Modern Workplace Advanced

Azure Hands On Lab November 15th 2019

v1.0 by : Gino van Essen
Stephan van de Kruis
Gido Veekens

2tCloud

COPACO

Content

Introduction

Estimated time to complete this lab

120 minutes

Objectives

During this lab, you will learn how to get started with Azure to;

-

Prequisites

To complete this course, you will be needing;

- Laptop/computer with Internet browser and WiFi connected
- Account with an Azure CSP Subscription
- A Microsoft 365 business subscription

Materials

All student materials are available for download here:

https://github.com/Copaco/handsonlab/

Activity 1 : Use Secure Score and Passwordless Sign-in to secure your tenant

Estimated time to complete this activity

45 minutes

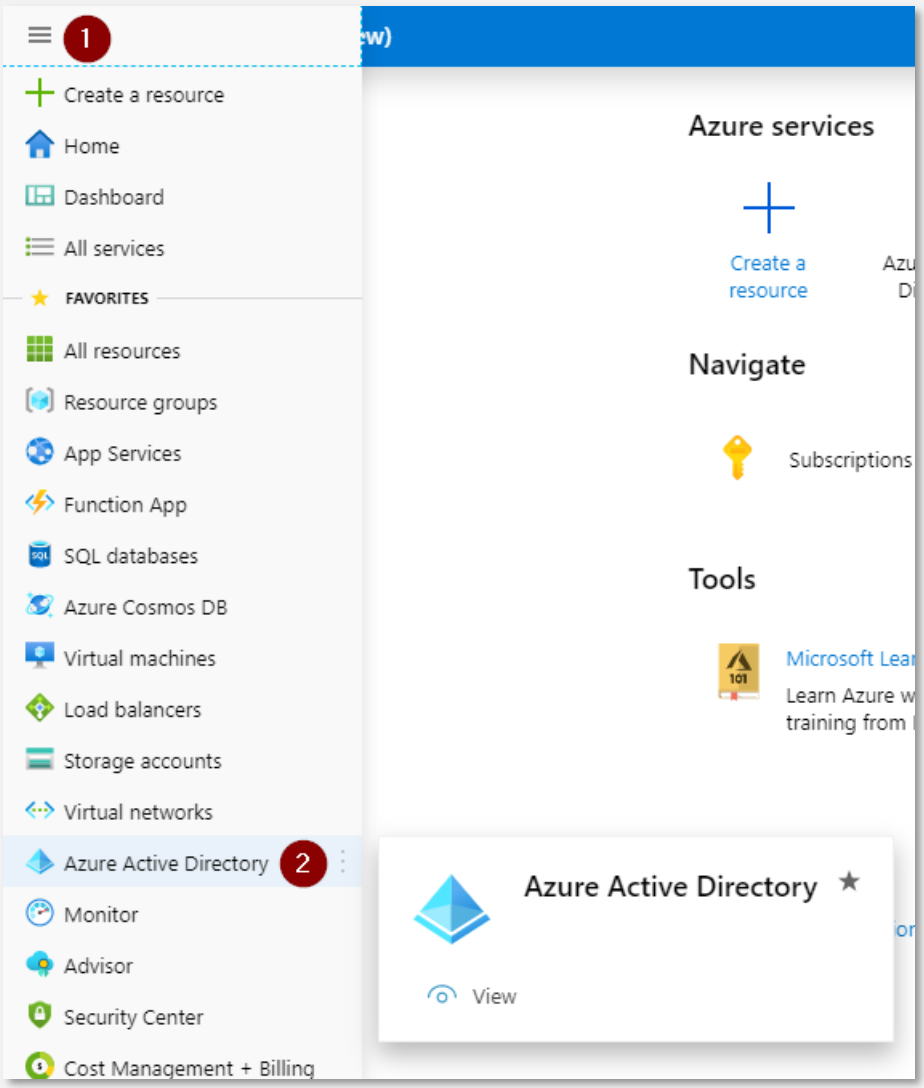
Objectives

In this activity, you will configure the components necessary to perform this lab;

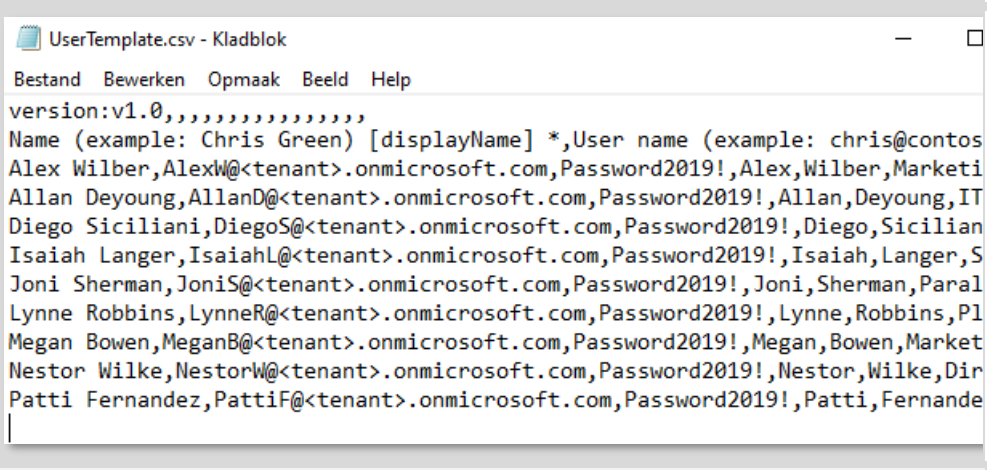
- Use Microsoft Secure Score to secure your Microsoft 365 tenant
- Enable baseline policies for MFA
- Enable passwordless authentication for users
- Troubleshoot Conditional Access

Exercise 1a : Setting up the user accounts

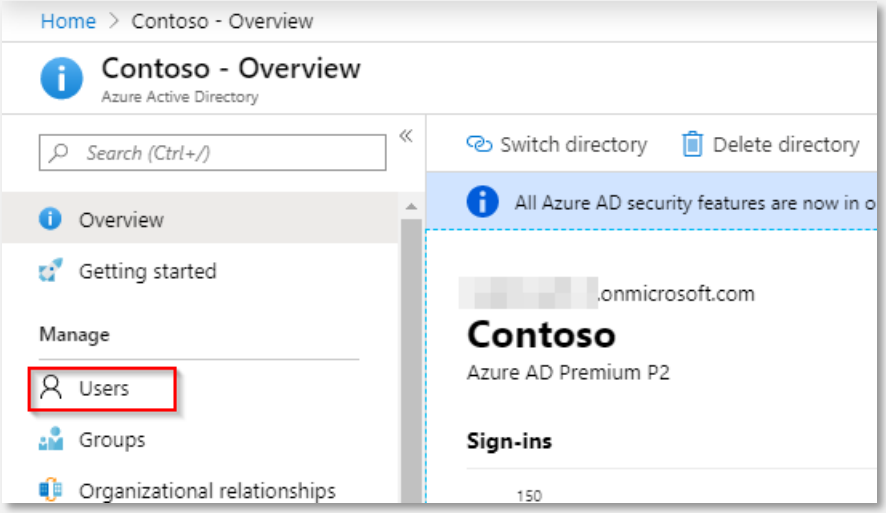
1) Using your *Work Account*, you can sign into the Azure Portal using: https://portal.azure.com	

2) Browse to the Azure Active Directory	
3) Note the tenant name from the Azure Active Directory blade	<tenant>.onmicrosoft.com
4) From GitHub, *download* **UserTemplate.csv** and *save* it to your local computer.	

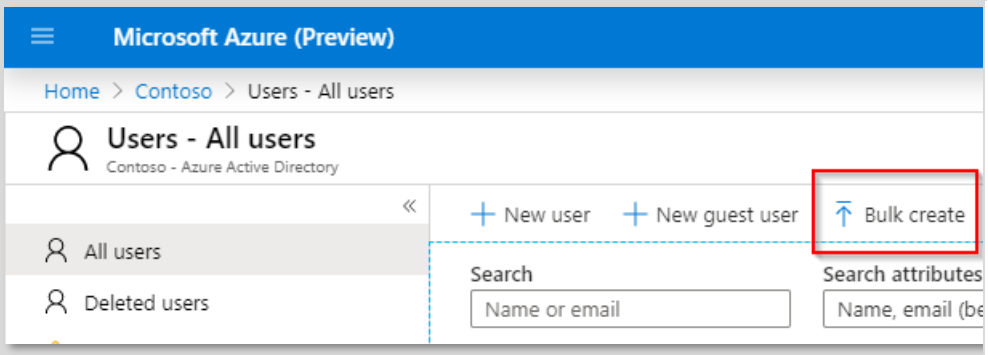
5) *Edit* the CSV-file using Notepad. Find and replace *<tenant>* so it corresponds with the tenant you're using. **Save** your changes.

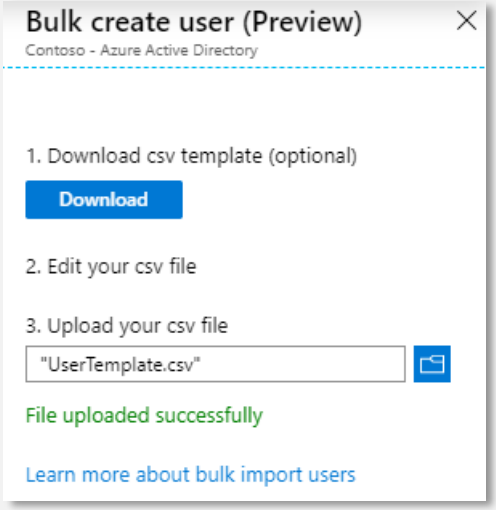
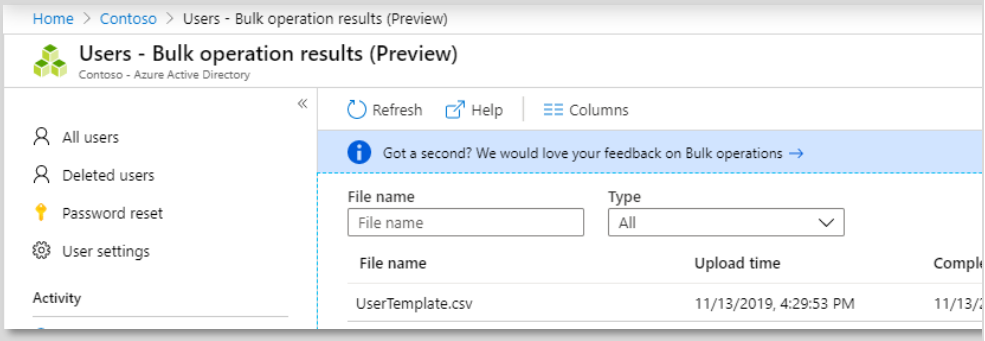
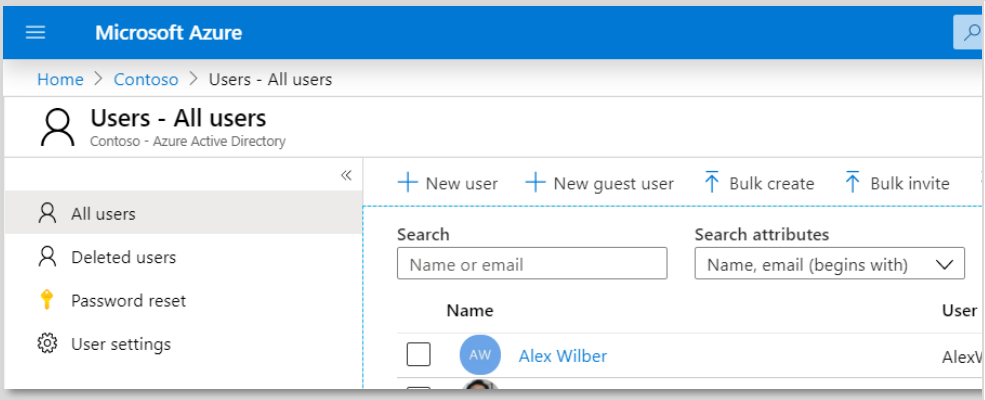


6) Open the *Users* blade

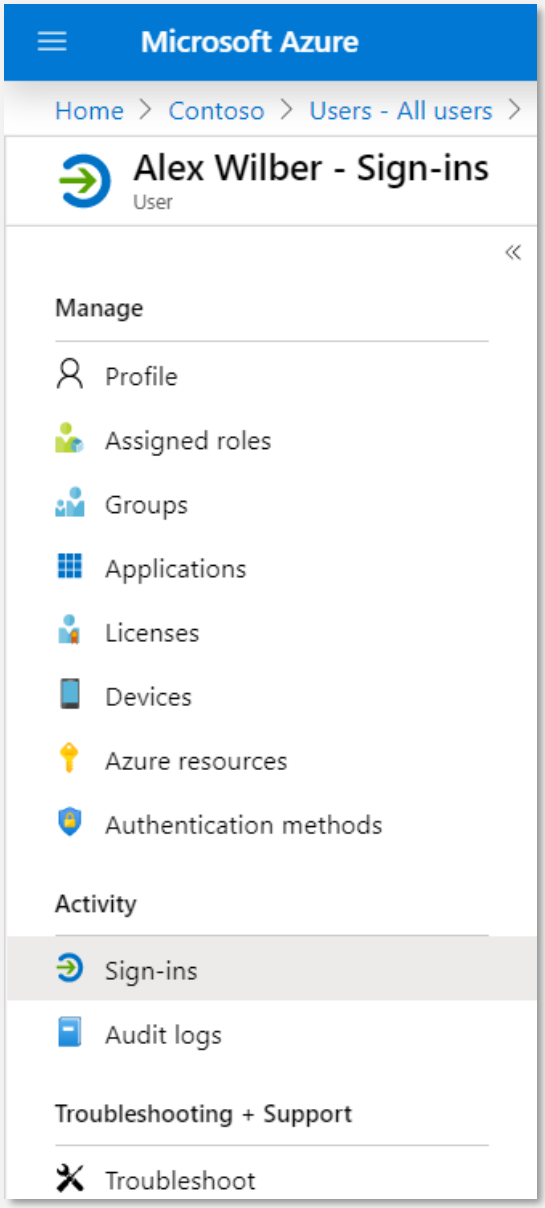


7) Choose *Bulk create*

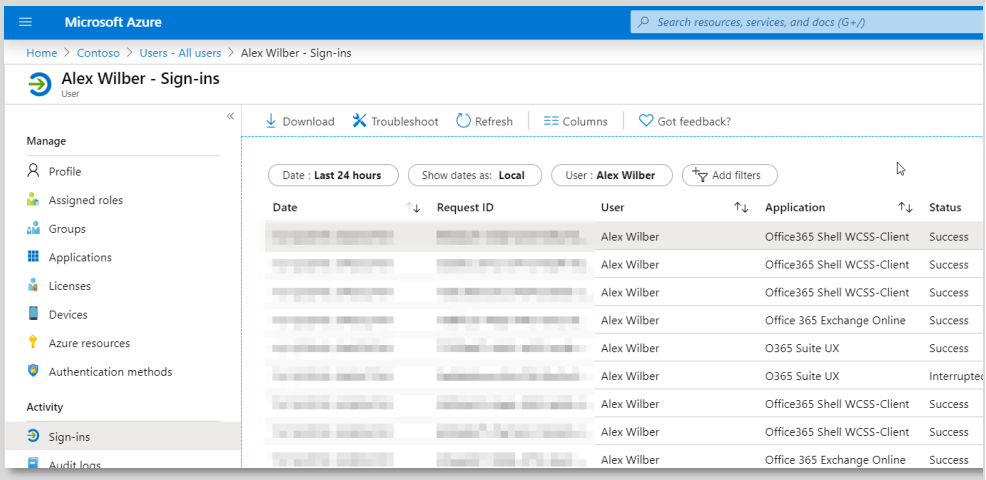


8) Upload the edited CSV and Submit.	
9) Click to watch the status of the import job. Make sure it's successful before you proceed. Refresh the blade for status update,	
10) Open a new InPrivate browser window	
11) Browse to Office.com and sign in with user Alex Wilber. Take note of the current time.	Username *AlexW@<tenant>.onmicrosoft.com* & Password *Password2019!*
12) Switch back to the regular browser with the opened Azure Portal	
13) Browse to the Azure Active Directory and then Users	

14) Open Alex Wilber and open Sign-ins from the navigation bar.



