# Creating a Custom Scope for Jamf Connect and Conditional Access policies

**Change log:**
24 AUG 2022 - Initial release. See notes of minimum system requirements for minimum version of Jamf Connect to use for these instructions.
25 AUG 2022 - Added additional screen shot to enable ROPG in public client app. Mirrors instructions in official documentation.

## Workflow overview:

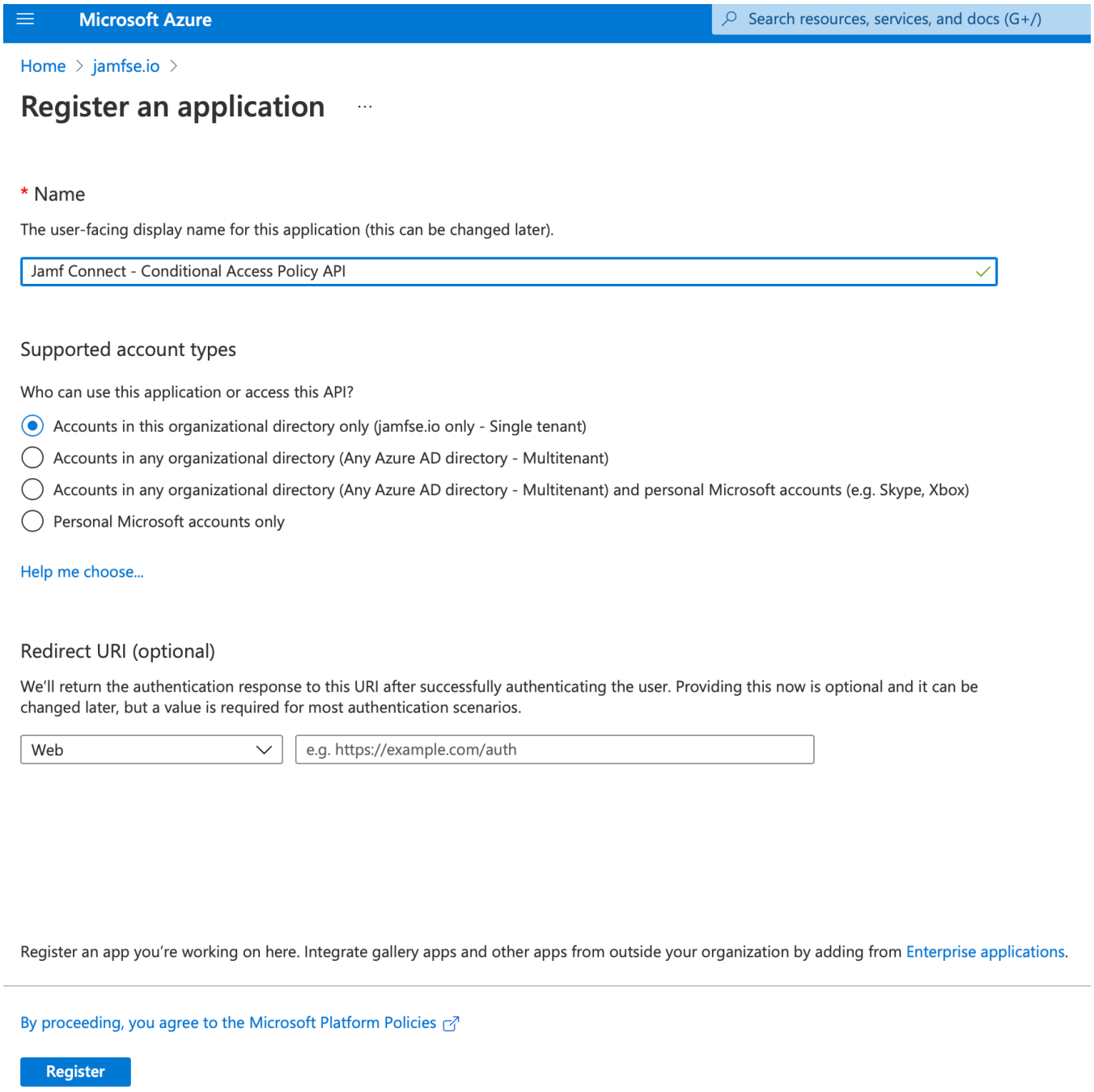
- Create a “private endpoint” application registration with a custom API
  - With API permissions for “User.read”
  - With “Expose an API” scope created
  - Define roles like “Admin” and “Standard” for elevating macOS account permissions
- Create a “public endpoint” application registration for OIDC to call that custom API
  - Add API permission for “My APIs” for the name of the application created in first step and the scope created in first step
- Create an Azure Conditional Access policy to require multifactor authentication
  - Apply to application created in first step
- Remove any CA policy applied to “All cloud apps” that would require MFA
- Create a Jamf Connect Login configuration profile
  - Azure as Identity Provider
  - Define a custom scope
  - Define the Discovery URL for OIDC and ROPG
  - Test with Jamf Connect Configuration

