



# AVD Tech Fest 2025

Roll the Dice in Berlin!



# Windows 365 License Management the Graph API edition

Start 13:30

Esther Barthel



citrix™

IGEL

LOGINVS

ezeep by ThinPrint

numecent

UltrArmor

Sr. Solutions Architect  
Sr. DevOps Engineer



Esther  
Barthel

<https://www.linkedin.com/in/ebarthel/>



# End User Computing





The Plex logo is displayed on a black square background. The word "plex" is written in a lowercase, sans-serif font. The letter "x" is unique, featuring a yellow vertical bar on its right side.

plex





AVD



W365 CloudPC



W365 CloudPC



W365 CloudPC



W365 CloudPC  
(Config-as-Code)

The Microsoft Graph logo is displayed on a blue gradient background. The text "Microsoft Graph" is on the left in white, sans-serif font. To the right is a 3D blue cube icon composed of several facets.

Microsoft Graph

W365 CloudPC  
(Config-as-Code)

## Every Bite a

Better Burger!

Burger Pasta Salads



Beef Burger

Onion With Cheese  
\$18.00



Chicken Burger

Cheese With Chicken  
\$12.00



Classic Burger

Beef Wit Lottuce  
\$24.00



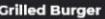
Grilled Burger

Griller Chicken  
\$14.00



Classic Burger

Beef Wit Lottuce  
\$24.00



Grilled Burger

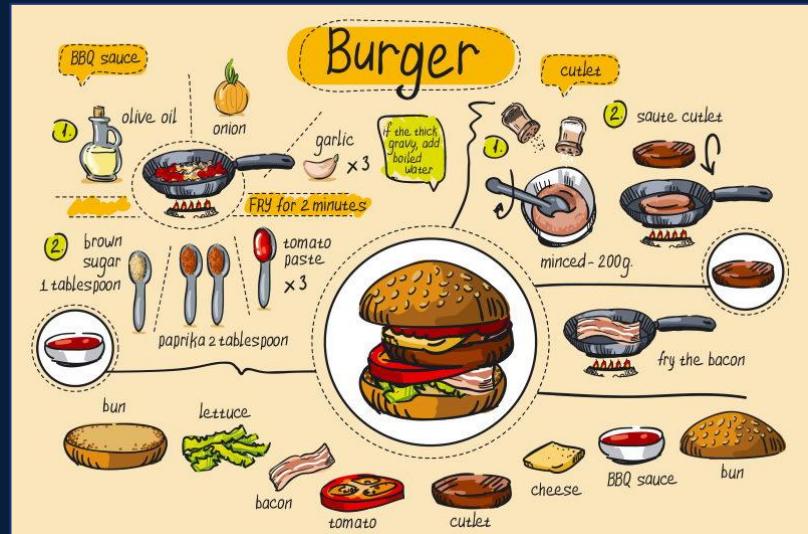
Griller Chicken  
\$14.00



Fresh Beef  
Burger With Cheese...

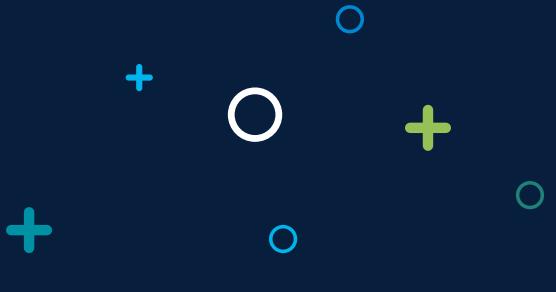
Beef burger With Fresh Cheese And Onion Served With Fries And Drink. Enjoy our 20% Off With New Promo Code.

Declarative code  
(templates)



Imperative code  
(scripts)

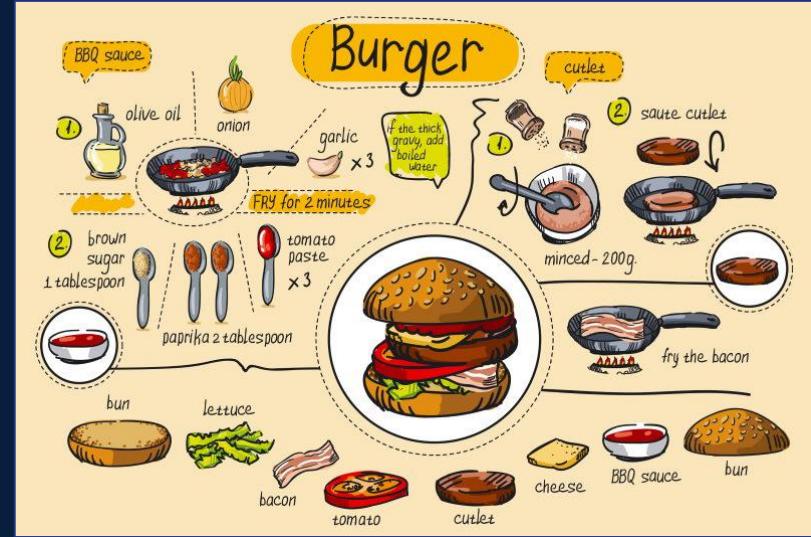




# Getting Started with the Graph API

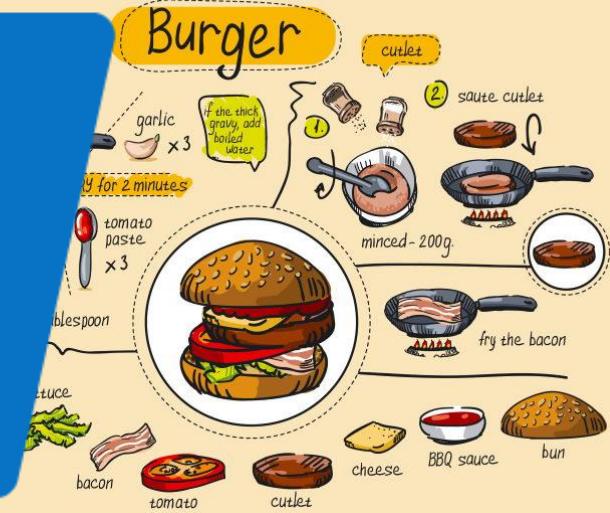


# Microsoft Graph API

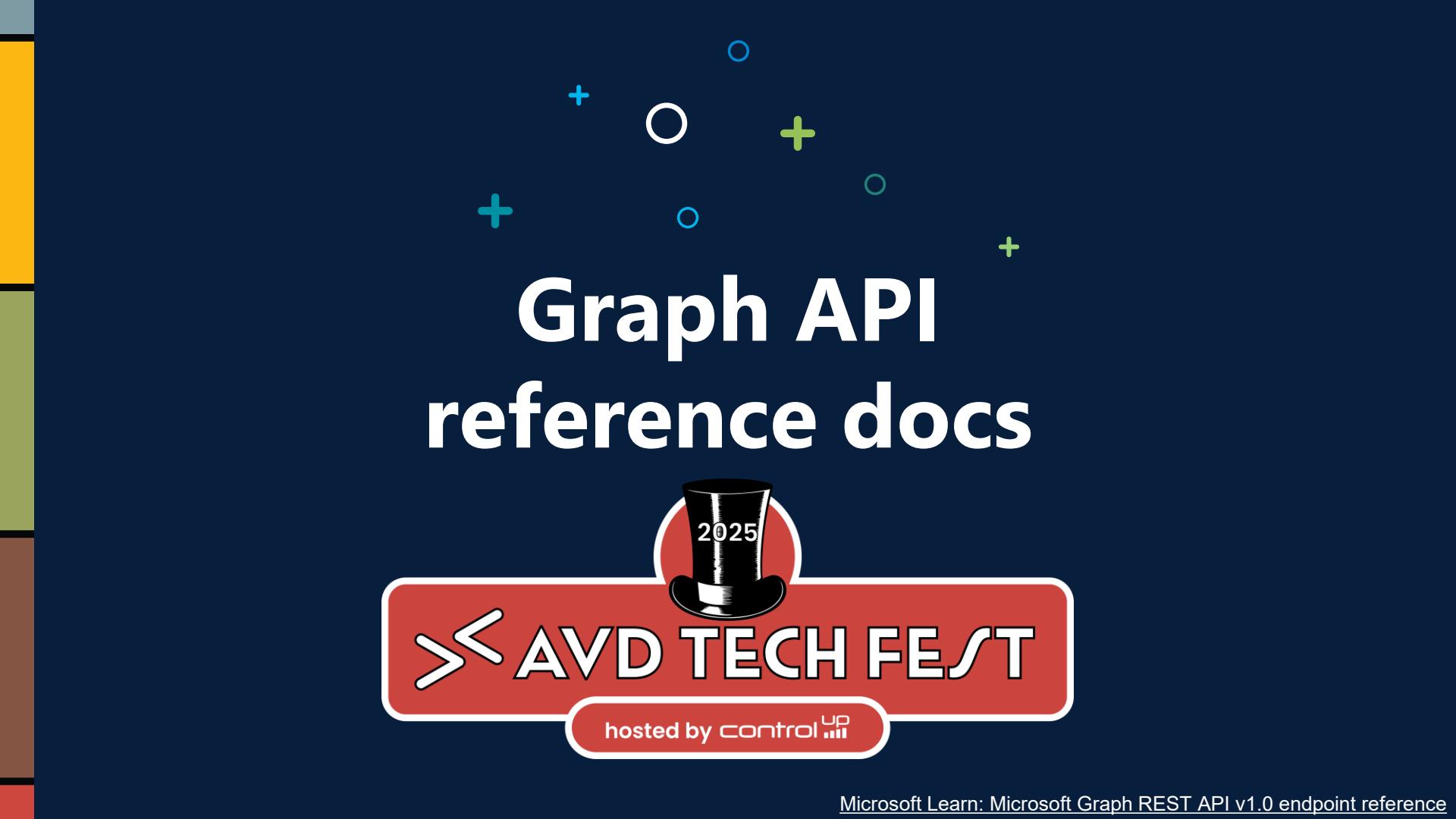


Imperative code (scripts)

# Microsoft Graph API



## Imperative code (scripts)



# Graph API reference docs



# Use the Microsoft Graph API

Microsoft Graph is a RESTful web API that enables you to access Microsoft Cloud service resources. After you [register your app](#) and [get authentication tokens](#) for a user or service, you can make requests to the Microsoft Graph API.

## Call a REST API method

To read from or write to a resource such as a user or an email message, you construct a request that looks like the following:

HTTP

 Copy

```
{HTTP method} https://graph.microsoft.com/{version}/{resource}?{query-parameters}
```

## 4. Request an access token

In the OAuth 2.0 client credentials grant flow, you use the application ID and client secret values that you saved when you registered your app to request an access token directly from the Microsoft identity platform /token endpoint.

You specify the preconfigured permissions by passing <https://graph.microsoft.com/.default> as the value for the scope parameter in the token request.

Send a POST request to the /token identity platform endpoint to acquire an access token. In this request, the client uses the client secret.

HTTP

cURL

HTTP

Copy

request, the client uses the client secret.

HTTP

cURL

HTTP

Copy

// Line breaks are for legibility only.

POST https://login.microsoftonline.com/{tenant}/oauth2/v2.0/token HTTP/1.1

Host: login.microsoftonline.com

Content-Type: application/x-www-form-urlencoded

client\_id=535fb089-9ff3-47b6-9bfb-4f1264799865

&scope=https%3A%2F%2Fgraph.microsoft.com%2F.default

&client\_secret=qWgdYA....L1qKv5bPX

&grant\_type=client\_credentials

request, the client uses the client secret.

HTTP

cURL

HTTP

Copy

// Line breaks are for legibility only.

POST https://login.microsoftonline.com/{tenant}/oauth2/v2.0/token HTTP/1.1

Host: login.microsoftonline.com

Content-Type: application/x-www-form-urlencoded

client\_id=535fb089-9ff3-47b6-9bfb-4f1264799865

&scope=https%3A%2F%2Fgraph.microsoft.com%2F.default

&client\_secret=qWgdYA....L1qKv5bPX

&grant\_type=client\_credentials

# IT Processes

# W365 License Management explaining the workflow



# W365 CloudPC License Management

Get-W365CPC-InactiveFor60Days-Report-Data (phase 1)



Revoke-W365CPC-License (phase 2)



Notify-W365CPC-User (phase 3)



# W365 CloudPC License Management

Get-W365CPC-InactiveFor60Days-Report-Data (phase 1)

Retrieve CloudPCReports:  
getTotalAggregatedRemoteConnectionReports

Enrich report data for  
readability & follow-up actions

Filter report data on  
CloudPC Types &  
60 days inactivity

# W365 CloudPC License Management



[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

## HTTP request

HTTP Copy  
POST /deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports

## Request headers

Expand table

Name	Description
Authorization	Bearer (token). Required. Learn more about <a href="#">authentication</a> and authorization.
Content-Type	application/json. Required.