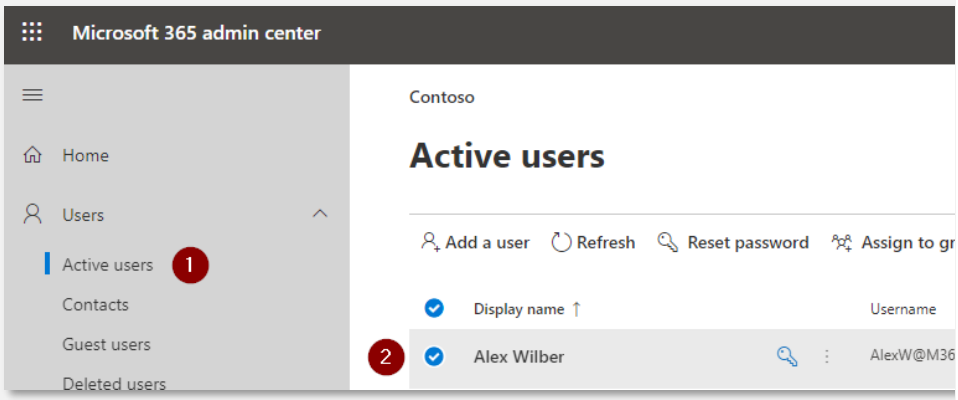
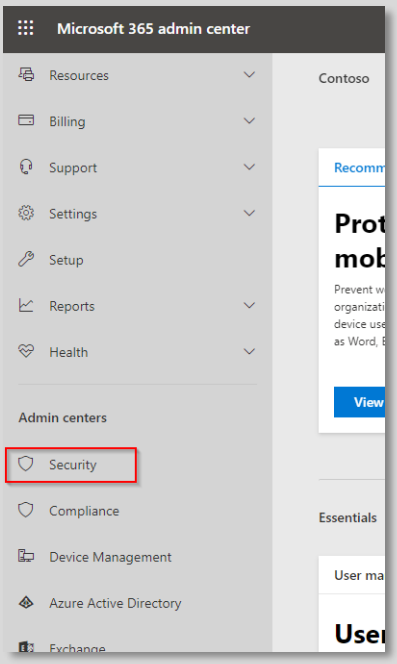
15) Select the most recent sign-in, which corresponds with the sign-in performed in step 11. There's some latency, so wait and refresh if the sign-in won't show.



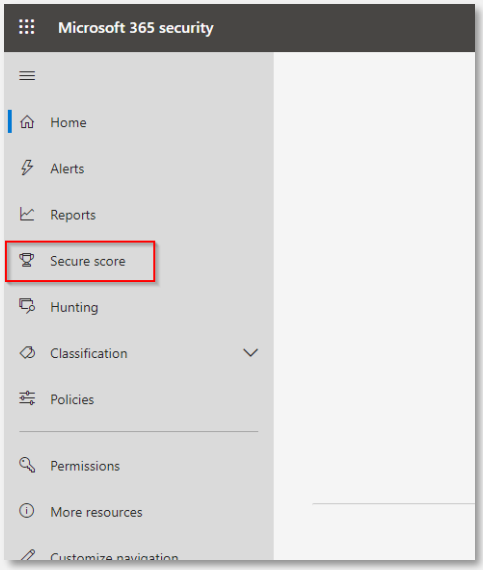
16) Take note of the IP-address that

was used to perform the sign-in. You will need this later.	
17) Your done for this exercise!	

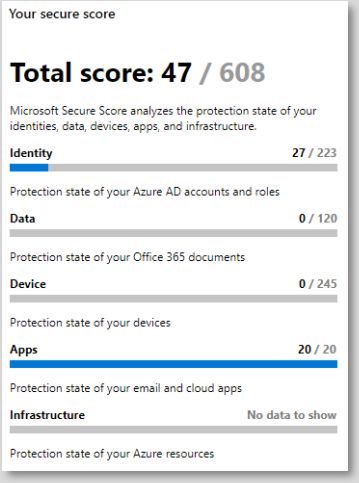
Exercise 1b : Use Secure Score to harden your security

1) Using your *Work Account*, you can sign into the Azure Portal using: https://admin.microsoft.com	
2) Browse to Users. Select user Alex from the list and assign the Microsoft 365 license to the user.	
3) From the Microsoft 365 Admin Center navigation panel, browse to the *Security Center*.	

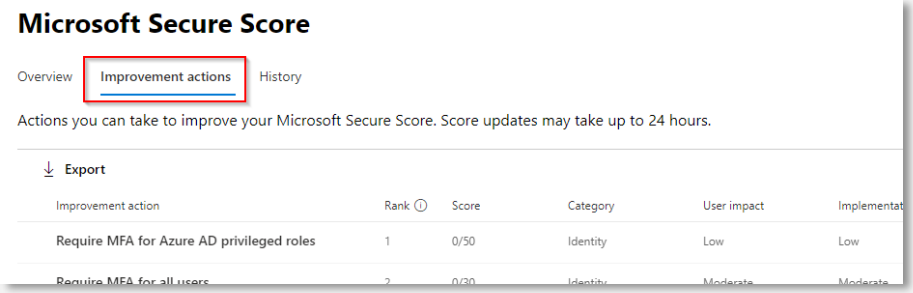
4) Open *Secure Score* from the navigation panel.



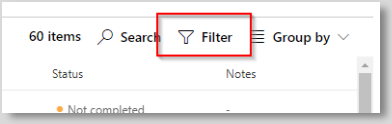
5) Review the score overview.



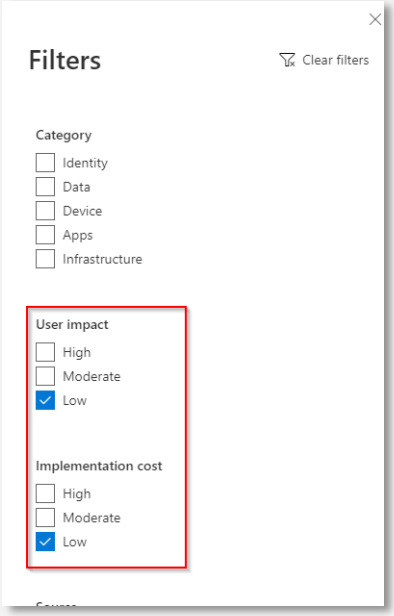
6) Open the *Improvement Actions* tab.



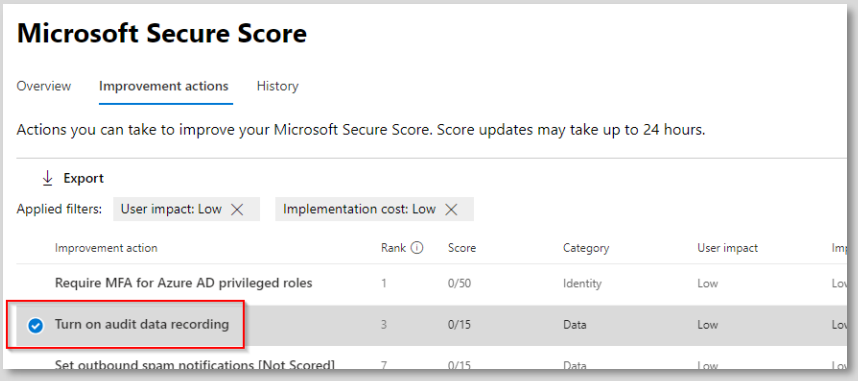
7) Click the *Filter* icon.



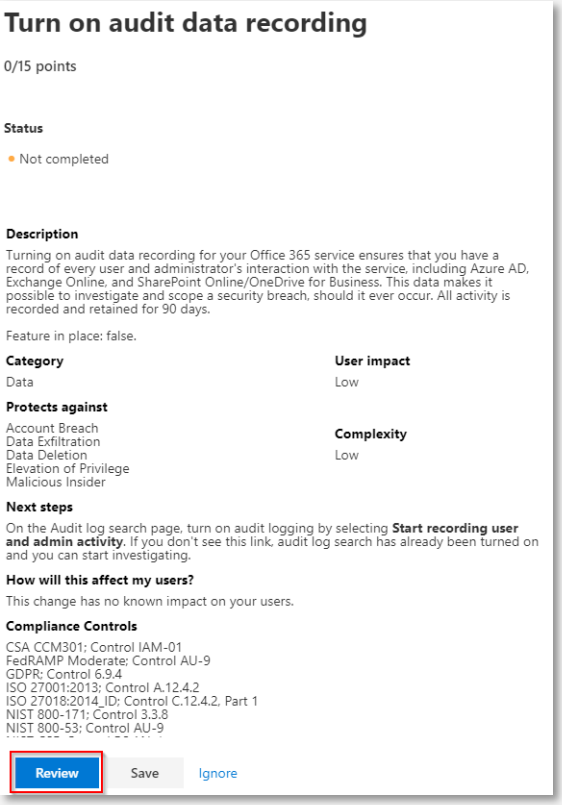
8) Make sure you only select the items with minimal *User impact* and minimal *Implementation Cost*. Click **Apply**.



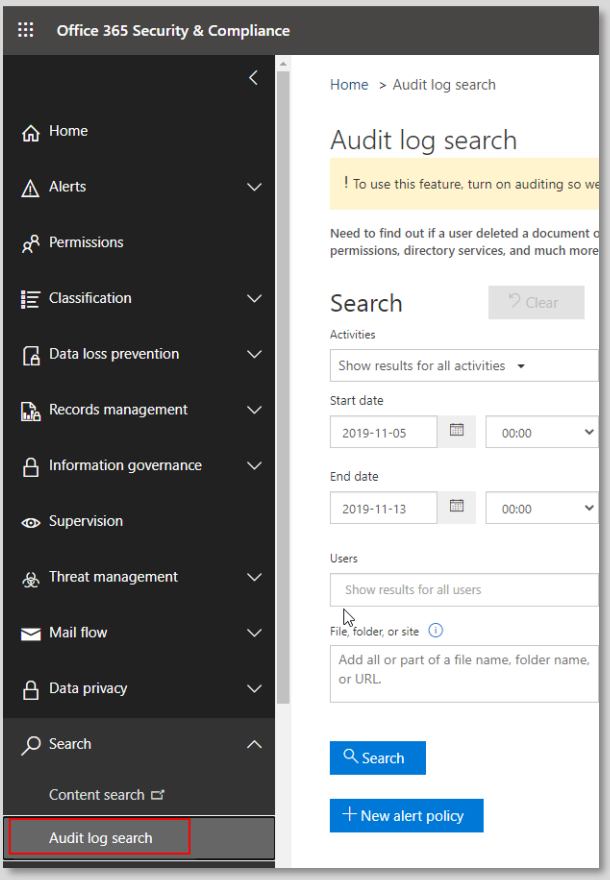
9) Find and select *Turn on audit data recording*

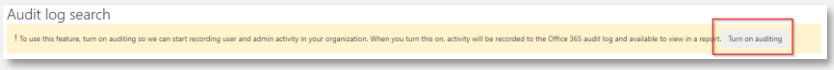
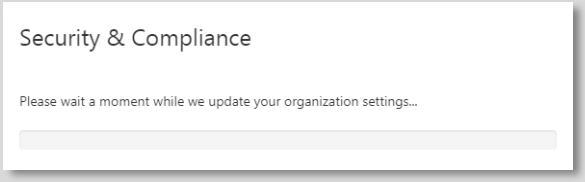


10) Review the *Description* and *Next steps* from the sidepanel. **Review** the setting. The *Security & Compliance Center* will be opened.



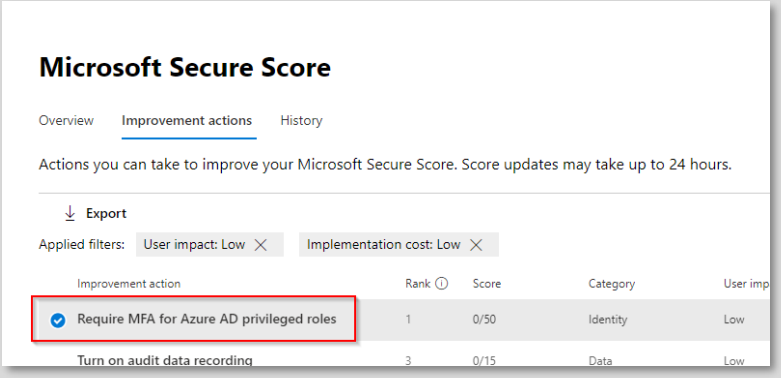
11) Browse to the *Audit log search*.



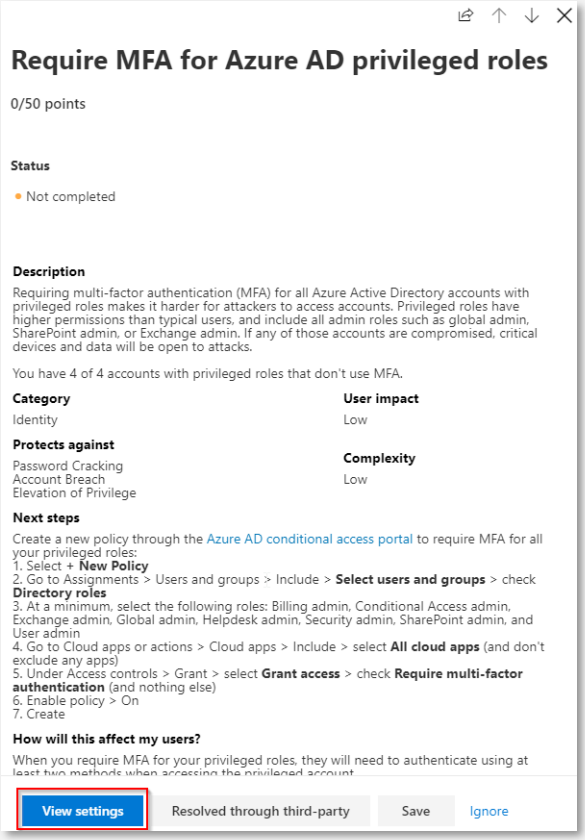
12) If not enabled already, **Turn on auditing**.	
13) Wait for the change to be effective. This could take a few minutes. Close the browser tab with the *Compliance and Security Center*.	
14) You can check Secure Score to see an updates score. However, this usually takes 24 hours to show. Come back later to see how your score has changed from the actions taken. Proceed with the lab for now.	
15) You're done with this exercise!	