# Minimum system requirements

These instructions were written assuming you are using Jamf Connect version 2.14 or greater.

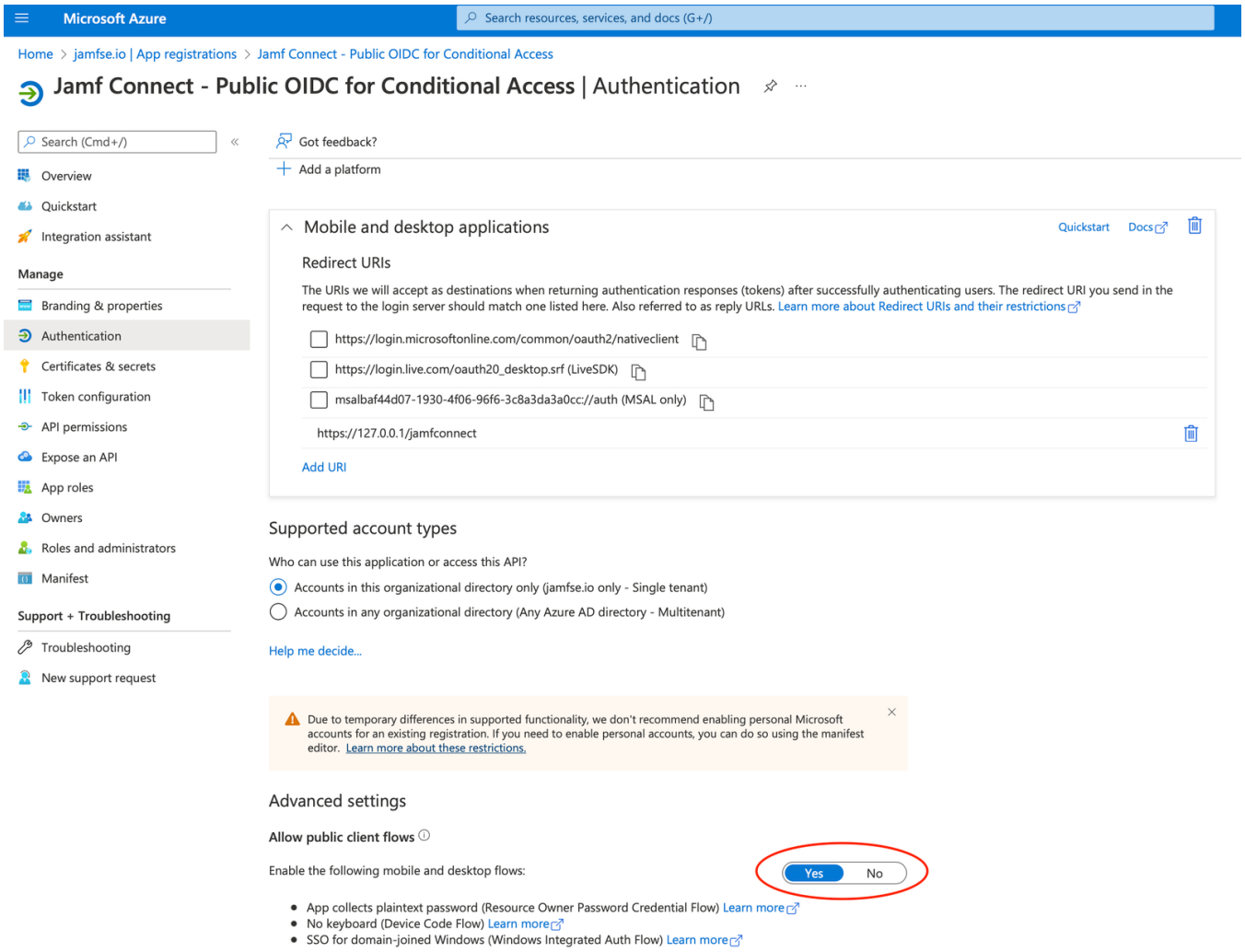
# Step One: Create an application registration with a custom API