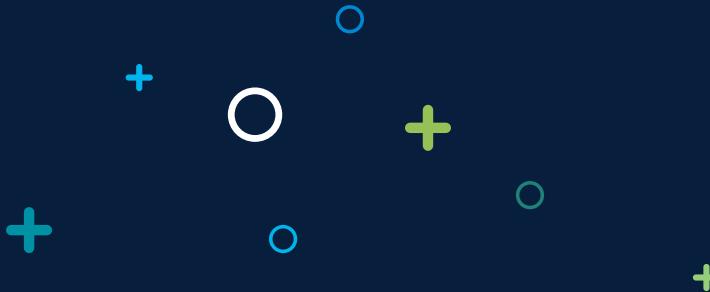
## Request body

In the request body, supply a JSON representation of the parameters.

The following table shows the parameters that can be used with this action.

Expand table

Parameter	Type	Description
filter	String	OData filter syntax. Supported filters include <code>and</code> , <code>or</code> , <code>lt</code> , <code>le</code> , <code>gt</code> , <code>ge</code> , and <code>eq</code> .
select	String collection	OData select syntax. Represents the selected columns of the reports.
search	String	Specifies a string to search
groupBy	String collection	Specify how to group the reports. If used, must have the same contents as select parameter
orderBy	String collection	Specify how to sort the reports.
skip	Int32	Number of records to skip.
top	Int32	The number of top records to return.



# W365 License Management

## Postman API testing



# W365 CloudPC License Management (Postman)

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, New, Import, Collections, Environments, Flows, APIs, Mock servers, Monitors, History.
- Current Collection:** 00. GraphAPI Demos
- Request:** POST CloudPCReports: getTotalAggregatedRemoteConnectionReports
- URL:** https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports?&\$count=1...
- Method:** POST
- Body:** raw JSON (selected)  
```json

```
1 {
2     "skip": 0,
3     "select": [],
4     "top": 100,
5     "search": "",
6     "orderBy": [
7         "TotalUsageInHour"
8     ]
9 }
```
- Response:** 200 OK, 1.07 s, 2.03 KB  
```json

```
1 {
2     "TotalRowCount": 2,
3     "Schema": [
4         ...
5     ],
6     "Values": [
7         [
8             ...
9         ],
10        [
11            [
12                "627a21fe-a554-47ca-9a3c-f14fe8c3e99b",
13                "CPC-test0-N61MG",
14                "test01@cognitionitdev.onmicrosoft.com",
15                null,
16                0.0,
17                "645f6084-98ba-4ed4-9299-817ed164860d",
18                ...
19            ]
20        ]
21    ]
22 }
```
- Bottom Navigation:** Postbot, Runner, Capture requests, Cloud Agent, Cookies, Vault, Trash

# W365 CloudPC License Management (Postman)



## cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

HTTP 00. GraphAPI Demos / Windo... / 03. License Management / [CloudPCReports: getTotalAggregatedRemoteConnectionReports](https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?$count=1...) Save Share

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?\$count=1... Send

Params ● Authorization Headers (12) Body ● Scripts Settings Cookies Beautify

Body raw binary GraphQL JSON

```
1 {  
2   "skip": 0,  
3   "select": [],  
4   "top": 1,  
5   "search": "",  
6   "orderBy": [  
7     "TotalUsageInHour"  
8   ]  
9 }
```

Body Cookies Headers (7) Test Results 200 OK • 783 ms • 1.64 KB • Save Response

{ } JSON Visualize

```
1 {  
2   "TotalRowCount": 2,  
3   "Schema": [...  
68   ],  
69   "Values": [...  
88   ]  
89 }
```

# W365 CloudPC License Management (Postman)



## cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

HTTP 00. GraphAPI Demos / Windo... / 03. License Management / [CloudPCReports: getTotalAggregatedRemoteConnectionReports](https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?$count=1...) Save Share

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?\$count=1...

Send

Params • Authorization Headers (12) Body • Scripts Settings Cookies Beautify

Body

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "skip": 0,
3   "select": [],
4   "top": 1,
5   "search": "",
6   "orderBy": [
7     "TotalUsageInHour"
8   ]
9 }
```

Body Cookies Headers (7) Test Results

200 OK • 783 ms • 1.64 KB • Save Response

{ } JSON Visualize

| ManagedDeviceName | DeviceType         | NeverSignedIn | LastActiveTime       | CreatedDate            | DaysSinceLastSignIn | TotalUsageInHour | UserPrincipalName                      | IntuneDeviceId      | PolicyName               | ServicePlanId                      | CloudPCSize  | CloudPcid |
|-------------------|--------------------|---------------|----------------------|------------------------|---------------------|------------------|--|---------------------|--------------------------|------------------------------------|--|-----------|
| TST-test0-B5J4M   | Enterprise         | TRUE          |                      | 2025-12-05 12:07       |                     |                  | 0 test@ognitionittst.onmicrosoft.com   | 12345678-9012-3TST  | Test Cloud PC            | 2de9c682-1234-1234-1234567890      | Cloud PC Enterprise 4vCPU/16GB/128GE 64379843-123- |           |
| TST-test1-KP47C   | Enterprise         | FALSE         | 6/20/2025 2:37:02 PM | 2025-12-05 12:07       | 56                  |                  | 0 test1@cognitionittst.onmicrosoft.com | 12345678-9012-3 TST | Test Cloud PC            | 69dc175c-1234-1234-1234567890      | Cloud PC Enterprise 8vCPU/32GB/256GE cad55617-123- |           |
| DEV-test0-VEOL3   | Enterprise         | FALSE         | 7/16/2025 9:24:11 AM | 7/14/2023 9:37:07 AM   | 30                  |                  | 0 test0@cognitionitdev.onmicrosoft.com | 12345678-9012-3 DEV | Development Cloud PC     | 2de9c682-1234-1234-1234567890      | Cloud PC Enterprise 4vCPU/16GB/128GB 82b02dbd-123- |           |
| DEV-test2-ZSSX8   | FrontlineDedicated | FALSE         | 7/16/2025 5:55:31 PM | 2025-09-07 08:38       | 30                  |                  | 0 test0@cognitionitdev.onmicrosoft.com | 12345678-9012-3 DEV | Development Cloud PC FFL | 057ebfe-1234-1234-1234567890       | Cloud PC Frontline 2vCPU/8GB/128GB 6a4fc5d-1234-   |           |
| DEV-test2-FAUF8   | Enterprise         | FALSE         | 6/30/2025 7:09:21 AM | 2024-08-04 06:28       | 46                  |                  | 0 test2@cognitionitdev.onmicrosoft.com | 12345678-9012-3 DEV | Development Cloud PC     | 3effff3f-1234-1234-1234-1234567890 | Cloud PC Enterprise 2vCPU/8GB/128GB 25b0a8c9-123-  |           |
| DEV-test3-D1U44   | FrontlineDedicated | TRUE          |                      | 8/13/2025 8:08:52 AM   |                     |                  | 0 test3@cognitionitdev.onmicrosoft.com | 12345678-9012-3 DEV | Development Cloud PC FFL | 057ebfe-1234-1234-1234-1234567890  | Cloud PC Frontline 2vCPU/8GB/128GB cccb673-1234-   |           |
| PRD-test1-01387   | FrontlineDedicated | TRUE          |                      | 2025-06-05 07:45       |                     |                  | 0 test1@ cognitionit .com              | 12345678-9012-3 PRD | Production Cloud PC      | 057ebfe-1234-1234-1234567890       | Cloud PC Frontline 2vCPU/8GB/128GB a65f5965-1234-  |           |
| PRD-test2-0EU22   | FrontlineDedicated | FALSE         | 2025-11-07 12:57     | 11/14/2024 10:04:12 AM | 35                  |                  | 0 test2@cognitionit .com               | 12345678-9012-3 PRD | Production Cloud PC      | 057ebfe-1234-1234-1234-1234567890  | Cloud PC Frontline 2vCPU/8GB/128GB 1d044521-123-   |           |

# W365 CloudPC License Management (raw)



## cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

HTTP 00. GraphAPI Demos / Windo... / 03. License Management / [CloudPCReports: getTotalAggregatedRemoteConnectionReports](https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?$count=1...) Save Share

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?\$count=1... Send

Params ● Authorization Headers (12) Body ● Scripts Settings Cookies Beautify

Body raw binary GraphQL JSON

```
1 {  
2   "skip": 0,  
3   "select": [],  
4   "top": 1,  
5   "search": "",  
6   "orderBy": [  
7     "TotalUsageInHour"  
8   ]  
9 }
```

Body Cookies Headers (7) Test Results 200 OK 783 ms 1.64 KB Save Response

{ } JSON Visualize

| ManagedDeviceName | DeviceType         | NeverSignedIn | LastActiveTime       | CreatedDate            | DaysSinceLastSignIn | TotalUsageInHour | UserPrincipalName                    | IoTuneDeviceId      | PolicyName           | ServicePlanId                 | CloudPCSize                                       | CloudPcid |
|-------------------|--------------------|---------------|----------------------|------------------------|---------------------|------------------|--------------------------------------|---------------------|----------------------|-------------------------------|---|-----------|
| TST-test0-B5J4M   | Enterprise         | TRUE          |                      | 2025-12-05 12:07       |                     |                  | test0@cognitionittst.onmicrosoft.com | 1345678-9012-3TST   | Test Cloud PC        | 2de9c682-1234-1234-1234567890 | Cloud PC Enterprise 4vCPU/16GB/128GE 64379843-123 |           |
| TST-test1-KP47C   | Enterprise         | FALSE         | 6/20/2025 2:37:02 PM | 2025-12-05 12:07       | 56                  |                  | test1@cognitionittst.onmicrosoft.com | 1345678-9012-3TST   | Test Cloud PC        | 69dc175c-1234-1234-1234567890 | Cloud PC Enterprise 8vCPU/32GB/256GE cad55617-123 |           |
| DEV-test0-VEOL3   | Enterprise         | FALSE         | 7/16/2025 9:24:11 AM | 7/14/2023 9:37:07 AM   | 30                  |                  | test0@cognitionitdev.onmicrosoft.com | 1345678-9012-3 DEV  | Development Cloud PC | 20896082-1234-1234-1234567890 | Cloud PC Enterprise 4vCPU/16GB/128GB 82b02dbd-123 |           |
| DEV-test2-ZSSX8   | FrontlineDedicated | FALSE         | 7/16/2025 5:55:31 PM | 2025-09-07 08:38       | 30                  |                  | test0@cognitionitdev.onmicrosoft.com | 1345678-9012-3 DEV  | Development Cloud PC | 057e1bfe-1234-1234-1234567890 | Cloud Frontline 2vCPU/8GB/128GB 6a4fc5d-123       |           |
| DEV-test2-FAUF8   | Enterprise         | FALSE         | 6/30/2025 7:09:21 AM | 2024-08-04 06:28       | 46                  |                  | test2@cognitionitdev.onmicrosoft.com | 1345678-9012-3 DEV  | Development Cloud PC | 3eff3f6e-1234-1234-1234567890 | Cloud PC Enterprise 2vCPU/8GB/128GB 25b0a8c9-123  |           |
| DEV-test3-D1U44   | FrontlineDedicated | TRUE          |                      | 8/13/2025 8:08:52 AM   |                     |                  | test3@cognitionitdev.onmicrosoft.com | 1345678-9012-3 DEV  | Development Cloud PC | 057e1bfe-1234-1234-1234567890 | Cloud Frontline 2vCPU/8GB/128GB cccb673-1234      |           |
| PRD-test1-01387   | FrontlineDedicated | TRUE          |                      | 2025-06-05 07:45       |                     |                  | 0 test1@cognitionit.com              | 12345678-9012-3 PRD | Production Cloud PC  | 057e1bfe-1234-1234-1234567890 | Cloud Frontline 2vCPU/8GB/128GB a65f5965-1234     |           |
| PRD-test2-0EU22   | FrontlineDedicated | FALSE         | 2025-11-07 12:57     | 11/14/2024 10:04:12 AM | 35                  |                  | 0 test2@cognitionit.com              | 12345678-9012-3 PRD | Production Cloud PC  | 057e1bfe-1234-1234-1234567890 | Cloud Frontline 2vCPU/8GB/128GB 1d044521-123      |           |

# W365 CloudPC License Management (enriched)



cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

HTTP 00. GraphAPI Demos / Windo... / 03. License Management / CloudPcReports: getTotalAggregatedRemoteConnectionReports

Save Share

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?\$count=1...

Send

Params • Authorization Headers (12) Body • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON Cookies Beautify

