

NEXXT



WORKPLACEDUDES

Meetup

Streamline your Windows 365 management with PSCloudPC

Stefan Dingemanse & Niels Kok

8 Februari 2024



WORKPLACEDUDES

Community
edition

Streamline your Windows 365 management with PS CloudPC

By Stefan Dingemanse and Niels Kok



WORKPLACEDUDES

Question for Audience

Who uses Windows 365?

In production environment?

Deployment?





BRAINPULSE IT
CONSULTANCY & TRAINING

Introduction

Stefan Dingemanse

- 34 years - Dordrecht, Netherlands
- Freelance Microsoft (Cloud) Consultant @ Brainpulse IT
- Microsoft MVP Enterprise Mobility & Microsoft Certified Trainer
- Modern Workplace, EUC, AVD/W365, Azure en DevOps
- Co-Founder AVD Community, DVDUG & Azure Thursday

- Website: StefanDingemanse.com
- Twitter: [@Sdingemanse](https://twitter.com/Sdingemanse)
- E-mail: Stefan@brainpulse.it



WORKPLACEDUDES

Introduction

Niels Kok

- 32 years, Enkhuizen
- Freelance Consultant
- Microsoft 365, AVD, Azure (DevOps)
- Community stuff: (Blog, Speaking etc..)



Agenda

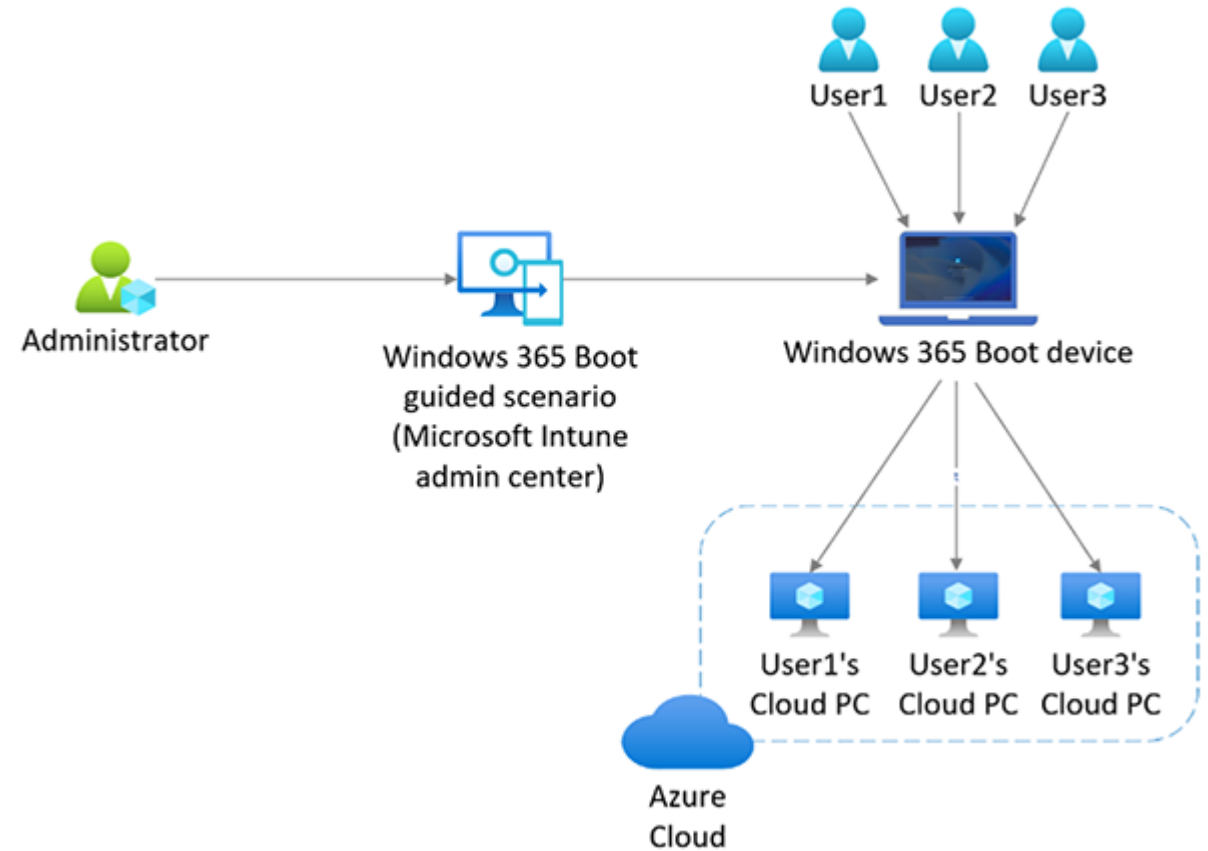
- What's new in Windows 365
- Use Cases
- Deploying Windows 365
- PSCloudPC PowerShell Module
- DEMO's



What's new Windows 365

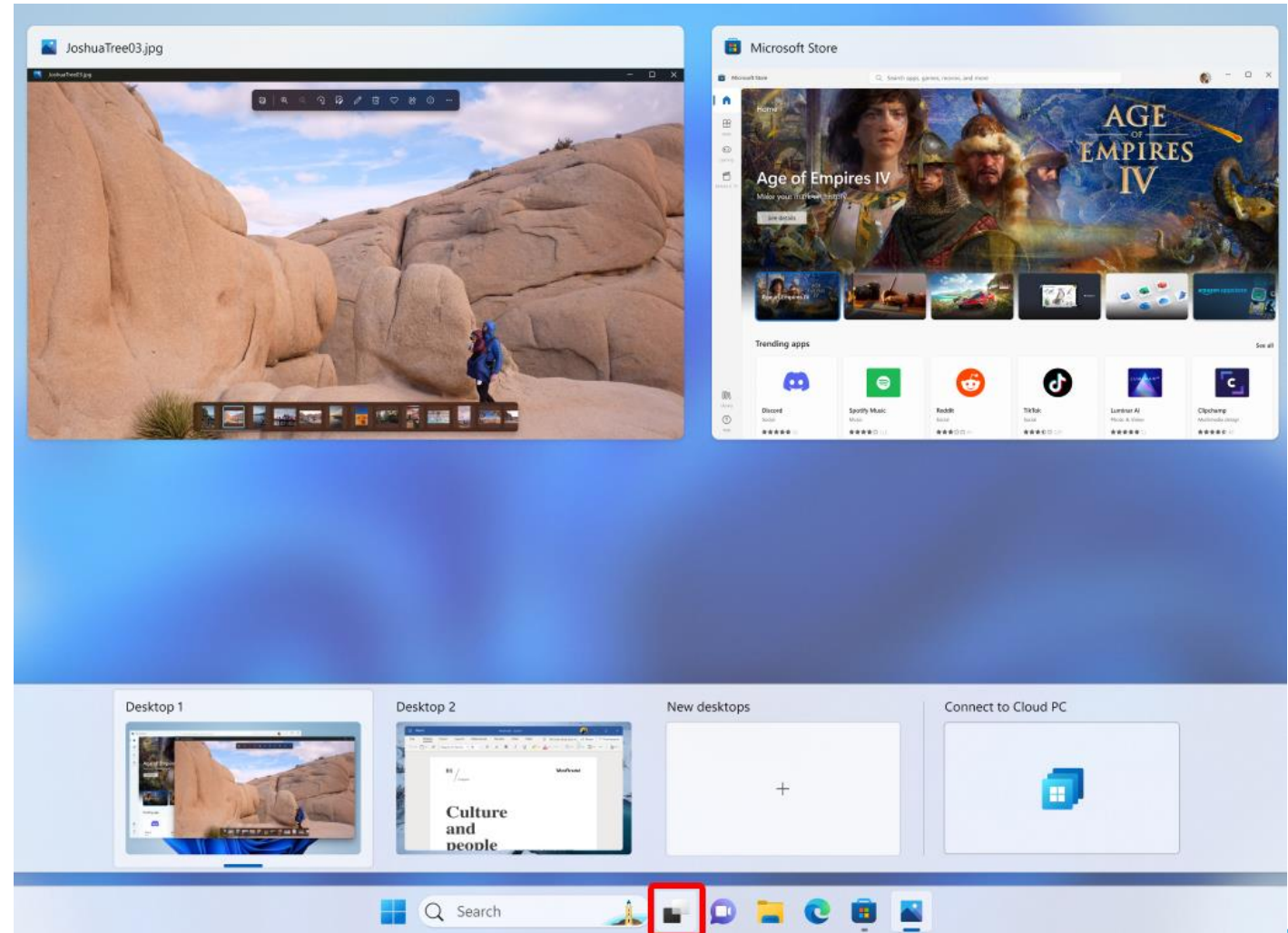
Windows 365 Boot

- Sign in directly to your Cloud PC
- Scenarios
 - Shared
 - Dedicated (Public Preview)
- Configured via Intune



Windows 365 Switch

- Sign in and Connect via the Windows 11 Task view



Windows 365 Frontline

- Unique Cloud PC for every employee
- No scheduling required
 - Signing out will free up licenses
 - Cloud PC will reboot before turning off
 - Cloud PC is turned off when not in use
- Ratio 1:3

What about Azure Virtual Desktop



Windows 365 vs AVD

- Built on Azure Virtual Desktop
- Uses the same AVD control plane
- Uses the same AVD clients
- Windows 365 is optimized for simplicity
- AVD is optimized for flexibility

Windows 365 vs AVD

Cloud PC – Windows 365	Cloud VDI – Azure Virtual Desktop
Windows 10/11	Windows 10/11, Windows Server 2012+ multi-session desktops
Full Desktops	Full Desktops and Remote Apps
Microsoft Endpoint Manager (Enterprise Edition)	Full control over configuration and management. Azure portal and MEM
Predictable per user per month pricing	Flexible consumption-based pricing
Dedicated VM's	Dedicated and pooled with user profile management
Limited SKU in 22 regions	All VM SKU's in all regions
GPU support	GPU Support
Simple management	Complex management
A PC in the cloud	Virtual Desktop Platform



