УДК 004.78:004.43

АВТОМАТИЗАЦИЯ РУТИННЫХ АДМИНИСТРАТИВНЫХ ЗАДАЧ В IT-ИНФРАСТРУКТУРЕ НА ОСНОВЕ СЦЕНАРИЕВ POWERSHELL

Гаах Т.В.

старший преподаватель,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Каримов Д.Р.

магистрант,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Аннотация

Статья посвящена вопросам повышения эффективности управления IT-инфраструктурой путем автоматизации повторяющихся задач на основе PowerShell. Рассматриваются типовые операции (администрирование AD, установка обновлений, мониторинг и бэкап), анализируется экосистема готовых модулей. Представлены практические сценарии для локальной Windows-среды, включая аудит конфигураций, сетевые проверки и управление пользователями. Показано, что применение предложенных скриптов сокращает время выполнения рутины на 70–90% и позволяет высвободить ресурсы для стратегических задач. Обоснована роль PowerShell как ключевого элемента в парадигме Infrastructure as Code.

Ключевые слова: автоматизация, PowerShell, IT-инфраструктура, управление конфигурациями, администрирование, скрипты, информационная безопасность, мониторинг.

AUTOMATION OF ROUTINE ADMINISTRATIVE TASKS IN IT-INFRASTRUCTURE BASED ON POWERSHELL SCRIPTS

Gaakh T.V.
senior lecturer,
Kaluga State University named after K.E. Tsiolkovsky,
Kaluga, Russia
Karimov D.R.
master's student,
Kaluga State University named after K.E. Tsiolkovsky,
Kaluga, Russia

Abstract

This article explores ways to improve IT infrastructure management efficiency by automating repetitive tasks using PowerShell. Typical operations (AD administration, update installation, monitoring, and backup) are covered, and the ecosystem of ready-made modules is analyzed. Practical scenarios for a local Windows environment are presented, including configuration auditing, network scans, and user management. It is shown that using the proposed scripts reduces the time spent on routine tasks by 70–90%, freeing up resources for strategic tasks. PowerShell's role as a key element in the Infrastructure as Code paradigm is substantiated.

Keywords: automation, PowerShell, IT-infrastructure, configuration management, administration, scripts, information security, monitoring.

Современная IT-инфраструктура характеризуется высоким уровнем сложности, масштабируемости и динамичности. В повседневной практике администраторам приходится выполнять множество однотипных операций, отличающихся значительной трудоёмкостью: развёртывание и

конфигурирование серверов, управление учётными записями и правами доступа, непрерывный мониторинг состояния аппаратных и программных компонентов, установка обновлений, резервное копирование и восстановление данных. Указанные процессы предъявляют жёсткие требования к точности исполнения, оперативности и минимизации ошибок, что критично для обеспечения отказоустойчивости ключевых сервисов и соответствия нормативным требованиям, включая внутренние политики информационной безопасности. В данном контексте автоматизация административных операций перестаёт быть вспомогательным инструментом и становится стратегической необходимостью, позволяющей повысить надёжность инфраструктуры, ускорить внедрение изменений и сократить операционные риски [6].

IT-инфраструктура современной организации объединяет физические серверы, рабочие станции, сетевое оборудование, системы хранения данных и облачные сервисы, формируя гибридные или полностью облачные конфигурации. Административные задачи в такой среде носят повторяющийся характер и требуют систематического вмешательства специалистов для поддержания работоспособности и защищённости. Автоматизация позволяет снизить долю рутинных операций, уменьшить вероятность человеческих ошибок и масштабировать управленческие процессы при возрастающих нагрузках [10].

1. Типовые административные задачи, подлежащие автоматизации. К числу наиболее частотных и ресурсоёмких операций относятся:

- Управление учётными записями и доступом – создание, модификация и удаление пользователей в каталогах (например, Active Directory), назначение ролей, групп и политик доступа [5]. Автоматизация этих процедур требует корректной идентификации субъектов, строгой обработки исключений и детального логирования.

- Развёртывание и обновление программного обеспечения – установка приложений на множестве узлов, применение патчей безопасности и обновлений операционных систем (в частности, Windows Server). Ключевыми принципами здесь выступают идемпотентность, проверка зависимостей и возможность отката при сбоях.