Exercise 1c: Implement Passwordless Sign-in

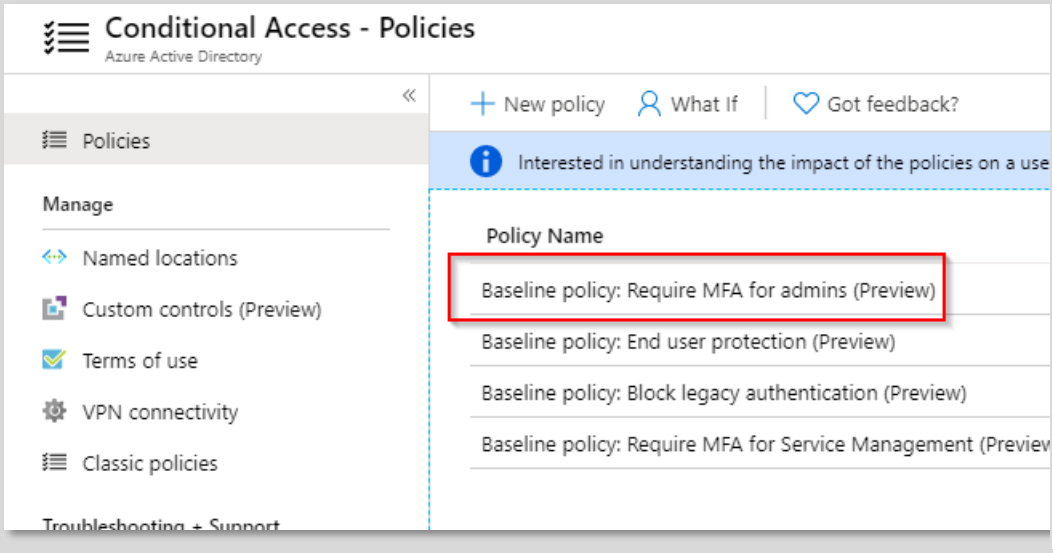
1) From the *Secure Score Improvement Actions*, select the *Require MFA for Azure AD privileged roles*.



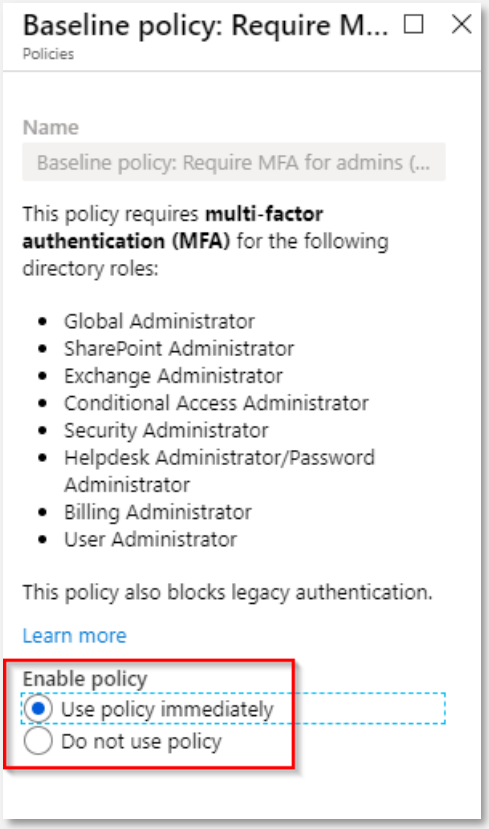
2) Review the details and click the **View settings** button.



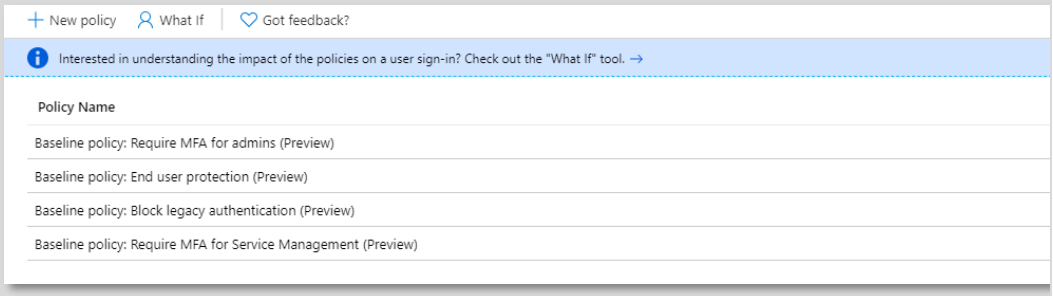
3) You'll be taken to the Conditional Access Policies. Open the Baseline policy for Require MFA for Admins.



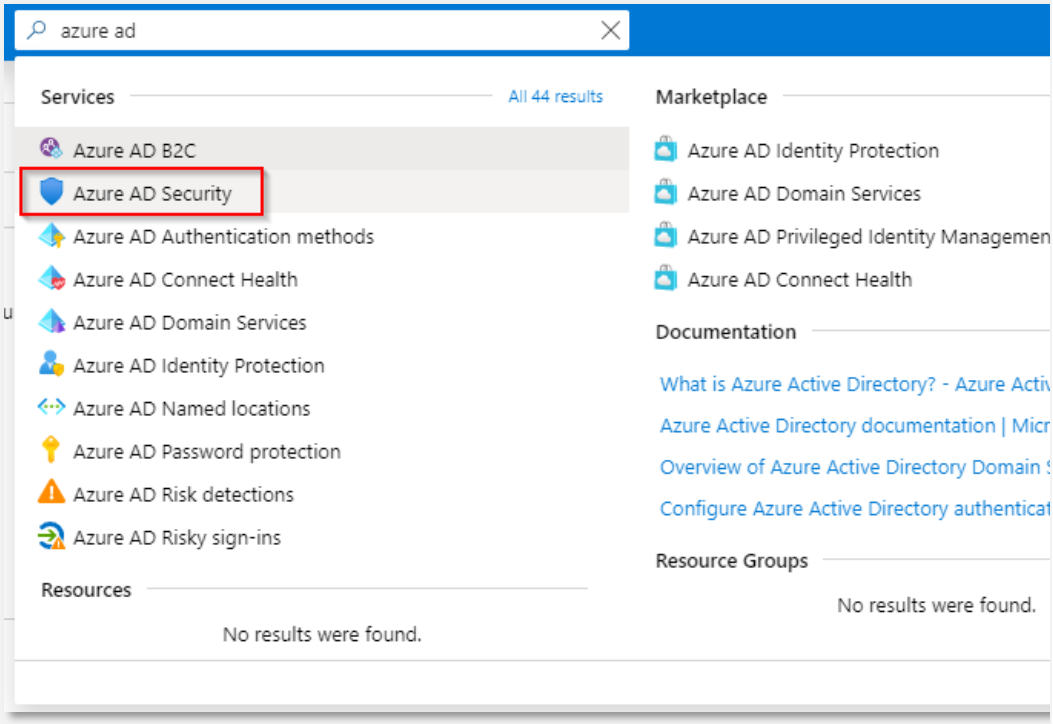
4) Review the description and *enable* the policy. **Save** the settings.



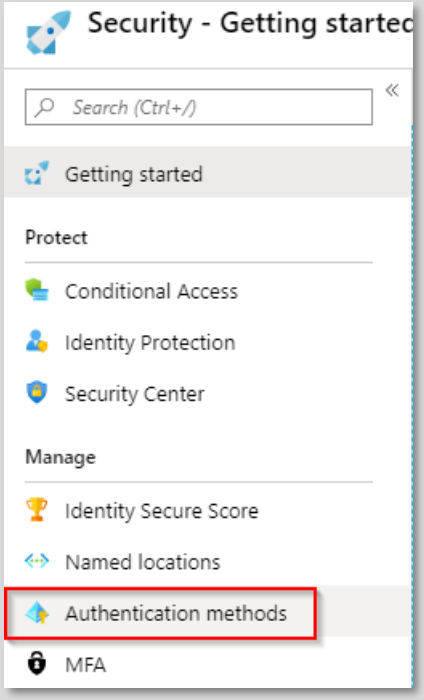
5) Repeat steps **14** and **15** for the other baseline policies.



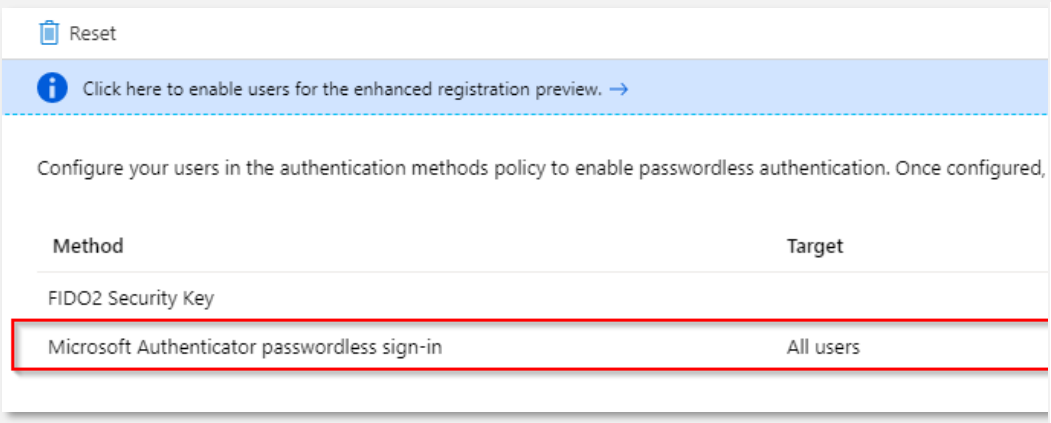
6) From the Search Bar, navigate to the Azure AD Security panel.



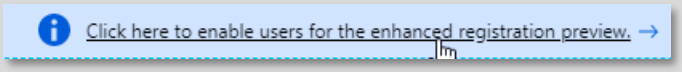
7) Browse to *Authenitcation Methods*.



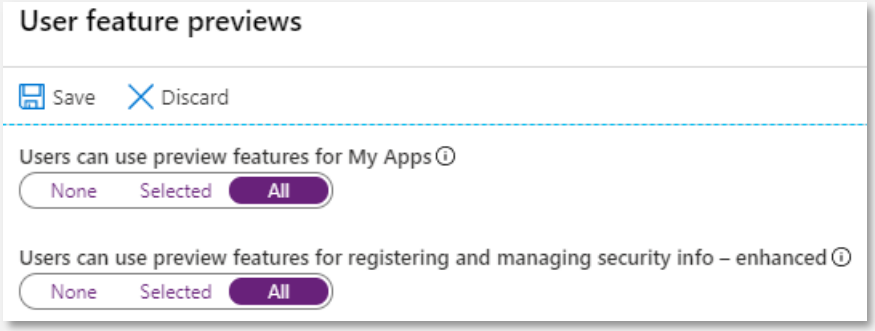
8) Enable *Microsoft Authenticator passwordless sign-in* for *All Users*.



9) Click the preview registration notification bar.



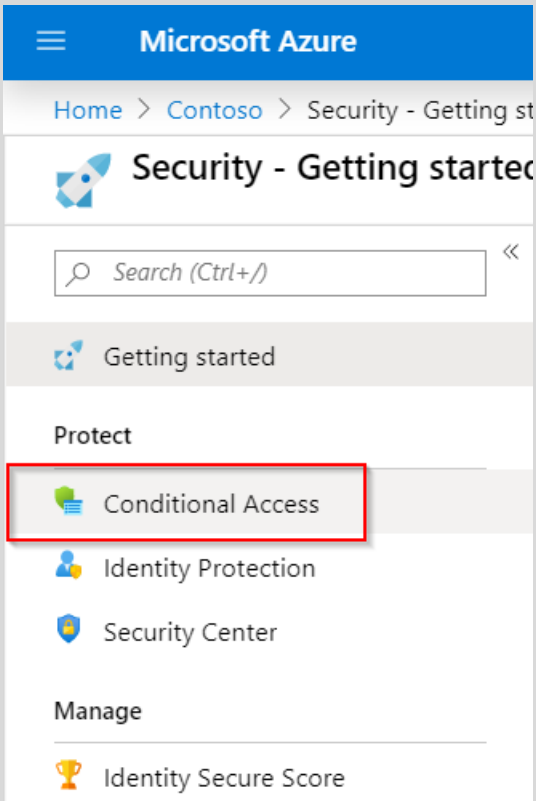
10) *Enable* all preview features and click **Save**.



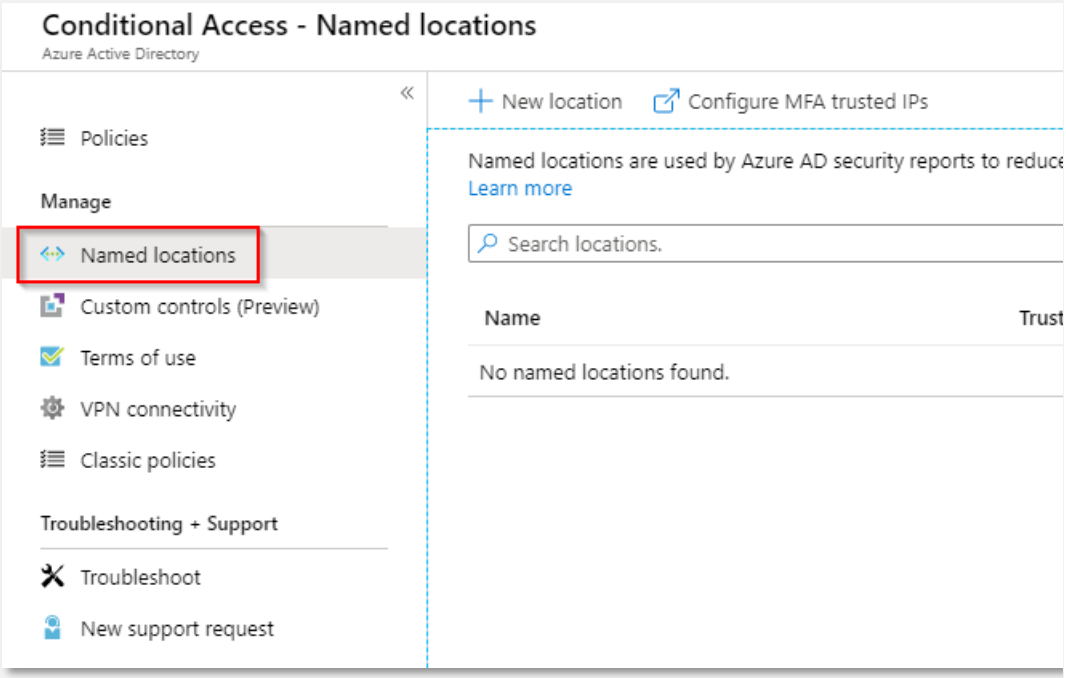
11) Your done for this exercise!

