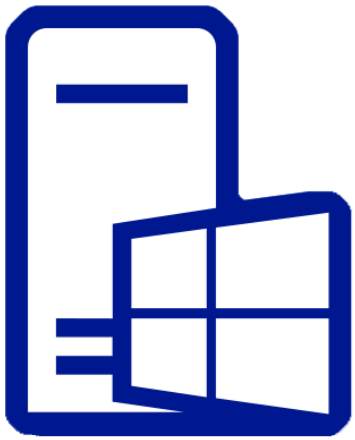
А.Д. Береснев, О.М. Ромакина

ПРАКТИЧЕСКИЕ РАБОТЫ ПО СИСТЕМНОМУ АДМИНИСТРИРОВАНИЮ В WINDOWS SERVER



Санкт-Петербург
2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А.Д. Береснев, О.М. Ромакина

ПРАКТИЧЕСКИЕ РАБОТЫ ПО СИСТЕМНОМУ АДМИНИСТРИРОВАНИЮ В WINDOWS SERVER

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлениям подготовки
09.03.03 Прикладная информатика,
11.03.02 Инфокоммуникационные технологии и системы связи,
45.03.04 Интеллектуальные системы в гуманитарной сфере
в качестве учебного пособия для реализации основных профессиональных образовательных программ высшего образования бакалавриата,

Санкт-Петербург

2025

Береснев А.Д., Ромакина О.М. Практические работы по системному администрированию в Windows Server – СПб: Университет ИТМО, 2025. – 39 с.

Рецензент: Маятин Александр Владимирович, доцент, кандидат педагогических наук, доцент факультета информационных технологий и программирования Университета ИТМО.

В учебно-методическом пособии содержатся описания практических работ по системному администрированию сетей на платформе Windows Server. Большое внимание уделено общим вопросам и подходам к автоматизации процессов управления инфраструктурой. Содержащиеся в пособии практические работы содержат в качестве учебных реальные, типовые задачи системного администрирования, с которыми часто сталкиваются инженеры, управляющие корпоративной инфраструктурой. Для выполнения работ используются технологии виртуализации, что снимает необходимость в специальном аппаратном обеспечении. Пособие снабжено приложениями, содержащими термины и определения, понимание которых обычно вызывает затруднения у студентов. Кроме этого, добавлено обширное приложение о языке сценариев PowerShell, содержащее сведения об основных языковых конструкциях и особенностях языка. Знакомство с этими рекомендациями может оказаться полезным не только для выполнения практических работ, но и для профессиональной деятельности.

Содержание

Введение

Данные практические работы предназначены для освоения дисциплин «Администрирование в инфокоммуникационных системах», «Администрирование Windows Server» и «Администрирование компьютерных сетей» в рамках образовательных программ направлений подготовки 09.03.03 Прикладная информатика, 11.03.02 Инфокоммуникационные технологии и системы связи, 45.03.04 Интеллектуальные системы в гуманитарной сфере, а также для студентов других направлений подготовки, интересующихся вопросами системного администрирования сетей на платформе Windows Server.

Основной целью пособия является формирование у обучающихся знаний, умений и практических навыков установки и настройки операционной системы Windows Server, управления конфигурациями серверов через командную строку и PowerShell, навыков создания и управления доменными структурами с использованием Active Directory, развертывания сетевых инфраструктурных сервисов на примере DHCP, организации мониторинга производительности и восстановления работоспособности серверов.

В пособии содержатся пошаговые инструкции и методические рекомендации по выполнению практических работ, направленных на изучение основных инструментальных средств управления Windows Server, принципов работы с Active Directory и основными средствами мониторинга и диагностики. Выполнение практических работ способствует формированию у студентов компетенций, включающих способность администрировать инфокоммуникационные системы, навыки работы с серверными операционными системами, умение использовать базовые инструменты администрирования и конфигурирования инфокоммуникационных систем.

Описания практических работ

Работа №1.
Основные инструментальные средства управления Windows Server

Цель работы: получить представление и практические навыки работы с основными инструментальными средствами управления Windows Server на примере управления локальными учетными записями и параметрами сетевых интерфейсов [1].

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жестких дисков операционных систем **Windows Server 2012/2016.**

Краткие теоретические сведения:

ОС Windows Server содержит в себе разнообразные средства управления. Для разовых и непериодических действий подходят средства GUI:

- элементы Панели управления,
- стандартные графические консоли Microsoft System Console,
- сборные графические консоли на основе MMC (Microsoft Management Console),
- утилита ServerManager.exe.

Для оркестрации и пакетного выполнения заданий используются:

- утилиты командной строки и скрипты (BAT/CMD),
- командлеты PowerShell и скрипты PowerShell. Это средство считается основным.

Элементы Панели управления (например **sysdm.cpl** или **firewall.cpl**) и стандартные графические консоли (например **fsmgmt.msc, compmgmt.msc**) могут быть вызваны через GUI прямо из командной строки.

Сборные графические консоли на основе MMC создаются с помощью утилиты **mmc.exe**. С ее помощью можно собрать в одной консоли стандартные графические консоли, скрипты и внешние программы. Причем консоли можно запускать для управления не только локальным, но и удаленным компьютером.

Утилиты командной строки - утилиты с текстовым вводом-выводом. Для работы с ними используется командный интерпретатор cmd.exe. Утилиты могут вызываться в командных файлах (BAT\CMD). В Windows текстовый файл является скрипом, если имеет расширение .bat или .cmd .

Современным консольным средством является PowerShell - расширяемое средство автоматизации от Microsoft с открытым исходным кодом, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев [2].

С ОС поставляется интегрированная среда сценариев Windows PowerShell ISE – облегченная IDE для PowerShell. Для разработки подходит MS Visual Studio Code.

Но скрипты, конечно, можно писать в текстовом редакторе.

Порядок выполнения работы:

Использование средств управления будет осуществляться на примере процессов создания локальных пользователей и групп, а также настройки сетевых интерфейсов.

Часть 1. Консоль MMC.

1. Запустите Windows Server, авторизуйтесь с правами администратора.
2. С помощью утилиты mmc.exe создайте свою консоль, включающую в себя консоли:
   - Управление компьютером,
   - Управление службами,
   - Просмотр журнала событий,
   - Управление общими папками
3. Добавьте в консоль Папку (назовите ее Tools), а в нее кнопку Задачи запуска окна Сетевые подключения.
4. Сохраните оснастку.
5. Создайте с помощью вашей консоли пользователя с именем UPart1*FIO* и группу GPart1FIO (где FIO - ваши инициалы),
6. Включите пользователя в группу.
7. Сохраните вашу консоль в папку C:\Console\
8. Зайдите в систему под новым пользователем.
9. Запустите вашу консоль, попробуйте перезапустить с ее помощью службу DNS-клиент.
10. С помощью команды runas или с помощью GUI запустите оснастку от имени Администратора. Попробуйте перезапустить с ее помощью службу DNS-клиент.
11. Настройте IP-адреса на сетевом интерфейсе по следующим параметрам:
    - **IP 192.168.1.10**
    - **mask 255.255.255.0**

- **gateway 192.168.1.1**
- **DNS 8.8.8.8**

Часть 2. Утилиты командной строки CMD

1. Запустите Windows Server, авторизуйтесь с правами администратора
2. Создайте скрипт script21.cmd с использованием консольной утилиты net. Скрипт выполняет следующие действия:
   - Запрашивает у пользователя Строку (без пробелов, 4 символа),
   - Создает пользователя с именем UPart2STR и группу GPart2STR (где STR полученная строка),
   - Включает пользователя в группу,
   - Активирует пользователя.
3. С помощью консольной утилиты netsh создайте скрипт script22.cmd, который:
   - Запрашивает у пользователя параметр Auto или Manual;
   - Если пользователь выбирает Auto, то настраивает получение IP адреса, маски, шлюза и DNS автоматически;
   - Если выбран Manual, то устанавливает параметры из п. 11 часть 1:
4. Добавьте скрипты в папку Tools вашей консоли.

Часть 3. Использование PowerShell

1. Запустите Windows Server, авторизуйтесь с правами администратора
2. Создайте PowerShell-скрипт script31.ps1. Скрипт выполняет следующие действия:
   - Запрашивает у пользователя Строку (без пробелов, 4 символа)
   - Создает пользователя с именем UPart3STR и группу GPart3STR (где STR полученная строка),
   - Включает пользователя в группу,
   - Активирует пользователя.
3. Создайте PowerShell-скрипт script32.ps1, который:
   - Запрашивает у пользователя параметр Auto или Manual
   - Если пользователь выбирает Auto, то настраивает получение IP-адреса, маски, шлюза и DNS автоматически.
   - Если выбран Manual, то устанавливает параметры из п. 11 часть 1:
4. Добавьте скрипты в папку Tools вашей консоли.
5. Сохраните копию консоли, так, чтобы ее было невозможно изменить пользователю.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет

содержит титульный лист, артефакты выполнения и ответы на вопросы.