Azure Virtual Desktop Gateway Service Geographies



Use cases

Daily Workstation for employees



Daily Workstation for non-Office workers



Consultants Workstation

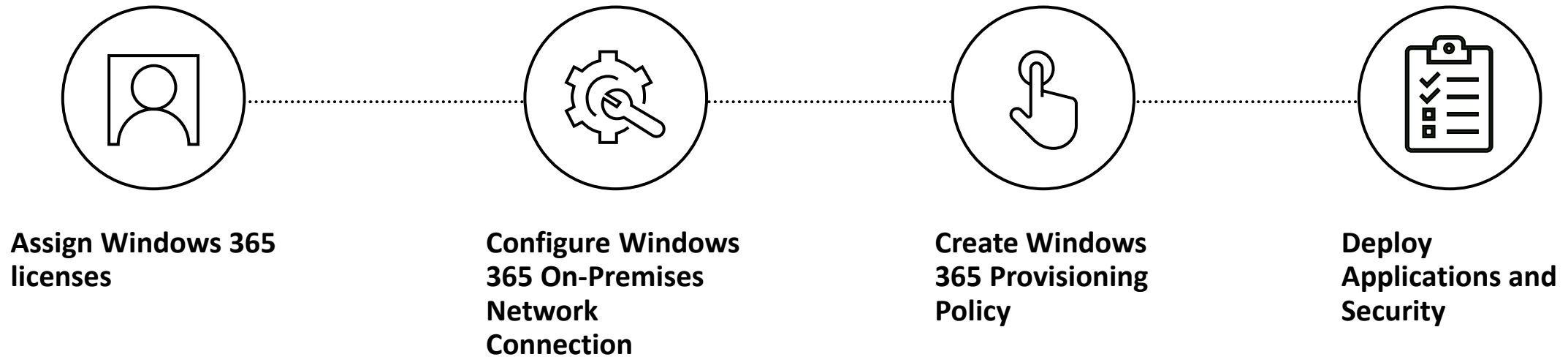


Developers Workstation



Deployment flow

Deployment flow



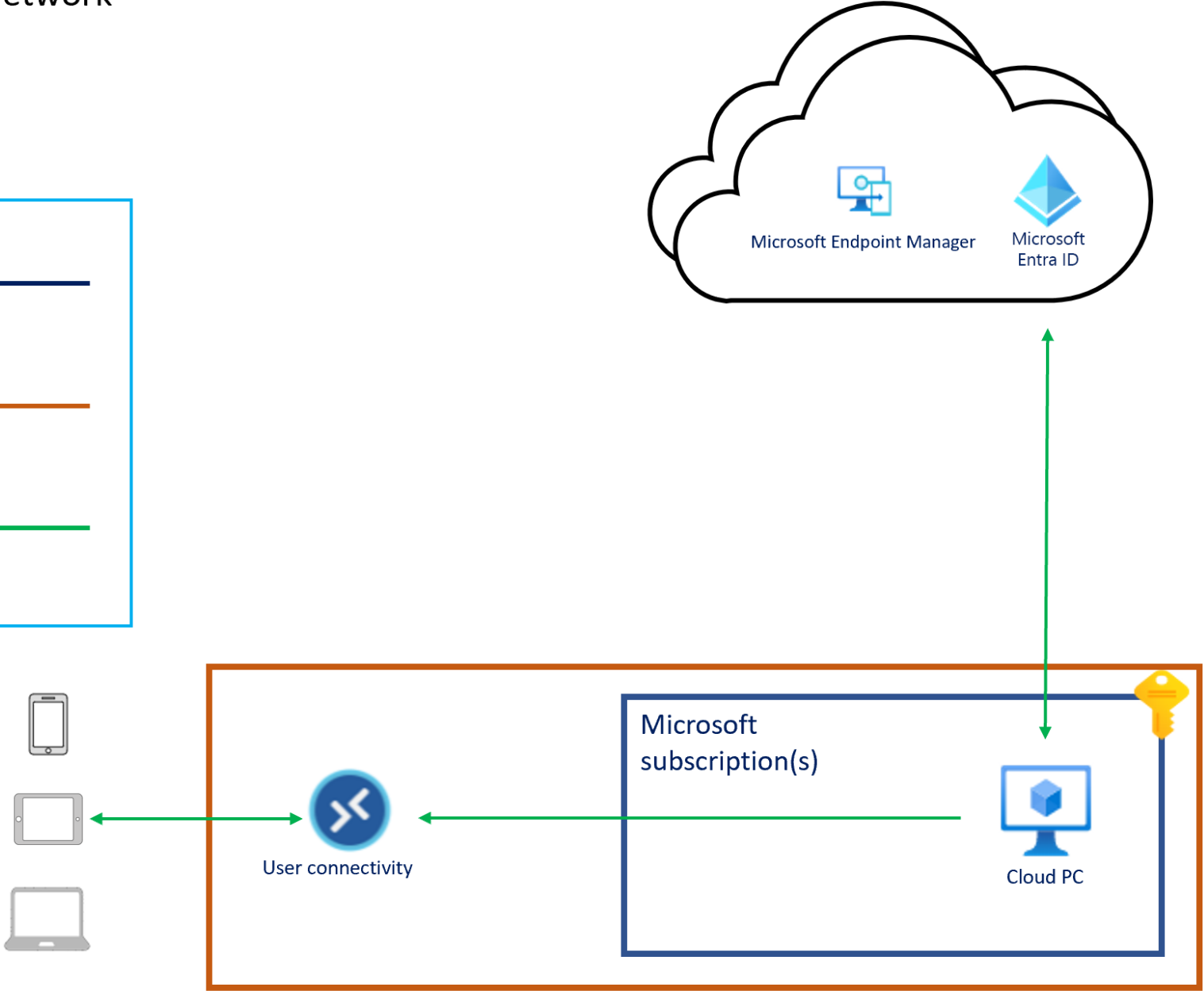
Microsoft Entra join

Microsoft Hosted Network

Azure Subscriptions

Managed by Microsoft

Network flow



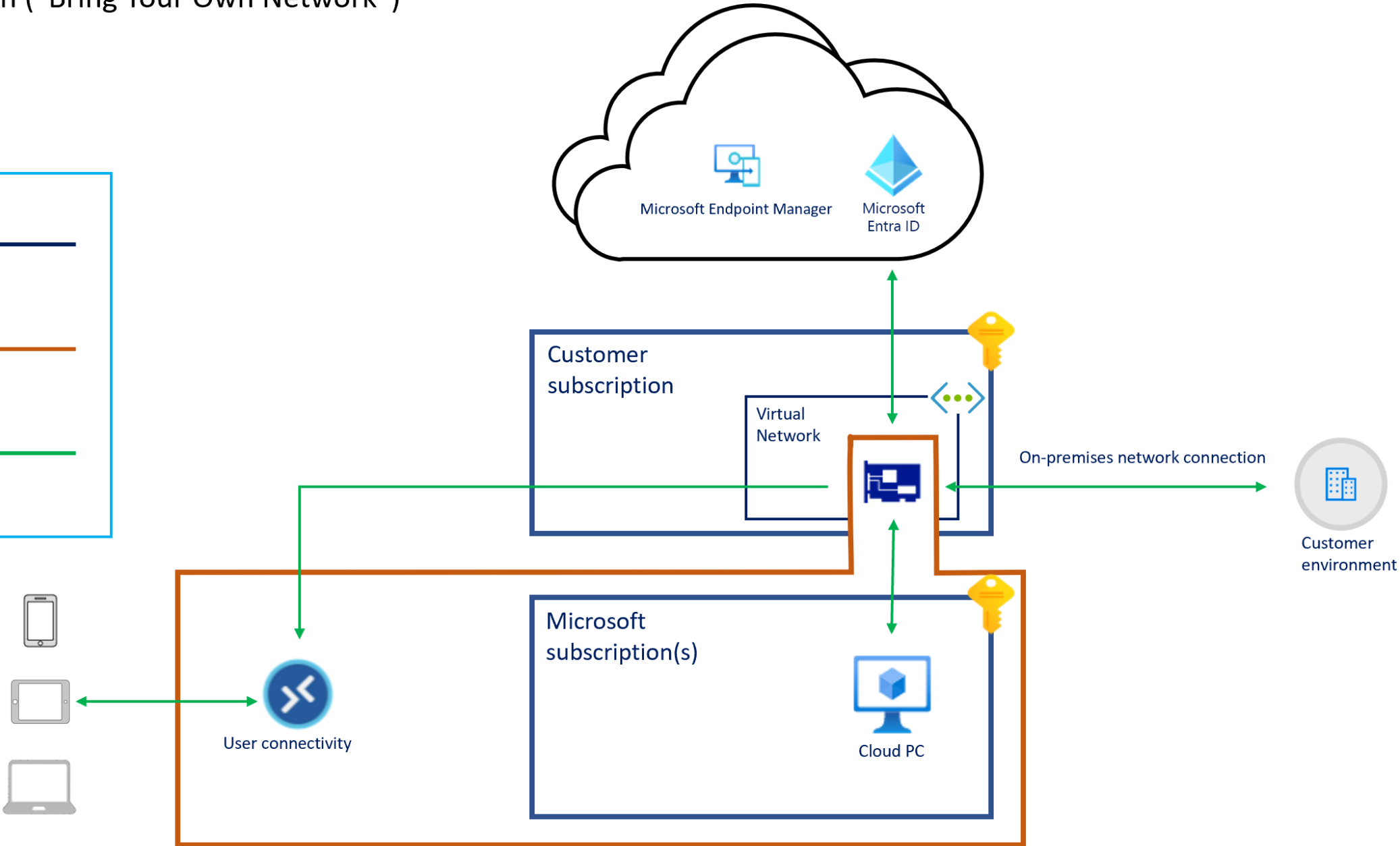
Microsoft Entra join

Network Connection (“Bring Your Own Network”)