- Мониторинг и сбор журналов событий – агрегация метрик производительности (загрузка ЦП, использование памяти, дисковая активность), анализ системных событий и генерация уведомлений о инцидентах. Требования включают обработку потоков данных в реальном времени и интеграцию с внешними системами мониторинга (например, Zabbix).

- Резервное копирование и восстановление – планирование регулярных копий баз данных [9], виртуальных машин и файловых хранилищ с последующей верификацией целостности. Автоматизированное решение должно обеспечивать соблюдение политик резервирования (например, правило 3-2-1), шифрование данных и возможность автоматического восстановления [4].

Применение скриптовых средств автоматизации позволяет сократить временные затраты на перечисленные задачи на 70–90 %, высвобождая ресурсы администраторов для решения более сложных стратегических вопросов. В среде PowerShell указанные возможности реализуются через командлеты и расширяемые модули.

2. PowerShell как платформа для автоматизации. PowerShell представляет собой объектно-ориентированную командную оболочку и язык сценариев, разработанный корпорацией Microsoft для задач автоматизации в Windows-экосистеме. В отличие от традиционной командной строки cmd.exe, оперирующей текстовыми потоками, PowerShell работает с объектами .NET, что позволяет передавать по конвейеру структурированные данные с

сохранением типов, свойств и метаданных. Командлеты строятся по шаблону «глагол-существительное» (например, Get-Process, Set-ADUser, New-VM) [4].

Ключевые функциональные возможности PowerShell:

- Управление инфраструктурой – модули, такие как ActiveDirectory (New-ADUser, Get-ADGroup), Hyper-V (New-VM, Export-VM) и ServerManager (Install-WindowsFeature), позволяют выполнять массовые операции над доменными объектами, виртуальными машинами и серверными ролями. Например, команда Get-ADUser -Filter * | Set-ADUser -Enabled $false отключает всех неактивных пользователей за минимальное время [2].

- Удалённое выполнение – через протокол WinRM (командлет Invoke-Command -ComputerName server1, server2 -ScriptBlock {Get-Process} -AsJob) сценарии могут запускаться параллельно на тысячах удалённых хостов. Поддержка постоянных сессий (New-PSSession) упрощает реализацию многошаговых процедур [3].

- Сбор данных и формирование отчётов – командлеты Get-EventLog, Get-Counter обеспечивают получение метрик производительности и событий; Export-Csv и ConvertTo-Json позволяют генерировать отчёты в различных форматах. Возможна интеграция с системами мониторинга (Zabbix, Prometheus) через REST API.

- Управление обновлениями и пакетами – модуль PSWindowsUpdate (Install-WindowsUpdate) автоматизирует установку патчей; Install-Module из репозитория PowerShell Gallery предоставляет доступ к более чем 2000 модулей, включая средства для облачных платформ (Az для Azure, AWS.Tools).

- Безопасность и соответствие политикам – политики выполнения скриптов (ExecutionPolicy – Restricted, RemoteSigned), поддержка цифровых подписей и ролевое управление доступом (RBAC) минимизируют риски несанкционированного использования [8].

PowerShell 7+ является кроссплатформенным решением (поддерживает Windows, Linux, macOS) и сохраняет обратную совместимость с Windows PowerShell 5.1. В новой версии реализована поддержка SSH-соединений наряду с WinRM, а также механизмы параллельной обработки (ForEach-Object -Parallel) [7]. Интеграция с системами CI/CD (GitHub Actions, Azure DevOps) ускоряет реализацию DevOps-пайплайнов. Встроенная справочная система (Get-Help, Update-Help) и активное сообщество (PowerShell Gallery) обеспечивают оперативную поддержку разработчиков [9]. По данным Microsoft, применение PowerShell для автоматизации позволяет сократить время выполнения типовых задач на 80 % в крупных распределённых средах.

3. Экосистема готовых решений и модулей. PowerShell предоставляет обширную экосистему готовых модулей и сценариев, охватывающих большинство типовых административных задач. Официальные галереи Microsoft и проекты сообщества содержат тысячи проверенных решений, существенно ускоряющих внедрение автоматизации в доменных средах.

