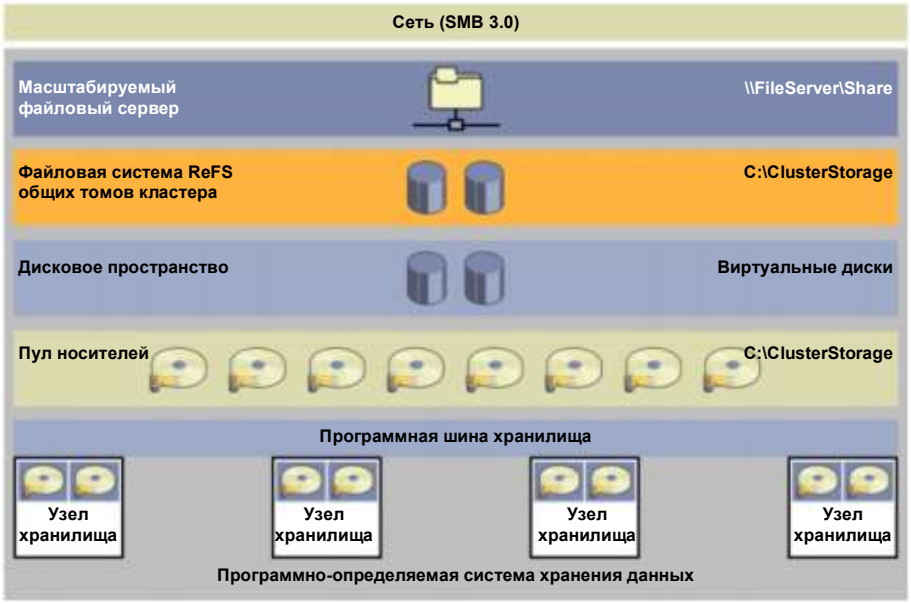


Рис. 2-60. Компоненты локальных дисковых пространств

Обновленный набор компонентов включает следующие элементы (снизу вверх):

- **Оборудование.** Система хранения данных содержит не менее двух узлов с локальными накопителями (рекомендуется от 4 узлов). Каждый узел хранилища может использовать внутренние накопители или внешний JBOD, подключенный по интерфейсу SAS. В качестве накопителей можно использовать диски и твердотельные накопители с интерфейсами SATA и SAS.
- **Программная шина хранилища.** Программная шина хранилища охватывает все узлы и объединяет локальные накопители в каждом узле, делая все накопители доступными для следующего уровня — дисковых пространств.
- **Дисковые пространства.** В дисковых пространствах содержатся пулы носителей и виртуальные диски. Пул носителей включает в себя все локальные накопители всех узлов. Виртуальные диски обеспечивают устойчивость в случае отказа дисков или узлов, поскольку копии данных хранятся на разных физических дисках.
- **Отказоустойчивая файловая система ReFS.** Файловая система ReFS используется для хранения файлов виртуальных машин Hyper-V. ReFS — основная файловая система Windows Server 2016 для развертывания виртуальной среды, она оптимизирована для технологии дисковых пространств и поддерживает такие возможности, как обнаружение ошибок и их автоматическое исправление. Кроме того, ReFS ускоряет работу VHD(X) в таких операциях, как создание фиксированных (fixed) VHD(X), их динамическое расширение и слияние. На уровнях CSVFS над ReFS все подключенные тома объединяются в одно пространство имен.
- **SOFS.** Это верхний уровень компонентов хранилища, он предоставляет удаленный доступ к системе хранения данных по протоколу SMB3.

Улучшенная масштабируемость

Локальные дисковые пространства можно развернуть на устройствах хранения, использующих локальные диски или наборы дисков JBOD без общего доступа. В предыдущих версиях Windows Server для масштабирования дисковых пространств требовалось одновременное масштабирование инфраструктуры SAS, соединявшей узлы хранилища с общими JBOD, состоявшими из дисков SAS. Используя локальные дисковые пространства, можно избежать сложности инфраструктуры SAS, а для масштабирования достаточно просто добавить новый узел хранилища с локальными дисками или с подключенным локальным JBOD. Масштабирование путем

добавления узлов повышает гибкость планирования хранилища, поскольку возможности расширения хранилища больше не ограничены количеством отсеков для дисков в JBOD с интерфейсом SAS, общим для всех узлов.

Дополнительному улучшению масштабирования по сравнению с прежними версиями Windows Server способствует и увеличение максимального количества устройств в одном пуле носителей. Большее число дисков в одном пуле позволяет обходиться меньшим количеством пулов, а значит, упрощает управление инфраструктурой хранения.

Оптимизация пула носителей для локальных дисковых пространств

Технология локальных дисковых пространств позволяет оптимизировать пул носителей путем равномерного распределения данных среди набора физических дисков, образующих пул. Со временем, по мере добавления и удаления физических дисков и по мере записи и удаления данных, распределение данных по физическим дискам, образующим пул, может стать неравномерным. Некоторые физические диски могут быть заполнены до предела, тогда как на других дисках в этом же пуле может остаться много места.

Чтобы задействовать новые ресурсы, при добавлении в пул новых дисков существующие данные следует оптимизировать. Это обеспечит повышение эффективности всего пула, а также увеличение производительности за счет дополнительной пропускной способности добавленного физического оборудования. Оптимизация пула — это задача обслуживания, выполняемая администратором.

После запуска команды на оптимизацию пула локальное дисковое пространство перемещает данные между физическими дисками в пуле. Это перемещение происходит в фоновом режиме, чтобы не затрагивать основные рабочие нагрузки или работу пользователей.

Сценарии неполадок

В локальных дисковых пространствах учитываются различные возможные сценарии неполадок. Для более полного понимания сначала поговорим о том, как работают виртуальные диски.

Виртуальный диск состоит из областей размером по 1 ГБ каждая. Таким образом, виртуальный диск объемом 100 ГБ состоит из 100 областей по 1 ГБ. Если используется зеркальное копирование виртуального диска (с помощью ResiliencySettingName), то в системе присутствует несколько копий этих областей. Количество копий можно узнать с помощью NumberOfDataCopies, у каждой области могут быть две или три копии. Например, зеркально дублированный виртуальный диск размером 100 ГБ с тремя копиями данных будет занимать 300 областей. Размещение областей определяется доменом сбоя, в локальных дисковых пространствах это узел (StorageScaleUnit). Как показано на рис. 2-61, три копии области (A) размещаются на трех разных узлах хранилища, в примере на этом рисунке — на узлах 1, 2 и 3. Три копии другой области (B) того же самого виртуального диска могут быть размещены на других узлах, на этом рисунке — на узлах 1, 3 и 4, и так далее. Это означает, что области виртуального диска распределяются по всем узлам хранилища, а копии каждой области размещаются на разных узлах. На рис. 2-61 показаны развертывание четырех узлов с зеркальным дублированием виртуального диска с тремя копиями и пример размещения областей.

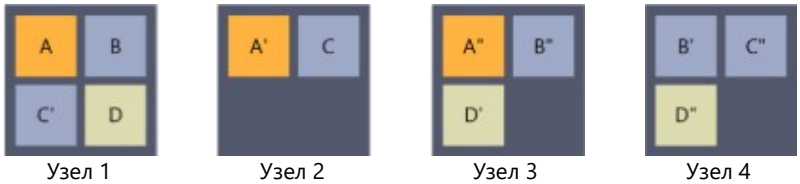


Рис. 2-61. Реализация с использованием четырех узлов

Теперь рассмотрим различные сценарии отказов и методы их обработки в дисковых пространствах.

Сценарий 1. Отказ одного или нескольких секторов на диске

В этом случае подсистема дисковых пространств перераспределяет область, затронутую отказом секторов. Диском назначения для перераспределения может быть другой диск этого же узла или диск другого узла, на котором еще нет копии этой области. Например, если три копии области находятся на узлах 1, 2 и 3, а область на узле 1 пострадала при отказе сектора, ее новая копия может быть создана на другом диске узла 1 или на любом диске узла 4. Диски на узлах 2 и 3 использовать нельзя, поскольку эти узлы уже содержат копию затронутой области.

Сценарий 2. Отказ диска

При обнаружении отказа диска подсистема дисковых пространств удаляет физический диск из пула носителей. После прекращения использования этого физического диска на всех виртуальных дисках начинается процесс восстановления. Поскольку физический диск был удален, виртуальные диски создают новую копию всех областей, которые были на удаленном физическом диске. Новые копии размещаются на узлах по той же логике, что описана в сценарии 1.

Сценарий 3. Отсутствие диска

В этом сценарии подсистема дисковых пространств выполняет одно из двух возможных действий:

- если отсутствует лишь физический диск, подсистема удаляет этот диск из пула носителей;
- если также отсутствуют узел хранилища (сервер) или дисковая полка, частью которых являлся физический диск, то подсистема не удаляет этот физический диск.

Причина, по которой во втором случае физический диск не удаляется, состоит в том, что при перезагрузке или при временном обслуживании узла хранилища все диски и дисковые полки, связанные с этим узлом, будут считаться отсутствующими. Автоматическое удаление всех этих дисков и корпусов может повлечь огромный объем работы по восстановлению: потребуется восстановить все области этих дисков на других узлах системы хранения данных. Объем восстанавливаемых данных при этом может достигать нескольких терабайтов. Если диски и дисковые полки действительно отсутствуют и больше не будут использоваться в системе хранения данных, администратор должен самостоятельно удалить их из пула носителей и запустить восстановление.

Сценарий 4. Перезапуск или обслуживание узла хранилища

В этом случае подсистема дисковых пространств не удаляет физические диски из пула носителей по причинам, описанным выше в сценарии 3. Когда узел хранилища снова будет включен, подсистема автоматически обновит все области, устаревшие по сравнению с копиями, которые не были затронуты перезапуском или обслуживанием.

Сценарий 5. Окончательный отказ узла хранилища

В этом случае администратор должен удалить все затронутые физические диски из пула носителей, при необходимости добавить дополнительные узлы хранилища в систему хранения данных, а затем приступить к исправлению. Этот процесс не выполняется автоматически, поскольку подсистема дисковых пространств не располагает информацией о том, является отказ временным или окончательным. Нежелательно начинать исправление, которое может повлечь значительный объем восстановительных работ.

> **Примечание.** Более подробная информация о локальных дисковых пространствах в Windows Server 2016 доступна по адресу https://technet.microsoft.com/library/mt126109.aspx.

Дедупликация

В подсистеме дедупликации Windows Server 2016 повышена производительность и значительно расширен масштаб использования этой технологии. В Windows Server 2016 поддерживаются следующие возможности:

- тома размером до 64 ТБ;
- файлы размером до 1 ТБ;
- улучшенная работа с виртуализированными приложениями резервного копирования
- поддержка Nano Server;
- последовательное обновление кластера.

С момента появления дедупликации в Windows Server 2012 основные принципы преобразования данных в блоки сохранились в неизменном виде. Рассмотрим изменения, которые позволили реализовать новые возможности и сценарии использования.

Подсистема оптимизации усовершенствована с однопоточной до параллельной многопоточной — используются несколько потоков ввода-вывода, как показано на рис. 2-62.

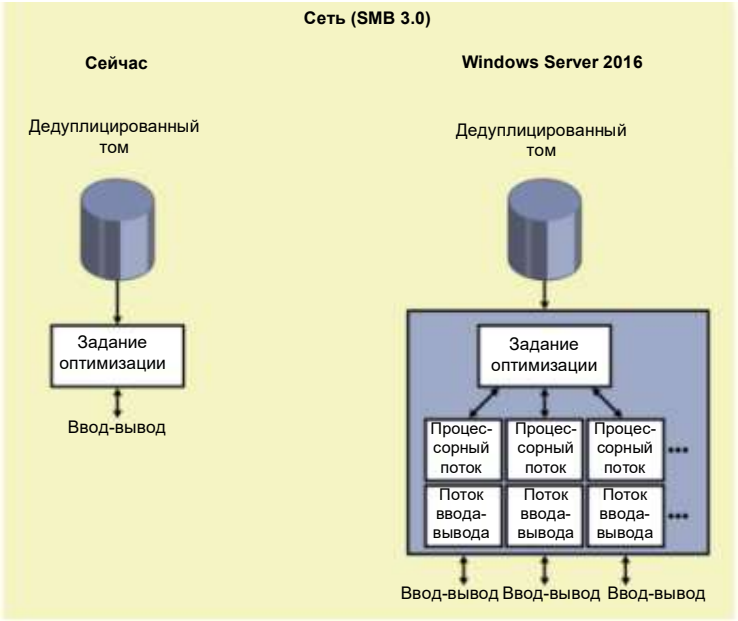


Рис. 2-62. Однопоточная обработка и параллельная обработка

Помимо параллельного выполнения, был переработан алгоритм обработки файлов: теперь используется новая структура карты потоков, улучшена частичная оптимизация файлов, увеличены масштабируемость и производительность, поддерживаются файлы размером до 1 ТБ. На рис. 2-63 показано изменение технологии сопоставления с целью более эффективной оптимизации тома в целом.

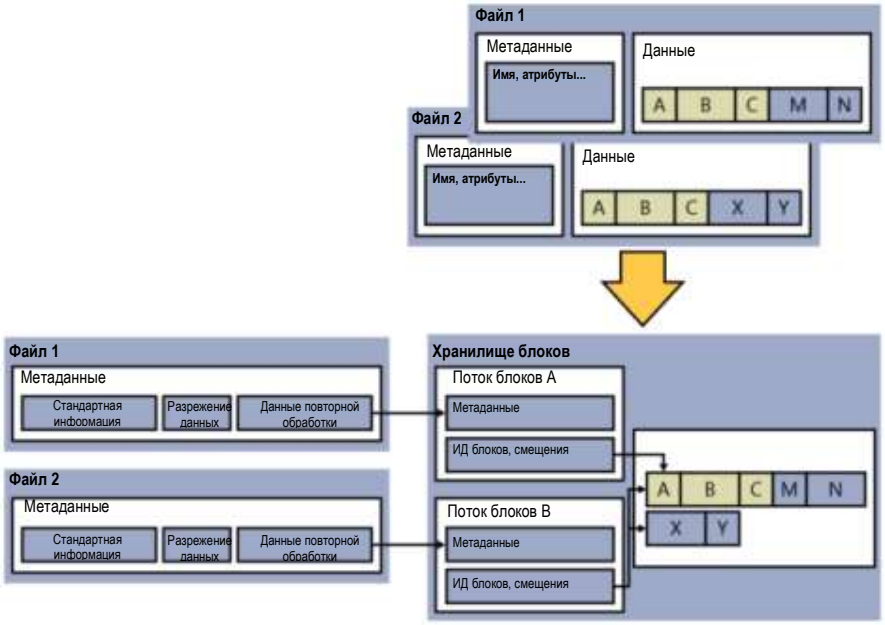


Рис. 2-63. Старая структура сопоставления и новая структура карты потоков в Windows Server 2016

Корпорация Microsoft предоставляла техническую поддержку и рекомендации по использованию дедуплицированных томов с DPM в Windows Server 2012 R2, но в Windows Server 2016 реализованы новые усовершенствования, упрощающие настройку и поддержку. Для настройки можно использовать графический пользовательский интерфейс Windows 2016 или командлет Enable-DeDup в Windows PowerShell, включая новый атрибут Backup.

```
Enable-DedupVolume -Volume <том> -UsageType Backup
```

Еще одна удобная новая возможность — поддержка последовательного обновления кластера. Можно начать обновление узлов кластера до Windows Server 2016, поддерживая при этом процесс дедупликации в Windows Server 2012, — ранее такой возможности не было. В процессе перехода на Windows 2016 все задания будут выполняться в режиме Windows 2012.

Еще одно небольшое обновление, достойное упоминания, — доступность интерфейса Storage Management API (SMAPI) для использования в System Center Virtual Machine Manager 2016. Благодаря этому нововведению можно настраивать дедупликацию хранилища и отчеты о состоянии в System Center Virtual Machine Manager 2016.

Качество обслуживания хранилища

В предыдущих версиях Windows Server возможность применять политики качества обслуживания (QoS) к трафику системы хранения данных отсутствовала, это приводило к возникновению серьезных проблем при виртуализации: если требовалось сделать определенные рабочие нагрузки приоритетными и предоставить им, к примеру, гарантированное количество операций ввода-вывода в секунду (IOPS), выполнить это было просто невозможно.

В Windows Server 2016 можно принудительно устанавливать равнодоступность или приоритет ресурсов в зависимости от политик, которые нужно настроить для хранилища. Применение политики обслуживания в отношении хранилища связано главным образом с виртуальными машинами Hyper-V, развернутыми на масштабируемых файловых серверах (SOFS) или в кластерах Hyper-V с общими томами (CSV).

На рис. 2-64 показано создание различных политик и их применение к разным виртуальным машинам, что позволяет использовать разные стандарты обслуживания или выделять приоритетные рабочие нагрузки.

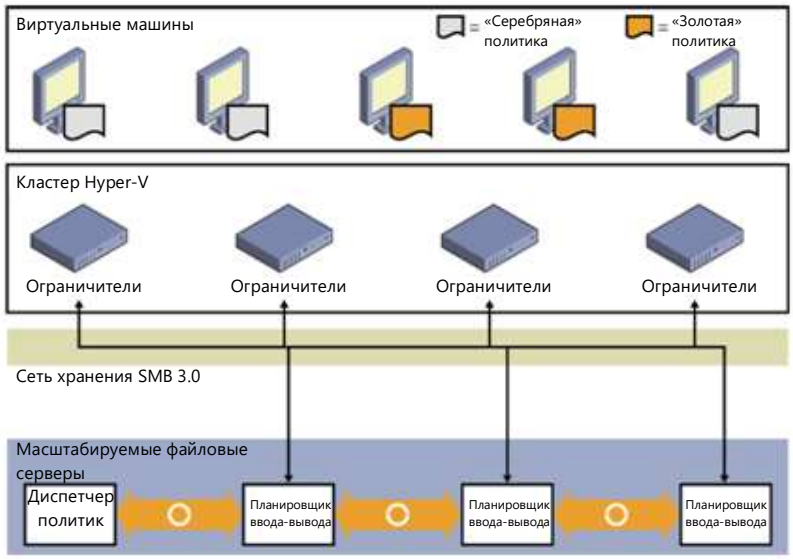


Рис. 2-64. Политики качества обслуживания хранилища и их применение к разным уровням виртуальных машин

В Windows Server 2016 функция качества обслуживания хранилища включена по умолчанию, и для ее использования не нужно устанавливать дополнительные роли или компоненты.

На рис. 2-65 показано, что если у вас есть отказоустойчивый кластер Hyper-V, то новый кластерный ресурс будет в списке.

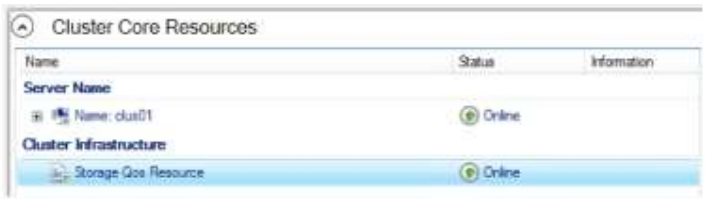


Рис. 2-65. Качество обслуживания хранилища как кластерный ресурс

Для отображения такого же результата можно использовать командлет Get-ClusterResource в Windows PowerShell:

```
Get-ClusterResource -Name "Storage Qos Resource"
```

Качество обслуживания хранилища эффективно работает только в случае, если определены соответствующие политики. С помощью политик можно управлять потоком трафика нужным образом в соответствии с требованиями. Есть возможность настроить политики качества обслуживания хранилища на масштабируемом файловом сервере (SOFS). Существуют два основных типа политик:

- **Политики единственного экземпляра**. С помощью политик единственного экземпляра можно указывать минимальное и максимальное количество операций ввода-вывода в секунду (IOPS) для каждой политики. Указанные значения являются суммарными для каждой виртуальной машины. Например, если виртуальная машина включает один виртуальный диск VHD/VHDX, она сможет использовать все количество операций ввода-вывода в секунду, назначенное политикой. Если же виртуальная машина содержит три диска VHD/VHDX, каждому из которых назначена одна и та же политика единственного экземпляра, максимальное количество операций ввода-вывода в секунду будет разделено между дисками, что приведет к снижению общей производительности. Можно создать несколько политик единственного экземпляра и настроить для каждого диска отдельную

политику, чтобы все диски получили доступ ко всей пропускной способности. Если используются две виртуальные машины, каждая с одним виртуальным диском VHD, то при назначении им одной и той же политики единственного экземпляра минимальное и максимальное количество операций ввода-вывода в секунду также будет разделено между ними.

- **Политики множественных экземпляров**. В политике множественных экземпляров тоже можно задать минимальное и максимальное количество операций ввода-вывода в секунду. Однако в этом случае, если на двух виртуальных машинах используется по одному диску VHD/VHDX, каждый из них воспринимает эти ограничения независимо от другого. При этом такой же принцип действует, и если у одной машины имеется несколько дисков: назначенное минимальное и максимальное количество операций ввода-вывода в секунду будет независимым для каждого диска, если не использовать индивидуальные политики.

Для создания политики используйте следующий командлет Windows PowerShell:

```
$GoldVmPolicy = New-StorageQosPolicy -Name Gold -PolicyType MultiInstance -Minimumlops 100
-Maximumlops 500
```

В этом примере информация о политике будет сохранена в переменной. Потребуется свойство, которое называется Policyid. Для доступа к свойству Policyid используйте следующий синтаксис:

```
$GoldVmPolicy.PolicyId
Guid
Cd5f6b87-fa15-402b-3545-32c2f456f6e1
```

GUID потребуется для применения этой политики к VHD с помощью следующей команды Windows PowerShell:

```
Get-VM -Name GoldSrv* | Get-VMHardDiskDrive | Set-VMHardDiskDrive -QoSPolicylD
Cd5f6b87-fa15-402b-3545-32c2f456f6e1
```

После применения политики нужно проверить, во-первых, включена ли она, а во-вторых, действует ли она нужным образом. Для этого можно использовать командлет Get-StorageQoSFlow.

В следующих выходных данных показано, какие политики применены и какую долю пропускной способности хранилища использует виртуальная машина:

```
Get-StorageQosFlow -InitiatorName GoldVml | Format-List

FilePath:c:\ClusterStorage\Volume1\VMS\Gold\GoldVM1.VHDX
Flowld: ebfecb54-e47a-5a2d-8ec0-0940994ff21c
Initiatorld             :     ae4e3dd0-3bde-42ef-b035-9064309e6fec
InitiatorlOPS           :     464
InitiatorLatency        :     26.2684
InitiatorName           :     GoldVM1
InitiatorNodeName       :     node1.contoso.com
Interval                :     300000
Limit                   :     500
Policyid                :     cd5f6b87-fa15-402b-3545-32c2f456f6e1
Reservation             :     500
Status                  :     Ok
StorageNodeIOPs         :     475
StorageNodeLatency      :     6.5625
StorageNodeName         :     node1.contoso.com
TimeStamp               :     2/12/2016 3:28:49 AM
Volumeld                :     2d34fc5a-2b3f-9922-23f4-43563b2a6787
PSComputerName          :
Maximumlops             :     100
Minimumlops             :     500
```

Перед созданием политик можно использовать командлет Get-StorageQoSFlow для проверки фактического количества операций ввода-вывода в секунду, которое используется виртуальными машинами.

```
Get-StorageQosFlow | Sort-Object StorageNodeIOPs -Descending | ft InitiatorName,
@{Expression={$_.InitiatorNodeName.Substring(0,$_.InitiatorNodeName.IndexOf('.'))};Label="Init
iatorNodeName"}, StorageNodeIOPs, Status,
@{Expression={$_.FilePath.Substring($_.FilePath.LastIndexOf('\')+1)};Label="File"}
```

```
-AutoSize

InitiatorName     InitiatorNodeName StorageNodeIOPs   Status      File
GoldVM5           node1             2482              Ok          IOMETER.VHDX
GoldVM2           node2             344               Ok          BUILDVM2.VHDX
GoldVM1           node2             597               Ok          BUILDVM1.VHDX
GoldVM4           node1             116               Ok          BUILDVM4.VHDX
GoldVM3           node2             526               Ok          BUILDVM3.VHDX
GoldVM4           node1             102               Ok
```

Примечание. Более подробная информация об использовании качества обслуживания хранилища в Windows 2016 доступна по адресу https://technet.microsoft.com/library/mt126108.aspx.

Сети

По мере развития программно-определяемых центров обработки данных значительно возросла сложность обслуживания традиционных сетевых решений, в частности, виртуальных локальных сетей (VLAN), и управления ими. Для эффективного развития центров обработки данных требуется возможность централизованного программного управления сетью с динамическим созданием нужных ресурсов. Решение, получившее название «программно-определяемые сети», впервые появилось в Windows Server 2012 R2. В новой версии это решение усовершенствовано и работает аналогично соответствующим технологиям Azure.

В этом разделе мы рассмотрим следующие функциональные возможности:

- Виртуализация сетей
- Сетевой контроллер
- Мультитенантный BGP-маршрутизатор шлюза удаленного доступа
- Программный балансировщик нагрузки
- Брандмауэр центра обработки данных
- Прокси-служба веб-приложений (WAP)

Виртуализация сетей

Важную роль при принятии решений, касающихся управления ИТ-инфраструктурой играет эффективность использования ресурсов оборудования. За прошедшее десятилетие в корпоративных ИТ-средах прочно обосновались технологии виртуализации, повышающие их эффективность за счет выполнения множества рабочих нагрузок на одном сервере. В области же сетевых технологий сохраняется ряд традиционных ограничений, в частности, по количеству виртуальных локальных сетей (VLAN): их может быть не более 4096. Это немало, но в ряде сегментов отрасли ИТ (например, у поставщиков услуг) этот предел достигается очень быстро. Для решения подобных проблем можно применять виртуализацию сетей. Впрочем, это не единственный случай, когда виртуализация сетей является целесообразным и эффективным решением. Представьте себе, к примеру, компанию, которая расширяется путем приобретения других компаний. В таких случаях их IP-пространства нередко накладываются на схему или структуру корпоративной сети поглощающей компании. Казалось бы, ничего страшного. Но что если действующие лицензионные соглашения, имеющиеся у приобретенных компаний, «привязаны» к определенным IP-адресам серверов, причем изменить или расторгнуть такие соглашения невозможно без существенных издержек либо компания, первоначально приобретавшая лицензии, более не существует?

Как бы то ни было, виртуализация сетей образует основу для программно-определяемых сетей в центрах обработки данных, как показано на рисунке 2-66.

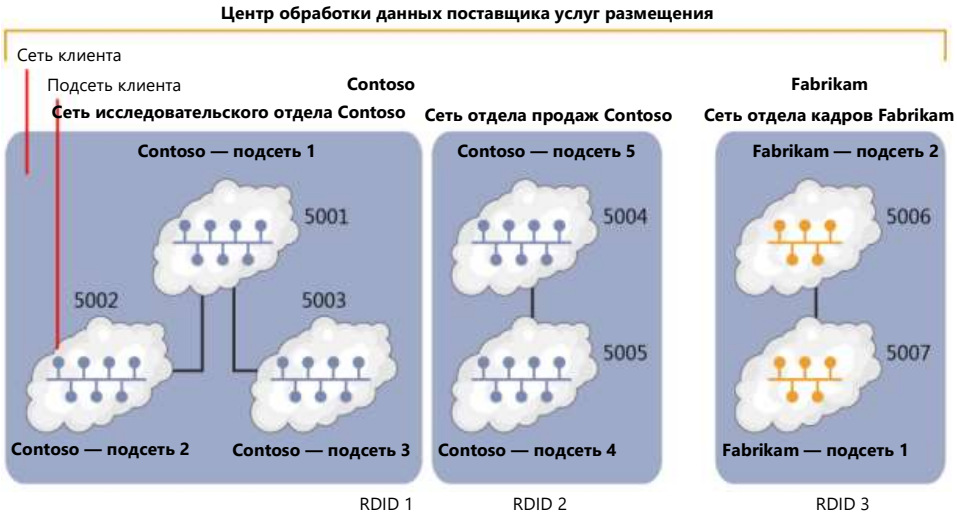


Рисунок 2-66. Виртуализация сети

На рисунке 2-66 показаны некоторые основные компоненты программно-определяемых сетей. Первый — *идентификатор домена маршрутизации* (RDID). Его также можно считать виртуальной сетью, такой же, как и в Azure. Каждая виртуальная сеть определяет границу этой сети: обмениваться данными между собой могут только подсети, находящиеся внутри одной и той же виртуальной сети. Виртуальные сети находятся на уровне 3 сетевой модели OSI (то есть IP), поэтому изолировать трафик разных сетей очень просто: изоляция обеспечивается с помощью полей идентификатора NVGRE или сетевого идентификатора VXLAN (VNI). В каждом домене маршрутизации можно создать несколько виртуальных подсетей (с разными идентификаторами VSID), которые смогут обмениваться данными между собой. На рисунке 2-66 домен маршрутизации RDID 1 содержит три подсети с разными идентификаторами VSID. Обмен данными между подсетью VSID 5001 и подсетью VSID 5002 может быть организован путем переадресации в виртуальном коммутаторе на уровне 2. Когда пакет проходит через коммутатор, он инкапсулируется, к нему применяются сопоставления (заголовок инкапсуляции), после чего пакет отправляется на порт или коммутатор назначения в Hyper-V. Таким образом, если у трафика один и тот же идентификатор RDID, его можно переадресовывать между разными VSID, но если RDID отличается, то придется использовать шлюз.

Привычные сетевым администраторам понятия, такие как ARP-запросы, MAC-адреса и широковещательные домены, формально сохранены, но относятся к разным областям. Например, виртуальная подсеть VSID 5001 может рассматриваться и как широковещательный домен, и как виртуальная локальная сеть. Если два компьютера в подсети VSID 5001 хотят установить соединение, они находят MAC-адрес, отправив ARP-запрос к коммутатору Hyper-V. Этот коммутатор имеет порт для каждого сетевого адаптера виртуальных машин, а у каждого сетевого адаптера, в свою очередь, есть MAC-адрес. Коммутатор Hyper-V поддерживает таблицу подстановки или таблицу ARP с этими данными, чтобы при поступлении ARP-запросов знать, куда направлять трафик. Если нужной записи нет, коммутатор отправляет ARP-запрос, пытаясь определить, к какому порту подключен компьютер, на который требуется передать трафик. Этой подсети (или VSID) также будет назначена IP-подсеть. например, 192.168.0.0/24, которая может быть подмножеством выделенных сетевых адресов 192.168.0.0/16.

Виртуализация сети Hyper-V на одном сервере устроена сравнительно просто. Однако картина несколько усложняется при использовании нескольких серверов в рамках одного центра обработки данных, когда требуется соблюдать границы RDID и VSID изолированных сетей и подсетей — возникает необходимость ввести определение ряда понятий адресного пространства. На рисунке 2-67 показана схема из нескольких подсетей, которые накладываются одна на другую.

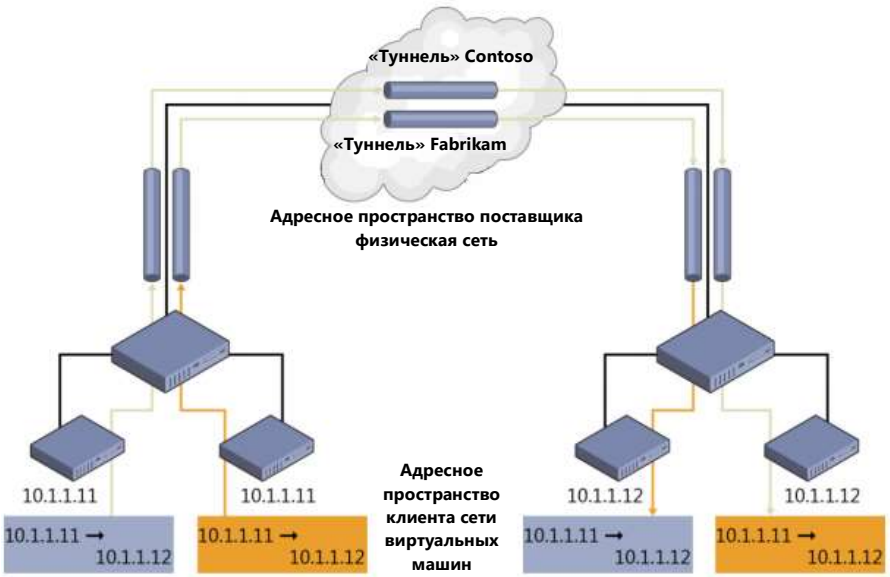


Рисунок 2-67. Адресные пространства.

Если бы эти подсети не были изолированы, то в сети возникло бы множество неполадок, многие компоненты попросту не смогли бы работать. Благодаря виртуализации сетей, в которой пространства IP-адресов изолированы, они могут сосуществовать в рамках одной и той же физической сети. Рассмотрим синюю и оранжевую сети на рисунке 2-67: эти адресные пространства означают *адресное пространство клиента*. Чтобы эти адресные пространства могли обмениваться данными между узлами физической сети, их необходимо инкапсулировать с помощью общей IP-подсети. Эта общая подсеть, к которой подключен сервер, называется *адресным пространством поставщика*. Далее происходит следующее: адресное пространство клиента сопоставляется с IP-адресом в адресном пространстве поставщика, и когда виртуальные машины с адресами в пространстве клиента хотят обмениваться информацией между разными узлами, они используют для этого соответствующие IP-адреса в пространстве поставщика.

Методика сопоставления зависит от типа используемой технологии. При виртуализации сети можно использовать VXLAN или NVGRE.

Ниже приведен фрагмент статьи на портале TechNet (https://technet.microsoft.com/library/mt238303.aspx) с описанием VXLAN и NVGRE:

Протокол виртуальной расширяемой локальной сети (Virtual eXtensible Local Area Network, VXLAN) (RFC 7348) повсеместно принят в отрасли и поддерживается крупнейшими поставщиками, в том числе компаниями Cisco, Brocade, Arista, Dell, HP и другими. В качестве транспортного протокола применяется UDP. Для VXLAN используется назначенный ассоциацией IANA порт назначения UDP 4789, причем номер исходного порта UDP должен представлять собой хэш информации из внутреннего пакета, используемой для распространения ECMP. После заголовка UDP к пакету присоединяется заголовок VXLAN, он содержит зарезервированное 4-байтовое поле, за которым следует 3-байтовое поле идентификатора сети VXLAN (VNI) — VSID, за которым следует еще одно зарезервированное однобайтовое поле. После заголовка VXLAN присоединяется исходный кадр клиентского адреса уровня 2 (без FCS Ethernet-кадра).

NVGRE используется в составе заголовка туннеля. В NVGRE пакет виртуальной машины инкапсулируется внутрь другого пакета. Заголовок этого нового пакета имеет соответствующие IP-адреса источника и назначения в пространстве поставщика в дополнение к идентификатору VSID, который хранится в поле Key заголовка GRE, как показано на рисунке 2-68.

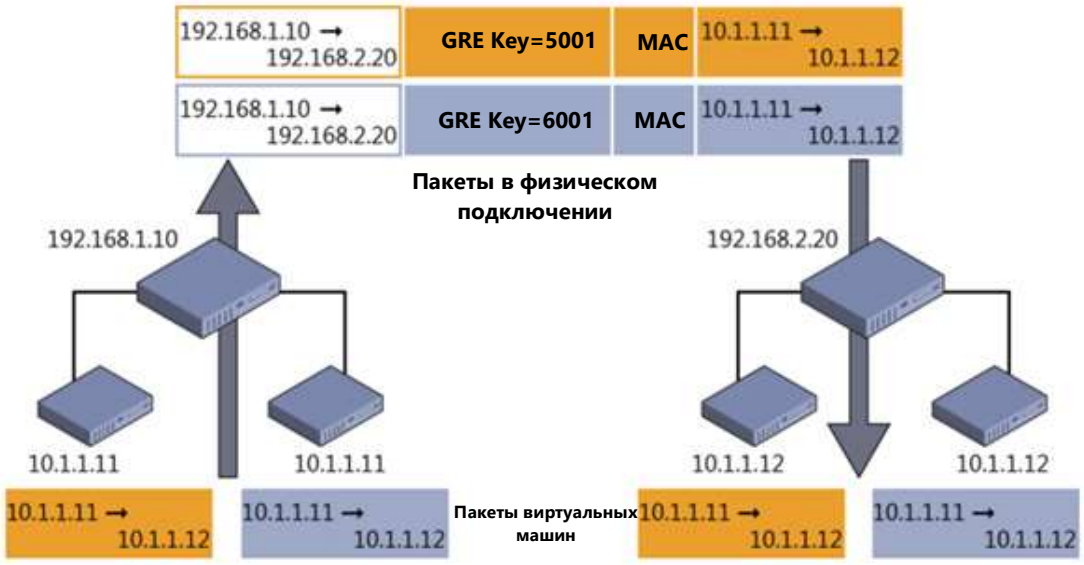


Рисунок 2-68. Инкапсуляция маршрутов при виртуализации сети (NVGRE).

Сетевой контроллер

В Windows Server 2016, что очень важно, используется совершенно новый подход к виртуализации сетей (назовем его версией 2): многие компоненты коренным образом изменены так, чтобы обеспечить однородность реализации сети и структуры с реализацией в Azure. Общий принцип виртуализации сетей остался в целом таким же, как мы описали ранее, но некоторые технологии в Windows Server 2016 были переработаны. Перед тем, как подробно рассмотреть сетевые контроллеры и программную балансировку нагрузки, следует отметить, что был переделан и расширяемый коммутатор Hyper-V. Возможности расширения, доступные в виртуализации сетей версии 1, не будут работать в версии 2. Вместе с тем, в версии 2 реализована технология виртуальной платформы фильтрации Azure, единая для частных и общедоступных облаков.

В этом разделе описывается сетевой контроллер — это, по сути, «мозг» внедряемого решения для виртуализации сетей. В крупных и сложных сетях с традиционными сетевыми технологиями целесообразно задействовать централизованный инструмент для управления такими сетями. Эти централизованные инструменты предоставляют нам единую точку управления, дают возможность автоматизировать настройку, обслуживание, резервное копирование и устранение неполадок в физической среде. В виртуальной сетевой среде за все это отвечает сетевой контроллер.

Сетевой контроллер может взаимодействовать с сетью. С ним самим можно взаимодействовать посредством двух API-интерфейсов, у каждого из которых имеется свое предназначение. «Восходящий» (Northbound) API-интерфейс (реализован в виде REST API) служит для взаимодействия с сетевым контроллером, мониторинга сети и внедрения изменений конфигурации. «Нисходящий» (Southbound) API-интерфейс служит для взаимодействия с сетевыми устройствами, обнаружения конфигурации служб и анализа сети в целом. Управлять сетью и отслеживать ее работу непосредственно из консолей этих решений можно с помощью System Center Operations Manager и System Center Virtual Machine Manager.

Сетевой контроллер может управлять всеми технологиями виртуализации сетей, которые входят в состав Windows Server 2016. В следующей таблице перечислены области, которыми может управлять сетевой контроллер, с описанием поддерживаемых задач управления.

Компонент	Управляемые области
Управление сетевой структурой	• Управление физической структурой • IP-подсети • Виртуальные локальные сети (VLAN) • Коммутаторы уровня 2 • Коммутаторы уровня 3 • Сетевые адаптеры на узлах
Управление брандмауэром	• Управление правилами брандмауэра для портов виртуальных коммутаторов в сети виртуальных машин • Централизованный журнал трафика на коммутаторе
Мониторинг сетей	• Мониторинг физической сети • Мониторинг виртуальной сети • Активный мониторинг сетей • Сбор данных элементов с помощью ловушек и опросов SNMP • Анализ воздействия
Топология сети и управление обнаружением	• Обнаружение сетевой топологии с помощью элементов
Программная балансировка нагрузки	• Управление и настройка правил балансировки нагрузки
Управление виртуальными сетями	• Политики виртуальной сети • Все элементы виртуализации сетей
Управление шлюзом службы удаленного доступа	• Добавление и удаление виртуальных машин шлюза из кластера, настройка требуемого уровня резервного копирования • Подключение шлюза VPN «сеть-сеть» между удаленными сетями арендаторов и вашим центром обработки данных по протоколу IPsec • Подключение шлюза VPN типа «сеть-сеть» между удаленными сетями арендаторов и вашим центром обработки данных по протоколу GRE • Подключение шлюза VPN «точка-сеть», чтобы администраторы арендаторов могли работать со своими ресурсами, размещенными в вашем центре обработки данных, из любого места • Возможности переадресации на уровне 3 • Маршрутизация BGP с возможностью управления маршрутизацией сетевого трафика между сетями виртуальных машин арендаторов и их удаленными сайтами

Примечание. В этой книге мы не можем предоставить исчерпывающее описание всех функций сетевого контроллера и особенностей его работы. Дополнительные сведения представлены в статье на портале TechNet по адресу https://technet.microsoft.com/en-US/library/dn859239.aspx.

Сетевой контроллер отличается сложностью в развертывании и настройке перед использованием. В следующих статьях представлена наиболее актуальные сведения о развертывании и настройке сетевого контроллера в Windows Server 2016:

«Требования к установке и подготовке к развертыванию сетевого контроллера»: https://technet.microsoft.com/library/mt691521.aspx

«Развертывание сетевого контроллера с помощью Windows PowerShell»: https://technet.microsoft.com/library/mt282165.aspx

Мультитенантный BGP-маршрутизатор шлюза удаленного доступа

При развертывании виртуализации сетей и использовании методов инкапсуляции и изоляции, которые описаны выше в этой главе, возникает интересный вопрос: как виртуальные машины в этих изолированных сетях будут обмениваться данными с устройствами, которые находятся вне изолированной сети? И как внешние устройства будут обмениваются данными с этими изолированными виртуальными машинами?

В Windows Server 2016 в роль шлюза удаленного доступа добавлена поддержка протокола BGP. В прежних версиях поддерживались следующие возможности такого шлюза:

- VPN типа «сеть-сеть»
- VPN типа «точка-сеть»
- Туннели GRE
- Преобразование сетевых адресов (NAT)

Все эти возможности доступны и в нынешнем шлюзе удаленного доступа, что позволяет добиться следующего:

- Виртуальные машины могут обмениваться данными с сетями, находящимися вне домена маршрутизации, которому они назначены.
- При необходимости можно создавать конечные точки для виртуальной сети.
- Есть возможность соединять виртуальные и физические сети.

С появлением протокола BGP открываются новые возможности построения сетевой среды. Например, на BGP основана вся работа Express Route, не говоря уж о том, насколько широко BGP используется в Интернете.

Маршрутизатор BGP динамически определяет подключенные сети и объявляет о них другим маршрутизаторам, поддерживающим BGP. Эти маршрутизаторы на основании полученных записей могут формировать свои таблицы маршрутизации, и если маршрутизатор BGP получит запрос на передачу трафика в сеть клиента, он будет знать, по какому маршруту отправить этот трафик. Важная способность маршрутизатора BGP — дублирование маршрутов и автоматическое повторное вычисление оптимального маршрута к нужной сети. Если подключены несколько маршрутизаторов и существует несколько маршрутов, ведущих к нужной сети, то BGP выберет оптимальный маршрут. В случае отказа компонентов сети маршрутизатор BGP заново рассчитает маршрут и объявит его другим устройствам BGP.