Navigate to portal.azure.com → Azure Active Directory → App Registrations. Create a new app registration. Name the application something like “Jamf Connect - Conditional Access Policy API”. Select the supported account types to “Accounts in this organizational directory only”. Leave Redirect URI section blank. Register the application.



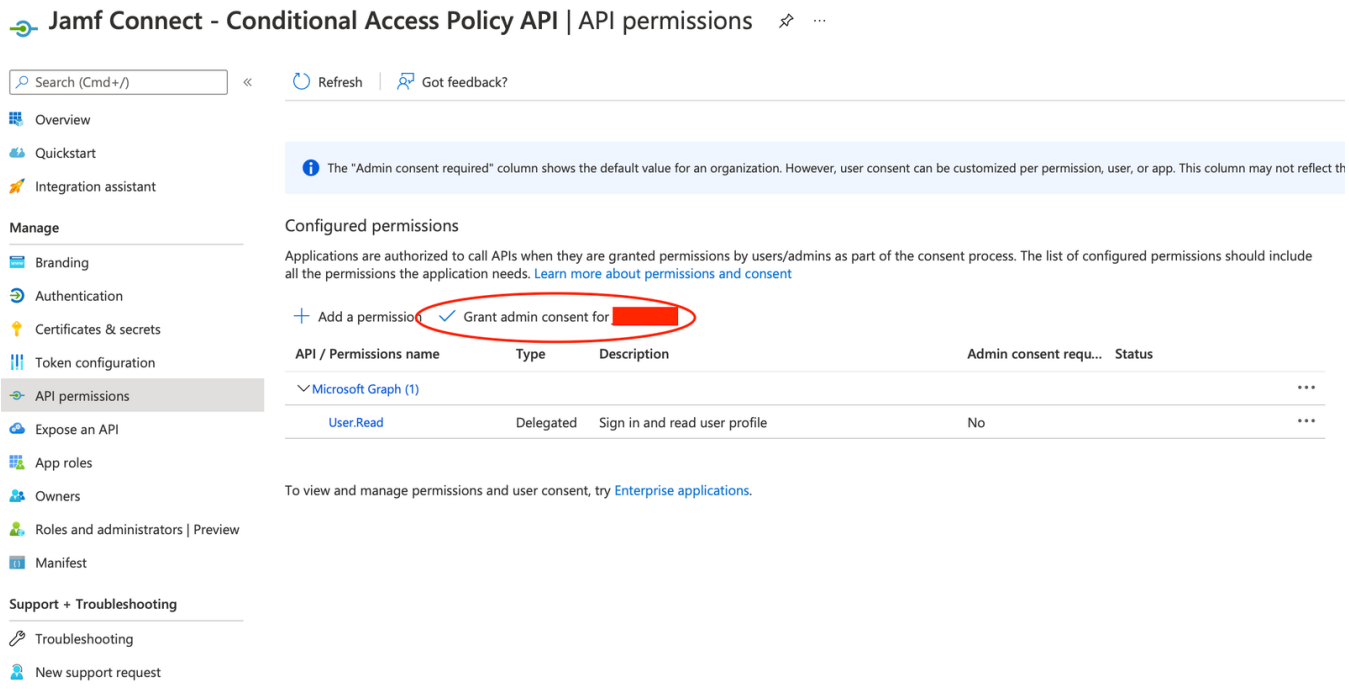

*Application Registration Screen (as of 06DEC2021)*

Navigate to Authentication on the left hand navigation bar. Set “Allow public client flows” to “Yes.” (This feature enables Resource Owner Password Grant or ROPG to validate passwords.)