Наиболее востребованные модули и репозитории:

- PowerShell Gallery (более 2000 модулей) – содержит PSWindowsUpdate (управление патчами), PoshWSUS (настройка WSUS), dbatools (администрирование SQL Server), Carbon (конфигурации в стиле Infrastructure as Code).

- Microsoft Learn Samples – официальные коллекции скриптов для Active Directory, Hyper-V, мониторинга (Get-WinEvent, Get-Counter).

- Community-репозитории – проекты на GitHub, материалы на специализированных форумах (например, Habr) с примерами автоматизации взаимодействия с Outlook и Active Directory.

В таблице 1 приведены типовые сценарии автоматизации с указанием используемых модулей и достигаемых результатов.

Таблица 1 – Типовые сценарии автоматизации на базе PowerShell

Сценарий	Модули/Cmdlets	Пример кода	Эффект
Управление пользователями AD	ActiveDirectory	Get-ADUser -Filter * -Properties LastLogonDate | Where-Object LastLogonDate -lt (Get-Date).AddDays(-90) | Remove-ADUser -Confirm:$false	Очистка неактивных учёток за 30 сек.
Обновление серверов	PSWindowsUpdate	Get-WUList | Install-WUUpdate -AcceptAll -AutoReboot	Патчи на 100+ хостах параллельно.
Мониторинг инфраструктуры	Get-Counter, Zabbix	Get-Counter "\Processor(_Total) \% Processor Time" -Continuous | Export-Csv "CPU_Report.csv"	Метрики CPU/память в реальном времени
Резервное копирование	Hyper-V, Veeam	Get-VM | Export-VM -Path "D:\Backups"	Ежедневные снапшоты ВМ по 3-2-1 rule.
Аудит безопасности	Search-ADAccount	Search-ADAccount -LockedOut -UsersOnly | Export-Csv "LockedUsers.csv"	Отчёт по блокировкам (NETLOGON cred).
Обработка почты	Outlook COM	Get-Process Outlook -ErrorAction SilentlyContinue | Stop-Process; $outlook = New-Object -ComObject Outlook.Application	Автообработка заявок из Inbox.

Многообразие модулей PowerShell позволяет автоматизировать не только отдельные операции, но и полный жизненный цикл задач – от запуска до мониторинга, логирования и механизмов самовосстановления, что формирует замкнутый контур управления в парадигме DevOps [1].

4. Разработка специализированных сценариев для локальной инфраструктуры. В рамках настоящего исследования был определён набор ключевых PowerShell-сценариев, ориентированных на типичную локальную Windows-инфраструктуру, включающую Active Directory, серверы и рабочие станции. Выбор задач основывался на анализе частоты их выполнения, временных затрат, а также особенностей конкретной среды (интеграция с

Zabbix, сетевая топология). К выделенным направлениям относятся: аудит конфигураций, сетевые проверки и развёртывание программного обеспечения.

4.1. Сценарии для Active Directory. Разработанные сценарии решают следующие задачи (таблица 2):

Таблица 2 – Функционал сценариев автоматизации для Active Directory

Сценарий	**Основной функционал**	**Частота запуска**	**Способ инициации**
Автоматический переброс УЗ компьютера в его «рабочую OU-ку»	Переброс нововведённой учетной записи ПК в Active Directory с дефолтной OU в рабочую OU	Каждые 10 минут	Ручной, планировщик задач
Блокировка УЗ ПК после неактивности более 90 дней	Блокировка УЗ ПК в Active Directory с последующим выводом в отчетный документ	Каждый день	Ручной, планировщик задач
Автоматическая установка времени новым пользователям домена	Установка разрешенных часов работы для пользователей Active Directory	Каждый час	Ручной, планировщик задач
Установка времени работы для пользователей AD	GUI-форма для установки разрешенного времени работы в домене	По запросу	Ручной

4.2. Сценарии для аудита конфигураций и развёртывания ПО (таблица 3).