Azure Subscriptions

Managed by Microsoft

Network flow



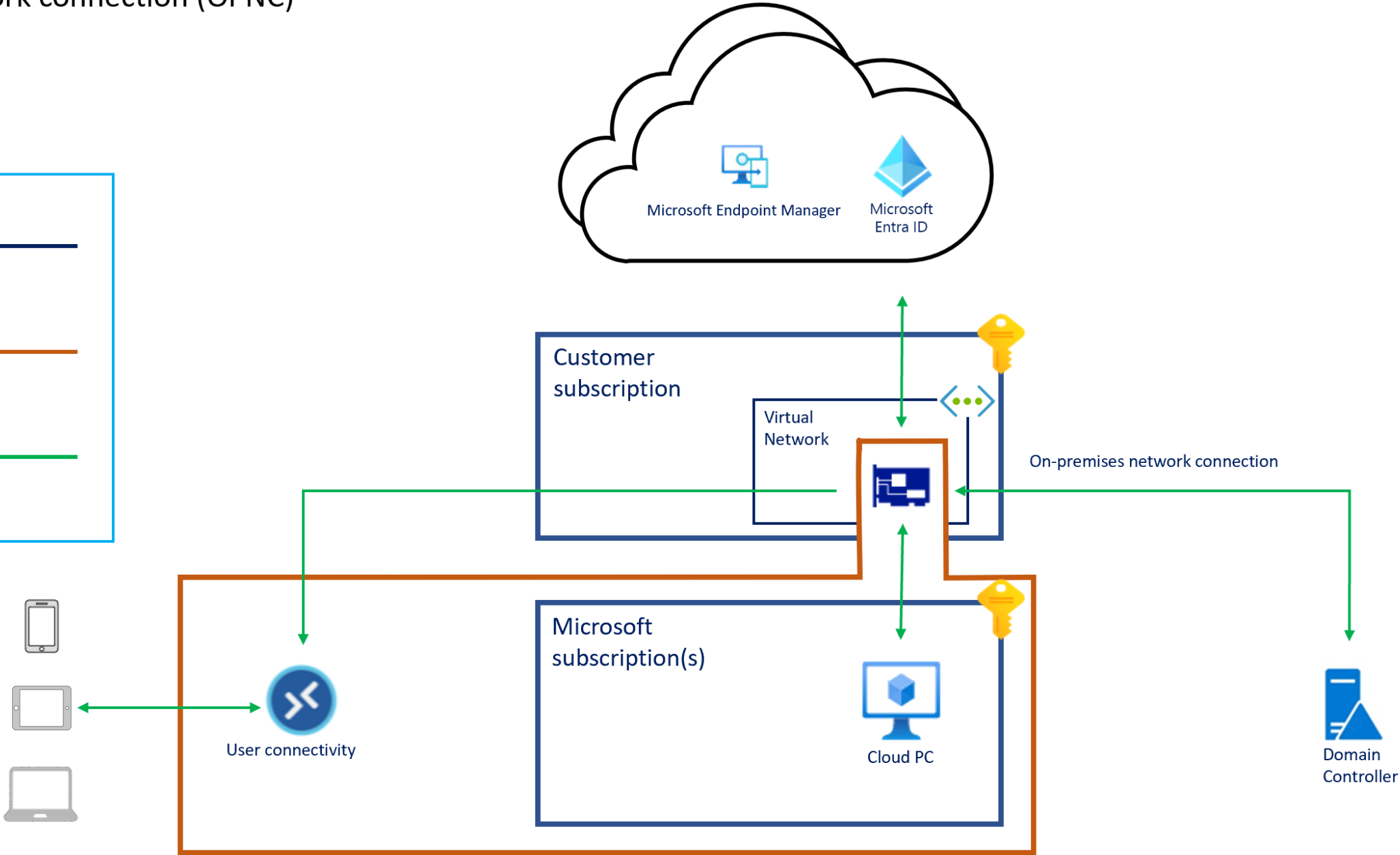
Microsoft Entra hybrid join

On-premises network connection (OPNC)

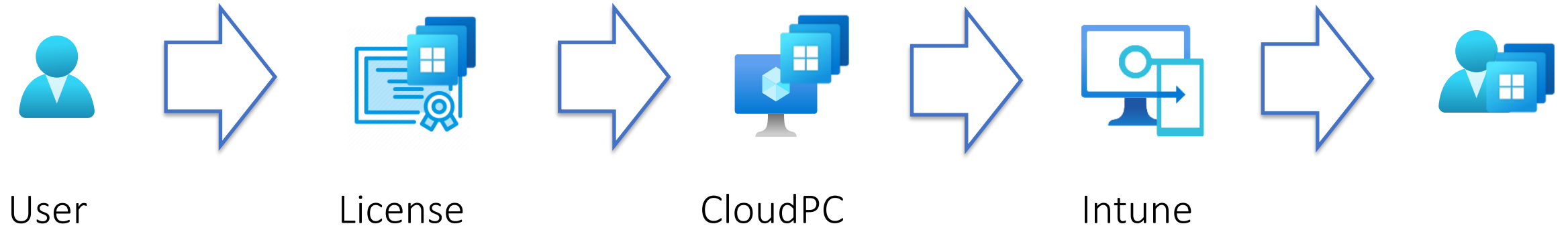
Azure Subscriptions

Managed by Microsoft

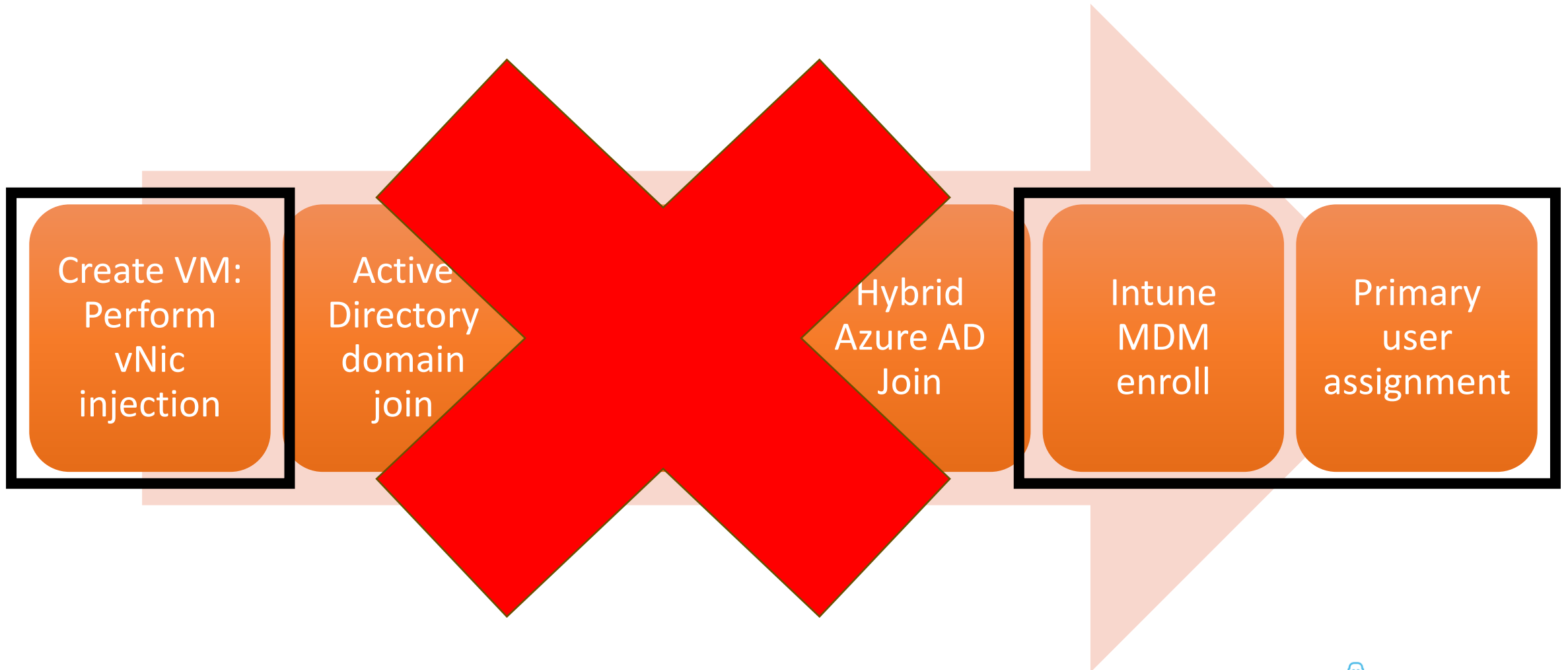
Network flow



Provisioning flow



Provisioning steps



DEMO

Manual deployment

PSCloudPC

Why?

Authentication

Flow example

Tools

Why?

- It's fun!
- Learning
- Portal vs automation
- Consistency



Why?

Microsoft Endpoint Manager admin center

Home > Devices | Windows 365 >

Create a provisioning policy

Windows 365

General Image Configuration Assignments **Review + create**

General

Name	Demo - Windows 365
Description	NielsKok.tech
Use single sign-on (preview)	Yes
Join type	Azure AD Join
Geography	European Union
Region	West Europe

Image

Image type	Gallery
	Windows 11 Enterprise + Microsoft 365 Apps
	22H2

Configuration

Language & Region	English (United States)
Additional Services	None

Assignments

Groups

MDM_UserGroup

Microsoft Graph X-Ray

Save script Clear session

```
GET /groups/653967ee-8c68-446f-93b7-e9db78eae72/photos/48x48/$value
```

```
GET /groups/474494ef-3631-4d20-92fa-762b41f07396/photos/48x48/$value
```

```
POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/provisi
```

```
Import-Module Microsoft.Graph.DeviceManagement.Administration

$params = @{
    DisplayName = "Demo - Windows 365"
    Description = "NielsKok.tech"
    ProvisioningType = "dedicated"
    ManagedBy = "windows365"
    ImageId = "MicrosoftWindowsDesktop_windows-ent-cpc_win11-22h2-ent-cpc-m365"
    ImageDisplayName = "Windows 11 Enterprise + Microsoft 365 Apps 22H2"
    ImageType = "gallery"
    MicrosoftManagedDesktop = @{
        Type = "notManaged"
        Profile = ""
    }
    EnableSingleSignOn = $true
    DomainJoinConfiguration = @{
        Type = "azureADJoin"
        RegionName = "westeurope"
        RegionGroup = "europeUnion"
        OnPremisesConnectionId = $null
    }
    WindowsSettings = @{
        Language = "en-US"
    }
}

New-MgDeviceManagementVirtualEndpointProvisioningPolicy -BodyParameter $params
```

Previous Create

WORKPLACEDUDES

Why?

```
PowerShell 7 (x64)
PS C:\Users\NKO> New-CPCProvisioningPolicy -Name PP_W365_Powershell_Demo -Description W365_Powershell_Demo -RegionName westeu
rope -RegionGroup europeUnion -ImageId "MicrosoftWindowsDesktop_windows-ent-cpc_win11-22h2-ent-cpc-m365" -EnableSingleSignOn
>true|
```


Why?

Custom images Azure network connection User settings ...

+ Add Refresh

Search Add filter

Name ↑↓	Assigned ↑↓	Date created ↑↓	
Windows 365 Demo - Portal	Yes	2/13/2023, 12:09:11 PM	...
Windows 365 Demo	Yes	2/13/2023, 11:53:20 AM	...
UserSettings_NielsKok.tech	Yes	1/17/2023, 8:55:00 AM	...

```
GET /groups/93f2df51-93d1-4221-8574-585888a5d06d/photos/48x48/$value
```

```
POST
https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/userSet
...

Import-Module Microsoft.Graph.DeviceManagement.Administration

$params = @{
    DisplayName = "Windows 365 Demo - Portal"
    LocalAdminEnabled = $false
    RestorePointSetting = @{
        UserRestoreEnabled = $true
        FrequencyInHours = 6
    }
}

New-MgDeviceManagementVirtualEndpointUserSetting -BodyParameter $params
```

```
POST
https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/userSet
edf2-4b63-9ed1-05e0ee5739c3/assign
...

Import-Module Microsoft.Graph.DeviceManagement.Actions

$params = @{
    Assignments = @(
        @{
            Target = @{
                GroupId = "82d63e26-0bc2-48cd-84a7-
1ad212650725"
            }
        }
    )
}

Set-MgDeviceManagementVirtualEndpointUserSetting -CloudPcUserSettingId
$cloudPcUserSettingId -BodyParameter $params
```



Why?