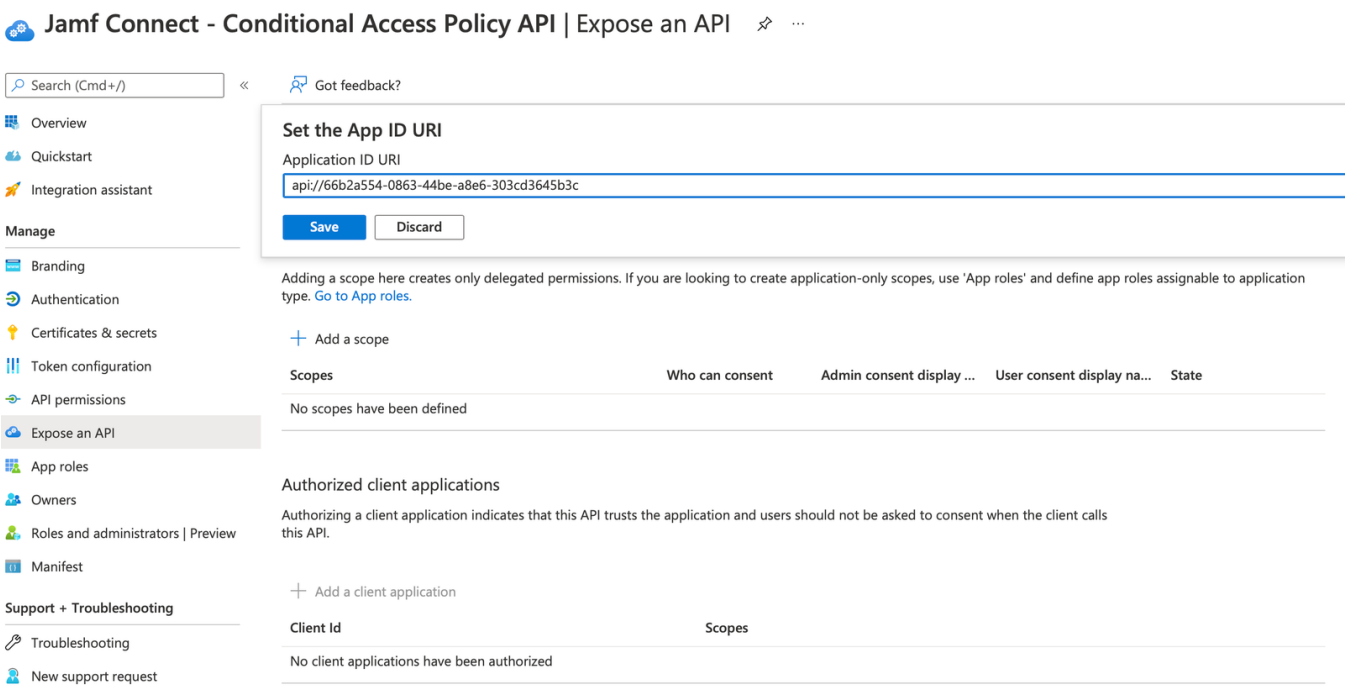



*Authentication page for an App registration (25AUG2022)*

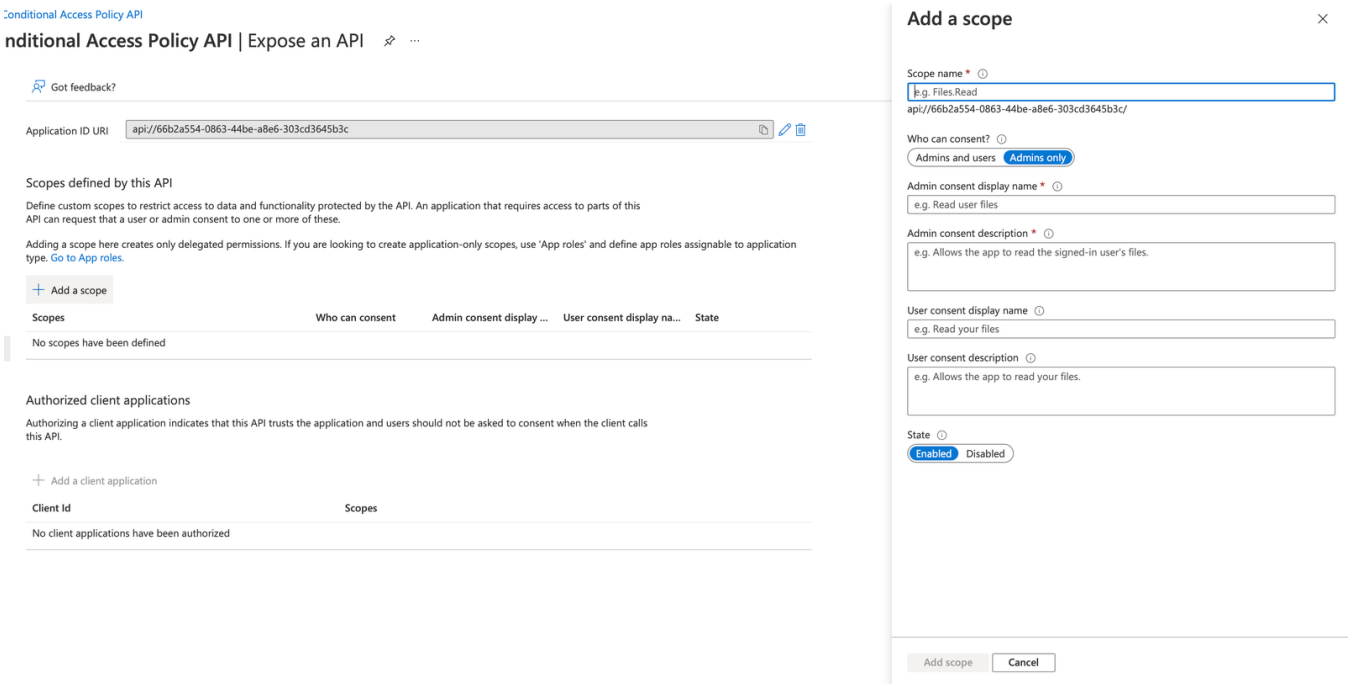
Navigate to API permissions on the left hand navigation bar. Grant admin consent for the organization.