Exercise 1d: Implement custom Conditional Access policy

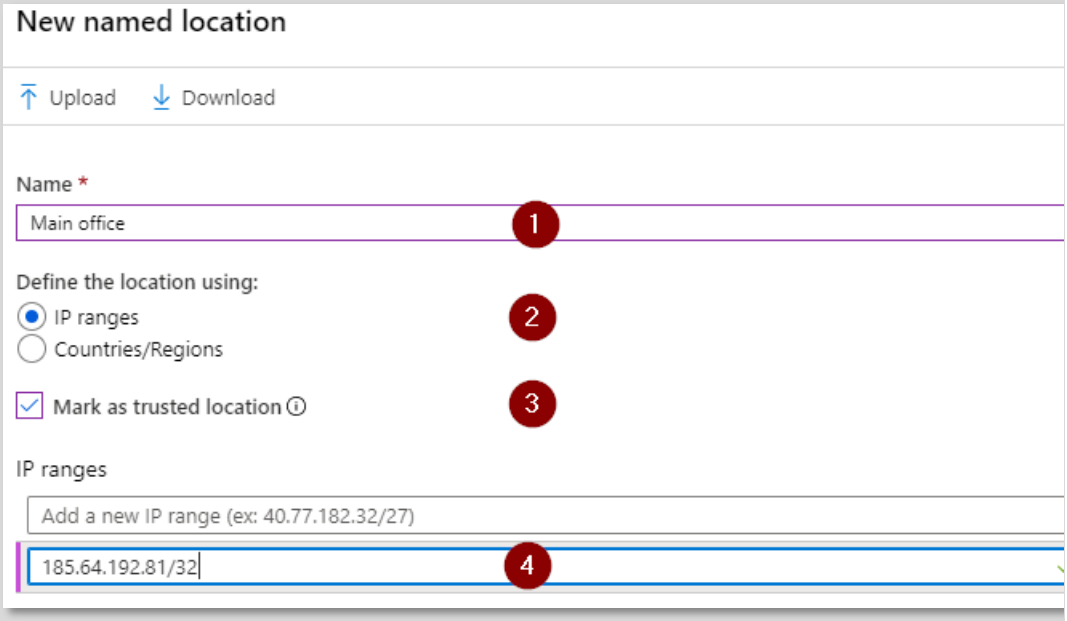
1) From the Azure Portal, navigate to Azure Active Directory > Security > Conditional Access



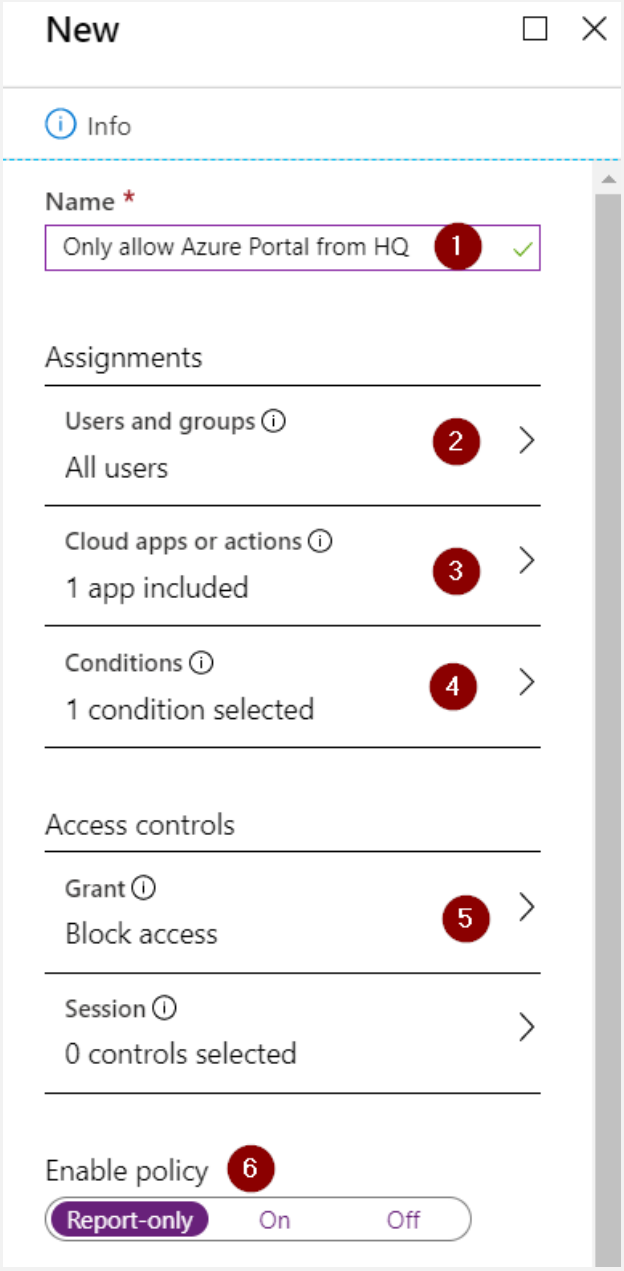
2) From the navigation panel, go to Named Locations.



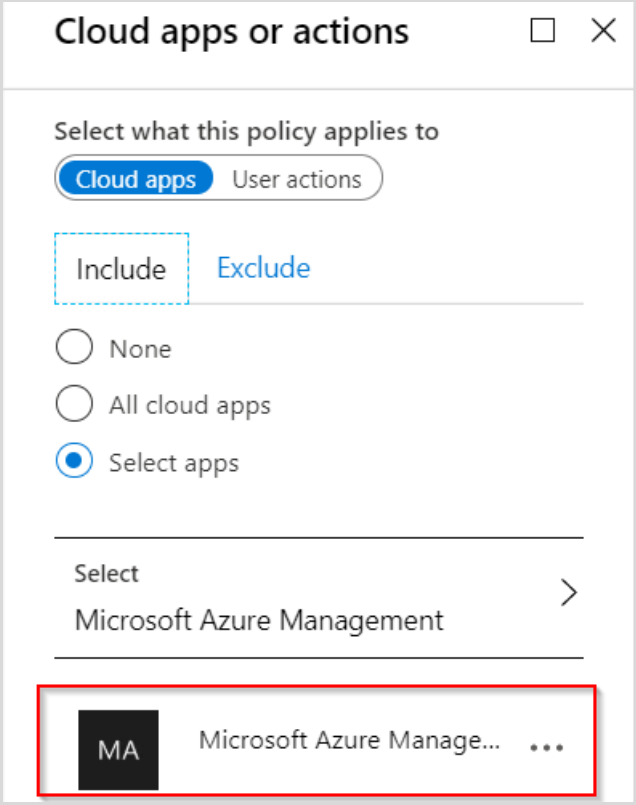
3) Choose to add a New location and fill in the form. Use the IP-address you retrieved earlier.



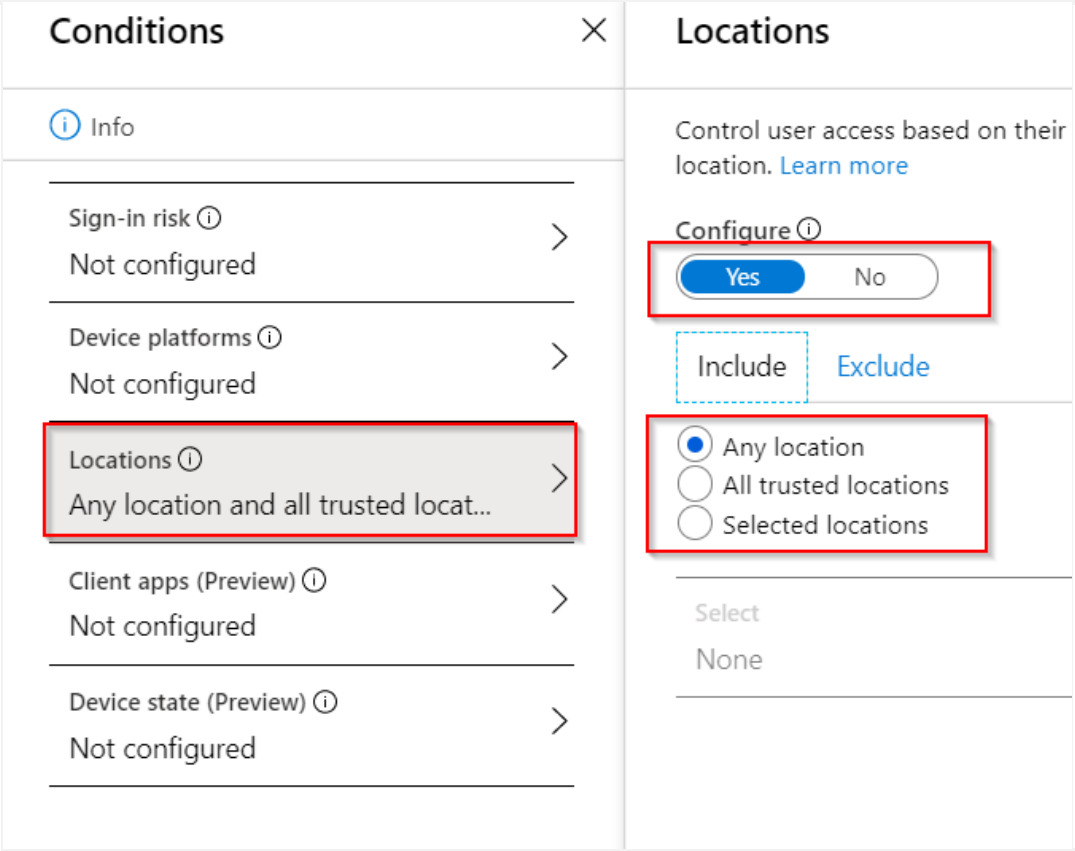
4) Create a new policy and define the settings following the instructions.

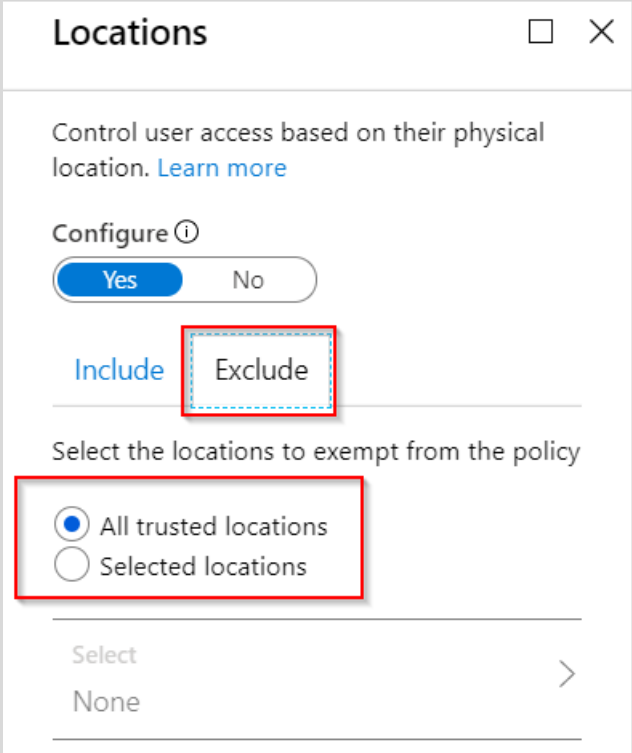
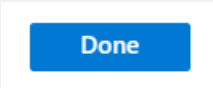
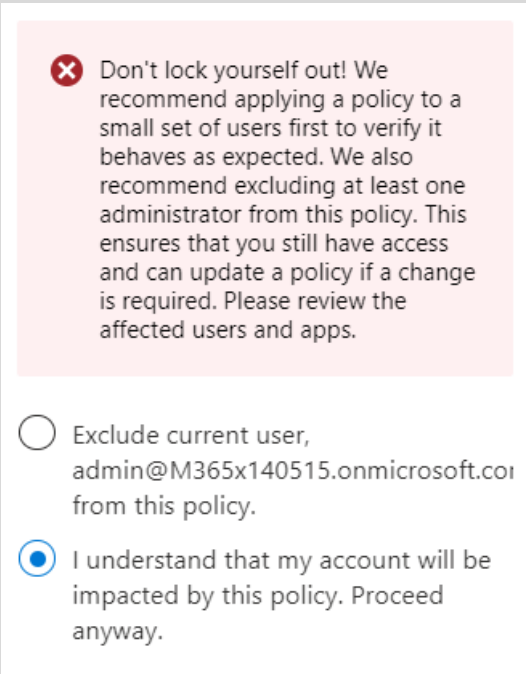
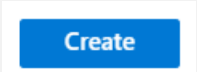


5) Select the Azure Portal as a condition for this policy.



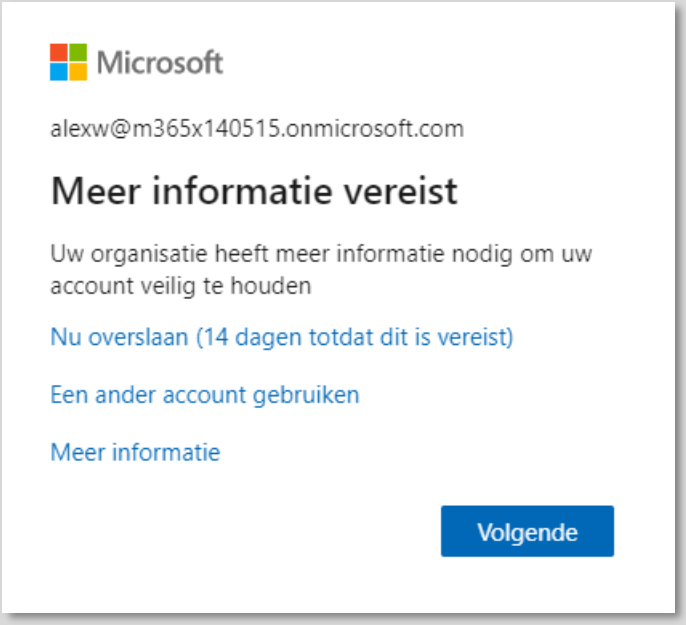
6) We want every location to be included. ..