Одна из проблем, связанных с виртуализацией сетей и шлюзом удаленного доступа в Windows Server 2012, касалась взаимосвязи шлюзов: если в Windows Server 2012 требовалось развернуть высокодоступный пул шлюзов, то разделить такой пул для разных функций или клиентов было невозможно. Также существовали строгие требования по размещению узлов шлюза, из-за чего в корпоративных кластерах возникало множество трудностей.

В Windows Server 2016 реализована гораздо более удобная модель пулов, в которой можно создавать пулы для какой-либо одной определенной функции или для набора функций. На рисунке 2-69 показано развертывание пулов для разных функций.

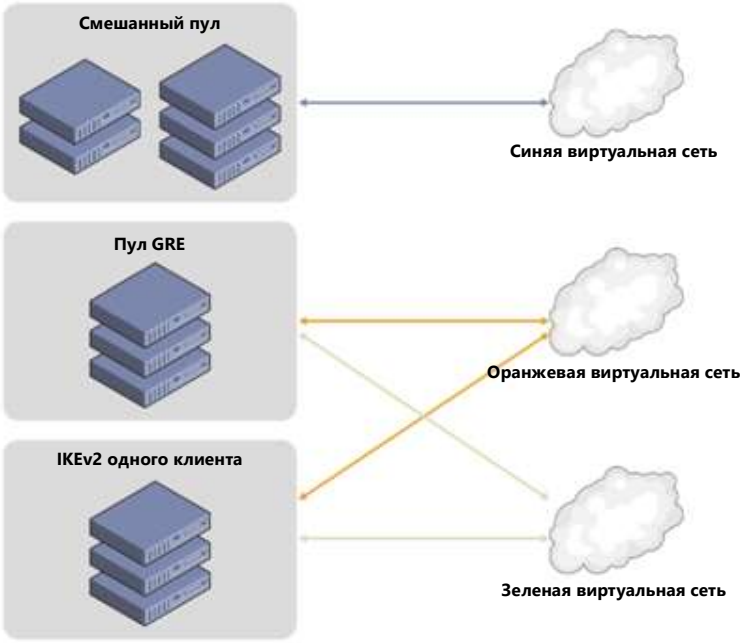


Рисунок 2-69. Пулы шлюзов удаленного доступа.

Примечание. Мультитенантный BGP-маршрутизатор шлюза удаленного доступа — достаточно сложная тема для обсуждения, к тому же его функционал постоянно обновляется. Дополнительные сведения доступны по адресу https://technet.microsoft.com/library/mt679502.aspx.

Программный балансировщик нагрузки

В Windows Server 2016 появилась функция программной балансировки нагрузки (SLB), обеспечивающая рабочим нагрузкам клиентов высокую доступность и масштабируемость. Программная балансировка нагрузки поддерживает следующие возможности:

- Службы балансировки нагрузки на уровне 4 для «вертикального» («North-South») и «горизонтального» («East-West») трафика TCP/UDP.
- Балансировка нагрузки сетевого трафика в общедоступных и внутренних сетях.
- Поддержка динамических IP-адресов в VLAN и виртуальных сетях, созданных с помощью виртуализации сетей Hyper-V.
- Поддержка проверки работоспособности.
- Облачное масштабирование, в том числе возможность горизонтального и вертикального масштабирования для мультиплексоров и агентов узлов.

При проектировании программного балансировщика нагрузки учитывалась возможность работы с пропускной способностью в десятки гигабайт на каждый кластер, поэтому получившееся решение позволило получить достойную альтернативу традиционным аппаратным решениям для балансировки нагрузки.

Перед подробным описанием программной балансировки нагрузки определим пару терминов:

- **Виртуальный IP-адрес** — IP-адрес, на который будут направлены все внешние подключения.

- **Динамический IP-адрес** — набор IP-адресов виртуальных машин, поддерживающих работу службы.

При наличии службы, которая запрашивает программную балансировку нагрузки, сетевой контроллер получает уведомление о запросе и предоставляет мультиплексор программного балансировщика нагрузки. В одной среде может быть несколько разных мультиплексоров. Каждому из них назначается виртуальный IP-адрес, после чего BGP объявляет сети о доступности этого виртуального IP-адреса. Мультиплексор также отвечает за прием подключений и их перенаправление виртуальным машинам, поддерживающим работу службы. Виртуальный IP-адрес объявляется с помощью BGP и находится под управлением сетевого контроллера, поэтому при отказе мультиплексора сетевой контроллер может восстановить оборвавшиеся подключения, для этого он запускает новый мультиплексор и заново объявляет маршруты через BGP. На рисунке 2-70 показана архитектура программной балансировки нагрузки.

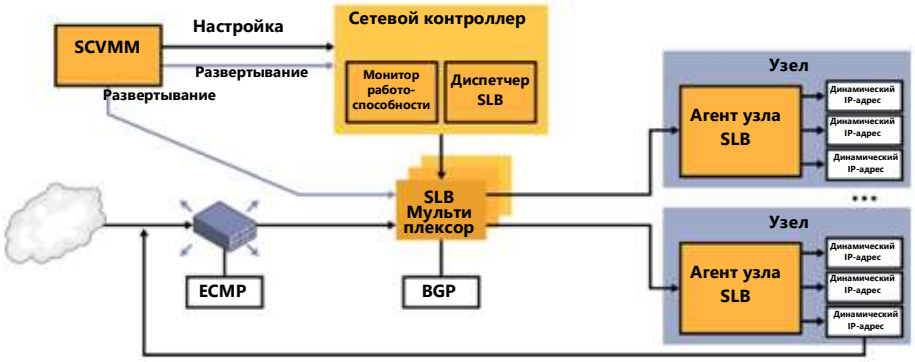


Рисунок 2-70. Общее устройство программного балансировщика нагрузки.

Примечание. Прежде чем использовать программный балансировщик нагрузки, необходимо установить и настроить сетевой контроллер. Дополнительные сведения представлены в статье на портале TechNet по адресу https://technet.microsoft.com/library/mt632286.aspx.

Брандмауэр центра обработки данных

В Windows Server 2016 появился новый компонент — брандмауэр центра обработки данных, это брандмауэр сетевого уровня, обладающий следующими возможностями:

- Проверка пакетов с сохранением состояния
- Поддержка мультитенантного доступа
- Проверка по правилам с пятью кортежами (протокол, номера портов источника и назначения, IP-адреса источника и назначения)

Это мультитенантное решение, его можно использовать для защиты рабочих нагрузок в виртуальных машинах арендаторов, его настройкой занимаются администраторы пользователей. Брандмауэр центра обработки данных (показан на рисунке 2-71) позволяет реализовать политики безопасности, которым подчиняется ИТ-инфраструктура вашей организации.

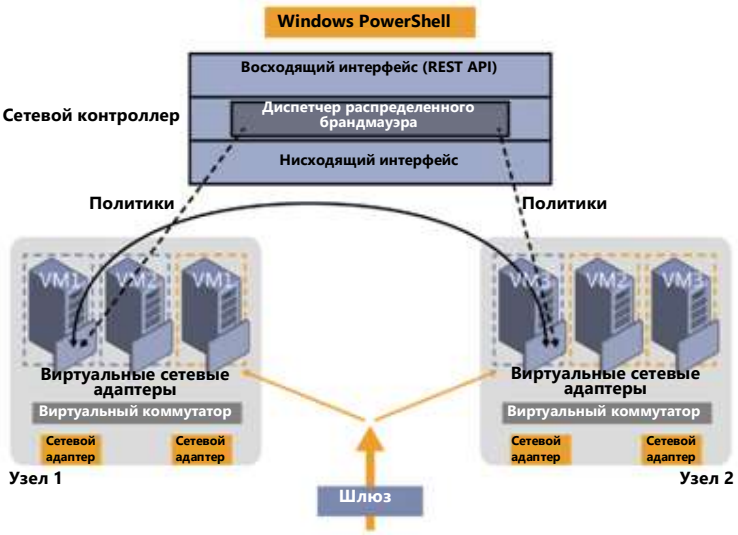


Рисунок 2-71. Брандмауэр центра обработки данных

Брандмауэром центра обработки данных управляет сетевой контроллер. Администратор арендатора может настраивать политики и напрямую применять их к виртуальным портам коммутатора Hyper-V. Кроме того, при перемещении рабочей нагрузки клиента по центру обработки данных между его узлами вместе с рабочей нагрузкой может передаваться и политика, которую определил этот клиент.

Прокси-служба веб-приложения

В этом разделе Юрий Диогенес (Yuri Diogenes) и Дэвид Брэнскам (David Branscome) расскажут об использовании в Windows Server 2016 обновленной прокси-службы веб-приложения для обеспечения удобного доступа к информации из любого места, где бы пользователь ни находился.

Усовершенствования возможностей публикации

Пользователи могут обращаться к корпоративным данным, применяя для этого различное клиентское оборудование (ноутбуки, планшеты, смартфоны), которое для простоты будем называть *устройствами*. Они могут отправлять запросы из разных географических точек, но при этом пользователи ожидают таких же возможностей работы, как если бы они находились в офисе компании. ИТ-служба должна обеспечить безопасность всего канала передачи данных от места их хранения в центре обработки данных (в локальной среде или в облаке) и до попадания на устройство назначения. Там эти данные будут сохранены, поэтому их также необходимо защитить.

Чтобы пользователи могли безопасно обращаться к данным компании, в Windows Server 2016 были расширены возможности прокси-службы веб-приложения с учетом возможности использования личных устройств для служебных нужд (BYOD). В частности, поддерживается предварительная проверка подлинности на Microsoft Exchange Server (об этом мы подробнее поговорим далее). Для проверки подлинности и авторизации прокси-служба веб-приложения по-прежнему использует службы федерации Active Directory (AD FS) и доменные службы Active Directory (AD DS). Такая интеграция важна для использования личных устройств на работе, поскольку дает возможность создавать разные настраиваемые правила для пользователей, которые обращаются к ресурсам, находясь при этом в офисе, и для пользователей, работающих удаленно через Интернет.

Примечание. Если вы не знаете, как работает прокси-служба веб-приложения в Windows Server 2012 R2, ознакомьтесь со статьей по адресу http://technet.microsoft.com/library/dn584107.aspx.

Процедура установки прокси-службы веб-приложения такая же, как и в Windows Server 2012 R2, поэтому для установки в Windows Server 2016 нужно выполнить те же действия. После установки будет предложено выполнить настройку, как показано на рисунке 2-72.

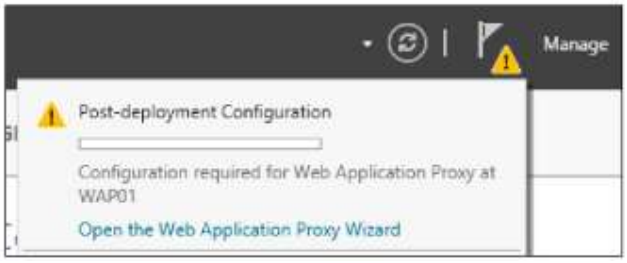


Рисунок 2-72. Настройка прокси-службы веб-приложения после развертывания

> **Примечание.** Перед развертыванием прокси-службы веб-приложения нужно спланировать инфраструктуру согласно рекомендациям, приведенным в статье по адресу http://technet.microsoft.com/library/dn383648.aspx. Данная статья была написана для Windows Server 2012 R2, но изложенные в ней рекомендации применимы и к Windows Server 2016.

По окончании действий, требуемых после развертывания (по сути, нужно подключить прокси-службу веб-приложения к серверу служб федерации Active Directory), можно использовать мастер публикации новых приложений. В новой версии мастер несколько изменен. Первое изменение, на которое следует обратить внимание: если нажать кнопку «Опубликовать» в инструменте управления прокси-службы веб-приложения, то в левой части окна появятся доступные настройки. Например, на странице «Предварительная проверка подлинности» в качестве метода проверки подлинности можно выбрать либо «Службы федерации Active Directory (AD FS), либо «Сквозной режим» (рисунок 2-73).

Рисунок 2-73. Выбор метода проверки подлинности

В этом примере выберите «Службы федерации Active Directory (AD FS)» и нажмите кнопку «Далее». На странице «Поддерживаемые клиенты» можно выбрать «Интернет и MSOFBA», «Обычная проверка подлинности HTTP» или OAuth2, как показано на рисунке 2-74.

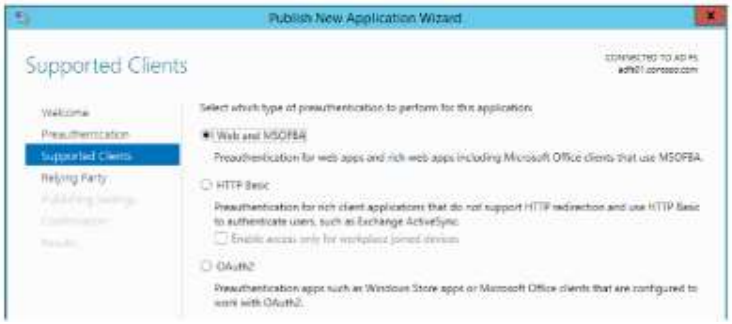


Рисунок 2-74. Поддерживаемые клиенты

Выберите «Интернет и MSOFBA», чтобы выполнять предварительную проверку подлинности клиентов с помощью протокола MSOFBA, который поддерживает проверку подлинности на основе форм вместо базовой проверки, или NTLM при использовании клиентских приложений Office. Второй вариант — хорошо известная базовая проверка подлинности HTTP, которую можно использовать, например, для Exchange Active Sync (ActiveSync), это новая возможность, реализованная в этой версии прокси-службы веб-приложения. В сценарии с ActiveSync процесс проверки подлинности включает четыре основных этапа:

1. Прокси-служба веб-приложения принимает запрос и передает все учетные данные в службу федерации Active Directory.
2. Служба AD FS проверяет данные, применяет политику и отправляет ответный маркер.
3. В случае успеха прокси-служба веб-приложения разрешает прохождение запроса к серверу Exchange Server.
4. Прокси-служба веб-приложения кэширует полученный маркер для дальнейшего использования.

Третий вариант — платформа авторизации OAuth2, применяющаяся во многих компаниях, в том числе и в корпорации Microsoft. Прокси-служба веб-приложения поддерживает OAuth2, начиная с Windows Server 2012 R2, но ранее этот вариант не был доступен для выбора в пользовательском интерфейсе.

> **Примечание.** Более подробная информация о протоколе OAuth2 доступна по адресу http://tools.ietf.org/html/rfc6749. Дополнительные сведения о поддержке OAuth2 в AD FS можно найти по адресу http://technet.microsoft.com/library/dn383640.aspx.

Выбрав нужный клиент для публикации, нажмите на кнопку «Далее». На странице «Параметры публикации» появился один новый параметр, с помощью которого можно включить перенаправление HTTP — HTTPS (рисунок 2-75).

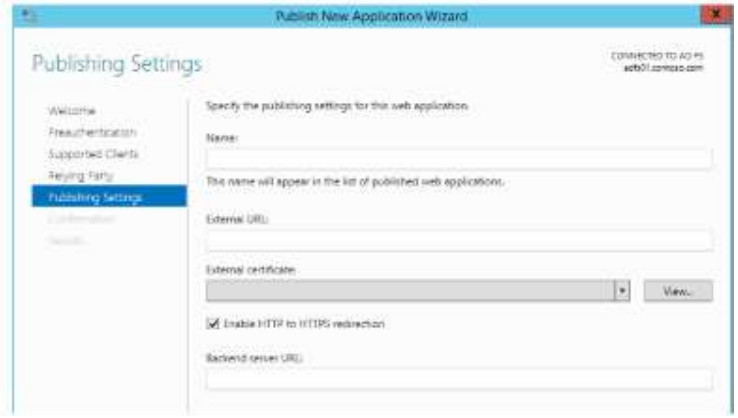


Рисунок 2-75. Параметры публикации

Это очень удобное дополнение, поскольку в Windows Server 2012 R2 для включения перенаправления HTTP — HTTPS нужно было устанавливать и настраивать Internet Information Services (IIS). Обратите внимание, что эта функция по умолчанию включена, но перед тем, как нажимать кнопку «Далее» и переходить на страницу подтверждения, убедитесь, что она выбрана и для вашего приложения.

Использование шлюза удаленных рабочих столов

В эту версию включены изменения, которые затрагивают способ публикации шлюза удаленных рабочих столов в прокси-службе веб-приложения. Эти новшества впервые появились в пакете обновления для Windows Server 2012 R2 в августе 2014 года, они упрощают развертывание ИТ-администраторам, планирующим опубликовать RDP через прокси-службу веб-приложения.

Шлюз удаленных рабочих столов может использовать файл cookie сеанса, использованный службой веб-доступа к удаленным рабочим столам для проверки подлинности RDP по протоколу HTTP.

Аудит доступа к ресурсам

В Windows Server 2016 реализована новая возможность, предоставляющая ИТ-администраторам более удобный доступ к аудиту опубликованных ресурсов. Чтобы проверить, существует ли уже данный заголовок XFF (X-Forwarded-For), прокси-служба веб-приложения добавляет этот заголовок к каждому запросу. Если существует, то прокси-служба веб-приложения сопоставляет с этим заголовком IP-адрес клиента.

> **Примечание.** XFF — нестандартный заголовок HTTP, который стал стандартом де-факто. Он широко используется прокси-серверами для идентификации IP-адреса источника запроса. Более подробная информация доступна в RFC по адресу http://tools.ietf.org/html/rfc7239.

Еще одна важная особенность аудита в прокси-службе веб-приложения — события, записываемые в программе просмотра событий. В новой версии в программу просмотра событий добавлено намного больше событий, в том числе аналитика и журналы отладки. С примерами этих событий можно ознакомиться ниже в этой главе в разделе «Устранение неполадок прокси-службы веб-приложения».

Прокси приложений в современной ИТ-среде

Несколько лет назад в нашей команде возникла непростая ситуация. В то время на рынке было два наших продукта — Forefront Threat Management Gateway и Forefront Unified Access Gateway. Оба эти продукта развивались в течение долгого времени после появления их первых версий в девяностые годы, выпускались уже много лет, ими пользовались десятки тысяч клиентов.

Но в обоих продуктах были схожие проблемы: эти продукты были очень сложные, работать с ними было очень трудно на всех этапах — от развертывания до устранения неполадок и обслуживания. Отчасти это объяснялось тем, что в ходе постепенного развития в продуктах накопилось очень много функций, ставших со временем ненужными. И в то же время не хватало поддержки современных технологий, таких как федерация или OAuth2, либо их поддержка была ограниченной. Кроме того, это были дорогостоящие продукты с собственными лицензиями.

Мы приняли непростое решение: решили начать с чистого листа, изучить всю функциональность обратных прокси-серверов и отобрать только актуальные технологии, действительно востребованные в современной ИТ-среде, а затем реализовать их, переписав код заново в соответствии с самыми современными стандартами. Одним из основных соображений в пользу этого решения было наше желание встроить в Windows Server обратный прокси-сервер. Мы стремились реализовать наше решение, подобное любым другим службам ролей, доступных для установки в диспетчере серверов. Это означало, что мы были обязаны соблюдать строжайшие стандарты в отношении кода и управления. Пользователи решений Microsoft ожидают, что управление всеми службами ролей в Windows Server осуществляется единообразно, в том числе с помощью Windows PowerShell, пользовательского интерфейса администратора, удаленного интерфейса администратора, счетчиков производительности, пакета System Center Operations Manager, журналов событий и т. п.

Именно так в составе Windows Server 2012 R2 появилась прокси-служба веб-приложения. Мы ни в чем не отступили от жестких требований безопасности кода, управления и стандартизации. Пользователи хорошо приняли наше решение. Компании с легкостью развертывали и встраивали прокси-службу веб-приложения в свою инфраструктуру.

Недостаток этого подхода состоит в том, что нам не удалось реализовать всю функциональность, которую мы хотели поддерживать, — все те функции, наличие которых дало бы

возможность всем пользователям перейти с Threat Management Gateway и Unified Access Gateway на новое решение. Впрочем, мы создали надежный базис, поэтому теперь добавлять дополнительные функции стало удобнее. Прокси-служба веб-приложения уже стала стандартом де-факто для доступа удаленных пользователей к ресурсам в локальной ИТ-среде, таким как Microsoft SharePoint, Lync и Exchange. Эта версия — важная веха на пути, по которому мы пошли много лет назад.

Настало время отправиться в новый путь и реализовать удаленный доступ к современной облачной среде. В качестве еще одного инструмента для публикации приложений в облачных решениях мы создали прокси приложений Azure Active Directory. К счастью, у прокси-службы веб-приложения Windows Server и у прокси приложения Azure Active Directory значительная часть кода общая. Более того, оба решения реализованы на базе одинаковых принципов и одинаковой идеологии удаленного доступа, простоты развертывания и удобства обслуживания. Мы продолжим разработку обоих продуктов. Кроме того, мы будем предлагать заказчикам Microsoft использование первой или второй архитектуры на выбор. В облаке пользователи получают уникальный и очень эффективный способ реализации удаленного доступа с помощью мощных функций и надежных средств безопасности Azure Active Directory без необходимости изменять сеть периметра. Ваши локальные приложения будет обслуживать та же самая служба, которая еженедельно обрабатывает 18 миллиардов запросов проверки подлинности.

Меир Менделович (Meir Mendelovich), старший руководитель программы

Публикация Exchange Server 2013

Как уже было сказано, после того, как была прекращена разработка Forefront Threat Management Gateway, многие администраторы Exchange оказались в затруднительном положении: требовался удобный способ публикации серверов Exchange Server в Интернете. Для решения этой задачи крупные организации могли использовать имеющуюся аппаратную инфраструктуру балансировки нагрузки, но для управления балансировщиками нагрузки в компаниях малого и среднего бизнеса обычно оказывается недостаточно средств и квалифицированных сотрудников. Именно в таких случаях сервер веб-приложений может оказаться весьма полезным.

Базовые принципы публикации Exchange 2013 Outlook Web App и Центра администрирования Exchange с помощью прокси-службы веб-приложения подробно описаны по адресу http://technet.microsoft.com/library/dn635116(v=exchg.150).aspx. Чтобы лучше понять некоторые возможности прокси-службы веб-приложения в Windows Server 2016, рассмотрим очень простой сценарий, показанный на рисунке 2-76.

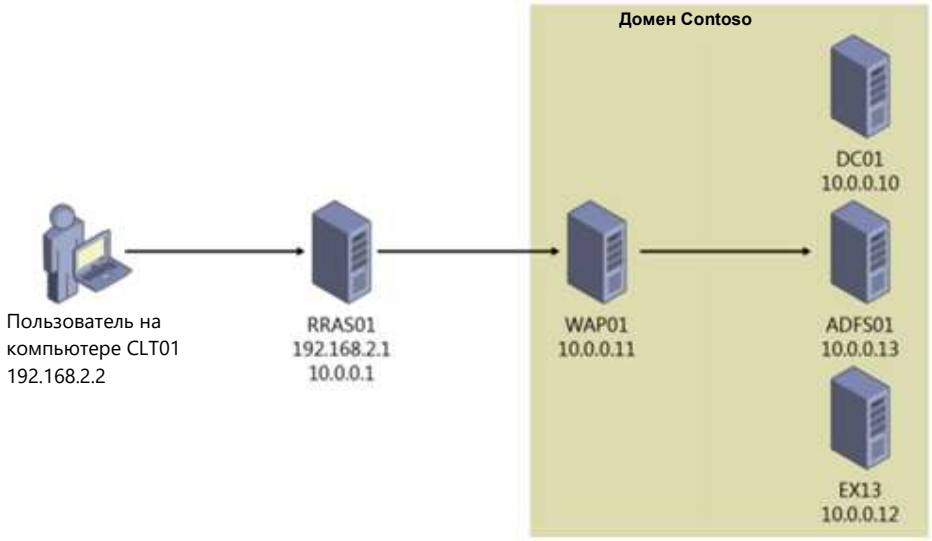


Рисунок 2-76. Сценарий, демонстрирующий прокси-службу веб-приложения в Windows Server 2016.

В этом сценарии пользователь на компьютере CLT01, не присоединенном к домену, устанавливает подключение к Outlook Web App по URL-адресу https://mail.contoso.com/owa. Запрос пользователя передается по внешней сети на сервер маршрутизации и удаленного доступа (RRAS01). Этот сервер выполняет внешнее разрешение DNS для зоны contoso.com и направляет трафик внешних пользователей во внутреннюю сеть Contoso. Сервер RRAS01 направляет запрос, высланный на адрес https://mail.contoso.com/owa, во внутреннюю сеть Contoso к прокси-службе веб-приложения (WAP01), работающей под управлением Windows Server 2016.

На сервере WAP01 опубликовано приложение Outlook Web App с использованием предварительной проверки подлинности AD FS. Отметим, что через этот сервер опубликована не одна, а целый ряд различных служб Exchange, использующих как предварительную, так и сквозную проверку подлинности AD FS, как показано на рисунке 2-77.

Рисунок 2-77. Опубликованные веб-приложения.

В этом случае мы исходим из того, что используется разделенная конфигурация DNS, т. е. внутренний и внешний DNS-серверы разрешают имя mail.contoso.com в разные IP-адреса в зависимости от расположения пользователя. Таким образом, значения «Внешний URL-адрес» и «URL-адрес внутреннего сервера» в этом примере одинаковы, но они могут быть и разными, как будет проиллюстрировано ниже. Итак, когда пользователь, находящийся во внутренней сети Contoso, переходит по адресу https://mail.contoso.com/owa, используется встроенная проверка подлинности Windows. Для этого URL-адреса mail.contoso.com и adfs.contoso.com должны быть перечислены в списке «Доверенная зона» местной интрасети в Internet Explorer. В этом случае пользователь сможет подключаться к своему почтовому ящику, причем при подключении у пользователя учетные данные запрашиваться вообще не будут .

Если же пользователь находится вне корпоративной сети, то при попытке подключения он будет переходить на веб-страницу проверки подлинности на основе форм (рисунок 2-78), на которой потребуется указать учетные данные для входа.

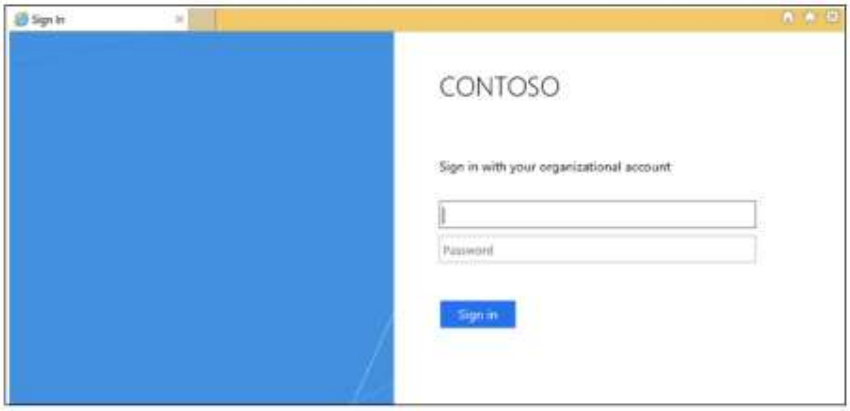


Рисунок 2-78. Страница проверки подлинности на основе форм.

Впрочем, отсутствует одна очень важная служба, широко использующаяся практически во всех организациях, — Microsoft Server ActiveSync. Есть возможность определить для ActiveSync отношение доверия с проверяющей стороной и настроить сквозную проверку подлинности;

в прокси-службе веб-приложения в Windows Server 2012 R2 только так и можно было сделать. Но, как мы уже говорили в этой главе, прокси-служба веб-приложения в Windows Server 2016 теперь поддерживает клиентов, использующих обычную проверку подлинности HTTP (HTTP Basic) для служб, подобных ActiveSync, не поддерживающих перенаправление и использующих аутентификацию HTTP Basic.

HTTP Basic — это метод проверки подлинности, применяемый многими протоколами, включая службу ActiveSync, используемую для подключения полнофункциональных клиентов и смартфонов к почтовым ящикам Exchange (дополнительные сведения об обычной проверке подлинности HTTP содержатся в RFC 2617 по адресу http://www.ietf.org/rfc/rfc2617.txt). Прокси-служба веб-приложения обычно взаимодействует с AD FS с помощью перенаправлений, которые не поддерживаются клиентами ActiveSync. Если опубликовать приложение с помощью обычной проверки подлинности HTTP, то будет обеспечена поддержка клиентов ActiveSync в прокси-службе веб-приложения, поскольку HTTP-приложение сможет получить отношение доверия с проверяющей стороной для использования AD FS.Процесс проверки подлинности для клиентов, применяющих HTTP Basic, состоит из следующих шагов:

1. Пользователь пытается получить доступ к опубликованному веб-приложению из клиентского приложения, установленного на телефоне.
2. Приложение отправляет HTTPS-запрос по URL-адресу, опубликованному прокси-службой веб-приложения.
3. Если запрос не содержит учетных данных, прокси-служба веб-приложения возвращает приложению отклик «HTTP 401» с URL-адресом сервера AD FS, отвечающего за проверку подлинности.
4. Пользователь снова отправляет HTTPS-запрос приложению, при этом в заголовке запроса www-authenticate указывается тип авторизации Basic, а также имя пользователя и пароль в кодировке Base64.
5. Поскольку перенаправить устройство непосредственно к AD FS невозможно, прокси-служба веб-приложения отправляет к AD FS запрос проверки подлинности, указывая имеющиеся учетные данные: имя пользователя, пароль, а также сертификат устройства (если он есть). Прокси-служба от имени устройства получает маркер проверки.
6. Чтобы ограничить количество запросов, отправляемых к AD FS, прокси-служба веб-приложения подтверждает последующие запросы клиента, используя кэшированные маркеры, пока они действительны. Прокси-служба веб-приложения периодически очищает кэш. Его размер можно узнать с помощью счетчика производительности.
7. Если маркер проверки действителен, то прокси-служба веб-приложения перенаправляет запрос на сервер, реализующий службу, и пользователь получает доступ к опубликованному веб-приложению.

Чтобы проделать эту последовательность шагов, нужно вернуться на сервер ADFS01 и создать для ActiveSync проверяющую сторону, которая не требует утверждений, как показано на рисунке 2-79.

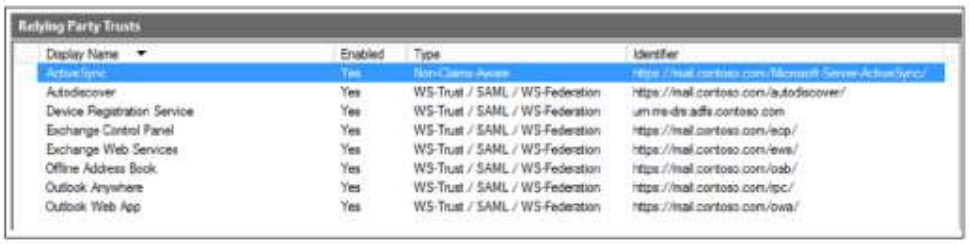


Рисунок 2-79. Доверие проверяющей стороны.

Далее нужно перейти на сервер WAP01, опубликовать с помощью HTTP Basic приложение ActiveSync, а затем выбрать проверяющую сторону ActiveSync, не требующую использования утверждений, как показано на рисунке 2-80.

Рисунок 2-80. Поддерживаемые клиенты

Обратите внимание, что эта проверяющая сторона отображается лишь в том случае, если она была определена на сервере AD FS, как показано на рисунке 2-81.

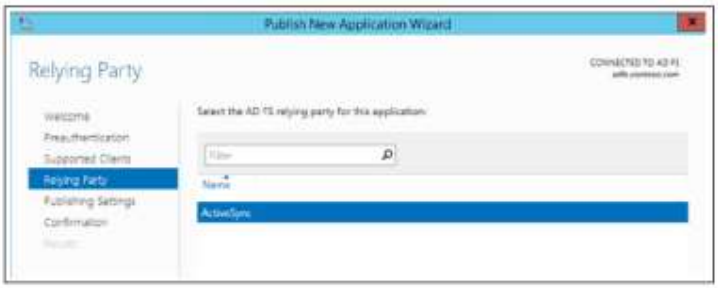


Рисунок 2-81. Параметры проверяющей стороны.

Выполните оставшиеся инструкции мастера, настроив внешний URL-адрес и URL-адрес сервера, реализующего службу. Конечный результат показан на рисунке 2-82.

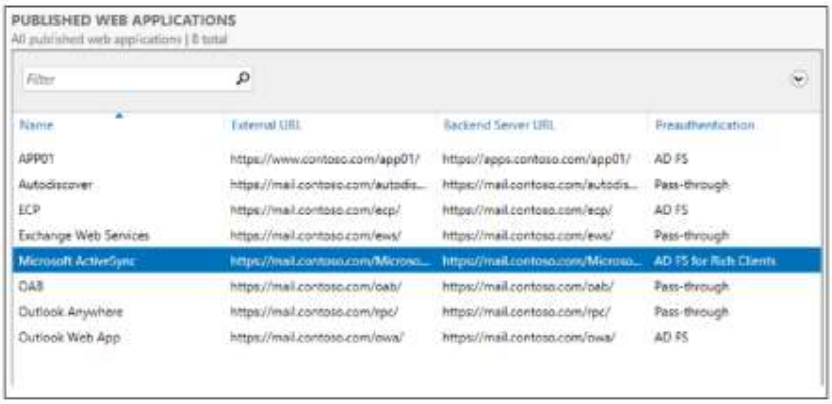


Рисунок 2-82. Все опубликованные веб-приложения.

Обратите внимание, что опубликованное приложение Microsoft ActiveSync использует метод предварительной проверки подлинности AD FS для клиентских приложений.

Определение утверждений

Определение утверждений не является функцией прокси-службы веб-приложения в Windows Server 2016, тем не менее, важно понимать место и значение утверждений в транзакциях. Утверждения определяются в разделе Outlook Web App в области «Действия» на сервере AD FS, как показано на рисунке 2-83.

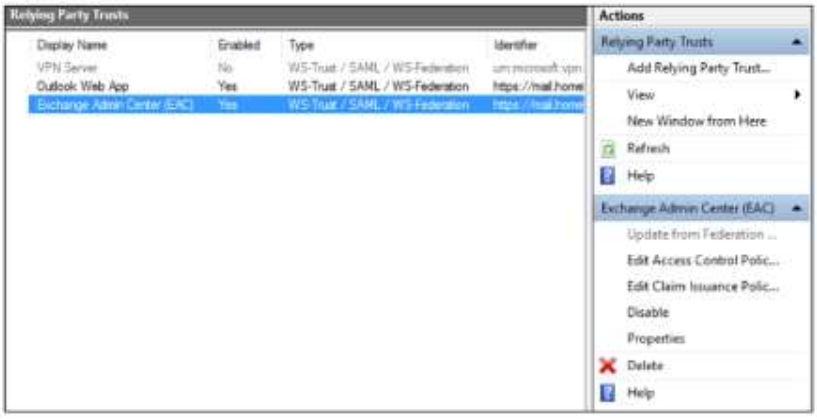


Рисунок 2-83. Изменение правил утверждений.

Выберите отношение доверия с проверяющей стороной, для которого нужно определить утверждения, а затем в области «Действия» нажмите «Изменить утверждения».

В модели удостоверений на основе утверждений служба федерации Active Directory выдает маркер, содержащий набор утверждений. Правила утверждений определяют решения службы AD FS, принимаемые в отношении утверждений. Правила утверждений и все данные конфигурации сервера хранятся в базе данных конфигурации AD FS.

Чтобы опубликовать Outlook Web App и Центр администрирования Exchange, нужно создать три настраиваемых правила утверждений:

- SID пользователя Active Directory
- SID группы Active Directory
- Имя участника-пользователя (UPN) Active Directory

При настройке правил утверждений необходимо использовать соответствующий синтаксис языка. В частности, для правила ActiveDirectoryUserSID используйте следующий код:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname".
   Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory",
   types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"),
   query = ";objectSID;{0}", param = c.Value);
```

Получившееся правило содержит имя и текст правила, как показано на рисунке 2-84.

Рисунок 2-84. Редактирование правила утверждения.

Затем настройте следующее правило утверждения ActiveDirectoryGroupSID:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory",
   types = ("http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid"),
   query = ";tokenGroups(SID);{0}", param = c.Value);
```

Наконец, настройте еще одно правило утверждения ActiveDirectoryUPN:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"]=> issue(store = "Active Directory",
   types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"),
   query = ";userPrincipalName;{0}", param = c.Value);
```

По завершении нажмите кнопку «Применить», а затем кнопку OK. Названия правил преобразования отображаются на вкладке «Правила преобразования выдачи» в окне «Изменение правил утверждений», как показано на рисунке 2-85.

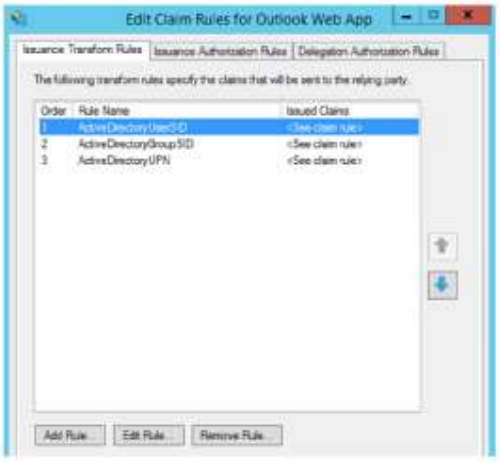


Рисунок 2-85. Окно «Изменение правил утверждений».

> **Примечание.** Эту последовательность потребуется проделать для каждого отношения доверия проверяющей стороны.

Преобразование имен узлов в URL-адресах

Следует помнить о том, что прокси-служба веб-приложения способна преобразовывать имена узлов в URL-адресах. Например, для доступа к какому-либо приложению внешние пользователи могут использовать URL-адрес http://www.contoso.com/app01, тогда как для внутренних пользователей это же приложение будет доступно по адресу http://apps.contoso.com/app01. Это вполне приемлемо, прокси-служба веб-приложения учитывает различия в URL-адресах, как показано на рисунке 2-86: на этом рисунке внешний URL-адрес — http://www.contoso.com/app01, а URL-адрес внутреннего сервера — http://apps.contoso.com/app01.

Рисунок 2-86. Опубликованные веб-приложения.

> **Примечание.** Невозможно изменить путь таким образом, чтобы для внешних пользователей использовался адрес http://www.contoso.com/app1, а для внутренних — http://apps.contoso.com/applicationXYZ.

Это важно помнить при публикации Exchange, поскольку для внутреннего и внешнего доступа могут использоваться разные пространства имен DNS. Для Outlook Web App целесообразно опубликовать разные URL-адреса для внутренних пользователей и для внешнего доступа. На рисунке 2-87 показано, что прокси-служба веб-приложения поддерживает такой сценарий, если используется одинаковый путь.

Рисунок 2-87. Параметры публикации приложения

Чтобы разрешить такое преобразование, нужно сначала получить идентификатор приложения, для которого требуется использовать преобразование. Для этого можно использовать следующую команду Windows PowerShell:

```
Get-WebApplicationProxyApplication | Format-Table ID, Name, ExternalURL
```

На рисунке 2-88 показаны выходные данные после выполнения этой команды.

Рисунок 2-88. Выходные данные, полученные в результате выполнения команды Get-WebApplicationProxyApplication.

После этого нужно взять идентификатор приложения из данных, показанных на рисунке 2-88, и ввести следующую команду Windows PowerShell:

```
Set-WebApplicationProxyApplication -ID <ИД приложения> -DisableTranslateUrlInRequestHeaders:$false
```

Включение AD FS для организации Exchange

При настройке AD FS для проверки подлинности на основе утверждений с Outlook Web App и Центром администрирования Exchange в Exchange 2013 необходимо включить AD FS для вашей организации Exchange. Для этого можно использовать командлет Set-OrganizationConfig. В примере среды, описываемой в этой главе, потребовалось бы сделать следующее:

- Задать для издателя AD FS адрес https://adfs.contoso.com/adfs/ls.
- Задать URI службы AD FS https://mail.contoso.com/owa и https://mail.contoso.com/ecp.
- Найти отпечаток сертификата для подписи маркеров AD FS с помощью командлета Windows PowerShell Get- ADFSCertificate -CertificateType "Token-signing" на сервере AD FS и затем назначить найденный сертификат для подписи маркеров.

В командной консоли Exchange Shell введите следующий код:

```
Get-ADFSCertificate -CertificateType "Token-signing
```

При этом вы получите отпечаток сертификата для подписи маркеров, с использованием которого можно запустить следующие командлеты Set-OrganizationConfig:

```
$uris = @("https://mail.contoso.com/owa","https://mail.contoso.com/ecp")

Set-OrganizationConfig -AdfsIssuer "https://adfs.contoso.com/adfs/ls/" -AdfsAudienceUris $uris
-AdfsSignCertificateThumbprint "1a2b3c4d5e6f7g8h9i10j11k12l13m14n15o16p17q"
```

Устранение неполадок прокси-службы веб-приложения

В следующих разделах приводятся советы по устранению возможных неполадок, которые могут возникнуть в среде, где развернута прокси-служба веб-приложения.

Сбор информации о среде

Для управления прокси-службами веб-приложения и для устранения неполадок необходимо хорошо знать Windows PowerShell и, в частности, командлеты, относящиеся к прокси-службе веб-приложения. При устранении неполадок в прокси-службе веб-приложения прежде всего ознакомьтесь со всеми сообщениями об ошибках, которые отображаются в консоли. Если нет очевидных ошибок, проверьте журналы событий. Для этого можно войти на каждый сервер и просмотреть журналы событий, но можно поступить проще — использовать Windows PowerShell.

К примеру, следующая команда Windows PowerShell собирает все события, созданные прокси-службой веб-приложения за последние 24 часа (для каждого события указывается идентификатор, уровень и сообщение):

```
$yesterday = (Get-Date) - (New-TimeSpan -Day 1) ;
Get-WinEvent -FilterHashTable @{LogName='Microsoft-Windows-WebApplicationProxy/Admin';
StartTime=$yesterday}
| group -Property ID,LevelDisplayName,Message -NoElement |
sort Count, Name -Descending | ft -Name -HideTableHeaders }
```

Предположим, что на одном из серверов периодически возникает событие с идентификатором 12000; при этом вы используете несколько серверов с прокси-службой веб-приложения и хотите проверить, возникает ли такая же ошибка на всех серверах. Выполните следующую команду, чтобы собрать все события с идентификатором 12000, созданные за последние 10 часов, для набора серверов с прокси-службой веб-приложения:

```
Foreach ($Server in (gwpc).ConnectedServersName) {Get-WinEvent -FilterHashTable
@{LogName='Microsoft-Windows-WebApplicationProxy/Admin'; ID=12000; StartTime=(Get-Date) -
(New-TimeSpan -hour 10)} -ComputerName $Server -ErrorAction SilentlyContinue | group MachineName
-NoElement | ft Name -HideTableHeaders }
```

Теперь у нас имеется список всех серверов, на которых возникает такая проблема. В нашем примере представим, что такая ошибка возникает только на одном сервере.

Очень полезной для устранения проблемы может оказаться таблица кодов ошибок, доступная на портале TechNet (http://technet.microsoft.com/en-us/library/dn770156.aspx). В ней рекомендуется проверить подключение этого конкретного сервера AD FS к прокси-службе веб-приложения. Для этого перейдите по адресу https://<полное_доменное_имя_прокси_AD_FS>/FederationMetadata/2007-06/FederationMetadata.xml и убедитесь в наличии отношений доверия между сервером AD FS и сервером с прокси-службой веб-приложения. Если отношения не работают, исправьте неполадку с помощью командлета Install-WebApplicationProxy.

Использование анализатора соответствия рекомендациям Microsoft Exchange

На сервере с прокси-службой веб-приложения можно запустить анализатор соответствия рекомендациям Exchange. Для этого можно использовать графическую оболочку диспетчера серверов. В области слева выберите «Локальный сервер», затем прокрутите вниз, пока на средней панели не появится «Анализатор соответствия рекомендациям», как показано на рисунке 2-89.

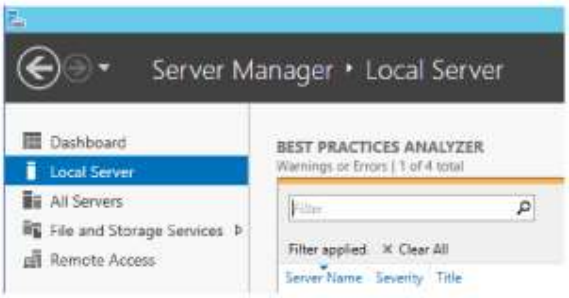


Рисунок 2-89. Анализатор соответствия рекомендациям в диспетчере серверов.

Справа в диспетчере серверов нажмите «Задачи», затем выберите «Начать проверку BPA», как показано на рисунке 2-90.

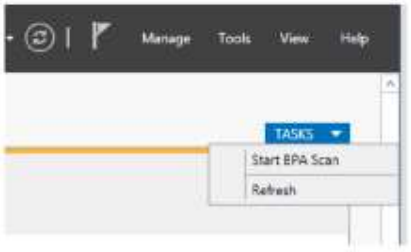


Рисунок 2-90. Запуск проверки BPA.

Также можно запустить анализатор соответствия рекомендациям с помощью следующего командлета Windows PowerShell:

```
Invoke-BpaModel Microsoft/Windows/RemoteAccessServer
Get-BpaResult Microsoft/Windows/RemoteAccessServer
```

В этом случае неполадка обусловлена проблемами с сертификатом (рисунок 2-91), поэтому отобразится сообщение об ошибке со следующим текстом: «Прокси-службе веб-приложения не удалось опубликовать приложение из-за проблем с сертификатом».

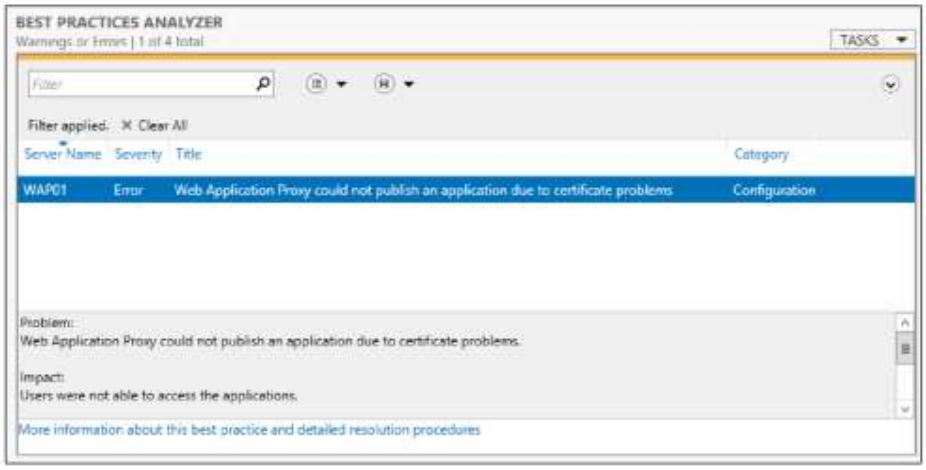


Рисунок 2-91. Просмотр результатов анализатора соответствия рекомендациям.

Событие, указанное в окне анализатора соответствия рекомендациям, содержит информацию об устранении неполадки, как показано на рисунке 2-92.

Рисунок 2-92. Просмотр подробной информации в анализаторе соответствия рекомендациям.

В таблице на портале TechNet для события 12021 приведены следующие рекомендации:

Убедитесь, что отпечатки сертификата, настроенные для приложений прокси-службы веб-приложения, установлены на всех серверах с прокси-службой веб-приложения с закрытым ключом в хранилище локального компьютера.

Располагая этой информацией, можно проверить сертификаты на сервере с прокси-службой веб-приложения и убедиться в правильности имен и сроков действия, а также в том, что их отпечаток соответствует отпечатку на сервере. Затем можно проверить сертификаты на сервере и убедиться в их правильности; в случае возникновения ошибок следует выпустить сертификаты заново.

Неполадки, связанные с сертификатами

Сертификаты имеют большое значение для служб федерации Active Directory и для функционирования прокси-службы веб-приложений, поэтому для правильной ее работы с AD FS крайне важно получить правильные сертификаты — с правильными именами в сертификатах на соответствующих серверах.

Неполадки с сертификатами могут привести к получению таких вот сообщений об ошибках:

```
Сертификат доверия ("ADFS ProxyTrust - WAP01") недействителен.
```

Причин у этой проблемы может быть несколько:

- Возможны какие-либо перебои в сети между сервером с прокси-службой веб-приложения и сервером AD FS.
- Сервер с прокси-службой веб-приложения мог быть выключен в течение длительного времени.
- Возможны неполадки при проверке сертификата из-за сбоев в инфраструктуре центра сертификации.
- Неполадки синхронизации времени между прокси-службой веб-приложения и сервером AD FS могут привести к потере синхронизации.

Для устранения этих проблем проверьте текущее время на прокси-службе веб-приложения и на сервере AD FS, а затем заново запустите командлеты Install-WebApplicationProxy.

Данные конфигурации в AD FS не согласованы или повреждены

Также могут возникать ошибки, связанные с данными конфигурации в AD FS: такие данные могут либо отсутствовать, либо оказаться непригодными для сервера с прокси-службой веб-приложения. Могут отображаться следующие сообщения об ошибках:

```
Данные конфигурации не найдены в AD FS.
```

или:

```
Данные конфигурации, хранящиеся в AD FS, повреждены, или прокси-службе веб-приложения не удалось разобрать их.
```

или:

```
Прокси-службе веб-приложения не удалось получить список проверяющих сторон из AD FS.
```

Эти ошибки могут быть вызваны различными причинами. Возможно, прокси-служба веб-приложения не была полностью установлена и настроена, либо изменения в базе данных AD FS привели к повреждению данных. Также возможна ситуация, когда из-за неполадок в сети не удается связаться с сервером AD FS, поэтому база данных AD FS недоступна.

Устранить подобные ошибки можно разными способами:

- Еще раз запустите командлет Install-WebApplicationProxy, чтобы устранить неполадки настройки.
- Убедитесь в наличии действующего сетевого подключения между сервером с прокси-службой веб-приложения и сервером AD FS.
- Убедитесь, что на сервере с прокси-службой веб-приложения запущена служба WebApplicationProxy.

Поддержка клиентов, не использующих SNI

Указание имени сервера (SNI) — это функция протокола SSL/TLS, используемая на сервере с прокси-службой веб-приложения и на сервере AD FS с целью снижения требований к сетевой инфраструктуре. Обычно сертификат SSL соответствует сочетанию IP-адреса и номера порта. Из этого следует, что если нужно привязать другой сертификат к этому же номеру порта на сервере, придется настраивать еще один IP-адрес. При использовании SNI сертификат привязывается не к IP-адресу, а к имени узла и к номеру порта, что позволяет экономить IP-адреса и упростить настройку.

Важно понимать, что возможность использования SNI всецело зависит от того, поддерживает ли эту функцию запрашивающий клиент. Если Hello-пакет клиента SSL не содержит заголовок SNI, то http.sys не сможет определить, какой сертификат следует выдать клиенту, и поэтому сбросит подключение.

Большинство современных клиентов поддерживают SNI, но с некоторыми клиентами часто возникают проблемы. Устаревшие браузеры, операционные системы, аппаратные средства балансировки нагрузки, средства проверки работоспособности, устаревшие версии WebDAV, ActiveSync в Android, а также некоторые устаревшие VoIP-устройства для конференций могут не поддерживать SNI.

Чтобы обеспечить поддержку клиентов, не совместимых со SNI, проще всего нужно создать привязку резервного сертификата в http.sys. Резервный сертификат должен включать все полные доменные имена (FQDN), которые требуется поддерживать, в том числе полное доменное имя самой службы AD FS (adfs.contoso.com), полные доменные имена всех приложений, опубликованных при помощи прокси-службы веб-приложения (mail.contoso.com), а также полное доменное имя Enterprise Registration (enterpriseregsitration.contoso.com), если используется присоединение к рабочему месту Workplace Join.

После создания сертификата нужно получить отпечаток GUID приложения и сертификата. Для этого можно использовать следующий командлет Windows PowerShell:

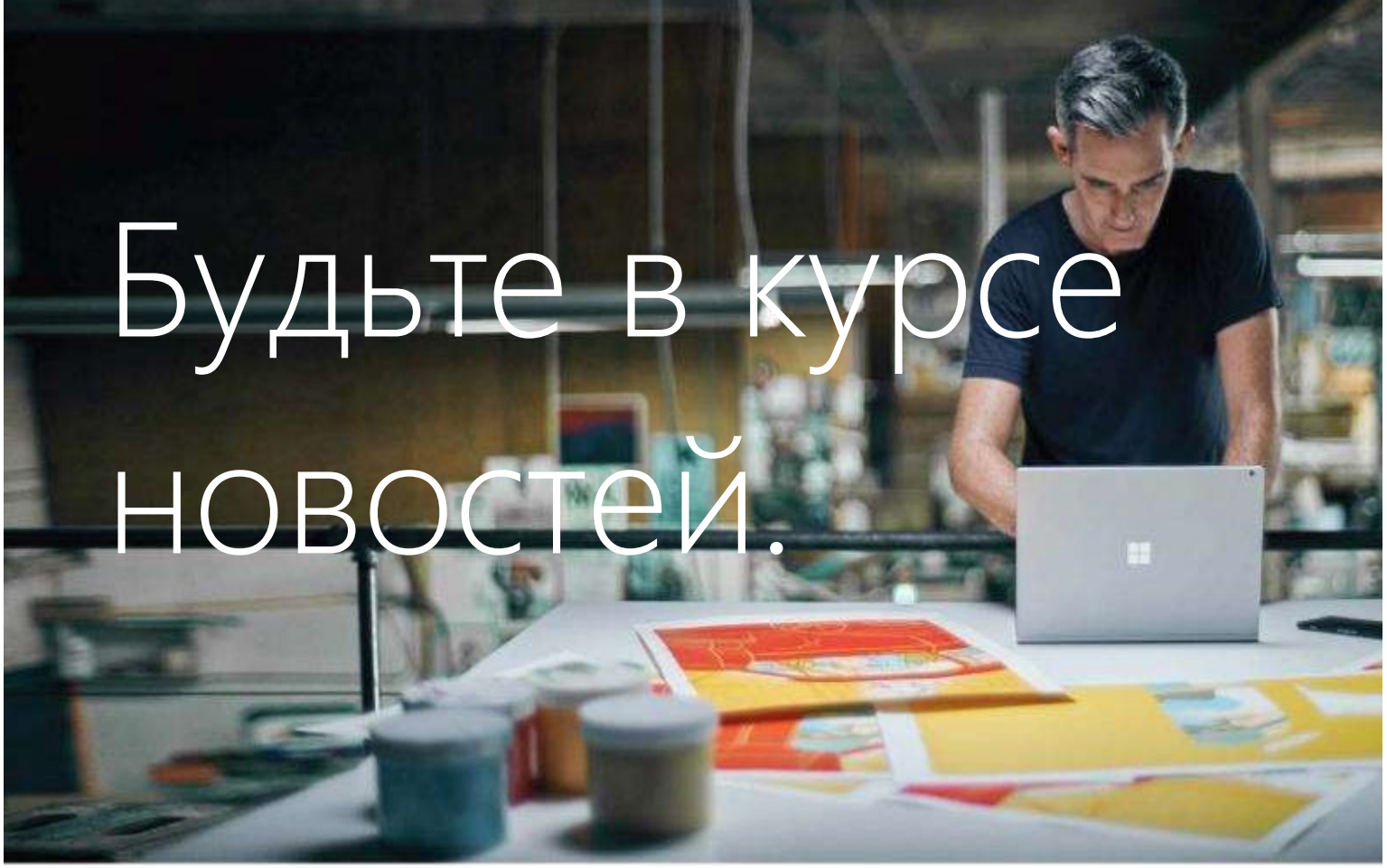
```
Get-WebApplicationProxyApplication | fl Name,ExternalURL,ExternalCertificateThumbprint
```

Теперь, когда имеется отпечаток GUID и сертификата, его можно привязать к шаблону IP-адресов и к порту 443, используя следующий синтаксис:

```
netsh http add sslcert ipport=0.0.0.0:443 certhash=certthumprint appid={applicationguid}
```

Обратите внимание, что выполнить эту команду потребуется на каждом сервере фермы AD FS, а также на сервере с прокси-службой веб-приложения.

> **Примечание.** Технические подробности о SNI, как о подмножестве расширений TLS, доступны по адресу https://tools.ietf.org/html/rfc3546#section-3.1.



Будьте в курсе новостей.

Получайте последние новости
от издательства Microsoft Press.

- Новые и будущие книги
- Специальные предложения
- Бесплатные электронные книги
- Учебные статьи

Подпишитесь по адресу MicrosoftPressStore.com/Newsletters

Microsoft

Платформа приложений

В этой главе мы расскажем о надежной платформе Microsoft для приложений, которые могут быть размещены как в общедоступном, так и в частном облаке. Мы обсудим две новые технологии, которые появились в Microsoft Windows Server 2016: *Nano Server* и *контейнеры*. Эти технологии помогут вам получить максимум выгоды от использования оптимизированной, масштабируемой и безопасной платформы приложений.

Модернизация традиционных приложений

Облачные решения дают компаниям возможность быстрее внедрять инновации и ускорять их окупаемость за счет использования облачных приложений и архитектуры микрослужб. Многие компании, планирующие воспользоваться с новыми возможностями, сталкиваются с непростой задачей: как обеспечить управление тысячами приложений и обновлений? Требуется решение, способное улучшить работу существующих приложений и позволяющее создавать новые, изначально предназначенные для облачной среды. Windows Server 2016 позволяет делать и то, и другое.

Windows Server помогает обеспечить информационную безопасность и обновление существующих корпоративных серверных приложений практически без изменения кода, упаковывать приложения в контейнеры; воспользоваться преимуществами гибкой разработки по методике DevOps; развертывать упакованные приложения в локальной среде и в любых облаках, в том числе гибридных. Разработчики могут создавать оптимизированные для облачной среды приложения и службы с помощью таких технологий, как контейнеры и Nano Server.

Windows Server 2016 позволяет модернизировать приложения и быстрее внедрять инновации, используя в качестве основы мощную платформу приложений, полностью приспособленную к облачным средам. На рис. 3-1 показаны ключевые области применения доступных в Windows Server 2016 технологий, предназначенных для подготовки существующих приложений к работе в облачной среде или для разработки новых облачных приложений.

Рис. 3-1. Три составляющие платформы приложений

Во-первых, переход на Windows Server 2016 позволит практически сразу же повысить безопасность серверной структуры, обеспечив защиту виртуальных машин и расширив возможности имеющихся приложений. Этого можно добиться, воспользовавшись усовершенствованными функциями безопасности и управления. В частности, для защиты важных приложений можно использовать экранированные виртуальные машины, дающие возможность запускать приложения только в доверенной структуре. Кроме того, можно ограничить доступ администраторов к определенным задачам: поддерживается как ограничение доступных возможностей (Just Enough Administration, JEA), так и ограничение времени, в течение которого предоставляется доступ (Just-in-Time, JIT).

> **Примечание.** Дополнительные сведения об экранированных виртуальных машинах, а также об администрировании с ограничениями JEA и JIT приведены в главе 4.

Во-вторых, есть возможность поместить существующие приложения в контейнеры и перенести традиционные приложения в современную среду разработки и эксплуатации практически без изменения кода. При этом вы получите однородность при разработке, тестировании и использовании за счет применения одинаковых инструментов. Также поддерживаются ускоренное развертывание, непрерывная интеграция и доставка приложений, при этом обеспечивается более высокий, чем прежде, уровень информационной безопасности. Для лучшего управления и поддержания целостности можно использовать контейнеры: одни и те же приложения можно развертывать в локальной среде, в любых облаках и в гибридной архитектуре. Если требуется более высокий уровень изоляции, приложение можно развернуть в контейнере Hyper-V: в этот контейнер упаковывается точно такой же образ, а низкоуровневая оболочка (гипервизор) обеспечивает дополнительную изоляцию.

В-третьих, можно создавать облачные и гибридные приложения. Операционная система Windows Server 2016 адаптирована для использования подходов Agile при создании облачных приложений на основе архитектуры микрослужб. Модель развертывания Nano Server дает возможность создавать в автономном режиме настраиваемые образы операционной системы, глубоко оптимизированные для конкретных задач и приложений. Nano Server — это сверхкомпактная и очень быстро загружающаяся операционная система, к числу ее преимуществ относятся более высокая плотность и ограничение числа уязвимостей.

Корпорация Microsoft стремится упростить создание наилучших приложений на любых платформах, будь то Microsoft .Net Framework или платформы с открытым исходным кодом, такие как .Net Core и Node.JS. Используя надежную технологию Microsoft Azure Service Fabric, вы можете разрабатывать надежные и масштабируемые распределенные приложения и запускать их в Azure, в локальной и гибридной среде. Кроме того, есть возможность сочетать преимущества контейнеров, Nano Server, Service Fabric и надежной платформы Windows Server для повышения гибкости бизнеса с помощью облачных приложений.

Выбор тех или иных технологий зависит от потребностей каждого заказчика и от особенностей разрабатываемых приложений. В Windows Server 2016 поддерживается множество возможностей, позволяющих с минимальными затратами перейти к современной облачной инфраструктуре.

Микрослужбы

В том, что касается приложений, предназначенных для веб-доступа, мы отошли от традиционной многоуровневой архитектуры в сторону сервис-ориентированной архитектуры (SOA). Это было весьма непросто, и многим заказчикам пришлось переписывать свои приложения. При использовании SOA приложение разделяется на компоненты, обменивающиеся данными посредством какого-либо протокола.

Можно сказать, что архитектура SOA была прообразом микрослужб: при использовании микрослужб приложение разделяется на еще более мелкие компоненты, каждый из которых существует и работает в виде отдельного процесса и обменивается информацией с другими компонентами вне зависимости от использованного языка программирования.

Микрослужбы дают возможность ускорить разработку по сравнению с SOA. Это обусловлено тем, что компоненты, задействованные в микрослужбах, гораздо компактнее, чем в SOA. Если нужно изменить один из компонентов микрослужб, то есть возможность быстро разработать, обновить и развернуть только этот компонент, не затрагивая работу других. Каждый компонент с технической точки зрения является независимым «контрагентом», работает по-своему и поддерживает свой способ обмена информацией. Все компоненты используют общую модель обмена данными, что позволяет с легкостью вносить изменения в отдельные части приложения, созданного на основе микрослужб.

Service Fabric — это распределенная системная платформа, упрощающая создание микрослужб и преобразование приложений в архитектуру микрослужб. В состав платформы входят средства для управления всем жизненным циклом приложения. Платформа доступна как в локальной среде, так и в Azure в виде Azure Service Fabric — достаточно один раз написать приложение, чтобы развернуть его как в локальной среде, так и в Azure без изменения используемых API-интерфейсов, применяя распространенные средства разработки, такие как Microsoft Visual Studio.

На основе платформы Service Fabric работают многие современные службы Microsoft: СУБД Azure SQL, Azure DocumentDB, Cortana, Power BI, Intune, концентраторы событий Azure, службы Интернета вещей для Azure, Skype для бизнеса и многие другие важные службы Azure. При создании продукта Service Fabric были учтены результаты анализа использования перечисленных решений. Платформа микрослужб — это оптимальный выбор, если вам требуется исключительно надежное и масштабируемое решение для создания и развертывания приложений.

Примечание. Дополнительные сведения о возможностях Service Fabric доступны по адресу https://azure.microsoft.com/documentation/articles/service-fabric-overview/.

Программа преимуществ гибридного использования Azure

Как мы уже отметили выше, платформа Windows Server 2016 очень удобна в том, что касается перехода на облачные вычисления. Для поддержки такого перехода корпорация Microsoft разработала программу лицензирования, позволяющую в полной мере воспользоваться преимуществами локальных лицензий и использовать их в Azure, чтобы контролировать затраты при использовании программ в общедоступном облаке.

Лицензионная программа преимуществ гибридного использования Azure (Azure Hybrid Use Benefit, AHUB) предоставляет заказчикам, использующим Windows Server с подпиской Software Assurance, возможность использования этих лицензий в Azure. Для виртуальных машин Windows Server, запущенных в Azure, действует льготная тарифная ставка. Проще говоря, заказчик оплачивает лишь базовый вычислительный тариф, и экономия по сравнению с оплатой работы виртуальных машин по тарифу D2 может достигать 41%.

Примечание. Дополнительные сведения о лицензировании AHUB доступны по адресу http://azure.microsoft.com/pricing/hybrid-use-benefit.

Nano Server

Nano Server — это новый удобный вариант установки Windows Server 2016. В этом случае системе требуется еще меньше места, чем при установке только основных серверных компонентов (Server Core).

Знакомство с Nano Server

Nano Server — это новый сверхкомпактный вариант установки Windows Server 2016, представляющий собой глубоко переработанный Windows Server, оптимизированный для использования в облачной среде. Nano Server в составе Windows Server 2016 отлично подходит для следующих сценариев использования:

- вычислительный узел для Hyper-V или часть отказоустойчивого кластера Windows;
- узел контейнеров;
- узел хранилища для масштабируемого файлового сервера (Scale-Out File Server, SOFS);
- DNS-сервер;
- веб-сервер с IIS;
- Платформа для развертывания приложений, созданных на основе облачных методик разработки и запускаемых в контейнере и/или в гостевой виртуальной машине.

Nano Server не имеет графического интерфейса, поэтому организациям, которые не используют возможности удаленного управления своими серверами, потребуется определенным образом пересмотреть процедуры управления и эксплуатации.

В свое время пользователи Windows Server выдвигали следующие нарекания:

- Перезагрузки отрицательно влияют на мой бизнес. Почему мне приходится перезагружать сервер из-за установки исправления для компонента, который я никогда не использую?
- Когда необходима перезагрузка, нужно, чтобы серверы возобновляли работу как можно скорее.
- Большие образы серверов долго развертываются и потребляют много сетевого трафика.
- Если бы операционная система потребляла меньше ресурсов, можно было бы повысить плотность размещения виртуальных машин.
- Мы больше не можем позволить себе подвергать систему рискам безопасности, используя подход «устанавливаем все компоненты на все серверы».

В Nano Server эти проблемы решены: устанавливаются только те компоненты, которые действительно необходимы для предполагаемых сценариев использования, и ничего более. За счет этого ограничивается уязвимая область, исключаются ненужные перезагрузки, снижается объем потребляемых ресурсов. При этом повышается скорость перезагрузок и развертывания, а высвобожденные ресурсы могут использоваться для других задач.

Улучшения безопасности

Ограничение в Nano Server количества компонентов, устанавливаемых по умолчанию, также приводит к уменьшению количества драйверов, служб и открытых портов, как показано на рис. 3-2.

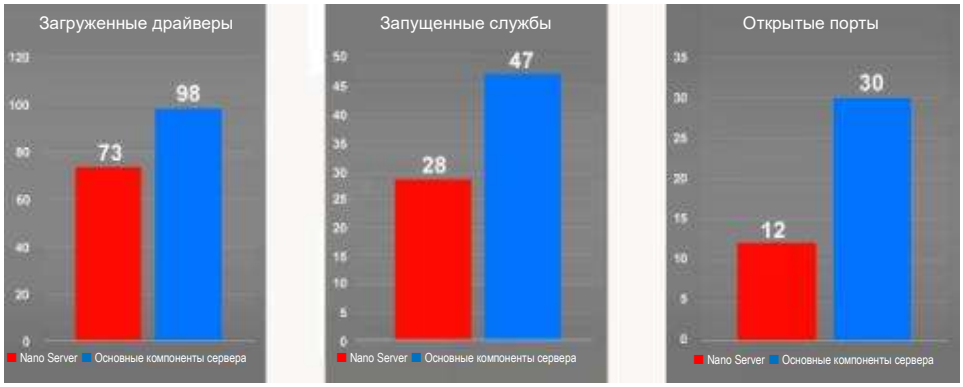


Рис. 3-2. Сравнение характеристик при установке Nano Server и основных компонентов сервера (Server Core)

Использование ресурсов

Благодаря снижению объема ресурсов, используемых сервером Nano Server, высвобождается больше возможностей для повышения плотности виртуальных машин. Кроме того, ускоряется перезагрузка, как показано на рис. 3-3.

Рис. 3-3. Сравнение использования ресурсов при установке Nano Server и основных компонентов сервера (Server Core)

Усовершенствования при развертывании

Установка Nano Server, включая специализацию, производится существенно быстрее, чем Server Core. Значительно снижен и объем требуемого дискового пространства (рис. 3-4). За счет этого обеспечивается более быстрое развертывание, при повторном развертывании уменьшается объем трафика в сети, кроме того, при развертывании снижается учитываемая общая нагрузка на сеть.

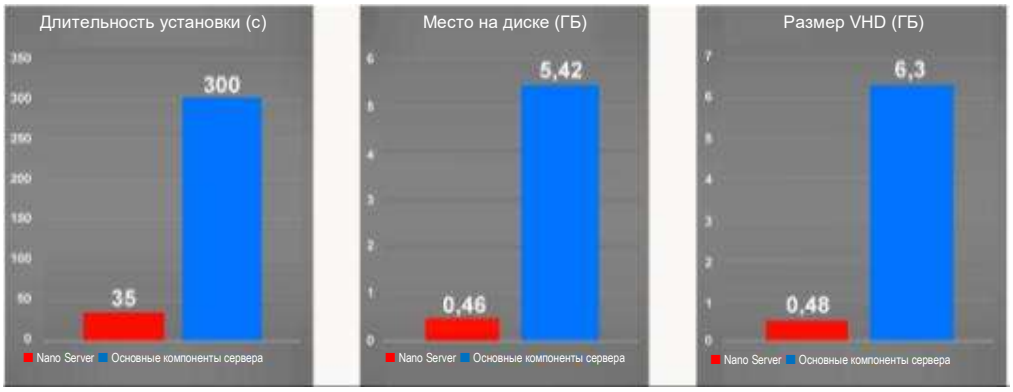


Рис. 3-4. Сравнение требований к развертыванию для Nano Server и для основных компонентов сервера (Server Core)

Устройство основных компонентов сервера

На рис. 3-5 видно, как вариант установки Nano Server соотносится с другими вариантами установки Windows Server 2016 (аналогичным образом основные компоненты сервера были выделены в Windows Server 2008 и в Windows Server 2008 R2).

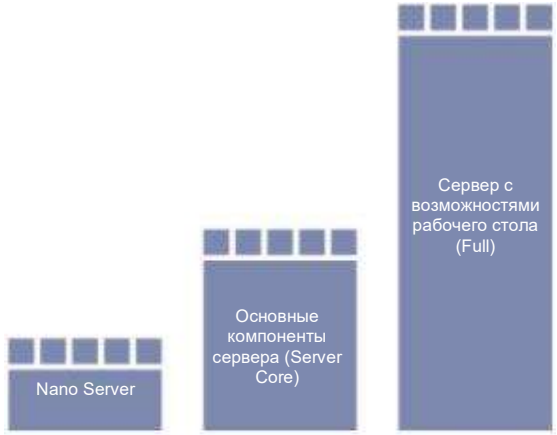


Рис. 3-5. Архитектура различных вариантов установки Windows Server 2016

> **Примечание.** Для перехода с Nano Server на другие варианты установки потребуется переустановка.

Развертывание Nano Server

Nano Server — это новый вариант установки Windows Server 2016. Тем не менее, в отличие от основных компонентов сервера (Server Core), он не отображается на экране как один из возможных вариантов при запуске программы установки. Это обусловлено тем, что перед развертыванием Nano Server необходимо настроить образ в соответствии с оборудованием и с предполагаемым назначением (подробнее — далее в этой главе). Nano Server находится на установочном носителе Windows Server 2016 в папке \NanoServer. Все пакеты, которые можно установить в Nano Server, доступны в папке \NanoServer\Packages или в сетевом хранилище.

> **Примечание.** Актуальные сведения о развертывании Nano Server доступны в руководстве, которое опубликовано по адресу https://msdn.microsoft.com/library/mt126167.aspx.

Драйверы

Nano Server не поддерживает технологию Plug-and-play в пользовательском режиме, поэтому все драйверы оборудования перед развертыванием нужно добавить в образ Nano Server. Nano Server использует такие же драйверы, что и Windows Server, поэтому можно использовать Nano Server с любым поддерживаемым оборудованием, для которого есть драйверы для Windows Server, в том числе:

- сетевые адаптеры;
- контроллеры систем хранения данных;
- диски.

В особых версиях драйверов для Nano Server нет необходимости, но если для настройки оборудования требуется определенная программа, а существующая программа не поддерживает удаленную работу, то поставщику оборудования потребуется предоставить обновленную программу или инструкции по настройке для Nano Server.

> **Примечание.** Для добавления драйверов в образ Nano Server используется сценарий New-NanoServerImage.

Роли и компоненты

Хранилище пакетов в Nano Server отделено от образа, поэтому при развертывании Nano Server в папке WinSXS не появится никаких двоичных файлов ролей и компонентов — их нужно будет добавить в образ перед развертыванием Nano Server.

Благодаря этому можно настроить развертываемый образ Nano Server так, чтобы в нем были только те компоненты, которые необходимы для предполагаемой роли сервера.

> **Примечание.** Для установки ролей и компонентов используется командлет New-NanoServerImage.

Список ролей, поддержка которых добавлена в Nano Server, доступен по адресу https://msdn.microsoft.com/library/mt126167.aspx.

Приложения

Nano Server неплохо показывает себя в ряде описанных выше сценариев. А как насчет запуска приложений на Nano Server в качестве гостевой виртуальной машины или рабочей нагрузки контейнера? Это тоже возможно, при этом ресурсы используются более рационально, к тому же исключается ряд возможных рисков безопасности. Вопрос в том, как установить приложение на Nano Server.

Помните, что Nano Server — это глубокая переработка Windows, и если в состав не включить компоненты, от которых зависит работа приложения, то приложение не будет работать. Чтобы оно заработало, его потребуется изменить для обеспечения поддержки Nano Server.

> **Примечание.** Пошаговые руководства по разработке приложений для Nano Server можно найти в примере по адресу http://blogs.technet.microsoft.com/nanoserver/2016/04/27/developing-native-apps-on-nano-server или https://blogs.technet.microsoft.com/nanoserver/2016/04/27/nanoserverapiscan-exe-updated-for-tp5/.

Если вы переделали приложение для совместимости с Nano Server, можно воспользоваться новой программой — установщиком Windows Server Apps (WSA), чтобы с его помощью упаковать приложение в пакет APPX и установить его на Nano Server. APPX и WSA образуют новую структуру для установки приложений, в них устранены некоторые ограничения, присутствовавшие в установщике MSI.

> **Примечание.** Дополнительные сведения об установке приложений Windows Server на Nano Server доступны по адресу https://blogs.technet.microsoft.com/nanoserver/2015/11/18/installing-windows-server-apps-on-nano-server.

Индивидуализация Nano Server

Как и в случае с установкой основных компонентов сервера (Server Core), для индивидуализации образа Nano Server можно использовать подмножество компонентов, доступных при автоматической установке. Чтобы не только уменьшить объем занимаемого дискового пространства, но и дополнительно ускорить развертывание Nano Server, появилась возможность настроить два часто используемых параметра автоматической установки в автономном режиме:

- имя компьютера;
- присоединение к домену с помощью Djoin.exe.

> **Примечание.** Дополнительные сведения о присоединении к домену в автономном режиме с помощью Djoin.exe доступны по адресу https://docs.microsoft.com/en-us/windows-server/get-started/deploy-nano-server.

При развертывании образа Nano Server с этими параметрами, заданными в разделе автономного режима файла автоматической установки (Unattend), сервер Nano Server будет настроен уже при первой загрузке. При этом нет необходимости во второй перезагрузке, которая происходит, к примеру, в Server Core. Таким образом, существенно уменьшается время развертывания Nano Server с уже предопределенными параметрами.

Удаленное управление Nano Server

Nano Server не содержит компонентов графической оболочки и RDP, поэтому невозможно установить подключение к его удаленному рабочему столу. Все задачи по управлению Nano Server необходимо выполнять удаленно с помощью Windows PowerShell, инструментария управления Windows (WMI), удаленной оболочки Windows (WinRS), службы аварийного управления (EMS) или удаленных инструментов с графической оболочкой.

> **Примечание.** Обсуждаемые в этом разделе средства для удаленного управления Nano Server также можно использовать и для удаленного управления основными серверными компонентами (Server Core). Единственное отличие в управлении Nano Server состоит в том, что всё необходимо делать удаленно.

Windows PowerShell

В состав Nano Server входит переработанное подмножество Windows PowerShell под названием Core PowerShell, созданное на базе CoreCLR. Core PowerShell обладает следующими возможностями:

- полная совместимость с языком Windows PowerShell;
- полные возможности удаленного использования Windows PowerShell;
- основные возможности ядра;
- поддержка всех типов командлетов, в том числе C#, Windows PowerShell и CIM.

В состав Nano Server входит Core PowerShell, поэтому для управления Nano Server можно использовать удаленное взаимодействие Windows PowerShell. При этом нужны права администратора на компьютере Nano Server, а кроме того, IP-адрес этого компьютера нужно добавить в список доверенных узлов на компьютере управления. Для этого в командной строке Windows PowerShell с правами администратора выполните следующую команду (в этом примере подразумевается, что IP-адрес компьютера Nano Server — 192.168.1.10):

```
PS C:\> Set-Item WSMan:\localhost\Client\TrustedHosts "192.168.1.10"
```

Ниже приведен пример запуска интерактивного сеанса удаленного взаимодействия:

```
PS C:\NanoServer> $ip = "192.168.1.10"
PS C:\NanoServer> $user = "Administrator"
PS C:\NanoServer> Enter-PSSession -ComputerName $ip -Credential $user
```

После этого вы можете запускать все доступные команды Windows PowerShell таким же образом, как если бы работали непосредственно с консолью Nano Server. Например:

```
[192.168.1.10]: PS C:\users\userl\Documents> Get-Process w*
[192.168.1.10]: PS C:\users\userl\Documents> ipconfig /all
```

В Nano Server доступны не все команды Windows PowerShell. Чтобы узнать, какие командлеты доступны, выполните следующую команду:

```
[192.168.1.10]: PS C:\us ers\userl\Documents> Get-Command -CommandType Cmdlet
```

Чтобы завершить удаленный сеанс, выполните команду:

```
[192.168.1.10]: PS C:\users\userl\Documents> Exit-PSSession
```

WMI

Nano Server поддерживает WMI версий 1 и 2, а также поставщики соответствующих функций.

Удаленное управление Windows (WinRM)

В Windows PowerShell можно использовать сеансы и экземпляры CIM для удаленного выполнения команд WMI с помощью удаленного управления Windows, как показано здесь:

```
PS C:\NanoServer> $ip = "192.168.1.10"
PS C:\NanoServer> $user = "Administrator"
PS C:\NanoServer> $cim = New-CimSession -Credential $user -ComputerName $ip
```

После установки сеанса CIM можно выполнять различные команды WMI, например, как показано ниже:

```
PS C:\NanoServer> Get-CimInstance -CimSession $cim -ClassName Win32_ComputerSystem | Format-List *
PS C:\NanoServer> Get-CimInstance -Query "SELECT * from Win32_Process WHERE name LIKE 'p%'"
```

Удаленная оболочка Windows (WinRS)

WinRS дает возможность запускать программы удаленно. Перед ее использованием нужно настроить службу WinRM и задать кодовую страницу:

```
C:\NanoServer> winrm quickconfig
C:\NanoServer> winrm set winrm/config/client @{TrustedHosts="*"}
```

После настройки службы WinRM можно удаленно выполнять команды точно так же, как из командной строки:

```
C:\NanoServer> winrs -r:192.168.1.10 -u:Administrator -p:Tuva ipconfig
```

Примечание. Дополнительные сведения о WinRS доступны по адресу http://technet.microsoft.com/library/hh875630.aspx.

Службы аварийного управления

Службы аварийного управления (EMS) — еще один инструмент для удаленного управления Nano Server. При этом для передачи данных между компьютером управления и компьютером Nano Server нужно использовать последовательный кабель.

Настроив EMS в данных конфигурации загрузки на компьютере Nano Server и запустив Nano Server, нужно запустить из командной строки эмулятор терминала (например, PuTTY) с правами администратора, а затем выполнить следующие действия:

1. установить такую же скорость передачи данных, как в данных конфигурации загрузки Nano Server;
2. выбрать значение «Последовательное» для параметра «Тип подключения»;
3. ввести действительное значение для параметра «Последовательная линия».

Удаленные инструменты с графическим интерфейсом

Помимо средств удаленного управления из командной строки, описанных выше, для удаленного управления Nano Server можно использовать и существующие инструменты с графической оболочкой. Поскольку в Nano Server нет локального входа в систему и удаленного рабочего стола, в составе даже основных компонентов сервера есть инструменты, не имеющие работающих удаленно аналогов с графической оболочкой (например, диспетчер задач), и существует набор веб-инструментов под названием «Средства управления сервером», который доступен в Azure. Это удаленные веб-инструменты для управления серверами Nano Server, основными компонентами сервера (Server Core) и остальными вариантами установки.

Примечание. Дополнительные сведения о средствах управления сервером приведены в главе 5.

Отслеживание работы Nano Server

Работу Nano Server можно отслеживать с помощью System Center 2016 Operations Manager. Для этого потребуются обновленные пакеты управления, они позволят отслеживать работу базовой

операционной системы Nano Server, отказоустойчивого кластера, DNS, IIS и других. Агент можно развернуть удаленно с помощью консоли System Center 2016 Operations Manager.

Запустив обычным образом мастер обнаружения, на сервер Nano Server можно установить агент. Также можно настроить Nano Server на переадресацию событий, связанных с безопасностью, службе сборщика аудита System Center 2016 Operations Manager.

Консоль восстановления Nano Server

В состав Nano Server входит консоль восстановления, с помощью которой можно получить доступ к Nano Server даже в случае, когда из-за неправильной настройки сети установить подключение к Nano Server обычным образом не удается. С помощью консоли восстановления можно исправить параметры сети, а затем воспользоваться обычными средствами удаленного управления.

При загрузке Nano Server на виртуальной машине или на физическом компьютере с монитором и клавиатурой отобразится экран входа в систему (в текстовом режиме). Чтобы увидеть имя компьютера и IP-адрес Nano Server, войдите в систему с учетными данными администратора. Для навигации в этой консоли можно использовать:

- клавиши со стрелками для прокрутки;
- клавишу Tab для перехода к любому тексту, начинающемуся с «>», затем необходимо нажать клавишу Enter для выбора;
- клавишу Esc, чтобы вернуться назад на один экран или на одну страницу (если открыта главная страница, то при нажатии клавиши Esc вы выйдете из системы).

На некоторых экранах отображаются дополнительные возможности, они перечисляются в последней строке экрана. Например, чтобы отключить сетевой адаптер, можно при работе с сетевым адаптером нажать F4.

> **Примечание.** Консоль восстановления поддерживает только основные функции клавиатуры. Индикаторы клавиатуры, 10-клавишные секции и переключение раскладки клавишами Caps Lock и Num Lock не поддерживаются.

Модели обслуживания

Предыдущие версии Windows Server обслуживались и поддерживались по модели «5 + 5»: обычная поддержка в течение пяти лет и расширенная поддержка в течение еще пяти лет. Этот принцип сохранится и для Windows Server 2016. Заказчики, выбирающие Windows Server 2016 с рабочим столом или основные компоненты сервера (Server Core), сохранят эту возможность обслуживания. Теперь она называется «долгосрочным обслуживанием» — Long-Term Servicing Branch (LTSB).

Заказчики, выбирающие установку Nano Server, получают более активную модель обслуживания, аналогичную используемой для Windows 10. Эти периодические выпуски называются «текущая ветвь для бизнеса» — Current Branch for Business (CBB). Этот подход предназначен для поддержки заказчиков, использующих ускоренные циклы разработки и стремящихся быстрее внедрять новые возможности. Поскольку при этом типе обслуживания по-прежнему предоставляются новые компоненты и возможности, для развертывания и использования Nano Server в рабочей среде также требуется подписка Software Assurance (табл. 3-1).

Таблица 3-1.

Вариант установки	Долгосрочное обслуживание (LTSB)	Текущая ветвь для бизнеса (CBB)
Сервер с возможностями рабочего стола	Да	Нет
Основные компоненты сервера (Server Core)	Да	Нет
Nano Server	Нет	Да

Цель заключается в том, чтобы выпускать новые компоненты для Nano Server два-три раза в год. Эта модель будет похожа на модель обслуживания клиентских операционных систем Windows, но с некоторыми отличиями. У нас общая задача — быстро предоставлять заказчикам новые и полезные технологии с полным пониманием особенностей и требований серверной среды.

Несмотря на то что потребуется выполнять обновление до актуальных версий по мере их выхода, на серверах новые версии не будут устанавливаться автоматически — администратор должен установить их вручную по своему усмотрению. Nano Server будет обновляться чаще, поэтому заказчики Nano Server CBB смогут отставать от последней версии не более чем на два выпуска. В любое время будут обслуживаться только два выпуска CBB, поэтому после выхода третьего выпуска Nano Server поддержка первого прекратится и будет необходимо перейти с него на более поздний. Когда появится четвертый выпуск, нужно будет перейти со второго на более поздний и т. д.

Контейнеры

Автор: Джон Мак-Кейб (John McCabe)

В этом разделе описывается новая технология в составе Windows Server 2016 — *контейнеры*. Существует два типа контейнеров:

- контейнеры Windows Server;
- контейнеры Hyper-V.

В этом разделе мы поговорим о том, что такое контейнеры и почему это важно знать.

Примечание. Поддержка контейнеров Linux будет также добавлена в скором времени.

Что такое контейнер?

Контейнер в простейшей форме — это то, что отделяет какой-либо объект от окружающих объектов. В Windows Server это изолированная среда, в которой можно запускать приложения, не опасаясь того, что при этом изменятся другие приложения или параметры. У контейнеров есть общие компоненты (ядро, системные драйверы и пр.), благодаря чему ускоряется их запуск и обеспечивается более высокая плотность, чем при использовании виртуальных машин. На рис. 3-6 показано общее устройство контейнера.

Рисунок 3-6. Диаграмма использования контейнеров

Как видно из этого рисунка, в операционной системе физического сервера может быть размещено множество контейнеров. При этом они полностью изолированы друг от друга, но совместно используют основные компоненты операционной системы, например ядро.

Интерес представляет поведение приложений в контейнерах. У приложения может быть ряд зависимых компонентов, необходимых для его работы. Такие зависимые компоненты могут существовать только внутри того же контейнера, в котором находится приложение. Это означает, что в случае каких-либо неполадок приложения А и двоичных файлов, от которых оно зависит, работа приложения Б (и его двоичных файлов) никак не будет затронута. Если в обычной среде приложение А, к примеру, удалит реестр, это нарушит работу и приложения А, и приложения Б. В случае если используются контейнеры, приложения А и Б друг от друга изолированы, поэтому изменение реестра приложением А никак не повлияет на приложение Б.

Все двоичные файлы и зависимые компоненты размещены внутри контейнера, поэтому приложение, работающее в контейнере, полностью переносимо. Это означает, что контейнер можно развернуть на любом узле, на котором запущен диспетчер контейнеров, после чего контейнер можно запустить и использовать без каких-либо дополнительных изменений и настроек. Например, разработчик может начать развертывать приложение и развернуть его в контейнере Hyper-V в Windows 10. Когда приложение будет готово к развертыванию в рабочей среде, его можно запустить в Windows Server 2016, в том числе в Nano Server, в общедоступном, частном или гибридном облаке.

Контейнеры состоят из нескольких уровней. Первый уровень является базовым. Это образ операционной системы, на основе которого выстраиваются все остальные уровни. Образ хранится в репозитории образов, поэтому в нужных случаях можно воспользоваться ссылкой на него. Следующим (и иногда последним) уровнем является уровень платформы приложений, который может быть общим для всех приложений. Например, если базовым уровнем является ядро Windows Server, то уровнем платформы приложений могут быть .NET Framework и Internet Information Services (IIS). Второй уровень тоже может храниться в виде образа, который при вызове также описывает зависимость от базового уровня — ядра Windows Server. И наконец, уровень приложений: на этом уровне хранится само приложение, ссылающееся на уровень платформы приложений и, в свою очередь, на базовый уровень.

На базовый уровень и на уровень платформы приложений могут в любое время ссылаться все прочие создаваемые контейнеры приложений. Каждый уровень доступен только для чтения, исключая верхний уровень развертываемого образа. Например, если развертывается контейнер, зависящий только от образа ядра Windows Server, этот уровень ядра Windows Server будет являться верхним уровнем контейнера; для сохранения всех операций записи и изменений, сделанных во время выполнения, создается песочница. После этого сделанные изменения можно сохранять в качестве другого образа для использования в дальнейшем. Этот же принцип применяется и при развертывании образа с уровнем платформы приложений: этот уровень получит собственную песочницу, а при развертывании приложения можно сохранить эту песочницу в виде образа для последующего использования.

Общий принцип таков: при развертывании контейнера на узле узел сам определяет наличие базового уровня. Если этого уровня нет, узел получает его из репозитория образов.

В дальнейшем этот процесс повторяется для уровня платформы приложений, а затем создается контейнер приложения, который требуется развернуть. Если после этого потребуется создать другой контейнер с такими же зависимостями, то будет достаточно выполнить команду на создание нового контейнера приложений. Этот контейнер будет подготовлен к работе практически мгновенно, поскольку все зависимости уже есть в наличии. Если имеется контейнер приложения, зависящий от другого уровня платформы приложений и от исходного базового уровня ядра Windows Server, то можно просто получить другой контейнер платформы приложений из хранилища образов и запустить новый контейнер приложения.

Для чего нужны контейнеры?

Контейнеры предоставляют существенные преимущества по сравнению с традиционной моделью развертывания приложений на виртуальной машине или на физическом компьютере.

Первое преимущество относится к развертыванию. Традиционная проблема, с которой вынуждены сталкиваться все разработчики приложений, связана с перемещением приложения из среды разработки в тестовую, а затем в рабочую среду. Разработчикам приходится затрачивать немало времени и усилий на проверку зависимостей приложения при его перемещении. Если же приложение развернуто в контейнере, можно перемещать весь контейнер, поскольку он изолирован, все двоичные файлы находятся внутри него.

Еще одно преимущество контейнеров — расширенные возможности масштабирования по сравнению с развертыванием приложений на виртуальных машинах. Для создания трех разных сред для разработки, тестирования и работы в модели с виртуальными машинами потребуются по крайней мере три виртуальные машины. В модели с контейнерами будет достаточно одной виртуальной машины. Одна виртуальная машина, на которой запущен диспетчер контейнеров, может поддерживать работу сразу трех контейнеров, имитирующих среду разработки, тестовую среду и рабочую среду. При использовании контейнеров для работы нескольких сред потребуется меньше виртуальных машин, поэтому можно добиться более высокой плотности размещения в облачной среде.

Контейнеры также поддерживают быстрое развертывание и быструю работу приложений. В отличие от виртуальных машин, в контейнерах нет базовой операционной системы как таковой.

Рассмотрим работу контейнеров с точки зрения развертывания. Если требуется создать новое приложение или запустить дополнительные экземпляры существующего приложения, чтобы поддерживать возросшую нагрузку, достаточно загрузить новый контейнер; операционная система уже есть. По этой причине развертывание и запуск контейнеров происходят значительно быстрее, чем развертывание и запуск виртуальных машин: нет необходимости дожидаться запуска операционной системы.

Контейнеры Windows Server и контейнеры Hyper-V

В Windows Server 2016 доступно два типа контейнеров:

- контейнеры Windows Server;
- контейнеры Hyper-V.

Контейнеры Windows Server можно считать равноценными контейнерам Linux. Контейнеры Windows Server изолируют приложения на одном и том же узле. Каждый контейнер получает собственное представление системы узла, в том числе ядра, процессов, файловых систем, реестра и других компонентов. Контейнеры Windows Server работают между уровнем пользовательского режима и уровнем режима ядра.

Контейнеры Hyper-V созданы на основе технологии, опирающейся на виртуализацию с аппаратной поддержкой. За счет аппаратной виртуализации приложения в контейнерах Hyper-V получают среду с высокой степенью изоляции, в которой запущенные контейнеры никоим образом не могут повлиять на операционную систему физического сервера.

На рис. 3-7 показан принцип функционирования обеих технологий контейнеров, доступных в Windows Server 2016.

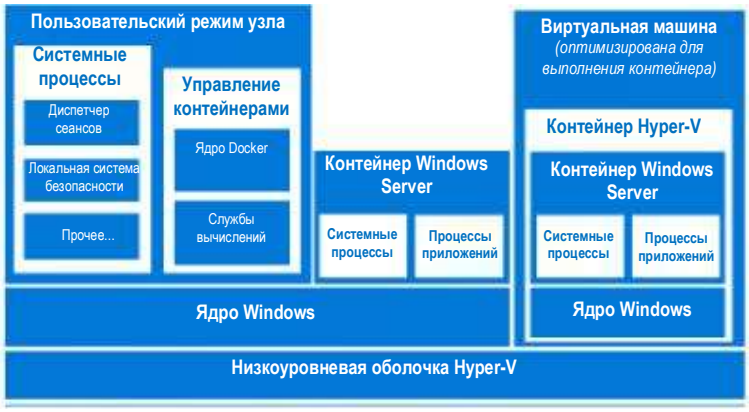


Рис. 3-7. Контейнеры Windows Server 2016 и контейнеры Hyper-V на одном и том же физическом сервере

Как показано на рисунке, на одном физическом сервере может одновременно использоваться целый набор из контейнеров Hyper-V, виртуальных машин и контейнеров Windows Server. Поскольку доступны два типа контейнеров, возникает вопрос, в каких случаях для разработки и развертывания приложений следует использовать контейнеры Windows Server, а в каких — контейнеры Hyper-V. Выбор зависит от требований приложения (или заказчика, использующего контейнеры) к масштабированию и изоляции с аппаратной поддержкой.

Например, если требуется расширенное масштабирование, его проще достичь при использовании контейнеров Windows Server. Если же требуются расширенные возможности изоляции, доступные при виртуализации с аппаратной поддержкой, лучше использовать контейнеры Hyper-V.

Выбирая тот или иной тип контейнеров, важно понимать жизненный цикл разработки. С точки зрения разработчика, привычные средства и инструменты, такие как Visual Studio, дают возможность создавать и развертывать приложения непосредственно в контейнерах. С помощью этих же инструментов можно описывать, какие основные функции будут нужны в базовой операционной системе, какие библиотеки могут быть общими для нескольких приложений, а какие — выделенными для одного контейнера. На рис. 3-8 показаны зависимости и режим выполнения контейнеров.

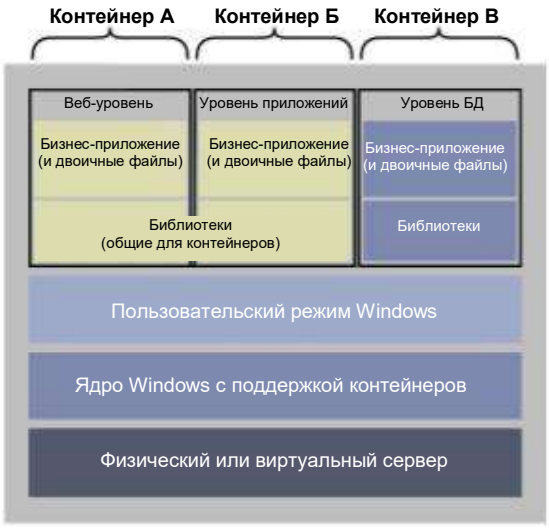


Рис. 3-8. Контейнеры и зависимости

Вне зависимости от выбранного типа контейнера приложение, развернутое в контейнере, будет совместимо с обоими типами. Разработчик может создать приложение, разместить его в контейнере Windows Server, а затем переместить в контейнер Hyper-V без каких бы то ни было изменений. За счет этого обеспечивается высокая гибкость, особенно если требования в отношении масштабирования или изоляции изменяются по сравнению с использовавшимися при планировании.

Управление контейнерами

В 2015 году корпорация Microsoft объявила о поддержке подсистемы Docker на виртуальных машинах Linux в Azure. Это здорово, но остается вопрос: как насчет поддержки Docker непосредственно в Windows? С появлением контейнеров Windows Server и Hyper-V механизм Docker становится еще полезнее, поскольку теперь с его помощью можно управлять контейнерами Docker как в традиционной среде Linux, так и в Windows. Кроме того, теперь есть доступ ко всем образам, доступным через Docker, поэтому можно загружать и развертывать их.

Среда выполнения Docker работает в виде приложения с контейнерами Windows Server и Hyper-V. Docker содержит все необходимые средства для разработки и работы с контейнерами Windows обоих типов — Hyper-V и Windows Server. Поддерживается уже описанный уровень гибкости: достаточно разработать приложение в одном контейнере, чтобы иметь возможность запускать его где угодно.

На рис. 3-9 показано положение Docker относительно контейнеров Windows Server и Hyper-V, а также сравнивается работа Docker в Windows Server и Linux.

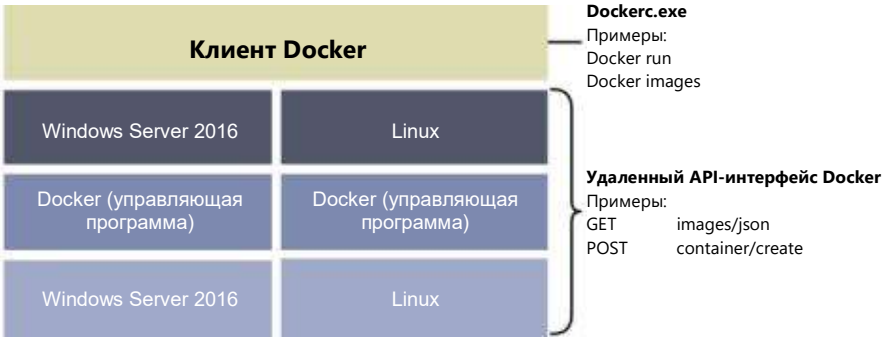


Рис. 3-9. Docker в Windows и в Linux

Docker работает на одном и том же уровне и в среде с контейнерами Windows Server, и в среде с контейнерами Linux. При этом Windows Server или Linux находятся над уровнем подсистемы Docker. Клиент Docker подключается к любой подсистеме, поэтому конечные пользователи получат одинаковые возможности управления в любых условиях.

Основной принцип — ускорение разработки и реализация возможности однократного создания приложения для его дальнейшего развертывания в любой среде, как показано на рис. 3-10.

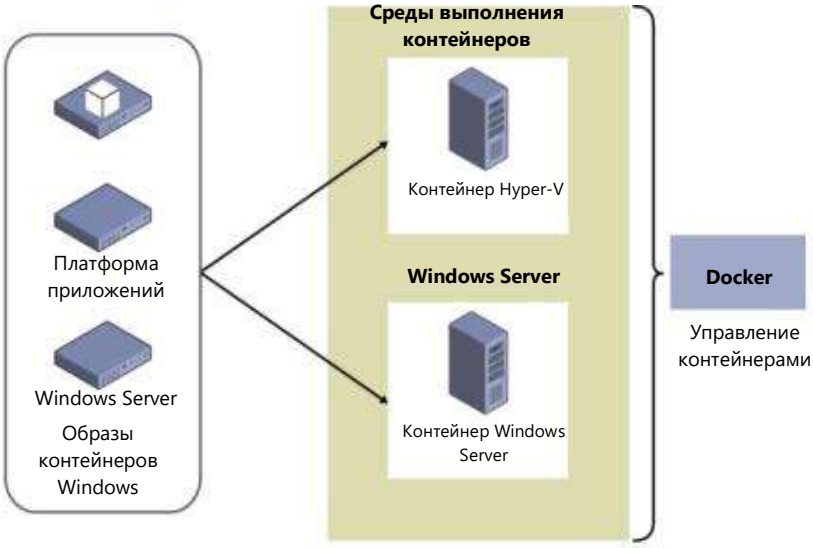


Рис. 3-10. Достаточно однократно создать контейнер, чтобы развертывать его в любой среде

Сети

Необходимо подумать о том, каким образом контейнеры будут обмениваться данными с корпоративной сетью и с внешними ресурсами. На рис. 3-11 показана схема подключения контейнера к «внешнему миру».

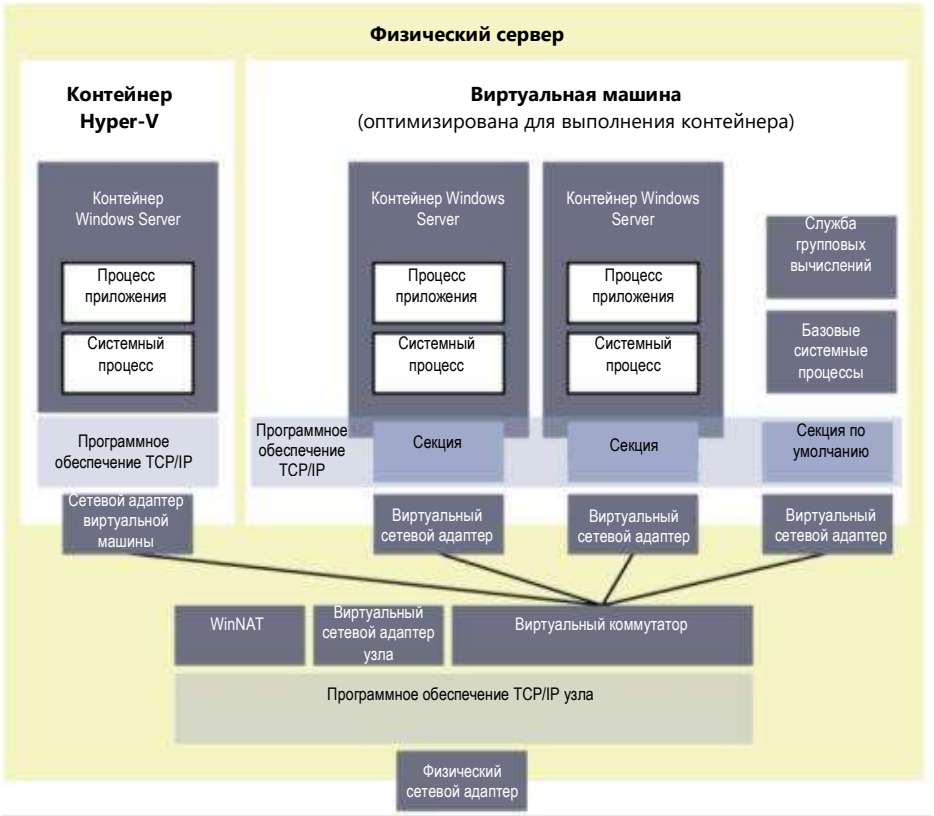


Рис. 3-11. Сетевые подключения контейнеров

Как показано на рис. 3-11, каждый контейнер подключается посредством виртуального сетевого адаптера (контейнер Windows Server) или сетевого адаптера виртуальной машины (контейнер Hyper-V) к виртуальному коммутатору, настроенному на узле. Каждый виртуальный сетевой адаптер изолирован от следующего и рассматривается как отдельная секция. Эти виртуальные сетевые адаптеры подключаются к виртуальному коммутатору через порты (аналогично Hyper-V). Виртуальный сетевой адаптер физического узла изолирован от контейнеров. Сетевые подключения к контейнерам Hyper-V являются прозрачными для виртуальной машины и проходят через сетевой адаптер виртуальной машины.

Подключение к внешним ресурсам можно реализовать разными способами. Каждый из них зависит от конкретного сценария, принятого для использования контейнеров. Например, если нужно создать среду контейнера для разработчиков, лучше всего использовать преобразование сетевых адресов (NAT) для сети контейнеров. При этом предоставляется частное IP-пространство (IP-адреса назначаются DHCP-сервером), изолированное от внешней среды. Подключения между контейнерами ограничены, но поддерживается переадресация портов в контейнерную среду, с которой вы работаете. Весь трафик, поступающий на общедоступный IP-адрес NAT (IP-адрес внешнего сетевого адаптера сервера) будет сравниваться с таблицей, управляемой через WinNAT, и передаваться в контейнер.

Если разработчикам или бизнесу требуется развертывание в небольшом масштабе, причем контейнеры должны находиться в пространстве IP-адресов компании, можно использовать прозрачные сетевые подключения контейнеров. При этом используется существующее пространство IP-адресов (статических или назначаемых DHCP-сервером), в котором IP-адреса назначаются запускаемым контейнерам. Если вы не используете DHCP, то не сможете получить IP-адрес шлюза. При прозрачном сетевом подключении контейнеры смогут обмениваться данными друг с другом и с внешними службами, такими как SQL и т. п.

И наконец, если требуется развертывание облачного уровня, можно использовать туннелирование уровня 2 или мост уровня 2. Оба этих метода представляют собой виртуализацию для контейнеров с возможностью полной изоляции трафика при многоузловом развертывании контейнеров в центре обработки данных.

В режиме моста уровня 2 расширение виртуальной платформы фильтрации (VFP) виртуального коммутатора в узле контейнеров будет работать в качестве моста и будет переписывать MAC-адрес в нужных случаях. Уровни 3 и 4 остаются без изменений.

Режим туннелирования на уровне 3 используется, когда нужна сетевая политика в сценарии облачного развертывания. Внешний виртуальный коммутатор предоставляет все возможности подключений для контейнера. Весь трафик контейнера перенаправляется через физический сервер, а MAC-адрес переписывается перед входом в структуру сети.

По умолчанию механизм Docker пытается использовать NAT. Если сеть NAT не обнаружена, выполняется попытка создать такую сеть. Все контейнеры, созданные после этого, будут присоединены к сети NAT. Это поведение, используемое по умолчанию, можно изменить, выполнив, к примеру, следующую команду:

```
Docker -b "none"
```

Где «none» — имя сети, а -b — мост. В этом случае мы ничего ни к чему не присоединяем.

Для создания прозрачной сети можно использовать следующую команду:

```
Docker network create -d transparent -gateway 192.168.0.254 "TransparentNET"
```

Несколько сетей контейнеров

Если необходимо поддерживать несколько сетей на узле контейнеров, действуют следующие правила:

- В каждом узле контейнеров можно создать только одну сеть NAT.

- Если несколько сетей используют внешний виртуальный коммутатор (например, прозрачное подключение, мост уровня 2, прозрачное подключение уровня 2), то каждая из них должна использовать собственный сетевой адаптер.
- Разные сети должны использовать разные виртуальные коммутаторы.

Брандмауэры

На каждом узле контейнеров по умолчанию включен набор правил брандмауэра. Правила настроены с использованием стандартного набора. Если нужно, чтобы контейнеры отвечали на эхо-запросы (ICMP) или получали IP-адрес через DHCP, убедитесь в том, что эти правила включены на узле контейнеров. Чтобы разрешить передачу различных типов трафика в контейнеры, потребуется настроить дополнительные правила.

Если используется NAT и требуется включить переадресацию портов для контейнеров, соответствующие правила брандмауэра будут созданы при настройке автоматически.

Устранение неполадок

Docker больше не ведет отдельный файл журнала, он записывает все события в журнал приложений, доступный в программе просмотра событий.

Можно отфильтровать этот журнал, выбрав Docker в качестве источника событий, или использовать Windows PowerShell для просмотра журнала, выполнив следующую команду:

```
Get-EventLog -LogName Application -Source Docker
```

Кроме того, можно запустить Docker в режиме отладки с помощью команды:

```
Dockerd.exe -D
```

Чтобы запустить режим отладки, можно также использовать файл конфигурации Docker (он находится в папке [или его нужно поместить в эту папку, если его там нет] по адресу C:\ProgramData\docker\config\daemon.json):

```
{
"debug": true
}
```

Установка контейнеров

Чтобы начать использовать контейнеры Windows Server, необходимо установить компонент «Контейнеры». Если также нужно использовать контейнеры Hyper-V, необходимо установить Hyper-V.

После этого потребуется установить и настроить программу Docker, которая не входит в состав Windows Server 2016.

> **Примечание.** Дополнительные сведения об установке этих компонентов, а также о настройке и установке Docker доступны в кратких руководствах, которые опубликованы по адресу https://aka.ms/windowscontainers.

Правила для контейнеров

В Windows загружаемый образ контейнера *должен* быть образом операционной системы, совпадающей с узлом контейнера по номеру сборки и по уровню исправлений. В табл. 3-2 указаны данные о поддержке базового образа на разных типах узлов контейнеров.

Таблица 3-2. Поддерживаемые образы контейнеров на разных типах узлов контейнеров

		Среда выполнения операционной системы контейнера	
		Основные компоненты сервера	**Nano Server (AB1)**
Операционная система узла	**Сервер с пользовательским интерфейсом (LTSB)**	Контейнеры Windows Server и/или контейнеры Hyper-V	Контейнеры Windows Server и/или контейнеры Hyper-V
	Основные компоненты сервера (LTSB)	Контейнеры Windows Server и/или контейнеры Hyper-V	Контейнеры Windows Server и/или контейнеры Hyper-V
	Nano Server (AB1)	Контейнеры Hyper-V	Контейнеры Windows Server или контейнер Hyper-V

После установки исправлений на узел контейнеров потребуется установить исправление на образ контейнера и зафиксировать его.

Корпорация Microsoft будет ежемесячно выпускать обновленные образы Windows Docker для использования их при сборке образа контейнера.

Если планируется использовать контейнеры Hyper-V внутри гостевой виртуальной машины, должны быть выполнены следующие условия:

- На узле контейнеров должна быть включена вложенная виртуализация.
- На узле должно быть не менее 4 ГБ оперативной памяти.
- В качестве операционной системы узла необходимо использовать Windows Server 2016 или Windows 10.
- У гостевой виртуальной машины узла контейнеров должно быть по крайней мере два виртуальных процессора.

В репозитории Docker есть дополнительные образы, отвечающие таким же требованиям.

> **Примечание.** В этой книге не приводятся подробные инструкции по развертыванию контейнеров Windows Server. Дополнительные сведения доступны по адресу https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-docker/configure-docker-daemon.

ГЛАВА 4

Безопасность и идентификация

В последние годы тема безопасности компьютерных систем стала важнейшей в области ИТ. Это неудивительно: известно множество случаев взлома ИТ-ресурсов ведущих компаний и государственных организаций, связанных с утечкой личных данных клиентов и сотрудников.

Злоумышленники, располагающие мощными инструментами и не сталкивающиеся с адекватными защитными мерами, могут проникать в ИТ-среду крупных организаций так, что их атака долгое время остается незамеченной. Это дает возможность нападающим похищать конфиденциальные данные или атаковать внутренние ресурсы.

В этой главе вы узнаете об уровнях защиты Microsoft Windows Server 2016 от новых угроз и о том, как использовать систему Windows Server 2016, чтобы повысить уровень безопасности всей ИТ-среды. Сначала мы поговорим о новом решении — экранированных виртуальных машинах. Это решение защищает виртуальные машины от атак, нацеленных на базовую структуру, в которой они работают.

Затем мы опишем встроенные в Windows Server 2016 компоненты обеспечения устойчивости в отношении различных угроз, а также расширенные события аудита, помогающие средствам безопасности быстро обнаруживать вредоносные действия.

И наконец, мы продемонстрируем план защиты привилегированного доступа на основе прежних и новых возможностей Windows Server.

Экранированные виртуальные машины

Автор: Джон Сэвилл (John Saville)

В большинстве современных виртуальных сред доступ к ресурсам виртуальных машин, например к хранилищам, обычно имеют разного рода системные администраторы: администраторы виртуализации, администраторы хранилищ, сетевые администраторы, администраторы резервного копирования и другие. Многим организациям, в том числе поставщикам услуг размещения, требуется способ защитить виртуальные машины, в том числе и от администраторов. Именно эту задачу решают экранированные виртуальные машины. Помните, что защита от администраторов необходима по ряду причин. Вот некоторые из них:

- фишинговые атаки;
- похищенные учетные данные администраторов;
- инсайдерские атаки.

Экранированные виртуальные машины обеспечивают защиту данных и состояния виртуальных машин от просмотра, кражи и подмены пользователями с правами администратора. Экранированные виртуальные машины можно использовать на виртуальных машинах второго поколения, поддерживающих необходимые функции. В частности, требуется поддержка безопасного запуска, UEFI и виртуального модуля TPM версии 2.0. Несущие узлы Microsoft Hyper-V при этом должны работать под управлением Windows Server 2016, а в качестве гостевой операционной системы на виртуальных машинах можно использовать Windows Server 2012 или более поздней версии.

В среде развертывается новый экземпляр службы защиты узла, в котором хранятся ключи, необходимые одобренному узлу Hyper-V для подтверждения своей работоспособности. Такое подтверждение требуется для запуска экранированных виртуальных машин.

Экранированные виртуальные машины обладают следующими преимуществами:

- шифрование дисков с помощью BitLocker (с использованием vTPM);
- защищенный рабочий процесс виртуальных машин (VMWP), шифрование трафика динамического перемещения, а также файлов состояния выполнения, сохраненного состояния, контрольных точек и даже файлов реплики Hyper-V;
- отсутствие консольного доступа, блокирование Windows PowerShell Direct, компонентов гостевых служб (для работы Copy-VMFile) и других сервисов, предоставляющих возможность доступа к виртуальной машине для пользователей и процессов с правами администратора.

Как добиться такого уровня безопасности? Прежде всего необходимо убедиться в том, что безопасность узла Hyper-V не была нарушена перед выпуском ключей, необходимых для доступа к ресурсам виртуальной машины, службой защиты узла (HGS). Аттестация может проводиться двумя способами. Рекомендуемый способ — использовать TPM 2.0 на узле Hyper-V. С помощью TPM проверяется процесс загрузки сервера, что дает возможность удостовериться в том, что на сервере отсутствуют вредоносное ПО или пакеты rootkit, нарушающие безопасность. Модуль TPM защищает данные, передаваемые в службу аттестации HGS и из нее. Если узлы не поддерживают TPM 2.0, то можно использовать способ аттестации на основе Active Directory. Впрочем, в этом случае проверяется лишь принадлежность узла к определенной группе Active Directory, поэтому не обеспечивается такой же уровень надежности и защиты от подмены двоичных файлов и, следовательно, от получения злоумышленником прав администратора узла. Впрочем, при этом доступны такие же возможности экранированных виртуальных машин.

После того как узел прошел аттестацию, он получает от службы аттестации в HGS сертификат работоспособности, разрешающий узлу получить ключи у службы защиты ключей, также

работающей в HGS. Эти ключи передаются в зашифрованном виде, их можно расшифровать только внутри защищенного анклава — это новая возможность в Windows 10 и Windows Server 2016 (подробности ниже). Полученные ключи можно использовать для расшифровки vTPM, после чего виртуальная машина сможет получить доступ к своему хранилищу, защищенному с помощью BitLocker, что даст возможность запуститься. Узел сможет получить нужный ключ и открыть доступ виртуальной машины к зашифрованному хранилищу только в том случае, если этот узел получил соответствующее разрешение и не скомпрометирован. При этом администратор такого доступа не получит, поскольку виртуальный жесткий диск (VHD) находится на физическом диске в зашифрованном виде.

На этом этапе может показаться, что такой подход неработоспособен. Предположим, что я администратор Hyper-V. Если узел получил ключи для запуска виртуальной машины, то я смогу получить доступ к памяти узла и, следовательно, к этим ключам, поэтому любые меры защиты виртуальных машин от доступа администратора заведомо несостоятельны. К счастью, решение проблемы обеспечит еще одна новая служба Windows 10 и Windows Server 2016. Эта служба — упомянутый ранее защищенный анклав, который также называется виртуальным безопасным режимом (VSM). Этой службой пользуются многие компоненты, в том числе Credential Guard. VSM — это безопасная среда выполнения, в которой содержатся секреты и ключи, при этом важные процессы безопасности выполняются в *компактных безопасных процессах* в защищенном виртуальном разделе.

VSM — это не виртуальная машина Hyper-V, а своего рода небольшой виртуальный сейф, защищенный виртуализацией на основе таких технологий, как преобразование адресов второго уровня (SLAT), позволяющее предотвратить прямой доступ пользователей к памяти, блок управления вводом-выводом памяти (IOMMU), предназначенный для защиты от атак с прямым доступом к памяти (DMA), и др. Операционная система Windows, в том числе ее ядро, не имеет доступа к VSM. Такой доступ разрешен только безопасным процессам, подписанным корпорацией Microsoft. Безопасный процесс vTPM используется для виртуального модуля TPM каждой виртуальной машины, он отделен от остальных процессов виртуальных машин, которые выполняются внутри защищенного рабочего процесса виртуальных машин нового типа. Это означает, что даже при полном доступе к ядру получить доступ к памяти, в которой хранятся ключи, невозможно. Например, если запустить систему с присоединенным отладчиком, этот этап будет помечен как часть процесса аттестации, проверка работоспособности не пройдет и ключи не будут предоставлены узлу. Как вы наверняка помните, ключи, предоставляемые службой защиты ключей, отправляются в зашифрованном виде, их расшифровывает VSM, поэтому расшифрованные ключи всегда защищены от операционной системы узла.

В результате у вас есть возможность создать безопасную среду виртуальных машин, защищенную от администраторов любого уровня (если узел использует TPM 2.0), что дает возможность устранить многие уязвимости, которые сохраняются в других случаях.

> **Примечание.** Подробные руководства о реализации этого сценария доступны по адресу https://gallery.technet.microsoft.com/Shielded-VMs-and-Guarded-70c5b471/view/Discussions.

Технологии устойчивости к угрозам

В Windows Server 2016 встроены технологии обеспечения устойчивости к угрозам, играющие важную роль в повышении общего уровня безопасности. Возможности этих технологий включают блокирование внешних злоумышленников, пытающихся воспользоваться уязвимостями (защита потока управления), а также защиту от атак злонамеренных пользователей и программ, получивших доступ к серверу с правами администратора (Credential Guard и Device Guard). Об этих новых возможностях мы поговорим в следующих разделах.

Защита потока управления

В Windows Server 2016 и Windows 10 для обеспечения безопасности операционной системы применена защита потока управления. Эта оптимизированная функция безопасности платформы существенно затрудняет запуск произвольного кода посредством использования уязвимостей, таких, например, как *переполнение буфера*.

Кроме того, когда разработчик компилирует код, компилятор выполняет дополнительные проверки безопасности кода и выявляет набор функций, считающихся источником косвенных вызовов. Эти косвенные вызовы могут поступать из уязвимостей: в качестве параметров функции могут передаваться данные, специально сформированные таким образом, чтобы вызвать нарушение работы. Косвенный вызов в коде, не поддерживающем защиту потока управления, может вызвать переполнение буфера памяти, что, в свою очередь, приводит к сбоям других приложений и к выполнению вредоносного кода с расширенными правами. Если же компилятор определил эти наборы функций в качестве потенциальных уязвимостей и пометил их соответствующим образом, среда выполнения обнаруживает их и вызывает дополнительные функции, проверяющие допустимость косвенных вызовов. Если косвенный вызов не прошел проверку, работа приложения завершается, не позволяя этому приложению нанести ущерб системе.

Device Guard в Windows Server 2016

Каждый день создаются тысячи новых вредоносных файлов, поэтому использование традиционных методов защиты, таких как антивирусы, выявляющие вредоносные программы по характерным сигнатурам, не всегда дает нужный результат. Технология Device Guard в Windows Server 2016 меняет весь подход к безопасности: вместо режима, в котором все приложения считаются доверенными, если они не заблокированы антивирусом, используется режим, в котором операционная система доверяет только тем приложениям, которые разрешены вашей компанией.

Что такое Device Guard

Device Guard защищает программное обеспечение, работающее в режиме ядра и в пользовательском режиме. В режиме ядра Device Guard проверяет, имеют ли драйверы известную подпись (например, подпись WHQL). Также он дает возможность дополнительно ограничить драйверы, поместив их в список безопасных программ в политике. Device Guard не позволяет драйверам загружать динамический код и блокирует все драйверы, не вошедшие в список безопасных программ. Если на компьютер попадет измененный злоумышленником драйвер, который попытается изменить код в памяти, то этот драйвер невозможно будет запустить. Device Guard также обеспечивает защиту в пользовательском режиме (UMCI): можно создавать политики целостности кода (CI), определяющие доверенные и разрешенные программы для отдельных серверов.

Более подробная информация о Device Guard доступна в следующих источниках (этот список не является исчерпывающим):

- «Введение в Device Guard»;
- «Требования к планированию развертывания Device Guard»;
- «Политики целостности кода».

Расширенная защита в режиме ядра с помощью проверки целостности кода низкоуровневой оболочки

Защита и основные функции Device Guard работают на аппаратном уровне. Устройства, процессоры которых поддерживают технологию преобразования адресов второго уровня (SLAT) и расширения виртуализации, такие как Intel VT-x и AMD-V, смогут воспользоваться средой безопасности на основе виртуализации (VBS). Эта среда существенно повышает уровень безопасности в Windows за счет изоляции важнейших служб Windows от самой операционной системы.

Device Guard использует VBS для изоляции службы проверки целостности кода низкоуровневой оболочки (HVCI). За счет этого Device Guard защищает процессы и драйверы уровня ядра от уязвимостей и атак «нулевого дня». Чтобы заставить все программы, работающие в режиме ядра, безопасно выделять память, HVCI использует функциональность процессора. Это означает, что после выделения памяти ее состояние изменяется с возможности записи на доступ только для чтения или только для запуска. Принудительный перевод памяти в такое состояние предотвращает внедрение вредоносного кода в процессы и драйверы уровня ядра посредством таких методик, как переполнение буфера или атака, использующая ошибки в работе с памятью приложения (heap spraying).

Для реализации этого уровня безопасности технологии Device Guard требуются следующие аппаратные компоненты и программное обеспечение:

- безопасная загрузка UEFI (в некоторых случаях — с удаленным из базы данных UEFI центром сертификации стороннего производителя);
- включенная по умолчанию поддержка виртуализации в системах BIOS;
- расширения виртуализации (например, Intel VT-x и AMD RVI);
- SLAT — преобразование адресов второго уровня (например, Intel EPT и AMD RVI);
- IOMMU — блок управления памятью для операций ввода-вывода (например, Intel VT-d, AMD-Vi).

В UEFI BIOS должно быть настроено предотвращение доступа неавторизованных пользователей к отключению аппаратных функций безопасности, от которых зависит Device Guard (например, безопасной загрузки).

Драйверы режима ядра должны быть подписаны и совместимы с принудительной проверкой целостности низкоуровневой оболочки.

Для развертывания HVCI можно использовать групповую политику. Рекомендуется включить HVCI на всех серверах Windows Server 2016. Более подробная информация о конфигурации групповой политики доступна по адресу https://technet.microsoft.com/itpro/windows/keep-secure/deploy-device-guard-enable-virtualization-based-security.

Развертывание настраиваемой политики целостности кода

Вредоносные программы обычно не подписаны. Простое развертывание политик целостности кода обеспечивает защиту от неподписанных вредоносных программ, являющихся источником подавляющего большинства атак. С помощью политик целостности кода также можно указать, какие именно двоичные файлы разрешено запускать в пользовательском режиме и в режиме ядра. При полном применении политики допускается загрузка только конкретных приложений или только программ с определенными подписями. Эта возможность сама по себе фундаментальным образом меняет уровень безопасности корпоративной ИТ-среды.

Можно запускать настраиваемые политики целостности кода независимо от HVCI, что позволяет использовать их на устройствах, не соответствующих аппаратным требования HVCI.

С помощью настраиваемых политик целостности кода администраторы могут точно выбрать уровень доверия для программного обеспечения на сервере, от возможности подписания программы надежными издателями (например, корпорацией Microsoft) до совпадения с хэшем конкретного файла.

Политики целостности кода рекомендуется всегда сначала развертывать в режиме аудита: это даст возможность отследить, какие двоичные файлы не смогут загрузиться при включенной политике. В этом случае можно будет правильно настроить политику, устранить возможные неполадки, а уже потом запустить политику в режиме применения.

В этом разделе мы поговорим о двух распространенных типах политик целостности кода: для обычных серверов и для защищенных серверов.

- **Обычные серверы** — это серверы, на которых выполняются различные рабочие нагрузки и иногда устанавливаются новые программы. Такие серверы используются в достаточно гибком режиме.
- **Защищенные серверы** — это серверы, предназначенные для выполнения определенной важной рабочей нагрузки с высокой надежностью, например узлы Hyper-V или контроллеры домена.

Создание политики целостности кода для обычных серверов

Чтобы создать политику целостности кода, нужно настроить эталонный сервер на стандартном оборудовании и установить на него все программы, которые должны быть на таких серверах. Затем необходимо выполнить следующий командлет:

```
New-CIPolicy -Level Publisher -Fallback Hash -UserPEs -FilePath C:\CI\Publisher.xml
```

> **Примечание.** Более подробная информация о параметре level доступна по адресу https://technet.microsoft.com/itpro/powershell/windows/configci/new-cipolicy

Этот командлет создает политику следующим образом: он анализирует файлы на сервере, извлекает из файлов информацию об издателях и добавляет эту информацию в политику. Политика создается в режиме аудита: в таком режиме файлы, на которые политика целостности кода не распространяется, также смогут загружаться, при этом они будут записаны в канал журнала событий Microsoft\Windows\CodeIntegrity. Для выявления угроз безопасности администраторы могут проверять журналы.

В ходе обычной работы администраторы имеют возможность получать обновления программ, а также, возможно, добавлять другие программы тех же поставщиков программного обеспечения. Поскольку у этих обновлений и у добавляемых программ издатель остается прежним, обновлять политику целостности кода не нужно.

Одну и ту же политику целостности кода можно развертывать на серверах одной категории, использующих одинаковое оборудование.

Создание политики целостности кода для защищенных серверов

Этот процесс аналогичен созданию политики целостности кода для обычной категории серверов, но с другим уровнем контроля доверенного программного обеспечения. Для серверов этого типа рекомендуется использовать значение параметра -Level FilePublisher, чтобы на сервере можно было загружать только разрешенные файлы. Для создания политики целостности кода запустите следующий командлет:

```
New-CIPolicy -Level FilePublisher -Fallback Hash -UserPEs -FilePath C:\CI\FilePublisher.xml
```

Этот командлет создает политику следующим образом: он анализирует файлы на сервере и создает список безопасных файлов, добавляя в политику информацию об имени файлов, их версии и издателе. Доверенными считаются только те файлы, которые перечислены в списке безопасных. При этом должны совпадать имя файла и наименование издателя, а версия может совпадать с указанной или быть более поздней. У файлов, на которые распространяется эта политика, при обновлении программного обеспечения будет более поздний номер версии, поэтому создавать политику заново не потребуется. Если на сервер добавляются новые файлы, их нужно будет проанализировать и добавить информацию о них в существующую политику.

Предыдущий командлет создает политику в режиме аудита, поэтому сначала можно проверить политику и убедиться в том, что она распространяется на все доверенные файлы. Удостоверившись, что политика правильно настроена, можно переключить ее в режим применения с помощью следующего командлета:

```
Set-RuleOptions -FilePath C:\CI\FilePublisher.xml -Option 3 -delete
```

Развертывание политики целостности кода

XML-файл, созданный командлетом New-CIPolicy, пока еще не готов к использованию системой. Для развертывания политики нужно будет преобразовать этот файл в двоичный формат и скопировать в папку CodeIntegrity внутри папки System32.

Для преобразования XML-файла запустите следующий командлет:

```
ConvertFrom-CIPolicy C:\CI\FilePublisher.xml C:\CI\FilePublisher.bin
```

Разверните политику целостности кода:

```
Copy-Item C:\CI\FilePublisher.bin C:\Windows\System32\CodeIntegrity\SiPolicy.p7b
```

После этого перезагрузите сервер, чтобы служба целостности кода загрузила политику.

> **Примечание.** Более подробная информация о том, как приступить к использованию политик целостности кода, а также о создании политик аудита и их развертывании с помощью групповой политики доступна по адресу https://docs.microsoft.com/en-us/windows/device-security/device-guard/deploy-code-integrity-policies-policy-rules-and-file-rules.

Credential Guard

С помощью технологий виртуализации Credential Guard изолирует секреты пользователей таким образом, что доступ предоставляется только привилегированным системам. Credential Guard обладает следующими возможностями:

- **Аппаратная защита:** повышение безопасности получаемых учетных данных домена за счет использования средств обеспечения безопасности платформы, включая безопасную загрузку и виртуализацию.
- **Безопасность на основе виртуализации:** службы Windows, управляющие полученными учетными данными домена и другими секретами, работают в защищенной среде, изолированной от работающей операционной системы.
- **Улучшенная защита от усовершенствованных постоянных угроз:** защита полученных учетных данных домена при помощи средств безопасности на основе виртуализации. В этом случае блокируются технологии и инструменты кражи учетных данных, используемые во множестве атак. Вредоносные программы, запущенные в операционной системе с правами администратора, не смогут извлечь секреты, защищенные с помощью виртуализации.
- **Управляемость**: управление с помощью групповой политики, инструментария WMI, из командной строки и с помощью Windows PowerShell.

Обычно секреты хранятся в памяти процесса локальной системы безопасности (LSA) в Windows. При использовании Credential Guard процесс LSA обменивается данными с новым компонентом,

который называется *изолированным процессом LSA (LSAIso)*. Он работает на основе виртуализации и недоступен для остальной части операционной системы. На рис. 4-1 показана изоляция, обеспечиваемая безопасностью на основе виртуализации для процесса LSAIso по сравнению с процессом LSASS.

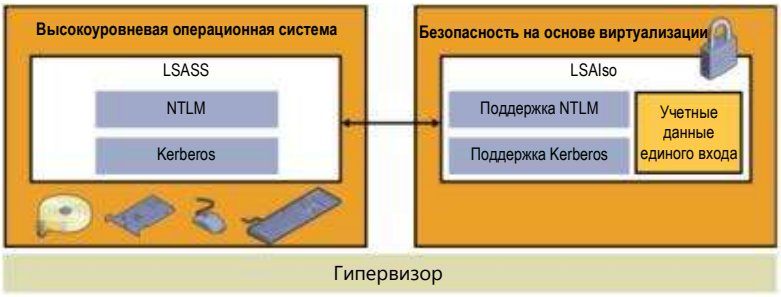


Рис. 4-1. Процесс LSA на основе виртуализации.

Если защита Credential Guard включена, устаревшие варианты NTLM и Kerberos (NTLM v1, MS-CHAPv2 и пр.) не поддерживаются.

Технология Credential Guard работает на основе виртуализации, поэтому для ее использования требуется определенная аппаратная поддержка. Некоторые из требований перечислены в табл. 4-1.

Таблица 4-1.

Требование	Описание
Windows Server 2016	Решение Credential Guard доступно во всех выпусках Windows Server 2016, кроме Nano Server (поскольку в Nano Server поддерживается только удаленное управление).
Микропрограмма UEFI версии 2.3.1 или более поздней и безопасная загрузка	Проверить, что микропрограмма использует UEFI 2.3.1 или более позднюю версию и безопасную загрузку можно, выполнив проверку соответствия системным требованиям System.Fundamentals.Firmware.CS.UEFISecureBoot.ConnectedStand
Расширения виртуализации	Для поддержки безопасности на основе виртуализации требуются следующие расширения виртуализации: • Intel VT-x или AMD-V; • преобразование адресов второго уровня (SLAT).
Архитектура x64	Компоненты, используемые средствами безопасности на основе виртуализации в низкоуровневой оболочке Windows, поддерживаются только на 64-разрядных системах.
Блок управления памятью для операций ввода-вывода (IOMMU) VT-d или AMD-Vi	Блок IOMMU повышает устойчивость системы к атакам, направленным на память.
TPM версии 1.2 или 2.0	Если модуль TPM не установлен, решение Credential Guard будет работать, при этом ключи, используемые для шифрования в Credential Guard, не будут защищены модулем TPM.
Обновленная микро-программа с поддержкой безопасного бита MOR	Для предотвращения определенных атак, нацеленных на память, требуется поддержка безопасного бита MOR.
Физический компьютер или виртуальная машина	Credential Guard поддерживается как на физических компьютерах, так и на виртуальных машинах. На виртуальной машине низкоуровневая оболочка должна поддерживать вложенную виртуализацию.

Самый простой способ использовать Credential Guard — включить это решение с помощью групповой политики и назначить компьютеры, к которым оно будет применено.

Для этого создайте в консоли управления групповыми политиками новую групповую политику или измените существующую. Перейдите в раздел «Конфигурация компьютера > Административные шаблоны > Система > Device Guard».

Дважды нажмите левую кнопку мыши на поле «Включить средство обеспечения безопасности на основе виртуализации», в отобразившемся окне выберите «Включено» (рис. 4-2). В списке «Выберите уровень безопасности платформы» выберите элементы «Безопасная загрузка» или «Безопасная загрузка и защита DMA». В списке «Конфигурация Credential Guard» выберите элемент «Включено с блокировкой UEFI» и нажмите на кнопку OK. Если нужна возможность удаленного отключения Credential Guard, в списке «Конфигурация Credential Guard» выберите элемент «Включено без блокировки» вместо «Включено с блокировкой UEFI».

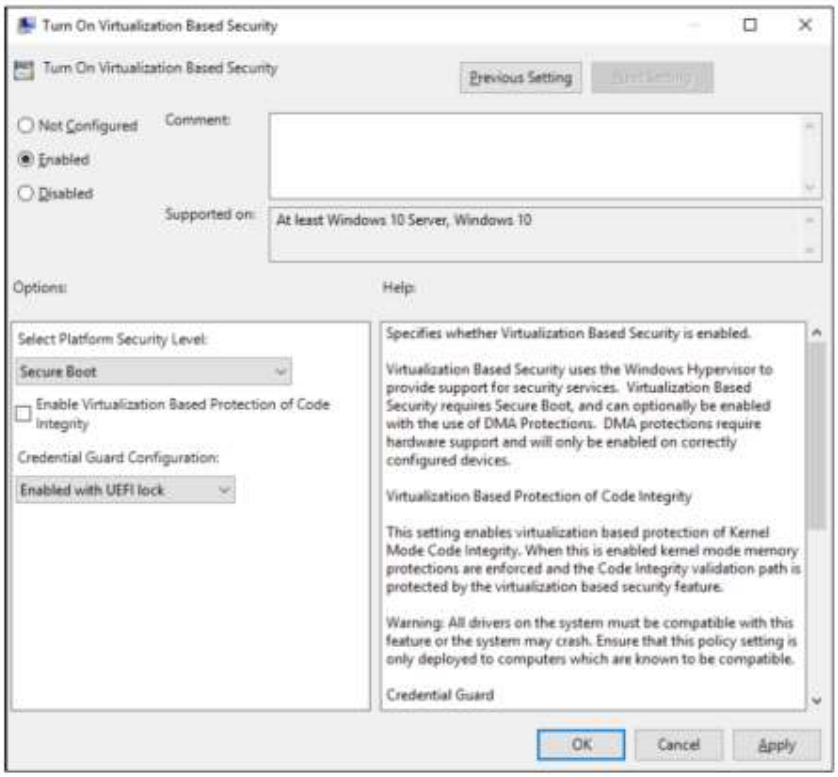


Рис. 4-2. Параметры групповой политики для Credential Guard

> **Примечание.** Более подробная информация доступна по адресу https://technet.microsoft.com/library/mt483740(v=vs.85).aspx.

Remote Credential Guard

Remote Credential Guard обеспечивает защиту от кражи учетных данных при удаленном подключении к системе с помощью сеанса удаленного рабочего стола.

Когда пользователь пытается установить подключение удаленного рабочего стола к узлу, запрос Kerberos для проверки подлинности перенаправляется на исходный узел. Учетные данные на удаленном узле отсутствуют. Если на удаленном узле (то есть на компьютере конечного пользователя или сервере) работает вредоносный код, способный получить учетные данные, Remote Credential Guard устранит эту уязвимость благодаря тому, что учетные данные не будут переданы на удаленный узел.

Для использования Remote Credential Guard действуют определенные требования:

- Пользователь должен быть присоединен к тому же самому домену Active Directory, или сервер удаленных рабочих столов должен быть присоединен к домену, обладающему отношением доверия с доменом, в котором находится клиентское устройство.
- На этих системах должна использоваться проверка подлинности Kerberos.
- Необходимо использовать операционную систему Windows 10 сборки 1607 или более поздней либо Windows Server 2016.
- Требуется классическое приложение «Удаленный рабочий стол» для Windows. Приложение «Удаленный рабочий стол» универсальной платформы Windows не может работать с удаленным Credential Guard.

Чтобы включить Remote Credential Guard, необходимо настроить его с помощью групповой политики и развернуть в ИТ-среде.

Для настройки Remote Credential Guard с помощью групповой политики откройте консоль управления групповыми политиками и перейдите в раздел «Конфигурация компьютера → Административные шаблоны → Система → Передача учетных данных». Дважды нажмите левую кнопку мыши на элементе «Ограничить делегирование учетных данных удаленным серверам», затем выберите «Требуется Remote Credential Guard». После этого нажмите на кнопку OK и выполните команду gpudpate /force, чтобы применить групповую политику. Remote Credential Guard можно также включить с помощью реестра.

Примечание. Более подробная информация о развертывании Remote Credential Guard доступна по адресу https://technet.microsoft.com/itpro/windows/keep-secure/remote-credential-guard.

Windows Defender

Windows Defender входит в состав Windows Server 2016 и запускается по умолчанию.

Если в организации используется другое решение для защиты от вредоносных программ, Windows Defender можно удалить с помощью Windows PowerShell:

```
Uninstall-WindowsFeature -Name Windows-Server-Antimalware
```

Windows Defender получает обновления посредством центра обновления Windows. Если в организации управление центром обновления Windows осуществляется с помощью средства развертывания обновлений, убедитесь в том, что это средство загружает обновления Windows Defender и актуальных определений вредоносных программ.

Настроить Windows Defender для централизованного управления и администрирования можно с помощью групповой политики.

Технологии обнаружения угроз

Чтобы убедиться в действенности мер защиты среды, независимо от того, какие именно меры используются, необходимо провести аудит. В Windows Server 2016 появились две новые подкатегории аудита, которые дают возможность получить более полную информацию о событиях:

- **Членство в группе аудита**. Такие события относятся к категории событий входа и выхода. События из этой подкатегории записываются при перечислении или запросе членства в группе на компьютере, на котором был создан сеанс входа.
- **PNP-действие аудита**. Эта подкатегория относится к категории «Подробное отслеживание», ее можно использовать в случаях, когда подсистема Plug-and-Play обнаруживает внешнее устройство. В этой категории записывается только успешный аудит.

В Windows Server 2016 реализованы дополнительные изменения, помогающие быстрее обнаруживать и устранять угрозы. Более подробная информация приведена в табл. 4-2.

Таблица 4-2.

Область	Усовершенствования
Политика аудита ядра по умолчанию	В прежних версиях для получения информации о некоторых событиях ядро использовало LSA. В Windows Server 2016 политика аудита событий создания процессов автоматически включается до тех пор, пока из LSA не будет получена фактическая политика аудита. За счет этого расширяются возможности аудита тех служб, которые могут быть запущены до службы LSA.
Системный список управления доступом (SACL) по умолчанию для LSASS.exe	Используемый по умолчанию SACL добавлен к LSASS.exe для регистрации процессов, пытающихся получить доступ к LSASS.exe. Используется SACL L"S:(AU;SAFA;0x0010;;;WD)". Для включения выберите «Конфигурация расширенной политики аудита → Доступ к объектам → Аудит объектов ядра».
Новые поля события входа	Событие входа с идентификатором 4624 обновлено и включает более подробную информацию для более удобного анализа. В событие 4624 добавлены следующие поля: • **MachineLogon** (строка: «да» или «нет») Если учетная запись, использованная для входа на компьютер, является учетной записью компьютера, то это поле имеет значение «да»; в противном случае — значение «нет». • **ElevatedToken** (строка: «да» или «нет») Если учетная запись, использованная для входа на компьютер, обладает правами администратора, то это поле имеет значение «да»; в противном случае — значение «нет». Кроме того, если эти данные являются частью разделенного маркера, также будет показан связанный идентификатор входа (LSAP_LOGON_SESSION). • **TargetOutboundUserName** (строка) и **TargetOutboundUserDomain** (строка) Имя пользователя и домен учетных данных, созданных методом LogonUser для исходящего трафика. • **VirtualAccount** (строка: «да» или «нет») Если учетная запись, использованная для входа на компьютер, является виртуальной учетной записью, то это поле имеет значение «да»; в противном случае — значение «нет». • **GroupMembership** (строка) Список всех групп в маркере пользователя. • **RestrictedAdminMode** (строка: «да» или «нет») Если пользователь входит на компьютер в режиме ограниченного администрирования с удаленного рабочего стола, то это поле имеет значение «да».

Новые поля события создания процесса	Событие создания процесса с идентификатором 4688 обновлено и включает более подробную информацию для повышения удобства анализа. В событие 4688 добавлены следующие поля: • **TargetUserSid** (строка) SID целевого пользователя. • **TargetUserName** (строка) Имя учетной записи целевого пользователя. • **TargetDomainName** (строка) Домен целевого пользователя. • **TargetLogonId** (строка) Идентификатор входа целевого пользователя. • **ParentProcessName** (строка) Имя создающего процесса. • **ParentProcessId** (строка) Указатель на фактический родительский процесс, если он отличается от создающего процесса.
События диспетчера учетных записей безопасности (SAM)	Добавлены новые события SAM для API-интерфейсов SAM, выполняющих операции чтения и запросы. В прежних версиях Windows аудит поддерживался только для операций записи. Новые события получили идентификаторы 4798 и 4799. Теперь осуществляется аудит следующих API-интерфейсов: • SamrEnumerateGroupsInDomain; • SamrEnumerateUsersInDomain; • SamrEnumerateAliasesInDomain; • SamrGetAliasMembership; • SamrLookupNamesInDomain; • SamrLookupIdsInDomain; • SamrQueryInformationUser; • SamrQueryInformationGroup; • SamrQueryInformationUserAlias; • SamrGetMembersInGroup; • SamrGetMembersInAlias; • SamrGetUserDomainPasswordInformation.
События базы данных конфигурации загрузки (BCD)	Добавлено событие с идентификатором 4826 для отслеживания следующих изменений BCD: • параметры DEP/NEX; • тестовая подпись; • моделирование PCAT SB; • отладка; • отладка загрузки; • службы целостности; • отключение меню отладки Winload.
События PNP	Для отслеживания подключения внешних устройств с помощью Plug-and-Play добавлено событие с идентификатором 6416 . Основной сценарий его использования — подключение внешнего устройства, содержащего вредоносные программы, к особо важному компьютеру, на котором такие действия не предусмотрены, например к контроллеру домена.

Защита привилегированного доступа

В этом разделе мы поговорим о принципах, касающихся защиты привилегированного доступа. Сначала рассмотрим принципы администрирования с ограничением по времени (Just-in-Time, JIT) и по области воздействия (Just Enough Administration, JEA). Затем мы расскажем, как объединить все инструменты и технологии, описанные в этой главе, в стратегию реализации для вашей организации.

Администрирование с ограничением по времени (JIT) и по области воздействия (JEA)

Администрирование с ограничением по времени устроено очень просто. Суть такого подхода в том, что не существует постоянных администраторов, точнее, не существует учетных записей, которые всегда обладают правами администратора. Используется простой процесс, при котором необходимые права запрашиваются непосредственно перед тем, как они становятся действительно нужны. Запрос прав одобряется, после чего права на определенный период времени предоставляются запросившей их учетной записи. Благодаря этому удается выполнить нужную задачу с нужным объемом прав за назначенное время. Для предоставления нужного набора прав администрирование JIT работает совместно с ограничением по области воздействия. В Windows Server 2016 эти две технологии объединены под названием Privileged Access Management (PAM).

> **Примечание.** Более подробная информация о PAM доступна по адресу https://technet.microsoft.com/library/dn903243.aspx.

Теперь вкратце рассмотрим JEA. Это компонент пакета Windows Management Framework 5.0, он поддерживается с версии Windows Server 2008 R2. С помощью JEA можно назначить учетной записи пользователя определенные права — только необходимый набор прав для выполнения нужной функции, поэтому нет необходимости назначать пользователю права администратора и при этом помнить, что потом эти права нужно будет отозвать. Администрирование с ограничением области воздействия (JEA) предоставляет возможности ролевого управления доступом (RBAC), востребованного в современных компаниях для повышения безопасности.

Для использования JEA в какой-либо системе требуется файл конфигурации сеанса Windows PowerShell. Для создания файла с расширением .PSSC, который нужен для управления доступом, используйте командлет New-PSSessionConfigurationFile со следующим синтаксисом:

```
New-PSSessionConfigurationFile -Path "$env:Programdata\<папка конфигурации JEA>\<имя файла>.pssc"
```

Ниже приведен пример файла конфигурации по умолчанию, создаваемого этой командой:

```
@{
# Номер версии схемы, используемой для этого документа, SchemaVersion = '2.0.0.0'
# Уникальный идентификатор этого документа GUID = '1da190ce-fc94-4f8b-98e0-7d70fd9154b1'
# Автор этого документа Author = 'john'
# Описание функций, обеспечиваемых этими параметрами, Description = ''
# Применимые значения типа сеанса по умолчанию для этой конфигурации сеанса. Возможные значения:
«RestrictedRemoteServer» (рекомендуется), «Empty» или «Default». SessionType = 'Default'
# Папка для сохранения записей сеанса для этой конфигурации сеанса TranscriptDirectory =
'C:\Transcripts\'
# Нужно ли выполнять эту конфигурацию сеанса в качестве учетной записи администратора компьютера
(виртуального) RunAsVirtualAccount = $true
# Группы, связанные с (виртуальной) учетной записью администратора на компьютере,
RunAsVirtualAccountGroups = 'Remote Desktop Users', 'Remote Management Users'
# Сценарии, запускаемые при применении к сеансу, ScriptsToProcess = 'C:\ConfigData\InitScript1.ps1',
'C:\ConfigData\InitScript2.ps1'
# Роли пользователей (группы безопасности) и возможности роли, которые должны быть применены к ним
при применении к сеансу, RoleDefinitions = @{ 'CONTOSO\SqlAdmins' = @{ RoleCapabilities =
'SqlAdministration' }; 'CONTOSO\ServerMonitors' = @{ VisibleCmdlets = 'Get-Process' } }}
```

Главный интересующий нас параметр — SessionType, для которого установлено значение «Default». Чтобы использовать JEA, нужно установить для этого параметра значение «RestrictedRemoteServer». Затем нужно раскомментировать строку # RunAsVirtualAccount = $True, благодаря чему у сеанса будут права «виртуального» администратора. Наконец, нужно изменить раздел Roledefintions в соответствии с вашей средой и раскомментировать его.

После создания и настройки файла конфигурации нужно зарегистрировать его с помощью командлета Register-PSSessionConfiguration.

```
Register-PSSessionConfiguration -Name <имя> -Path "$env:Programdata\<папка конфигурации JEA>\<имя файла>.pssc"
```

Для проверки можно установить подключение к компьютеру, как при обычном удаленном сеансе Windows PowerShell.

```
Enter-PSSession -ComputerName <имя компьютера> -ConfigurationName <имя конфигурации JEA>
-Credential $cred
```

Можно создать еще один файл — файл возможности роли с расширением .PSRC. Этот файл можно использовать, чтобы определить команды и приложения, доступные для заданных ролей. Для создания пустого шаблона можно использовать командлет New-PSRoleCapabilityFile.

В результате получим файл, содержащий раздел, в котором можно указать, какие модули следует импортировать, какие функции и командлеты следует сделать доступными.

```
# ModulesToImport = 'MyCustomModule',
@{ ModuleName = 'MyCustomModule2'; ModuleVersion = '1.0.0.0';
GUID = '4d30d5f0-cb16-4898-812d-f20a6c596bdf'}
# VisibleFunctions = 'Invoke-Function1', @{ Name = 'Invoke-Function2'; Parameters = @{ Name =
'Parameter1'; ValidateSet = 'Item1', 'Item2' },
@{ Name = 'Parameter2'; ValidatePattern = 'L*' } }
# VisibleCmdlets = 'Invoke-Cmdlet1',
@{ Name = 'Invoke-Cmdlet2'; Parameters = @{ Name = 'Parameter1'; ValidateSet = 'Item1', 'Item2' },
@{ Name = 'Parameter2'; ValidatePattern = 'L*' } }
```

> **Примечание.** JEA — достаточно сложная область, поэтому здесь мы можем привести лишь краткое описание. Более подробная информация и список всех параметров конфигурации доступны по адресу http://aka.ms/JEA.

Стратегия защиты привилегированного доступа

Сколь бы надежно ни была защищена операционная система или служба, она безопасна лишь настолько, насколько безопасен самый нестойкий пароль. Предположим, у вас хранятся самые конфиденциальные в мире данные, вы зашифровали их с помощью самого совершенного в мире алгоритма шифрования, но использовали при этом пароль вида «Password01», в результате чего использование целого набора технологий безопасности потеряло всякий смысл.

Рассмотрим другой сценарий. Пройдите по офису и обратите внимание на то, у скольких пользователей пароль записан на бумажках, приклеенных к клавиатуре или монитору. Также посмотрите, у скольких пользователей на столах стоят фотографии их родственников или домашних животных. Когда этим пользователям нужно придумать пароль, какова вероятность того, что он будет личным, связанным с этими фотографиями?

Рассмотрим еще один сценарий: атака на основе социальной инженерии. При атаке такого рода (это, к слову, основная причина нарушения безопасности) злоумышленник неожиданно звонит пользователю, выдает себя за сотрудника ИТ-отдела и заявляет, что ему по какой-то причине нужно проверить данные учетной записи. Если атакующий хорошо владеет своим ремеслом, то с высокой вероятностью его жертва с готовностью предоставит запрошенную информацию.

Таким образом, злоумышленник получает доступ к каким-либо данным и теоретически может использовать его для дальнейшего развития атаки. Ситуация усугубится, если это была не обычная учетная запись, а привилегированная, то есть с расширенными правами.

Защита привилегированного доступа не обеспечивается какой-либо одной технологией, это целый набор методик, которые можно внедрить для повышения общего уровня безопасности. Внедрить и проверить все политики, связанные с безопасностью, а также провести необходимую подготовку, чтобы ознакомить сотрудников с потенциальными уязвимыми местами, будет полезно всем организациям.

Если у пользователей есть доступ к какой-либо сети, то такая сеть никогда не будет безопасной на 100%. Следует надежно защитить привилегированный доступ к системам и сетям, начав со следующих основных мер:

- **Обновления.** Развертывание обновлений на контроллерах домена в течение семи дней после выпуска.
- **Удаление пользователей из группы локальных администраторов.** Отслеживание пользователей и их удаление из группы локальных администраторов, если этим пользователям не нужны такие права доступа. Для централизованного управления членством пользователей в группах при необходимости можно использовать Active Directory.
- **Базовые политики безопасности.** Развертывание политик, образующих стандартную конфигурацию для организации. Разумеется, будут и исключения (в зависимости от приложений и определенных требований), но их необходимость следует регулярно проверять, чтобы поддерживать всю систему в как можно более безопасном состоянии.
- **Антивирусные программы.** Регулярное обновление антивирусных программ и регулярные проверки среды. Очистка и устранение угроз непосредственно после обнаружения.
- **Журналы и анализ.** Запись информации о безопасности, регулярный просмотр, выявление аномалий в журналах. Дальнейший анализ каждого обнаруженного элемента с целью определения его источника и оценки безопасности риска.
- **Развертывание и инвентаризация программного обеспечения.** Управление программным обеспечением, установленным в среде, крайне важно, для того чтобы не дать пользователям установить вредоносные программы. Необходимо знать, какие программы установлены, и поддерживать их список, чтобы отслеживать изменения в системах.

От этих основных мер перейдем к более подробному описанию стратегии защиты привилегированного доступа. Следует понимать, что выработать и внедрить такую стратегию за один день невозможно. Ее внедрение следует осуществлять постепенно и последовательно, чтобы принятая в организации практика успела измениться и приспособиться к новым принципам.

При этом, как и при разработке любой стратегии, целесообразно установить краткосрочные, среднесрочные и долгосрочные цели. В табл. 4-3 перечислены цели и сроки, а также области работ по каждой из целей, которых следует достичь.

Таблица 4-3.

Цель	Срок	Описание
Краткий срок	План на 2–4 недели	Быстрое устранение наиболее часто применяемых атак
Средний срок	План на 1–3 месяца	Достижение прозрачности и контроля деятельности администрирования
Долгий срок	План на 6 месяцев и более	Формирование профилактического подхода к безопасности

Краткосрочный план

Для достижения краткосрочных целей в любой организации важно предотвращать наиболее часто используемые атаки, чтобы таким образом обеспечить основы безопасности.

Для этого одной из первых мер должно быть *разделение обязанностей*. Это означает, что если требуется выполнить задачу, для которой нужен расширенный доступ, то в организации должна

быть учетная запись с расширенными правами для выполнения этой задачи. Учетным записям обычных пользователей ни в коем случае нельзя предоставлять расширенный доступ в сети для выполнения каких-либо задач. Эти учетные записи всегда должны рассматриваться именно как принадлежащие обычным пользователям. Для привилегированной учетной записи, созданной для выполнения задач с расширенными правами, можно проводить более подробный аудит и тщательнее отслеживать действия. Поскольку вы поддерживаете отдельный набор учетных данных для этой учетной записи с более строгими требованиями, вы сможете отразить атаку, если учетная запись обычного пользователя окажется скомпрометирована.

Настройка мер безопасности для учетной записи администратора раньше проводилась при развертывании, после чего все обычно оставалось без изменений. Чаще всего пароль сохранялся неизменным в течение всего срока эксплуатации рабочих станций, что в случае его утечки приводило к крайне серьезным проблемам. Но если не использовать одинаковый пароль администратора для всех компьютеров в ИТ-среде, то возникает другая проблема: придется помнить уникальный пароль для каждой рабочей станции. Для управления паролями локального администратора как для рабочих станций, так и для серверов корпорация Microsoft предлагает решение под названием Local Administrator Password Solution (LAPS).

LAPS позволяет создать уникальный пароль для каждого сервера и каждой рабочей станции и сохранить эти пароли в Active Directory в качестве конфиденциального атрибута в объекте-компьютере. К объектам будет применен соответствующий список управления доступом, поэтому обращаться к паролям и получать их смогут только те учетные записи, которым это разрешено. Более подробная информация о LAPS доступна по адресу http://aka.ms/LAPS.

Конечная часть краткосрочного плана посвящена созданию рабочих станций с привилегированным доступом (PAW). PAW — это надежно защищенные рабочие станции, созданные специально в качестве управляемых точек администрирования с целью укрепления безопасности систем. Чтобы свести к минимуму уязвимые участки, рабочие станции PAW лишены доступа к Интернету и к небезопасным ресурсам. Вход на рабочие станции PAW разрешен лишь ограниченному числу полномочных пользователей, что, в свою очередь, повышает устойчивость защищенной части сетей к атакам. Более подробная информация о рабочих станциях PAW доступна по адресу http://aka.ms/CyberPAW.

На рис. 4-3 показаны действия в рамках краткосрочного плана.

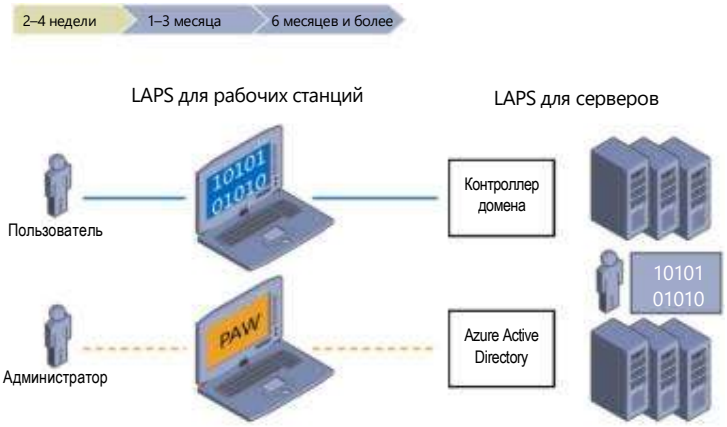


Рис. 4-3. Краткосрочный план

На этом рисунке показаны четыре отдельные области:

1. Создание отдельной учетной записи администратора для административных задач (на рисунке — «Администратор»).
2. Развертывание рабочих станций PAW для администраторов Active Directory. Более подробная информация доступна по адресу http://aka.ms/cyberPAW, где этот шаг обозначен как «Этап 1».

3. Создание уникальных паролей для рабочих станций с использованием LAPS. Более подробная информация доступна по адресу http://aka.ms/LAPS.
4. Создание уникальных паролей для серверов с использованием LAPS. Более подробная информация доступна по адресу http://aka.ms/LAPS.

Среднесрочный план

Первое, что нужно сделать в среднесрочном плане, — расширить развертывание рабочих станций PAW, чтобы охватить больше систем, которыми можно управлять только с этих рабочих станций.

После этого нужно сосредоточить усилия на реализации ограничений прав доступа по времени: пользователь может запросить права доступа, которые действуют только в течение заранее заданного периода времени. Это означает, что можно обойтись без отдельных администраторов как таковых: пользователи получат возможность запросить нужные им права доступа и после подтверждения таких запросов выполнять необходимые задачи. Такой подход работает на основе диспетчера удостоверений Microsoft, а нужные функции предоставляются подсистемой администрирования с ограничением по области воздействия (JEA).

Чтобы дополнительно оградить системы от атак, следует также внедрить многофакторную проверку подлинности для привилегированного доступа. Для этого можно использовать средства безопасности на основе токенов, обратные вызовы или смарт-карты. После этого можно приступить к внедрению JEA. Принцип работы JEA очень прост: учетной записи, которой нужно выполнить какое-либо действие, предоставляется минимальный набор прав, необходимый для выполнения этого действия. Более подробно мы поговорим о JEA далее в этой главе.

Следующий шаг — укрепление защиты контроллеров домена, для чего потребуется реализовать обнаружение угроз с помощью средства расширенной аналитики угроз (ATA). ATA дает возможность выявлять аномальное поведение в работе систем и быстро оповещает о таких ситуациях. Для этого формируются профили поведения пользователей и шаблоны их обычной работы. Если какие-либо действия пользователя отклоняются от обычного шаблона, ATA вас оповестит. Впрочем, это краткое и упрощенное описание не в полной мере отражает настоящие возможности ATA: на самом деле это гораздо более сложное и совершенное решение. Более подробная информация доступна по адресу http://aka.ms/ata.

На рис. 4-4 показана схема среднесрочного плана.

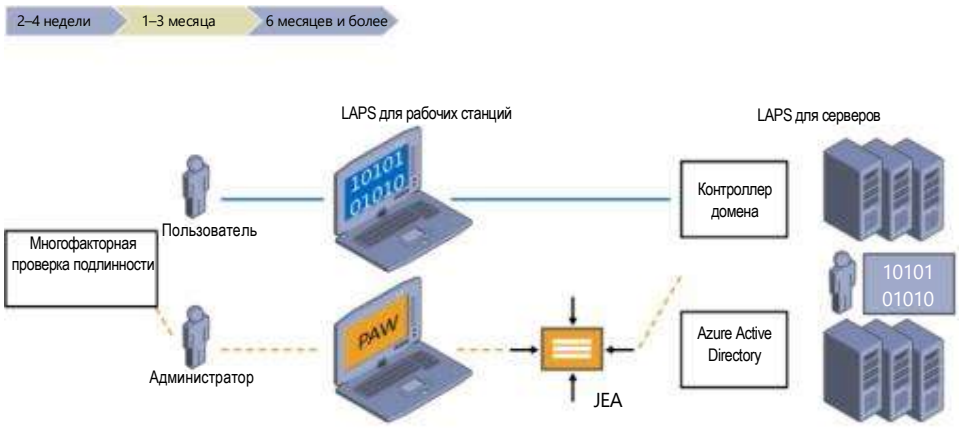


Рис. 4-4. Среднесрочный план

На этом рисунке показано шесть отдельных областей:

1. Увеличение количества рабочих станций PAW, предоставление их всем администраторам для использования дополнительных защитных мер, таких как Credential Guard и ограничен-

ное администрирование RDP. Более подробная информация доступна по адресу http://aka.ms/CyberPAW, где этот шаг соответствует этапам 2 и 3.

2. Внедрение механизмов, предоставляющих расширенные права с ограничением по времени (отсутствие постоянных администраторов). Более подробная информация доступна по адресу http://aka.ms/AzurePIM.
3. Внедрение многофакторной проверки при повышении прав. Более подробная информация доступна по адресу http://aka.ms/PAM.
4. Использование JEA для обслуживания контроллеров домена. Более подробная информация доступна по адресу http://aka.ms/JEA.
5. Ограничение уязвимых мест в доменах и контроллерах домена. Более подробная информация доступна по адресу http://aka.ms/HardenAD.
6. Внедрение средств обнаружения атак для серверов и контроллеров домена. Более подробная информация доступна по адресу http://aka.ms/ata.

Долгосрочный план

Цели долгосрочного плана (рис. 4-5) относятся к заключительной части постоянно развивающейся стратегии. Защита ИТ-среды — постоянный процесс, который не прекращается никогда. Со временем стратегию безопасности необходимо пересматривать и приспосабливать к новым обстоятельствам, но перечисленные здесь действия помогут создать основу для дальнейшей работы.

Как и при разработке программного обеспечения, в отношении управления доступом к ресурсам необходимо использовать понятие жизненного цикла. Подход должен основываться на самых современных принципах и использовать JEA. Всем администраторам должны быть предоставлены механизмы проверки подлинности повышенной надежности, например с использованием смарт-карт или паспорта.

Существенного укрепления защиты можно добиться, создав безопасный лес, изолированный от леса обычных пользователей. Здесь можно разместить самые безопасные системы вашей среды, полностью отделив их от рабочей сети. Следующий раздел посвящен защите целостности кода: в системах будет разрешено запускать только авторизованный код.

Наконец, можно использовать новую возможность Hyper-V в Windows Server 2016 — экранированные виртуальные машины. Эта функция поддерживается для виртуальных машин второго поколения, используется шифрование виртуальных машин. В этом случае следует сначала сосредоточить усилия на контроллерах домена, чтобы злоумышленники не смогли просмотреть виртуальную машину и скопировать ее с дисков либо атаковать несущий узел для получения доступа к виртуальным машинам. Экранированные виртуальные машины более подробно описываются в этой главе далее.

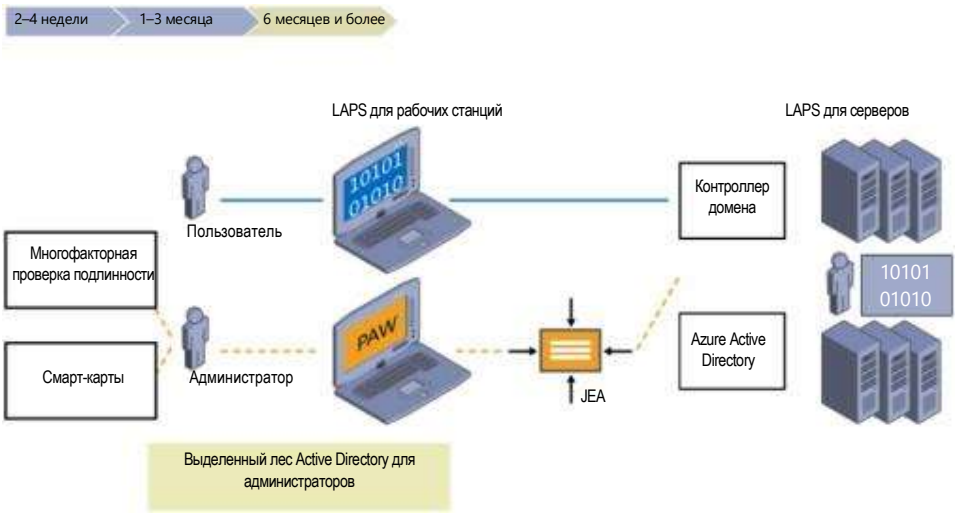


Рисунок 4-5. Долгосрочный план.

На этом рисунке показаны следующие области:

1. Модернизация ролей и модели делегирования.
2. Внедрение проверки подлинности по смарт-картам или паспорту для всех администраторов (http://aka.ms/passport).
3. Создание в Active Directory отдельного леса для администраторов (http://aka.ms/ESAE).
4. Внедрение политики целостности кода для контроллеров домена в Windows Server 2016.
5. Использование экранированных виртуальных машин для контроллеров домена в Windows Server 2016 и в структуре Hyper-V (http://aka.ms/shieldedvms).

Идентификация

Настало время поговорить о некоторых других элементах, также касающихся области безопасности, — об улучшениях в наборе технологий удостоверений в Windows Server 2016.

Доменные службы Active Directory

В новой версии основные усовершенствования были сделаны в трех основных областях:

- Privileged Access Management;
- присоединение к Azure Active Directory;
- Microsoft Passport.

Давайте подробнее рассмотрим каждую из этих областей и узнаем о новых возможностях, появившихся в новой версии платформы.

Управление привилегированным доступом (PAM)

Киберугрозы с каждым днем становятся все более совершенными и изощренными, в большинстве случаев их крайне сложно обнаружить, поэтому для устранения всех возможных уязвимых мест необходимо применять меры безопасности на всех уровнях. Решение для управления привилегированным доступом, получившее название Privileged Access Management (PAM), было создано для борьбы с распространенными угрозами кражи учетных данных на основе фишинга, получения хэша и т. п. Для использования PAM необходимо развернуть Microsoft Identify Manager (MIM).

Примечание. Описание и информация о развертывании MIM доступны по адресу https://aka.ms/vaz62m.

Для большинства сред Active Directory принято считать, что вредоносная деятельность полностью отсутствует, но на практике в этом не может быть полной уверенности. Поэтому одним из новых механизмов безопасности, реализованных в PAM, стал новый лес со средой бастиона, для которого гарантируется полное отсутствие вредоносных действий. Устанавливается новый тип отношений доверия — доверие PAM. В ходе начального развертывания MIM подготавливает этот лес со средой бастиона. На рис. 4-6 показан принцип устройства нового леса и доверия PAM.

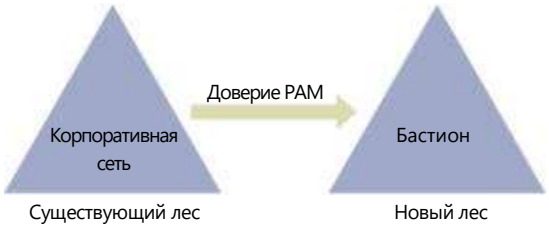


Рис. 4-6. Новый лес со средой бастиона и доверие PAM

PAM предоставляет возможность изолировать использование учетных записей с расширенными правами путем хранения их в этом лесу со средой бастиона, благодаря чему злоумышленникам становится сложнее получать привилегированный доступ. MIM предоставляет пользователям методы для безопасного запроса и получения прав администрирования, когда такие права необходимы. После «одобрения» рабочими процессами MIM в лесу со средой бастиона создается теневой субъект безопасности. Эти теневые субъекты безопасности «привязываются» с помощью ссылки, хранящейся в атрибуте Active Directory, который указывает на SID привилегированной группы в исходном лесу.

Пользователи могут запросить привилегированный доступ следующими способами:

- Веб-API служб MIM;
- конечная точка REST;
- Windows PowerShell (командлет New-PAMRequest).

Эти простые методы можно интегрировать в другие средства, такие как Runbook автоматизации и системы отправки запросов, — это позволит получить дополнительный контроль над процессом.

Ранее в этой главе мы описали устройство и принципы работы администрирования JIT и JEA. Реализовать эти технологии в вашей среде можно с помощью PAM. Подобно JIT и JEA, решение PAM предоставляет ограниченные по времени права запросившей их учетной записи и, разумеется, связывает эту учетную запись с привилегированной группой, обладающей нужными для выполнения требуемой задачи разрешениями.

Также можно настроить время жизни билета Kerberos, чтобы значение срока жизни (TTL) было как можно меньше. В этом случае, когда вы входите в систему и получаете билет Kerberos, его время жизни будет связано с периодом, оставшимся от общего времени, на которое решение PAM предоставило вам доступ к привилегированной группе.

PAM также поддерживает множество новых средств мониторинга, дающих возможность получать более подробную информацию о том, кто запросил доступ, какой тип доступа был фактически предоставлен, а также, что важнее, какие действия были выполнены пользователем, получившим доступ, в течение того времени, пока он обладал привилегированным доступом.

Эту информацию можно просматривать в MIM или в программе «Просмотр событий». Если в среде используются System Center Operations Manager 2012 и службы сбора аудита (ACS), можно создавать наглядное представление этой информации. В будущем наглядное

отображение подобных данных будет также поддерживаться в других сторонних программах и в пакете Operations Management Suite (OMS).

Присоединение к Azure Active Directory

По мере того как компании внедряют облачные решения, а работники становятся все более мобильными, возникает необходимость управлять ИТ-средой, практически не соприкасающейся с корпоративной сетью, вследствие чего появляются очевидные затруднения. Может возникнуть и ряд других проблем — например, как предоставить доступ к ресурсам организации стороннему устройству. Впрочем, какими бы ни были трудности, функция присоединения к домену Azure Active Directory (Azure AD) в Windows Server 2016 предоставляет новые способы управления удостоверениями и новые возможности, доступные как для корпоративных, так и для личных устройств. На рис. 4-7 показаны возможности присоединения к Azure AD.

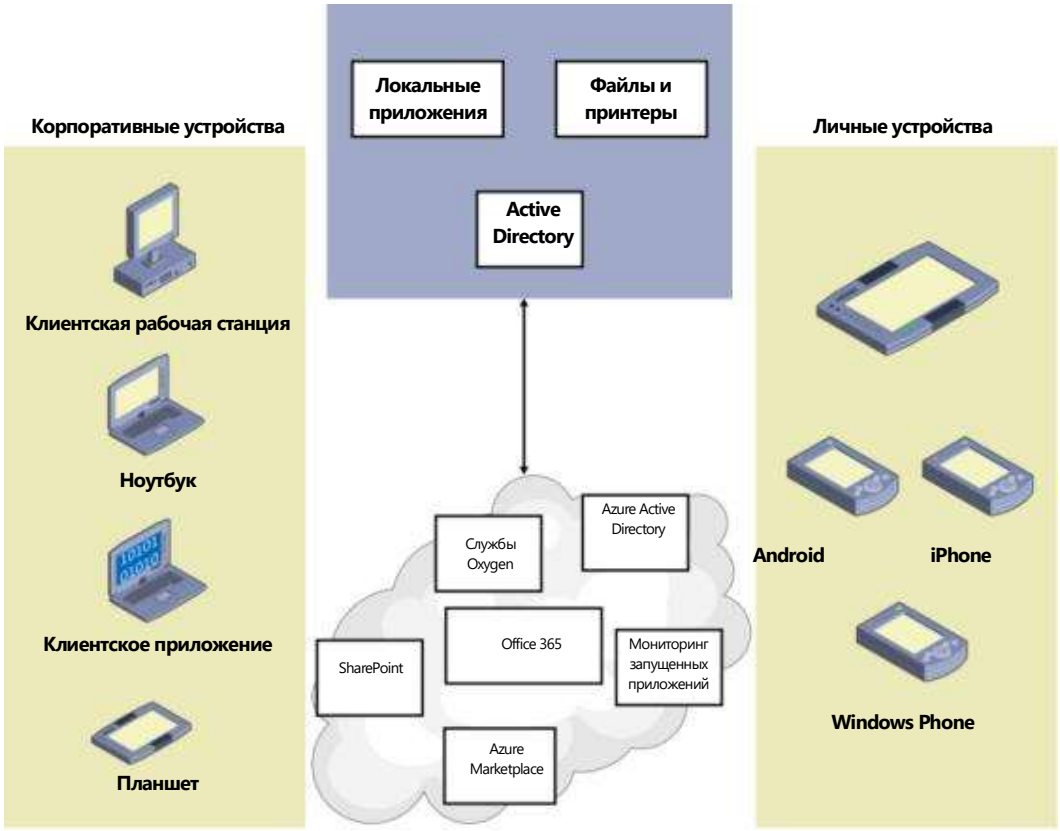


Рис.4-7. Присоединение к Azure AD

Вот некоторые возможности, имеющиеся в Azure AD уже сейчас:

- Доступность современных настроек.

  С любого устройства, подключенного к домену Windows или присоединенного к клиенту Azure AD, можно, используя свои корпоративные учетные данные, получить доступ к следующим параметрам:

  - перемещаемые параметры, параметры персонализации, настройки доступности и учетные данные;
  - резервное копирование и восстановление;
  - доступ к магазину Windows с помощью корпоративных учетных данных;
  - живые плитки и уведомления.

- Доступ к ресурсам организации.

  Теперь возможен доступ к корпоративным ресурсам с любых устройств, которые ранее было невозможно присоединить к домену.

- Единый вход (SSO).

  Возможности единого входа в Microsoft Office 365, внутренние ресурсы, решения с архитектурой «Программное обеспечение как услуга» (SaaS) и т. п.

- Использование сотрудниками личных устройств на работе (BYOD).

  Чтобы предоставить доступ к корпоративным ресурсам с личного устройства, можно указать рабочую учетную запись, которая будет использоваться для доступа, и воспользоваться новыми технологиями, такими как условный доступ.

- Интеграция с MDM.

  Эта функция предназначена для использования личных устройств на работе: с помощью автоматической регистрации личные устройства можно превратить в ресурсы, контролируемые корпоративными политиками. Для этого используется решение для управления мобильными устройствами (MDM), например Microsoft Intune.

- Режим киоска для множества пользователей.

  Чтобы предоставить нескольким пользователям возможность работать с одним современным приложением (например, приложением для входа в систему в приемной), можно настроить устройство в режиме киоска.

- Возможности разработки.

  Разработчики могут создавать приложения, использующие одинаковый набор технологий для использования как в рабочих, так и в личных целях.

- Образы.

  Можно предоставить конечным пользователям возможность выбора — использовать корпоративный образ или разрешить настройку корпоративных политик при первом включении компьютера.

Это все, конечно, полезные возможности, но для чего именно следует начать использовать присоединение к Azure AD? Причин может быть несколько — в зависимости от типа организации, в которой вы работаете. Например, если вы работаете в стартапе, где используется много мобильных устройств, то предоставление пользователям возможности использовать их собственные ноутбуки или настольные компьютеры, подключаясь к Azure AD, поможет сэкономить немало времени и усилий, необходимых для развертывания корпоративной политики. То же самое верно и для более зрелых организаций, осваивающих новые рынки: такой подход позволит избежать трудностей, связанных с транспортировкой компьютеров из главного офиса корпорации в удаленные офисы. Еще одна возможность относится к учебным заведениям и облачным решениям электронной почты, например, как в Office 365: такие организации могут управлять всеми пользователями в Azure AD, предоставлять доступ к облачным решениям электронной почты и управлять доступом к другим ресурсам, например Microsoft SharePoint Online.

Наконец, важно отметить не только различия между разными методами и особенностями их работы для пользователей, но и различия между предоставляемыми возможностями. Некоторые основные различия перечислены в табл. 4-4.

Таблица 4-4. Разные методы подключения

Корпоративное устройство (подключено к домену в локальной среде)	Корпоративное устройство (подключено к облаку)	Личное устройство
Пользователи могут входить в Windows с рабочими учетными данными (как они делают в настоящее время).	Пользователи могут входить в Windows с рабочими учетными данными, управление которыми осуществляется в Azure AD. Такой сценарий применим для корпоративных устройств в трех случаях: • У организации нет Active Directory в локальной среде (например, в компаниях малого бизнеса). • Организация использует Active Directory не для всех учетных записей пользователей (например, учетные записи в Active Directory не создаются для студентов, консультантов или сезонных работников). • Организация использует корпоративные устройства, которые невозможно присоединить к домену (в локальной среде), например телефоны или планшеты с мобильными операционными системами (дополнительные устройства в производственных помещениях или в торговых залах и т. п.). К Azure AD можно присоединять корпоративные устройства и в управляемых, и в федеративных организациях.	Пользователи входят в Windows, используя свои личные учетные записи Microsoft (без изменений).
Пользователи получают доступ к перемещаемым параметрам и к корпоративному Магазину Windows. Эти службы работают с учетными записями, им не требуется личная учетная запись Microsoft. Для этого организации должны подключить локальную службу каталогов Active Directory к Azure AD.	Пользователи могут использовать самообслуживание при настройке. При этом можно использовать процесс настройки при первом включении компьютера с использованием рабочей учетной записи вместо подготовки устройств к работе силами ИТ-подразделения, хотя поддерживаются оба метода.	Пользователи могут добавлять рабочую учетную запись, находящуюся под управлением Active Directory или Azure AD.
Пользователи имеют возможность единого входа с рабочего стола в рабочие приложения, на веб-сайты, на ресурсы, включая локальные ресурсы и облачные приложения, использующие Azure AD для проверки подлинности.	Устройства автоматически регистрируются в корпоративной службе каталогов (Azure AD) и автоматически подключаются к системе управления мобильными устройствами (функция Azure AD Premium).	Пользователи располагают возможностями единого входа для приложений, веб-сайтов и ресурсов, используя свою рабочую учетную запись.

Пользователи могут добавлять свои личные учетные записи Microsoft для доступа к личным файлам и изображениям, не затрагивая корпоративные данные (при этом перемещаемые параметры по-прежнему связаны с рабочими учетными записями). Учетная запись Microsoft поддерживает единый вход и больше не управляет перемещением параметров.	Пользователям доступен самостоятельный сброс пароля при входе в Windows, то есть они могут сбросить забытый пароль (функция Azure AD Premium).	Пользователи имеют доступ к корпоративному магазину Windows, чтобы приобретать и использовать бизнес-приложения на своих личных устройствах.

Microsoft Passport

Методы проверки подлинности сегодня развиваются быстрее, чем когда-либо ранее. Представьте себе следующую ситуацию: вы входите в систему на ноутбуке, затем открываете веб-браузер и переходите на ваши любимые веб-сайты, на которых снова выполняете процедуру входа. При этом вы не всегда используете корпоративные учетные данные. Если вы узнали о новой службе и хотите ее использовать, велика вероятность того, что вам будет предложено воспользоваться учетными данными вашей собственной (а не корпоративной) учетной записи Microsoft, Facebook, Google и т. п. Традиционная парадигма использования выделенного поставщика проверки подлинности удостоверений развивается, теперь все чаще для этого используются «хорошо известные» службы, такие как упомянутые ранее.

Microsoft Passport — это новый метод проверки подлинности на основе ключей, более надежный по сравнению с обычными паролями и более устойчивый к традиционным атакам, направленным на средства проверки подлинности. Пользователь регистрируется в Microsoft Passport, но при этом он должен убедиться в том, что поставщик проверки подлинности поддерживает механизм проверки подлинности FIDO. В двухэтапном процессе пользователь настраивает Microsoft Passport на своем устройстве, а также настраивает биометрический жест или ПИН-код, после чего их можно использовать для проверки подлинности через Microsoft Passport.

При настройке на устройстве сохраняется сертификат пары асимметричных ключей. Закрытый ключ хранится в аппаратном модуле TPM на устройстве. В процессе проверки подлинности закрытый ключ не выходит за пределы устройства. Открытый ключ регистрируется в Azure Active Directory и в Active Directory в Windows Server. Учетная запись пользователя представляет собой сопоставление между открытым и закрытым ключами. Дополнительные средства контроля реализуются с помощью одноразовых паролей, использования телефона (голосовой или SMS-проверки подлинности) и пр.

> **Примечание.** Более подробная информация о развертывании Microsoft Passport доступна по адресу https://aka.ms/bh1m24.

Службы федерации Active Directory

По мере распространения облачных решений в мире ИТ, способность управлять удостоверениями пользователей становится все более важной. Нужно подумать о том, как использовать корпоративные учетные данные для доступа к приложениям, которые, строго говоря, нам больше не принадлежат. Нужно позаботиться и о том, как предоставить доступ другим организациям к нашим приложениям с высоким уровнем безопасности и надежными средствами контроля, но без громоздких процессов управления пользователями.

Такую возможность предоставляют службы федерации Active Directory (AD FS). Они позволяют устанавливать подключения к приложениям, находящимся в локальной среде или в облаке (архитектура «платформа как услуга» [PaaS] или «программное обеспечение как услуга» [SaaS]), используя корпоративные удостоверения.

Службы федерации Active Directory используются уже давно, и в Windows Server 2016 эта технология получила ряд усовершенствований, чтобы соответствовать новому уровню потребностей организаций, использующих облачные технологии. Ниже приведен список основных улучшений AD FS:

- Многофакторная проверка подлинности.

  Windows Server 2016 содержит встроенный адаптер Azure MFA, упрощающий процесс использования многофакторной проверки подлинности (MFA) Azure в качестве основного поставщика удостоверений. Развертывать сервер MFA в локальной среде больше не требуется.

- Регистрация устройств для гибридного условного доступа.

  Теперь можно настроить службы AD FS для распознавания состояния устройств. Это означает, что можно управлять устройствами и применять политики по мере необходимости. Благодаря этому устройства всегда будут отвечать требованиям корпоративных политик, к тому же снижается потенциальный риск, которому подвергаются корпоративные ресурсы.

Примечание. Более подробная информация доступна по адресу https://aka.ms/i4jy7h.

- Интеграция Windows 10 и Microsoft Passport.

  Реализована возможность интеграции Microsoft Passport и служб AD FS, чтобы предоставлять удобные возможности проверки подлинности для пользователей Windows 10.

- Интеграция протокола LDAP для защиты каталогов без использования Active Directory.

  Многие организации не используют Active Directory для управления удостоверениями. В подобных случаях службы AD FS можно интегрировать с другими службами каталогов, поддерживающими протокол LDAP версии 3. Такой подход обеспечивает дальнейшую интеграцию в облако с использованием этих поставщиков удостоверений, при этом сохраняются такие же возможности, как при использовании Active Directory.

Примечание. Более подробная информация доступна по адресу https://aka.ms/qqupdh.

- Улучшенные средства аудита.

  Ранее аудит в службах AD FS был устроен довольно сложно: приходилось иметь дело с очень большим количеством подробной информации, которую было непросто отслеживать. В Windows Server 2016 возможности аудита были упорядочены: теперь доступны более удобные методы отслеживания нужной информации в журналах.

Примечание. Более подробная информация доступна по адресу https://aka.ms/ftbvm1.

- Усовершенствования SAML 2.0.

  В Windows Server 2016 улучшена поддержка SAML, добавлена возможность импорта доверия на основе метаданных, содержащих несколько записей. За счет этого службы AD FS можно настроить для участия в конфедерациях, таких как InCommon Federations, а также в других решениях, соответствующих стандарту eGov 2.0.

Примечание. Более подробная информация доступна по адресу https://aka.ms/d1xw4q.

- Настраиваемая процедура единого входа.

В Windows Server 2016 можно настраивать сообщения, изображения, эмблемы и темы для каждого приложения, поэтому организации, состоящие из нескольких подразделений, могут использовать одно развертывание вместо нескольких для каждого подразделения. Кроме того, можно использовать отдельные параметры и для каждой проверяющей стороны.

Примечание. Более подробная информация доступна по адресу https://aka.ms/f6rxu8.

- Упрощенное управление паролями для федеративных пользователей Office 365.

Службы AD FS теперь могут отправлять утверждения (claims) об окончании срока действия пароля субъектам, доверяющим проверяющей стороне. Пользователи приложения получат уведомления о том, что срок действия их паролей истекает, и смогут вовремя изменить свои пароли.

Примечание. Более подробная информация доступна по адресу https://aka.ms/i8jq9x.

- Настройка политик управления доступом без использования языка правил утверждений.

  В Windows Server 2016 существуют новые шаблоны политик управления доступом, с помощью которых удобнее настраивать правила утверждений. Эти шаблоны дают возможность использовать простой процесс на основе пользовательского интерфейса для быстрого и безопасного создания правил утверждений для вашей организации.

Примечание. Более подробная информация доступна по адресу https://aka.ms/rf833l.

- Обновление с предыдущих версий AD FS.

  В Windows Server 2016 значительно упрощен процесс обновления AD FS. Достаточно установить экземпляр AD FS Windows Server 2016 в существующую версию, проверить функциональность, а затем удалить прежние версии. Службы AD FS в Windows Server 2016 могут имитировать режим предыдущей версии AD FS.

Примечание. Более подробная информация доступна по адресу https://aka.ms/qo74pk.

Поделитесь своим мнением!

Эта книга пригодилась вам?
Она оправдала ваши ожидания?
Что-то можно было улучшить?

Напишите нам по адресу
http://aka.ms/tellpress

Ваши отзывы поступают непосредственно сотрудникам издательства Microsoft Press, мы внимательно читаем все ваши замечания и предложения. Спасибо!

Microsoft

ГЛАВА 5

Управление системами

Авторы: Джон Мак-Кейб (John McCabe), Ритеш Моди (Ritesh Modi)

В этой главе описываются новые возможности Windows Server 2016, связанные с управлением системами. Мы поговорим о новых возможностях Windows PowerShell версий 5/5.1, затем перейдем к System Center 2016 и к использованию Microsoft Operations Management Suite для получения полного спектра возможностей управления гибридными средами. Наконец, мы рассмотрим новые инструменты управления серверами с Windows с помощью графического веб-интерфейса и командной строки.

Усовершенствования Windows PowerShell

Windows PowerShell является стандартом де-факто в области управления решениями Microsoft. Едва ли вы найдете какой-либо иной продукт этого класса, обладающий такими же широкими возможностями, какие доступны в Windows PowerShell уже сегодня, включая встроенные в эту среду функции расширенной поддержки частных и общедоступных облаков. Среда Windows PowerShell стала настолько популярной, что многие сторонние поставщики встраивают поддержку Windows PowerShell непосредственно в свои приложения. Исходный код Windows PowerShell открыт — это означает, что сообщество разработчиков может принять участие в доработке этой среды и в ее дальнейшем усовершенствовании. Еще одно замечательное нововведение, о котором корпорация Microsoft объявила в 2016 году, — поддержка Linux. Теперь для управления средой Linux можно использовать те же интерфейсы и стандарты написания кода, что и созданные прежде в Windows PowerShell. Дальнейшее развитие Windows PowerShell — исключительно перспективное направление, поэтому корпорация Microsoft прилагает немало усилий для того, чтобы среда PowerShell стала наиболее предпочитаемым и распространенным средством управления не только для Windows, но и для Linux.

Windows PowerShell входит в состав платформы Windows Management Framework. Версия 5.1 вышла вместе с Windows Server 2016. Эта версия поддерживает обратную совместимость с прежними версиями от Windows 7 до Windows Server 2012 R2.

В этой главе представлен обзор некоторых новых возможностей версий 5 и 5.1. Мы постараемся затронуть наиболее интересные темы. Тем не менее в арсенале любого пользователя Windows PowerShell должна быть следующая полезная ссылка:

http://microsoft.com/PowerShell. Этот наиболее полный информационный ресурс содержит следующие разделы:

- галерея Windows PowerShell — http://www.powershellgallery.com/;
- блог Windows PowerShell — https://blogs.msdn.microsoft.com/powershell;
- репозиторий Windows PowerShell в GitHub — https://github.com/PowerShell/PowerShell.

Кроме того, вы можете:

- загрузить Windows Management Framework — https://www.microsoft.com/en-us/download/details.aspx?id =50395;
- ознакомиться с документацией Windows PowerShell — https://msdn.microsoft.com/en-us/powershell/scripting/powershell-scripting;
- отправить отзыв разработчикам Windows PowerShell — https://windowsserver.uservoice.com/forums/301869-powershell и многое другое.

Вот лишь некоторые из новых и усовершенствованных возможностей Windows PowerShell, о которых мы поговорим:

- управление пакетами;
- классы Windows PowerShell;
- отладка сценариев Windows PowerShell;
- настройка требуемого состояния.

Полное описание возможностей и функций WMF 5.0 и WMF 5.1 доступно по следующим ссылкам:

WMF 5.0 — https://msdn.microsoft.com/powershell/wmf/5.0/releasenotes;

WMF 5.1 — https://msdn.microsoft.com/powershell/wmf/5.1/release-notes.

Управление пакетами

Есть несколько способов установить программы в Windows традиционным образом. Например, можно создать EXE- или MSI-файл, запуск которого обеспечит установку программы. У таких файлов могут быть зависимые компоненты, поставляемые в составе различных пакетов. Также эти файлы могут зависеть от различных исправлений, содержащихся в файлах с расширениями .msu. В общем, установка программ — вовсе не такой простой процесс, как может показаться. В Linux данная проблема решена с помощью управления пакетами: используются диспетчеры пакетов, такие как APT-GET, YUM и другие. Эти диспетчеры анализируют программу, которую требуется установить, определяют все ее зависимые компоненты, а затем их устанавливают.

Как и в Linux, у нас приняты определенные термины и понятия в области управления пакетами:

- **пакет** — автоматизированный установщик программного обеспечения, в том числе элементов PowerShell;
- **репозиторий** — хранилище пакетов, которое может располагаться как в Интернете, так и в локальной ИТ-среде;

- **диспетчер пакетов** — интерфейс командной строки для установки пакетов;
- **поставщик** — код, предназначенный для обмена информацией с репозиторием;
- **источник/галерея** — репозиторий, содержащий программы для загрузки и установки.

Это вполне стандартные термины, их полезно знать для ознакомления с управлением пакетами в целом.

В следующем разделе мы подробно рассмотрим использование Windows PowerShellGet и Nuget для установки пакетов и приведем соответствующие технические подробности.

Windows PowerShellGet и NuGet

Традиционный способ установки модуля Windows PowerShell был таков: найти модуль в Интернете, загрузить и установить его. В Windows Server 2016 способ управления модулями изменен: в операционную систему встроен модуль PowerShellGet, который помогает находить, загружать и устанавливать модули, а также управлять ими.

PowerShellGet поддерживает работу с несколькими поставщиками. Эти поставщики представляют собой клиентские инструменты, подключающиеся к репозиторию модулей, который представлен источником (расположение определяется по URI). Самый главный поставщик, с которым работает PowerShellGet в Windows Server 2016, — это NuGet, диспетчер пакетов для Windows, предоставляющий возможность взаимодействовать не только с модулями, но также с приложениями и с пакетами. NuGet может работать с разными источниками, однако в качестве источника для PowerShellGet чаще всего используется репозиторий PSGallery.

Модуль PowerShellGet включает все функции для управления модулями. Первый шаг в управлении модулями — импорт модуля в консоль Windows PowerShell. Для этого нужно запустить интегрированную среду сценариев (ISE) Windows PowerShell , а затем выполнить следующую команду для загрузки модуля PowerShellGet:

```
PS C:\users\me» import-module PowerShellGet -Verbose

ПОДРОБНО:      Выполняется загрузка модуля с использованием пути PS C:\Windows\system32>
import-module PowerShellGet -Verbose
ПОДРОБНО:      Выполняется загрузка модуля с использованием пути ‘C:\Program
Files\windowsPowerShell\Modules\PowerShellGet\1.0.0.1\PowerShellGet.psdr.
ПОДРОБНО:      Выполняется загрузка "FormatsToProcess" с использованием пути "C:\Program
Files\WindowsPowerShell\Modules\PowerShellGet\1.0.0.1\PSGet. Format. pslxml".
ПОДРОБНО:      Выполняется загрузка модуля с использованием пути "C:\Program
Files\windowsPowerShell\Modules\PowerShellGet\1.0.0.1\PSModule.psmr.
ПОДРОБНО:      Импорт функции "Find-Command¹".
ПОДРОБНО:      Импорт функции "Find-DscResource".
ПОДРОБНО:      Импорт функции "Find-Module¹".
ПОДРОБНО:      Импорт функции "Find-RoleCapability".
ПОДРОБНО:      Импорт функции "Find-Script".
ПОДРОБНО:      Импорт функции "Get-InstalledModule".
ПОДРОБНО:      Импорт функции "Get-InstalledScript".
ПОДРОБНО:      Импорт функции "Get-PSRepository".
ПОДРОБНО:      Импорт функции "Install-Module".
ПОДРОБНО:      Импорт функции "Install-Script".
ПОДРОБНО:      Импорт функции "New-ScriptFilelnfo".
ПОДРОБНО:      Импорт функции "Publish-Module".
ПОДРОБНО:      Импорт функции "Publish-Script".
ПОДРОБНО:      Импорт функции "Register-PSRepository".
ПОДРОБНО:      Импорт функции "Save-Module".
ПОДРОБНО:      Импорт функции "Save-Script".
ПОДРОБНО:      Импорт функции "Set-PSRepository".
ПОДРОБНО:      Импорт функции "Test-ScriptFilelnfo".
ПОДРОБНО:      Импорт функции "Uninstall-Module".
ПОДРОБНО:      Импорт функции "Uninstall-Script".
ПОДРОБНО:      Импорт функции 'Unregister-PSRepository'.
ПОДРОБНО:      Импорт функции "Update-Module".
ПОДРОБНО:      Импорт функции "Update-ModuleManifest".
ПОДРОБНО:      Импорт функции "Update-Script".
ПОДРОБНО:      Импорт функции "Update-ScriptFileInfo".
ПОДРОБНО:      Импорт псевдонима "fimo".
ПОДРОБНО:      Импорт псевдонима "inmo".
ПОДРОБНО:      Импорт псевдонима "pumo".
ПОДРОБНО:      Импорт псевдонима "upmo".
```

При использовании параметра Verbose с командлетом Import-Module отображаются все импортируемые функции, командлеты и псевдонимы. В этом модуле доступны восемь функций, две переменные и четыре псевдонима.

При первом использовании командлета PowerShellGet проверяется, установлен ли диспетчер пакетов NuGet. Если NuGet не установлен, отображается сообщение подтверждения:

```
Для взаимодействия параметра PowerShellGet с коллекциями на основе NuGet требуется объект
NuGet-anycpu.exe. Объект NuGet-anycpu.exe должен быть доступен в
"C:\ProgramData\OneGet\ProviderAssemblies" или

"C:\Users\<имя_пользователя>\AppData\Local\OneGet\ProviderAssemblies". Более подробная
информация о поставщике NuGet доступна по адресу http://www.nuget.org. Разрешить PowerShellGet
загрузить объект NuGet-anycpu.exe сейчас?
```

Нажмите «Да», и модуль NuGet будет загружен и установлен на компьютер.

Командлеты, предоставляемые модулем PowerShellGet, разделяются на две общие категории: модули и командлеты репозитория. PowerShellGet предоставляет командлеты для поиска, установки, публикации и обновления модулей из репозитория. Также он содержит командлеты для чтения текущих параметров репозитория, их обновления, регистрации и отмены регистрации.

По умолчанию доступно два репозитория: PSGallery и MSPSGallery. Для подключения к ним PowerShellGet использует поставщик NuGet. Если выполнить командлет Get-PSRepository, отображаются все репозитории, имеющиеся на компьютере.

```
PS C:\users\me>> Get-PSRepository

Name        SourceLocation                                  OneGetProvider    InstallationPolicy
PSGallery   https://msconfiggallery.cloudapp.net/api/v2/  NuGet             Untrusted
MSPSGallery http://www.microsoft.com/                       NuGet             Trusted
```

Если запустить командлет Get-PSRepository, указав имя репозитория, отображаются все конфигурации, связанные с этим репозиторием.

```
PS C:\WINDOWS\system32>get-PSRepository -name PSGallery |Format-list *
Name                      :PSGallery
SourceLocation            :https://www.powershellgallery.com/api/v2/
Trusted                   :False
Registered                :True
InstallationPolicy        :Untrusted
PackageManagementProvider :NuGet
PublishLocation           :https://www.powershellgallery.com/api/v2/package/
ScriptSourceLocation      :https://www.powershellgallery.com/api/v2/items/psscrip
ScriptPublishLocation     :https://www.powershellgallery.com/api/v2/package/
ProviderOptions           :{}
```

В выходных данных: SourceLocation — это URL-адрес расположения репозитория; PackageManagementProvider — поставщик пакетов, использованный для подключения к репозиторию (в данном случае NuGet); Trusted — отметка о том, является ли репозиторий доверенным; PublishLocation — URL-адрес, использованный для отправки модулей.

Командлет Set-PSRepository устанавливает значения конфигурации репозитория. Например, приведенный ниже командлет Set-PSRepository изменяет значение конфигурации Untrusted на Trusted для репозитория PSGallery. После изменения значения можно просмотреть новую конфигурацию, запустив командлет Get-PSRepository.

```
PS C:\Users\me> Set-PSRepository -Name "PSGallery" -InstallationPolicy Trusted

PS C:\Users\me> Get-PSRepository

Name                 InstallationPolicy   SourceLocation
PSGallery            Trusted              https://www.powershellgallery.com/api/v2/
```

Командлет Register-PSRepository предназначен для добавления и регистрации нового репозитория. Для правильной работы с репозиторием командлету требуются следующие параметры: имя репозитория, адрес источника для загрузки модулей, адрес для публикации новых модулей в репозиторий, политика установки и диспетчер пакетов. Если запустить этот

командлет, указав имя Chocolatey, http://chocolatey.org/api/v2/ в качестве адреса источника и публикации, Trusted в качестве значения политики установки и NuGet в качестве имени диспетчера пакетов, то будет добавлен новый репозиторий модулей:

```
Register-PSRepository -Name "Chocolatey" -SourceLocation "http://chocolatey.org/api/v2/" `
-PublishLocation "http://chocolatey.org/api/v2/" -InstallationPolicy Trusted `
-PackageManageMentProvider NuGet
```

После регистрации командлет Find-Module сможет выполнять поиск в репозиториях, а командлет Install-Module может загружать и устанавливать модули. Имя Chocolatey указано исключительно для примера: это может быть любой репозиторий с модулями Windows PowerShell.

Если запустить командлет Unregister-PSRepository с параметром Name, то зарегистрированный ранее репозиторий будет удален.

```
PS C:\Users\me> Unregister-PSRepository -Name Chocolatey
```

Самая важная функция модуля Powershell Get — поиск и установка модулей. Если запустить командлет Find-Module, не указывая никаких параметров, будут выведены все доступные модули во всех репозиториях, как показано здесь:

```
PS C:\Users\me> Find-Module
Version  Name               Repository      Description
2.0.1    AzureRM.profile    PSGallery       Microsoft Azure PowerShell - Profile credential ...
2.0.1    Azure.Storage      PSGallery       Microsoft Azure PowerShell - Storage service cmd...
1.7.6    Posh-SSH           PSGallery       PowerShell module for automating tasks using the...
2.0.1    AzureRM            PSGallery       Azure Resource Manager Module
```

Запустив командлет Find-Module с параметром Name, вы получите список модулей, связанных с этим именем. Например, если запустите этот командлет, указав для параметра Name значение Bing, то получите информацию о Bing:

```
PS C:\Users\me> Find-Module -Name "Bing"

Version  Name          Repository        Description
5.0      Bing          PSGallery         A few functions for working with the new Bing APIs
5.0      Bing          Choclatey         A few functions for working with the new Bing APIs

PS C:\Users\me> Find-Module -Name "*ing"

Version  Name                    Repository  Description
2.11.0.  xNetworking             PSGallery   Module with DSC Resources for Networking area
0
0.9.4    AzureRM.MachineLearning PSGallery   Microsoft Azure PowerShell - Machine Learning We..
1.0.0.0  xWindowsEventForwarding PSGallery   This module can be used to manage configuration ..
2.5.2    PSLogging               PSGallery   Creates and manages log files for your scripts.
1.2.1    PowerShellLogging       PSGallery   Captures PowerShell console output to a log file.
5.0      Bing                    PSGallery   A few functions for working with the new Bing APIs
2.0.1    Remote PSRemoting       PSGallery   Enable PSRemoting Remotely using WMI
```

Примечание. Параметр Name также принимает подстановочные знаки.

Кроме того, командлет Find-Module использует дополнительные параметры MinimumVersion и RequiredVersion. Одновременно можно применять только один из них. Чтобы загрузить определенную версию, используйте параметр RequiredVersion. Чтобы загрузить самую последнюю версию, соответствующую указанной, или более позднюю, используйте параметр MinimumVersion.

Если запустить Find-Module, указав значение Bing для параметра Name и значение 4.0 для параметра MinimumVersion, то будет найден модуль Bing версии 5.0.

```
PS C:\Users\me> Find-Module -Name "Bing" -MinimumVersion "4.0"

Version    Name          Repository        Description
5.0        Bing          PSGallery         A few functions for working with the new Bing APIs
5.0        Bing          Choclatey         A few functions for working with the new Bing APIs
```

Если же запустить Find-Module, указав значение Bing для параметра Name и значение 4.0 для параметра RequiredVersion, то появится сообщение об ошибке:

```
PS C:\Users\me> Find-Module -Name "Bing" -RequiredVersion "6.0"
PackageManagement\Find-Package : Для указанных условий поиска и имен модулей "Bing" не найдено
никаких совпадений.
```

```
Используйте
Get-PSRepository для просмотра всех доступных репозиториев модулей.
At C:\Program Files\WindowsPowerShell\Modules\PowerShellGet\1.0.0.1\PSModule.psm1:1249 char:3
+ PackageManagement\Find-Package @PSBoundParameters | Microsoft ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo : ObjectNotFound: (Microsoft.Power...ets.FindPackage:FindPackage)
[Find-Package], Exception
    + FullyQualifiedErrorId :
NoMatchFoundForCriteria,Microsoft.PowerShell.PackageManagement.Cmdlets.FindPackage
```

Если вы запустите Find-Module, указав значение Bing для параметра Name и значение 5.0 для параметра RequiredVersion, то получите информацию о модуле Bing:

```
PS C:\Users\me> Find-Module -Name "Bing" -RequiredVersion "5.0"

Version   Name         Repository       Description
5.0       Bing         PSGallery        A few functions for working with the new Bing APIs
5.0       Bing         Choclatey        A few functions for working with the new Bing APIs
```

После обнаружения нужных модулей можно перейти к установке. Для установки модулей в PowerShellGet предусмотрен командлет Install-Module. Он работает аналогично командлету Find-Module, при этом используются параметры Name, RequiredVersion и MinimumVersion.

В приведенном ниже фрагменте кода показана работа командлета Install-Module с параметрами Name и Verbose. Вместе с параметром Name можно использовать параметры MinimumVersion или RequiredVersion. Обратите внимание на последнюю строку выходных данных: она указывает, что модуль установлен. Кроме того, обратите внимание на то, что по умолчанию модули загружаются в папку $env:ProgramFiles\WindowsPowershell\Modules. Среда Windows PowerShell использует эту папку для установки модулей.

```
PS C:\Users\me> Install-Module -Name bing -Repository PSGallery -Verbose -AllowClober

ПОДРОБНО: Данные репозитория, имя = "SGallery", расположение =
"https://www.powershellgallery.com/api/v2/";
надежный =
"True"; зарегистрированный = "True".
ПОДРОБНО: Для поиска пакетов используется поставщик "PowerShellGet".
ПОДРОБНО: Использование указанных имен источников: "PSGallery".
ПОДРОБНО: Получение объекта поставщика PackageManagement "NuGet".
ПОДРОБНО: Указанное расположение: "https://www.powershellgallery.com/api/v2/" и поставщик
PackageManagement:
"NuGet".
ПОДРОБНО: Searching repository
'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='bing'' for ''.
ПОДРОБНО: Total package yield:'1' for the specified package 'bing'.
ПОДРОБНО: Выполнение операции "Install-Module" над целевым объектом ""Version '5.0' модуля 'Bing'".
ПОДРОБНО: Вид установки указан как "AllUsers".
ПОДРОБНО: Указанный модуль будет установлен в "C:\Program Files\WindowsPowerShell\Modules".
ПОДРОБНО: Указанное расположение: "NuGet", PackageManagementProvider: "NuGet".
ПОДРОБНО: Загрузка модуля "Bing" с версией "5.0" из репозитория "{2}".
"https://www.powershellgallery.com/api/v2/".
ПОДРОБНО: Searching repository
'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='bing'' for ''.
ПОДРОБНО: Searching repository
'https://www.powershellgallery.com/api/v2/FindPackagesById()?id='BetterCredentials'' for
''.
ПОДРОБНО: InstallPackage' - name='BetterCredentials',
version='4.4',destination='C:\Users\johm\AppData\Local\Temp\1152878524'
ПОДРОБНО: DownloadPackage' - name='BetterCredentials',
version='4.4',destination='C:\Users\johm\AppData\Local\Temp\1152878524\BetterCredentials\Bette
rCredentials.nu pkg',
uri='https://www.powershellgallery.eom/api/v2/package/BetterCredentials/4.4.0¹
ПОДРОБНО: Загрузка "https://www.powershellgallery.eom/api/v2/package/BetterCredentials/4.4.0".
ПОДРОБНО: Завершена загрузка
"https://www.powershellgallery.com/api/v2/package/BetterCredentials/4.4.0¹".
ПОДРОБНО: Завершена загрузка "BetterCredentials".
ПОДРОБНО: InstallPackageLocal' - name='BetterCredentials¹,
version='4.4'destination='C:\Users\johm\AppData\Local\Temp\1152878524'
ПОДРОБНО: InstallPackage¹ - name='Bing'J
version='5.0'destination='C:\Users\johm\AppData\Local\Temp\1152878524'
ПОДРОБНО: DownloadPackage¹ - name='Bing'J
version='5.0',destination=''C:\Users\johm\AppData\Local\Temp\1152878524\Bing\Bing.nupkg',
uri='https://www.powershellgallery.com/api/v2/package/Bing/5.0.0'
ПОДРОБНО: Загрузка 'https://www.powershellgallery.com/api/v2/package/Bing/5.0.0¹.
```

```
ПОДРОБНО: Завершена загрузка "https://www.powershellgallery.com/api/v2/package/Bing/5.0.0".
ПОДРОБНО: Завершена загрузка "Bing".
ПОДРОБНО: InstallPackageLocal - name='Bing'
version='5.0' destination='C:\Users\johm\AppData\Local\Temp\1152878524'
ПОДРОБНО: Catalog file 'BetterCredentials.cat' is not found in the contents of the module
'BetterCredentials
being installed.
ПОДРОБНО: Установка модуля зависимости "BetterCredentials" с версией 4.4 для модуля "Bing".
ПОДРОБНО: Модуль "BetterCredentials" успешно установлен в путь "C:\Program
Files\WindowsPowerShell\Modules\BetterCredentials\4.4\
ПОДРОБНО: Catalog file 'Bing.cat is not found in the contents of the module 'Bing being installed.
ПОДРОБНО: Модуль "Bing" успешно установлен в путь "C:\Program
Files\WindowsPowerShell\Modules\Bing\5.0\
```

Теперь можно использовать модуль Bing, импортировав его в текущее пространство выполнения Windows PowerShell с помощью командлета Import-Module. После первоначальной установки командлет Update-Module обновляет существующие модули. Этот модуль использует параметры Name и RequiredVersion, но не поддерживает параметр MinimumVersion:

```
PS C:\Users\me> Update-Module -Name Bing
PS C:\Users\me> Update-Module -Name "Bing" -RequiredVersion "5.0"
```

Также есть командлет Publish-Module, предназначенный для добавления в репозиторий более новых модулей.

Классы Windows PowerShell

Классы Windows PowerShell предоставляют новый способ расширения возможностей управления Windows PowerShell, ориентированный на разработчиков и ИТ-специалистов. С помощью классов PowerShell можно создавать объекты Windows PowerShell традиционным образом, используя формальный синтаксис и семантику объектно-ориентированного программирования.

Например, разработчики наверняка хорошо знакомы с такими конструкциями ООП, как классы или методы. Теперь в Windows PowerShell есть возможность определить их на встроенном языке и затем использовать.

Описание классов Windows PowerShell выходит за рамки учебного материала уровня 200, к которому относится эта книга, тем не менее полезно знать, что это важное усовершенствование Windows PowerShell и что эти возможности становятся доступными для более широкой аудитории.

Вот некоторые элементы, поддерживаемые классами Windows PowerShell:

- определение ресурсов настройки требуемого состояния на собственном языке Windows PowerShell;
- определение настраиваемых типов (то есть классов, свойств и методов);
- поддержка типов отладки;
- создание и обработка исключений с использованием формальных методов.

Эти возможности могут показаться слишком сложными, но мы рассказываем о них уже сейчас, чтобы вы могли оценить развитие Windows PowerShell — мощной и полезной среды, ставшей одной из фундаментальных технологий при создании современных бизнес-приложений.

> **Примечание.** Более подробная информация о создании настраиваемых типов с помощью Windows PowerShell доступна по адресу http://msdn.microsoft.com/powershell/wmf/5.0/class_overview.

Отладка сценариев Windows PowerShell

В Windows Server 2016 усовершенствованы следующие функции отладки сценариев Windows PowerShell:

- «прервать все»;
- удаленная правка;
- удаленная отладка;
- отладка заданий;
- отладка пространств выполнения;
- удаленная отладка настройки требуемого состояния.

Давайте рассмотрим каждую из них подробнее.

«Прервать все»

Это очень полезная функция для остановки запущенного сценария, позволяющая открыть отладчик, посмотреть, как работает сценарий, и узнать текущее состояние переменных и других элементов. Поддержка такого прерывания добавлена в консоль Windows PowerShell и в интегрированную среду сценариев.

Чтобы использовать отладчик, работая в сеансе консоли, нажмите клавиши Ctrl+Break.

В интегрированной среде сценариев Windows PowerShell нажмите клавиши Ctrl+B или выберите в меню «Отладка» и «Прервать все».

Удаленная правка

В текущем сеансе интегрированной среды сценариев Windows PowerShell можно открывать и напрямую редактировать файл в удаленном сеансе Windows PowerShell. С помощью новой команды PSEdit можно редактировать файлы как локально, так и в удаленных сеансах. Это действие показано в следующем примере кода:

[Cloudei]: PS C:\> PSEdit C:\WinDemo\Get-ComputerInfo.psl

При выполнении команды PSEdit файл открывается в интегрированной среде сценариев. В ней можно внести в файл нужные изменения, сохранить его на удаленном компьютере и заново запустить код.

Удаленная отладка

Теперь возможности удаленной работы расширены: поддерживается не только удаленная правка, но и отладка сценария, запущенного в удаленном сеансе в интегрированной среде сценариев.

Командлет Set-PsBreakpoint устанавливает в коде точку останова, после чего можно использовать командлет Write-Debug, чтобы вывести нужную информацию. Это будет полезным в сценариях при переходе к точке останова: открывается отладчик, в котором можно выполнить дополнительные действия и просмотреть доступную информацию. Это показано на рис. 5-1.

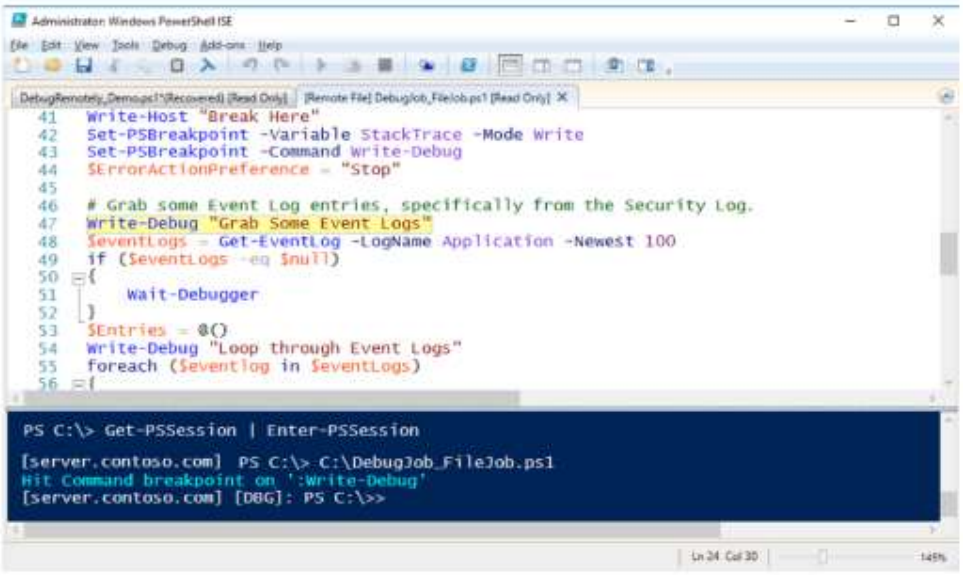


Рис. 5-1. Образец кода в интегрированной среде сценариев Windows Powershell с удаленной отладкой

Когда сценарий дойдет до точки останова в удаленном сеансе, он отобразит соответствующее сообщение. Пример такого сообщения показан на рис. 5-2.

Рис. 5-2. Предупреждение о точке останова в удаленном сеансе

Не все удаленные сеансы поддерживают удаленную отладку, но при подключении к удаленному сеансу с помощью командлета Connect-PSSession на экране появится информация, подобная той, что показана на рис. 5-3. Кроме того, указывается, доступна ли отладка.

Рис. 5-3. Доступность удаленной отладки

Если удаленная отладка доступна, можно установить подключение к этому сеансу с помощью командлета Enter-PSSession, при этом обеспечивается подключение непосредственно к отладчику (рис. 5-4).

Рис. 5-4. Удаленный отладчик в действии

Отладка заданий

Одна из полезных функций Windows PowerShell — возможность запуска сценариев в качестве фоновых заданий. Это позволяет выполнять дополнительные задачи не загромождая основной экран консоли. Несмотря на все преимущества такого подхода, раньше отладка таких заданий была затруднена: разработать надежные сценарии, способные работать в виде заданий, можно было лишь приложив значительные усилия — методом проб и ошибок.

В Windows Server 2016 появился командлет Debug-Job, дающий возможность эффективнее отлаживать эти фоновые задания. Использование этого командлета демонстрируется на рис. 5-5.

Обратите внимание на то, что командлет Debug-Job показывает строку и символ, на котором произошла ошибка выполнения задания.

Рис. 5-5. Командлет Debug-Job выводит сообщение об ошибке в фоновом задании

Чтобы получить это состояние прерывания, приостановив сценарий и войдя в отладчик, используется командлет Set-PSBreakPoint или Wait-Debugger. Есть возможность запустить эти командлеты в сценарии, проверить состояние задания и удостовериться в том, что сценарий дошел до точки останова и готов к отладке. Пример показан на рис. 5-6.

Рис. 5-6. Состояние фонового задания

Отладка пространств выполнения

Пространства выполнения были добавлены в среду Windows PowerShell, чтобы обеспечить решение ряда проблем, возникающих при появлении фоновых заданий, в том числе проблем с доступом к ресурсам и со снижением производительности. Пространства выполнения отличаются от заданий тем, что создают в среде новый параллельный поток выполнения и при этом позволяют избежать издержек, характерных для фоновых заданий.

> **Примечание.** Более подробная информация о пространствах выполнения и о том, как их использовать, доступна по адресу http://blogs.technet.microsoft.com/heyscriptingguy/2015/11/26/beginning-use-of-powershell-runspaces-part-1/.

На рис. 5-7 показано создание пространства выполнения.

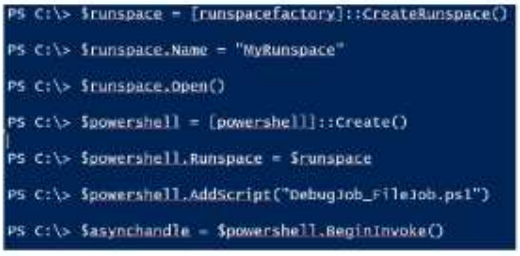


Рис. 5-7. Создание пространства выполнения

Как и в случае с заданиями, для отладки пространства выполнения необходимо получить его идентификатор. На рис. 5-8 показаны доступные пространства выполнения и вход в сеанс отладки с помощью командлета Debug-Runspace.

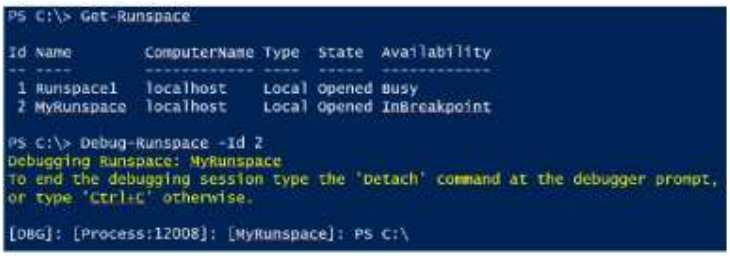


Рис. 5-8. Отладка пространства выполнения

Первое пространство выполнения (ID 1) всегда является исходным сеансом Windows PowerShell, в котором вы работаете. Более подробная информация о пространствах выполнения доступна по указанному выше адресу.

Настройка требуемого состояния

Настройка требуемого состояния (DSC) Windows PowerShell широко обсуждается в среде ИТ-специалистов. DSC — это новая платформа управления конфигурациями, с помощью которой администраторы могут использовать Windows PowerShell для развертывания программных служб и управления ими, а также для управления средой, в которой выполняются эти службы. В Windows Server 2016 реализован ряд усовершенствований DSC. В этой главе мы рассмотрим два из них: новый локальный диспетчер конфигураций и новую функцию неполных конфигураций.

Локальный диспетчер конфигураций настройки требуемого состояния

Важнейшим компонентом DSC является локальный диспетчер конфигураций (LCM). Это ядро DSC, отвечающее за обработку и использование файлов конфигураций (имеющих расширение .mof).

LCM принимает (режим push), получает (режим pull), применяет, отслеживает, сравнивает файлы конфигураций и выдает отчеты об изменениях. Безусловно, LCM — это основа DSC.

В состав Windows Server 2016 входят Windows Management Framework 5.1 и DSC. Платформа DSC появилась в Windows Management Framework 4.0 (ее также можно загрузить и для более ранних версий Windows), она входит в состав Windows Server 2012 R2 и Windows 8.1. Как и следовало ожидать, в Windows Management Framework 5.1 реализовано много новых возможностей DSC и LCM и много изменений.

Для настройки LCM и ядра DCS можно использовать документ *метаконфигурации* (meta.mof). Поведение и действия LCM можно изменять, модифицируя свойства метаконфигурации.

Модуль DSC версии 2 в Windows Server 2016 создан на основе предыдущей версии. Были упразднены некоторые свойства LCM, добавлены новые командлеты для управления конфигурациями, обновлен диспетчер LCM — в нем появилась дополнительная функциональность, введены новые атрибуты метаконфигурации и новые возможности, такие как неполные конфигурации и синхронизация между компьютерами.

Диспетчер LCM реализован в качестве класса модели CIMMSFT_DSCLocalConfigurationManager в пространстве имен root\Microsoft\Windows\DesiredStateConfiguration.

В этом разделе мы поговорим о новых возможностях и функциях LCM v2 в Windows Server 2016.

В DSC предусмотрено два командлета для доступа к LCM, просмотра и обновления свойств LCM: Get-DSCLocalConfigurationManager для просмотра и Set-DSCLocalConfigurationManager для установки. При запуске командлета Get-DSCLocalConfigurationManager в консоли PowerShell в Windows Server 2016 выводятся все свойства метаконфигурации LCM и их текущие значения. Значения по умолчанию показаны ниже:

```
PS C:\> Get-DscLocalConfigurationManager

ActionAfterReboot                : ContinueConfiguration
AgentId                          : C8F7308B-6E6D-11E6-899F-B4AE2BEB7DE5
AllowModuleOverWrite             : False
CertificateID                    :
ConfigurationDownloadManagers    : {}
ConfigurationID                  :
ConfigurationMode                : ApplyAndMonitor
ConfigurationModeFrequencyMins : 15Credential :
DebugMode                        : {NONE}
DownloadManagerCustomData        :
```

```
DownloadManagerName             :
LCMCompatibleVersions           : {1.0, 2.0}
LCMState                        : Idle
LCMStateDetail                  :
LCMVersion                      : 2.0
StatusRetentionTimelnDays       : 10
SignatureValidationPolicy       : NONE
SignatureValidations            : {}
MaximumDownloadSizeMB           : 500
PartialConfigurations           :
RebootNodeIfNeeded              : False
RefreshFrequencyMins            : 30
RefreshMode                     : PUSH
ReportManagers                  : {}
ResourceModuleManagers          : {}
PSComputerName                  :
```

Чтобы обеспечить обратную совместимость, LCM в Windows Server 2016 содержит все свойства первой версии. Большинство этих свойств необходимы и в новой версии, однако некоторые упразднены, и потому их нельзя использовать для настройки LCM в Windows PowerShell версии 5.

> **Примечание.** Подробная справочная информация обо всех доступных свойствах метаконфигурации доступна по адресу https://msdn.microsoft.com/powershell/dsc/metaconfig.

В новых свойствах LCM можно использовать несколько фрагментов конфигурации вместо одной конфигурации. Свойства лучше упорядочены, четко очерчена сфера применения каждого из них. Можно опрашивать текущее состояние LCM, включать и отключать кэширование, использовать разные URL-адреса конечных точек для конфигураций и ресурсов.

В DSC в Windows PowerShell версии 4 для настройки свойств метаконфигурации использовался особый ресурс LocalConfigurationManager. Этот ресурс упразднен в LCM в Windows PowerShell версии 5. Его по-прежнему можно использовать для настройки LCM версии 2, но с его помощью невозможно настраивать новые свойства метаконфигурации. Вместо него для настройки свойств LCM рекомендуется использовать новый ресурс Settings.

Для настройки свойств метаконфигурации нужно выполнить несколько действий в определенной последовательности. Как уже было сказано выше, в LCM версии 2 для настройки свойств мета-конфигурации LCM используется новый ресурс, который называется Settings. Его нужно поместить в сценарий конфигурации и затем запустить. В ходе работы ресурса формируется файл meta.mof, который передается в LCM целевого сервера, после чего этот LCM применяет и изменяет значения свойств метаконфигурации. Обратите внимание на то, что настройка LCM не допускается в обычной конфигурации с использованием обычных ресурсов DSC. В LCM версии 2 вместе с новым ресурсом Settings появилось еще несколько ресурсов, предназначенных для LCM. Эти ресурсы повышают удобство создания и изменения свойств, доступных в ресурсе Settings. Их список приведен далее:

- **Settings** — основной ресурс метаконфигурации LCM.
- **ConfigurationRepositoryWeb** — конечная точка протокола OData служб Internet Information Services (IIS) для опрашивающих серверов. Данный ресурс изменяет свойство ConfigurationDownloadManagers ресурса Settings. У него имеются следующие свойства:
  - ConfigurationNames;
  - ServerUrl;
  - AllowUnsecureConnection;
  - CertificateID;
  - RegistrationKey.
- **ConfigurationRepositoryShare** — конечная точка ресурсов SMB для опрашивающих серверов. Этот ресурс изменяет свойство ConfigurationDownloadManagers ресурса Settings. Он обладает следующими свойствами:
  - SourcePath;

    - Credential.
- **ResourceRepositoryWeb** — конечная точка протокола OData служб IIS для загрузки ресурсов DSC, используемых службами IIS. Этот ресурс изменяет свойство ResourceModuleManagers ресурса Settings. Его свойства следующие:
    - AllowUnsecureConnection;
    - ServerUrl;
    - CertificateID;
    - RegistrationID.
- **ResourceRepositoryShare** — конечная точка протокола OData служб IIS для загрузки ресурсов DSC с помощью общих ресурсов SMB. Этот ресурс изменяет свойство ResourceModuleManagers ресурса Settings. У него имеются следующие свойства:
    - SourcePath;
    - Credential.
- **ReportServerWeb** — конечная точка протокола OData служб IIS для предоставления данных отчетов, связанных с узлами, их текущих конфигураций и изменений. Этот ресурс изменяет свойство ResourceModuleManagers ресурса Settings. Он обладает следующими свойствами:
    - ServerUrl;
    - CertificateID;
    - RegistrationKey;
    - AllowUnsecureConnection.
- **PartialConfiguration** — имя конфигурации, которую следует получить с сервера. Одна конфигурация может содержать несколько ресурсов PartialConfiguration. Этот ресурс изменяет свойство PartialConfigurations ресурса Settings. Его свойства следующие:
    - DependsOn;
    - RefreshMode;
    - RefreshModuleSource;
    - Description;
    - ExclusiveResources;
    - ConfigurationSource.

Примечание. Подробное описание перечисленных блоков доступно по адресу https://msdn.microsoft.com/powershell/dsc/metaconfig.

Ниже приведена типовая реализация свойства метаконфигурации в LCM версии 2:

```
[DSCLocalConfigurationManager()]
Configuration ChangeLCMProperties {
   Node DemoServerWin {
      Settings
      {
          AllowModuleOverwrite = $false
          RebootNodeIfNeeded = $true
          RefreshMode = "Pull"
          ConfigurationMode = "ApplyAndAutoCorrect"
          ConfigurationID = "fcd03a8d-5a64-4982-92b3-5c89680add39"
      }
      ConfigurationRepositoryWeb PullServer1
      {
      ServerURL = "http://demoserverwin10:8090/PSDSCPullServer.svc/" AllowUnsecureConnection =
      $true
      }
```

```
        ConfigurationRepositoryWeb PullServer2
        {
                  ServerURL = "http://demoserverwin10:8080/PSDSCPullServer.svc/"
            AllowUnsecureConnection = $true
        }
        ReportServerWeb ComplianceServer
        {
            ServerURL = "http://demoserverwin10:8000/PSDSCComplianceServer.svc/"
            AllowUnsecureConnection = $true
        }
        PartialConfiguration IISInstall
        {
            Description = 'Configuration for IIS Web Server'
            ConfigurationSource = '[ConfigurationRepositoryWeb]PullServer1'
         }
        PartialConfiguration IndexFile
        {
            Description = 'Configuration for Index File'
            ConfigurationSource = '[ConfigurationRepositoryWeb]PullServer2'
            DependsOn = '[PartialConfiguration]IISInstall'
        }
    }
}

ChangeLCMProperties -OutputPath "C:\DSC"

Set-DscLocalConfigurationManager -Path "C:\DSC"
```

Приведенный выше сценарий аналогичен общей настройке DSC с помощью конфигурации ChangeLCMProperties, но с заданным атрибутом DSCLocalConfigurationManager. Этот атрибут устанавливает, что все ресурсы в конфигурации должны быть связаны только с LCM. Если в этой конфигурации использовать другие общие ресурсы, возникнет ошибка. Сценарий содержит один раздел узла для сервера DemoServerWin.

Ресурс Settings является основным ресурсом для настройки свойств LCM. В приведенном примере мы указываем некоторые его свойства и назначаем им значения. Например: устанавливается режим обновления Pull, чтобы операционная система была перезапущена (когда это требуется ресурсу); режим настройки изменен на ApplyandAutoCorrect; для свойства ConfigurationID предоставлен допустимый идентификатор GUID. Конфигурация, представленная идентификатором GUID, будет получена с опрашивающего сервера.

В конфигурации указано два опрашивающих сервера: PullServer1 и PullServer2. Свойство ServerURL показывает, что они находятся на одном и том же сервере, но с разными номерами портов. Кроме того, свойство AllowUnsecureConnection дает возможность использовать протокол HTTP вместо HTTPS. С помощью ReportServerWeb также предоставляется информация о совместимости сервера. LCM загружает две неполные конфигурации и применяет их как одну конфигурацию на сервере. Неполная конфигурация

IISInstall отвечает за загрузку конфигурации с именем IISInstall с сервера PullServer1. Неполная конфигурация с именем IndexFile отвечает за загрузку конфигурации с именем IndexFile с сервера PullServer2. Более того, запуск неполной конфигурации IndexFile зависит от завершения конфигурации IISInstall — на это указывает свойство DependsOn. Конфигурация IndexFile может быть запущена только после применения конфигурации IISInstall.

После того как конфигурация определена, она выполняется, и в результате создается файл с расширением .mof (DemoServerWin.Meta.mof) по адресу C:\DSC. Расположение папки указывается явным образом с помощью параметра OutputPath. После создания MOF-файла командлет Set-DSCLocalConfigurationManager передает его и применяет к серверу DemoServerWin.

При применении конфигурации происходит настройка LCM на сервере DemoServerWin, чтобы обеспечить получение неполных конфигураций с нескольких опрашивающих серверов, а также периодическое применение этих конфигураций.

После этого можно снова прочесть новую конфигурацию, используя Get-DSCLocalConfigurationManager, как показано в следующем примере:

```
PS C:\Users\me> Get-DscLocalConfigurationManager

ActionAfterReboot                   : ContinueConfiguration
AgentId                             : C8F7308B-6E6D-11E6-899F-B4AE2BEB7DE5
AllowModuleOverWrite                : False
CertificateID                       :
ConfigurationDownloadManagers       : {[ConfigurationRepositoryWeb]PullServer1,
                                      [ConfigurationRepositoryWeb]PullServ
                                      er2}
ConfigurationID                     : fcd03a8d-5a64-4982-92b3-5c89680add39
ConfigurationMode                   : ApplyAndAutoCorrect
ConfigurationModeFrequencyMins      : 15
Credential                          :
DebugMode                           : False
DownloadManagerCustomData           :
DownloadManagerName                 :
LCMCompatibleVersions               : {1.0, 2.0}
LCMState                            : Ready
LCMVersion                          : 2.0
MaximumDownloadSizeMB               : 500
StatusRetentionTimeInDays           : 7
PartialConfigurations               : {[PartialConfiguration]IISInstall,
[PartialConfiguration]IndexFile}
RebootNodeIfNeeded                  : True
RefreshFrequencyMins                : 30
RefreshMode                         : PULL
ReportManagers                      : [ReportServerWeb]ComplianceServer
ResourceModuleManagers              : {}
PSComputerName                      :
```

В приведенном выше фрагменте кода свойство ConfigurationDownloadManagers имеет два значения, которые представляют два опрашивающих сервера: PartialConfigurations имеет два значения, представленные конфигурациями IISInstall и IndexFile, а ReportManagers имеет значение ComplianceServer.

Новые методы в локальном диспетчере конфигураций

В LCM версии 2 реализовано три новых метода: GetConfigurationStatus, GetConfigurationResultOutput и SendConfigurationApplyAsync. Давайте их кратко рассмотрим.

GetConfigurationStatus

Метод GetConfigurationStatus получает текущее состояние конфигурации с сервера. Новый командлет DSC Get-DSCConfigurationStatus вызывает метод CIM. В следующем коде показан пример метода Get-DSCConfigurationStatus:

```
PS C:\> $cimsession = New-CimSession -ComputerName DemoServerWin10 PS C:\>
Get-DscConfigurationStatus -CimSession $cimsession
Status       Start Date    Type          Mode       Reboot      Number Of      Resources PS
                                                    Requested   Configuration  Computer Name
Success      2016/07/12    Consistency   PUSH       False       1              DemoServerWin10
             16:23:03
```

GetConfigurationResultOutput

Метод GetConfigurationResultOutput предоставляет подробную информацию о текущей конфигурации и об изменениях конфигурации. Командлетов DSC, вызывающих этот метод CIM, не существует. Тем не менее его можно вызвать с помощью командлета CIM Invoke-CIMMethod, как показано ниже:

```
PS C:\>
$ConsistencyCheck = (Invoke-CimMethod -ClassName "MSFT_DSCLocalConfigurationManager" `
            -Namespace "root\Microsoft\Windows\DesiredStateConfiguration" `
            -MethodName getConfigurationResultOutput)

for($i=0; $i -le 100; $i++)
{
   $ConsistencyCheck[$i].ItemValue.Message
}
```

```
[DEMOSERVERWIN10]:                            [] Запуск модуля обеспечения согласованности.
[DEMOSERVERWIN10]: LCM: [Start Resource]     [[WindowsFeature]XPS]
[DEMOSERVERWIN10]: LCM: [Start Test]         [[WindowsFeature]XPS]
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Выполняется функциональность
Test компонента XPS-Viewer.
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Выполняется запрос компонента
XPS-Viewer с помощью командлета Get-WindowsFeature диспетчера серверов.
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Запущена операция
"Get-WindowsFeature": XPS-Viewer
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Запущен метод поставщика
GetServerComponentsAsync: XPS-Viewer
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Вызов метода поставщика
GetServerComponentsAsync успешно выполнен.
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Операция "Get-WindowsFeature"
успешно выполнена: XPS-Viewer
[DEMOSERVERWIN10]:                            [[WindowsFeature]XPS] Завершено выполнение
функциональности Test functionality компонента XPS-Viewer.
[DEMOSERVERWIN10]: LCM: [ End Test ]         [[WindowsFeature]XPS] за 0,4667 с.
[DEMOSERVERWIN10]: LCM: [ End Resource ]     [[WindowsFeature]XPS]
[DEMOSERVERWIN10]:                            [] Проверка согласованности выполнена.
```

SendConfigurationApplyAsync

Метод SendConfigurationApplyAsync асинхронно применяет конфигурацию к целевому серверу. Это означает, что LCM вызывает этот метод, но не дожидается его завершения. Для вызова этого метода также нет командлетов DSC, но можно использовать командлет CIM, как показано в следующем примере:

```
PS C:\> Configuration PushDemo
{
   Node DemoServerWin10
   {
      WindowsFeature XPS
      {
         Name = "XPS-Viewer"
         Ensure = "Absent"
      }
   }
}

PushDemo -OutputPath "C:\DSC"
$mofString = get-content "C:\dsc\DemoServerWin10.mof"
$mofbytes = [System.Text.Encoding]::ASCII.GetBytes($mofString)
$AsyncApply = Invoke-CimMethod -ClassName "MSFT_DSCLocalConfigurationManager" `
            -Namespace "root\Microsoft\Windows\DesiredStateConfiguration" `
            -MethodName SendConfigurationApplyAsync `
            -Arguments @{ConfigurationData=$mofbytes;Force=$true}
$AsyncApply
   Directory: C:\DSC
Mode                 LastWriteTime     Length                 Name
07/12/2016           2:28 PM          1226                    DemoServerWin10.mof
PSComputerName:
```

Неполные конфигурации настройки требуемого состояния (DSC)

Одна из самых ожидаемых и интересных возможностей DSC версии 2 — поддержка неполных конфигураций. До выпуска DSC версии 2 было сложно разделить конфигурацию на несколько файлов конфигурации для сервера. Неполные конфигурации дают возможность разделять конфигурацию на несколько фрагментов, каждый из которых сохраняется в отдельном файле. Неполные конфигурации реализуются точно таким же образом, что и обычные конфигурации DSC. LCM на целевом сервере отвечает за объединение всех фрагментов в одну конфигурацию и за ее применение.

Неполные конфигурации представляют собой готовые законченные объекты, их можно применять к любому серверу независимо друг от друга (так же как полные конфигурации). Метаконфигурация LCM, настроенная на целевом сервере, дает возможность применять на этом сервере неполные конфигурации.

В Windows Server 2016 неполные конфигурации работают с режимами отправки и получения DSC. Это означает, что для получения конфигураций с опрашивающего сервера (IIS или общего ресурса SMB) и для идентификации конфигураций на таких серверах нужно настроить диспетчер LCM серверов в сети.

Вот некоторые из преимуществ неполных конфигураций:

- Создавать конфигурации могут несколько авторов независимо друг от друга, причем одновременно для нескольких серверов в сети.
- Можно применять дополнительные конфигурации к серверам, не меняя при этом существующие конфигурации.
- Можно создавать конфигурации по модульному принципу.
- Нет необходимости использовать только один MOF-файл. Именно такое ограничение было в DSC версии 1: в любой момент времени можно было использовать и применить к серверу только один MOF-файл. Текущие конфигурации в DSC версии 1 заменяются новыми конфигурациями (.mof).

Для использования неполных конфигураций в Windows Server 2016 необходимо выполнить следующие действия:

1. Создать опрашивающий сервер.
2. Настроить метаконфигурацию LCM серверов в сети.
3. Создать конфигурации.
4. Развернуть конфигурации на опрашивающем сервере.

Мы подробнее рассмотрим все эти шаги, кроме создания опрашивающего сервера, поскольку это делается точно так же, как для DSC версии 1.

Настройка метаконфигурации локального диспетчера конфигураций

Для подготовки метаконфигурации LCM сервера необходимо настроить следующие свойства:

- RefreshMode — значение Pull.
- ConfigurationMode — любое значение (важно, чтобы сервер находился в нужном состоянии).
- ConfigurationRepositoryWeb — экземпляр ресурса, представляющий собой опрашивающий сервер, это может быть веб-служба или общий ресурс SMB.
- Несколько экземпляров ресурса PartialConfiguration, каждый из которых представляет конфигурацию опрашивающего целевого сервера.

Еще одно свойство метаконфигурации в LCM, которое нужно настроить, — это либо ConfigurationID, либо ConfigurationName. Более подробная информация по этим свойствам доступна по следующим адресам:

- ConfigurationName —

  https://msdn.microsoft.com/powershell/dsc/pullclientconfignames;

- ConfigurationID —

  https://msdn.microsoft.com/powershell/dsc/pullclientconfigid.

Для демонстрации неполных конфигураций в следующем примере используется среда с опрашивающим сервером (DemoServerWin10). Имеются две конфигурации, каждая из которых развернута на одном из опрашивающих серверов. Эти серверы и конфигурации настроены

в диспетчере LCM компьютера назначения. Конфигурация LCM, примененная к серверу DemoServerWin, показана ниже:

```
[DSCLocalConfigurationManagerQ]
Configuration ChangeLCMProperties
{
   Node DemoServerWin
   {
      Settings
      {
         RebootNodeIfNeeded = $true
         RefreshMode = "Pull"
         ConfigurationMode = "ApplyAndAutoCorrect"
         ConfigurationID = ᴹfcd03a8d-5a64-4982-92b3-5c89680add39"
      }
      ConfigurationRepositoryWeb PullServerl
      {
         ServerURL = ’’http://demoserverwinl0:8080/PSDSCPullServer.svc/‟
         AllowUnsecureConnection = $true
      }
      PartialConfiguration HSInstall
      {
         Description = 'Configuration for IIS Web Server¹
         ConfigurationSource = ‘[ConfigurationRepositoryWebpullServerl’
      }
      PartialConfiguration IndexFile
      {
         Description = 'Configuration for Index File’
         ConfigurationSource = ‘[ConfigurationRepositoryWeb]PullServerl’
         DependsOn= ‘[PartialConfiguration]IISInstall’
      }
   }
}
ChangeLCMProperties -OutputPath "C:\DSC"
Set-DscLocalConfigurationManager -Path "C:\DSC"
```

Если исходить из того, что опрашивающие серверы на указанных серверах уже доступны, необходимо изменить свойства метаконфигурации LCM на всех машинах, которые получают от них конфигурации. В приведенном выше примере конфигурации есть раздел узла с именем DemoServerWin. Он указывает, что эта конфигурация изменяет конфигурацию LCM сервера DemoServerWin. Атрибут DSCLocalConfigurationManager указывает, что в этой конфигурации можно использовать только ресурсы, применимые к метаконфигурации LCM, и что эта метаконфигурация выдаст файл с расширением .meta.mof. В таких конфигурациях, как уже было сказано, невозможно использовать обычные ресурсы. С помощью этого атрибута можно указать для DSC, что эта конфигурация относится только к LCM.

Ресурс Settings настроен следующим образом: свойство RefreshMode имеет значение Pull, свойство ConfigurationMode — ApplyandAutoCorrect, свойство ConfigurationID — fcd03a8d-5a64-4982-92b3-5c89680add39, а свойство RebootNodeIfNeeded — значение True. LCM загружает с опрашивающего сервера файлы конфигурации с таким GUID, который был назначен свойству ConfigurationID. Таким образом, в приведенном выше примере LCM загрузит с опрашивающего сервера файлы конфигурации, имена которых содержат «fcd03a8d-5a64-4982-92b3-5c89680add39».

Также нужно указать в LCM данные об опрашивающем сервере. В LCM версии 2 это можно сделать с помощью ресурса WebConfigurationRepository. В конфигурации можно определить несколько опрашивающих серверов (ресурсов WebConfigurationRepository). В приведенном выше примере определены два опрашивающих сервера: PullServer1 с URL-адресом http://demoserverwin:8080/PSDSCPullServer.svc/ и свойством AllowUnsecureConnection со значением True; сервер PullServer2 с URL-адресом http://Demoserverwin10:8090/PSDSCPullServer.svc/ и свойством AllowUnsecureConnection со значением True. В этом примере также показано свойство AllowUnsecureConnection. Если установить для этого свойства значение True, LCM сможет запрашивать конфигурацию по протоколу HTTP вместо HTTPS, однако мы настоятельно рекомендуем всегда использовать HTTPS, если вы находитесь в обычной рабочей среде.

Ресурс PartialConfiguration определяет фрагменты конфигурации. Определены две неполные конфигурации: IISInstall и IndexFile. Конфигурация IISInstall доступна на сервере PullServer1, а конфигурация IndexFile — на сервере PullServer2. Важно обратить внимание на имена неполных конфигураций, поскольку они должны в точности совпадать с именами конфигураций на опрашивающем сервере. В следующем разделе показаны создание конфигурации IISInstall и ее доступность на сервере PullServer1, а также доступность конфигурации IndexFile на сервере PullServer2. Свойство ConfigurationSource прикрепляет опрашивающий сервер к неполной конфигурации.

Кроме того, обратите внимание на то, что комбинация из URL-адреса опрашивающего сервера, ConfigurationID и Configuration Name предоставляет диспетчеру LCM полную информацию, позволяющую уникальным образом идентифицировать конфигурацию на опрашивающем сервере. LCM не может получить неполные конфигурации, если какая-либо часть этой информации отсутствует.

Показанная ранее конфигурация создает файл DemoServerWin.meta.mof в папке C:\DSC. Для передачи и применения MOF-файла к серверу DemoServerWin можно использовать командлет Set-DSCLocalConfigurationManager.

Создание конфигураций

Теперь мы рассмотрим создание конфигураций, которые будут участвовать в неполной конфигурации. В этом разделе мы создадим две конфигурации: IISInstall и IndexFile.

Конфигурация IISInstall

Это простая конфигурация, отвечающая за установку IIS (веб-сервера) на сервер с помощью ресурса WindowsFeature. При запуске конфигураций, перечисленных в этом и следующем разделах, формируются MOF-файлы. Сценарий конфигурации в этом примере выполняется на сервере ServerWin10. Имя MOF-файла совпадает с именем, определенным в сценарии конфигурации. Для конфигураций, участвующих в неполных конфигурациях, действует особое требование в отношении именования: их имя должно иметь формат *<ConfigurationName>.<ConfigurationID>*.mof. Конфигурация, показанная в этом примере, для определения имени узла использует ConfigurationData — структуру данных для передачи значений сценарию конфигурации. $AllNodes.NodeName получает все имена узлов из данных конфигурации. В нашем случае узел только один, поскольку задано только одно свойство NodeName.

IISInstall.fcd03a8d-5a64-4982-92b3-5c89680add39.MOF формируется следующим сценарием:

```
$ConfigInfoIIS = @{
   AllNodes = @(
      @{
         NodeName = "IISInstall.fcd03a8d-5a64-4982-92b3-5c89680add39"
      }
   )
}
Configuration IISInstall
{
   Node $AllNodes.NodeName
   {
      WindowsFeature IIS
      {
         Name = "Web-server"
         Ensure = "Present"
      }
   }
}
IISInstall -OutputPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
-ConfigurationData $ConfigInfoIIS
New-DSCCheckSum -ConfigurationPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
-OutPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
```

MOF-файлы следует разместить на опрашивающем сервере WinServer10 в заданной папке, обычно это C:\Program Files\WindowsPowerShell\DSCService\Configuration. При создании MOF-файла расположение папки передается в качестве параметра атрибута OutPath.

После создания MOF-файла необходимо создать контрольную сумму файла конфигурации. По контрольной сумме можно определить, существует ли новая конфигурация на опрашивающем сервере. Также с ее помощью можно проверять целостность конфигураций при их передаче между опрашивающим сервером и LCM. Имя файла контрольной суммы должно быть таким же, как у MOF-файла, а расширение должно быть .mof.checksum. В DSC для создания файла контрольной суммы предусмотрен командлет New-DSCCheckSum. Его параметр ConfigurationPath указывает расположение папки для сохранения конфигураций. Командлет создает файл контрольной суммы для каждого файла конфигурации и сохраняет файлы контрольной суммы в папке, указанной параметром OutPath.

Конфигурация IndexFile

IndexFile — это простая конфигурация, отвечающая за создание файла Index.htm на сервере в корневой папке IIS по умолчанию (C:\Inetpubwwwroot\). Смысл этого файла в том, чтобы обеспечить отображение текста «На веб-сайте ведутся технические работы». Конфигурация использует ресурс File и создает файл Index.htm в папке C:\Inetpub\wwwroot с отображаемым содержимым формата HTML, в этом примере оно следующее: «Если вы видите эту страницу, это означает, что веб-сайт сейчас находится на обслуживании, а DSC работает». Приведенный ниже сценарий конфигурации выполняется на сервере DemoServerWin10. Имя сценария — IndexFile.fcd03a8d-5a64-4982-92b3-5c89680add39, значение GUID здесь то же, что указано выше в примере с конфигурацией IISInstall.

```
$ConfigInfoIndex = @{
   AllNodes = @(
      @{
         NodeName = "IndexFile.fcd03a8d-5a64-4982-92b3-5c89680add39"
      }
   )
}
Configuration IndexFile
{
   Node $AllNodes.NodeName
   {
      File IndexFile
      {
         DestinationPath = "C:\inetpub\wwwroot\index.htm"
         Ensure = "Present"
         Type = "File"
         Force = $true
         Contents = "<HTML><HEAD><Title> На веб-сайте ведутся технические
         работы.</Title></HEAD><BODY>
         <h1>Если вы видите эту страницу, это означает, что веб-сайт сейчас находится на обслуживании,
         а DSC работает!</h1></BODY></HTML>"
      }
   }
}
IndexFile -OutputPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
-ConfigurationData $ConfigInfoIndex
New-DSCCheckSum -ConfigurationPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
-OutPath 'C:\Program Files\WindowsPowerShell\DscService\Configuration'
```

Для работы конфигурация *IndexFile* должна совпадать с неполной конфигурацией, определенной в метаконфигурации.

MOF-файл создается на сервере получения DemoWinServer10 в папке C:\Program Files\WindowsPowerShell\DSCService\Configuration. Контрольная сумма этого файла конфигурации формируется таким же образом, как в предыдущем примере.

Развертывание конфигураций

После настройки LCM и создания неполных конфигураций следует применить их к целевому серверу, в данном случае — к серверу DemoServerWin. В новой версии DSC предусмотрен командлет Update-DSCConfiguration для получения конфигурации и применения ее к серверу. Если запустить этот командлет, указав для параметра ComputerName значение localhost, с опрашивающего сервера будут одновременно загружены все подходящие конфигурации (MOF-файлы), определенные с помощью ресурса LCM PartialConfiguration. Затем LCM объединяет все эти конфигурации в единый MOF-файл и применяет его к серверу. В следующем примере демонстрируется командлет Update-DSCConfiguration, предназначенный для применения конфигурации:

```
PS C:\Users\me> Update-DSCConfiguration -ComputerName localhost

Id    Name   PSDobTypeName      State       HasMoreData   Location     Command
47    Dob47  Configuratio...    Running     True          localhost    Update-DscConfiguration
```

При запуске командлета Get-DSCConfiguration на этом сервере все ресурсы (File и WindowsFeature) применяются в составе конфигурации DSC, как показано здесь:

```
PS C:\Users\riteshmodi> Get-DSCConfiguration

ConfigurationName                 :
DependsOn                         :
ModuleName                        :
ModuleVersion                     :
ResourceId                        :
SourceInfo                        :
Credential                        :
DisplayName                       : Web Server (IIS)
Ensure                            : Present
IncludeAllSubFeature              : False
LogPath                           :
Name                              : Web-Server
Source                            :
PSComputerName                    :
ConfigurationName                 :
DependsOn                         :
ModuleName                        :
ModuleVersion                     :
ResourceId                        :
SourceInfo                        :
Attributes                        : {archive}
Checksum                          :
Оглавление                        :
CreatedDate                       : 7/8/2016 1:40:50 PM
Credential                        :
DestinationPath                   : C:\inetpub\wwwroot\index.htm
Ensure                            : present
Force                             :
MatchSource                       :
ModifiedDate                      : 7/14/2016 7:09:02 AM
Recurse                           :
Size                              : 197
SourcePath                        :
SubItems                          :
Type                              : file
PSComputerName                    :
```

Примечание. Более подробная информация о работе с Windows PowerShell DSC доступна по адресу

https://msdn.microsoft.com/powershell/dsc/overview. Более подробная информация о неполных конфигурациях PowerShell DSC доступна по адресу https://msdn.microsoft.com/powershell/dsc/partialconfigs.

System Center 2016

Одновременно с новым Windows Server выпущена и новая версия решения System Center. В этом разделе мы поговорим о новых возможностях System Center 2016. Решение System Center 2016 предназначено главным образом для управления в гибридной среде: оно позволяет управлять облаком прямо из System Center, также есть возможность использовать облако для расширения возможностей System Center или управлять гибридной средой из облака. В состав System Center 2016 входят облачные средства управления Microsoft Operations Management Suite и Microsoft Intune.

Многие нововведения System Center 2016 предназначены для поддержки новых возможностей Windows Server 2016. Кроме того, пакет System Center 2016 поддерживает программно-определяемые центры обработки данных (SDDC) и содержит все необходимые инструменты для управления такими ресурсами. В табл. 5-1 перечислены некоторые новые возможности System Center 2016 и общие области управления, к которым они относятся.

Таблица 5-1. <Наименование таблицы>

Область	Возможности
Управление устройствами	Поддержка развертывания Windows 10 Регистрация MDM в Microsoft Azure Active Directory Ограничение доступа на основе регистрации устройств и политики
Выделение ресурсов	Поддержка функций Hyper-V в Windows Server 2016 Technical Preview Последовательные обновления кластеров Упрощенное управление сетями Подготовка защищенных виртуальных машин (Shielded VM) Управление защищенными узлами Поддержка VMware vCenter 5.5
Мониторинг	Nano Server Хранилище Windows Поддержка SMS-S Усовершенствования каталога пакетов управления Повышение производительности Усовершенствованная визуализация данных Улучшенная поддержка Linux Улучшенная поддержка сетей
Автоматизация	Переход в облако Пакеты интеграции SCO и Runbook SMA с поддержкой синтаксиса Windows PowerShell Windows Management Framework 5.0 Поддержка подключаемого модуля интегрированной среды сценариев Windows PowerShell для модулей Runbook SMA
Самообслуживание	Более удобное использование и повышенная производительность Портал самообслуживания на основе HTML 5 Новый соединитель Microsoft Exchange
Защита данных	Поддержка Azure ExpressRoute Поддержка защищенных виртуальных машин (Shielded VM) Локальные дисковые пространства (Storage Spaces Direct)

Платформа System Center изначально была предназначена для управления локальной инфраструктурой. Эти возможности управления развивались в предыдущих версиях, усовершенствования продолжились и в System Center 2016.

Можно использовать System Center 2016 и для управления облачной средой. Например, хотели бы вы знать состояние подписки на Office 365? System Center 2016 позволяет эту информацию получить. На рис. 5-9 показан пример панели мониторинга, которая доступна в пакете управления для Microsoft Office 365.

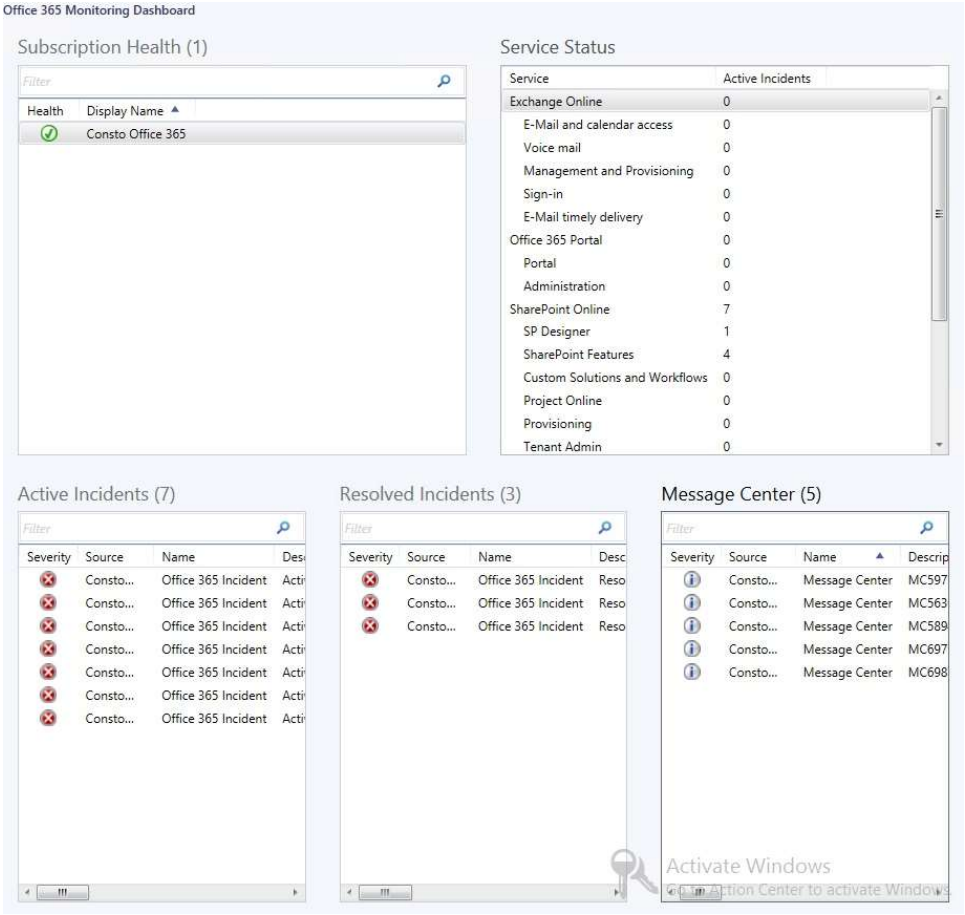


Рис. 5-9. Панель мониторинга Office 365

В окне «Состояние подписки» (Subscription Health), показанном на рис. 5-9, можно проверить правильность конфигурации и наличие подключения к Office 365. При развертывании этого пакета управления необходимо настроить набор учетных данных, поскольку для подключения к подписке пакету управления потребуются соответствующие разрешения. На рис. 5-9 также показаны области подписки Office 365 с активными инцидентами (Active Incidents). Активные инциденты (Active Incidents), как и другие оповещения Operations Manager, содержат информацию о состоянии работоспособности и ссылку на центр знаний, в котором содержится информация о решении проблемы.

В приведенном выше примере показан лишь Office 365, но это только небольшая часть возможностей, доступных в System Center 2016 для управления публичной облачной службой. Возможности управления доступны в публичных облачных средах, устроенных по принципу «инфраструктура как услуга» (IaaS), «платформа как услуга» (PaaS) и «программное обеспечение как услуга» (SaaS).

Чтобы понять, как использовать пакет в разных средах, рассмотрим пример сценария. Компания Contoso Limited создала простой виджет, который оказался настолько удобным, что спрос на него растет в геометрической прогрессии. Теперь компании требуется ИТ-решение, полностью соответствующее принципам Agile, отвечающее нуждам и потребностям не только сотрудников Contoso, но и клиентов компании.

Клиенты Contoso хотят иметь возможность купить этот виджет в любое время и из любого места. И разумеется, с любого устройства. Что это означает на практике? То, что, проще говоря, компании Contoso нужна система, доступная круглосуточно и без выходных. Что это означает с точки зрения инфраструктуры и управления? Итак, у нас два веб-сайта, подключающихся к серверам СУБД, которые размещены в облаке. Первый веб-сайт предназначен для покупателей, а второй — для сотрудников отдела продаж Contoso: сайт позволяет проверить складские запасы, просмотреть заказы и т. п. Также есть мобильные службы для подключения и размещения заказов в любое время с помощью приложений для смартфонов. Для проверки подлинности пользователей веб-сайт для сотрудников отдела продаж использует Azure Active Directory (Azure AD). Это означает, что в ИТ-среде Contoso служба каталогов Active Directory расширена в облако Azure. Для совместной работы и доступа к электронной почте в Contoso также используется Office 365. И наконец, чтобы использовать возможности глобальной связи, компания интегрировала свою систему телефонии с Office 365.

ИТ-среда Contoso включает публичное и частное облака. Компания обладает глобальной базой клиентов, поддержкой которой занимается глобальная база сотрудников.

Используя System Center 2016, компания может: выделять ресурсы инфраструктуры, когда это необходимо; управлять состоянием приложения; интегрировать процессы разработки и эксплуатации так, чтобы в цепочке разработки отображались фактические метрики использования приложения; устранять неполадки; проводить диагностику и т. п.

Может показаться, что этот пример не слишком близок к реалиям существующих сегодня инфраструктур, но современные компании постоянно развиваются. Покупатели могут находиться где угодно, в любых местах, поэтому размещение систем в регионе, расположенном далеко от клиентов, приведет к неудовлетворительным результатам. Предположим, что основные центры данных размещены в США, а офисы и клиенты находятся в Индии. Клиентам будет неудобно пользоваться сайтом хотя бы из-за задержек, неизбежно возникающих из-за значительного расстояния до центра данных. Предположим также, что вы решили не использовать облако для размещения приложений, а вместо этого собираетесь открыть локальный центр данных в Индии. В этом случае придется продублировать структуру управления, вследствие чего управлять станет сложнее. Существует немало сценариев, демонстрирующих развитие ИТ и принципы управления ИТ-средами.

При разработке System Center 2016 значительное внимание уделялось схемам использования, подобным сценарию компании Contoso: была поставлена задача реализовать не только управление облаком из локальной среды, но и обратную схему. Еще одним ключевым направлением стало обеспечение соответствия темпам развития облачных технологий. С одной стороны, важно было инвестировать в изменение и развитие System Center 2016 в соответствии с темпами эволюции облачных технологий, но в то же время всегда целесообразны и вложения в создание гибридного решения для управления, работающего и в облаке, и в локальной среде. В следующем разделе мы поговорим о пакете Operations Management Suite, который используется для решения задач управления средами.

Operations Management Suite

Operations Management Suite (OMS) — это облачное решение для управления, которое можно использовать совместно с System Center 2016, приобретенным и развернутым как локально, так и независимо, на другой площадке. На рис. 5-10 показана схема управления гибридной ИТ-средой с программными решениями разных производителей с помощью System Center 2016 и OMS.

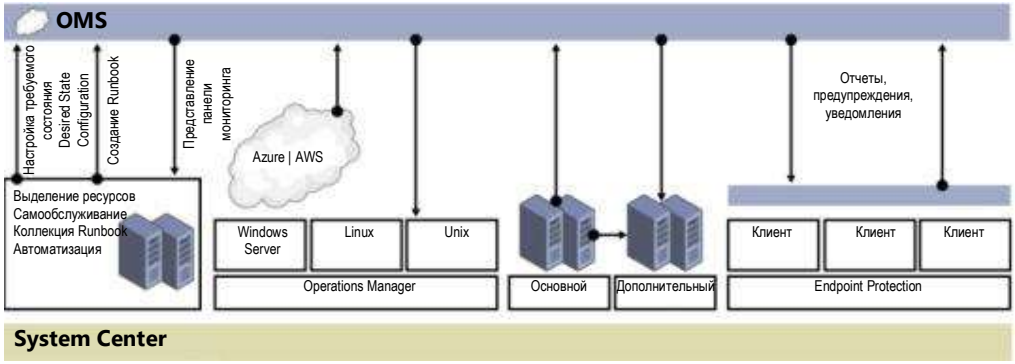


Рис. 5-10. Управление ИТ-средой, в которой развернуты решения различных поставщиков, с помощью System Center 2016 и OMS

Перед обсуждением преимуществ совместного использования OMS и System Center 2016 давайте рассмотрим возможности, доступные в OMS. Их можно разделить на четыре основные области (табл. 5-2).

Таблица 5-2. <Наименование таблицы>

Область	Описание
Аналитика журналов	Поиск закономерностей, выявление проблем с использованием различных источников журналов, предоставление информации о событиях в ИТ-среде в реальном времени; интеграция с панелями мониторинга Microsoft Power BI для наглядного отображения информации
Автоматизация ИТ-среды	Автоматизация простых и сложных задач в ИТ-среде; непосредственная интеграция с приложениями и системой управления версиями для среды автоматизации, подключение и управление ресурсами во всех центрах обработки данных
Резервное копирование и восстановление	Резервное копирование рабочих нагрузок непосредственно в облако, использование облака в качестве точки восстановления; также можно реплицировать рабочие нагрузки из VMware или Hyper-V и использовать облако в качестве сайта восстановления
Безопасность и соответствие внутренним или внешним требованиям и нормам	Непрерывный анализ всех аспектов работы среды — от пользователей, входящих в систему, до возникновения новых рисков

Основной вывод, который можно сделать, гладя на табл. 5-2, состоит в том, что возможность управления гибридной средой существует. Это особенно важно, если вы вложили немало средств в локальную среду и собираетесь использовать OMS вместе с System Center 2016. Впрочем, если вы не эксплуатируете System Center в локальной среде, а желаете пользоваться OMS, это также вполне возможно: используйте преимущества OMS для управления существующей облачной или локальной ИТ-средой.

Прежде всего, вне зависимости от того, развернуто ли решение OMS, необходимо создать рабочую область OMS. Для этого войдите на сайт https://portal.azure.com и нажмите «Создать» (New). Введите **Служба анализа журналов (OMS) (Log Analytics)**, нажмите «Служба анализа журналов (OMS)» (Log Analytics) (рис. 5-11), затем на следующей странице нажмите «Создать».

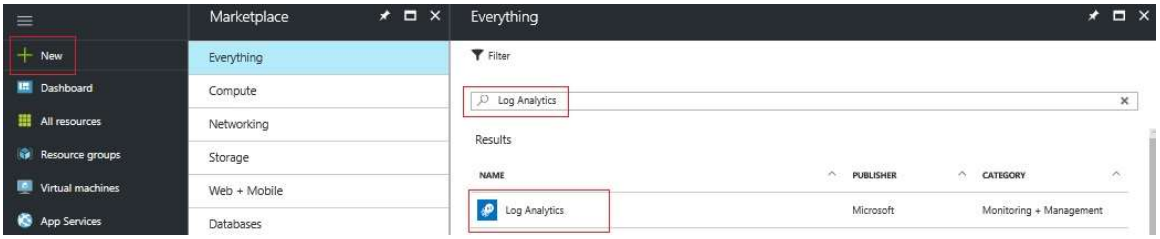


Рис. 5-11. Создание рабочей области OMS, часть 1

Отобразится окно рабочей области OMS (OMS Workspace). Нужно настроить параметры, как показано на рис. 5-12, а затем нажать «OK». Также нужно выбрать ценовую категорию. Для начала подойдет ценовая категория «Бесплатно» (Free), позволяющая изучить возможности и преимущества OMS.

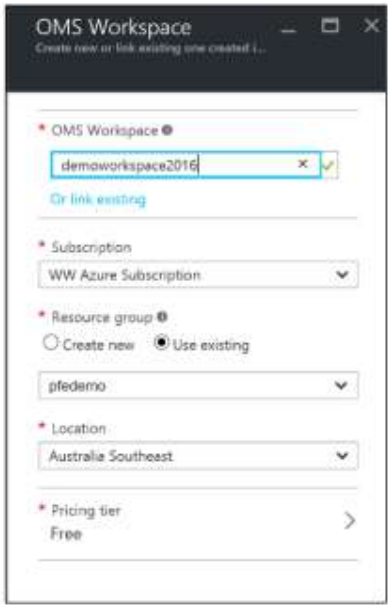


Рис. 5-12. Создание рабочей области OMS, часть 2

> **Примечание.** После выбора поля «Оперативная аналитика» (Operational Insights) вы будете перенаправлены на портал «Управление службами Azure (ASM)», а затем снова вернетесь на портал Azure Resource Manager (ARM). Аналогичный результат дает и выбор поля «Служба анализа журналов (OMS)» (Log Analytics).

После создания рабочей области перейдите в раздел Log Analytics. Для вашей рабочей области будет отображаться состояние «Активно» (Active).

На этой странице доступно очень много параметров (слишком много, чтобы описывать их в этой книге), поэтому для примера, обратившись к разделу «Источники данных рабочей области» (Workspace Data Sources), рассмотрим два параметра: «Виртуальные машины» (Virtual machines) и «Журналы учетных записей хранения» (Storage accounts logs). Если нажать «Виртуальные машины» (Virtual machines), откроется новая страница, на которой перечислены виртуальные машины, существующие в той группе ресурсов, в которой вы опубликовали рабочую область Log Analytics, как показано на рис. 5-13.

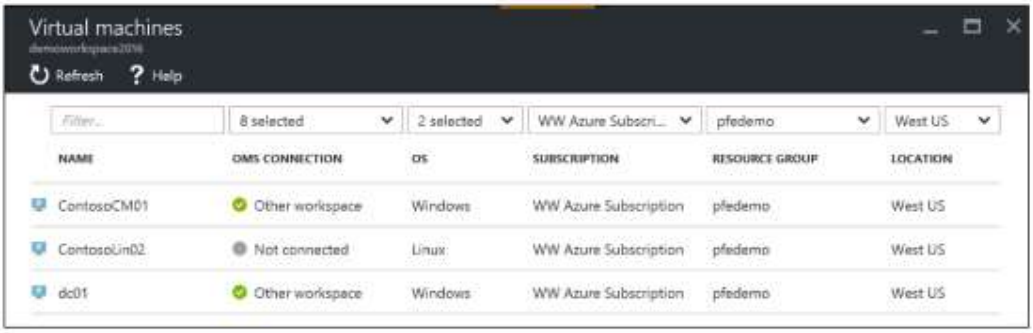


Рис. 5-13. Виртуальные машины в группе ресурсов

Здесь список содержит три виртуальные машины, из которых две подключены к другой рабочей области OMS, а третья не подключена ни к одной. Если нажать кнопку мыши на этой виртуальной машине, отобразится возможность использовать расширения виртуальных машин Azure для установки агента OMS и автоматической регистрации этой виртуальной машины в вашей рабочей области, как показано на рис. 5-14.

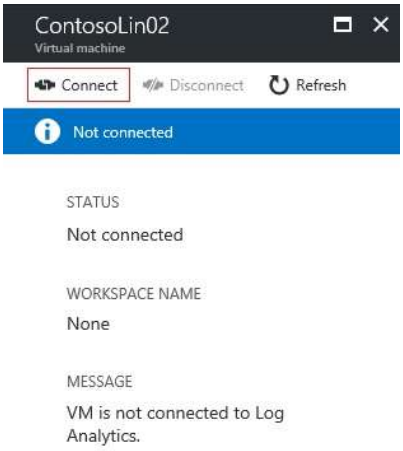


Рис. 5-14. Автоматическое подключение виртуальной машины к OMS

Вернитесь на главную страницу OMS и в разделе «Источники данных рабочей области» (Workspace Data Sources) нажмите «Журналы учетных записей хранения» (Storage accounts logs). По умолчанию это окно пусто, как показано на рис. 5-15. Необходимо добавить учетную запись хранения, в которой можно будет сохранять данные журналов из различных источников. OMS будет использовать эту учетную запись для получения информации.

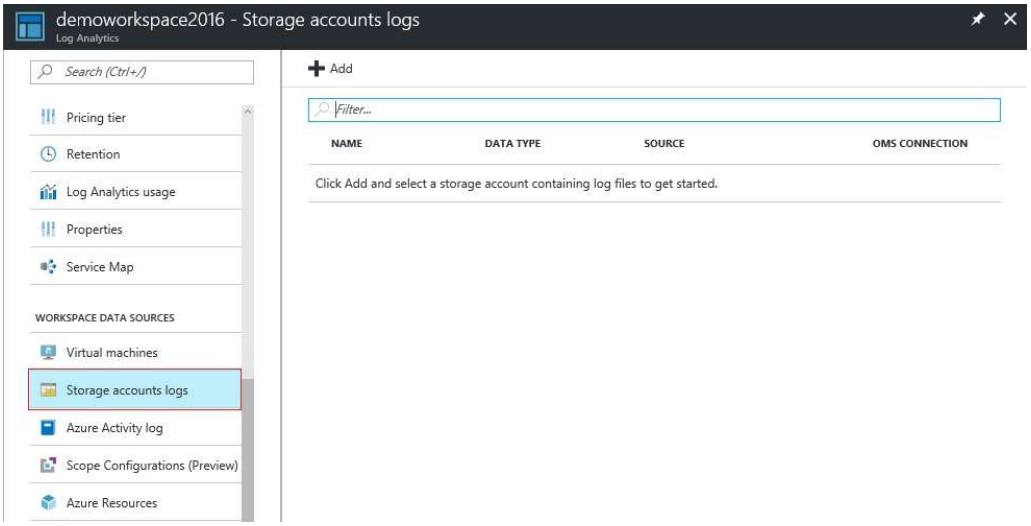


Рис. 5-15. Подключение учетной записи хранения для OMS

Нажмите кнопку «Добавить» (Add), чтобы открыть окно «Добавление журнала учетной записи хранения» (Add storage account log). Здесь необходимо указать определенную информацию, в первую очередь — выбрать учетную запись хранения, которую нужно использовать. Затем следует выбрать тип данных. Например, можно выбрать:

- журналы IIS;
- события;
- системные журналы (Linux);
- журналы ETW;
- события Service Fabric.

На рис. 5-16 показан пример окна с выбранными параметрами.

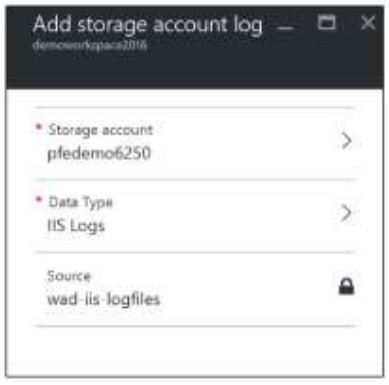


Рис. 5-16. Окно «Добавление учетной записи хранения» с выбранными параметрами

Здесь вам доступно множество параметров. Можно выбирать их в различных сочетаниях, добавлять, если требуется, дополнительные ресурсы по мере расширения или дополнительные учетные записи хранения, но пока не будем переходить на главный портал OMS.

Вернитесь на страницу Log Analytics и нажмите на рабочую область, с которой собираетесь работать. На открывшейся странице в разделе «Рабочая область OMS» (OMS Workspace) нажмите «Портал OMS» (OMS Portal), как показано на рис. 5-17.

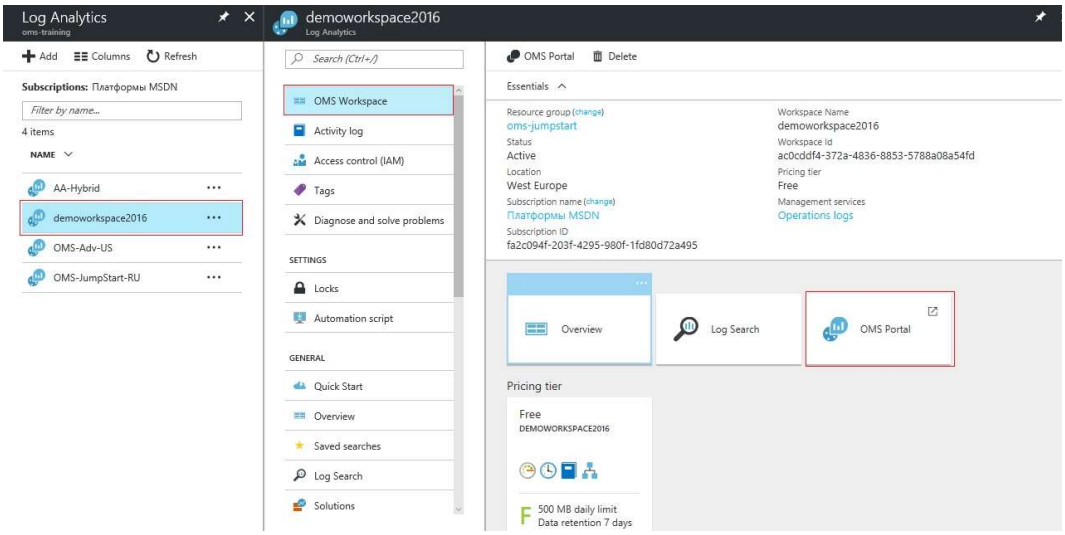


Рис. 5-17. Нажмите на рабочую область, чтобы перейти к управлению ею

В окне появятся некоторые базовые параметры, одним из которых имеет смысл воспользоваться. В Azure у многих служб есть возможность записывать файлы журналов непосредственно в учетную запись хранения. Можно добавить эту учетную запись в рабочую область, чтобы затем анализировать собранные данные.

При первом входе в эту рабочую область нажмите «Настройки» (Settings), как показано на рис. 5-18.

> **Примечание.** Если уже есть заранее настроенные источники данных портала Azure, то они будут здесь показаны.

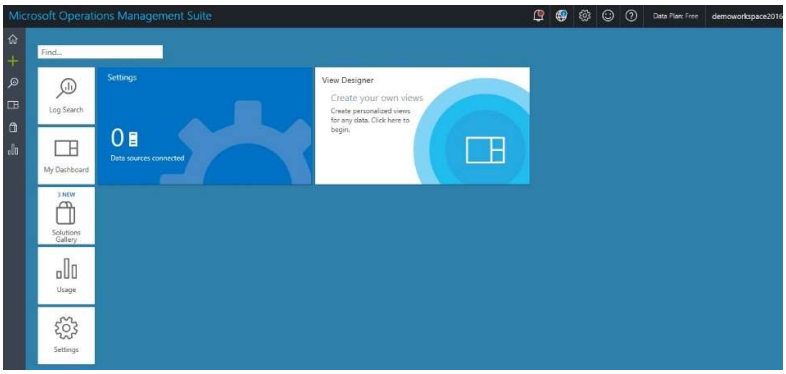


Рис. 5-18. Главная рабочая область

В начале работы с OMS необходимо выполнить три основные задачи. Если нажать «Настройки» (Settings), откроется окно «Решения» (Solutions). Решения — это аналог пакетов управления в OMS, они содержат все функции аналитики и правила, по которым будет оцениваться работа виртуальных машин в выбранной среде. Решения часто обновляются, новые решения постоянно разрабатываются на основе требований и запросов клиентов и добавляются в общий портфель.

На рис. 5-19 показан первый шаг настройки OMS. Чтобы приступить к работе, нужно выбрать решения. В области слева нажмите «Решения» (Solutions). Решения не будут работать до тех пор, пока вы не выберете компьютеры, к которым их следует применить. Можно выбрать все решения или только те, работа которых вас интересует.

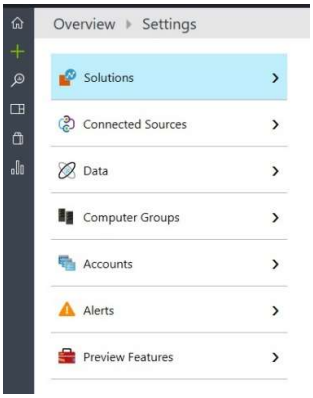


Рис. 5-19. Шаг 1. Выбор решений

Затем нажмите «Подключенные источники» (Connected Sources). На рис. 5-20 показан набор источников, доступных для вашей среды.

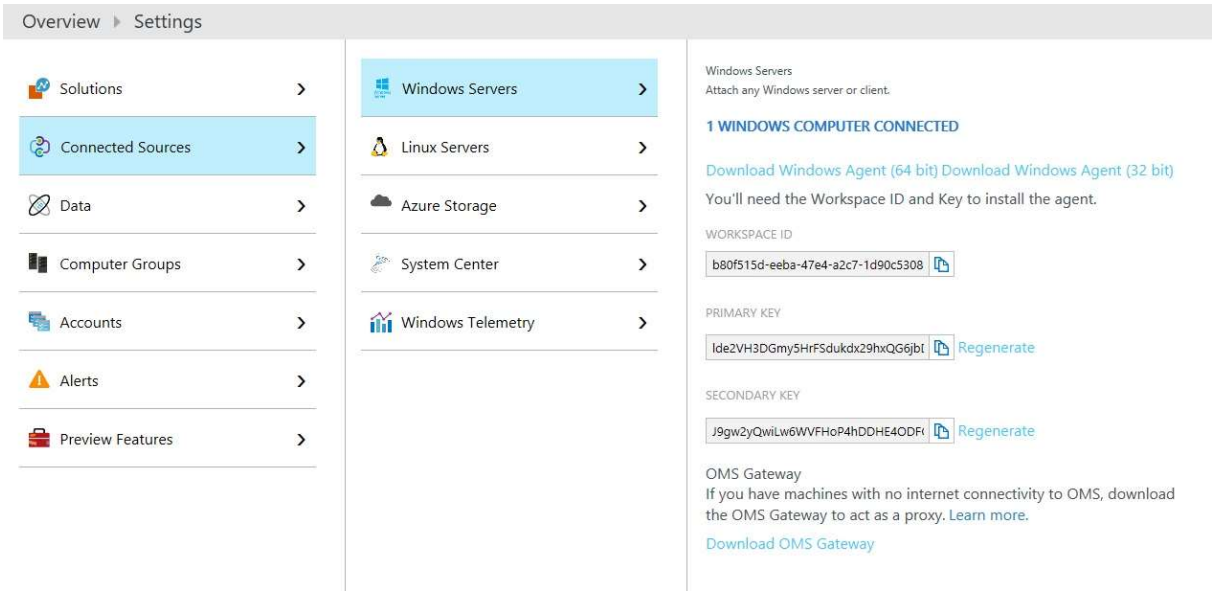


Рис. 5-20. Шаг 2. Подключение источников

Здесь нужно ответить на три простых вопроса:

- Требуется ли развернуть агент непосредственно на компьютере и зарегистрировать его непосредственно в OMS?
- Требуется ли подключить Operations Manager к OMS?
- Нужно ли добавить учетную запись хранения с данными журналов?

От ответов на эти вопросы зависят действия при установке. Если нужно, чтобы компьютер назначения предоставлял информацию напрямую в OMS, загрузите агент и установите его на этот компьютер. При установке потребуется выбрать тип развертывания, для которого надо зарегистрировать агент. В качестве агента используется Microsoft Monitoring Agent. Его можно зарегистрировать непосредственно в OMS или на сервере Operations Manager, как показано на рис. 5-21.

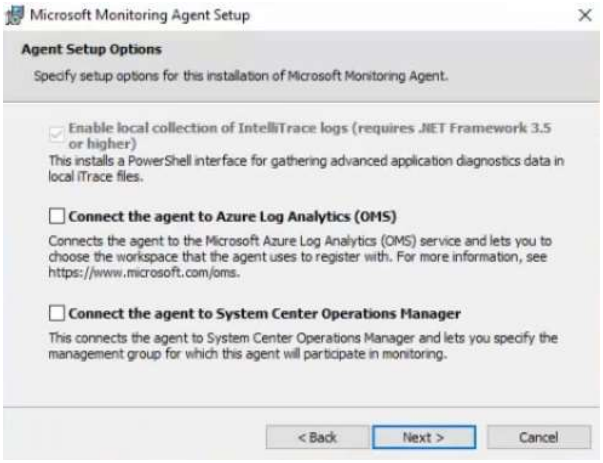


Рис. 5-21. Установка Microsoft Monitoring Agent

Если установить флажок «Подключить агент к службе анализа журналов Azure (OMS)» (Connect the agent to Azure Log Analytics (OMS)), потребуется ввести идентификатор и ключ рабочей области. Их можно получить в области «Подключенные источники» (Connected Sources), как показано на рис. 5-20, и скопировать в соответствующие поля, как показано на рис. 5-22.

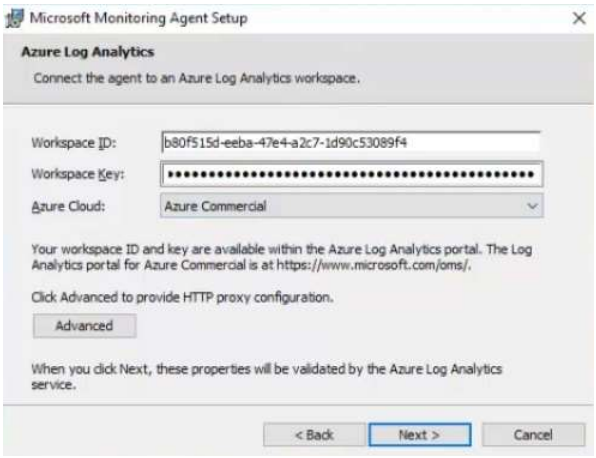


Рис. 5-22. Настройка идентификатора и ключа рабочей области

Агент завершит установку и зарегистрируется в рабочей области OMS. После регистрации агента в OMS в рабочей области OMS появится один подключенный сервер.

Далее можно настроить сбор дополнительных данных из используемых источников. На рис. 5-23 показаны различные типы журналов, доступные для выбора. Например, **в разделе «Журналы событий Windows» (Windows Event Logs)** можно ввести в поле поиска слово **free** или **System**, и вы увидите список доступных журналов. Не забудьте нажать на кнопку «Сохранить» (Save).

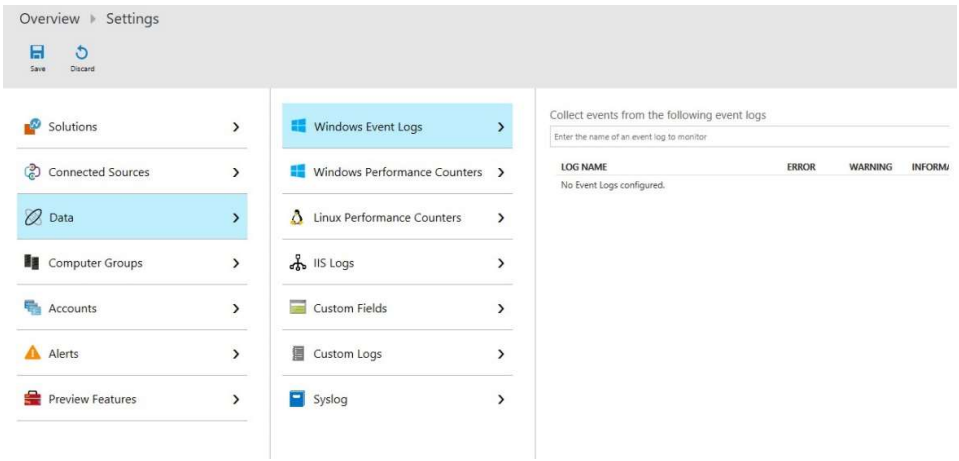


Рис. 5-23. Добавление журналов

Правила будут загружены в агент обычным образом и затем обработаны. Данные будут передаваться на портал и анализироваться. Основная коллекция решений будет пополняться последними данными, полученными из системы. При необходимости можно добавить дополнительные решения из коллекции решений.

На рис. 5-24 показана обновленная панель мониторинга после передачи информации.

Рис. 5-24. Обновленная панель мониторинга

Можно нажать на каждую панель мониторинга — это позволит вам просмотреть подробные данные. Здесь можно настроить дополнительные функции: автоматизацию, резервное копирование и Azure Site Recovery. Все три функции можно использовать в гибридных сценариях для управления облачными ресурсами и локальными ресурсами из облака.

С этого момента можно использовать поиск по журналу (Log Search) и все дополнительные решения, как показано на рис. 5-25.

Рис. 5-25. Коллекция решений в OMS

> **Примечание.** Более подробная информация об OMS доступна по адресу https://www.microsoft.com/cloud-platform/operations-management-suite-resources.

Инструменты управления серверами

В современных компаниях гибридная инфраструктура и гибридное развертывание систем используются все чаще, рабочие нагрузки распределяются по облакам, и в результате трудоемкость управления всеми этими решениями многократно возрастает. Это, разумеется, нехорошо: требуется более контролируемый способ управления ресурсами, которые могут находиться как в локальной среде, так и в Azure.

В составе инструментов управления серверами (SMT) появился графический веб-интерфейс, размещенный в Azure, и средства командной строки, реализующие необходимые возможности управления в Windows Server 2016. Например, с помощью этого графического интерфейса администраторы могут управлять сервером Nano Server или Server Core, не устанавливая дополнительных компонентов.

В настоящее время поддерживаются следующие возможности:

- просмотр и изменение конфигурации системы;
- просмотр производительности ресурсов, управление процессами и службами;
- управление устройствами, подключенными к серверу;
- просмотр журналов событий;
- просмотр списка установленных ролей и компонентов;
- использование консоли Windows PowerShell для управления и автоматизации.

На рис. 5-26 показана схема развертывания с инструментами управления серверами.

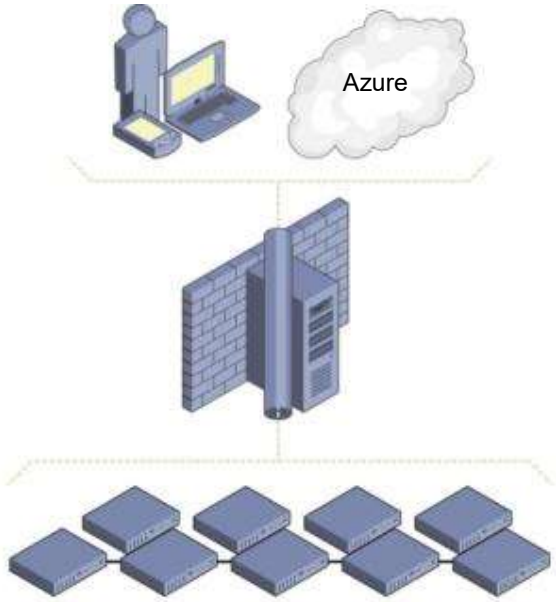


Рис. 5-26. Пример развертывания инструментов управления серверами

На этой схеме видно, что сервер шлюза (в центре схемы) необходим, чтобы локальная инфраструктура могла обмениваться данными со службой в Azure.

Поддержка SMT в Windows 2012 и более поздних версиях

В Windows Server 2016 не нужна установка никаких дополнительных компонентов, но если вы используете одну из предыдущих версий Windows (например, 2012 или 2012 R2), то для управления серверами Windows Server 2016, в том числе Nano Server, необходимо установить WMF 5.0.

Все инструменты SMT, кроме центра обновления Windows и диспетчера устройств, работают с Windows Server 2012 и 2012 R2. При планировании SMT и использовании этих инструментов для управления прежними версиями Windows следует учесть один важный фактор — взаимозависимость приложений, установленных на сервере. Например, будет ли нарушена работа приложения, если установить более новую версию WMF?

> **Примечание.** Чтобы проверить, можно ли установить WMF 5.0 перед подключением сервера к SMT, перейдите по адресу https://msdn.microsoft.com/en-us/powershell/wmf/5.0/productincompat.

Чтобы убедиться в правильности работы приложений с WMF 5.0, можно провести дополнительные проверки.

Постоянные учетные данные

В инструментах управления серверами (SMT) Windows Server 2016 можно сохранить учетные данные: они сохраняются в Azure в зашифрованном виде (используется алгоритм AES256). За шифрование учетных данных перед отправкой в Azure отвечает шлюз. Он должен их зашифровать с помощью имеющегося сертификата (существующего только на шлюзе). Расшифровать учетные данные тоже может только шлюз, используя тот же сертификат, который использовался для шифрования. Как уже было сказано, этот сертификат всегда находится только на шлюзе и никогда не выходит за его пределы.

Правила брандмауэра

Централизованное управление брандмауэром Windows обеспечивает ряд преимуществ для серверов за счет применения стандартной политики. К сожалению, работа с брандмауэром Windows вне традиционных корпоративных средств мониторинга обычно не отличалась удобством: невозможно было управлять сразу множеством серверов; не всегда просто было

получить полную информацию о том, какие правила включены и в каком состоянии они находятся.

В SMT имеется графический интерфейс для поиска правил брандмауэра на каждом конкретном компьютере, это помогает быстрее и точнее оценить ситуацию (рис. 5-27).

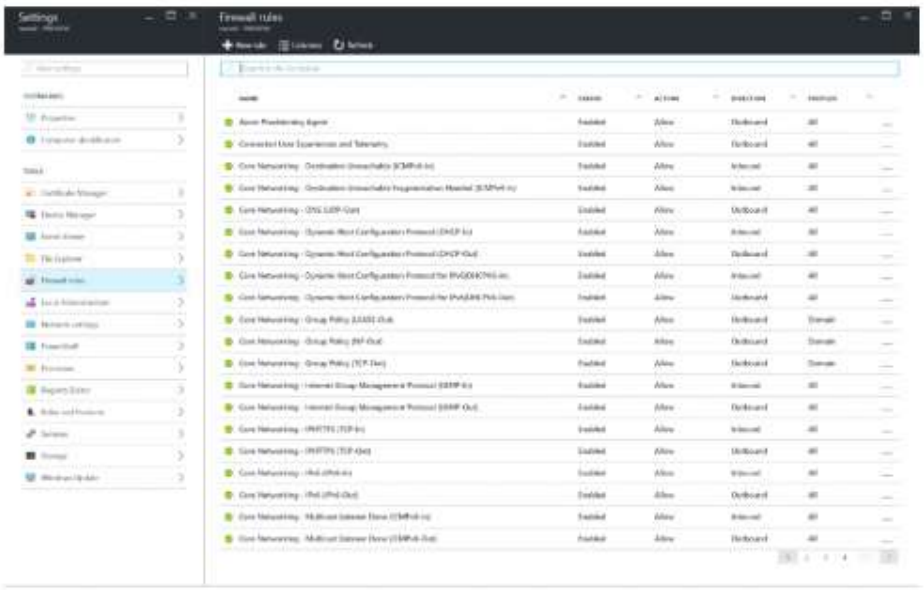


Рис. 5-27. Правила брандмауэра в SMT

Усовершенствования редактора сценариев Windows PowerShell

Редактор сценариев Windows PowerShell в SMT обновлен, теперь он поддерживает возможности работы с файлами на управляемых компьютерах: сценарии на них можно открывать, редактировать и сохранять.

Редактор сценариев также получил возможность подключаться напрямую к хранилищу BLOB-объектов Azure (рис. 5-28) и сохранять сценарии в нем. После этого сценарии становятся доступными для всех серверов, на которые действует подписка, и даже для других систем.

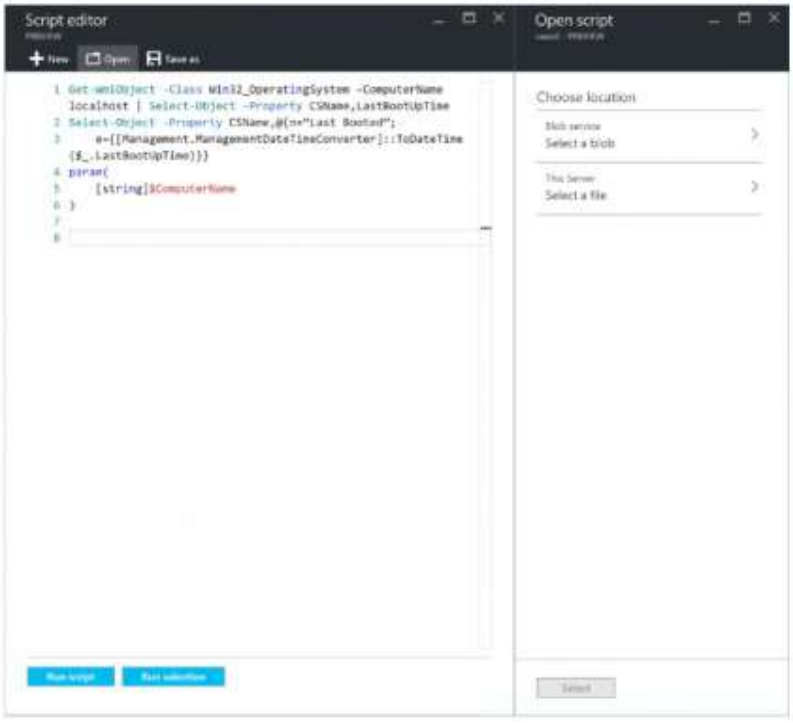


Рис. 5-28. Редактор сценариев Windows PowerShell в SMT подключен к хранилищу BLOB-объектов

Проводник

Редактор сценариев Windows PowerShell обладает базовыми возможностями по взаимодействию и работе со сценариями на определенных компьютерах. Кроме того, доступны простые действия по управлению файлами: можно просматривать файлы, переименовывать и удалять их. На рис. 5-29 показан пример проводника в SMT.

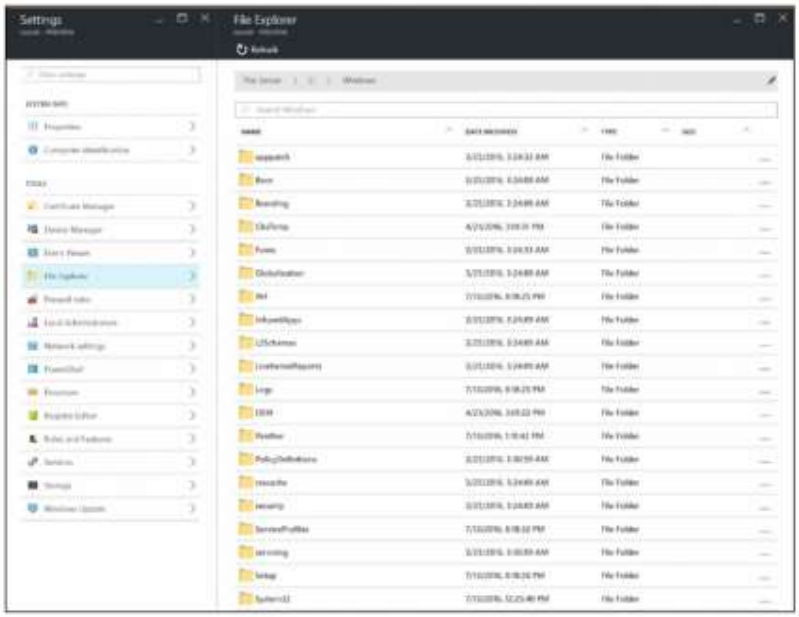


Рис. 5-29. Проводник отображает содержимое компьютера в SMT

Локальные хранилища

SMT теперь может предоставлять подробную информацию о хранилищах данных на каждом компьютере (рис. 5-30). Доступна информация о накопителях, томах и общих файловых ресурсах. Пока эта информация доступна только для чтения, но со временем будут реализованы и другие возможности.

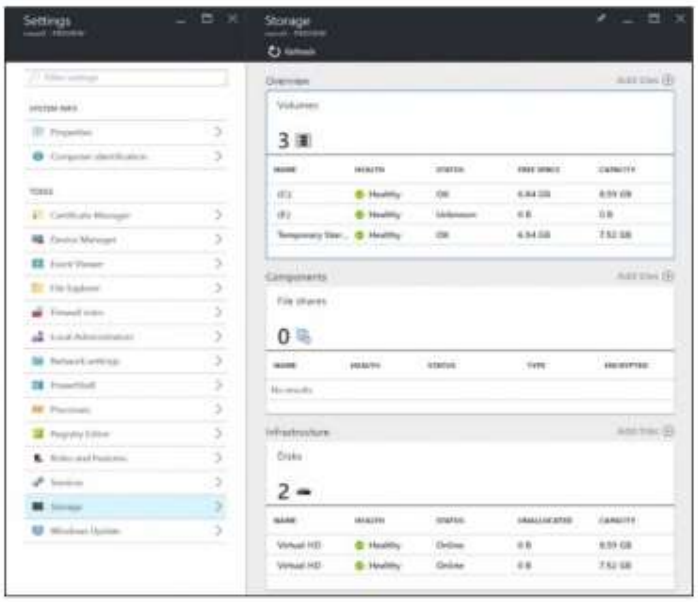


Рис. 5-30. Информация о хранилище в SMT

Диспетчер сертификатов

Управление сертификатами в любой ИТ-организации может быть сопряжено с определенными трудностями. Например, как проверить сертификаты для множества компьютеров, если нет своего центра сертификации? В SMT диспетчер сертификатов появился, и теперь с его помощью можно удаленно управлять сертификатами на указанных компьютерах. На рис. 5-31 показаны функции просмотра всех сертификатов или их подмножества, просмотра журнала событий и управления жизненным циклом сертификатов с возможностью их импорта, экспорта и удаления.

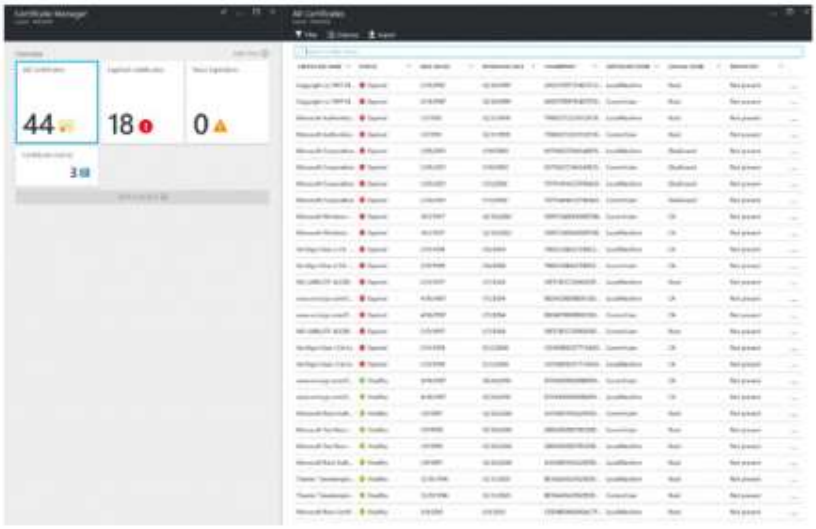


Рис. 5-31. Диспетчер сертификатов в SMT

Развертывание

Развертывание SMT осуществляется без особых затруднений. Тем не менее при этом используется Azure, поэтому требуется подписка Azure. Получить ее можно разными способами, самый простой из них — перейти на сайт https://azure.microsoft.com/free/. Здесь можно создать подписку, если у вашей организации ее еще нет.

Серверу шлюза, который вы создадите, также требуется доступ к Интернету, поэтому он должен находиться в маршрутизируемой подсети вашей организации.

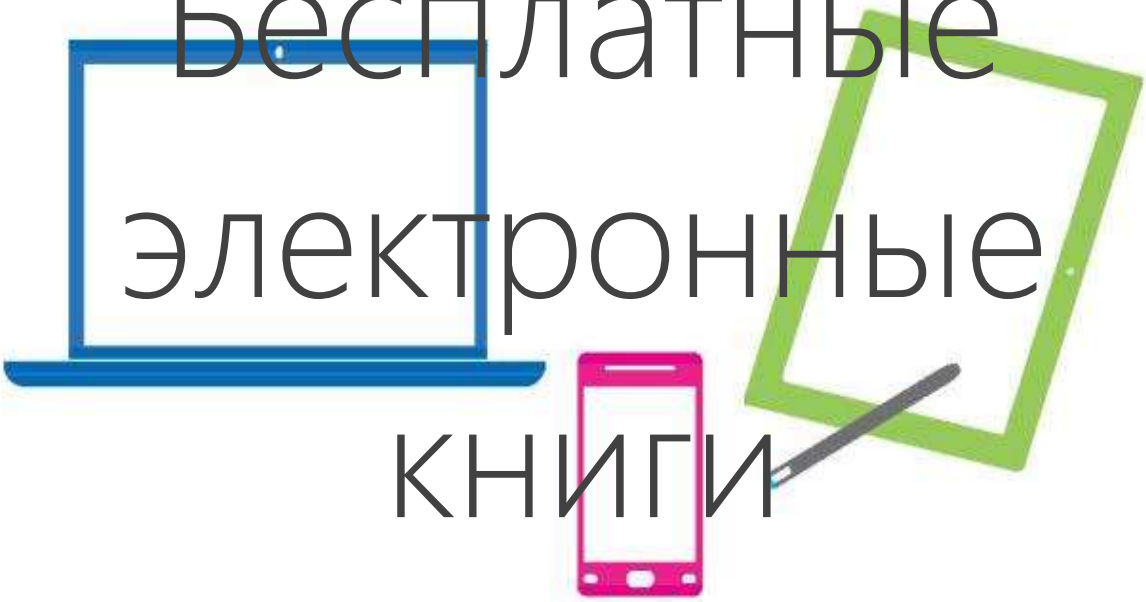
Существуют два способа развертывания SMT: с помощью портала Azure и с помощью Windows PowerShell. Для развертывания графического пользовательского интерфейса перейдите по адресу https://blogs.technet.microsoft.com/servermanagement/2016/08/17/deploy-setup-server-management-tools/. Чтобы использовать Windows PowerShell, перейдите по адресу http://social.technet.microsoft.com/wiki/contents/articles/35196.microsoft-azure-managing-nano-server-with-server-management-tools.aspx.

> **Примечание.** Более подробная информация об SMT доступна в блоге группы разработчиков этого продукта по адресу https://blogs.technet.microsoft.com/servermanagement/

Об авторе

Джон Мак-Кейб (John McCabe) работает в корпорации Microsoft в должности старшего инженера по эксплуатации. На этой позиции он работал с крупнейшими в мире клиентами, занимался поддержкой и внедрением передовых решений на основе технологий Microsoft. Он также отвечает за разработку основных служб для отделов корпоративных услуг. Джон был соавтором нескольких книг, в том числе *«Освоение Windows Server 2012 R2» (Mastering Windows Server 2012 R2)* издательства Sybex, *«Освоение Lync 2013» (Mastering Lync 2013)* издательства Sybex и *«Введение в Microsoft System Center 2012» (Introducing Microsoft System Center 2012)* издательства Microsoft Press. Джон выступал на множестве конференций в Европе, в том числе на конференциях TechEd и TechReady. Перед переходом в корпорацию Microsoft Джон работал ведущим специалистом в компании Unified Communications и располагает 15-летним опытом консультаций по различным технологическим направлениям, включая сети, безопасность и архитектуру.

Бесплатные электронные книги



От технических обзоров до подробного анализа специализированных тем — получите *бесплатные* электронные книги издательства Microsoft Press по адресу:

www.microsoftvirtualacademy.com/ebooks

Загрузите бесплатные электронные книги в формате PDF, EPUB или Mobi для устройств Kindle.

Множество других интересных и полезных ресурсов вы найдете на сайте Microsoft Virtual Academy, где можно приобрести новые навыки и пройти бесплатное обучение у специалистов Microsoft, полезное для дальнейшего карьерного роста.

Microsoft