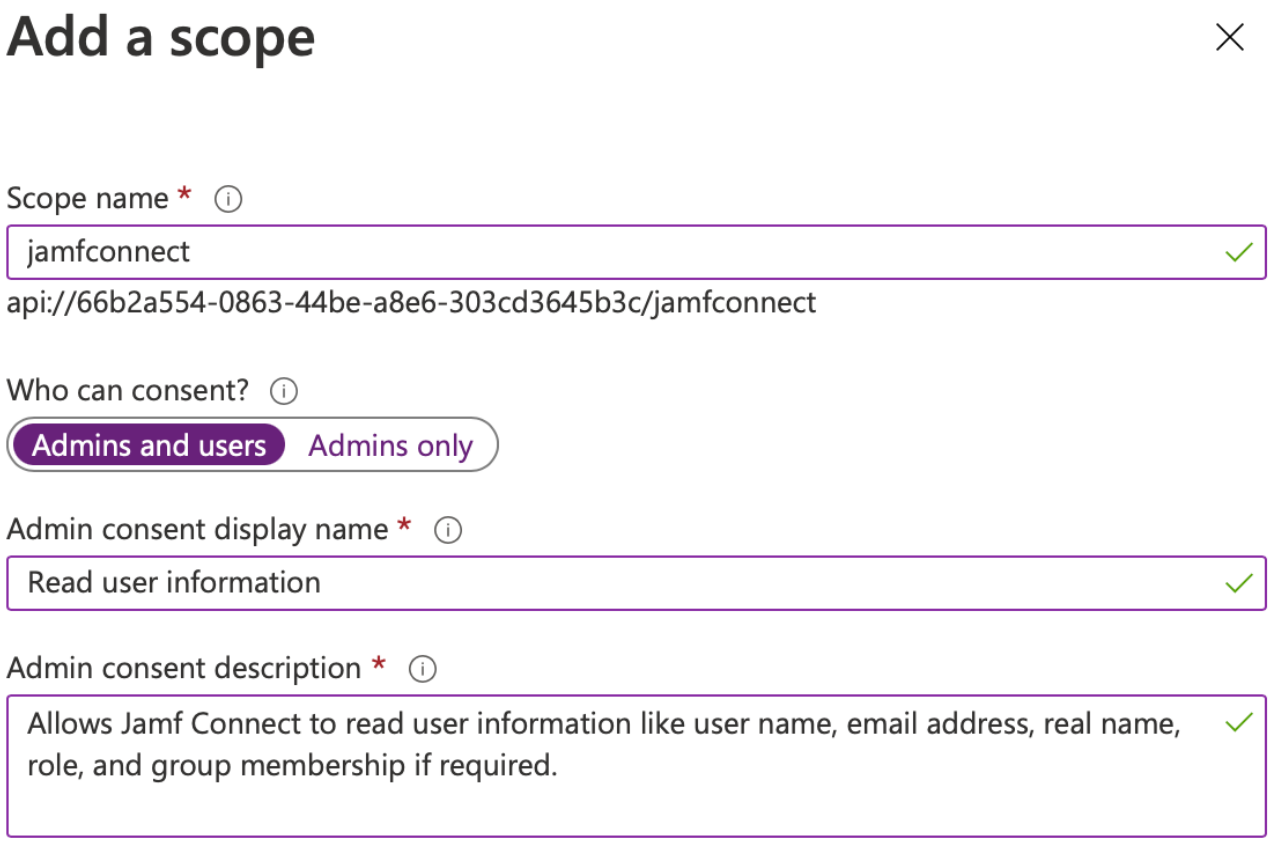

Using the left hand navigation bar, select “Expose an API”. Set the Applicaiton ID URI. A default entry will be created based on the pattern of `api://[application ID]`. This may be modified if desired but default entry is acceptable.