Таблица 3 – Функционал сценариев автоматизации для аудита и деплоя

Сценарий	**Основной функционал**	**Частота запуска**	**Способ инициации**
Сбор информации о компьютерах	Инвентаризация железа (CPU, RAM, BIOS), ОС, сетевые адаптеры (Get-CimInstance Win32_ComputerSystem, Win32_NetworkAdapterConfiguration)	После аутентификации клиента за ПК	Ручной, планировщик задач
Установка сложных приложений	Тихий деплой MSI/EXE с зависимостями (Start-Process -Wait, Chocolatey integration), верификация	По запросу/ежемесячно	Ручной
Автоматическое обновление ПО	Проверка версии/ПО и последующая поэтапная установка	По запросу/ежемесячно	Ручной

4.3. Сценарии для сетевых проверок (таблица 4).

Таблица 4 – Функционал сценариев автоматизации для сетевых проверок

Сценарий	**Основной функционал**	**Частота запуска**	**Способ инициации**

Аудит шлюза по умолчанию	Сбор ПК с неверным default gateway (Get-NetIPConfiguration), сравнение с эталоном, отчет	После включения ПК	GPO, ручной, планировщик задач
Аудит DNS	Сбор ПК с неверным DNS	После включения ПК	GPO, ручной, планировщик задач

Разработанный комплекс PowerShell-сценариев охватывает около 85 % типовых административных задач, характерных для локальной IT-инфраструктуры, и обеспечивает автоматизацию рутинных операций с учётом различных способов инициации – регулярные фоновые проверки через планировщик задач, разовые запуски по запросу, а также цепные вызовы в рамках ежедневного аудита.

Таким образом, применение PowerShell в качестве основного инструмента автоматизации позволяет существенно повысить эффективность управления IT-инфраструктурой, снизить нагрузку на персонал и минимизировать риски, связанные с человеческим фактором. Предложенный набор сценариев демонстрирует практическую реализацию автоматизированного подхода к решению наиболее частотных административных задач и может служить основой для дальнейшего развития систем самообслуживания и самовосстановления в корпоративных средах.

Библиографический список:

1. 11 языков программирования для DevOps и их применение // Habr URL: https://habr.com/ru/companies/first/articles/678994/ (дата обращения: 01.02.2026)

2. Get-ADUser: получаем информацию о пользователях Active Directory из PowerShell // itpro URL: https://winitpro.ru/index.php/2015/05/21/powershell-get-aduser-poluchenie-dannih-o-polzovatelyax-active-directory/ (дата обращения: 23.11.2025)

3. Invoke-Command: запуск команды/скрипта на удаленном компьютере // itpro URL: https://winitpro.ru/index.php/2020/07/27/powershell-invoke-command-zapuska-komand-na-udalennyx-kompyuterax/ (дата обращения: 12.12.2025)

4. PowerShell // Википедия URL: https://ru.wikipedia.org/wiki/PowerShell (дата обращения: 12.12.2025)

5. Доуст, Мэтью. PowerShell: практическая автоматизация: эффективная разработка скриптов от консоли до облака / Мэтью Доуст; перевел с английского А. Бойков. - Санкт-Петербург; Москва; Минск: Питер, 2025. - 415 с.

6. ИТ-инфраструктура предприятия: принципы построения и главный критерий эффективности // Корус URL: https://korusconsulting.ru/infohub/it-infrastruktura/?yclid=3370783406267564031 (дата обращения: 11.12.2025)

7. Переход с Windows PowerShell 5.1 на PowerShell 7 // GitHub URL: https://github.com/MicrosoftDocs/PowerShell-Docs/blob/main/reference/docs-conceptual/whats-new/Migrating-from-Windows-PowerShell-51-to-PowerShell-7.md (дата обращения: 10.11.2025)

8. Политика выполнения скриптов PowerShell // Заметки о Windows URL: https://windowsnotes.ru/powershell-2/politika-vypolneniya-skriptov-powershell/ (дата обращения: 16.12.2025)

9. Центральное хранилище для обмена и получения кода PowerShell // Галлерея Powershell URL: // https://www.powershellgallery.com/ (дата обращения: 10.11.2025)

10. Что такое ИТ-инфраструктура компании: понятия, задачи и особенности, в том числе, на примере НРД // Хабр URL: https://habr.com/ru/articles/684902/ (дата обращения: 12.12.2025)