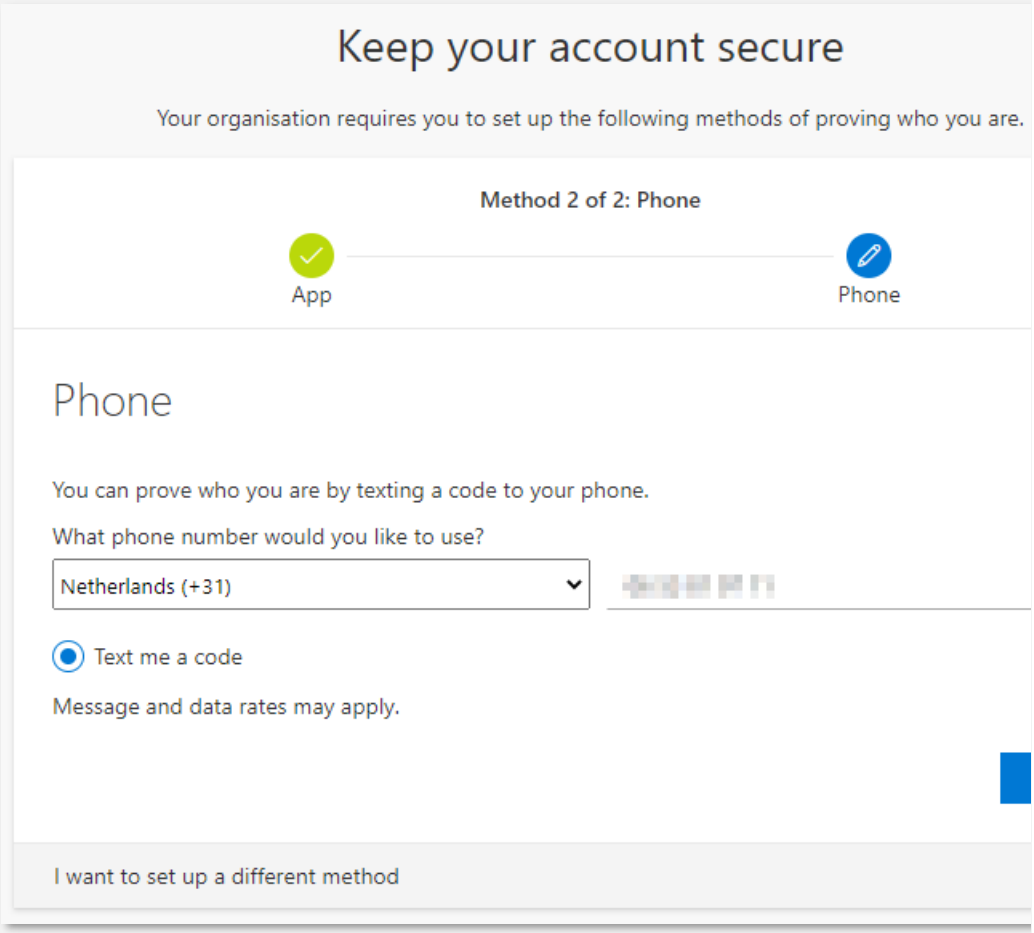
7)	Except the trusted location we	
8)	Apply the changes by clicking Done twice.	
9)	The Azure Portal warns on locking yourself out. As we choose to report-only, we can safely ignore the suggested exclusion.	
10)	Create the policy.	

Exercise 1e: Validate Passwordless Sign-in

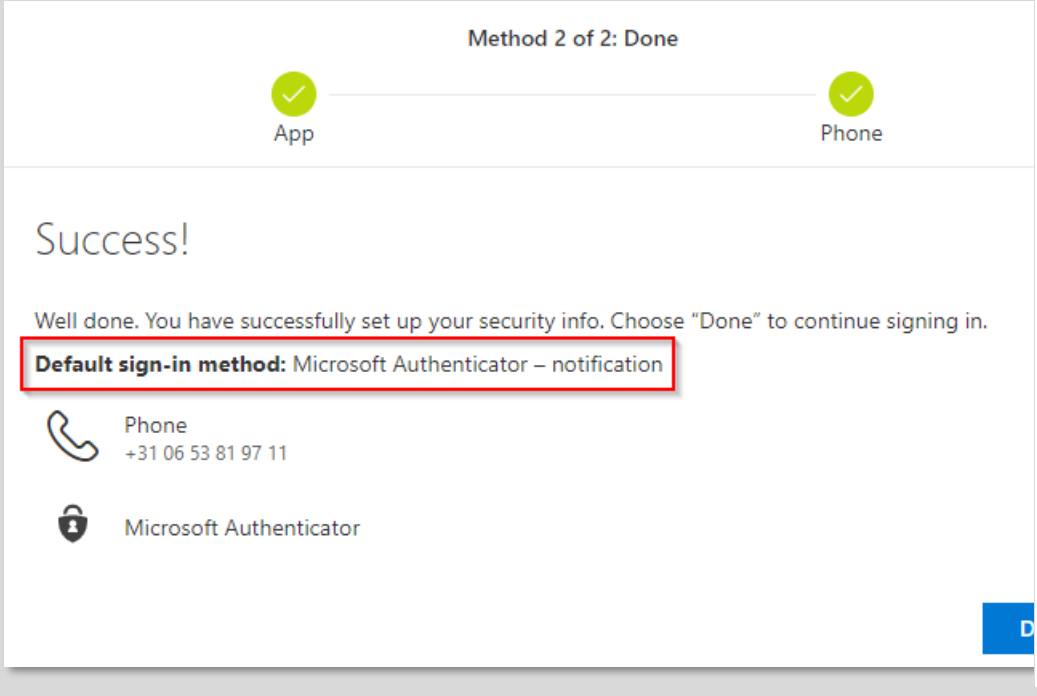
11) Switch back to the in-private browser window sign in again by using the *user account.*

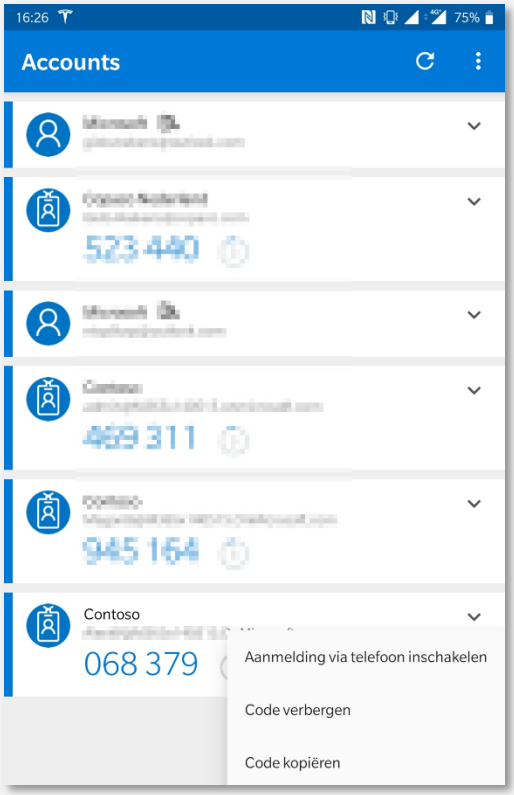


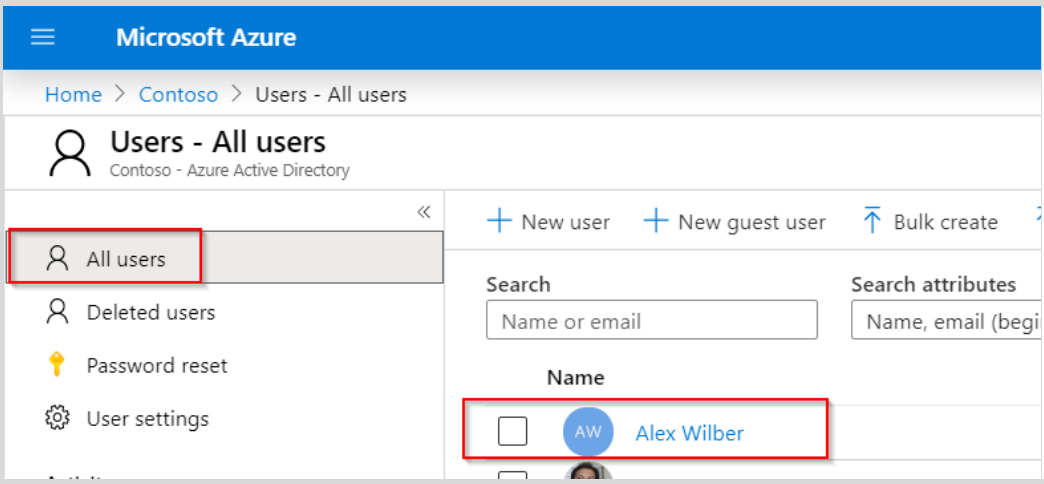
12) Please note you're required to register for MFA. Follow the instructions to download and activate the Microsoft Authenticat or app. You'll be asked to add a phone number as a secondary method. Please activate your mobile phone number.



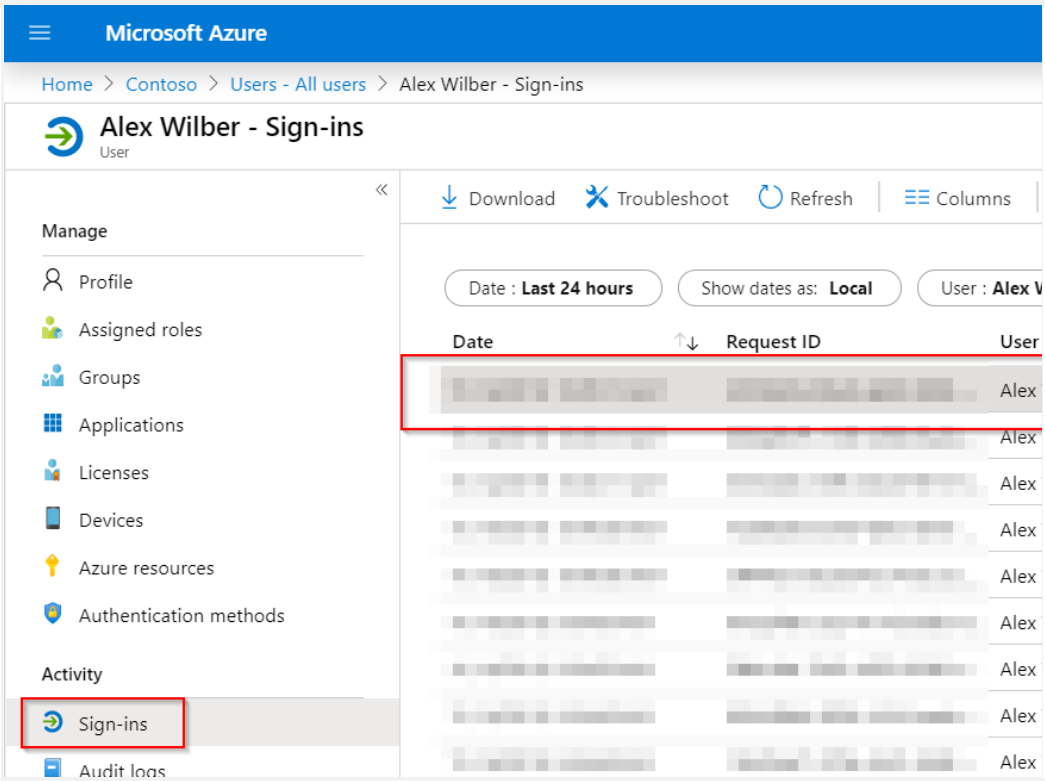
13) When done, both the Authenticat or should be the default sign-in method.



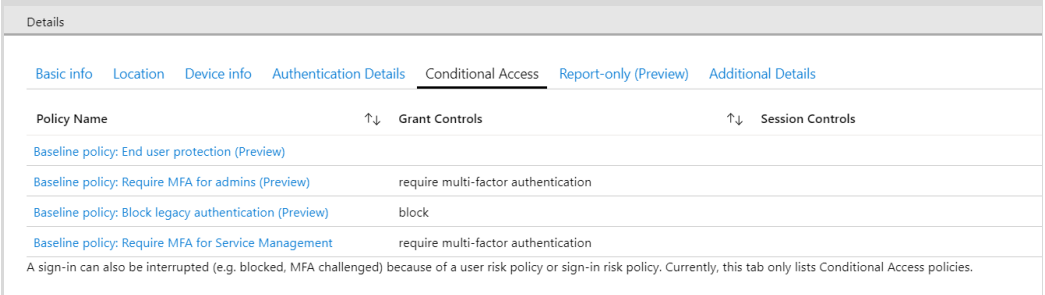
14) From the Authenticator app on your phone, start the Phone Sign-In registration.	
15) Sign in using the user credentials and register the device for Phone Sign-In.	
16) From the webbrowser, sign out from the account. Sign right back in using the same user account.	
17) Please not you won't be prompted for a	

password. Instead, the Authenticator app now validates using a numeric value. Choose the corresponding number to validate.	
18) Switch back to the browser instance with the Azure Portal	
19) Navigate to Azure Active Directory > Users and select Alex Wilber	

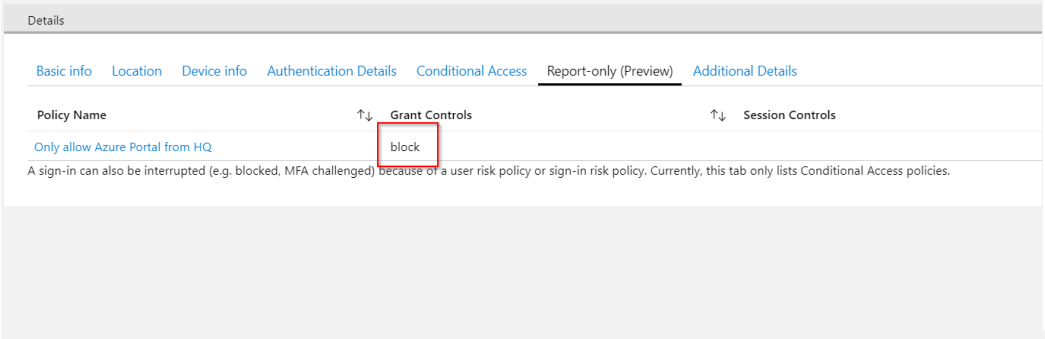
20) Select the sign-in that corresponds with your last sign-in as Alex. Wait and refresh if it's not being displayed yet.



21) Browse to the Conditional Access tab. Validate if the policies are applied as expected. This helps you understand behavior and troubleshooting concerning sign-ins.



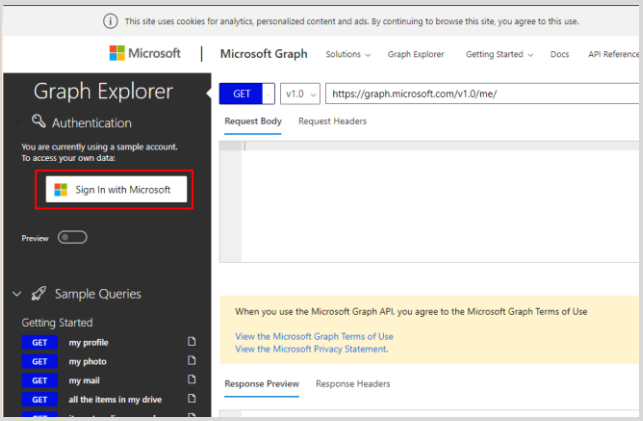
22) Browse to the Report-Only tab. Verify that the policy should block access, but hasn't because of the report-only setting. This is a great way



	to check your policies before taking them into production.	
23)	Your done for this exercise!	

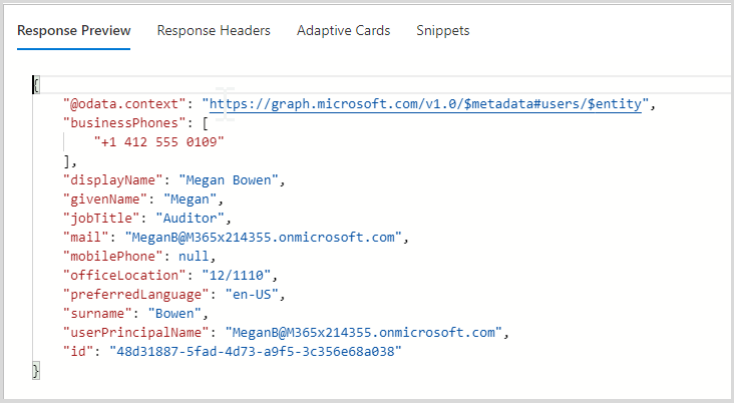
Activity 2 : Graph API Explorer

1) Go to https://developer.microsoft.com/en-us/graph/graph-explorer
2) Sign in with your demo administrator account
3) If prompted, provide consent for the access



4) There is a preview version available, you can enable it to view the lasted features.

