

# Notify license shortage with automation

Requirements: Create an App Registration with `LicenseAssignment.ReadWrite.All` permissions and a client secret.

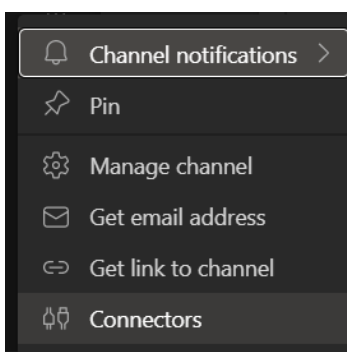
With this automation all licenses and their usage are evaluated via Graph API. A message is posted in a team channel when a certain amount of usage of a license is reached.

## Use case

This automation helps large organizations with many Microsoft 365 licenses to keep track. If this automation is carried out regularly, interruptions in operation and overbooked licenses can be noticed at an early stage. To run such scripts on a regular basis, an Azure Runbook is a good choice.

## Teams webhook notification

To send a notification to a Teams Channel you can use a webhook. To create a webhook for a channel you can rightclick on a channel and then choose "Connectors".



There you can then configure a new incoming webhook:

## Connectors for your team

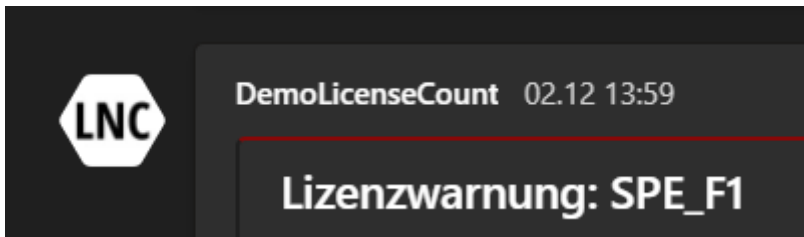


### Incoming Webhook

Send data from a service to your Office 365 group in real time.

[Configure](#)

You have to name the webhook first. This name will be displayed as the sender of the message. You can upload an image which will then be used as a profile picture of the sender. After that you can click "create". This will return an URL which then can be used to send a payload message. This will then look like that:



## PowerShell script

This PowerShell script needs an App Registration and the according LicenseAssignment.ReadWrite.All Graph API permission. This script uses the three Azure Automation variables "\$tenantid", "\$clientid" and "\$clientsecret" as well as the perviously created "\$WebhookURI", which should be added before.

```
$tenantId=Get-AutomationVariable -Name "<nameoftenantidrunbookvariable>"
$ClientId=Get-AutomationVariable -Name "<nameofclientidrunbookvariable>"
$CredentialObject=Get-AutomationPSCredential -Name '<nameofclientsecretrunbookcredentials>'
$ClientSecret = $CredentialObject.GetNetworkcredential().password

$WebhookURI = ""
$PercentageAlert = "98"

$Body = @{
    "tenant" = $TenantId
    "client_id" = $ClientId
    "scope" = "https://graph.microsoft.com/.default"
    "client_secret" = $ClientSecret
    "grant_type" = "client_credentials"
}

$Params = @{
    "Uri" = "https://login.microsoftonline.com/$TenantId/oauth2/v2.0/token"
```

```

"Method" = "Post"
"Body" = $Body
"ContentType" = "application/x-www-form-urlencoded"
}

$AuthResponse = Invoke-RestMethod @Params

$Header = @{
    "Authorization" = "Bearer $($AuthResponse.access_token)"
}

$AllLicenses = Invoke-RestMethod -Method GET -Uri "https://graph.microsoft.com/v1.0/subscribedSkus" -Header
$Header

Foreach($License in $AllLicenses.value){
    if($License.prepaidUnits.enabled -ge 50){
        try{
            $LicensePercentage = ($License.consumedUnits/$License.prepaidUnits.enabled*100)
        }
        catch{
            $null
        }
        if($LicensePercentage -ge $PercentageAlert){
            $CurrentTime = Get-Date
$JsonBody = @"
{
    "@context": "https://schema.org/extensions",
    "@type": "MessageCard",
    "themeColor": "880808",
    "title": "License warning: $($License.skuPartNumber)",
    "text": "License has more than 98% allocations. Please order new licenses in order not to jeopardize the
operation.<br><br>Licensename: $($License.skuPartNumber) <br><br>Licenses available:
$($License.prepaidUnits.enabled - $License.consumedUnits) <br><br> Licenses total:
$($License.prepaidUnits.enabled) <br> Licenses assigned: $($License.consumedUnits) <br><br> Time of the
evaluation: $($CurrentTime.addHours(2))<br><br> More details about the license:
https://admin.microsoft.com/#/licensedetailpage/$($License.skuld)",
    }
"@

    Invoke-RestMethod -Method Post -Body $JsonBody -Uri $WebhookURI -Header @{"content-type" =
"application/json; charset=UTF-8"}

```

```
}  
}  
}
```

---

Revision #16

Created 23 December 2022 08:33:24 by Luca Noah Caprez

Updated 11 September 2023 08:16:38 by Luca Noah Caprez