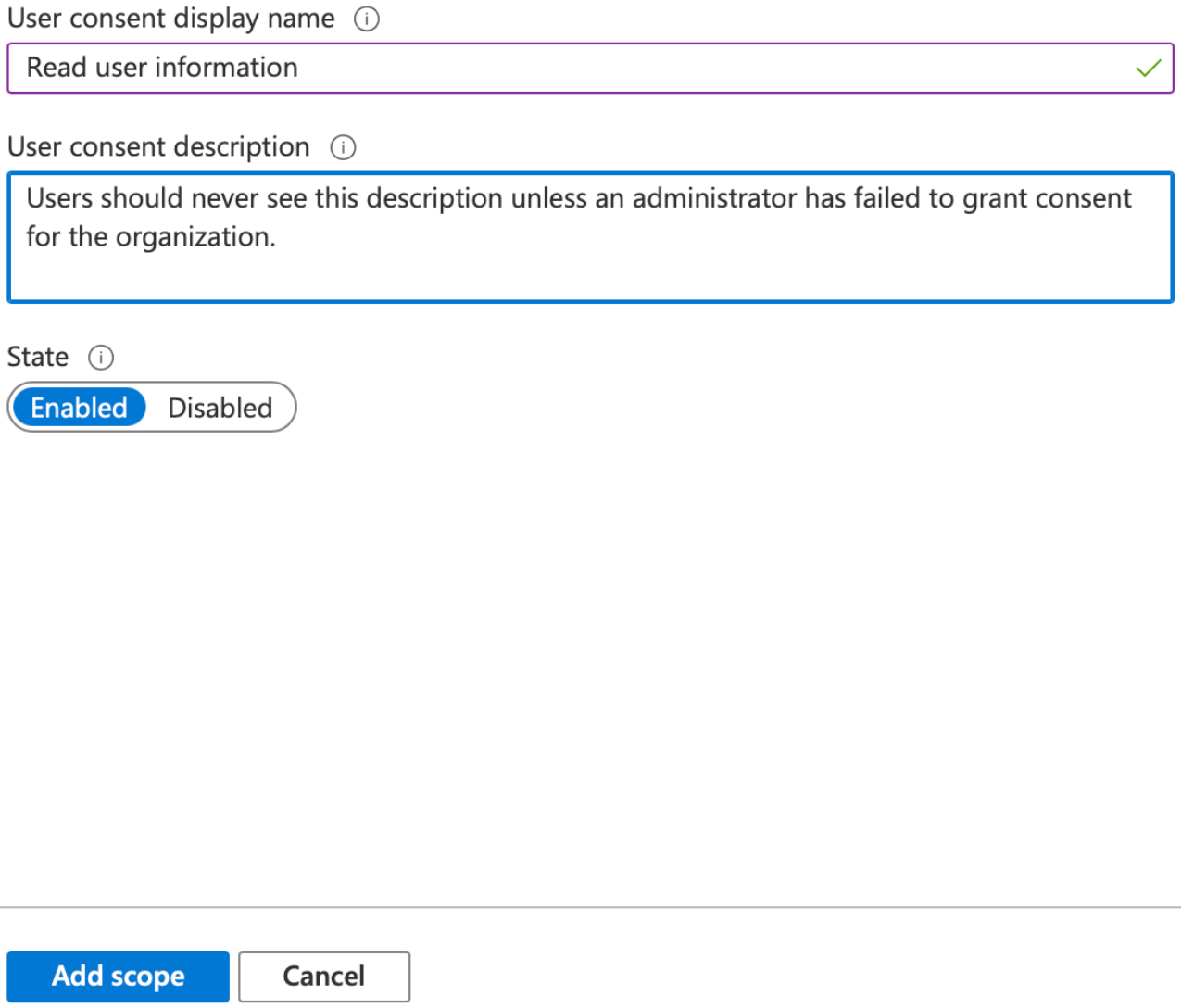



Select the option for “Add a scope”




Set the scope name to `jamfconnect`. Set “Who can consent” to “Admins”. Enter information into the Admin consent display name and Admin consent description. Any text is acceptable - this will be accepted by the admin in the next step. Press “Add scope” to save.