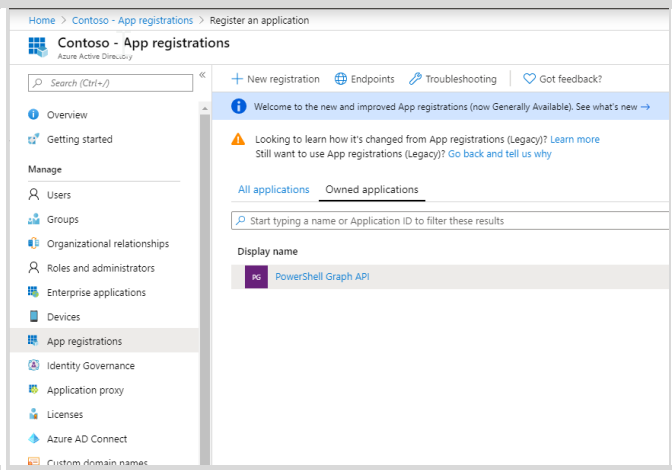
5) At the top of the screen, choose Run Query.
6) Review the output



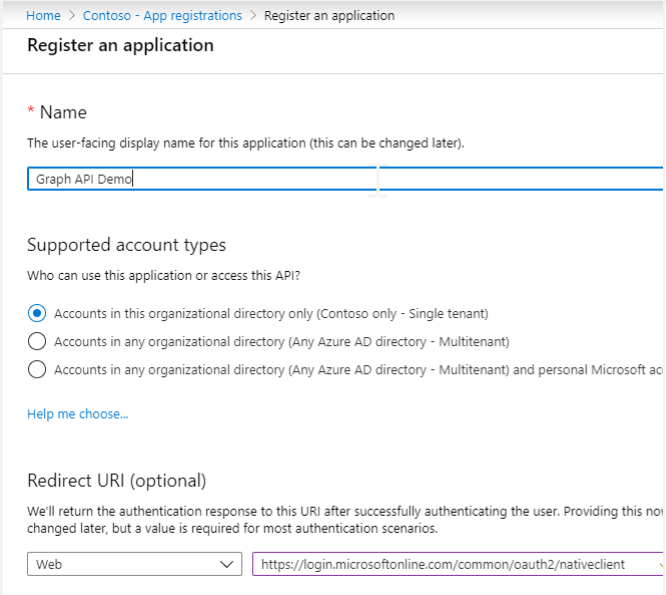
7) Also make note of the Permissions tab. This is an easy way to discover what permissions are required for the query.

Activity 3 : Connect with the Graph API

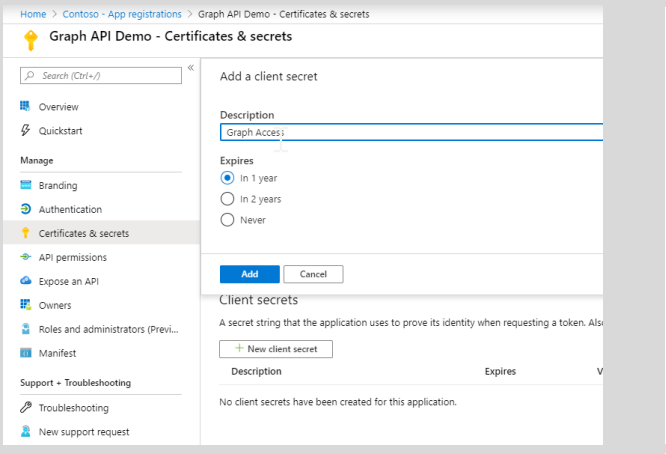
1) Log in to portal.azure.com
2) Go to Azure Active Directory
3) Choose App Registrations
4) Choose New Registration



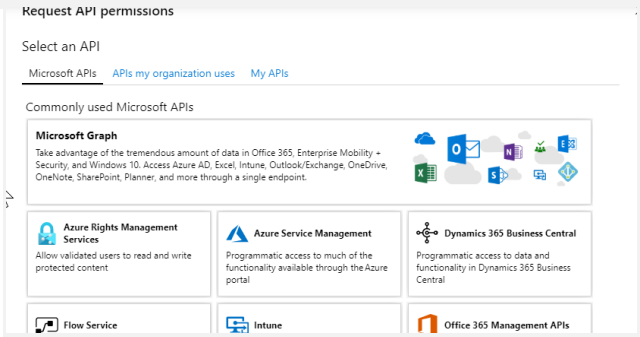
5) Give the Application a Name
6) Choose Single Tenant
7) For the Redirect URI enter https://login.microsoftonline.com/common/oauth2/nativeclient
8) Finally click Register



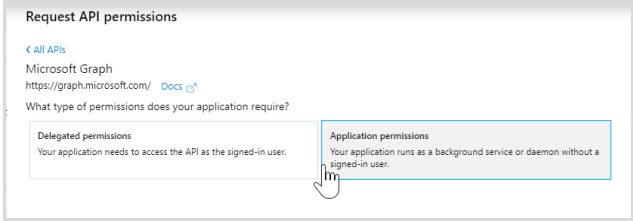
9) Go to Certificates & secrets and choose New Client Secret
10) Choose 1 Year and give it an Description
11) Choose Add
12) **Copy the Client Secret value** you wont be able to get it after you leave the page. And store it some where save



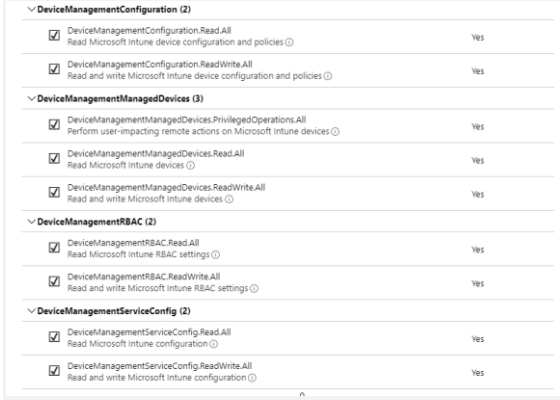
13) Go to API permissions
14) Choose Add a permission
15) Select Microsoft Graph



16) Choose Applicatoin permissions (for the demo this is easiest

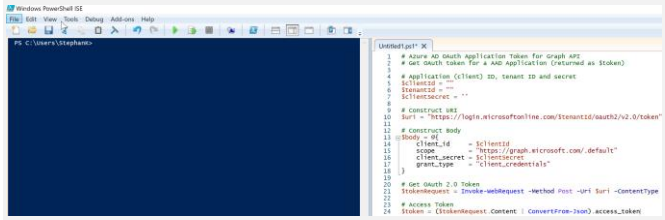


17) You can now select which permissions are needed.
18) Select all user and devices permissions and choose add permissions
19) Choose Add permissions



20) Scroll to the bottom of the screen and select Grant admin consent for "name of your tenant" and choose yes

21) Open Powershell ISE, and copy the script Get-AuthToken.ps1 in the Script Pane



22) Go to https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps and select the app that you have registered. In the overview you can find the Application ID, Directory ID

23) In PowerShell copy the client id Tenant ID and the client secret. The client secret you got in activity 12.

Untitled1.ps1*

```
# Azure AD OAuth Application Token for G
# Get OAuth token for a AAD Application

# Application (client) ID, tenant ID and
$clientId = "8e3ba
$tenantId = "380aa
$clientSecret = '

# Construct URI
$uri = "https://login.microsoftonline.co

# Construct Body
$body = @{
    client_id     = $clientId
    scope         = "https://graph.micro
    client_secret = $clientSecret
    grant_type    = "client_credentials"
}

# Get OAuth 2.0 Token
$tokenRequest = Invoke-WebRequest -Metho

# Access Token
$token = ($tokenRequest.Content | Conver
```

24) Run the Script
25) If you run $token then you should receive the token for authorization

```
PS C:\Users\StephanK> $token
```

eyJ0eXAiOiJKV1QiLCJub25jZSI6IllhRzRKUnA5VTZLSldFanRDRHZNXOhRVGR1MzU0RFZVak45aFkzc2FivnciLC
JhbGciOiJSUzI1NiIsIng1dCI6IkJCOENlRlZxeWFHckdOdWvoSklpTDRkZmp6dyIsImtpZCI6IkJCOENlRlZxeWFH
ZGdWvoSklpTDRkZmp6dyI9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1pY3Jvc29mdC5jb2OiLCJpc3MiOiJodHRwc
Ovl3N0cy53aW5kb3dzLm5ldC8zODBhYWVmOCO5MzE5LTQzZTktYjdiYy0xZmRMNzQyYmM2YWEvIiwiaWF0IjoxNTc
zMONzE3LCJuYmYiOjE1NzM3MzQ3MTcsImv4cCI6MTU3MzczODYxNywiYWlvIjoiNDJWZ1loavk5bOI3VjZmZHNRb
nWGU0bDZ5LOJRQTOiLCJhcHBfZGlzcGxheW5hbWUiOiJQb3dlclNoZWxsIEdyYXBoIEFQSSIsImFwcGlkIjoiOGU
mExYTgtODFjMCOONzJlLTllNDutODY0YzkOOTkOODFlIiwiYXBwaWRhY3IioiIxIiwiaWRwIjoiaHROcHM6Ly9zd
ud2luZG93cy5uZXQvMzgwYWflZjgtOTMxOSOOM2U5LWI3YmMtMWZkZjcOMmJjNmFhLyIsIm9pZCI6ImZhNWY5OTg
WVjNjMtNGZjYi1hZjg3LWM4MTMyZGQzOTBjZiIsInJvbGVzIjpbIlBsYWNlLjlYWQuQWxsIiwiQ2hhdC5vcGRhd
Qb2xpY3lwaW9sYXRpb24uQWxsIiwiSWRlbnRpdHlvc2VyRmxvdy5SZWFkV3JpdGUuQWxsIiwiQ2FsbHMuSm9pbkd
3VwQ2FsbC5BbGwiLCJFZHVSb3NOZXIuUmVhZC5BbGwiLCJEZXZpY2VNYW5hZ2VtZW5OTWFuYWdlZERldmljZXMuU
nZC5BbGwiLCJPbmxpbmVNZWV0aW5ncy5SZWFkLkFsbCIsIkFjY2Vzc1Jldmlldy5SZWFkV3JpdGUuTWVtYmVyc2h
CIsIkJpdGxvY2tlcktlesS5SZWFkQmFzaWMuQWxsIiwiQXBwcm92YWxSZXF1ZXNOLljlYWRxcml0Zs5DdXNOb21lc
WY2tib3giLCJUZWFtcOFjdGl2aXR5LljlYWQuQWxsIiwiQml0bG9ja2vyS2V5LljlYWQuQWxsIiwiVGVhbXNBcHA
mvhZC5BbGwiLCJNYWlsLljlYWRxcml0ZSIsIklkZW50aXR5UmlzaOV2ZW5OLljlYWRxcml0Zs5BbGwiLCJPcmdDb
OYWNOLljlYWQuQWxsIiwiT25saW5lTWVldGluZ3MuUmVhZFdyaXRlkFsbCIsIkRldmljZS5SZWFkV3JpdGUuQWx
iwiVXNlci5SZWFkV3JpdGUuQWxsIiwiRG9tYWluLljlYWRxcml0Zs5BbGwiLCJBcHBsaWNhdGlvbi5SZWFkV3Jpd
UT3duZWRCeSIsIlNlY3VyaXR5QWN0aW9ucy5SZWFkV3JpdGUuQWxsIiwiU2VjdXJpdHlFdmVudHMuUmVhZC5BbGw
CJEZWxlZ2FOZWRQZXJtaXNzaW9uR3JhbnQuUmVhZFdyaXRlkFsbCIsIkNhbGVuZGFycy5SZWFkIiwiTWFpbC5Sz
kQmFzaWMuQWxsIiwiUHJpdmlsZWdlZEFjY2Vzcy5SZWFkLkF6dXJlQUQiLCJSb2xlTWFuYWdlbWVudC5SZWFkLkR
mvjdG9yeSIsIkVkdUFzc2lnbmllbnRzLljlYWRxcml0ZS5BbGwiLCJQZW9wbGUuUmVhZC5BbGwiLCJBcHBsaWNhd
Vbi5SZWFkV3JpdGUuQWxsIiwiQ2FsbHMuSW5pdGlhdGVHcm9lcENhbGwuQWxsIiwiR3JvdXAuUmVhZC5BbGwiLCJ
Gipbmlzdh3hdGl2ZyVuaXQuUmVhZC5BbGwiLCJBY2NlC3NSZXZpZXcuUmVhZFdyaXRlkFsbCIsIk9uUHJlbWlzZ
QdWJsaXNoaW5nUHJvZmlsZXMuUmVhZFdyaXRlkFsbCIsIkRpcmVjdG9yeS5SZWFkV3JpdGUuQWxsIiwiVGVhbXN
BRpdmlOeS5TZW5kIiwiSWRlbnRpdHlsaXNreVvzZXIuUmVhZFdyaXRlkFsbCIsIkVkdUFzc2lnbmllbnRzLljlY
Xcml0ZUJhc2ljLkFsbCIsIk1haWxib3hTZXR0aW5ncy5SZWFkIiwiRWRlQWRtaW5pc3RyYXRpb24uUmVhZC5BbGw
CJDYWxscy5Kb2luR3JvdXBDYWxsQXNHdWvzdC5BbGwiLCJUaHJlYXRJbmRpY2F0b3JzLljlYWRxcml0Zs5Pd2Slz
5IiwiU2l0ZXMuUmVhZC5BbGwiLCJEZXZpY2VNYW5hZ2VtZW5OU2VydmljZUNvbmZpZy5SZWFkLkFsbCIsIlNpdGV

Activity 4 : Custom Graph API commands

1) From github copy the Lab Graph cmds in a new page in Powershell ISE
2) Make sure to adjust the parameters that match your tenant + **some of the scripts assume that you have configuration policies and compliance policies in place. If you don't have these in the tenant create some!**

3) Get all users

```
#get all users
$uri = "https://graph.microsoft.com/beta/users"
$users = Invoke-RestMethod -Method GET -Uri $uri -Headers @{Authorization = "Beare
$users.value
$users.value | Select-Object DisplayName, ID, UserPrincipalName
```

4) Get specific user
5) Make sure that you adjust the upn for the specific

```
#get specific user
$uri = "https://graph.microsoft.com/beta/users/CameronW@M
$megan = invoke-restmethod -Method GET -Uri $uri -Headers
$megan.displayName
$megan | Select-Object DisplayName, MobilePhone, City
```

6) Update user info

```
#update user info
$PatchJSON = @{
    "mobilephone" = "+31640409642"
    "city"        = "Eindhoven"
} | ConvertTo-Json

Invoke-RestMethod -Uri $uri -Method PATCH -Headers @{Aut
```

7) Check updated info
8) Make sure that you adjust the upn for the specific

```
#Check if user info is updated
$uri = "https://graph.microsoft.com/beta/users/CameronW@M3
$megan = invoke-restmethod -Method GET -Uri $uri -Headers
$megan.displayName
$megan | Select-Object DisplayName, MobilePhone, City
```

