Power_shell

Что такое PowerShell?

PowerShell — это кроссплатформенное решение для автоматизации задач, которое включает оболочку командной строки, скриптовый язык и платформу управления конфигурацией. PowerShell поддерживается в Windows, Linux и macOS.

Оболочка командной строки

PowerShell — это современная командная оболочка, в которой реализованы лучшие возможности других популярных оболочек. В отличие от большинства оболочек, которые только принимают и возвращают текст, PowerShell принимает и возвращает объекты .NET. Это решение предлагает следующие возможности:

- надежный журнал командной строки;
- заполнение нажатием клавиши TAB и подстановка команд (см. раздел about_PSReadLine);
- поддержка псевдонимов команд и параметров;
- создание конвейера для объединения команд;
- система справки в консоли, похожая на страницы `man` в Unix.

Язык сценария

В качестве скриптового языка PowerShell обычно используется для автоматизации процессов управления системами. Это решение также часто используется для создания, тестирования и развертывания решений в средах CI/CD. В основе PowerShell лежит среда CLR .NET. Все входные и выходные данные являются объектами .NET. Вам не нужно анализировать текстовые выходные данные для извлечения информации из них. Скриптовый язык PowerShell предлагает следующие возможности:

- расширяемость с использованием функций, классов, скриптов и модулей;
- расширяемая система форматирования для удобного вывода;

- расширяемая система типов для создания динамических типов;
- встроенная поддержка распространенных форматов данных, таких как CSV, JSON и XML.

Платформа автоматизации

Расширяемый характер PowerShell позволил создать экосистему модулей PowerShell для развертывания и администрирования практически любой технологии, с которой вы работаете. Пример:

Microsoft

- Azure
- Windows
- Exchange
- SQL

Сторонний производитель

- AWS
- VMWare
- Google Cloud

Управление конфигурацией

PowerShell Desired State Configuration (DSC) — это платформа управления в PowerShell, которая позволяет управлять корпоративной инфраструктурой, используя конфигурацию как код. С помощью DSC можно выполнять следующие задачи:

- создавать декларативные конфигурации и пользовательские скрипты для повторяемых развертываний;
- применять параметры конфигурации и настраивать информирование о смещении конфигурации;
- развертывать конфигурации с помощью моделей принудительной отправки или опроса.

Чем PowerShell отличается от командной строки

Как упоминалось ранее, cmd — это очень старый инструмент, который никогда не предназначался для удаленного администрирования системы. Для расширения его функциональности требуются дополнительные утилиты, такие как Microsoft Sysinternals PsExec.

PowerShell — это вполне себе законченная среда для написания и исполнения скрипта. Так что можно создавать очень сложные и объёмные скрипты для управления системой, чем те, на какие была способна консоль cmd.

С другой стороны, PowerShell предоставляет множество командлетов для упрощения задач системного администрирования. Он поддерживает автоматизацию широкого круга задач, таких как администрирование Active Directory , управление пользователями и разрешениями, а также извлечение данных о конфигурациях безопасности. Более того, PowerShell теперь поддерживает Linux.

В следующей таблице приведены основные различия между командной строкой и PowerShell с точки зрения программирования и эксплуатации.

Командная строка Windows (или сокращенно cmd) — это командная оболочка, основанная на операционной системе MS-DOS 1980 года. Он присутствует в ОС Microsoft, начиная с Windows NT. Это просто приложение Win32, которое помогает пользователям взаимодействовать с операционной системой с помощью текстовых инструкций (команд) и параметров.

Пользователи полагаются на возможности CMD для взаимодействия со всеми другими объектами и приложениями Win32. Это помогает выполнять различные задачи в Windows, например:

- Запускайте различные приложения и утилиты.
- Активировать / деактивировать важные настройки Windows.
- Создавать или изменять скрипты для автоматизации задач.
- Диагностировать и устранять различные проблемы с ОС, например, сканирование на наличие ошибок.

Хотя интерфейс командной строки широко используется для выполнения вышеуказанных и других задач, он имеет определенные ограничения.

- Он не может получить доступ к большинству элементов системного администрирования Windows.
- Среди других недостатков он не идеален для создания сложных скриптов.

Windows PowerShell — это и командная оболочка, и язык сценариев, обеспечивающий более глубокую интеграцию с ОС Windows. Он основан на платформе Microsoft .Net, что обеспечивает доступ к различным уже существующим инструментам и функциям. Вы можете легко создавать лучшие команды и сложные скрипты с помощью PowerShell.

Он использует текстовые команды, известные как командлеты и язык программирования C #, помогая пользователю более эффективно управлять инфраструктурой Windows. Проще говоря, PowerShell сочетает в себе функциональность командной строки с мощной средой сценариев для упрощения администрирования системы.

Помимо выполнения задач, аналогичных командной строке, PowerShell позволяет пользователям и системным администраторам выполнять следующие действия.

