

Как собрать статистику с MS Exchange 2019

Инструкция для администраторов

Оглавление

Статистика по наполнению системы	3
Статистика по активности пользователей	7

Статистика по наполнению системы

Статистику нужно собирать на пассивном сервере (Passive Node):

1. Скопируйте скрипт в файл с именем Get-ExchangeStats.ps1 и сохраните его на диске. Например: C:\Scripts\.

```
#Requires -Version 5.1
<#
.SYNOPSIS
    Простой сбор статистики Exchange 2019

.DESCRIPTION
    Собирает метрики по почте и календарям

.PARAMETER ResultPath
    Путь для сохранения отчёта (по умолчанию текущая папка)

.EXAMPLE
    .\Get-ExchangeStats.ps1
    .\Get-ExchangeStats.ps1 -ResultPath "C:\Reports"
#>

param(
    [string]$ResultPath = $PSScriptRoot
)

$ErrorActionPreference = "Stop"

if (-not (Get-Command Get-Mailbox -ErrorAction SilentlyContinue)) {
    Write-Error "Запустите скрипт из Exchange Management Shell"
    exit 1
}

Write-Host "`n"
ForegroundColor Cyan
Write-Host "    СБОР СТАТИСТИКИ EXCHANGE 2019" -ForegroundColor Cyan
Write-Host "`n"
ForegroundColor Cyan

if (-not (Test-Path $ResultPath)) {
    New-Item -ItemType Directory -Path $ResultPath -Force | Out-Null
}

Write-Host "[1/4] Получение списка пользователей..." -ForegroundColor Yellow
$mailboxes = Get-Mailbox -ResultSize Unlimited -Filter "RecipientTypeDetails -eq 'UserMailbox'"
$totalUsers = $mailboxes.Count
Write-Host "    Найдено пользователей: $totalUsers" -ForegroundColor Green

Write-Host "[2/4] Сбор статистики почты..." -ForegroundColor Yellow

$mailStats = @()
$totalMessages = 0

foreach ($mbx in $mailboxes) {
    Write-Progress -Activity "Обработка почтовых ящиков" -Status $mbx.DisplayName -
```

```

PercentComplete ((($mailStats.Count / $totalUsers) * 100)

    try {
        $stats = Get-MailboxFolderStatistics -Identity $mbx.Alias -FolderScope Inbox -
ErrorAction SilentlyContinue
        $itemsCount = ($stats | Measure-Object -Property ItemsInFolder -Sum).Sum

        $totalMessages += $itemsCount
        $mailStats += [PSCustomObject]@{
            DisplayName = $mbx.DisplayName
            EmailAddress = $mbx.PrimarySmtpAddress
            TotalMessages = $itemsCount
        }
    }
    catch {
        Write-Warning "Ошибка: $($mbx.DisplayName)"
    }
}

Write-Progress -Activity "Обработка почтовых ящиков" -Completed

$maxMessagesUser = $mailStats | Sort-Object TotalMessages -Descending | Select-Object -
First 1

Write-Host "[3/4] Сбор статистики календарей..." -ForegroundColor Yellow

$calendarStats = @()
$calendarFolders = @()
$usersWithCalendar = 0

foreach ($mbx in $mailboxes) {
    try {
        $calendars = Get-MailboxFolderStatistics -Identity $mbx.Alias -FolderScope Calendar
-ErrorAction SilentlyContinue

        if ($calendars.Count -gt 0) {
            $usersWithCalendar++
            $calendarCount = $calendars.Count
            $totalEvents = ($calendars | Measure-Object -Property ItemsInFolder -Sum).Sum

            $calendarStats += [PSCustomObject]@{
                DisplayName = $mbx.DisplayName
                EmailAddress = $mbx.PrimarySmtpAddress
                CalendarCount = $calendarCount
                TotalEvents = $totalEvents
            }

            foreach ($cal in $calendars) {
                $calendarFolders += [PSCustomObject]@{
                    DisplayName = $mbx.DisplayName
                    CalendarName = $cal.Name
                    EventsCount = $cal.ItemsInFolder
                }
            }
        }
    }
    catch {
    }
}

$totalCalendarEvents = ($calendarFolders | Measure-Object -Property EventsCount -Sum).Sum
$maxCalendarsUser = $calendarStats | Sort-Object CalendarCount -Descending | Select-Object
-First 1