Артефакты:

1. Скриншот финальной версии консоли
2. Тексты скриптов из части 2 и части 3.

Вопросы:

1. В каких группах оказался пользователь после п.6 Части 1?
2. Сравните организацию диалога в скриптах CMD и PowerShell. Приведите результаты сравнения.
3. Как используя вашу оснастку, управлять службами и пользователями на удаленном компьютере Windows?

Работа №2.
Основы работы с Active Directory в Windows Server

Цель работы: получить базовые навыки развертывания службы каталогов Active Directory на основе Windows Server, управления объектами AD, их правами и групповыми политиками [3, 4].

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жёстких дисков операционных систем **Windows Server 2012 R2/2016/2019.**
- Доступ к Microsoft Evaluation Center (https://www.microsoft.com/ru-ru/evalcenter)

Краткие теоретические сведения:

Для централизованного управления ресурсами сети применяют распределенные системы – службы каталогов. Эти системы позволяют хранить данные об объектах и субъектах безопасности в специализированной распределенной, защищенной базе данных - службе каталогов. На рынке существуют несколько популярных служб каталогов. Например, Novell eDirectory, OpenLDAP и Microsoft Active Directory (далее AD). Последняя является службой каталогов для сетей Windows. Структурно AD построена по принципу DNS и имеет подобную древовидную структуру. Сама AD использует механизмы DNS для поиска служб и организации взаимодействия компонентов сервиса.

Доступ к объектам каталога осуществляется по протоколу LDAP. В службах каталогов присутствуют объекты двух типов - контейнеры и листья (по ассоциации с деревом).

Основной единицей хранения в AD является домен. Домен – контейнерный объект, представляющий собой фрагмент AD хранящийся на специальном компьютере с Windows Server. Домен может содержать объекты-контейнеры (Organization Unit) и конечные объекты (User, Group,

Computer и т.п.). Домены AD могут объединяться в деревья, деревья в конгломераты более высокого уровня – леса. В AD относительно домена может строиться распределенная система, в которой копии домена хранятся на нескольких Windows Server, работающих в режиме контроллера домена.

Домены и другие контейнеры предназначены для объединения других объектов и распространения групповых политик. Групповые политики — это шаблоны, которые накладываются на реестр Windows и применяются для ассоциированных с ними объектов. Так, если в домене firma.loc создан Organization Unit с именем dev , а в нем пользователь supervisor, то при регистрации пользователя supervisor к его рабочей станции применяются среди прочих, групповые политики, привязанные к контейнеру dev [3].

Для управления объектами AD используются средства GUI, консольные утилиты dsquery, dsmod, dsadd, dsrm, dsget и набор командлетов Power Shell.

Для разграничения прав на доступ к файловым объектам на платформе Windows используется механизм ACL в файловой системе NTFS, в которой реализована возможность достаточно гибкого управления правами доступа к файлам и каталогам.

Совет 1. После выполнения работы необходимо сохранить снимки состояния виртуальных машин для использования в последующих работах.

Совет 2. Перед выполнением работы ознакомьтесь с требованиями к содержанию отчета, чтобы собирать необходимые артефакты выполнения.

Порядок выполнения работы:

Часть 1. Подготовительная

1. Для выполнения работы понадобится две виртуальные машины Windows Server и Windows 10 Pro или Enterprise.
2. Дистрибутивы операционных систем со сроком действия 90 дней можно скачать с сайта Microsoft Evaluation Center (https://www.microsoft.com/ru-ru/evalcenter).
3. Установите операционные системы, сделайте снимки машин. Переименуйте виртуальные машины в ad-srv, и ad-client соответственно версии операционной системы.
4. Настройте виртуальные машины так, чтобы они оказались в одной, изолированной LAN. Для сервера выберите и настройте адрес из сети 10.0.0.0/8. В качестве DNS севера установите адресе самого сервера.

Часть 2. Развертывание Active Directory

1. Подготовьте компьютер «AD-Srv» к развертыванию AD (новый домен, новый лес) с установкой DNS на «Ad-srv». С помощью мастера добавления ролей и компонентов и диспетчера серверов развернуть домен с именем «ваши_FIO».local. Автоматически

установите и настройте DNS.

2. После установки перезагрузите компьютер.

3. Установите DHCP-сервер и произведите его настройку (используйте адресный пул 10.0.0.100-10.0.0.110, обеспечьте получение клиентами адреса DNS и шлюза равных адресу сервера). Проведите авторизацию DHCP сервера. После установки перезагрузите компьютер.

4. Убедитесь, что компьютер ad-client получил необходимую конфигурацию ip. Подключите компьютер ad-client к домену.

5. Войдите на ad-client с учетной записью администратора домена.

6. На контроллере домена ad-srv в оснастке «Active Directory пользователи и компьютеры» найдите объект компьютера ad-client и компьютера ad-srv.

Часть 3. Объектами AD и правами на NTFS и SMB.

1. Используя административную оснастку «Active Directory пользователи и компьютеры», создайте в новом домене 2 подразделения (Organization Unit): ouSellers, ouManagers. В каждом подразделении создайте пользователя: uSeller1, uManager1 и группы gSellers и gManagers.

2. На сервере на диске C:\ создайте каталог «AllUsers» и дайте всем пользователям домена право на чтение этого каталога. В нем создайте каталоги Sellers и Managers, дайте членам групп gSellers и gManagers все права на уровне NTFS для соответствующих каталогов кроме возможностей изменения прав и удаления самих каталогов. При этом следует сохранить возможность создавать, удалять и модифицировать файлы и каталоги внутри самих каталогов. Создайте каталог AllUsers\BlackHole, в который пользователи созданных групп смогли бы копировать файлы "drag-and-drop", но не просматривать содержимое. Создайте каталог AllUsers\Common, в который все пользователи домена смогли бы писать файлы, но удалять смогли бы только свои. Откройте общий доступ через сеть к каталогу AllUsers с необходимыми разрешениями и назначьте сетевое имя AllUsersCom.

3. На диске C: сервера создайте папку UsersHome. Для каждого созданного в п. 1 части 3 пользователя создайте домашнюю папку c:\UsersHome\”имя пользователя“. Обеспечьте пользователю возможность записи через сеть (протокол SMB) в свой домашний каталог, причем имя сетевой папки должно быть скрытым, т. е. при просмотре списка папок компьютера в «Сетевом окружении» папку не должно быть видно.

4. В свойствах каждого пользователя задайте подключение

домашней папки на диск X: и место хранения перемещаемого профиля. Обратите внимание на то, что необходимо использовать сетевые пути UNC.

5. Используя машину «Ad-client», авторизуйтесь в системе под пользователем uSeller1, перезагрузите клиентский компьютер, выполните повторную аутентификацию и изучите данные в каталоге x:_profile.

Часть 4. Работа с групповыми политиками.

1. С помощью консоли Управление групповой политикой измените групповую политику домена, так чтобы пароли могли быть длиной 6 символов без контроля сложности.

Примечание: после создания политики она применяется не мгновенно, а согласно периоду обновления, заданному политикой домена. Для принудительного обновления политики можно использовать команду gpupdate [1].

2. Создайте групповую политику для контейнера ouSellers, с помощью которой будет:
   a. Запрещен доступ к Панели управления,
   b. Установлена блокировка экрана при периоде неактивности 1 минута, с отключением возможности менять этот параметр.
   c. Запретить пользователю редактировать реестр
   d. Скрыть в проводнике диск C:
3. Создайте групповую политику в контейнере ouManagers, которая будет определять приложения, которые может запускать пользователь:
   a. Paint;
   b. calc;
   c. Notepad.
4. Создайте контейнер для объектов – компьютеров и создайте в нем групповую политику, которая:
   a. отключает сбор и передачу в Microsoft сообщений об ошибках,
   b. отключит локальные учетные записи Администратор (Administrator)
   c. запретит пользователю пользоваться механизмом Offline Files
   d. установит на клиентских компьютерах для всех файловых объектов на диске C:\ следующий ACL (Администраторы, Система – полный доступ, Пользователи домена – чтение, просмотр каталогов, выполнение файлов).
5. Создайте отдельную групповую политику, с помощью которой разверните на клиентском компьютере программу 7-zip

(инсталлятор MSI).