```
1 {
2   "skip": 0,
3   "select": [],
4   "top": 1,
5   "search": "",
6   "orderBy": [
7     "TotalUsageInHour"
8   ]
9 }
```

Body Cookies Headers (7) Test Results

200 OK • 783 ms • 1.64 KB • Save Response

{ } JSON Visualize

| ManagedDeviceName | DeviceType         | NeverSignedIn | LastActiveTime       | CreatedDate            | DaysSinceLastSignIn | UserPrincipalName                      | contactEmail                        | ServicePlanId                                      | ServicePlanName                                      | provisioningPolicyGroupId | provisioningPolicyGroupName | ExpectedEnterpriseLicenseGroupName |
|-------------------|--------------------|---------------|----------------------|------------------------|---------------------|--|-------------------------------------|--|--|---------------------------|-----------------------------|------------------------------------|
| TST-test0-B5J4M   | Enterprise         | TRUE          |                      | 2025-12-05 12:07       |                     | test0@ognitionitst.onmicrosoft.com     | test0@ognitionitst.onmicrosoft.com  | 2de9c682-1234-Cloud PC Enterprise 4vCPU/16GB/128GB | 736b9fc8-1234-1234-W365-TST Test Cloud PC            |                           | W365-LIC-4vCPU-16GB-128GB   |                                    |
| TST-test1-KP47C   | Enterprise         | FALSE         | 6/20/2025 2:37:02 PM | 2025-12-05 12:07       |                     | 56 test1@ognitionitst.onmicrosoft.com  | test1@ognitionitst.onmicrosoft.com  | 69dc175c-1234-Cloud PC Enterprise 8vCPU/32GB/256GB | 736b9fc8-1234-1234-W365-TST Test Cloud PC            |                           | W365-LIC-8vCPU-32GB-256GB   |                                    |
| DEV-test1-VEOL3   | Enterprise         | FALSE         | 7/16/2025 9:24:11 AM | 7/14/2023 9:37:07 AM   |                     | 30 test0@ognitionitdev.onmicrosoft.com | test0@ognitionitdev.onmicrosoft.com | 2de9c682-1234-Cloud PC Enterprise 4vCPU/16GB/128GB | b8483202-1234-1234-W365-DEV Development Cloud PC     |                           | W365-LIC-4vCPU-16GB-128GB   |                                    |
| DEV-test1-ZSSX8   | FrontlineDedicated | FALSE         | 7/16/2025 5:55:31 PM | 2025-09-07 08:38       |                     | 30 test1@ognitionitdev.onmicrosoft.com | test1@ognitionitdev.onmicrosoft.com | 057efbfe-1234-Cloud PC Frontline 2vCPU/8GB/128GB   | a0249f00-1234-1234-W365-DEV Development Cloud PC FFL |                           | W365-LIC-2vCPU-8GB-128GB    |                                    |
| DEV-test2-FAUFB   | Enterprise         | FALSE         | 6/30/2025 7:09:21 AM | 2024-08-04 06:28       |                     | 46 test2@ognitionitdev.onmicrosoft.com | test2@ognitionitdev.onmicrosoft.com | 3eff3fe-1234-1Cloud PC Enterprise 2vCPU/8GB/128GB  | b8483202-1234-1234-W365-DEV Development Cloud PC     |                           | W365-LIC-2vCPU-8GB-128GB    |                                    |
| DEV-test3-D1U44   | FrontlineDedicated | TRUE          |                      | 8/13/2025 8:08:52 AM   |                     | test3@ognitionitdev.onmicrosoft.com    | test3@ognitionitdev.onmicrosoft.com | 057efbfe-1234-Cloud PC Frontline 2vCPU/8GB/128GB   | a0249f00-1234-1234-W365-DEV Development Cloud PC FFL |                           | W365-LIC-2vCPU-8GB-128GB    |                                    |
| PRD-test1-01387   | FrontlineDedicated | TRUE          |                      | 2025-06-05 07:45       |                     | test1@ognitionit.com                   | test1@ognitionit.com                | 057efbfe-1234-Cloud PC Frontline 2vCPU/8GB/128GB   | db320d66-1234-1234-W365-PRD Production Cloud PC      |                           | W365-LIC-2vCPU-8GB-128GB    |                                    |
| PRD-test2-EU022   | FrontlineDedicated | FALSE         | 2025-11-07 12:57     | 11/14/2024 10:04:12 AM |                     | 35 test2@ognitionit.com                | test2@ognitionit.com                | 057efbfe-1234-Cloud PC Frontline 2vCPU/8GB/128GB   | db320d66-1234-1234-W365-PRD Production Cloud PC      |                           | W365-LIC-2vCPU-8GB-128GB    |                                    |

# W365 CloudPC License Management (filtered)



cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

HTTP 00. GraphAPI Demos / Windo... / 03. License Management / [CloudPCReports: getTotalAggregatedRemoteConnectionReports](https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?$count=1...) Save Share

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/reports/getTotalAggregatedRemoteConnectionReports/?\$count=1... Send

Params ● Authorization Headers (12) Body ● Scripts Settings Cookies Beautify

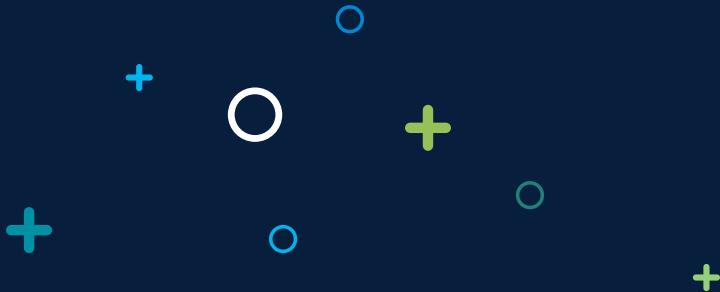
Body ● none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
1 {
2   "skip": 0,
3   "select": [],
4   "top": 1,
5   "search": "",
6   "orderBy": [
7     "TotalUsageInHour"
8   ]
9 }
```

Body Cookies Headers (7) Test Results 200 OK • 783 ms • 1.64 KB • Save Response ...

{ } JSON Visualize

| ManagedDeviceName | DeviceType         | NeverSignedin | LastActiveTime       | CreatedDate           | DaysSinceLastSignIn | UserPrincipalName                   | contactEmail          | ProvisioningPolicyName      | provisioningPolicyGroupName      | ExpectedEnterpriseLicenseGroupName | ServicePlanType |
|-------------------|--------------------|---------------|----------------------|-----------------------|---------------------|-------------------------------------|-----------------------|-----------------------------|----------------------------------|------------------------------------|-----------------|
| TST-test0-B5JM4   | Enterprise         | TRUE          | 2025-11-06 15:37     | 1/18/2024 2:18:49 PM  | 65                  | test0@cognitionitst.onmicrosoft.com | TST Test Cloud PC     |                             | W365-TST Test Cloud PC           | W365-LIC-4vCPU-16GB-128GB          | enterprise      |
| TST-test1-KP47C   | Enterprise         | FALSE         |                      | 2025-11-06 08:09      |                     | est1@cognitionitst.onmicrosoft.com  | TST Test Cloud PC     |                             | W365-TST Test Cloud PC           | W365-LIC-8vCPU-32GB-256GB          | enterprise      |
| DEV-test0-VEOL3   | Enterprise         | FALSE         | 2025-10-06 08:38     | 1/23/2025 1:10:08 PM  | 66                  | est0@cognitionitdev.onmicrosoft.com | test0@cognitionit.com | DEV Development Cloud PC    | W365-DEV Development Cloud PC    | W365-LIC-4vCPU-16GB-128GB          | enterprise      |
| DEV-test1-ZSSX8   | FrontlineDedicated | FALSE         |                      | 2025-10-06 14:37      |                     | est1@cognitionitdev.onmicrosoft.com | test1@cognitionit.com | DEV Development Cloud PC FL | W365-DEV Development Cloud PC FL | W365-LIC-4vCPU-16GB-128GB          | enterprise      |
| DEV-test2-FAUF8   | Enterprise         | FALSE         | 6/13/2025 1:26:57 PM | 3/21/2025 10:35:49 AM | 63                  | est2@cognitionitdev.onmicrosoft.com | test2@cognitionit.com | DEV Development Cloud PC    | W365-DEV Development Cloud PC    | W365-LIC-2vCPU-8GB-128GB           | enterprise      |



# W365 License Management PowerShell script challenges



# W365 CloudPC License Management

Get-W365CPC-InactiveFor60Days-Report-Data

Retrieve CloudPCReports:  
getTotalAggregatedRemoteConnectionReports

Enrich report data for  
readability & follow-up actions

Filter report data on  
CloudPC Types &  
60 days inactivity

# W365 CloudPC License Management



## Challenges:

- response data pagination
- response formatting (PSObject)

# Graph API PowerShell challenge (GET pagination)

Paging Microsoft Graph data in your app - Microsoft Graph | Microsoft Learn

## Paging Microsoft Graph data in your app

04/03/2025

Paging involves requesting or receiving data in batches. It's a performance technique that's crucial for efficiently handling large datasets and that helps improve the performance of your app and the response time of Microsoft Graph.

Some GET queries against Microsoft Graph return multiple pages of data either due to server-side paging or client-side paging. In this article, we explore how paging works for Microsoft Graph and how you can use it to optimize your applications.

Body Cookies Headers (10) Test Results ⌂

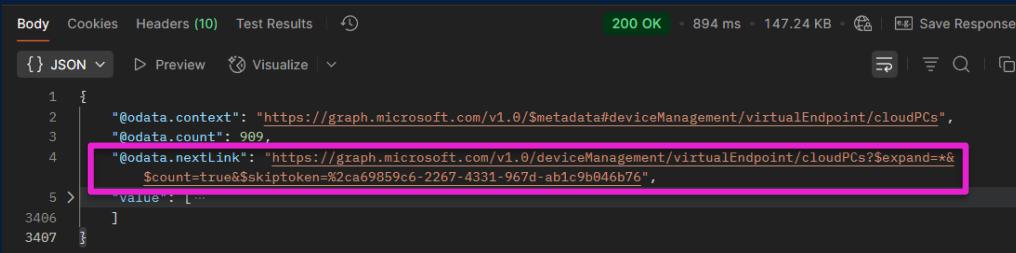
200 OK • 894 ms • 147.24 KB • ⓘ | ⏺ Save Response

{ } JSON ▾ ▷ Preview ⏺ Visualize ▾

```
1 {  
2   "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#deviceManagement/virtualEndpoint/cloudPCs",  
3   "@odata.count": 909,  
4   "@odata.nextLink": "https://graph.microsoft.com/v1.0/deviceManagement/virtualEndpoint/cloudPCs?$expand=&$count=true&$skiptoken=%2ca69859c6-2267-4331-967d-ab1c9b046b76",  
5   "value": [  
3406 ]  
3407 }
```

# Graph API PowerShell challenge (GET pagination)

Paging Microsoft Graph data in your app - Microsoft Graph | Microsoft Learn



Body Cookies Headers (10) Test Results | 200 OK • 894 ms • 147.24 KB • Save Response