```

```

$maxEventsCalendar = $calendarFolders | Sort-Object EventsCount -Descending | Select-Object
-First 1

Write-Host "[4/4] Сбор статистики писем за период..." -ForegroundColor Yellow

$endDate = Get-Date
$startDate1Day = $endDate.AddDays(-1)
$startDate7Days = $endDate.AddDays(-7)

$messages1Day = 0
$messages7Days = 0

try {
    $tracking1Day = Get-MessageTrackingLog -Start $startDate1Day -End $endDate -EventId
"DELIVER" -ResultSize Unlimited -ErrorAction SilentlyContinue
    $messages1Day = $tracking1Day.Count

    $tracking7Days = Get-MessageTrackingLog -Start $startDate7Days -End $endDate -EventId
"DELIVER" -ResultSize Unlimited -ErrorAction SilentlyContinue
    $messages7Days = $tracking7Days.Count

    Write-Host "        Данные за период собраны" -ForegroundColor Green
}
catch {
    Write-Warning "Не удалось собрать данные MessageTracking (требуется права
администратора)"
    $messages1Day = "N/A"
    $messages7Days = "N/A"
}

Write-Host "`n"
ForegroundColor Green
Write-Host "        РЕЗУЛЬТАТЫ СТАТИСТИКИ" -ForegroundColor Green
Write-Host "`n"
ForegroundColor Green

$results = @(
    [PSCustomObject]@{ Метрика = "Количество пользователей"; Значение = $totalUsers }
    [PSCustomObject]@{ Метрика = "Всего писем"; Значение = $totalMessages }
    [PSCustomObject]@{ Метрика = "Макс. количество писем у пользователя"; Значение = if
($maxMessagesUser) { "$($maxMessagesUser.TotalMessages)" } else { "-" } }
    [PSCustomObject]@{ Метрика = ""; Значение = "" }
    [PSCustomObject]@{ Метрика = "Количество пользователей с календарем"; Значение =
$usersWithCalendar }
    [PSCustomObject]@{ Метрика = "Всего событий календаря (items)"; Значение =
$totalCalendarEvents }
    [PSCustomObject]@{ Метрика = "Макс. календарей у пользователя"; Значение = if
($maxCalendarsUser) { "$($maxCalendarsUser.CalendarCount)" } else { "-" } }
    [PSCustomObject]@{ Метрика = "Макс. событий в одном календаре"; Значение = if
($maxEventsCalendar) { "$($maxEventsCalendar.EventsCount)" } else { "-" } }
    [PSCustomObject]@{ Метрика = ""; Значение = "" }
    [PSCustomObject]@{ Метрика = "Писем созданных за 1 день"; Значение = $messages1Day }
    [PSCustomObject]@{ Метрика = "Писем созданных за 7 дней"; Значение = $messages7Days }
)

$results | Format-Table -AutoSize -Property Метрика, Значение

$timestamp = Get-Date -Format "yyyyMMdd_HHmms"
$csvPath = Join-Path $ResultPath "ExchangeStats_{$timestamp}.csv"
$results | Export-Csv -Path $csvPath -NoTypeInformation -Encoding UTF8 -Delimiter ";"

Write-Host "`n"
ForegroundColor Cyan

```

```
Write-Host " Отчёт сохранён: $csvPath" -ForegroundColor Green
Write-Host "`n" -
ForegroundColor Cyan
```

2. Запустите Exchange Management Shell от имени администратора.