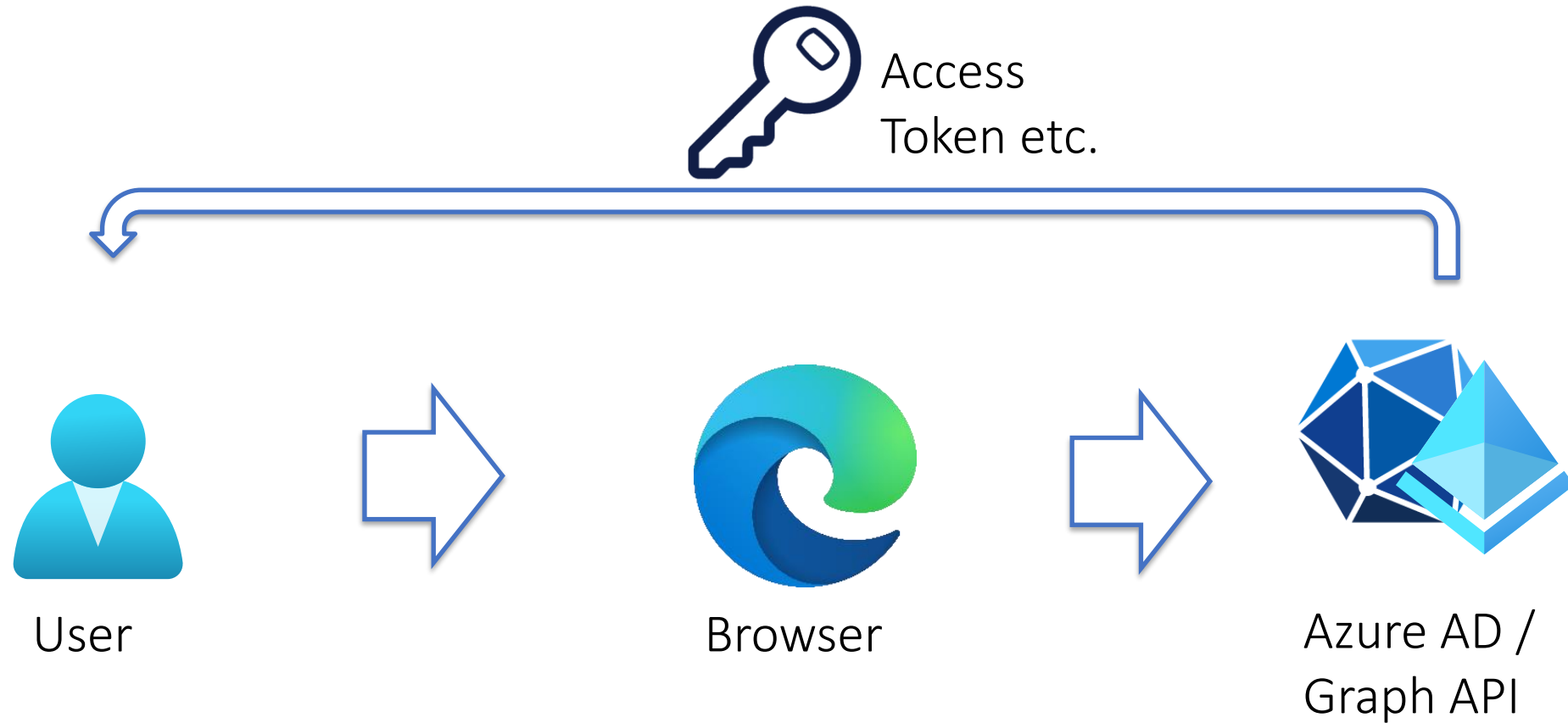
```
PowerShell 7 (x64) X + v
PS C:\Users\NKO> New-CPCUserSettingsPolicy -Name US_W365_Powershell_Demo -LocalAdminEnabled $false -UserRestoreEnabled $true -FrequencyInHours 6
```

```
PowerShell 7 (x64) X + v
PS C:\Users\NKOAdmin> Set-CPCUserSettingsPolicyAssignment -Name "Windows 365 Demo Powershell" -GroupName 'ExpertLiveDemo'
```

Prerequisites

- Authentication (Interactive / Service Principal)
- PowerShell Core (7)
- Microsoft Graph
- Easy to use / automation friendly
- Not depending on other modules
- Available via the Microsoft PowerShell gallery

Authentication (Interactive)



Authentication (Interactive)

- CloudPC.ReadWrite.All
- DeviceManagementConfiguration.ReadWrite.All
- DeviceManagementManagedDevices.ReadWrite.All
- Directory.Read.All

<https://login.microsoftonline.com/common/oauth2/nativeclient>

Microsoft Azure

Search resources, services, and docs (G+ /)

Home > NielsKok.Tech - LAB | Enterprise applications > Enterprise applications

Enterprise applications | All applications

NielsKok.Tech - LAB - Azure Active Directory

Overview

- Overview
- Diagnose and solve problems

Manage

- All applications
- Application proxy
- User settings
- App launchers
- Custom authentication extensions (Preview)

Security

- Conditional Access
- Consent and permissions

Activity

- Sign-in logs
- Usage & insights
- Audit logs

+ New application Refresh Download (Export) Preview info Columns

View, filter, and search applications in your organization that are set up to use your Azure AD tenant as their IdP.

The list of applications that are maintained by your organization are in [application registrations](#).

Search by application name or object ID Application type == Enterprise Applications

8 applications found

Name	Object ID
Microsoft Graph PowerShell	109abe28-5b42-4c7b-8ece-4f0a06501082
Graph Explorer	14d47a46-a307-4984-9267-721e27d414fe
AccountingCloudPro	...
...	...
...	...
...	...
...	...
...	...

Authentication (Interactive)

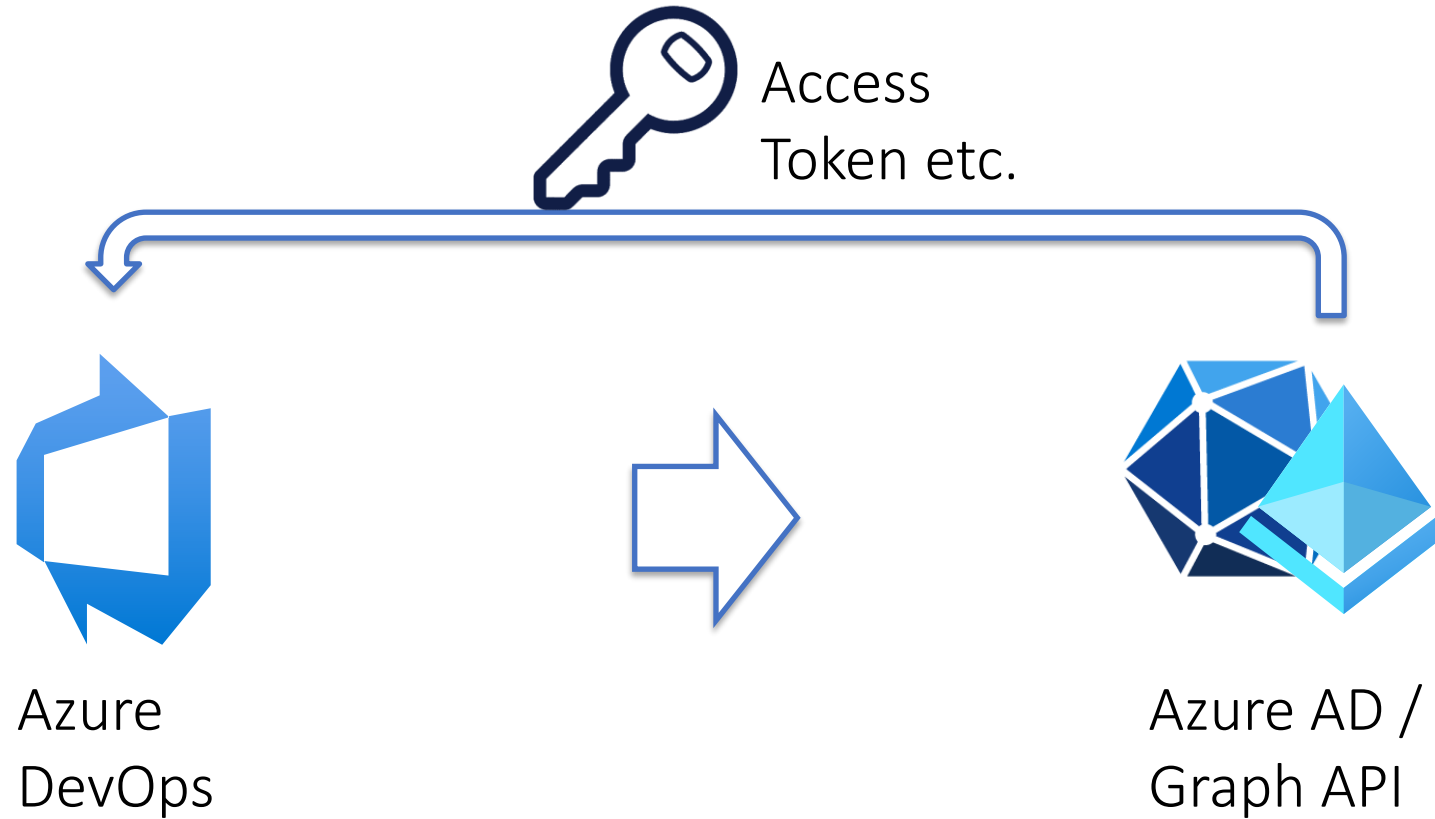
Authtoken

[illegible]

Authtoken

[illegible]

Authentication (SPN)



Authentication (SPN)

The screenshot shows the Microsoft Azure portal interface for the 'NielsKok.Tech - LAB' subscription. The left sidebar contains navigation links for Overview, Preview features, Diagnose and solve problems, and a Manage section with links to Users, Groups, External Identities, Roles and administrators, Administrative units, Delegated admin partners, Enterprise applications, Devices, App registrations (selected), Identity Governance, and Application proxy. The main content area is titled 'App registrations' and includes a search bar and a list of actions: New registration, Endpoints, Troubleshooting, Refresh, Download, and Preview features. A notification banner states that starting June 30th, 2020, new features will no longer be added to the Azure Active Directory Authentication Library (ADAL) and applications will need to be upgraded to the Microsoft Authentication Library (MSAL) and Microsoft Graph. Below this, there are tabs for All applications, Owned applications (selected), and Deleted applications. A search bar prompts the user to 'Start typing a display name or application (client) ID to filter these r...'. A list of 5 applications is shown, with 'ExpertsLive-W365' highlighted by a red box. The application's icon is a red square with 'EX' in white.