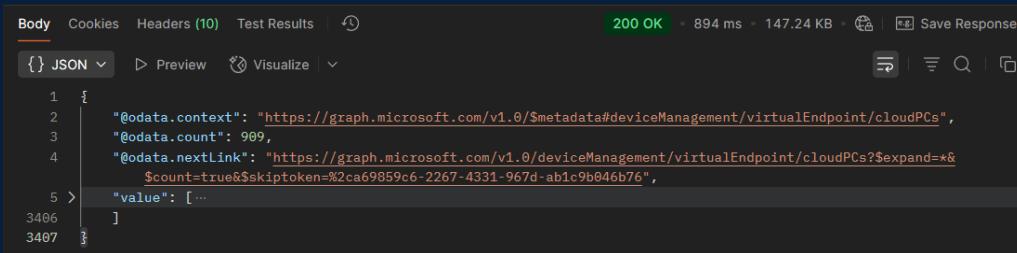
{ } JSON ▾ ▷ Preview ⚡ Visualize | ↻ 🔍 🔍 🔍 🔍

```
1 {  
2   "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#deviceManagement/virtualEndpoint/cloudPCs",  
3   "@odata.count": 909,  
4   "@odata.nextLink": "https://graph.microsoft.com/v1.0/deviceManagement/virtualEndpoint/cloudPCs?$expand=&count=true&$skiptoken=%2ca69859c6-2267-4331-967d-ab1c9b046b76",  
5   "value": [  
3406 ]  
3407 ]
```

```
# Make the web request and convert result content to PSObject to retrieve the Windows365 CloudPCs  
$webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)  
  
# Create an array to store all CloudPCs  
$allCloudPCs = @()  
$allCloudPCs += $webrequestResult.value  
  
# Resolve Graph API pagination to retrieve all CloudPCs  
while ($null -ne $webrequestResult.'@odata.nextLink') {  
    $uri = $webrequestResult.'@odata.nextLink'  
    $webRequestParams = @{  
        Uri      = $uri  
        Headers = $authHeader  
        Method   = $method  
    }  
    $webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)  
    # Store next page of CloudPCs in the allCloudPCs array  
    $allCloudPCs += $($webrequestResult.value)  
}  
# Return the array of all CloudPCs  
return $allCloudPCs
```

# Graph API PowerShell challenge (GET pagination)

Paging Microsoft Graph data in your app - Microsoft Graph | Microsoft Learn



Body Cookies Headers (10) Test Results | 200 OK • 894 ms • 147.24 KB •

{ } JSON ▾ ▷ Preview ⚡ Visualize | ↻ 🔍

```
1 {  
2   "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#deviceManagement/virtualEndpoint/cloudPCs",  
3   "@odata.count": 909,  
4   "@odata.nextLink": "https://graph.microsoft.com/v1.0/deviceManagement/virtualEndpoint/cloudPCs?$expand=&$count=true&$skiptoken=%2ca69859c6-2267-4331-967d-ab1c9b046b76",  
5   "value": [...]  
3406 }  
3407 }
```

```
# Make the web request and convert result content to PSObject to retrieve the Windows365 CloudPCs  
$webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)  
  
# Create an array to store all CloudPCs  
$allCloudPCs = @()  
$allCloudPCs += $webrequestResult.value  
  
# Resolve Graph API pagination to retrieve all CloudPCs  
while ($null -ne $webrequestResult.'@odata.nextLink') {  
    $uri = $webrequestResult.'@odata.nextLink'  
    $webRequestParams = @{  
        Uri      = $uri  
        Headers = $authHeader  
        Method   = $method  
    }  
    $webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)  
    # Store next page of CloudPCs in the allCloudPCs array  
    $allCloudPCs += $($webrequestResult.value)  
}  
# Return the array of all CloudPCs  
return $allCloudPCs
```

# Graph API PowerShell challenge (POST pagination)

Body Cookies Headers (7) Test Results | ⏱

{ } JSON ▾ ⚡ Visualize ▾

```
1  {
2      "TotalRowCount": 906,
3      "Schema": [...]
4      ],
5      "Values": [...]
6  ]
7 }
```



NO @odata.nextLink !!!!!

# Graph API PowerShell challenge (POST pagination)



Body Cookies Headers (7) Test Results | ⏱

{ } JSON ▾ Visualize ▾

```
1  {
2    "TotalRowCount": 906,
3    "Schema": [
4      ...
5    ],
6    "Values": [
7      ...
8    ]
9  }
```



NO @odata.nextLink !!!!!

Paging data from the Microsoft Graph API using the `POST` method—especially for endpoints like **Cloud PC reports**—requires a slightly different approach than traditional `GET` requests. Here's how you can handle pagination using the `skip` parameter in this context:

## Scenario: Paging Cloud PC Reports via POST

Microsoft Graph's reporting endpoints (like `/reports/cloudPCPerformance`, `/reports/cloudPCConnection`, etc.) often use `POST` for complex queries. These endpoints typically support pagination via the `skip` parameter in the request body.

### 3. Loop Through Pages

To get all data, loop through the pages by increasing `skip`:

Python ^ Copy

```
skip = 0
top = 100
results = []

while True:
    response = post_to_graph_api(skip=skip, top=top)
    data = response["value"]
    results.extend(data)

    if len(data) < top:
        break # No more data
    skip += top
```

# Graph API PowerShell challenge (POST pagination)



Body Cookies Headers (7) Test Results

{ } JSON Visualize

```
1  {
2    "TotalRowCount": 906,
3  >   "Schema": [...]
4  >     ],
5  >   "Values": [...]
6  >     ]
7  >   }
8  > }
```

```
# Create an array to store all AggregatedRemoteConnections
[array]$rawAggregatedRemoteConnections = @()

do {
    # Make the web request to retrieve the Total Aggregated Remote Connection Reports
    $webRequestParams = @{
        Uri      = $uri
        Headers = $authHeader
        Method   = $method
        Body     = $(ConvertTo-Json -InputObject $bodyObject -Depth 100)
    }

    # Use Invoke-RestMethod to handle base64-encoded webrequest.content
    $webrequestResults = Invoke-RestMethod @webRequestParams

    # Process the result of the web request
    [int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
    foreach ($item in $webrequestResults.Values) {
        $aggregatedRemoteConnection = @{}
        for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
            $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
        }
        $rawAggregatedRemoteConnections += $aggregatedRemoteConnection
    }

    # Update skip and check for pagination
    $retrievedRows = ($webrequestResults.Values | Measure-Object).Count
    $skip += $retrievedRows
    $totalRowCount = $webrequestResults.TotalRowCount

    # Update the request body for the next page
    $bodyObject["skip"] = $skip
} while ($skip -lt $totalRowCount)

# Return the array of all Aggregated Remote Connections
return $rawAggregatedRemoteConnections
```

# Graph API PowerShell challenge (POST pagination)



Body Cookies Headers (7) Test Results

{ } JSON Visualize

```
1  {
2    "TotalRowCount": 906,
3  >   "Schema": [...]
4  >     ],
5  >   "Values": [...]
6  >     ]
7  >   }
8  > }
```

```
# Create an array to store all AggregatedRemoteConnections
[array]$rawAggregatedRemoteConnections = @()

do {
    # Make the web request to retrieve the Total Aggregated Remote Connection Reports
    $webRequestParams = @{
        Uri      = $uri
        Headers = $authHeader
        Method   = $method
        Body     = $(ConvertTo-Json -InputObject $bodyObject -Depth 100)
    }

    # Use Invoke-RestMethod to handle base64-encoded webrequest.content
    $webrequestResults = Invoke-RestMethod @webRequestParams

    # Process the result of the web request
    [int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
    foreach ($item in $webrequestResults.Values) {
        $aggregatedRemoteConnection = @{}
        for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
            $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
        }
        $rawAggregatedRemoteConnections += $aggregatedRemoteConnection
    }
}

# Update skip and check for pagination
$retrievedRows = ($webrequestResults.Values | Measure-Object).Count
$skip += $retrievedRows
$totalRowCount = $webrequestResults.TotalRowCount

# Update the request body for the next page
$bodyObject["skip"] = $skip
} while ($skip -lt $totalRowCount)

# Return the array of all Aggregated Remote Connections
return $rawAggregatedRemoteConnections
```

# Graph API PowerShell challenge (response -> PSObject)

List cloudPCs - Microsoft Graph beta | Microsoft Learn

Body Cookies Headers (10) Test Results ↻

200 OK • 882 ms • 1.54 KB • ⏺ Save Response ⋮

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

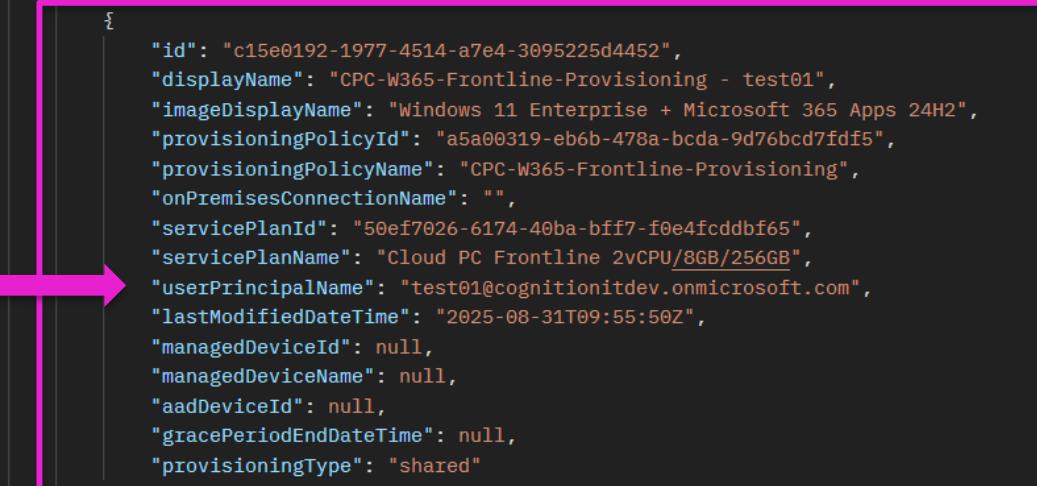
200 OK • 882 ms • 1.54 KB • ⏺ Save Response ⋮

200 OK • 882 ms • 1.54 KB • ⏺ Save Response ⋮

200 OK • 882 ms • 1.54 KB • ⏺ Save Response ⋮

200 OK • 882 ms • 1.54 KB • ⏺ Save Response ⋮

```
1 {  
2     "@odata.context": "https://graph.microsoft.com/v1.0/\$metadata#deviceManagement/virtualEndpoint/cloudPCs",  
3     "@odata.count": 2,  
4     "@odata.nextLink": "https://graph.microsoft.com/v1.0/deviceManagement/virtualEndpoint/cloudPCs?\$expand=\*&\$count=true&\$filter=startswith\(userPrincipalName%2c+%27test01%27\)&\$top=1&\$skiptoken=%2c1c60d512-df61-4e6a-9e11-3088259bf376",  
5     "value": [  
6         {  
7             "id": "c15e0192-1977-4514-a7e4-3095225d4452",  
8             "displayName": "CPC-W365-Frontline-Provisioning - test01",  
9             "imageDisplayName": "Windows 11 Enterprise + Microsoft 365 Apps 24H2",  
10            "provisioningPolicyId": "a5a00319-eb6b-478a-bcda-9d76bcd7fdf5",  
11            "provisioningPolicyName": "CPC-W365-Frontline-Provisioning",  
12            "onPremisesConnectionName": "",  
13            "servicePlanId": "50ef7026-6174-40ba-bff7-f0e4fcddbf65",  
14            "servicePlanName": "Cloud PC Frontline 2vCPU/8GB/256GB",  
15            "userPrincipalName": "test01@cognitionitdev.onmicrosoft.com",  
16            "lastModifiedDateTime": "2025-08-31T09:55:50Z",  
17            "managedDeviceId": null,  
18            "managedDeviceName": null,  
19            "aadDeviceId": null,  
20            "gracePeriodEndDateTime": null,  
21            "provisioningType": "shared"  
22        }  
23    ]  
24 }
```



# Graph API PowerShell challenge (response -> PSObject)

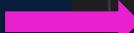
[List cloudPCs - Microsoft Graph beta | Microsoft Learn](#)

```
# Make the web request and convert result content to PSObject to retrieve the Windows365 CloudPCs
$webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)

# Create an array to store all CloudPCs
$allCloudPCs = @()
$allCloudPCs += $webrequestResult.value

# Resolve Graph API pagination to retrieve all CloudPCs
while ($null -ne $webrequestResult.'@odata.nextLink') {
    $uri = $webrequestResult.'@odata.nextLink'
    $webRequestParams = @{
        ...
    }
    $webrequestResult = ConvertFrom-Json -InputObject $($($Invoke-WebRequest @webRequestParams).Content)
    # Store next page of CloudPCs in the allCloudPCs array
    $allCloudPCs += $($webrequestResult.value)
}

# Return the array of all CloudPCs
return $allCloudPCs
```