3. Перейдите в папку с скриптом Get-ExchangeStats.ps1:

```
cd C:\Scripts
```

4. Запустите скрипт, чтобы сохранить отчет в текущей папке:

```
\Get-ExchangeStats.ps1
```

Чтобы сохранить в конкретной папке:

```
\Get-ExchangeStats.ps1 -ResultPath "D:\ExchangeReports"
```

Как выглядит отчет после выполнения скрипта

РЕЗУЛЬТАТЫ СТАТИСТИКИ

Метрика	Значение
-----	-----
Количество пользователей	13
Всего писем	1234
Макс. количество писем у пользователя	1006
Количество пользователей с календарем	11
Всего событий календаря (items)	146
Макс. календарей у пользователя	6
Макс. событий в одном календаре	60
Писем созданных за 1 день	0
Писем созданных за 7 дней	0

Отчёт сохранён: c:\ExchangeReports\ExchangeStats_20260414_101421.csv

Чтобы получать данные о количестве созданных писем за 1 и 7 дней убедитесь, что включено отслеживание писем (Message Tracking):

```
Get-TransportService | Format-List Name, MessageTrackingLogEnabled
```

Чтобы получать подробную информацию формируйте отчет ежедневно в течение недели.

Статистика по активности пользователей

Статистику нужно собирать на серверах с ролью Client Access (CAS):

1. Скопируйте скрипт в файл с именем Get-OWAStats.ps1 и сохраните его на диске каждого сервера CAS. Например: C:\Scripts\.

```
# Requires -Version 5.1

<#
.SYNOPSIS
    Сбор метрик активности OWA с Exchange сервера

.DESCRIPTION
    Скрипт собирает производительность OWA счетчиков и сохраняет в CSV файл.
    Счетчики соответствуют стандартному набору производительности Exchange.

.PARAMETER ResultPath
    Путь для сохранения CSV файла (по умолчанию: C:\OWA_Metrics)

.PARAMETER IntervalSeconds
    Интервал сбора данных в секундах (по умолчанию: 5)

.EXAMPLE
    .\Get-OWAStats.ps1
    .\Get-OWAStats.ps1 -ResultPath "D:\Reports" -IntervalSeconds 10
#>

param(
    [string]$ResultPath = "C:\OWA_Metrics",
    [int]$IntervalSeconds = 5
)

$counters = @(
    "\MSExchange OWA\Attachments Uploaded Since OWA Start",
    "\MSExchange OWA\Average Search Time",
    "\MSExchange OWA\Calendar View Refreshed",
    "\MSExchange OWA\Calendar Views Loaded",
    "\MSExchange OWA\Current Unique Users",
    "\MSExchange OWA\Current Users",
    "\MSExchange OWA\Logons/sec",
    "\MSExchange OWA\Mail View Refreshes",
    "\MSExchange OWA\Mail Views Loaded",
    "\MSExchange OWA\Messages Sent",
    "\MSExchange OWA\Peak User Count",
    "\MSExchange OWA\Requests/sec",
    "\MSExchange OWA\Searches",
    "\MSExchange OWA\Total Unique Users"
)

if (-not (Test-Path $ResultPath)) {
    New-Item -ItemType Directory -Path $ResultPath -Force | Out-Null
}

$timestamp = Get-Date -Format "yyyyMMdd_HH:mm:ss"
$outputFile = Join-Path $ResultPath "OWA_Metrics_{$timestamp}.csv"

$header = "Timestamp," + ($counters -join ",")
$header | Out-File -FilePath $outputFile -Encoding UTF8
```

```

Write-Host "`n"
ForegroundColor Cyan
Write-Host "        СБОР МЕТРИК OWA (Outlook Web Access)" -ForegroundColor Cyan
Write-Host "`n" -
ForegroundColor Cyan

Write-Host "    Счётчиков: $($Counters.Count)" -ForegroundColor Yellow
Write-Host "    Интервал: $IntervalSeconds секунд" -ForegroundColor Yellow
Write-Host "    Файл: $outputFile" -ForegroundColor Yellow
Write-Host "`n    Для остановки нажмите Ctrl+C...`n" -ForegroundColor Green

$iteration = 0

while ($true) {
    $iteration++

    try {
        $data = Get-Counter -ComputerName "." -Counter $counters -SampleInterval
$IntervalSeconds -MaxSamples 1 -ErrorAction Stop

        if ($data -ne $null) {
            $timestamp = $data.Timestamp.ToString("yyyy-MM-dd HH:mm:ss")

            $values = @()
            foreach ($counter in $counters) {
                $sample = $data.CounterSamples | Where-Object { $_.Path -like "$counter" }
                if ($sample -ne $null) {
                    $rawValue = $sample.CookedValue

                    if ($counter -like "*Average Search Time*") {
                        $val = [math]::Round($rawValue, 2)
                    }
                    elseif ($rawValue -ge 0 -and $rawValue -lt 1000000) {
                        $val = [math]::Round($rawValue, 0)
                    }
                    else {
                        $val = [math]::Round($rawValue, 2)
                    }
                    $values += $val
                } else {
                    $values += "N/A"
                }
            }

            $line = "$timestamp," + ($values -join ",")
            Add-Content -Path $outputFile -Value $line

            $currentUsers = if ($values[4] -ne "N/A") { $values[4] } else { $values[5] }
            $requestsPerSec = if ($values[11] -ne "N/A") { $values[11] } else { "N/A" }

            Write-Host "[ $timestamp ] Итерация $iteration | Текущие пользователи:
$currentUsers | Requests/sec: $requestsPerSec" -ForegroundColor Green
        }
    }
    catch {
        Write-Warning "Ошибка при сборе данных: $($_.Exception.Message)"
        Write-Host "Повторная попытка через $IntervalSeconds секунд..." -ForegroundColor
Yellow
        Start-Sleep -Seconds $IntervalSeconds
        continue
    }
}