9) Create new user
10) Make sure that you adjust the upn your tenant

```
#create new user
$uri = "https://graph.microsoft.com/beta/users"
$NewUserJSON = @{
    "accountEnabled"    = $true
    "displayName"       = "EL Demo User"
    "mailNickname"      = "eldemouser"
    "userPrincipalName" = "eldemouser@M365x428595.
    "mobilephone" = "+31640409642"
    "city"        = "Eindhoven"
    "passwordProfile"   = @{
```

11) Check created user

```
#check created user
$uri = "https://graph.microsoft.com/beta/user
$DemoUser = invoke-restmethod -Method GET -Ur
$DemoUser.displayName
$DemoUser | Select-Object DisplayName, Mobile
```

12) Delete created user

```
#get and delete created user
$uri = "https://graph.microsoft.com/beta/users"
$uri = $uri + '/' + $response.id
invoke-restmethod -Method GET -Uri $uri -Headers @

Invoke-RestMethod -Method DELETE -Uri $uri -Header
```

13) Get all groups	`#get all groups`
	`$uri = "https://graph.microsoft.com/beta/gro`
	`Invoke-RestMethod -Method GET -Uri $uri -Hea`
	`$groups = Invoke-RestMethod -Method GET -Uri`
	`$groups.value`
	`$groups.value | ft DisplayName`

14) Get Member of first group	`#get member of first group`
	`$groups.value[0]`
	`$groupid = $groups.value[0].id`
	`$uri = $uri + '/' + $groupid`
	`Invoke-RestMethod -Method GET -Uri $uri -Header`
	`$uri = $uri + '/' + 'members'`
	`$members = Invoke-RestMethod -Method GET -Uri $`

15) Get groups a user is member of	`#get groups user is member of`
	`$uri = "https://graph.microsoft.com/beta/users`
	`$membership = Invoke-RestMethod -Method GET -U`
	`$membership.value | select DisplayName`

16) Export Device configuration Profiles	`1`
	`2 ###Export Device configuration profiles`
	`3 $uri = "https://graph.microsoft.com/beta/deviceMan`
	`4 $configs = Invoke-RestMethod -Method GET -Uri $uri`
	`5`
	`6 foreach ($config in $configs.value) {`
	`7 $configname = $config.displayName`
	`8 $configfile = "C:\temp\$configname" + '.json'`
	`9 $config | ConvertTo-Json | out-file $configfil`
	`10 }`
	`11`

17) Export Device compliance policies	`###Export Device Compliance Policies`
	`$uri = "https://graph.microsoft.com/beta/deviceMar`
	`$compliances = Invoke-RestMethod -Method GET -Uri`
	
	`foreach ($compliance in $compliances.value) {`
	`$configname = $compliance.displayName`
	`$configfile = "C:\temp\$configname" + '.json'`
	`$config | ConvertTo-Json | out-file $configfil`
	`}`

18) Hello For Business settings	`###Export Hello For Business Settings`
	`$uri = "https://graph.microsoft.com/beta/deviceManage`
	`$WHBusiness = Invoke-RestMethod -Method GET -Uri $uri`
	`$WHBusiness`
	
	`$configname = $WHBusiness.value | select Displayname`
	`$configfile = "C:\temp\$($configname[0])" + '.json'`
	`$config | ConvertTo-Json | out-file $configfile`

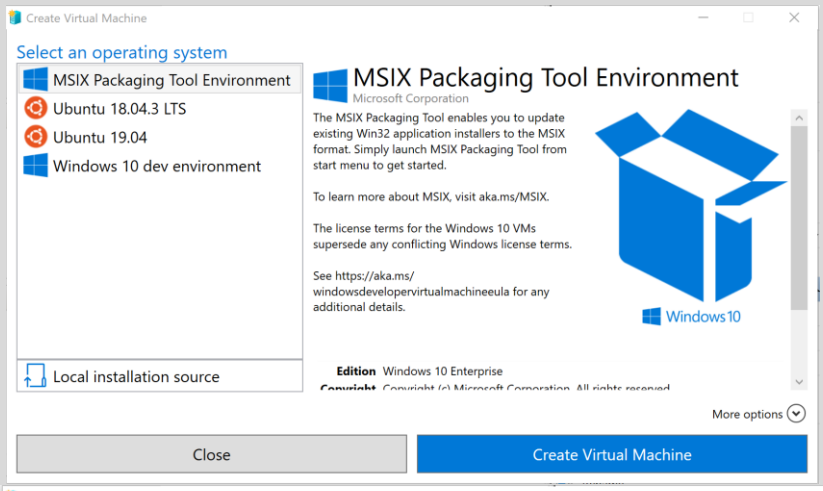
19) Import Device Configuration Policies
20) Download the new-policy-demo.json from Github to your local pc.
21) Edit the $newPolicy variable so that the json is imported to the variable
22) Run the script.
23)
	`#Import Device Configuration Policy`
	`$uri = "https://graph.microsoft.com/beta/deviceManagement/deviceconfigurations"`
	`$NewPolicy = get-content "C:\Users\StephanK\Desktop\MSIX demo\Graph Api\new-policy-demo.js`
	
	`$outputNieuwPolicy = Invoke-RestMethod -Method POST -Uri $uri -Headers @{Authorization = "`
	
	`$uri = "https://graph.microsoft.com/beta/deviceManagement/deviceconfigurations/$($outputNi`
	`Invoke-RestMethod -Method GET -Uri $uri -Headers @{Authorization = "Bearer $token" } -Erro`

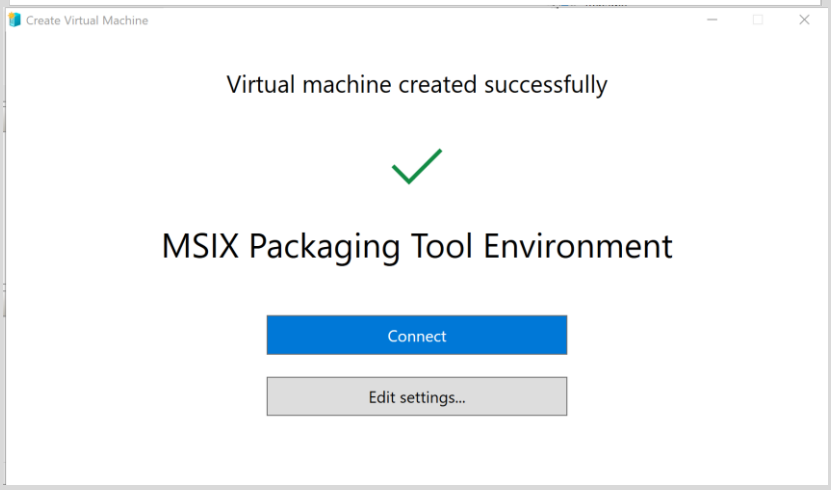
24) Check the Intune Portal to confirm that a new Configuration Policy is created.	

Activity 5 : MSIX

Exercise 1a: Create Packaging VM

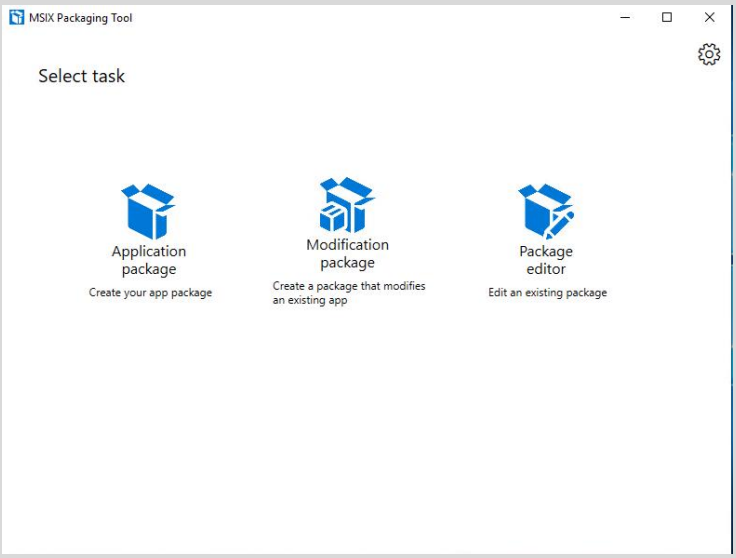
1) Open Hyper-V manager
2) Right click your host name and choose Quick Create
3) Pick MSIX Packaging Tool Environment and choose **Create Virtual Machine**. This will start to download the VM for you
4) When the VM is ready you can connect to the VM



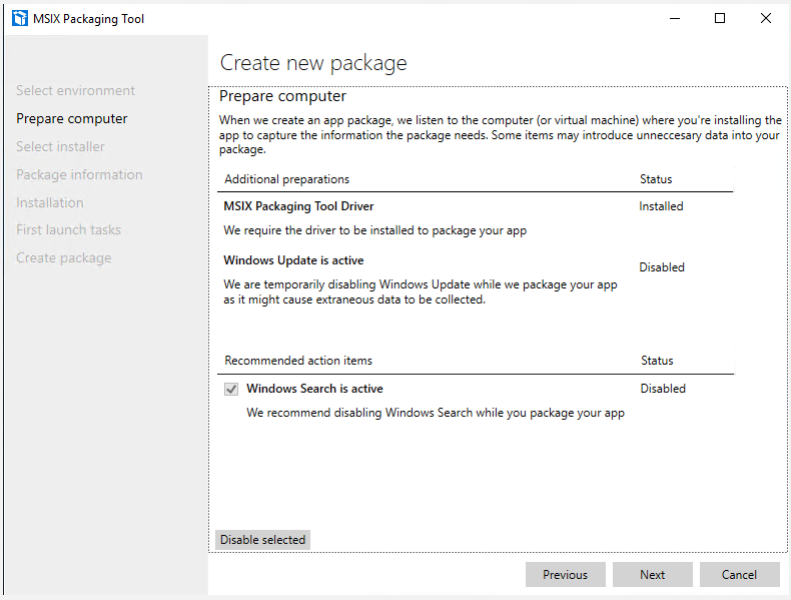


5) Start the VM and go through the installation process. Choose to domain join instead of a Microsoft account
6) When the VM is ready to use, it best that you create a snapshot (check point) so you can revert to this clean state in the future.
7) Download the Citrix receiver app and the pfx certificate from Github
8) Copy the installation files for the Citrix Receiver and the certificate to the VM. The files are available on Github

9) Start the MSIX Packaging tool
10) Choose Application Package



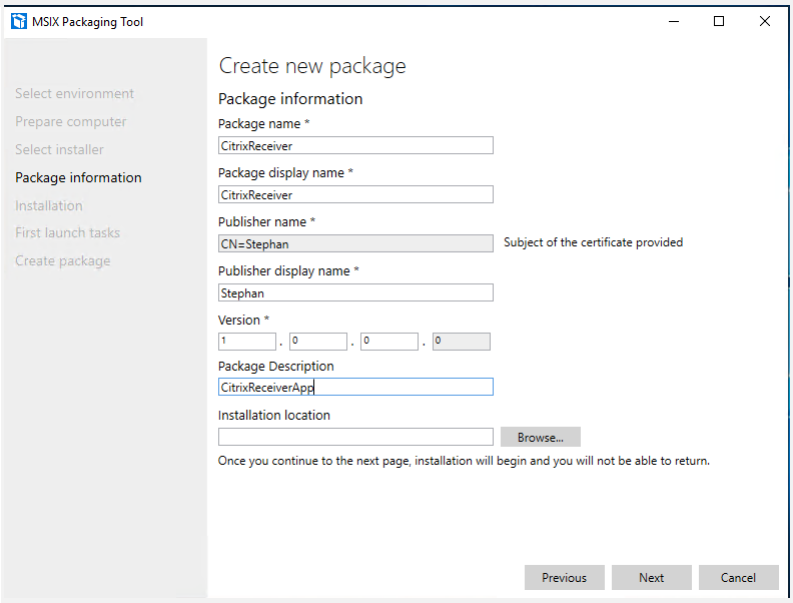
11) Create Package on this computer and click next.
12) Disable Windows Search and click next.



13) Browse to the Citrix receiver installation file
14) At Signing preference choose **Sign with a certificate .pfx.** Browse to the certificate.
The password is:
Welkom123$

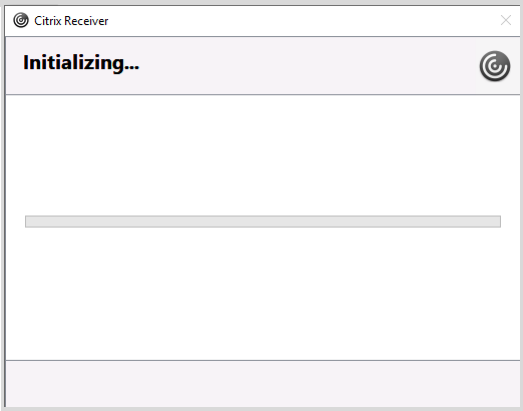
Choose Next

15) Enter the information for the application
16) Because the certificate uses the name **Stephan**, you must use this for the publisher display name. Choose next

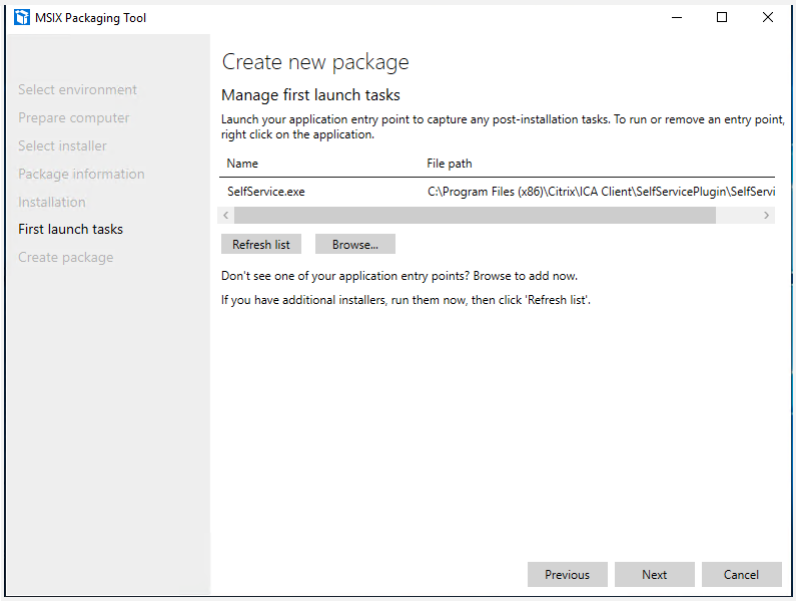


17) The Citrix receiver installation start now.