# Graph API PowerShell challenge (response -> PSObject)

List cloudPCs - Microsoft Graph beta | Microsoft Learn

```
> W365ilicenseReport.ps1 ×  
C: > Temp > > W365ilicenseReport.ps1 > ...  
591   $w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)  
592
```

PROBLEMS TERMINAL PORTS GITLENS AZURE OUTPUT DEBUG CONSOLE

● PS C:\Temp> \$w365CloudPCs[0]

```
id : c15e0192-1977-4514-a7e4-3095225d4452  
displayName : CPC-W365-Frontline-Provisioning - test01  
imageDisplayName : Windows 11 Enterprise + Microsoft 365 Apps 24H2  
provisioningPolicyId : a5a00319-eb6b-478a-bcda-9d76bcd7fdf5  
provisioningPolicyName : CPC-W365-Frontline-Provisioning  
onPremisesConnectionName :  
servicePlanId : 50ef7026-6174-40ba-bff7-f0e4fcddbf65  
servicePlanName : Cloud PC Frontline 2vCPU/8GB/256GB  
status : provisioning  
userPrincipalName : test01@cognitionitdev.onmicrosoft.com  
lastModifiedDateTime : 2025-08-31 09:53:38  
managedDeviceId :  
managedDeviceName :  
aadDeviceId :  
gracePeriodEndDateTime :  
servicePlanType : enterprise  
diskEncryptionState : notAvailable  
provisioningType : sharedByUser  
statusDetails :  
statusDetail :  
partnerAgentInstallResults : {}
```

PS C:\Temp>

# Graph API PowerShell challenge (response -> PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

Body Cookies Headers (7) Test Results ⏱

{ } JSON Visualize ▾

```
1 {  
2     "TotalRowCount": 2,  
3     "Schema": [  
4         {  
5             "Column": "CloudPcId",  
6             "PropertyName": "String"  
7         },  
8         {  
9             "Column": "ManagedDeviceName",  
10            "PropertyName": "String"  
11        },  
12        {  
13            "Column": "UserPrincipalName",  
14            "PropertyName": "String"  
15        },  
16        {  
17            "Column": "DaysSinceLastSignIn",  
18            "PropertyName": "Int64"  
19        },  
20        {  
21            "Column": "TotalUsageInHour",  
22            "PropertyName": "Double"  
23        },  
24    ]  
25 }  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106 }
```

Body Cookies Headers (7) Test Results ⏱

{ } JSON Visualize ▾

```
1 {  
2     "TotalRowCount": 2,  
3     "Schema": [  
4         {  
5             "Column": "CloudPcId",  
6             "PropertyName": "String"  
7         },  
8         {  
9             "Column": "ManagedDeviceName",  
10            "PropertyName": "String"  
11        },  
12        {  
13            "Column": "UserPrincipalName",  
14            "PropertyName": "String"  
15        },  
16        {  
17            "Column": "DaysSinceLastSignIn",  
18            "PropertyName": "Int64"  
19        },  
20        {  
21            "Column": "TotalUsageInHour",  
22            "PropertyName": "Double"  
23        },  
24    ]  
25 }  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
69  
70 }  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106 ]
```

# Graph API PowerShell challenge (response -> PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

```
# Create an array to store all AggregatedRemoteConnections
[array]$rawAggregatedRemoteConnections = @()

do {
    # Make the web request to retrieve the Total Aggregated Remote Connection Reports
    $webRequestParams = @{
        ...
    }

    # Use Invoke-RestMethod to handle base64-encoded webrequest.content
    $webrequestResults = Invoke-RestMethod @webRequestParams

    # Process the result of the web request
    [int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
    foreach ($item in $webrequestResults.Values) {
        $aggregatedRemoteConnection = {}
        for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
            $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
        }
        $rawAggregatedRemoteConnections += $aggregatedRemoteConnection
    }
} while ($skip -lt $totalCount)

# Return the array of all Aggregated Remote Connections
return $rawAggregatedRemoteConnections
```

# Graph API PowerShell challenge (response -> PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

```
# Create an array to store all AggregatedRemoteConnections
[array]$rawAggregatedRemoteConnections = @()

do {
    # Make the web request to retrieve the Total Aggregated Remote Connection Reports
    $webRequestParams = @{
        ...
    }

    # Use Invoke-RestMethod to handle base64-encoded webrequest.content
    $webrequestResults = Invoke-RestMethod @webRequestParams

    # Process the result of the web request
    [int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
    foreach ($item in $webrequestResults.Values) {
        $aggregatedRemoteConnection = @{}
        for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
            $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
        }
        $rawAggregatedRemoteConnections += $aggregatedRemoteConnection
    }
} while ($skip -lt $totalCount)

# Return the array of all Aggregated Remote Connections
return $rawAggregatedRemoteConnections
```

```
"Schema": [
    {
        "Column": "CloudPcId",
        "PropertyType": "String"
    },
    {
        "Column": "ManagedDeviceName",
        "PropertyType": "String"
    }
]
```

```
"Values": [
    [...],
    [
        ...
    ],
    [
        ...
    ]
]
```

```
[1]   "627a21fe-a554-47ca-9a3c-f14fe8c3e99b",
      "CPC-test0-N61MG",
```

# Graph API PowerShell challenge (response -> PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
C: > Temp > ➜ W365licenseReport.ps1 > ...
607 # Get Intune report data for Total Aggregated Remote Connections
608 $rawAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReport -AccessToken $($graphContext.access_token)
609 $enrichedAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReportWithEnrichedData -AccessToken $($graphContext.access_token) `-
610                                         -ServicePlanSearchable $w365ServicePlanSearchable -LicenseGroupSearchable $w365LicenseGroupsSearchTable `-
611                                         -CloudPcSearchable $w365CloudPCs -ProvisioningPolicySearchable $w365ProvisioningPoliciesSearchTable
612
```

PROBLEMS TERMINAL PORTS GITLENS AZURE OUTPUT DEBUG CONSOLE

- PS C:\Temp> \$rawAggregatedRemoteConnections[1] | ConvertTo-Json -Depth 10 | ConvertFrom-Json

```
LastActiveTime      :
CreatedDate        : 2025-08-27 20:37:09
ManagedDeviceName  : CPC-test0-N61MG
ServicePlanId      : 50ef7026-6174-40ba-bff7-f0e4fcddbf65
UserPrincipalName  : test01@cognitionitdev.onmicrosoft.com
CloudPcId          : 627a21fe-a554-47ca-9a3c-f14fe8c3e99b
DeviceType         : FrontlineDedicated
CloudPcSize        : Cloud PC Frontline 2vCPU/8GB/256GB
IntuneDeviceId    : 645f6084-98ba-4ed4-9299-817ed164860d
PcType             : Cloud PC Frontline 2vCPU/8GB/256GB
DaysSinceLastSignIn:
PolicyName         : CPC-W365-Frontline-Provisioning
NeverSignedIn      : True
HostRegion         : northeurope
TotalUsageInHour   : 0
```

PS C:\Temp>

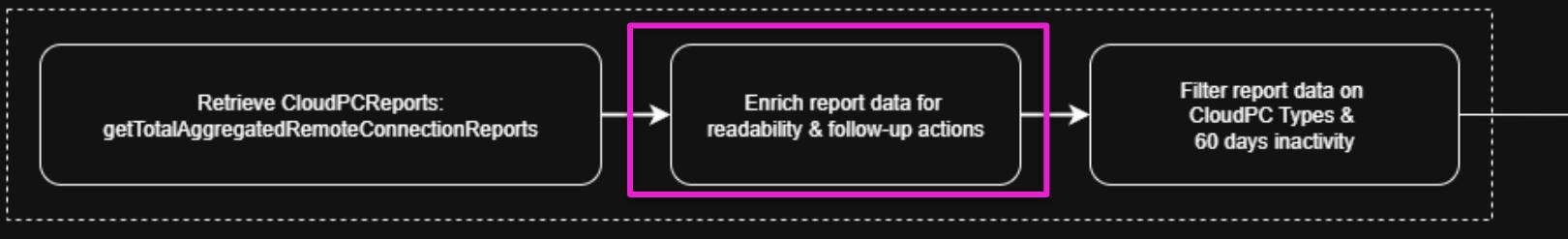
# W365 CloudPC License Management

Get-W365CPC-InactiveFor60Days-Report-Data

Retrieve CloudPCReports:  
getTotalAggregatedRemoteConnectionReports

Enrich report data for  
readability & follow-up actions

Filter report data on  
CloudPC Types &  
60 days inactivity



# W365 CloudPC License Management



## Challenges:

- add data to PSObject
- optimize additional API calls

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

```
# Create an array to store all AggregatedRemoteConnections
[array]$rawAggregatedRemoteConnections = @()

do {
    # Make the web request to retrieve the Total Aggregated Remote Connection Reports
    $webRequestParams = @{
        ...
    }

    # Use Invoke-RestMethod to handle base64-encoded webrequest.content
    $webrequestResults = Invoke-RestMethod @webRequestParams

    # Process the result of the web request
    [int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
    foreach ($item in $webrequestResults.Values) {
        $aggregatedRemoteConnection = @{}
        for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
            $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
        }
        $rawAggregatedRemoteConnections += $aggregatedRemoteConnection
    }
} while ($skip -lt $totalCount)

# Return the array of all Aggregated Remote Connections
return $rawAggregatedRemoteConnections
```

# Graph API PowerShell challenge (enrich PSObject)

cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

```
# Process the result of the web request
[int]$totalColumns = $webrequestResults.Schema | Measure-Object).Count
foreach ($item in $webrequestResults.Values) {
    $aggregatedRemoteConnection = {}
    for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
        $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])
    }

    # Add ServicePlan details
    if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {
        $servicePlan = $servicePlanSearchable.where({ $_.id -eq $item[$i] })
        $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)
        $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)
        # Add ServicePlanProvisioningType and ExpectedEnterpriseLicenseGroupName
        switch ($servicePlan.provisioningType) {
            'shared' {
                $servicePlanProvisioningType = 'Frontline'
                $enterpriseLicensedroupName = ''
            }
            'dedicated' {
                $servicePlanProvisioningType = 'Enterprise'
                $enterpriseLicensedroupName = "W365-LIC-$($servicePlan.vCpuCount)vCPU"
                $enterpriseLicensedroupName += "$($servicePlan.ramInGB)GB"
                $enterpriseLicensedroupName += "$($servicePlan.storageInGB)GB"
            }
            default {
                $servicePlanProvisioningType = 'Unknown'
                $enterpriseLicensedroupName = ''
            }
        }
        $aggregatedRemoteConnection.Add('ServicePlanProvisioningType', $servicePlanProvisioningType)
        $aggregatedRemoteConnection.Add('ExpectedEnterpriseLicenseGroupName', $enterpriseLicensedroupName)
    }
    # search EnterpriseLicenseGroupId in $w365LicenseGroupsSearchTable
    $licenseGroup = $licenseGroupSearchable.where({ $_.displayname -eq $enterpriseLicensedroupName })
    $aggregatedRemoteConnection.Add('ExpectedEnterpriseLicenseGroupId', $($licenseGroup.id))
}

# Add CloudPC details
if ($webrequestResults.Schema[$i].Column -eq 'CloudPcId') {
    $cloudPc = $cloudPcSearchable.where({ $_.id -eq $item[$i] })
    $aggregatedRemoteConnection.Add('CloudCName', $cloudPc.displayName)
    $aggregatedRemoteConnection.Add('ProvisioningPolicyId', $cloudPc.provisioningPolicyId)
    $aggregatedRemoteConnection.Add('ProvisioningPolicyName', $cloudPc.provisioningPolicyName)
    $aggregatedRemoteConnection.Add('ProvisioningType', $cloudPc.provisioningType)
    $aggregatedRemoteConnection.Add('LastModifiedDateTime', $cloudPc.lastModifiedDateTime)
    $aggregatedRemoteConnection.Add('provisioningStatus', $cloudPc.status)
    # Add provisioning policy groupname & groupid to dataset
    $assignmentServicePlanId = $cloudPc.servicePlanId
    $assignmentProvisioningPolicyId = $cloudPc.provisioningPolicyId
    $provisioningPolicy = $provisioningPolicySearchable.where({ $_.id -eq $assignmentProvisioningPolicyId })
    -and ( $_.assignmentServicePlan_id -eq $assignmentServicePlanId ) -or ($null -eq $_.assignmentServicePlanId) )
    $aggregatedRemoteConnection.Add('provisioningPolicyId', $provisioningPolicy.id)
    $aggregatedRemoteConnection.Add('provisioningPolicyName', $provisioningPolicy.name)
    $aggregatedRemoteConnection.Add('assignmentGroup', $provisioningPolicy.assignmentGroupName)
    $aggregatedRemoteConnection.Add('aadDeviceId', $cloudPc.aadDeviceId)
    # Convert user ID to contact email address
    $userPrincipalName = $cloudPc.userPrincipalName
    if ($userPrincipalName.endsWith('cognitionitdev.onmicrosoft.com')) {
        $userPrincipalName = $userPrincipalName.Replace('cognitionitdev.onmicrosoft.com', 'cognitionit.com')
    }
    if ($userPrincipalName.endsWith('cognitionitst.onmicrosoft.com')) {
        $userPrincipalName = $userPrincipalName.Replace('cognitionitst.onmicrosoft.com', 'cognitionit.com')
    }
    $contactEmail = $userPrincipalName
    $aggregatedRemoteConnection.Add('contactEmail', $contactEmail)
}