```

```
Start-Sleep -Milliseconds 100
}
```

2. Убедитесь, что запущена служба PerfHost на каждом сервере CAS.

```
Get-Service -Name "PerfHost" | Format-Table Name, Status
```

3. Запустите Exchange Management Shell от имени администратора на каждом сервере CAS.

4. Перейдите в папку с скриптом Get-ExchangeStats.ps:

```
cd C:\Scripts
```

5. Запустите скрипт, чтобы сохранить отчет в текущей папке:

```
\Get-OWAStats.ps1
```

Чтобы сохранить в конкретной папке:

```
\Get-OWAStats.ps1 -ResultPath "D:\ExchangeReports"
```

Как выглядит отчет после выполнения скрипта

СБОР МЕТРИК OWA (Outlook Web Access)

Счётчиков: 14

Интервал: 5 секунд

Файл: c:\ExchangeReports\OWA_Metrics_20260414_103825.csv

Для остановки нажмите Ctrl+C...

```
[2026-04-14 10:38:30] Итерация 1 | Текущие пользователи: 0 | Requests/sec: 0
[2026-04-14 10:38:36] Итерация 2 | Текущие пользователи: 0 | Requests/sec: 0
[2026-04-14 10:38:41] Итерация 3 | Текущие пользователи: 1 | Requests/sec: 1
[2026-04-14 10:38:46] Итерация 4 | Текущие пользователи: 1 | Requests/sec: 0
[2026-04-14 10:38:51] Итерация 5 | Текущие пользователи: 1 | Requests/sec: 0
[2026-04-14 10:38:56] Итерация 6 | Текущие пользователи: 1 | Requests/sec: 0
[2026-04-14 10:39:01] Итерация 7 | Текущие пользователи: 1 | Requests/sec: 0
[2026-04-14 10:39:06] Итерация 8 | Текущие пользователи: 1 | Requests/sec: 0
```

Чтобы получать подробную информацию формируйте отчет ежедневно в течение недели на каждом сервере CAS.

 Автор: Груздев Никита

🕒 14 апреля 2026 г.