PRTG – Verfügbare VDIs überwachen

Im letzten Artikel habe ich beschrieben, wir mittels PRTG der Load Index der Terminalserver überwacht werden kann.

Wenn wir uns die VDIs (single user OS), dann interessiert uns da normalerweise die Auslastung weniger. Dafür wollen wir wissen, wie viele VDIs überhaupt noch zur Verfügung stehen im Kontext von „random non-static“.

Im Vergleich zu den Terminalservern ist hier die Erstellung eines Custom Sensors sogar einfacher, da uns der Citrix Controller mit allen notwendigen Informationen versorgt:

```
###
# Prepare an array and get the information from a Citrix controller by an Invoke-Command
# Citrix PS commands are only available on those servers
$Data = $null
$Data=@()

$Data += Invoke-Command -Computername $server -credential $credentials -ScriptBlock {

    # Load Citrix PS Snapin
    add-pssnapin citrix*

    # Get delivery groups with the needed information
     $DeliveryGroups=Get-BrokerDesktopGroup | Select-Object -Property Name, TotalDesktops, DesktopsAvailable, InMaintenanceMode

    # Set the array for later work
    $DG=@()
    ForEach ($group in $DeliveryGroups){
                                                $DG      += [PSCustomObject]@{Name=$group.Name;Total=$group.TotalDesktops;Available=$group.DesktopsAvailable;Maintenance=$group.InMainte
```

```
nanceMode}
    }

    # Return data to PRTG probe
    return $DG
}
```

Diese Informationen können wir in einer weiteren Schlaufe dann auch gleich für die PRTG Ausgabe benutzen. Das komplette Skript steht zum Download bereit, daher beschreibe ich hier nur nochmals ein paar Eckpunkte.

Aus den Daten können die noch verfügbaren VDIs in Prozent umgerechnet und erneut für PRTG auf die nächste Ganzzahl gerundet werden:

```
    # Prepare variables based on the array data
    $DGName = $Dataset.Name
    $DGTotal = $Dataset.Total
    $DGAvailable = $Dataset.Available
    # Calculate the percentage of the available VDIs
    $DGPercent = "{0:N0}" -f (100/$DGTotal*$DGAvailable)
    $DGMaintenance = $Dataset.Maintenance
```

Kommt im Namen der Bereitstellungsgruppe „test“ vor, so wollen wir im PRTG nie einen Alarm sondern höchstens eine Warnung auslösen:

```
    # Determine if a delivery group is a test group
    # Test DGs will never go into alarm state
    # Test DGs in maintenance will have a percentage value of
100 to avoid an alarm
    If ($DGName -like "*test*"){
        $AlertString = $null
        $AlertLimit = $null
         $WarningString = "Warning - Test VDI capacity reachs
the end"
        If ([Int64]$DGPercent -ge 20) {
            $RetState = $returnStateOK
                                        $returnState   +=
[PSCustomObject]@{Name=$DGName;State=[Int64]$RetState}
        }
```

```
        ElseIf ($DGMaintenance){
            $RetState=$returnStateWarning
            $DGPercent=100
        }
        Else {
            $RetState = $returnStateWarning
                                        $returnState   +=
[PSCustomObject]@{Name=$DGName;State=[Int64]$RetState}
        }
    }
```

Ist eine Bereitstellungsgruppe im Wartungsmodus, so geben wir eine Verfügbarkeit von 100% an PRTG aus, um einen Alarm oder eine Warnung komplett zu vermeiden:

```
    # Determine if a delivery group is in maintenance
    # DGs in maintenance goes into warning state, but doesn't
affect the whole sensor
    # DGs in maintenance will not be stated based on her usage
    # DGs in maintenance will have a percentage value of 100
to avoid an alarm
    ElseIf ($DGMaintenance){
        $RetState=$returnStateWarning
        $DGPercent=100
    }
```

Die restlichen Bereitstellungsgruppen werden anhand der Prozentwerte eingestuft und entsprechende PRTG Ausgaben aufbereitet.

Die Ausgabe sieht dann folgendermassen aus, wobei die ausgegrauten Channels ehemalige bereits gelöschte Bereitstellungsgruppen sind:

Viel Spass beim Nachbauen :-)

PRTGCustomCitrixDGAvailableVDIs.ps1_Herunterladen