Microsoft Azure

Home > NielsKok.Tech - LAB

NielsKok.Tech - LAB | App registrations

Azure Active Directory

Overview

Preview features

Diagnose and solve problems

Manage

Users

Groups

External Identities

Roles and administrators

Administrative units

Delegated admin partners

Enterprise applications

Devices

App registrations

Identity Governance

Application proxy

+ New registration | Endpoints | Troubleshooting | Refresh | Download | Preview features

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications | **Owned applications** | Deleted applications

Start typing a display name or application (client) ID to filter these r... | Add filters

5 applications found

Display name ↑

EX	ExpertsLive-W365

Authentication (SPN)

The screenshot shows the Microsoft Azure portal interface. At the top, the navigation bar includes the Microsoft Azure logo and a search bar. Below the navigation bar, the breadcrumb trail reads: Home > NielsKok.Tech - LAB | App registrations > ExpertsLive-W365. The main heading is "ExpertsLive-W365 | API permissions", which is highlighted with a red box. Below the heading, there is a search bar, a refresh button, and a "Got feedback?" link. The left sidebar contains a "Manage" section with various options: Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions (highlighted with a red box), Expose an API, App roles, and Owners. The main content area is titled "Configured permissions" and includes a description: "Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)". Below this, there are two buttons: "+ Add a permission" and "✓ Grant admin consent for NielsKok.Tech - LAB". A table of configured permissions is displayed, which is also highlighted with a red box. The table has five columns: "API / Permissions name", "Type", "Description", "Admin consent requ...", and "Status". The table lists four permissions under the "Microsoft Graph (4)" group: "CloudPC.ReadWrite.All", "DeviceManagementConfiguratic", "DeviceManagementManagedDe", and "Directory.Read.All". Each permission is marked as "Granted for NielsKok.Te..." with a green checkmark.

Microsoft Azure

Search resources, services, and docs (G+)

Home > NielsKok.Tech - LAB | App registrations > ExpertsLive-W365

ExpertsLive-W365 | API permissions

Search

Refresh Got feedback?

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for NielsKok.Tech - LAB

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (4)				...
CloudPC.ReadWrite.All	Application	Read and write Cloud PCs	Yes	✓ Granted for NielsKok.Te... ...
DeviceManagementConfiguratic	Application	Read and write Microsoft Intune device configuration and ...	Yes	✓ Granted for NielsKok.Te... ...
DeviceManagementManagedDe	Application	Read and write Microsoft Intune devices	Yes	✓ Granted for NielsKok.Te... ...
Directory.Read.All	Application	Read directory data	Yes	✓ Granted for NielsKok.Te... ...

ExpertsLive-W365 - Microsoft A...

Pipelines - Run 20230511.3

W365Demo - Microsoft Azure

View Published Source - Microso...

+

dev.azure.com/nielskoktech/NielsKokTech/_build/results?buildId=215&view=results

dev.azure.com

Azure

DevOps

Preview Azure

HostPool Preview

Graph Explorer | Try...

Intune

Entra

Azure DevOps

nielskoktech

/

NielsKokTech

/

Pipelines

/

ExpertsLive_Auth_Demo

/

20230511.3

Search

NK

N

NielsKokTech

+

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Test Plans

Artifacts

Project settings

✓

#20230511.3 • TypoInVariable

ExpertsLive_Auth_Demo

Run new

ⓘ

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

Summary

Manually run by

NK

 Niels Kok

Repository and version

Time started and elapsed

Related

Tests and coverage

◆

 NielsKokTechTest

📅

 Today at 5:59 PM

🔗

 0 work items

🔗

 Get started

🌿

 main

🔗

 44c0b27a

🕒

 14s

📦

 0 artifacts

Jobs

Name	Status	Duration
<div>✓</div> RunningPowershellScript	Success	<div>🕒</div> 7s

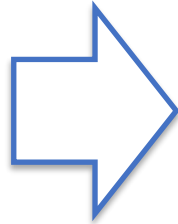
Flow



Authentication



Powershell
JSON Convert



Graph API



But wait! There's more!

Get-CloudPC

Get-CPCAzureNetworkConnection

Get-CPCCustomImage

Get-CPCGalleryImage

Get-CPCOrganizationSettings

Get-CPCProvisioningPolicy

Get-CPCRestorePoint

Get-CPCServicePlans

Get-CPCSupportedRegion

Get-CPCUserSettingsPolicy

Invoke-CPCEndGracePeriod

Invoke-CPCReprovision

Invoke-CPCRestore



Set-CPCProvisioningPolicyAssignment

Set-CPCUserSettingsPolicyAssignment

New-CPCAzureNetworkConnection

New-CPCCustomImage

New-CPCProvisioningPolicy

New-CPCUserSettingsPolicy

Remove-CPCAzureNetworkConnection

Remove-CPCCustomImage

Remove-CPCProvisioningPolicy

Update-CPCOrganizationSettings

Update-CPCUserSettingsPolicy

Update-CPCProvisioningPolicy

Tools

Graph X Ray

Microsoft Endpoint Manager admin center

Home > Devices | Windows 365 >

Create a provisioning policy

Windows 365

General Image Configuration Assignments **Review + create**

General

Name	Demo - Windows 365
Description	NielsKok.tech
Use single sign-on (preview)	Yes
Join type	Azure AD Join
Geography	European Union
Region	West Europe

Image

Image type	Gallery
	Windows 11 Enterprise + Microsoft 365 Apps
	22H2

Configuration

Language & Region	English (United States)
Additional Services	None

Assignments

Groups

MDM_UserGroup

Previous Create

Microsoft Graph X-Ray

Save script Clear session

GET /groups/653967ee-8c68-446f-93b7-e9db78eae72/photos/48x48/\$value

GET /groups/474494ef-3631-4d20-92fa-762b41f07396/photos/48x48/\$value

POST https://graph.microsoft.com/beta/deviceManagement/virtualEndpoint/provisi

```
Import-Module Microsoft.Graph.DeviceManagement.Administration

$params = @{
    DisplayName = "Demo - Windows 365"
    Description = "NielsKok.tech"
    ProvisioningType = "dedicated"
    ManagedBy = "windows365"
    ImageId = "MicrosoftWindowsDesktop_windows-ent-cpc_win11-22h2-ent-cpc-m365"
    ImageDisplayName = "Windows 11 Enterprise + Microsoft 365 Apps 22H2"
    ImageType = "gallery"
    MicrosoftManagedDesktop = @{
        Type = "notManaged"
        Profile = ""
    }
    EnableSingleSignOn = $true
    DomainJoinConfiguration = @{
        Type = "azureADJoin"
        RegionName = "westeurope"
        RegionGroup = "europeUnion"
        OnPremisesConnectionId = $null
    }
    WindowsSettings = @{
        Language = "en-US"
    }
}

New-MgDeviceManagementVirtualEndpointProvisioningPolicy -BodyParameter $params
```



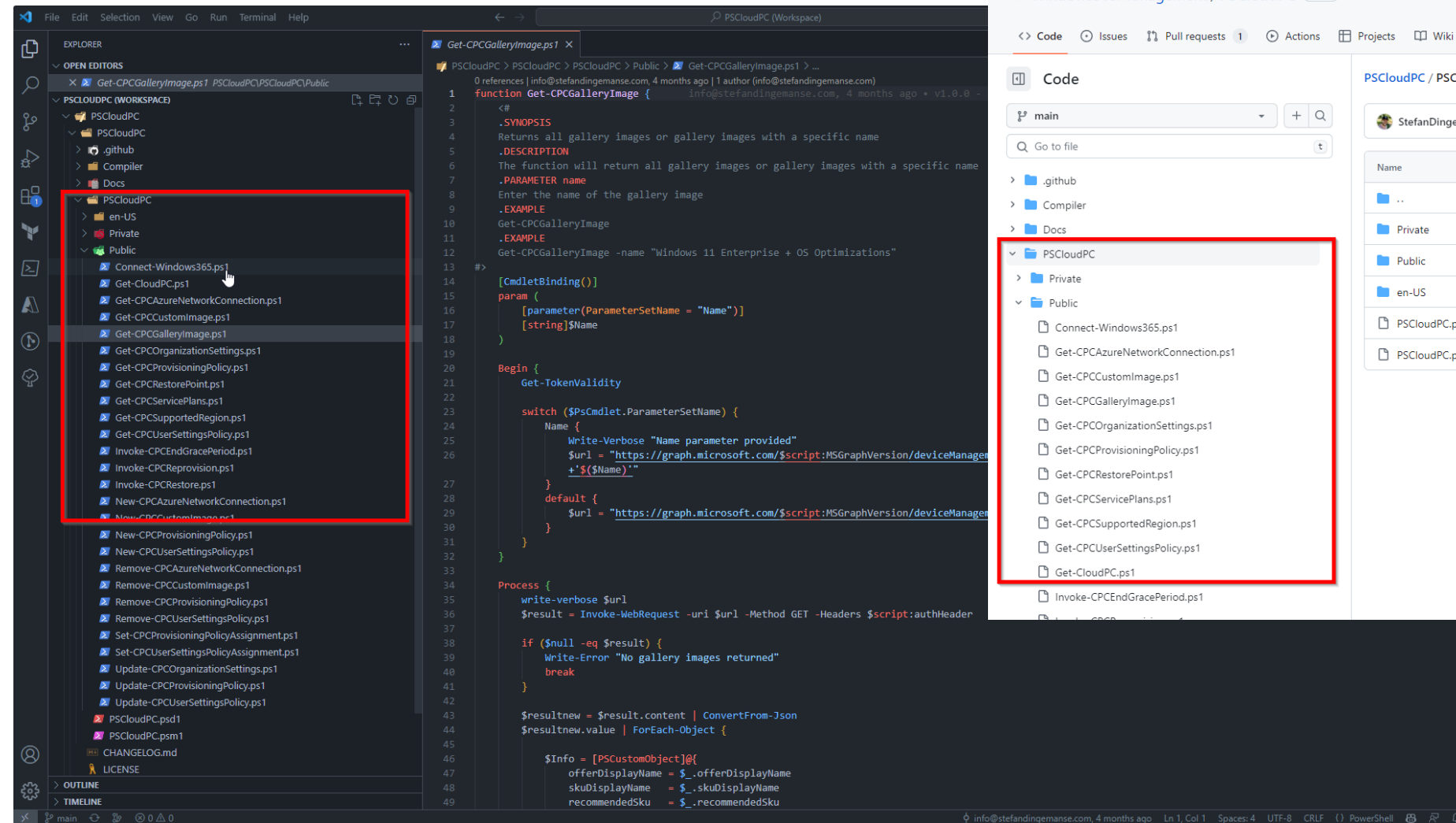
VS Code + GitHub

The screenshot displays the Microsoft Graph Explorer web application. The top navigation bar includes the Microsoft logo, the title "Microsoft Graph Dev Center", and links for Explore, Graph Explorer, Docs, API, Learn, Developer Program, and Support. On the right, it shows the tenant "NielsKok.Tech - LAB" and a search icon.