$allAggregatedRemoteConnections += $aggregatedRemoteConnection
}
```

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Process the result of the web request
[int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
foreach ($item in $webrequestResults.Values) {
    $aggregatedRemoteConnection = @{}
    for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
        $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])

        # Add ServicePlan details
        if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {
            $servicePlan = $ServicePlanSearchtable.where({ $_.id -eq $item[$i] })
            $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)
            $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)
```

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Process the result of the web request
[int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
foreach ($item in $webrequestResults.Values) {
    $aggregatedRemoteConnection = @{}
    for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
        $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])

        # Add ServicePlan details
        if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {
            $servicePlan = $ServicePlanSearchable.where({ $_.id -eq $item[$i] })
            $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)
            $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)
```

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Process the result of the web request
[int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
foreach ($item in $webrequestResults.Values) {
    $aggregatedRemoteConnection = @{}
    for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
        $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])

        # Add ServicePlan details
        if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {
            $servicePlan = $ServicePlanSearchtable.where({ $_.id -eq $item[$i] })
            $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)
            $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)
```

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Process the result of the web request
[int]$totalColumns = ($webrequestResults.Schema | Measure-Object).Count
foreach ($item in $webrequestResults.Values) {
    $aggregatedRemoteConnection = @{}
    for ($i = 0; $i -lt ($totalColumns - 1); $i++) {
        $aggregatedRemoteConnection.Add($webrequestResults.Schema[$i].Column, $item[$i])

        # Add ServicePlan details
        if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {
            $servicePlan = $ServicePlanSearchtable.where({ $_.id -eq $item[$i] })
            $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)
            $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)
```



```
CloudPcId      : 627a21fe-a554-47ca-9a3c-f14fe8c3e99b
ManagedDeviceName : CPC-test0-N61MG
UserPrincipalName : test01@cognitionitdev.onmicrosoft.com
ServicePlanId   : 50ef7026-6174-40ba-bff7-f0e4fcddbf65
ServicePlanName : Cloud PC Frontline 2vCPU/8GB/256GB
ServicePlanType : enterprise
```

# Graph API PowerShell challenge (enrich PSObject)

[cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn](#)

```
# Add ServicePlanProvisioningType and ExpectedEnterpriseLicenseGroupName
switch ($servicePlan.provisioningType) {
    'shared' {
        $ServicePlanProvisioningType = 'Frontline'
        $EnterpriseLicenseGroupName = ''
    }
    'dedicated' {
        $ServicePlanProvisioningType = 'Enterprise'
        $EnterpriseLicenseGroupName = "W365-LIC-$($servicePlan.vCPUCount)vCPU-"
        $EnterpriseLicenseGroupName += $($servicePlan.ramInGB)GB-
        $EnterpriseLicenseGroupName += $($servicePlan.storageInGB)GB"
    }
    default {
        $ServicePlanProvisioningType = 'Unknown'
        $EnterpriseLicenseGroupName = ''
    }
}
$aggregatedRemoteConnection.Add('ServicePlanProvisioningType', $ServicePlanProvisioningType)
$aggregatedRemoteConnection.Add('ExpectedEnterpriseLicenseGroupName', $EnterpriseLicenseGroupName)
#_search EnterpriseLicenseGroupId in #w365LicenseGroupsSearchTable
$licenseGroup = $LicenseGroupSearchable.where({ $_.displayName -eq $EnterpriseLicenseGroupName })
$aggregatedRemoteConnection.Add('ExpectedEnterpriseLicenseGroupId', $($licenseGroup.id))
}
```

CloudPcId : 627a21fe-a554-47ca-9a3c-f14fe8c3e99b  
ManagedDeviceName : CPC-test0-N61MG  
UserPrincipalName : test01@cognitionitdev.onmicrosoft.com  
ServicePlanProvisioningType : Enterprise  
ExpectedEnterpriseLicenseGroupName : W365-LIC-2vCPU-8GB-256GB  
ExpectedEnterpriseLicenseGroupId : 12345678-1234-1234-1234-123456789012

# Graph API PowerShell challenge (enrich PSObject)

cloudPcReports: getTotalAggregatedRemoteConnectionReports - Microsoft Graph beta | Microsoft Learn

```
# Add CloudPC details
if ($webrequestResults.Schema[$i].Column -eq 'CloudPcId') {
    $cloudPC = $CloudPcSearchable.where({ $_.id -eq $item[$i] })
    $aggregatedRemoteConnection.Add('CloudPCName', $CloudPC.displayName)
    $aggregatedRemoteConnection.Add('ProvisioningPolicyId', $CloudPC.provisioningPolicyId)
    $aggregatedRemoteConnection.Add('ProvisioningPolicyName', $CloudPC.provisioningPolicyName)
    $aggregatedRemoteConnection.Add('provisioningType', $CloudPC.provisioningType)
    $aggregatedRemoteConnection.Add('lastModifiedDateTime', $CloudPC.lastModifiedDateTime)
    $aggregatedRemoteConnection.Add('provisioningStatus', $CloudPC.status)
    # Add provisioning policy groupname & groupid to dataset
    $assignmentServicePlanId = $CloudPC.servicePlanId
    $assignmentProvisioningPolicyId = $CloudPC.provisioningPolicyId
    $provisioningPolicy = $ProvisioningPolicySearchable.where({ $_.id -eq $assignmentProvisioningPolicyId } ` 
        -and ( $_.assignmentServicePlanId -eq $assignmentServicePlanId ) -or ( $null -eq $_.assignmentServicePlanId ) )
    $aggregatedRemoteConnection.Add('provisioningPolicyGroupId', $provisioningPolicy.assignmentGroupId)
    $aggregatedRemoteConnection.Add('provisioningPolicyGroupName', $provisioningPolicy.assignmentGroupName)
    $aggregatedRemoteConnection.Add('aadDeviceId', $CloudPC.aadDeviceId)
    # Convert UPN to contact email address
    $userPrincipalName = $CloudPC.userPrincipalName
    if ($userPrincipalName.endsWith('cognitionitdev.onmicrosoft.com')) {
        $userPrincipalName = $userPrincipalName.Replace('cognitionitdev.onmicrosoft.com', 'cognitionit.com')
    }
    if ($userPrincipalName.endsWith('cognitionittst.onmicrosoft.com')) {
        $userPrincipalName = $userPrincipalName.Replace('cognitionittst.onmicrosoft.com', 'cognitionit.com')
    }
    $contactEmail = $userPrincipalName
    $aggregatedRemoteConnection.Add('contactEmail', $contactEmail)
}
```



|                             |   |  |
|-----------------------------|---|--|
| CloudPcId                   | : | 627a21fe-a554-47ca-9a3c-f14fe8c3e99b     |
| CloudPCName                 | : | CPC-W365-Frontline-Provisioning - test01 |
| ProvisioningPolicyId        | : | a5a00319-eb6b-478a-bcda-9d76bcd7fdf5     |
| ProvisioningPolicyName      | : | CPC-W365-Frontline-Provisioning          |
| provisioningType            | : | sharedByUser                             |
| lastModifiedDateTime        | : | 2025-08-27 23:14:00                      |
| provisioningStatus          | : | provisioned                              |
| ProvisioningPolicyGroupId   | : | 12345678-1234-1234-1234-123456789012     |
| ProvisioningPolicyGroupName | : | {\$null, grp-prv-W365-DEV-Users}         |
| aadDeviceId                 | : | 12345678-1234-1234-1234-123456789012     |
| UserPrincipalName           | : | test01@cognitionitdev.onmicrosoft.com    |
| contactEmail                | : | test01@cognitionit.com                   |

# Graph API PowerShell challenge (searchtables)

[List cloudPCs - Microsoft Graph beta | Microsoft Learn](#)

```
C: > Temp > ➔ W365licenseReport.ps1 > ⚡ function Get-W365AggregatedRemoteConnectionReportWithEnrichedData ()  
394     function Get-W365AggregatedRemoteConnectionReportWithEnrichedData {  
409         #region Create URI for Graph call to collect the Total Aggregated Remote Connection Reports  
455         |  
456             # Add ServicePlan details  
457             if ($webrequestResults.Schema[$i].Column -eq 'ServicePlanId') {  
458                 $servicePlan = $ServicePlanSearchtable.where({$_.id -eq $item[$i] })  
459                 $aggregatedRemoteConnection.Add('ServicePlanName', $servicePlan.displayName)  
460                 $aggregatedRemoteConnection.Add('ServicePlanType', $servicePlan.type)  
461             }  
|
```

PROBLEMS TERMINAL PORTS GITLENS AZURE OUTPUT DEBUG CONSOLE

PS C:\Temp> \$w365ServicePlanSearchtable.where({\$\_.id -eq '50ef7026-6174-40ba-bff7-f0e4fcddbf65'})

```
id          : 50ef7026-6174-40ba-bff7-f0e4fcddbf65  
displayName : Cloud PC Frontline 2vCPU/8GB/256GB  
type        : enterprise  
vCpuCount   : 2  
ramInGB     : 8  
storageInGB  : 256  
userProfileInGB : 25  
supportedSolution : windows365  
provisioningType : shared
```

PS C:\Temp>

# Graph API PowerShell challenge (searchtables)

[List cloudPCs - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Build the Searchtables
$w365ServicePlanSearchtable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchtable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) ` 
    -ProvisioningPolicyGroupDisplayName "grp-prv-W365"
$w365LicenseGroupsSearchtable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) ` 
    -LicenseGroupDisplayName "grp-lic-W365"
$w365ProvisioningPoliciesSearchtable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) ` 
    -ProvisioningPolicyGroupSearchTable $w365ProvisioningPolicyGroupSearchtable ` 
    -ServicePlanSearchTable $w365ServicePlanSearchtable
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)
```

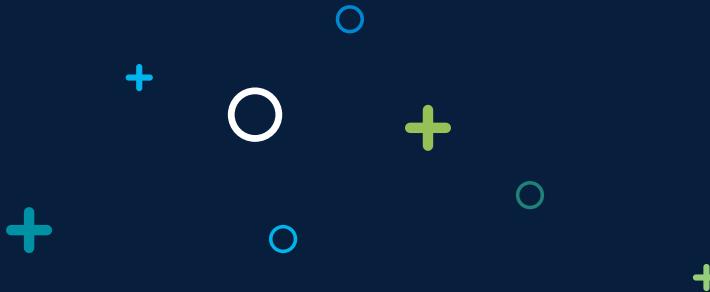
# Graph API PowerShell challenge (searchtables)

[List cloudPCs - Microsoft Graph beta](#) | [Microsoft Learn](#)

```
# Build the Searchtables
$w365ServicePlanSearchtable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchtable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) ` 
    -ProvisioningPolicyGroupDisplayName "grp-prv-W365"
$w365LicenseGroupsSearchtable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) ` 
    -LicenseGroupDisplayName "grp-lic-W365"
$w365ProvisioningPoliciesSearchtable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) ` 
    -ProvisioningPolicyGroupSearchTable $w365ProvisioningPolicyGroupSearchtable ` 
    -ServicePlanSearchTable $w365ServicePlanSearchtable
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)
```

PROBLEMS TERMINAL PORTS GITLENS AZURE OUTPUT DEBUG CONSOLE

```
Windows365 Service Plan SearchTable: 52
Windows365 Provisioning Policy Group SearchTable: 1
Windows365 License Groups SearchTable: 1
Windows365 Provisioning Policies SearchTable: 2
Windows365 CloudPCs SearchTable: 3
PS C:\Temp>
```