6. Проверьте функционирование политик.

Часть 5. Автоматизация работы с объектами AD

1. Напишите скрипт на PowerShell, получающий в качестве параметра путь к CSV файлу, содержащему:
   a. ФИО пользователя,
   b. Должность
   c. Название отдела
   d. E-mail
   e. Телефон
   f. Логин
   g. Пароль
   h. Имя контейнера, в который надо поместить пользователя
   i. Список групп, в которые нужно поместить пользователя
   j. Путь до домашней папки (подключается на диск X:).
   k. Путь до перемещаемого профиля.
2. Скрипт читает файл и создает необходимые объекты.
3. Существование групп и контейнеров необходимо проверять и создавать их в случае отсутствия.
4. Скрипт создает все необходимые каталоги в случае их отсутствия, назначает необходимые права NTFS и включает сетевой доступ
5. Формирует в формате HTML отчет, в котором указано сколько и каких групп, контейнеров и пользователей создано.
6. Все объекты создаются в домене, в котором запущен скрипт.

Часть 6. Восстановление удаленных объектов

1. Включите корзину AD (с помощью PowerShell или Центра администрирования AD).
2. С помощью скрипта из части 5 создайте 5 пользователей в контейнере unit-for-delete.
3. С помощью команд dsquery и dsrm удалите всех пользователей в контейнере unit-for-delete.
4. С помощью PowerShell восстановите всех удаленных пользователей в контейнере unit-for-delete.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы.

Вопросы:

1. Раскройте смысл терминов: дерево доменов, лес и схема Active Directory.
2. Где на контроллере домена хранится данные об объектах Active Directory в виде файлов? Какие файлы за что отвечают?
3. Где на контроллере домена хранятся файлы, содержащие групповые политики домена?
4. Какие компоненты автоматически устанавливаются мастером при добавлении ролей Active Directory?
5. Для чего нужен пароль DSRM?
6. Как восстановить пароль DSRM, если он был утерян после установки?
7. Зачем нужно имя домена NetBIOS?
8. Какие группы пользователей создаются в AD автоматически? Опишите минимум 5 из них.
9. Какие записи в DNS создаются специально для AD? Перечислите их, укажете их назначение.

Артефакты:

1. Приведите скриншоты групповых политик AD из части 4.
2. Приведите скрипт из части 5.
3. Как с помощью Powershell восстановить удаленный объект AD?
4. Приведите конвейер команд из ч.6 п.3
5. Приведите конвейер команд из ч.6 п.4

Работа №3.
Управление контроллерами домена в Active Directory

Цель работы: получить дополнительные навыки по управлению контроллерами домена Active Directory на основе Windows Server, работу с событиями и процессами.

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жёстких дисков операционных систем **Windows Server 2012/2016.**
- Доступ к Microsoft Evaluation Center (https://www.microsoft.com/ru-ru/evalcenter)

Краткие теоретические сведения:

Основной единицей хранения в AD является домен. Домен – контейнерный объект, представляющий собой фрагмент AD, хранящийся на специальном компьютере с Windows Server. Домен может содержать

объекты-контейнеры (Organization Unit) и конечные объекты (User, Group, Computer и т.п.). Домены AD могут объединяться в деревья, деревья в конгломераты более высокого уровня – леса. В AD относительно домена может строиться распределенная система, в которой копии домена хранятся на нескольких Windows Server, работающих в режиме контроллера домена.

Служба Active Directory Directory Service является распределенной. Домен хранится на одном или нескольких контроллерах доменов, которые являются равнозначными.

Однако существуют особые роли контроллеров домена – FSMO и функция глобального каталога. Эту функцию и каждую из ролей выполняет единственный контроллер. FSMO и функция глобального каталога могут быть перенесены или принудительно захвачены [5].

Совет 1. После выполнения работы необходимо сохранить снимки состояния виртуальных машин, для использования в последующих работах.

Совет 2. Перед выполнением работы ознакомитесь с требованиями к содержанию отчета, чтобы собирать необходимые артефакты выполнения.

Порядок выполнения работы:

Часть 1. Подготовительная.

1. Для выполнения работы понадобится две виртуальные машины Windows Server и Windows 10 Pro или Enterprise и Windows Server одной из версий: 2012 R2\2016\2019.
2. Дистрибутивы операционных систем со сроком действия 90 дней можно скачать с сайта Microsoft Evaluation Center (https://www.microsoft.com/ru-ru/evalcenter).
3. Для выполнения работы вы можете использовать готовую инфраструктуру из работы №2.
4. Если таковая инфраструктура имеется, установите дополнительный Windows Server с именем as-srv-2 со статическим адресом и в той же локальной сети, что и предыдущие машины. Введите его в домен.
5. Если инфраструктуры нет, то:
    a. Подготовьте виртуальные машины с windows server ad-srv и as-srv-2, и ad-client с Windows 10.
    b. Настройте виртуальные машины так, чтобы они оказались в одной, изолированной LAN. Для сервера выберите и настройте адрес из сети 10.0.0.0/8. В качестве DNS севера установите адресе сервера ad-srv.
    c. Подготовьте компьютер «AD-Srv» к развертыванию AD (новый домен, новый лес) с установкой DNS на «Ad-srv». С помощью мастера добавления ролей и компонентов и диспетчера серверов разверните домен с именем: «ваши_FIO».local. Автоматически установите и настройте DNS.

d. Введите в домен as-srv-2 и ad-client

6. Сделайте снимки всех машин.

Часть 2. Добавление контроллера домена

1. На компьютере ad-srv-2 установите роль AD DS.
2. Настройте на нем дополнительный контроллер домена в том же лесу, домене.
3. После установки перезагрузите компьютер.

Часть 3. Получение информации о домене

1. С помощью PowerShell установите, на каком контроллере домена функционируют FSMO
2. С помощью dsquery установите, на каком контроллере домена функционируют FSMO
3. Выясните, какие записи DNS появились с вводом нового контроллера домена.
4. На контроллере ad-srv-2 создайте пользователя в AD. Убедитесь, что от имени этого пользователя можно запускать процессы на ad-client с помощью GUI, утилиты runas и PowerShell.

Часть 4. Архивация Active Directory

1. С помощью PowerShell установите службу архивации windows на ad-srv.
2. С помощью консольной утилиты wbadmin создайте архивную копию ActiveDirectory.

Часть 5. Замена контроллера домена

Реализуйте сценарий замены контроллера домена, при котором все роли и gc будут переданы на ad-srv-2, а роль AD DS будет удалена с ad-srv.

1. Создайте снапшоты всех виртуальных машин.
2. Перенесите FSMO и gc на ad-srv-2 или с помощью утилиты ntdsutil, PowerShell или GUI. Убедитесь, что перенесен и DNS.
3. Подготовьте описание процесса для отчета.
4. С помощью утилиты dcdiag проверьте AD на ошибки.
5. С помощью PowerShell установите, на каком контроллере домена функционируют FSMO
6. Удалите роль AD DS на ad-srv. Перегрузите компьютер.
7. С помощью утилиты dcdiag проверьте AD на ошибки.
8. Убедитесь, что пользователи могут регистрироваться на ad-client.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы.

Вопросы:

1. Перечислите FSMO. Кратко раскройте их назначение.

2. Опишите, что произойдет, если не будет доступна каждая из ролей.

3. Как с помощью утилиты dcdiag проверить корректность настройки только dns?

4. Как с помощью утилиты dcdiag исправить ошибки в конфигурации?

5. Как ввести компьютер в домен с помощью утилиты netdom?

6. Как ввести компьютер в домен с помощью утилиты PowerShell?

7. Как запустить процесс от имени другого пользователя с помощью утилиты runas?

8. Как запустить процесс от имени другого пользователя с помощью командлета Invoke-Command?

Артефакты:

1. Консольные выводы по Части 3, п. 1-2.
2. Ответ на вопрос из Части 3., п. 3.
3. Командные строки из Части 4, п. 1-2.
4. Приведите описание процесса переноса ролей из Части 5. п.2
5. Командные строки и консольный вывод из Части 5, п. 5.

Работа №4.
Развертывание в среде Windows Server сетевых инфраструктурных сервисов на примере DHCP

Цель работы: получить представление и практические навыки работы по развертыванию и управлению сетевыми инфраструктурными сервисами на примере DHCP и IPAM в середе Windows Server, освоить основные понятия, связанные с работой DHCP сервера.