The left sidebar, titled "Graph Explorer", contains tabs for "Sample queries", "Resources", and "History". Below these is a search bar for "Search sample queries". A section titled "Getting Started (8)" lists several sample queries, each with a "GET" button. The bottom section, "Applications (8)", lists various application categories like "Batching (2)", "Compliance (beta) (10)", "Edge (4)", "Excel (7)", and "Extensions (7)".

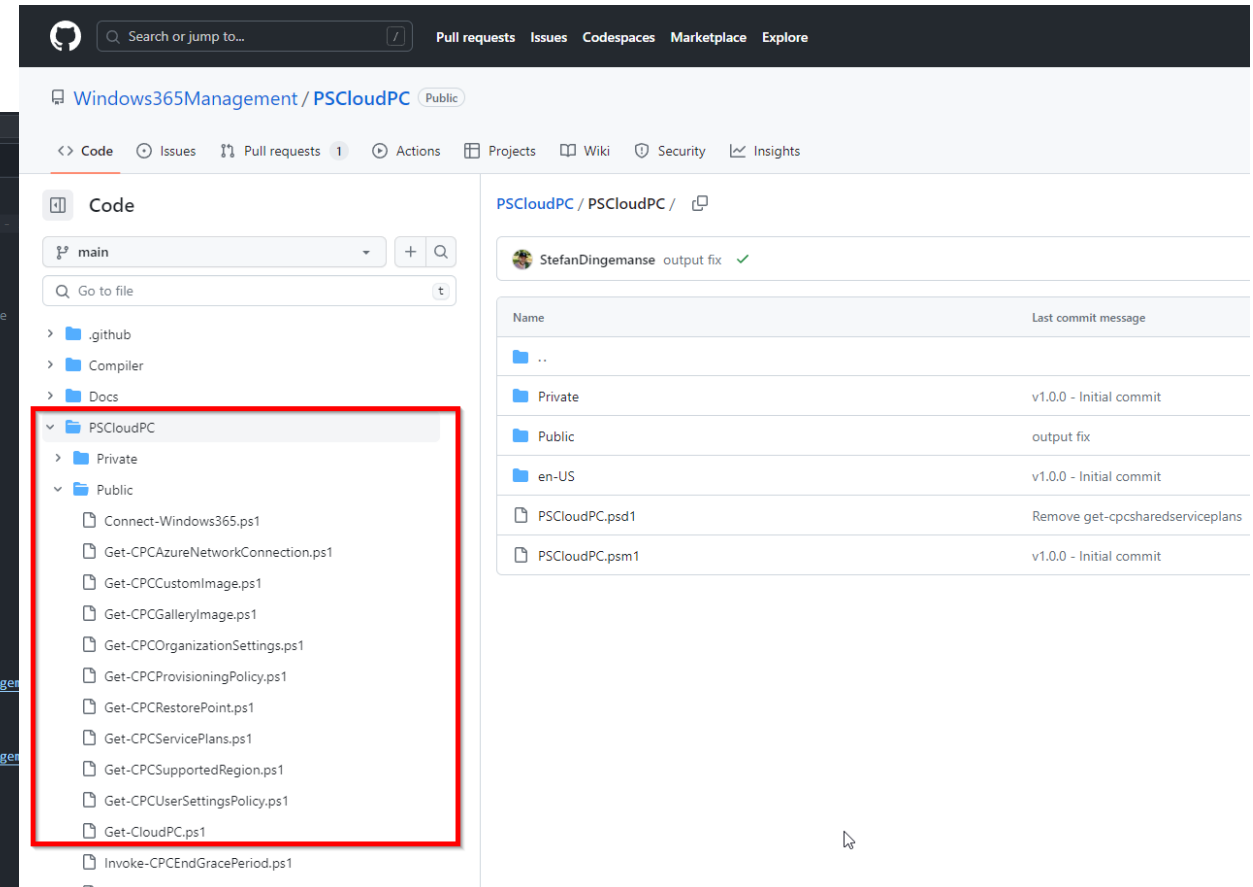
The main content area shows a GET request to the URL `https://graph.microsoft.com/beta/deviceManagement/managedDevices`. The request status is "OK - 200 - 428ms". Below the status bar, the "Response preview" tab is active, displaying a JSON response. The response includes metadata and a list of managed devices. One device is highlighted, showing details such as "id", "userId", "deviceName" (CPC-nt-6G1RRTLE), "ownerType", "managedDeviceOwnerType", "managementState", "enrolledDateTime", "lastSyncDateTime", "chassisType", "operatingSystem", "deviceType", "complianceState", "jailBroken", "managementAgent", "osVersion", "easActivated", and "easDeviceId".

Graph Explorer



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays a file tree for a project named 'PSCloudPC'. A red rectangle highlights a sub-folder named 'Public' which contains several PowerShell scripts, including 'Get-CPCGalleryImage.ps1'. The main editor window shows the content of 'Get-CPCGalleryImage.ps1'. The script is a PowerShell function that retrieves gallery images from the Microsoft Graph API. It includes a synopsis, description, parameter set, and a main body that uses 'Invoke-WebRequest' to fetch data and 'ConvertFrom-Json' to parse it. The script is authored by 'info@stefandingemane.com' and is version 1.0.0.

```
function Get-CPCGalleryImage {  
    <#  
    .SYNOPSIS  
    Returns all gallery images or gallery images with a specific name  
    .DESCRIPTION  
    The function will return all gallery images or gallery images with a specific name  
    .PARAMETER name  
    Enter the name of the gallery image  
    .EXAMPLE  
    Get-CPCGalleryImage  
    .EXAMPLE  
    Get-CPCGalleryImage -name "Windows 11 Enterprise + OS Optimizations"  
    #>  
    [CmdletBinding()]  
    param (  
        [parameter(ParameterSetName = "Name")]  
        [string]$Name  
    )  
    Begin {  
        Get-TokenValidity  
    }  
    switch ($PsCmdlet.ParameterSetName) {  
        Name {  
            Write-Verbose "Name parameter provided"  
            $url = "https://graph.microsoft.com/$script:MSGraphVersion/deviceManagement/galleryimages?filter=name eq '$($Name)'"  
        }  
        default {  
            $url = "https://graph.microsoft.com/$script:MSGraphVersion/deviceManagement/galleryimages"   
        }  
    }  
    Process {  
        write-verbose $url  
        $result = Invoke-WebRequest -uri $url -Method GET -Headers $script:authHeader  
        if ($null -eq $result) {  
            Write-Error "No gallery images returned"  
            break  
        }  
        $resultnew = $result.content | ConvertFrom-Json  
        $resultnew.value | ForEach-Object {  
            $Info = [PSCustomObject]{  
                offerDisplayName = $_.offerDisplayName  
                skuDisplayName = $_.skuDisplayName  
                recommendedSku = $_.recommendedSku  
            }  
        }  
    }  
}
```



The screenshot shows the GitHub repository page for 'Windows365Management/PSCloudPC'. The repository is public and has a commit history. A red rectangle highlights the 'PSCloudPC' folder in the file tree, which contains the same PowerShell scripts seen in the Visual Studio Code screenshot. The commit history table on the right shows the following commits:

Name	Last commit message
..	
Private	v1.0.0 - Initial commit
Public	output fix
en-US	v1.0.0 - Initial commit
PSCloudPC.psd1	Remove get-cpcsharedserviceplans
PSCloudPC.psm1	v1.0.0 - Initial commit

PowerShell Gallery

Welcome to the PowerShell Gallery

The central repository for sharing and acquiring PowerShell code including PowerShell modules, scripts, and DSC resources.

Search PowerShell packages:



11,546

Unique Packages

8,347,554,372

Total package downloads

217,742

Total packages



Learn

[What is PowerShell Gallery?](#)

Learn why the PowerShell Gallery is the most used resource for sharing and acquiring PowerShell code.

[Getting started](#)

Learn how to install and set up the PowerShellGet module, which is required in order to download packages from the Gallery.

Top package downloads (last 6 weeks):

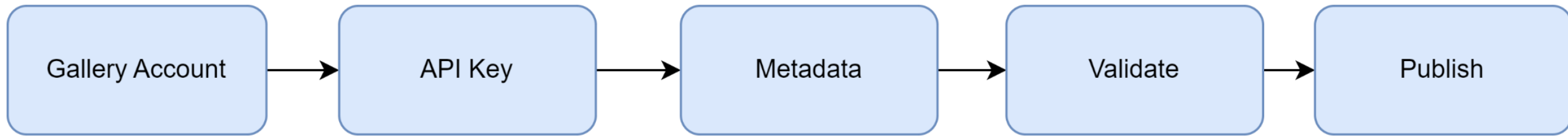
Package	Downloads
PackageManagement	28,141,457
PSWindowsUpdate	27,331,425
PowerShellGet	15,238,677
Az.Accounts	11,950,276
DellBIOSProvider	11,686,066

PowerShell Gallery

- Scripts
- Modules
- DSC resources



Requirements



Testing the code

- Pester
- PSScriptAnalyser

```
✓ Pester Module check

1 ▶ Run $testLocation = (Join-Path -Path "../Compiler" -ChildPath "Module.tests.ps1")
5 Pester v5.4.1
6
7 Starting discovery in 1 files.
8 Discovery found 9 tests in 268ms.
9 Running tests.
10
11 Running tests from '/home/runner/work/PSCloudPC/PSCloudPC/Compiler/Module.tests.ps1'
12 Describing PSCloudPC Global module tests
13 Context Module Setup
14 [+] PSCloudPC has the root module PSCloudPC.psm1 95ms (55ms|41ms)
15 [+] PSCloudPC has the a manifest file of PSCloudPC.psm1 7ms (6ms|2ms)
16 [+] PSCloudPC folder has functions folder 5ms (3ms|2ms)
17 [+] PSCloudPC folder has private functions folder 10ms (9ms|2ms)
18 [+] PSCloudPC root module should be in 39ms (35ms|4ms)
19 [+] PSCloudPC should have a semver version 6ms (5ms|2ms)
20 [+] PSCloudPC version should be greater than PSGallery 3.29s (3.29s|2ms)
21 [+] PSCloudPC project URL should be reachable 465ms (461ms|4ms)
22 [+] PSCloudPC is valid PowerShell code 17ms (15ms|1ms)
23 Tests completed in 4.61s
24 Tests Passed: 9, Failed: 0, Skipped: 0 NotRun: 0
```