# W365 License Management PowerShell script



# Graph API PowerShell script

```
#####
# Script Functions
#####
#region W365 license management functions

# Graph Authentication functions
1 reference
> function Get-GraphAccessToken { ...
}

# Search Table functions
1 reference
> function Get-W365ServicePlans { ...
}

1 reference
> function Get-W365ProvisioningPolicyGroups { ...
}

1 reference
> function Get-W365LicenseGroups { ...
}

1 reference
> function Get-W365ProvisioningPolicies { ...
}

1 reference
> function Get-W365CloudPCs { ...
}

# Intune Report functions
1 reference
> function Get-W365AggregatedRemoteConnectionReport { ...
}

1 reference
> function Get-W365AggregatedRemoteConnectionReportWithEnrichedData { ...
}

# User functions
1 reference
> function Get-W365UserId { ...
}
#endregion
```

# Graph API PowerShell script

```
#####
# Script workflow      #
#####
# Get the access token
$graphContext = Get-GraphAccessToken -tenantId $tenantId -appId $clientId -appSecret $clientSecret
# Build the SearchTables
$w365ServicePlanSearchable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) -ProvisioningPolicyGroupDisplayName "grp-prv-W365"
$w365LicenseGroupSearchable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) -LicenseGroupDisplayName "grp-lic-W365"
$w365ProvisioningPoliciesSearchable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) -ProvisioningPolicyGroupSearchTable $w365ProvisioningPolicyGroupSearchable -ServicePlanSearchTable $w365ServicePlanSearchable
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)

#####
# Retrieve Intune Total Aggregated Remote Connection Report for Windows365 #
#####
# Get Intune report data for Total Aggregated Remote Connections
$rawAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReport -AccessToken $($graphContext.access_token)
$enrichedAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReportWithEnrichedData -AccessToken $($graphContext.access_token) -ServicePlanSearchable $w365ServicePlanSearchable -LicenseGroupSearchable $w365LicenseGroupsSearchTable -Cloud

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $rawAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($fileOutputPath)\Windows365RawAggregatedRemoteConnections.json" -Force
$jsonRawData = Get-Content -Path "$($fileOutputPath)\Windows365RawAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonRawData | Export-Csv -Path "$($fileOutputPath)\Windows365RawAggregatedRemoteConnections.csv" -NoTypeInformation

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $enrichedAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($fileOutputPath)\Windows365EnrichedAggregatedRemoteConnections.json" -Force
$jsonEnrichedData = Get-Content -Path "$($fileOutputPath)\Windows365EnrichedAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonEnrichedData | Export-Csv -Path "$($fileOutputPath)\Windows365EnrichedAggregatedRemoteConnections.csv" -NoTypeInformation

#####
# FILTERING Enriched Aggregated Remote Connection Report for Windows365 #
#####
[Int]$daysToAdd = -59
$checkDate = (Get-Date).AddDays($daysToAdd)
$dateFormatString = "MM/dd/yyyy hh:mm:ss tt"
#rule 01: No license check for PRD Cloud PCs | Exclude PRD Cloud PCs as they will never expire (for now)
$filteredjsonData = $jsonEnrichedData | Where-Object { $_.provisioningPolicyName -notlike "PRD*" }

#rule 02: For never logged on provisioned CPCs (NeverLoggedIn = true) check CreateDate; if older than date limit, email user that license is revoked
#rule 03: For logged on provisioned CPCs (NeverLoggedIn = false) check LastLoggedInDate; if older than date limit, email user that license is revoked
# Filter CloudPCs with (NeverLoggedIn eq $true and CreatedDate lt $checkDate) or (NeverLoggedIn eq $false and LastActiveTime le $checkDate)
$totalFilteredjsonData = $filteredjsonData | Where-Object { (( $_.NeverSignedIn -eq $true ) -and ( $_.CreatedDate -lt $checkDate ) -and ( $_.provisioningStatus -eq 'provisioned' )) -or ( ( $_.NeverSignedIn -eq $false ) -and ( $_.LastActiveTime -le $checkDate ) -and ( $_.provisioningStatus -eq 'provisioned' ) ) }

# Loop through the filtered data and add the userId to the hashtable
$enrichedTotalFilteredjsonData = $totalFilteredjsonData | ForEach-Object {
    $userPrincipalName = $_.UserPrincipalName
    $userId = Get-W365UserId -AccessToken $($graphContext.access_token) -userPrincipalName $userPrincipalName
    $_ | Add-Member -MemberType NoteProperty -Name "UserDirectoryObjectId" -Value $userId # add the userId to the hashtable
}

# Export Array of Hashtables to JSON file
ConvertTo-Json -InputObject $enrichedTotalFilteredjsonData -Depth 100 | Out-File -FilePath "C:\Temp\W365CloudPCsInactiveLast600Days.json" -Force
$jsonFinalData = Get-Content -Path "C:\Temp\W365CloudPCsInactiveLast600Days.json" -Raw | ConvertFrom-Json
$jsonFinalData | Export-Csv -Path "C:\Temp\W365CloudPCsInactiveLast600Days.csv" -NoTypeInformation -Force
```

# Graph API PowerShell script

```
#####
# Script workflow          #
#####

# Get the access token
$graphContext = Get-GraphAccessToken -TenantId $tenantId -AppId $ClientId -AppSecret $ClientSecret
# Build the Searchtables
$w365ServicePlanSearchtable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchtable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) -ProvisioningPolicyGroupType "Group"
$w365LicenseGroupsSearchtable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) -LicenseGroupDisplayName "grp-1"
$w365ProvisioningPoliciesSearchtable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) -ProvisioningPolicyType "Policy"
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)

#-----
# Retrieve Intune Total Aggregated Remote Connection Report for Windows365 #
#-----#
# Get Intune report data for Total Aggregated Remote Connections
$rawAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReport -AccessToken $($graphContext.access_token)
$enrichedAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReportWithEnrichedData -AccessToken $($graphContext.access_token)

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $rawAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365RawAggregatedRemoteConnections.json"
$jsonRawData = Get-Content -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonRawData | Export-Csv -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.csv" -NoTypeInformation

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $enrichedAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json"
$jsonEnrichedData = Get-Content -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonEnrichedData | Export-Csv -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.csv" -NoTypeInformation
```

# Graph API PowerShell script

```
#####
# Script workflow          #
#####
# Get the access token
$graphContext = Get-GraphAccessToken -TenantId $tenantId -AppId $clientId -AppSecret $clientSecret
# Build the Searchtables
$w365ServicePlanSearchtable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchtable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) -ProvisioningPolicyGroupType "Group"
$w365LicenseGroupsSearchtable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) -LicenseGroupDisplayName "grp-1"
$w365ProvisioningPoliciesSearchtable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) -ProvisioningPolicyType "Policy"
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)

#-----
# Retrieve Intune Total Aggregated Remote Connection Report for Windows365  #
#-----#
# Get Intune report data for Total Aggregated Remote Connections
$rawAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReport -AccessToken $($graphContext.access_token)
$enrichedAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReportWithEnrichedData -AccessToken $($graphContext.access_token)

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $rawAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365RawAggregatedRemoteConnections.json"
$jsonRawData = Get-Content -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonRawData | Export-Csv -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.csv" -NoTypeInformation

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $enrichedAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json"
$jsonEnrichedData = Get-Content -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonEnrichedData | Export-Csv -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.csv" -NoTypeInformation
```

# Graph API PowerShell script

```
#####
# Script workflow          #
#####
# Get the access token
$graphContext = Get-GraphAccessToken -TenantId $tenantId -AppId $ClientId -AppSecret $ClientSecret
# Build the Searchtables
$w365ServicePlanSearchtable = Get-W365ServicePlans -AccessToken $($graphContext.access_token)
$w365ProvisioningPolicyGroupSearchtable = Get-W365ProvisioningPolicyGroups -AccessToken $($graphContext.access_token) -ProvisioningPolicyGroupType "Group"
$w365LicenseGroupsSearchtable = Get-W365LicenseGroups -AccessToken $($graphContext.access_token) -LicenseGroupDisplayName "grp-lic"
$w365ProvisioningPoliciesSearchtable = Get-W365ProvisioningPolicies -AccessToken $($graphContext.access_token) -ProvisioningPolicyType "Policy"
$w365CloudPCs = Get-W365CloudPCs -AccessToken $($graphContext.access_token)

#-----
# Retrieve Intune Total Aggregated Remote Connection Report for Windows365 #
#-----#
# Get Intune report data for Total Aggregated Remote Connections
$rawAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReport -AccessToken $($graphContext.access_token)
$enrichedAggregatedRemoteConnections = Get-W365AggregatedRemoteConnectionReportWithEnrichedData -AccessToken $($graphContext.access_token)

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $rawAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365RawAggregatedRemoteConnections.json"
$jsonRawData = Get-Content -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonRawData | Export-Csv -Path "$($FilePath)\Windows365RawAggregatedRemoteConnections.csv" -NoTypeInformation

# Export Array of Hashtables to JSON file and CSV
ConvertTo-Json -InputObject $enrichedAggregatedRemoteConnections -Depth 100 | Out-File -FilePath "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json"
$jsonEnrichedData = Get-Content -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.json" -Raw | ConvertFrom-Json
$jsonEnrichedData | Export-Csv -Path "$($FilePath)\Windows365EnrichedAggregatedRemoteConnections.csv" -NoTypeInformation
```

# Graph API PowerShell script

```
#-----#
# FILTERING Enriched Aggregated Remote Connection Report for Windows365 #
#-----#
[int]$daysToAdd = -59
$checkDate = (Get-Date).AddDays($daysToAdd)
$dateFormatString = "MM/dd/yyyy hh:mm:ss tt"
#rule 01: No license check for PRD Cloud PCs | Exclude PRD CloudPCs as they will never expire (for now)
$filteredJsonData = $jsonEnrichedData
$filteredJsonData = $filteredJsonData.Where({ ($_.provisioningPolicyName -notlike 'PRD*') })
#rule 02: For never logged on provisioned CPCs (NeverLoggedIn = true) check CreateDate; if older than date limit, email user that
#rule 03: For logged on provisioned CPCs (NeverLoggedIn = false) check LastLoggedInDate; if older than date limit, email user tha
# Filter CloudPCs with (NeverLoggedIn eq $true and CreatedDate lt $checkDate) or (NeverLoggedIn eq $false and LastActiveTime le $checkDate)
$totalFilteredJsonData = $filteredJsonData.Where({ (($_.NeverSignedIn -eq $true) -and ($_.CreatedDate -lt $checkDate)) -or ($_.NeverSignedIn -eq $false) -and ($_.LastActiveTime -le $checkDate) })
# Loop through the filtered data and add the userId to the hashtable
$enrichedTotalFilteredJsonData = $totalFilteredJsonData
$enrichedTotalFilteredJsonData | ForEach-Object {
    $userPrincipalName = $_.UserPrincipalName
    $userId = Get-W365UserId -AccessToken $($graphContext.access_token) -userPrincipalName $userPrincipalName
    $_ | Add-Member -MemberType NoteProperty -Name "UserDirectoryObjectId" -Value $userId -Force # add the userId to the hashtable
}

# Export Array of Hashtables to JSON file
ConvertTo-Json -InputObject $enrichedTotalFilteredJsonData -Depth 100 | Out-File -FilePath "C:\Temp\W365CloudPCsInactiveLast60Days.json"
$finalJsonData = Get-Content -Path "C:\Temp\W365CloudPCsInactiveLast60Days.json" -Raw | ConvertFrom-Json
$finalJsonData | Export-Csv -Path "C:\Temp\W365CloudPCsInactiveLast60Days.csv" -NoTypeInformation -Force
```