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жёстких дисков операционных систем **Windows Server 2012/2016.**

Краткие теоретические сведения:

ОС Windows Server содержит необходимые для работы корпоративной сети сервисы. Все эти серверы могут развертываться как с помощью GUI, так и с помощью PowerShell, а некоторые с помощью консольных команд, таких как netsh. Кроме того, в состав ОС Windows включены консольные команды для управления сервисами (net, sc и др.).

К таким сервисам относится DHCP сервер. Протокол DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки узла) — сетевой протокол, позволяющий узлам в сети автоматически получать необходимые параметры (IP-адрес, маску, адреса шлюза и DNS, имя хоста и другие параметры).

Обеспечивает работу сервера DHCP пара клиент и сервер, каждый из них является системной службой. DHCP-сервер содержит настройки одной или нескольких областей (scope). Для области настраиваются пул (pool) адресов и опции (options). Из пула по запросу клиентов сервер выделяет адреса на определенное время или навсегда. Можно создать предопределённые назначения адресов [6].

Порядок выполнения работы:

Часть 1. Подготовка инфраструктуры

1. Подготовьте инфраструктуру (можете использовать инфраструктуру от предыдущих работ), включающую:
   a. три виртуальные машины – 2 сервера (windows server) и 1 машину – клиента (любая ОС).
   b. Все машины в одном LAN.
   c. На одном из серверов должен быть развернута AD DS.
   d. Все ОС должны быть в одном домене.
2. Для удобства будем далее называть компьютеры так: s1 – первый сервер с AD DS, s2 – второй сервер, c1 - клиент.
3. Сделайте снимки исходного состояния для каждой из машин.

Часть 2. Развёртывание DHCP сервера

1. Запустите машину s1.
2. Назначьте на сетевой интерфейс адрес 10.0.0.1/8. Отключите IPv6.
3. Добавьте роль DHCP-сервер через ServerManager.
4. Запустите консоль DHCP-сервера и сконфигурируйте его так, чтобы:
   - Клиентам выдавали 100 адресов, начиная с 10.0.0.100
   - Из этого диапазона были исключены для назначения адреса 10.0.0.195-10.0.0.200
   - Адреса выдавались на 1 час.
   - Адрес шлюза и DNS – 10.0.0.1.
   - Родительский домен соответствовал названию вашего домена AD DS.
5. Создайте Резервирование для MAC адреса 00-01-02-03-04-05, для которого назначается IP адрес 10.0.0.199.
6. Создайте DHCP-политику (не политику AD DS!), которая

работает аналогично настройкам всей области, но для узлов с MAC адресами, начинающимися на AA-01-02, устанавливает адрес шлюза по умолчанию на 10.10.10.10

7. Сделайте архив конфигурации DHCP сервера в каталог C:\bak-dhcp\

8. С помощью команды netsh (контексты dhcp server) выведите дамп конфигурации. Сохраните его в текстовый файл. **(!)**

Часть 3. Работа клиента DHCP

1. Включите виртуальную машину c1.
2. На сетевом интерфейсе отключите IPv6 и для IPv4 включите получение адресов автоматически.
3. С помощью команды ipconfig определите полученные конфигурации и время аренды. Сохраните консольный вывод в файл. **(!)**
4. Найдите назначение адреса в консоли управления сервером DHCP.
5. На сервере s1 для DHCP сервера и протокола IPv4 отобразите сводную статистику работы сервера. Сохраните скриншот окна.
6. На c1 помощью утилиты ipconfig освободите резерв адреса и запросите адрес заново.
7. В диспетчере устройств, в параметрах сетевой платы задайте MAC адрес 00-01-02-03-04-05. С помощью команды ipconfig определите полученные конфигурации и время аренды. Сохраните консольный вывод в файл. **(!)**
8. В диспетчере устройств, в параметрах сетевой платы задайте MAC адрес AA-01-02-03-04-05. С помощью команды ipconfig определите полученные конфигурации и время аренды. Сохраните консольный вывод в файл. **(!)**
9. На сервере s1 в консоли управления DHCP сервером ознакомитесь с выданными лицензиями на адреса.

Часть 4. Организация отказоустойчивого DHCP сервиса

1. Запустите виртуальную машину s2.
2. Назначьте на сетевой интерфейс адрес 10.0.0.2/8. Отключите IPv6.
3. Установите DHCP сервер, но не настраивайте на нем области.
4. На сервере s1 проведите настройку отработки отказа для созданной области (настройте Failover).
5. Настройте сервер-партнер s2 так, чтобы:
    - Он работал в режиме Горячей замены в ждущем

режиме

- Имел 35% адресов пула для резерва
- Время упреждения клиента составляло 30 минут
- Интервал переключения 1 минуту
- Секретное слово для проверки – «123»

6. Визуально убедитесь в репликации области на s2.
7. На сервере s1 сделайте скриншот окна свойств области, закладка Обработка отказа. Сохраните скриншот. **(!)**
8. На машине c1 с помощью команды ipconfig определите, какие адреса получены и какой DHCP сервер их выдал. Сохраните консольный вывод в файл. **(!)**
9. В свойствах виртуальной машины s1 отключите сетевой кабель. На сервере s2 сделайте скриншот окна свойств области, закладка Обработка отказа. Сохраните скриншот. **(!)**
10. На машине c1 отключите и снова включите сетевой интерфейс. С помощью команды ipconfig определите, какие адреса получены и какой DHCP сервер их выдал. Сохраните консольный вывод в файл. **(!)**
11. В свойствах виртуальной машины s1 подключите сетевой кабель обратно.

Часть 5. Автоматизация управления DHCP сервисом с помощью PowerShell

Исходя из того, что, работают хосты s1, s2, c1 и на хостах s1 и s2 назначены адреса 10.0.0.1\8 и 10.0.0.2\8, написать скрипт, который добавляет роли DHCP-серверов на s1 и s2 и конфигурирует службы согласно п.5 части 2 и п.6 части 4. Параметры конфигурации (адреса, имена, значения времени и др.) следует хранить в текстовом файле. Для проверки скрипта можете восстановить снимки виртуальных машин.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы.

Вопросы:

1. Раскройте смысл понятий в контексте DHCP: область, опция, аренда, политика.
2. Какие компоненты устанавливаются мастером при добавлении роли DHCP-сервер?
3. Какие опции DHCP были задействованы в Части 2?
4. Какие режимы работы с точки зрения обеспечения надежности, существуют для DHCP сервера в Windows Server? Объясните разницу.
5. Поясните параметры Максимальное время упреждения для клиента (Maximum Client Lead Time) и Интервал переключения

состояния (State Switchover Interval). Что они означают? Что произойдет при сбое партнёра, если не задавать Интервал переключения состояния?

6. Что из себя представляет архивная копия DHCP-сервера?
7. Как переименовать хост с помощью PowerShell?

Артефакты:

1. Приведите секцию добавления области из файла дампа конфигурации из п.8, Часть 2.
2. Приведите консольный выводы и скриншоты п. 3, 7, 8, 9 части 3. Дайте пояснения, объясните различия.
3. Приведите консольный выводы и скриншоты п. 7-10 части 4. Дайте пояснения, объясните различия.
4. Приведите PS команду экспорта конфигурации DHCP сервера в файл.
5. Приведите текст скрипта Части 5.

Работа №5.
Работа со средствами мониторинга и диагностики в Windows

Цель работы: ознакомиться со встроенными средствами технического мониторинга, назначением и принципами работы Prefomance Monitor. Получить навыки сбора и анализа данных, позволяющих оценивать производительность системы. Получить практические навыки поиска "узких мест" в производительности системы. Получить дополнительные навыки по управлению Windows Server, управлению процессами и журналами работы.

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жёстких дисков операционных систем **Windows Server 2012/2016.**
- Доступ к Microsoft Evaluation Center (https://www.microsoft.com/ru-ru/evalcenter)

Краткие теоретические сведения

Одной из важнейших составляющих обеспечения функциональности системы является ее мониторинг. Современные ОС содержат средства для осуществления технического мониторинга. К целям технического мониторинга относятся:

1) наблюдение за текущими параметрами системы;
2) сбор статистики для ретроспективного анализа, построения профилей загрузки системы, отладки и настройки приложений, прогностического анализа;

3) автоматизация реакции системы на определенные ее состояния.

В системе ОС Windows содержится компонент Performance Monitor, реализующий технический мониторинг.