Pester

- Module Check
- Functions check
- Files check

```
$modulePath = Join-Path -Path (Join-Path "../.." -ChildPath "PScloudPC") -ChildPath "PScloudPC"
$moduleFunctions = (Import-PowerShellDataFile (Join-Path -Path $modulePath -ChildPath "PScloudPC.psd1")).Functions

BeforeAll {
    $modulePath = Join-Path -Path (Join-Path "../.." -ChildPath "PScloudPC") -ChildPath "PScloudPC"
    Write-Host $modulePath
    Write-Host "In before $modulePath"
    (Join-Path -Path $modulePath -ChildPath "Public")
    $psFiles = (Get-ChildItem -Path (Join-Path -Path $modulePath -ChildPath "Public")).BaseName
}

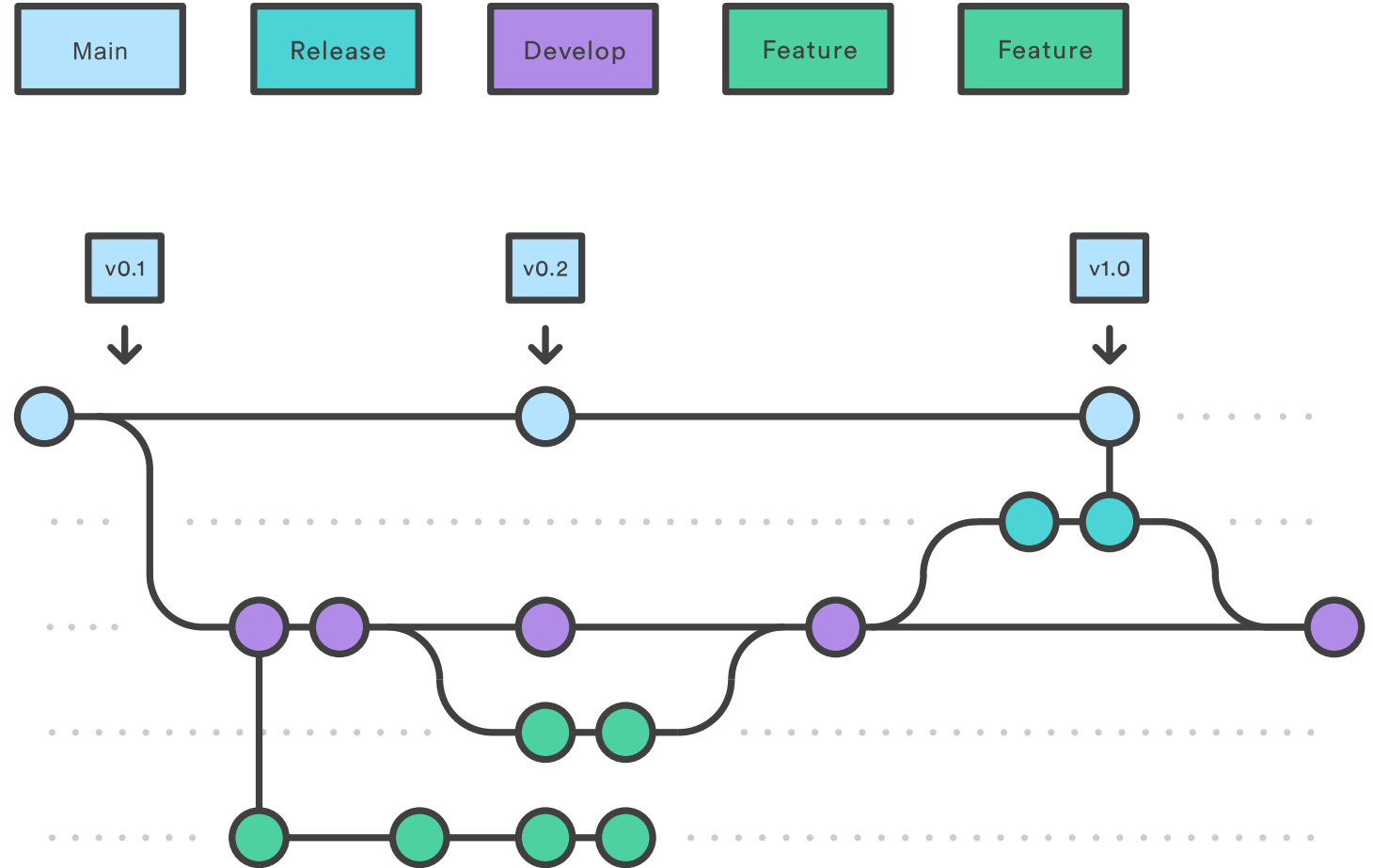
Run tests | Debug tests
Describe "Analyze code" -ForEach @(
    foreach ($function in $moduleFunctions) {
        @{
            function = $function
            helpInfo = Get-Help $function
            IgnoreRules = @('PSUseApprovedVerbs')
            fileobj = $file
        }
    }
){
    It "<function> should exist in file base" {
        $function -in $psFiles | Should -Be $true
    }

    It "<function> has a synopsis that is valid" {
        if ( $helpInfo.synopsis -eq $null ) {
            $helpInfo.synopsis | Should -not -Be $null -Because 'every ps1 file should have a synopsis block'
        }
        elseif ( $helpInfo.synopsis -eq 'Short description' ) {
            $helpInfo.synopsis | Should -not -Be 'Short description' -Because 'thats just lazy'
        }
    }

    It "<function> has a description that is valid" {
        if ( $helpInfo.description -eq $null ) {
            $helpInfo.description | Should -not -Be $null -Because 'every ps1 file should have a synopsis block'
        }
        elseif ( $helpInfo.description -eq 'Long description' ) {
            $helpInfo.description | Should -not -Be 'Long description' -Because 'thats just lazy'
        }
    }
}
```

Branching strategy

- Gitflow



Github actions – Release flow



Github actions – Release flow

The screenshot shows a GitHub Actions workflow run for the job 'Build and Release Binaries'. The workflow is titled 'Merge pull request #41 from Windows365Management/feature/37-feature-r... #9'. The 'Build' job is highlighted in the 'Jobs' section on the left, indicating it is the current step. The 'Build' job status is 'succeeded 4 days ago in 8s'. The 'Run details' section on the right lists the steps of the job, all of which are marked as successful with green checkmarks.

← Build and Release Binaries

✓ Merge pull request #41 from Windows365Management/feature/37-feature-r... #9

Summary

Jobs

- ✓ Build
- ✓ Publish

Run details

- Usage
- Workflow file

Build

succeeded 4 days ago in 8s


- > ✓ Set up job
- > ✓ Check out code
- > ✓ Get the version
- > ✓ Get previous tag version
- > ✓ Print the version
- > ✓ Build release changelog
- > ✓ Create Release
- > ✓ Post Check out code
- > ✓ Complete job





Documentation

PSCloudPC.com

- Mkdocs
- Static website
- Automatic

**PSCloudPC - Windows 365 Management PowerShell Module**

 Search

[Home](#) [Getting started](#) [User Guide](#) [Contributing](#) [About](#)

User Guide

Cmdlets

[Connect-Windows365](#)

Get-CPAzureNetworkConnection

Get-CPCCustomImage

Get-CPCGalleryImage

Get-CPCOrganizationSettings

Get-CPCProvisioningPolicy

Get-CPCRestorePoint

Get-CPCServicePlans

Get-CPCSupportedRegion

Get-CPCUserSettingsPolicy

Get-CloudPC

Invoke-CPCEndGracePeriod

Invoke-CPCReprovision

Invoke-CPCRestore

New-CPAzureNetworkConnection

New-CPCCustomImage

Connect-Windows365

SYNOPSIS

Connect to Windows 365 via Powershell

SYNTAX

```
Connect-Windows365 [[-AuthType] <String>] [[-ClientSecret] <String>] [-TenantID] <String> [[-ClientID] <String>] [<CommonParameters>]
```

DESCRIPTION

Connect to Windows 365 via Powershell via Interactive Browser or Service Principal

EXAMPLES

Table of contents

SYNOPSIS

SYNTAX

DESCRIPTION

EXAMPLES

EXAMPLE 1

EXAMPLE 2

PARAMETERS

-AuthType

-ClientSecret

-TenantID

-ClientID

CommonParameters

INPUTS

OUTPUTS

NOTES

RELATED LINKS

GitHub project site

Windows 365 Cloud PC Management PowerShell Module

Project information

CONTRIBUTORS	4	LICENSE	MIT	ISSUES	1 OPEN
--------------	---	---------	-----	--------	--------

Module statistics

LATEST	V1.0.5	POWERSHELL	7.0
--------	--------	------------	-----

PowerShell Gallery statistics

POWERSHELL GALLERY	V1.0.6	DOWNLOADS	230
--------------------	--------	-----------	-----

Description

This PowerShell module allows you to manage your Windows 365 environment from the command line. It provides a set of cmdlets that allow you to perform various tasks, such as creating, modifying and deleting policies, managing Cloud PCs, and more.

Getting Started

```
Install-Module -Name PScloudPC -Verbose
```

Then import the module into your session

```
Import-Module PScloudPC -Verbose -Force
```




The moment of truth



The moment of truth

The Result



229
Downloads

0
Downloads of 1.0.6
[View full stats](#)

5/15/2023
Last Published

Info
[Project Site](#)
[License Info](#)
[Contact Owners](#)
[Contact Support](#)
[Edit Package](#)
[Manage Owners](#)
[Unlist Module](#)

Search PowerShell packages:

Az, etc...

PSCloudPC 1.0.6

This PowerShell module allows you to manage your Windows 365 environment from the command line. It provides a set of cmdlets that allow you to perform various tasks, such as creating, modifying and deleting policies, managing Cloud PCs, and more.

Minimum PowerShell version
7.2

Installation Options

Install Module

Azure Automation

Manual Download

Copy and Paste the following command to install this package using PowerShellGet [More Info](#)

PS> Install-Module -Name PSCloudPC

Author(s)
Stefan Dingemanse Niels Kok

Copyright
(c) All rights reserved.