- Удаленный доступ к файловой системе, реестру и WMI (инструментарию управления Windows).
- Создавайте сложные сценарии с множеством условий.
- Удаленное выполнение задач и автоматизация с помощью многоразовых скриптов.
- Командный трубопровод среди прочего.

Основные возможности PowerShell

Windows PowerShell — это в первую очередь командная оболочка с языком сценариев, изначально созданная на основе платформы .NET Framework, а позднее — на .NET Core. В отличие от принимающих и возвращающих текстовые данные оболочек, Windows PowerShell работает с классами .NET, у которых есть свойства и методы. PowerShell позволяет выполнять обычные команды, а также дает доступ к объектам COM, WMI и ADSI. В ней используются различные хранилища, вроде файловой системы или реестра Windows, для доступа к которым созданы т.н. поставщики (providers). Стоит

отметить возможность встраивания исполняемых компонентов PowerShell в другие приложения для реализации различных операций, в т.ч. через графический интерфейс. Верно и обратное: многие приложения для Windows предоставляют доступ к своим интерфейсам управления через PowerShell.

Windows PowerShell позволяет:

- Менять настройки операционной системы;
- Управлять службами и процессами;
- Настраивать роли и компоненты сервера;
- Устанавливать программное обеспечение;
- Управлять установленным ПО через специальные интерфейсы;
- Встраивать исполняемые компоненты в сторонние программы;
- Создавать сценарии для автоматизации задач администрирования;
- Работать с файловой системой, реестром Windows, хранилищем сертификатов и т.д.

Журналы PowerShell и какая информация в них записывается

- **Журнал безопасности (Security Log)** содержит события, которые генерируются в том случае, когда на компьютере настроен аудит. Записи о событиях в журнале безопасности делятся на два типа:
  • успехи (Success events) указывают на то, что действие, для которого был настроен аудит, завершилось успешно (например, пользователь 276 успешно зарегистрировался в сети или успешно получил доступ к файлу на общедоступном ресурсе);
  • отказы (Failure events) протоколируют, что попытка выполнить действие, для которого был настроен аудит, завершилась безуспешно (например, пользователь попытался зарегистрироваться, однако ввел неправильный пароль, попытался обратиться к сетевому ресурсу, не имея соответствующих разрешений на доступ, и т. п.).
- **Журнал системы (System Log)** хранит события, которые были сгенерированы в результате действий операционной системы (например, запуск
  служб, сбои в работе драйверов, изменения роли сервера с рядового члена

до контроллера домена и т. д.). Записи о событиях системы делятся на три типа:
• уведомления (Information events) просто описывают произведенные действия (например, успешный запуск самой службы Event Log, установление удаленного подключения и т. п.). Некоторые уведомления также содержат информацию о сбоях при выполнении действий, которые реально не влияют на сетевые операции;
• предупреждения (Warning events) содержат информацию о событиях, которые могут стать источником различных проблем (например, сбой динамической регистрации DNS-имен из-за неправильной настройки клиента DNS, сбой службы Windows Time Service при поиске контроллера домена, нехватка пространства на диске и т. п.). Реагировать на подобные предупреждения следует как можно более оперативно;
• ошибки (Error events) сохраняют информацию о критических событиях, которые могут привести к потерям данных или другим серьезным проблемам (сбой при инициализации рабочей станции, отказ в динамическом обновлении со стороны сервера DNS, сбой драйвера устройства
и т. п.).

- **Журнал приложений (Application Log)** предназначен для записи событий, сгенерированных запущенными на компьютере приложениями. Для генерации подобных событий разработчики должны вставлять определенный код в свои приложения. Как правило, события приложений оказываются полезными лишь в том случае, когда вы отсылаете информацию
об этих событиях разработчикам для решения возникающих проблем. Тем не менее, в журнале приложений также сохраняются некоторые системные события Windows, например, события, возникающие в результате сбоя приложений (они записываются программой Dr. Watson), события, связанные с групповой политикой, нарушения ограничений криптографического экспорта для протокола IPSec, действия IIS, связанные с работой

В зависимости от того, какие дополнительные компоненты Windows установлены на компьютере, в системе кроме трех основных журналов могут вестись дополнительные журналы событий:

- **журнал службы каталога** (Directory service log), в который записывается информация о действиях Active Directory. Располагается журнал на

контроллерах домена Windows;

- **журнал сервера DNS** (DNS server log), где сохраняется информация о действиях сервера DNS;
- **журнал службы репликации файлов** (File Replication Service log), в котором сохраняются данные о действиях службы File Replication Service (FRS) на тех машинах, где настроена служба DFS (Distributed File System, распределенная файловая система).
- **Установка**. Журнал установки содержит события, связанные с установкой приложений.

  Кроме этого, собственный журнал событий ведет и сама оболочка Windows PowerShell. В этом журнале регистрируются запуск и остановка интерпретатора PowerShell, а также некоторые ошибки, приводящие к завершению
  сеанса работы оболочки.