Performance Monitor позволяет создавать:

4) Группу Сборщиков Данных – набор единообразно управляемых журналов сбора данных. В ней создавать:

5) Счетчик Производительности - журнал, в который с определенной периодичностью заносятся значения Счетчиков – атрибутов программных объектов, представляющих или аппаратные или программные подсистемы (Процессор, UDP, Файл Подкачки и т.п.).

6) Сборщик Данных отслеживания событий, куда заносятся все события, происходящие в подсистеме, которая выбрана в виде провайдера при создании сборщика.

7) Оповещение счетчика производительности – специального журнала, позволяющего автоматически реагировать на определённое состояние счетчика.

В Performance Monitor содержится раздел Отчетов, через который доступны обработанные сводные данные журналов [6].

Если журнал Счетчик Производительности ведется в бинарном формате, то конвертировать его в текстовый формат позволяет утилита **relog.exe**

Для конвертации журнала Сборщик Данных отслеживания событий используется утилита **tracerpt.exe**

Как любая зрелая операционная система, Windows содержит подсистему видения системных журналов. Доступны системные журналы (Приложения, Безопасность, Система, Установка, Перенаправленные события) и журналы приложений и служб.

Для работы с журналами служат оснастка Просмотр событий (eventvwr.msc), консольная утилита Wevtutil и PowerShell.

Windows содержит механизм запуска процессов по расписанию или событиям - Планировщик Заданий. Для работы с ним служат консоль taskschd.msc, утилиты schtasks.exe и at (устарела), а также PowerShell.

Порядок выполнения работы:

Часть 1. Работа с процессами. Разработка скриптов.

1. Напишите скрипт, который создает Журнал Работы с именем «ProcessMonitoringLog». Если журнал существует, то выводится сообщение об этом.

2. Напишите скрипт на PowerShell, который:

a. При запуске выводит список запущенных процессов (PID, Имя процесса, Путь к исполняемому файлу, Пользователь

процесса, Утилизация CPU, Занимаемая память, Время Получения данных).

b. Записывает эти данные в CSV файл.

c. При успешном сохранении данных пишет в журнал ProcessMonitoringLog сообщение об успехе, при ошибках сохранения – сообщение об ошибке.

Часть 2. Планирование периодического выполнения.

1. С помощью PowerShell добавьте автоматический запуск скрипта из Части 1. п.2 в планировщике заданий Windows (Task Scheduler), так чтобы он запускался каждые 3 минуты даже в тех случаях, когда питание идет не от батареи или ИБП.
2. Убедитесь в работоспособности решения.

Часть 3. Работа с журналом событий.

1. Ознакомитесь с журналом событий.
2. Создайте настраиваемое представление журнала, позволяющее увидеть все неудачные попытки входа в ОС под именем Администратора.
3. С помощью PowerShell напишите скрипт, который выводит в текстовый файл:
   a. время последних 10 включений компьютера,
   b. время 5 последних установок пакетов обновлений с указанием названий обновлений (например, KB1299393),
   c. количество ошибок и количество предупреждений за последние 24 часа.

Часть 4. Сбор и анализ данных

1) создать в программе Performance Monitor Группу Сборщиков Данных, которая будет содержать:
   a. Счетчик Производительности, записи которого позволят сравнить загрузку аппаратного обеспечения платформы. Счетчики для этого следует выбрать самостоятельно, но они должны отражать использование памяти, дисковой подсистемы, процессора и сети.
   b. Периодичность журнала установить в 5 секунд.
   c. Сборщик данных отслеживания событий, фиксирующий события ядра Windows.
2) С помощью Группы Сборщиков Данных сравните загрузку системы в двух разных ситуациях. Это может быть загрузка при использовании разных приложений одного типа (2 антивируса, 2 браузера, 2 СУБД, 2 кодека и т. п.), наборы разных программ (MS Word + MS Excel и MS Excel + MS Access и т.п.) или одно и то же приложение при разной его конфигурации.

3) С помощью механизма отчетов дайте первичный анализ загрузки в обоих случаях.

4) С помощью электронных таблиц или других средств анализа данных представите данные о загрузке в виде графиков.

Часть 5. Автоматизация реакции системы на ее состояние

1) Добавьте в виртуальную машину еще один жесткий диск объемом 200 Мб. Включите виртуальную машину и создайте на новом диске раздел.

2) Создайте скрипит, который постепенно заполняет новый логический диск файлами размером до 1 Мб.

3) Создайте скрипит, очищающий новый диск.

4) В Performance Monitor создайте новую Группу Сборщиков Данных с Оповещением счетчика производительности, который срабатывает в случае, если осталось менее 20% свободного места на новом разделе и выводит предупреждение в журнал событий и запускает скрипт из п.3.

Разработчики Performance Monitor предполагают, что нужно в Планировщике заданий создать задание, выполняющее скрипт из п. 3, и указать имя этого задания в настройках Сборщика данных отслеживания событий.

5) Проверьте срабатывание оповещений. Вероятно, вы обнаружите неожиданное поведение системы. Попробуйте выяснить причину, заменяя используемые счетчики.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы.

Артефакты:

1) Скрипты из Части 1
2) Скрипты из Части 2
3) Параметры Представления из Части 3, п.2
4) Скрипт из части 3. п. 3
5) Материалы и результаты анализа из Части 4 п. 3-4
6) Скрипты из части 5.

Вопросы:

1) В чем назначение каждого из разделов журнала событий?
2) Зачем нужен раздел Перенаправленные события?
3) Где находится журналы событий Windows в виде файлов?

4) Как с помощью графической оснастки журнала событий установить по известному VID коду, когда было подключено и настроено устройство?

5) Почему были выбраны конкретные счетчики в Части 4 п.1? Обоснуйте выбор.

6) Как получить на консоль подробные параметры запланированного задания с помощью утилиты schtasks.exe? Проиллюстрируйте ответ на примере задания из части 5.

7) Опишите ваши выводы по пункту 5 Части 5.

Работа №6.
Работа с томами хранения данных в Windows Server

Цель работы: получить представление и практические навыки работы по настройке томов хранения данных, организации программного RAID и использованию протокола iSCSI.

Необходимо:

- Установленная на компьютере среда виртуализации **ORACLE Virtual Box**
- Образы виртуальных жёстких дисков операционных систем **Windows Server 2012/2016.**

Краткие теоретические сведения:

ОС Windows Server позволяет гибко управлять файловым хранением, создавать программные RAID, использовать SAN и разворачивать отказоустойчивые сетевые файловые службы. Все эти возможности можно реализовать как с помощью GUI, так и с помощью PowerShell [7-9], а некоторые с помощью консольных команд, таких как diskpart. Windows Server поддерживает следующие программные RAID: JBOD, RAID0, RAID1, RAID5. В Windows Server встроены программные компоненты для работы с iSCSI.

Порядок выполнения работы:

Часть 1. Подготовительная.

1. Для выполнения работы понадобится две виртуальные машины Windows Server (могут использоваться машины, созданные в работе №4). Для удобства будем далее называть компьютеры так: s1 – первый сервер с AD DS, s2 – второй сервер, член домена.

2. Сделайте снимки исходного состояния для каждой из машин.

3. Если вы используете готовые машины из работы №4, то сделайте снимки исходного состояния для каждой из машин. Остановите на машинах s1 и s2 DHCP сервера. Настройте виртуальные машины так, чтобы они оказались в одной, изолированной LAN и для каждой из машин был выделен свой IP

адрес из сети 10.0.0.0/8.

4. Проверьте доступность по сети каждой машины с каждой машины с помощью утилиты ping и корректную работу домена AD.

Часть 2. Управление разделами

1. Добавьте в параметрах виртуальной машины s1 4 жестких диска d1-d4 по 4 Гб каждый (для экономии места используйте динамические диски).
2. На диске d1 с помощью Диспетчера дисков создайте простой том с файловой системой NTFS размером 1 Гб и смонтируйте его в каталог Volume1 на диск C:\
3. С помощью PowerShell выведите сведения о подключенных дисках как физических устройствах.
4. С помощью PowerShell выведите сведения о подключенных дисках как логических устройствах.
5. С помощью PowerShell выведите сведения о разделах.
6. С помощью PowerShell выведите сведения о томах.
7. В Диспетчере дисков переведите подключенные диски в offline режим (режим вне сети).
8. Напишите скрипт на Power Shell, который:
    a. Выводит перечень дисков
    b. Запрашивает номер диска
    c. Выводит предупреждение, что все данные на диске будут стерты.
    d. Если пользователь отказывается – завершает работу, если соглашается, то продолжает и
    e. Выполняет необходимые операции, чтобы создать на диске том с файловой системой NTFS и подключить его на букву T:
    f. Проверит диск T: на наличие ошибок, выведет результаты проверки.
    g. Выведет сведения о томе.
9. Удалите все созданные тома и разделы, переведите диски, кроме исходного, в состояние offline (вне сети).