> Package Details

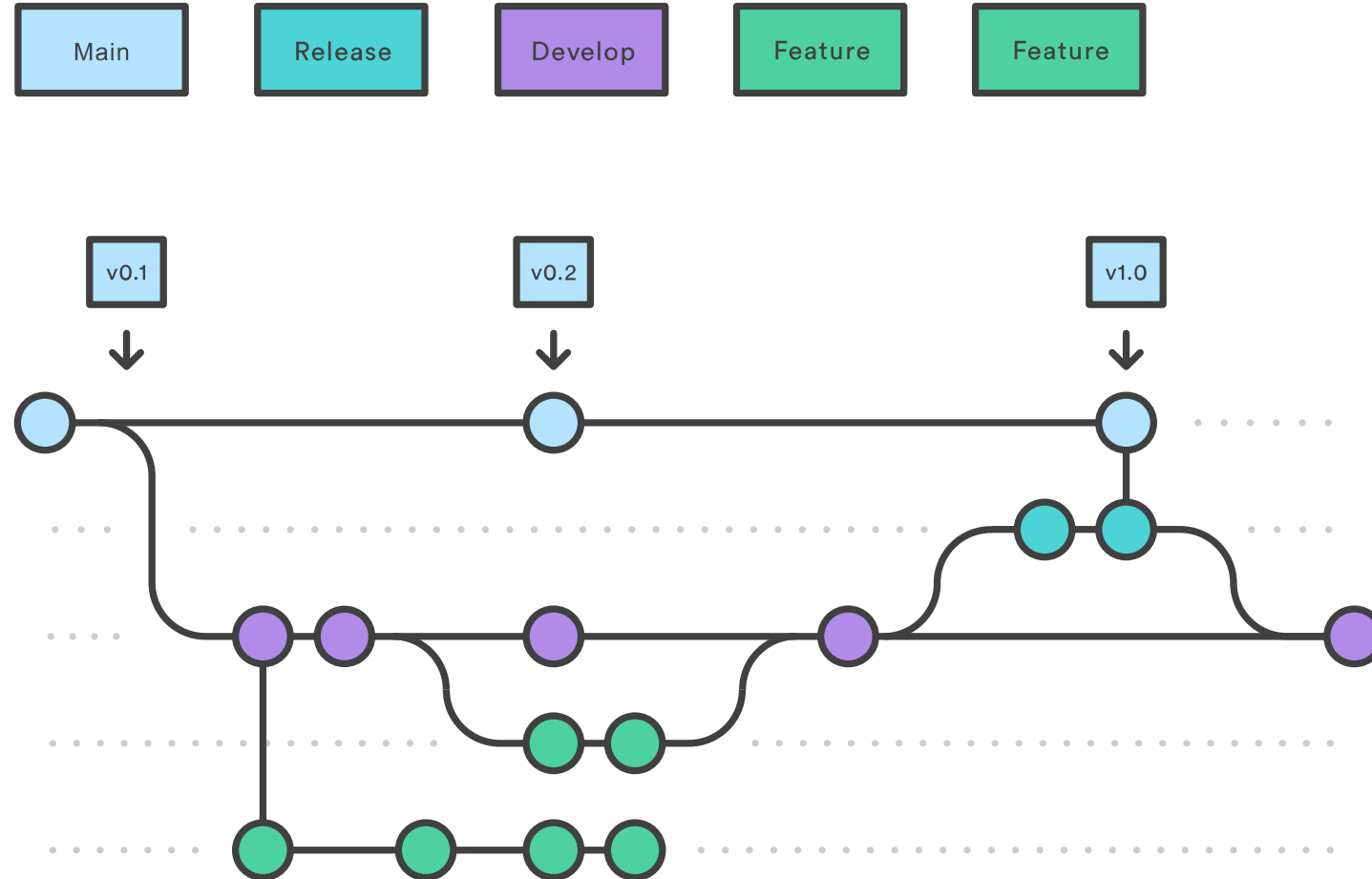
> FileList

Version History

Version	Downloads	Last updated	Status
1.0.6 (current version)	0	a few seconds ago by SDingemanse	Listed
1.0.5	32	15 days ago by SDingemanse	Listed
1.0.4	123	4 months ago by SDingemanse	Listed
1.0.3	25	4 months ago by SDingemanse	Listed
1.0.2	42	4 months ago by SDingemanse	Listed

+ Show more

Branching strategy



PSCloudPC.com

- Mkdocs
- Static website
- Automatic



The screenshot shows the PSCloudPC website, which is a static site generated by Mkdocs. The website has a blue header with the title "PSCloudPC - Windows 365 Management PowerShell Module" and a search bar. The main content area is divided into three columns: a left sidebar with links to "Home" and "About PSCloudPC", a central main content area, and a right sidebar with a "Table of contents" link and a list of sections: "Description", "Getting Started", "Connect to Windows 365 Cloud PC RestAPI", "Cmdlets", and "Troubleshooting". The main content area features a large heading "Windows 365 Cloud PC Management PowerShell Module", a "Description" section with text about the PowerShell module, and a "Getting Started" section with two code blocks for installing and importing the module. A large blue PowerShell icon is positioned to the right of the description text.

PSCloudPC - Windows 365 Management PowerShell Module

Home Getting started User Guide Contributing About

Home
About PSCloudPC

Windows 365 Cloud PC Management PowerShell Module

Description

This PowerShell module allows you to manage your Windows 365 environment from the command line. It provides a set of cmdlets that allow you to perform various tasks, such as creating, modifying and deleting policies, managing Cloud PCs, and more.

Getting Started

```
Install-Module -Name PSCloudPC -Verbose
```

Then import the module into your session

```
Import-Module PSCloudPC -Verbose -Force
```

Table of contents

- Description
- Getting Started
- Connect to Windows 365 Cloud PC RestAPI
- Cmdlets
- Troubleshooting

DEMO

Roadmap

Filters

is:issue is:open

Labels 10

Milestones 0

New issue

3 Open 3 Closed

Author Label Projects Milestones Assignee Sort

[Feature Request]: Support export and import of Windows 365 configurations and settings feature

#38 opened last week by Windows365Management

[Feature Request]: Add support for enabling users to reset their Cloud PCs within the user settings policy feature

#37 opened 2 weeks ago by Windows365Management

[Feature Request]: Support Frontline Service Plan feature

#36 opened 2 weeks ago by Windows365Management

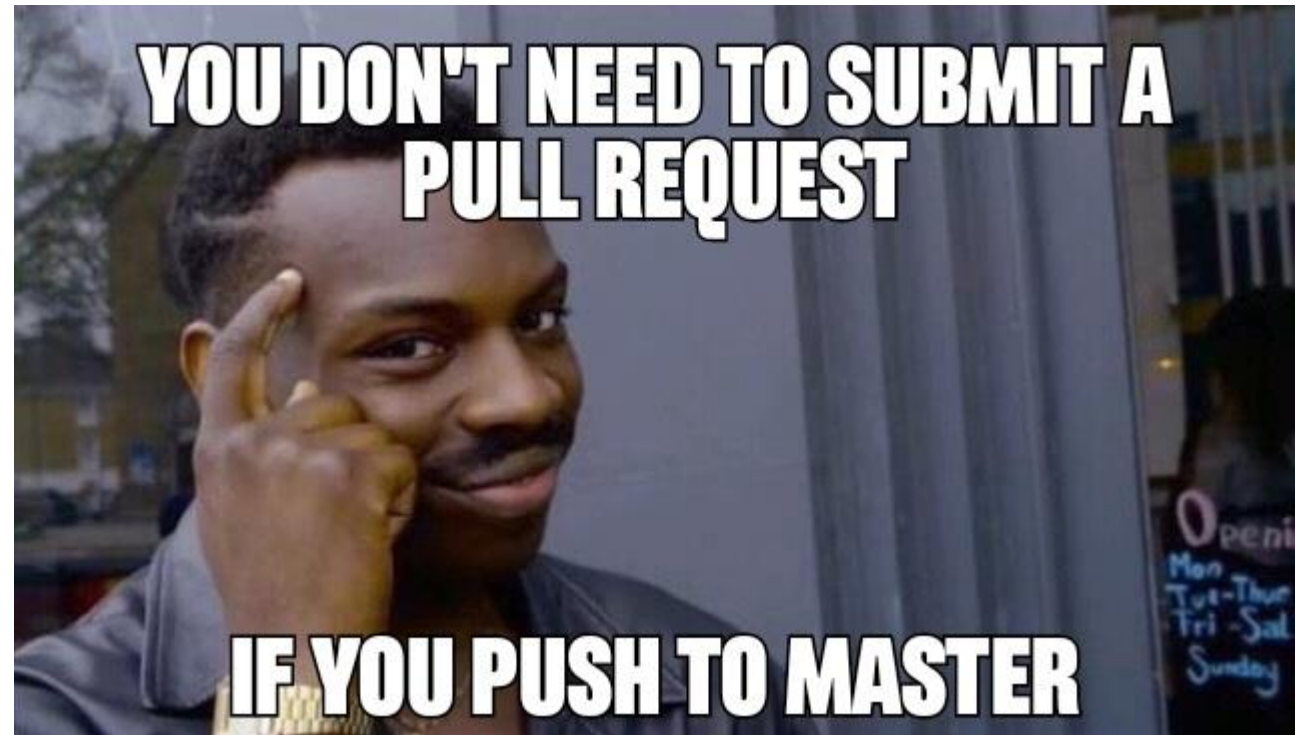




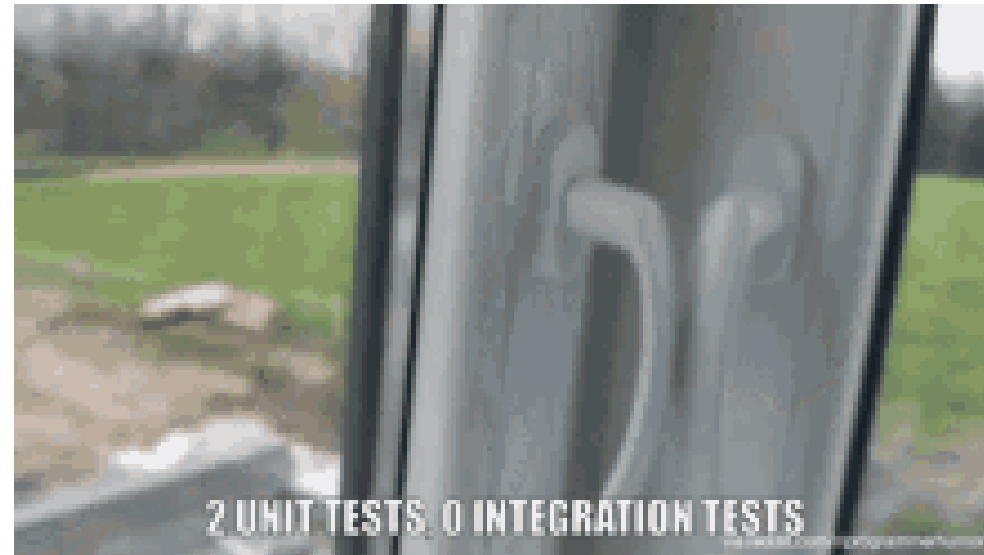
Lessons learned

Branching

- Protect your branches
- Branching strategy



Testing, testing, testing



Call to Action!

- Feedback
- Feature requests / issues



More than 1300 downloads



WORKPLACEDUDES