SCRIPT #09 · AUTOMAZIONE E REPORTISTICA GENERALE

Verifica generale salute infrastruttura

Esegue una serie combinata di controlli rapidi (rete, disco, servizi) e produce un esito complessivo pass/fail.

AUTORE	COPYRIGHT	PREREQUISITI
Francesco Larossa	© 2026 SysAdmin Italia	Connettivita verso i sistemi target

Cosa fa questo script

Lo script esegue in sequenza una serie di controlli di base (raggiungibilita, spazio disco, servizi critici) su un elenco di sistemi, aggregando i risultati in un unico esito complessivo per ciascun sistema: OK se tutti i controlli passano, altrimenti l'elenco specifico di cosa non va.

Come si usa

Prepara un file di testo con un sistema per riga, poi esegui lo script.

```
.\Verifica-SaluteInfrastruttura.ps1 -ListaServer "C:\Dati\server.txt"
```

In quale contesto usarlo

Pensato come primo controllo del mattino, un solo comando che da una risposta immediata su "e tutto a posto?" prima di scendere nel dettaglio con gli script piu specifici di questa raccolta solo dove effettivamente serve, risparmiando tempo quando, come nella maggior parte dei giorni, non c'e nulla di anomalo da segnalare.

Codice completo

Il codice è lo stesso del file Verifica-SaluteInfrastruttura.ps1: per usarlo scarica il file dalla pagina invece di copiarlo da questo PDF.

```
<#
.SYNOPSIS
    Esegue una serie combinata di controlli rapidi e produce un esito complessivo per sistema.
#>
param(
    [Parameter(Mandatory=$true)]
    [string]$ListaServer,
    [string[]]$ServiziCritici = @("Spooler","W32Time"),
    [int]$SogliaDiscoPercento = 15
)

$server = Get-Content -Path $ListaServer

foreach ($srv in $server) {
    $problemi = @()

    $raggiungibile = Test-Connection -ComputerName $srv -Count 1 -Quiet -ErrorAction SilentlyContinue
    if (-not $raggiungibile) {
        Write-Host "$srv : FALLITO (non raggiungibile)" -ForegroundColor Red
        continue
```

```
    }

    try {
        $disco = Get-CimInstance -ComputerName $srv -ClassName Win32_LogicalDisk -Filter "DeviceID='C:'" -ErrorAction Stop
        $percentualeLibera = ($disco.FreeSpace / $disco.Size) * 100
        if ($percentualeLibera -lt $SogliaDiscoPercento) { $problemi += `
            "disco C: sotto soglia ($([math]::Round($percentualeLibera,1))%)" }

        foreach ($svc in $ServiziCritici) {
            $stato = Get-CimInstance -ComputerName $srv -ClassName Win32_Service -Filter "Name='$svc'" `
                -ErrorAction SilentlyContinue
            if ($stato -and $stato.State -ne "Running") { $problemi += "servizio $svc non attivo" }
        }
    }
    catch {
        $problemi += "impossibile verificare disco/servizi: $_"
    }

    if ($problemi.Count -eq 0) {
        Write-Host "$srv : OK" -ForegroundColor Green
    } else {
        Write-Host "$srv : ATTENZIONE - $($problemi -join '; ')" -ForegroundColor Yellow
    }
}
```