Часть 3. Работа с RAID

1. С помощью Диспетчера дисков или утилиты diskpart создайте на дисках d1, d2, d3 том с RAID5. Подключите его на букву диска W:
2. Создайте на диске W: несколько файлов.
3. Выключите виртуальную машину s1. В свойствах виртуальной машины удалите диск d3.
4. Запустите машину s1.
5. Проверьте доступность файлов на диске W:
6. Через Диспетчер дисков определите состояние RAID5.

Сделайте скриншот, сохраните его в файл.

7. Через Диспетчер дисков восстановите RAID5 с помощью диска d4. Сделайте скриншот, сохраните его в файл.

8. Выключите виртуальную машину s1, подключите диск d3. Включите виртуальную машину и удалите все тома и разделы на дисках d1-d4.

9. Напишите скрипт для утилиты diskpart, который создает том RAID5 на дисках d1-d4 и монтирует его на букву диска V:. Используйте скрипт для создания диска.

Часть 4. Подключение дисков через iSCSI

1. Будет необходимо на виртуальных машинах s1 и s2 реализовать схему, при которой на машине s1 на диске V: нужно хранить виртуальный жесткий диск и монтировать его на машине s2 на букву диска R: по протоколу iSCSI.

2. Дайте определения понятиям роли iSCSI Initiator и iSCSI Target.

3. Определите, какая из машин будет выступать в роли iSCSI Initiator, а какая в роли iSCSI Target. Установите соответствующие компоненты на нужных виртуальных машинах через Диспетчер Серверов или Power Shell.

4. На виртуальной машине s1 создайте виртуальный диск iSCSI с именем LUN1 и объемом 5 Гб.

5. Подключите его к машине s2, указав ее по IP адресу.

6. На машине s2 подключите LUN1 в качестве диска (ключевые слова-подсказки «Обнаружение – Обнаружение портала» «Конечные объекты» или «Discovery – Discovery Portal» и «Targets»).

7. С помощью PowerShell выведите сведения о подключенных дисках как физических устройствах.

8. С помощью PowerShell выведите сведения о подключенных дисках как логических устройствах.

9. С помощью PowerShell выведите сведения о разделах.

10. С помощью PowerShell выведите сведения о томах.

11. С помощью PowerShell выведите только диски, подключенные к машине s2 по iSCSI.

Часть 5. Работа со Storage Spaces

1. Добавьте к конфигурации машины s1 7 дисков по 10 Гб ((для экономии места используйте динамические диски).

2. С помощью графического интерфейса Server Manager создайте пул из всех этих дисков.

3. С помощью PowerShell создайте на этом томе виртуальный диск в режиме Two-way-mirror с максимальным объемом. Командой PowerShell определите его объем.

4. Удалите виртуальный диск.

5. С помощью PowerShell создайте на этом томе виртуальный диск в режиме Parity с максимальным объемом и с отказоустойчивостью по отношению к потере двух дисков. Командой PowerShell определите его объем.

Содержание отчета

Требуется подготовить отчеты в формате DOC\DOCX или PDF. Отчет содержит титульный лист, артефакты выполнения и ответы на вопросы.

Вопросы:

1. В чем разница возможностей динамических и базовых дисков в Windows?
2. В чем разница устройства динамических и базовых дисков в Windows?
3. Сопоставьте данные, полученные в части 2, п. 3-6. Соотнесите объекты, с которыми вы работали, с элементами стека хранения Windows. Опишите результаты сопоставления и сравнения.
4. Каков будет размер каталога Volume1 после выполнения п. 2 части 2? Почему?
5. Какой будет объем диска W: после выполнения п.1 части 3 и диска V: после п.9 части 3? Почему?
6. Дайте определения понятиям iSCSI Initiator, iSCSI Target и IQN (iSCSI qualified name).
7. Как с помощью PowerShell установить iSCSI Target на локальный хост?
8. При создании виртуального диска iSCSI возможно выбрать три типа диска: Фиксированный, Динамический и Разностный. В чем разница этих типов? Придумайте реальные ситуации, когда целесообразно применять каждый из трех типов дисков.
9. Сравните данные, полученные в части 2, п. 3-6 с данными, полученными в части 4, п. 7-10.
10. Чем отличается, по-вашему, программный RAID и Storage Spaces? Какие аналогии из мира Linux вы можете привести?
11. Как вывести информацию о пуле Storage Spaces с помощью PowerShell?
12. Как создать пул Storage Spaces с помощью PowerShell?
13. Сравните объемы дисков из части 5 п.3 и п.5. Какой диск больше? Почему?

Артефакты:

1. Напишите конвейер PowerShell, который в гостевой Windows Server выводит информацию только о тех дисках, которые были подключены в части 3, п. 1.
2. Скрипт из части 2, п.8.
3. Скриншоты окон из части 3 п. 6 и 7.
4. Приведите команду из части 4, п. 11.
5. Команды из Части 5, п.3,4,5.

Приложения

Особенности командной оболочки PowerShell

PowerShell — это оболочка командной строки и язык сценариев, разработанный Microsoft. Она имеет множество особенностей, отличающих ее от традиционных командных оболочек, таких как `cmd` для Windows или `bash` для Linux. Перечислим основные отличия.

1) PowerShell предоставляет интерфейс командной строки (CLI), который позволяет пользователям взаимодействовать с компьютерной системой, вводя команды.
2) PowerShell является языком сценариев, который позволяет пользователям автоматизировать задачи, создавая и запуская сценарии.
3) PowerShell - это объектно-ориентированная оболочка, которая использует объекты для манипуляций с данными. Она может передавать объекты между командлетами, что позволяет выполнять более сложную и эффективную обработку данных.
4) PowerShell является кроссплатформенной и может работать в Windows, macOS и Linux.
5) PowerShell позволяет пользователям взаимодействовать с системой в режиме реального времени, что упрощает отладку и устранение неполадок.
6) PowerShell обладает высокой расширяемостью, с большой библиотекой командлетов (команд) и модулей, которые можно использовать для расширения его функциональности.
7) PowerShell — это безопасная оболочка, которая предоставляет несколько функций безопасности, включая политики подписи и выполнения скриптов, для защиты от вредоносного кода.
8) PowerShell интегрирован с другими технологиями Microsoft, такими как Active Directory и Exchange, что упрощает управление этими технологиями и их автоматизацию.
9) PowerShell можно настроить в соответствии с конкретными потребностями пользователей и организаций с помощью файлов конфигурации, профилей и других параметров настройки.
10) PowerShell обратно совместим с предыдущими версиями PowerShell, гарантируя, что сценарии и инструменты, разработанные в более ранних версиях, будут продолжать работать в последней версии.

Различия PowerShell .NET и .Core

PowerShell доступна в двух разных версиях: PowerShell .NET и, более новой, PowerShell .NET Core (просто .Core). Рассмотрим их основные различия.

1) Совместимость: PowerShell .NET совместим только с операционными системами Windows, а PowerShell .NET Core предназначен для работы в Windows, macOS и Linux.
2) Зависимости: PowerShell .NET использует полную среду выполнения .NET Framework, а PowerShell .NET Core использует среду выполнения .NET Core. Это означает, что PowerShell .NET Core имеет меньше зависимостей и может работать на большем количестве систем без дополнительной установки полной версии .NET Framework.
3) Производительность: PowerShell .NET Core, как правило, быстрее, чем PowerShell .NET, поскольку он оптимизирован для кроссплатформенного использования и имеет меньший объем исполняемых модулей.
4) Поддержка модулей: PowerShell .NET Core имеет меньшее количество встроенных модулей, чем PowerShell .NET, но все же может загружать и использовать модули, разработанные для PowerShell .NET.
5) Интеграция с облаком: PowerShell .NET Core лучше интегрируется с облачными службами, такими как Azure, чем PowerShell .NET, что делает его лучшим выбором для облачных задач автоматизации и управления.
6) Новые функции: PowerShell .NET Core имеет некоторые новые функции, недоступные в PowerShell .NET, такие как возможность запуска сценариев PowerShell в контейнерах Docker и возможность упаковывать сценарии PowerShell в виде автономных исполняемых файлов.