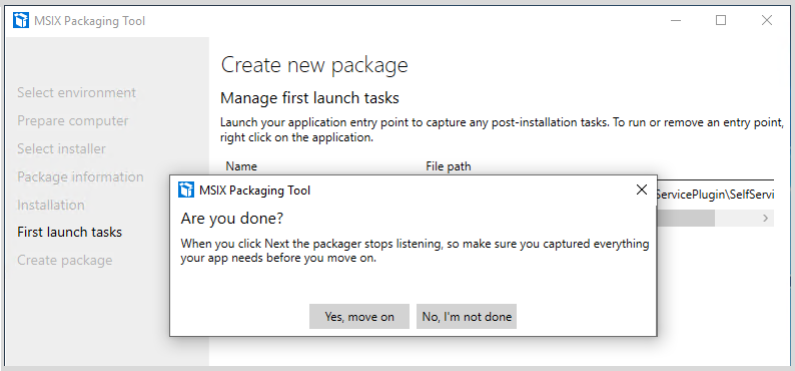
18) Following the installation steps
19) When the installation is done choose Finish
20) In the MSIX console choose Next



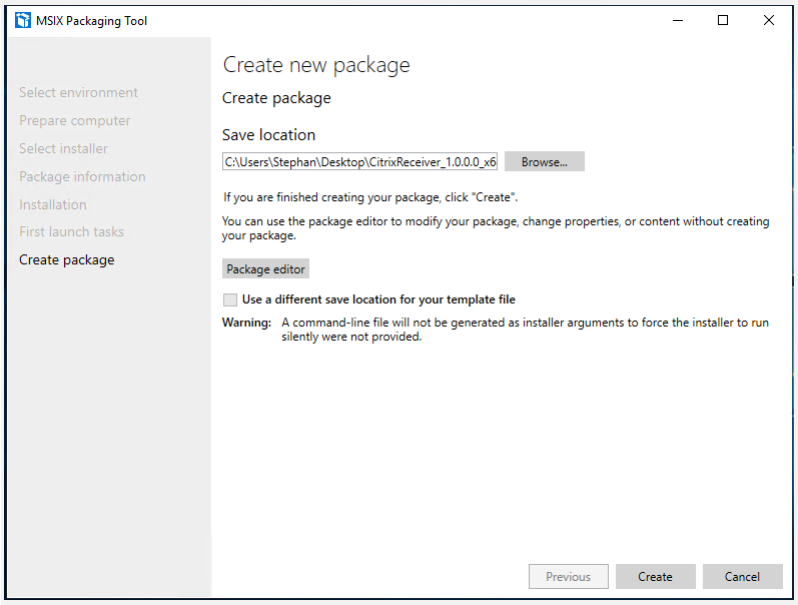
21) For now we can skip this, but for advanced installations you can continue to configure the first launch steps.
22) Choose next



23) Yes, move on
24) Your package is created



25) Choose a save location, and choose **Create**



Exercise 1b: Test your package

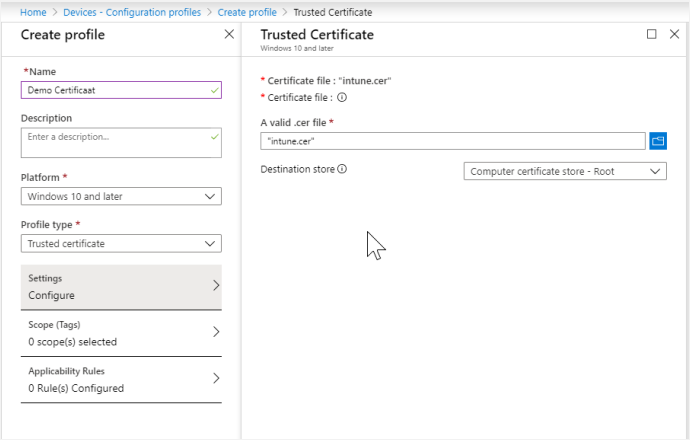
Now you created the package you proberly want to test it. There are a couple of ways you can do this. The first (and most easy way to do this) is to use the same MSIX machine you created. Since this being an M365 lab we can also deploy the MSIX with Intune, but this requires you to have a test VM which you can manage with Intune. This lab won't describe how to set up the test VM but you are free to set it up.

Deploy the MSIX application with Intune

Because the MSIX is signed with an self signed certificate that isn't trusted by default, we first have to deploy the certificate

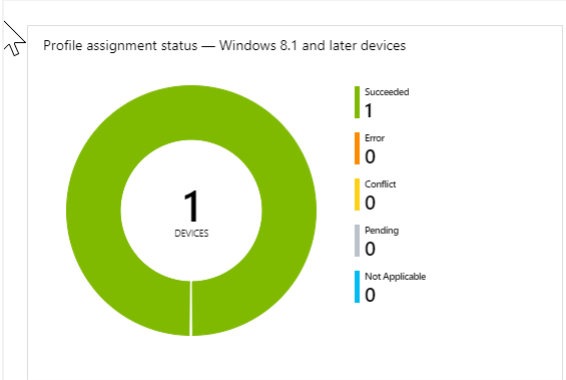
1) Go to https://devicemanagement.microsoft.com/ and to Devices. Here you can create a new Configuration Profile

2) Give the configuration a Name
3) For Platform choose **Windows 10 and later**
4) For Profile Type pick **Trusted certificate**
5) When you select Configure you can choose the cer file. This cer file can also be found at Github
6) For the Destination store you must choose **Computer certificate store – Root**
7) Choose OK and Create

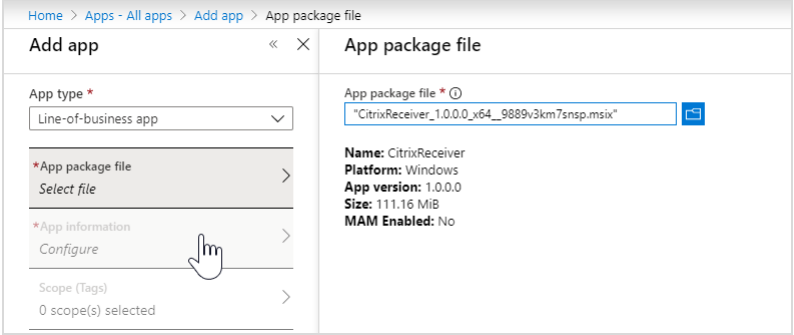


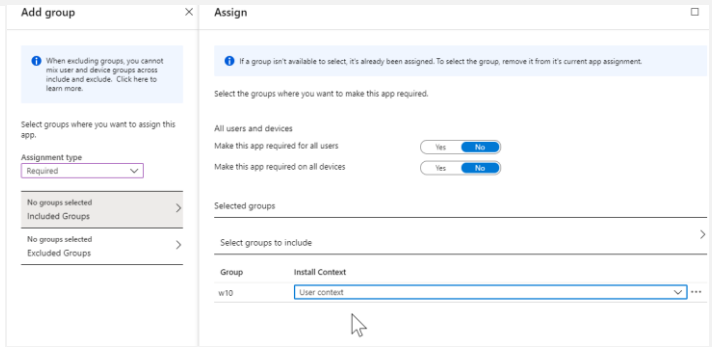
8) When the certificate is uploaded you can assign the Profile to a group which has the test VM in it.

9) **Wait and verify till the certificate is successfully is deployed to your test VM**



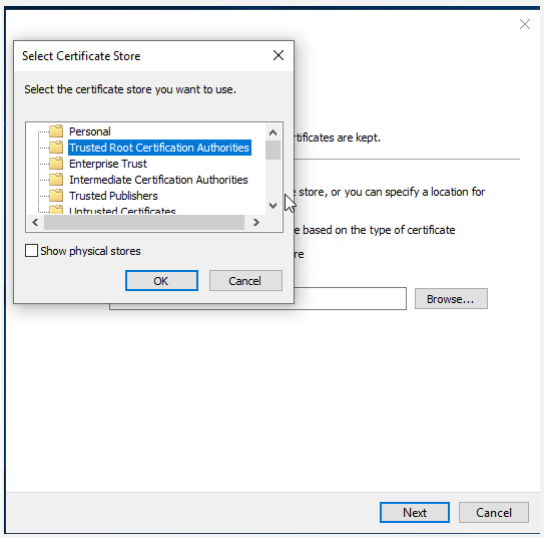
10) When the certificate is successfully deployed you can continue to deploy the Citrix receiver application
11) In the device management portal go to **Apps → All apps** and choose **Add**
12) App type is **Line-of-Business app**
13) At App package file navigate to your Citrix receiver MSIX package
14) Choose OK
15) Open the App information settings and reviews the settings. When ready choose OK
16) Choose Add
17) The application is now being uploaded. When ready you can continue to the next step



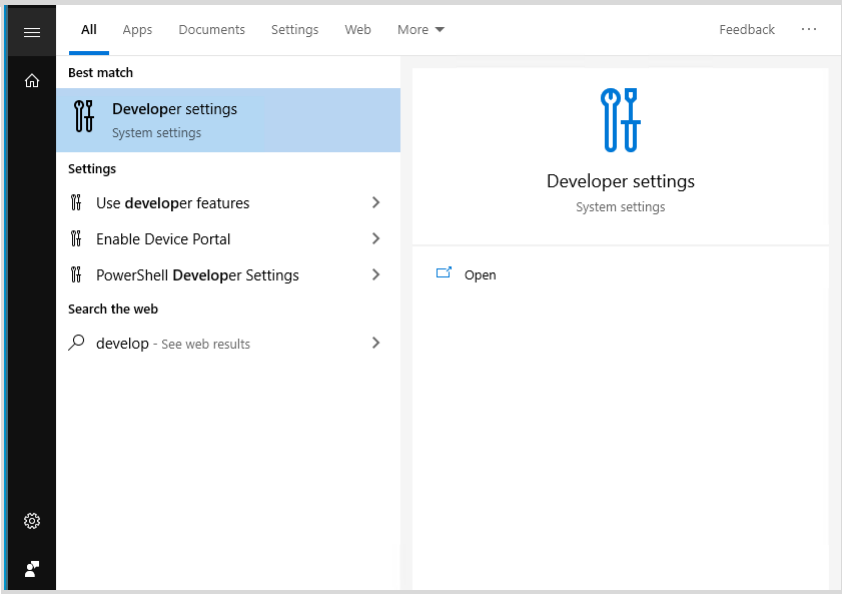
18) Go to assignment and choose **Add Group** 19) For assignment type choose **Required** 20) Pick the group which has your test vm in it and install in the **User Context** 21) Click Save	
22) Wait for the deployment to finish	
23)	

Test the Package locally

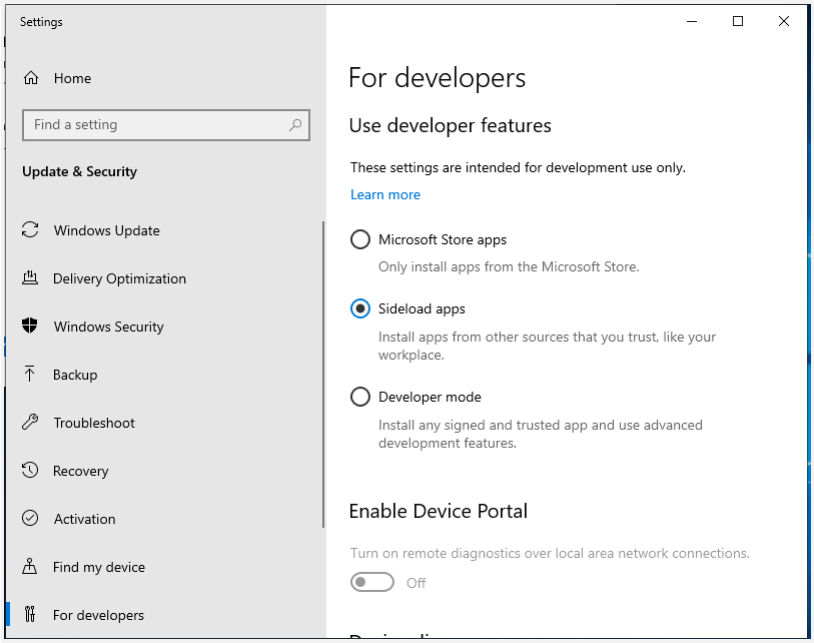
If you don't have a test VM which is enrolled in Intune you can also use the MSIX packaging machine to test your package.

1) Copy the MSIX package you created to your local PC.	
2) Revert to the clean state of the VM by applying the checkpoint you created	
3) Copy the MSIX package and the certificate to the machine	
4) Install the certificate by double clicking the certificate 5) Choose Local Machine 6) Next 7) Password is **Welkom123$**, and choose next 8) Choose Place all certificates in the following store 9) Select **Trusted Root Certification Authorities** Next 10) Finish	

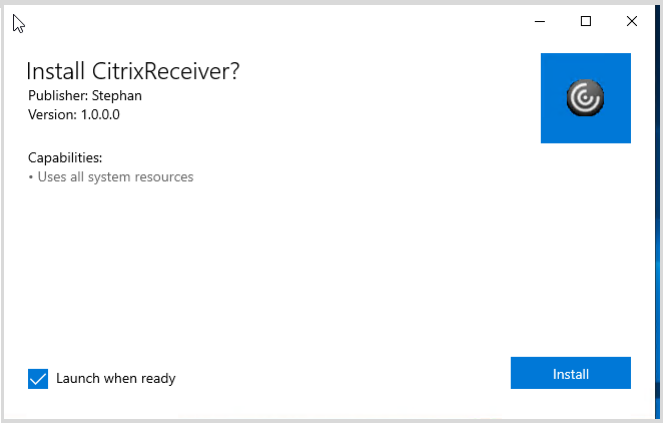
11) From the start menu search for **Developer settings**

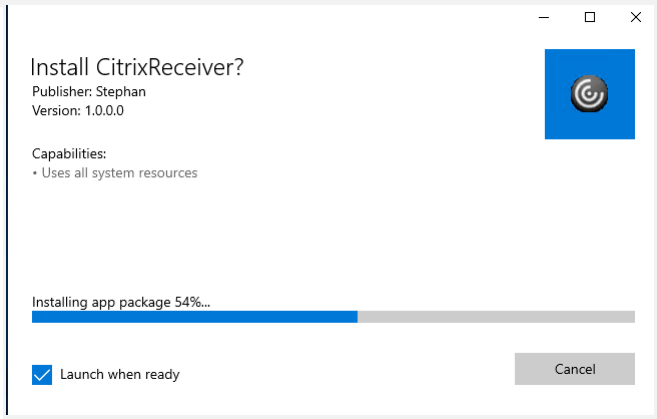
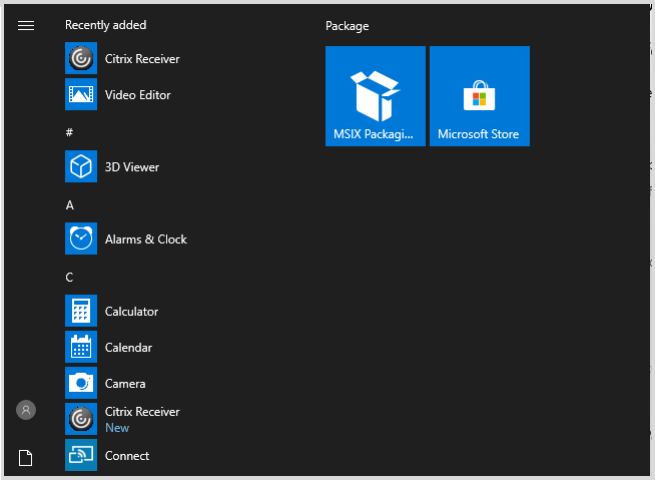


12) Choose Sideload apps



13) Then Double click your MSIX package
14) Choose Install



15) The applications is being installed	
16)	
17)	

Extra resources