# Graph API PowerShell script

```
#-----#
# FILTERING Enriched Aggregated Remote Connection Report for Windows365 #
#-----#
[int]$daysToAdd = -59
$checkDate = (Get-Date).AddDays($daysToAdd)
$dateFormatString = "MM/dd/yyyy hh:mm:ss tt"
#rule 01: No license check for PRD Cloud PCs | Exclude PRD CloudPCs as they will never expire (for now)
$jsonEnrichedData = Get-Content -Path "C:\Temp\W365CloudPCsInactiveLast60Days.json" -Raw | ConvertFrom-Json
$filteredJsonData = $jsonEnrichedData.Where({ ($_.provisioningPolicyName -notlike 'PRD*') })
#rule 02: For never logged on provisioned CPCs (NeverLoggedIn = true) check CreateDate; if older than date limit, email user that
#rule 03: For logged on provisioned CPCs (NeverLoggedIn = false) check LastLoggedInDate; if older than date limit, email user tha
# Filter CloudPCs with (NeverLoggedIn eq $true and CreatedDate lt $checkDate) or (NeverLoggedIn eq $false and LastActiveTime le $checkDate)
$totalFilteredJsonData = $filteredJsonData.Where({ (( $_.NeverSignedIn -eq $true) -and ($_.CreatedDate -lt $checkDate) -and ($_.LastActiveTime -le $checkDate)) -or (( $_.NeverSignedIn -eq $false) -and ($_.LastActiveTime -le $checkDate)) })  
  
# Loop through the filtered data and add the userId to the hashtable
$enrichedTotalFilteredJsonData = $totalFilteredJsonData
$enrichedTotalFilteredJsonData | ForEach-Object {
    $userPrincipalName = $_.UserPrincipalName
    $userId = Get-W365UserId -AccessToken $($graphContext.access_token) -userPrincipalName $userPrincipalName
    $_ | Add-Member -MemberType NoteProperty -Name "UserDirectoryObjectId" -Value $userId -Force # add the userId to the hashtable
}
# Export Array of Hashtables to JSON file
ConvertTo-Json -InputObject $enrichedTotalFilteredJsonData -Depth 100 | Out-File -FilePath "C:\Temp\W365CloudPCsInactiveLast60Days.json"
$finalJsonData = Get-Content -Path "C:\Temp\W365CloudPCsInactiveLast60Days.json" -Raw | ConvertFrom-Json
$finalJsonData | Export-Csv -Path "C:\Temp\W365CloudPCsInactiveLast60Days.csv" -NoTypeInformation -Force
```

# Automated workflow



# W365 CloudPC License Management

## Get-W365CPC-InactiveFor60Days-Report-Data

```
Retrieve CloudPCReports:  
getTotalAggregatedRemoteConnectionReports
```

```
Enrich report data for  
readability & follow-up actions
```

```
Filter report data on  
CloudPC Types &  
60 days inactivity
```

## Revoke-W365CPC-License

```
Remove User from  
CloudPC license group  
(Enterprise CPC)
```

```
Remove User from  
CloudPC provisioning group  
(Enterprise & Frontline CPC)
```

## Notify-W365CPC-User

```
Send notification email  
to CloudPC User
```

# W365 CloudPC License Management

## Get-W365CPC-InactiveFor60Days-Report-Data

Retrieve CloudPCReports:  
getTotalAggregatedRemoteConnectionReports

Enrich report data for  
readability & follow-up actions

Filter report data on  
CloudPC Types &  
60 days inactivity

| ManagedDeviceName | DeviceType         | NeverSignedIn | DaysSinceLastSignIn | contactEmail          | provisioningPolicyGroupName      | ExpectedEnterpriseLicenseGroupName | ExpectedEnterpriseLicenseGroupId   | provisioningPolicyGroupId                      | UserDirectoryObjectId |
|-------------------|--------------------|---------------|---------------------|-----------------------|----------------------------------|------------------------------------|--|--|-----------------------|
| TST-test0-B5IM4   | Enterprise         | TRUE          | 65                  | test0@cognitionit.com | W365-TST Test Cloud PC           | W365-LIC-4vCPU-16GB-128GB          | 3da17d79-1234-1234-1234-1234567736b9fc8-1234-1234-1;ae862dc8-1234-1234-12      |  |                       |
| TST-test1-KP47C   | Enterprise         | FALSE         |                     | test1@cognitionit.com | W365-TST Test Cloud PC           | W365-LIC-8vCPU-32GB-256GB          | 7bf95eb1-1234-1234-1234-1234567736b9fc8-1234-1234-1234-1;bbb9c44d-1234-1234-12 |  |                       |
| DEV-test0-VEOL3   | Enterprise         | FALSE         | 66                  | test0@cognitionit.com | W365-DEV Development Cloud PC    | W365-LIC-4vCPU-16GB-128GB          | 3da17d79-1234-1234-1234-1234567b8483202-1234-1234-1234-1;c5a187e5-1234-1234-12 |  |                       |
| DEV-test1-ZSX8    | FrontlineDedicated | FALSE         |                     | test1@cognitionit.com | W365-DEV Development Cloud PC FL |                                    |  | a0249b00-1234-1234-1234-1afb9d8b7-1234-1234-12 |                       |
| DEV-test2-FAUF8   | Enterprise         | FALSE         | 63                  | test2@cognitionit.com | W365-DEV Development Cloud PC    | W365-LIC-2vCPU-8GB-128GB           | e37795cc-1234-1234-1234-1234567b8483202-1234-1234-1234-107e742fa-1234-1234-12  |  |                       |

## Revoke-W365CPC-License

Remove User from  
CloudPC license group  
(Enterprise CPC)

Remove User from  
CloudPC provisioning group  
(Enterprise & Frontline CPC)

## Notify-W365CPC-User

Send notification email  
to CloudPC User

# W365 CloudPC License Management

## Get-W365CPC-InactiveFor60Days-Report-Data



| ManagedDeviceName | DeviceType         | NeverSignedIn | DaysSinceLastSignIn | contactEmail          | provisioningPolicyGroupName      | ExpectedEnterpriseLicenseGroupName | ExpectedEnterpriseLicenseGroupId   | provisioningPolicyGroupId | UserDirectoryObjectId |
|-------------------|--------------------|---------------|---------------------|-----------------------|----------------------------------|------------------------------------|--|---------------------------|-----------------------|
| TST-test0-B5IM4   | Enterprise         | TRUE          | 65                  | test0@cognitionit.com | W365-TST Test Cloud PC           | W365-LIC-4vCPU-16GB-128GB          | 3da17d79-1234-1234-1234-1234567736b9fc8-1234-1234-1;ae862dc8-1234-1234-1 |                           |                       |
| TST-test1-KP47C   | Enterprise         | FALSE         |                     | test1@cognitionit.com | W365-TST Test Cloud PC           | W365-LIC-8vCPU-32GB-256GB          | 7bf95eb1-1234-1234-1234-1234567736b9fc8-1234-1234-1;bbb9c44d-1234-1234-1 |                           |                       |
| DEV-test0-VEOL3   | Enterprise         | FALSE         | 66                  | test0@cognitionit.com | W365-DEV Development Cloud PC    | W365-LIC-4vCPU-16GB-128GB          | 3da17d79-1234-1234-1234-1234567b8483202-1234-1234-1-c5a187e5-1234-1234-1 |                           |                       |
| DEV-test1-ZSX8    | FrontlineDedicated | FALSE         |                     | test1@cognitionit.com | W365-DEV Development Cloud PC FL |                                    | a0249b00-1234-1234-1234-1afb9d8b7-1234-1234-12                           |                           |                       |
| DEV-test2-FAUF8   | Enterprise         | FALSE         | 63                  | test2@cognitionit.com | W365-DEV Development Cloud PC    | W365-LIC-2vCPU-8GB-128GB           | 7795cc-1234-1234-1234-1234567b8483202-1234-1234-1-07e742fa-1234-1234-12  |                           |                       |

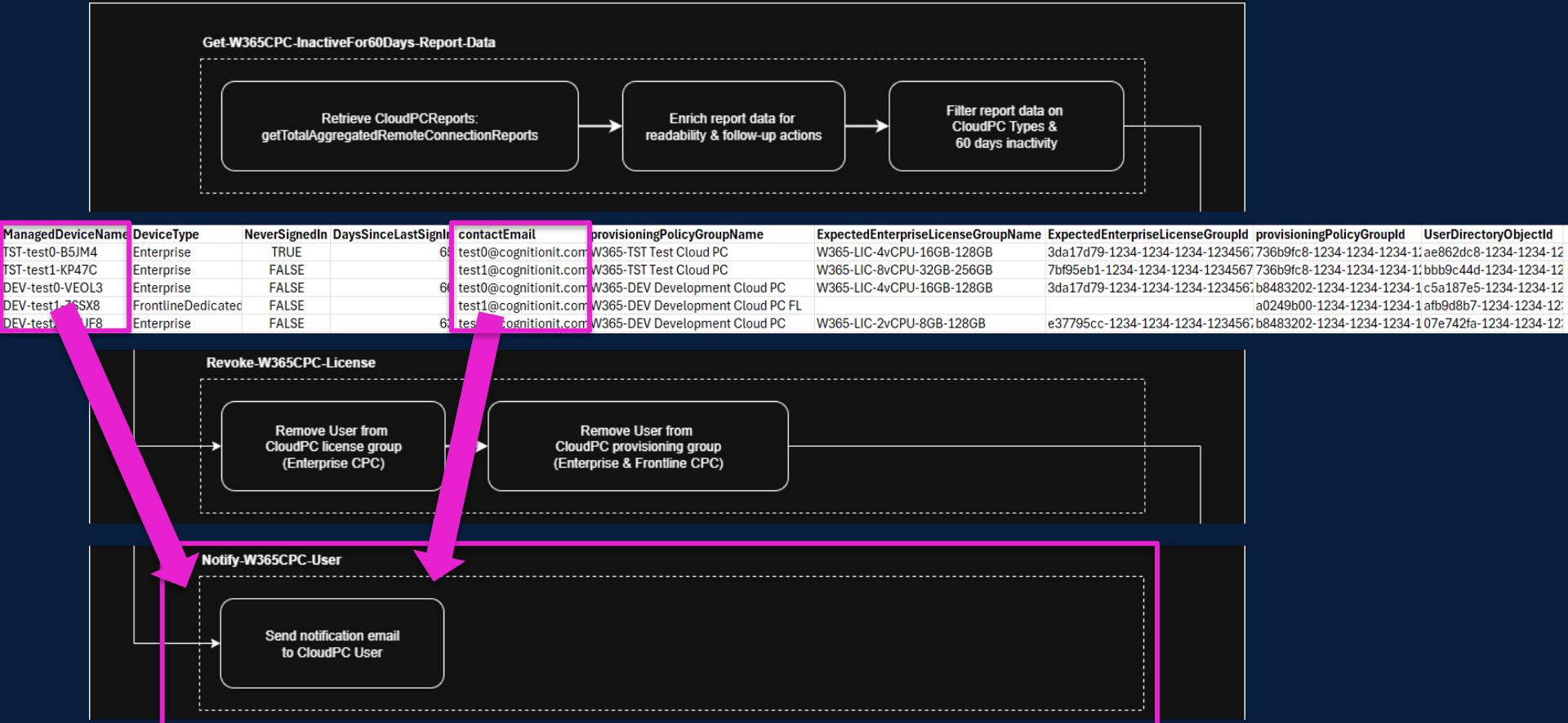
## Revoke-W365CPC-License



## Notify-W365CPC-User



# W365 CloudPC License Management





# THANK YOU

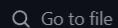
Are there any questions?

cloud-devops-ninja / myPresentations Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

Files



main



Go to file



Alyx



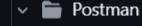
ExpertsLive



GraphAPI\_StarterKit



Code

[00\\_Create\\_Entra\\_Group.bicep](#)[00\\_Create\\_Entra\\_Group.bicepparam](#)[00\\_Create\\_Entra\\_Group.ps1](#)[01\\_AVD\\_Assign\\_AIM\\_Role\\_To\\_Group.ps1](#)[02\\_W365\\_Create\\_Provisioning\\_Policy.ps1](#)[03\\_W365\\_Provision\\_CloudPC\\_to\\_User.ps1](#)[bicepconfig.json](#)

Postman

[ExpertsLive\\_Collection.json](#)[createUUID.js](#)[storeAzureBearerToken.js](#)myPresentations / GraphAPI\_StarterKit / Code

cloud-devops-ninja Adding Connect SlideDecks

6f16799 · 3 months ago

[History](#)

| Name    | Last commit message       | Last commit date |
|---------|---------------------------|------------------|
| ..      |                           |                  |
| Code    | updating presentations    | 4 months ago     |
| Postman | updating presentations    | 4 months ago     |
| Slides  | Adding Connect SlideDecks | 3 months ago     |

code: [https://github.com/cloud-devops-ninja/myPresentations/tree/main/GraphAPI\\_StarterKit](https://github.com/cloud-devops-ninja/myPresentations/tree/main/GraphAPI_StarterKit)