Таким образом, PowerShell .NET оптимизирован для использования в системах Windows и имеет большее количество встроенных модулей, в то время как PowerShell .NET Core разработан для работы в кроссплатформенных средах с возможностью прозрачной интеграции с облачными решениями.

Основные сведения о синтаксисе языка сценариев PowerShell

Кратко рассмотрим основные сущности, языковые конструкции и приемы работы с PowerShell [8].

1) Предусмотрены следующие расширения для файлов PowerShell:

.ps1 - файлы скриптов,
.psd1 - файлы данных скриптов,
.psm1 - файлы модулей скриптов,
.ps1xml - файлы конфигурации.

2) Основным строительным элементом скрипта является к**омандлет** (*cmdlet*) – это предопределенная или разработанная пользователем команда PowerShell, с помощью которой можно осуществлять взаимодействие с окружением и объектами ОС. Имя командлета строится по шаблону:

«Глагол-Существительное»

Используются глаголы: Add – добавление данных, Clear – очистить; Enable – включить; Disable – выключить; New – создать; Remove – удалить; Set – задать; Start — запустить; Stop – остановить; Export – экспортировать; Import – импортировать [9].

3) Для получения справки можно использовать следующие командлеты:

`Get-Command` – поиск командлета и команд по шаблону имени,

`Get-Help` – справка по конкретной команде,

`Get-Member` – получение сведений об объектах PowerShell, их полях, типах полей и методов.

4) Командлеты взаимодействуют через переменные или конвейеры. Конвейер – это передача результата работы командлета через вертикальную черту (|) другому командлету. Стандартным результатом работы командлета является объект, так что в большинстве случаев передается объект. Например:

```
Get-ChildItem -Path "C:\Windows" -File | Sort-Object length -Descending | Select-Object -First 1
```

5) Выполнение скриптов, возможность их запуска в фоновом режиме, доступ к объектам ОС ограничены политиками. Существуют следующие политики выполнения скриптов [7]:

- Restricted – блокируется выполнение любых сценариев (значение по умолчанию);
- AllSigned – разрешено выполнение сценариев, которые имеют цифровую подпись;
- RemoteSigned – разрешено выполнение локальных сценариев, все скачанные сценарии должны иметь цифровую подпись;
- Unrestricted — разрешено выполнение любых сценариев (не рекомендуется, так как небезопасно!).

Для работы с политиками служат командлеты `Get-ExecutionPolicy` и `Set-ExecutionPolicy`.

Установить политику можно, например так:

```
Set-ExecutionPolicy RemoteSigned
```

6) Скрипты PowerShell можно запускать на удаленных хостах через службу WinRM, передающую данные через HTTP\HTTPS по порту tcp\5985. Выполнение команды на удаленном хосте возможно при указании имени хоста через атрибут `ComputerName` (если командлет его поддерживает). Этот вариант работает только для одной команды.

Можно установить сессии для выполнения в режиме «1-to-1» или «1-to-many».

Сессия «1-to-1» запускается командлетом `Enter-PSSession` (интерактивный сеанс), например:

```
Enter-PSSession -ComputerName SRV-01
```

При этом нельзя сделать последующее переподключение, использовать команды, имеющие графический интерфейс, запускать команды, имеющие свой собственный шел (например, nslookup, netsh), и взаимодействовать с пользователем на удаленной машине. Скрипты можно запускать, если политика запуска на удаленной машине позволяет это сделать.

Сессия «1-to-many» запускается командлетом `Invoke-Command`. Например:

```
Invoke-Command -Command { dir } -ComputerName
                SRV-01, SRV-02
```

Через параметр -ScriptBlock можно передать набор команд, а через FilePath – путь к скрипту на вашем компьютере.

7) В PowerShell есть возможность фонового исполнения заданий. Используются следующие командлеты [7]:

- `Start-Job` – запустить фоновую задачу;
- `Stop-Job` – остановить фоновую задачу
- `Get-Job` – посмотреть список фоновых задач;
- `Receive-Job` – посмотреть результат выполнения фоновой задачи;
- `Remove-Job` – удалить фоновую задачу;
- `Wait-Job` – перевести фоновую задачу на передний план, для того чтобы дожидаться ее окончания.

Например:

```
Start-Job {Get-Service}
```

8) Комментарии в PowerShell бывают

```
# строчные
<#
и
блочные
#>
```

9) Обращение к переменным осуществляется через знак «$»: `$имя_переменной`. Также можно использовать командлеты:

- `New-Variable` - создание переменной;
- `Set-Variable` - присваивание переменной значения;
- `Get-Variable` - получение переменной;
- `Clear-Variable` - очистка переменной;
- `Remove-Variable` - удаление переменой.

Если командлет через конвейер возвращает массив объектов или используется цикл по массиву, то к текущему элементу можно обращаться через «`$_`». Приведем пример:

```
Get-Service | WHERE {$_.status -eq "Running"} | SELECT displayname | sort-object displayname
```

У переменных есть область видимости, но можно объявить глобальную переменную, например так:

```
$Global: MyVar = “My Value”
```

10) Объявлять переменные можно с указанием типа или без этого. Переменные можно сразу инициализировать. Переменные могут менять свой тип, но только в том случае, если он не указан при объявлении. Перечислим основные типы:

- [int] — целое число, 32 бита;
- [single] — число с плавающей запятой одинарной точности;
- [double] — число с плавающей запятой двойной точности;
- [char] — один символ;
- [Boolean] — значения «Истина» или «Ложь»;
- [datetime] — дата или время;
- [string] — строка символов.

11) В PowerShell существуют массивы и хэштаблицы. Массивы нумеруются с 0.

Массивы объявляются так:

```
$Array = 1, 3, "Пример", 7, 9
#Выводим 3 элемент массива
```

```
$Array[2]
```

Хеш-таблицы строятся по принципу:

```
@{ ключ = «значение» }
```

Приведем примеры:

```
$ht = @{w1="I"; w2="LOVE"; w3="POWERSHELL"}
$ht
$ht.Add("w4","REALLY")
$ht
```

12) В PowerShell существуют все общепринятые условные операторы:

```
IF
IF…ELSE
IF…ELSEIF…ELSE
SWITCH
```

А вот для сравнения используются не обычные операторы:

- -eq – равно (знак =);
- -ne – не равно (эквивалентно знакам <> или !=);
- -gt – больше (знак >);
- -lt – меньше (знак <);
- -ge – больше или равно (эквивалентно знакам >=);
- -le – меньше или равно (эквивалентно знакам <=).

Приведем пример:

```
[int]$TestVar = 150
If ($TestVar -eq 100){
Write-Host "Переменная TestVar = 100"
}
ELSEIF ($TestVar -gt 100){
Write-Host "Переменная TestVar > 100"
}
ELSE {
Write-Host "Переменная TestVar < 100"
}
[int]$TestVar = 2

SWITCH ($TestVar)
{
0 {Write-Host "Переменная TestVar = 0"}
1 {Write-Host "Переменная TestVar = 1"}
2 {Write-Host "Переменная TestVar = 2"}
default {Write-Host "Неопределенное значение"}
}
```

13) Существуют дополнительные операторы сравнения PowerShell [8]:

- -like - символ подстановки

```
“PowerShell” -like “PowerS*" #true
```

- -notlike - не символ подстановки

```
“PowerShell” -nolike “PowerS*" #false
```

- -contains - содержит ли значение слева значение справа

```
1, 2, 3, 4, 5 -contains 3 #$true
```

- -notcontains - если значение слева не содержит значение справа, получим истину

```
1, 2, 3, 4, 5 -notcontains 3 #$false
```

- -match Использование регулярных выражений

```
$str = “http://ifmo.ru"; $str -match
"^http://(\S+)+(.ru)$" #$true
```

- -notmatch Использование регулярных выражений

```
$str = "http://ifmo.ru"; $str -notmatch
"^http://(\S+)+(.com)$" #true
```

14) В PowerShell существуют логические операторы:

- -and - логическое и
- -or - логическое или
- -not - логическое не

15) Есть привычные циклы:

- WHILE
- DO WHILE
- DO UNTIL
- FOR
- FOREACH

Приведем пример:

```
[int]$TestVar = 1
 DO {
 Write-Host $TestVar
 $TestVar = $TestVar + 1
 }
 UNTIL ($TestVar -gt 10)

$PSService = Get-Service
 FOREACH ($Service In $PSService){
 $Service.Name + " - " + $Service.Status
```

```
}
```

16) Обработка ошибок в PowerShell реализована через механизм `Try…Catch`. Здесь потенциально опасный код помещается в блок Try, а в блок Catch – код, выполняемый при возникновении ошибки.
Например:

```
Try {
[int]$Number = Read-Host "Введите число"
10 / $Number
} catch {
Write-Warning "Некорректное число"
}
```

17) Работа с вводом-выводом осуществляется с помощью командлетов:

- `Write-Host` – вывод на экран,
- `Read-Host` – чтение с консоли,
- `Out-File` - запись в файл (аналогично >),
- `Export-Csv` - экспорт данных в .csv файл,
- `ConvertTo-Html` - запись в файл HTML,
- `ConvertTo-Json` – запись в JSON файл.

18) Наконец, в PowerShell можно описать и вызвать функцию:

```
# часть описания функции
function Get-DayLog ($param1,$param2) {
Get-EventLog -LogName Application -Newest $param1
….
}
#вызов функции, вариант 1
Get-DayLog -param1 50 -param2 -14
#вызов функции, вариант 2
function Get-DayLog ($param1=50,$param2) {….}
return $InternalVariable
```

Основные сущности и понятия Microsoft Active Directory

1) Active Directory (AD) — это технология Microsoft, которая обеспечивает централизованное расположение для управления пользователями, компьютерами и другими сетевыми ресурсами. В контексте операционной системы AD – набор сервисов, работающих на Windows Server.
2) Домен — это логическая группа ресурсов в Active Directory. Он может

включать пользователей, компьютеры, принтеры и другие сетевые ресурсы. Домены обычно основаны на структуре организации и используются для упрощения управления ресурсами и обеспечения безопасности.

3) Лес — это набор из одного или нескольких доменов, имеющих общую схему, конфигурацию и глобальный каталог. Лес — это граница безопасности, а это означает, что политики и параметры безопасности применяются ко всем доменам в лесу.
4) Организационная единица (OU) — это контейнер внутри домена, который можно использовать для группировки ресурсов и применения групповых политик. Подразделения позволяют делегировать административные задачи и более эффективно управлять ресурсами.
5) Контроллер домена (DC) — это сервер, на котором работает Active Directory и который выполняет аутентификацию пользователей и компьютеров. Контроллеры домена хранят базу данных Active Directory, которая содержит информацию обо всех ресурсах в домене.
6) Глобальный каталог (GC) — это распределенная база данных, содержащая частичную копию всех объектов в лесу. Глобальный каталог используется для ускорения поиска и предоставления пользователям возможности искать ресурсы по всему лесу.
7) Доверительные отношения — это соединение между двумя доменами, которое позволяет пользователям одного домена получать доступ к ресурсам другого домена. Доверительные отношения могут быть односторонними или двусторонними, и они могут устанавливаться между доменами в одном и том же лесу или между доменами в разных лесах [10].
8) Групповая политика — это функция Active Directory, позволяющая администраторам управлять параметрами пользователей и компьютеров в сети. Параметры групповой политики можно использовать для управления безопасностью, параметрами рабочего стола, установкой программного обеспечения и другими параметрами конфигурации.

Некоторые особенности лицензирования Windows Server

Windows Server доступен в нескольких редакциях, включая Standard, Datacenter, Essentials и Hyper-V. Каждая редакция имеет разные функции и требования к лицензированию, поэтому важно выбрать ту редакцию, которая соответствует вашим потребностям.

Windows Server можно лицензировать по модели «на ядро процессора» или по модели «на сервер». Для лицензирования по количеству

ядер требуется лицензия для каждого физического или виртуального ядра на сервере, а для лицензирования по количеству серверов требуется лицензия для каждого сервера, на котором работает Windows Server.

Лицензии продаются как бессрочные лицензии или лицензии по подписке, в зависимости от выпуска и модели лицензирования. Активацию лицензий можно выполнить с помощью средства управления многопользовательской активацией (VAMT), центра обслуживания корпоративных лицензий (VLSC) или функции автоматической активации виртуальных машин (AVMA) для виртуальных машин.

Отличия операционных систем Windows Server и Linux

1) Одним из основных различий между Linux и Windows является их базовая архитектура. Linux основан на операционной системе Unix, известной своей модульной конструкцией с открытым исходным кодом. Windows, с другой стороны, построена на ядре Windows NT, которое представляет собой проприетарную операционную систему с закрытым исходным кодом [2].
2) Linux использует монолитную структуру ядра, что означает, что все основные функции операционной системы выполняются одним ядром. Windows, с другой стороны, использует гибридную конструкцию ядра, которая сочетает в себе элементы монолитного ядра с микроядерной архитектурой.
3) Linux обычно использует файловую систему ext4, которая разработана, чтобы быть быстрой, надежной и гибкой. Windows использует файловую систему NTFS, которая предлагает расширенные функции, такие как шифрование и сжатие файлов. Существуют более современные файловые системы: xfs для Linux и ReFS для Windows.
4) Для Linux существует множество интерфейсов командной строки, таких как bash, zsh и др. Для Windows основным средством командной строки является PowerShell.
5) GUI интегрирован в архитектуру Windows, тогда как в Linux он является внешним сервисом по отношению к ОС.
6) Linux — это операционная система с открытым исходным кодом, что означает, что исходный код находится в свободном доступе для любого использования и изменения. Windows, с другой стороны, является операционной системой с закрытым исходным кодом, что означает, что исходный код недоступен для общественности [3].

Список литературы

1. Власов Ю.В. Администрирование сетей на платформе MS Windows Server / Ю.В. Власов, Т.И. Рицкова. - Москва: Национальный Открытый Университет ИНТУИТ, 2016. - 622 с. - ISBN 978-5-94774-858-1. - URL: https://ibooks.ru/bookshelf/362758/reading (дата обращения: 02.04.2025).
2. Современный PowerShell / Андрей Попов. - 2-е изд. - Санкт-Петербург: БХВ-Петербург, 2024. - 416 с.
3. Станек, У. Р. Microsoft Windows Server® 2012 R2: хранение, безопасность, сетевые компоненты. Справочник администратора / У. Р. Станек. — Москва: Русская редакция, 2015. — 416 с. — ISBN 978-5-7502-0436-6. — http://ibooks.ru/reading.php?short=1&isbn=978-5-7502-0436-6 (дата обращения: 02.04.2025).
4. Нортроп, Тони. Проектирование сетевой инфраструктуры Windows Server 2008. Учебный курс Microsoft: офиц. пособие для самоподготовки: [пер. с англ.] / Т. Нортроп, Дж. К. Макин. — М.: Русская Редакция, 2009. — 570 с.
5. Карвальо, Л. Windows Server 2012 Hyper-V. Книга рецептов: / Карвальо Л. — Москва: ДМК Пресс, 2013. — ISBN 978-5-94074-905-9. — http://e.lanbook.com/books/element.php?pl1_id=58692 (дата обращения: 02.04.2025).
6. Платунова, С. М. Администрирование сети Winsows Server 2012: учебное пособие / С. М. Платунова. — Санкт-Петербург: НИУ ИТМО, 2015. — 102 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: https://e.lanbook.com/book/91548 (дата обращения: 02.04.2025). — Режим доступа: для авториз. пользователей.
7. Марк Руссинович, Дэвид Соломон, Алекс Ионеску, Павел Йосифович, Внутреннее устройство Windows – СПб, Питер, 2018 г. – 944 с. ISBN:978-5-4461-0663-9, 9780735684188.
8. PowerShell for Beginners: Learn PowerShell 7 Through Hands-On Mini Games / Ian Waters. - Apress, 2021.; ISBN 978-1-4842-7064-6; 978-1-4842-7063-9
9. Бетрам Адам. PowerShell для сисадминов. Пер. с англ. С. Черников. — СПб: Питер, 2021. — 416 с.: ил. — ISBN 978-5-4461-1732-1.
10. Романов, Н. О. Службы activedirectory в локальных вычислительных сетях / Н. О. Романов, Н. Н. Краснов, О. Л. Чернышев // Образование в XXI веке: проблемы и перспективы: сборник статей XV Международной научно-практической конференции, Пенза, 23–24 ноября 2023 года. – Пенза: Приволжский Дом знаний, 2023. – С. 120-126. – EDN BZGVDN.

Береснев Артем Дмитриевич
Ромакина Оксана Михайловна

Практические работы по системному администрированию в WINDOWS SERVER

Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н. Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49, литер А