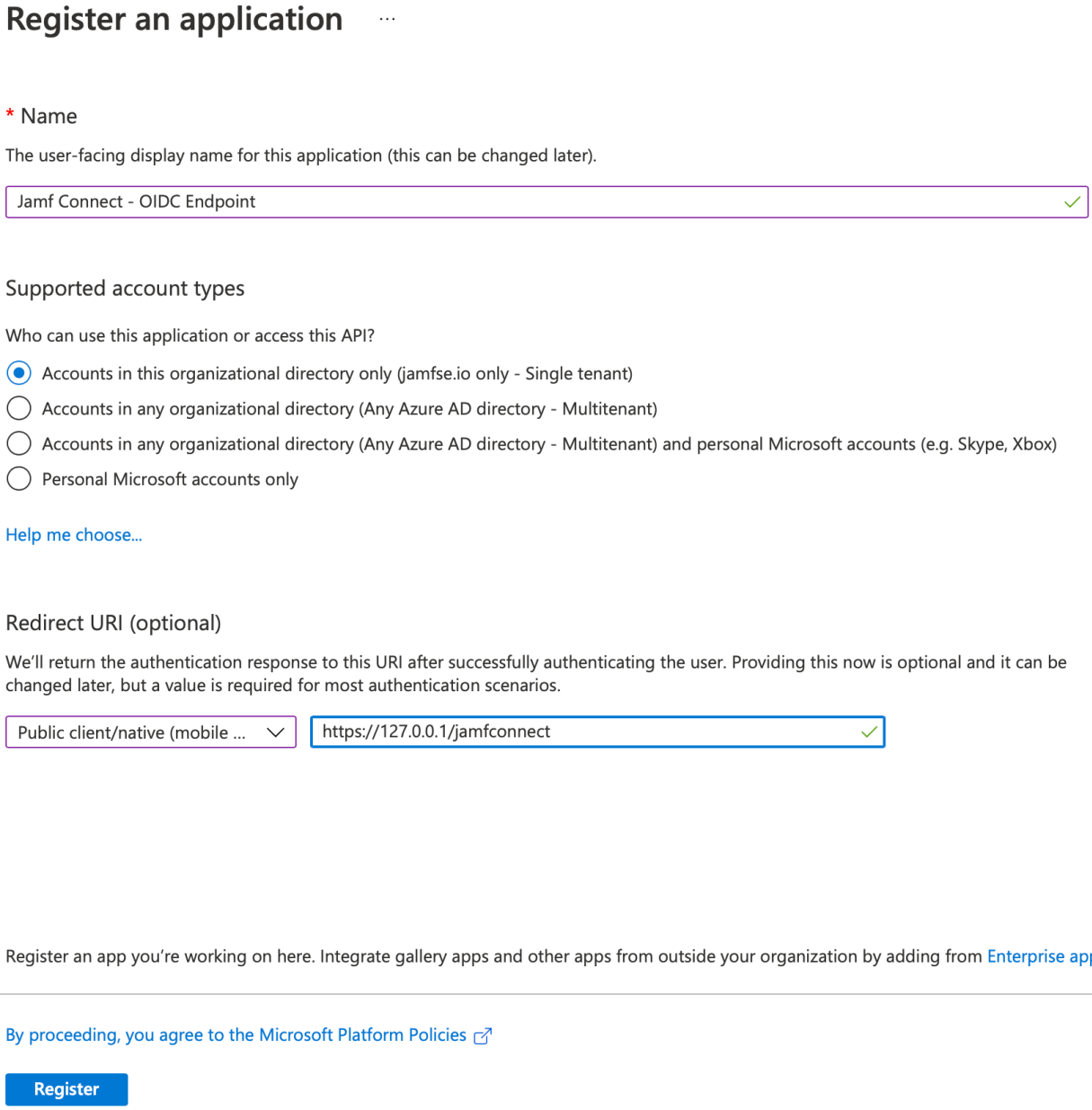



Copy the scope with the Copy button and save it for later. This will be used as the `OIDCScopes` later in Jamf Connect Configuration.

# Step Two: Create an application registration using this new API permission

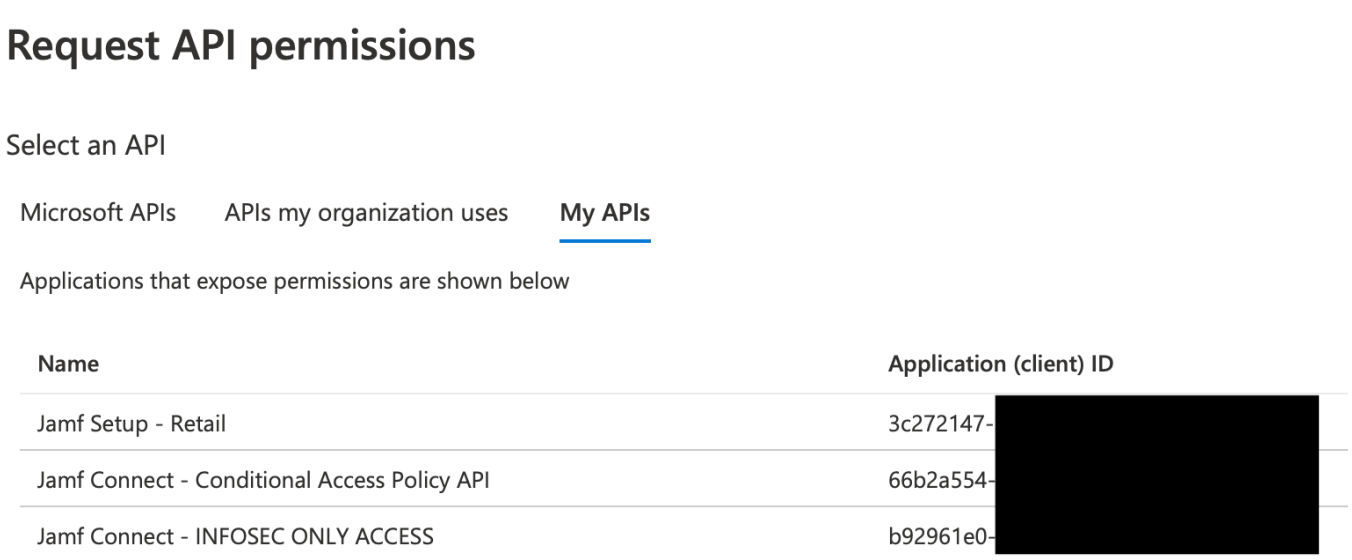
Return to Azure Active Directory → App Registrations. Create a new app registration. Name the app “Jamf Connect - OIDC Endpoint”. Set Supported account types to “Accounts in this organizational directory only”. Set Redirect URI to “Public client/native (mobile & desktop)” with the value

`https://127.0.0.1/jamfconnect`. Register the application.

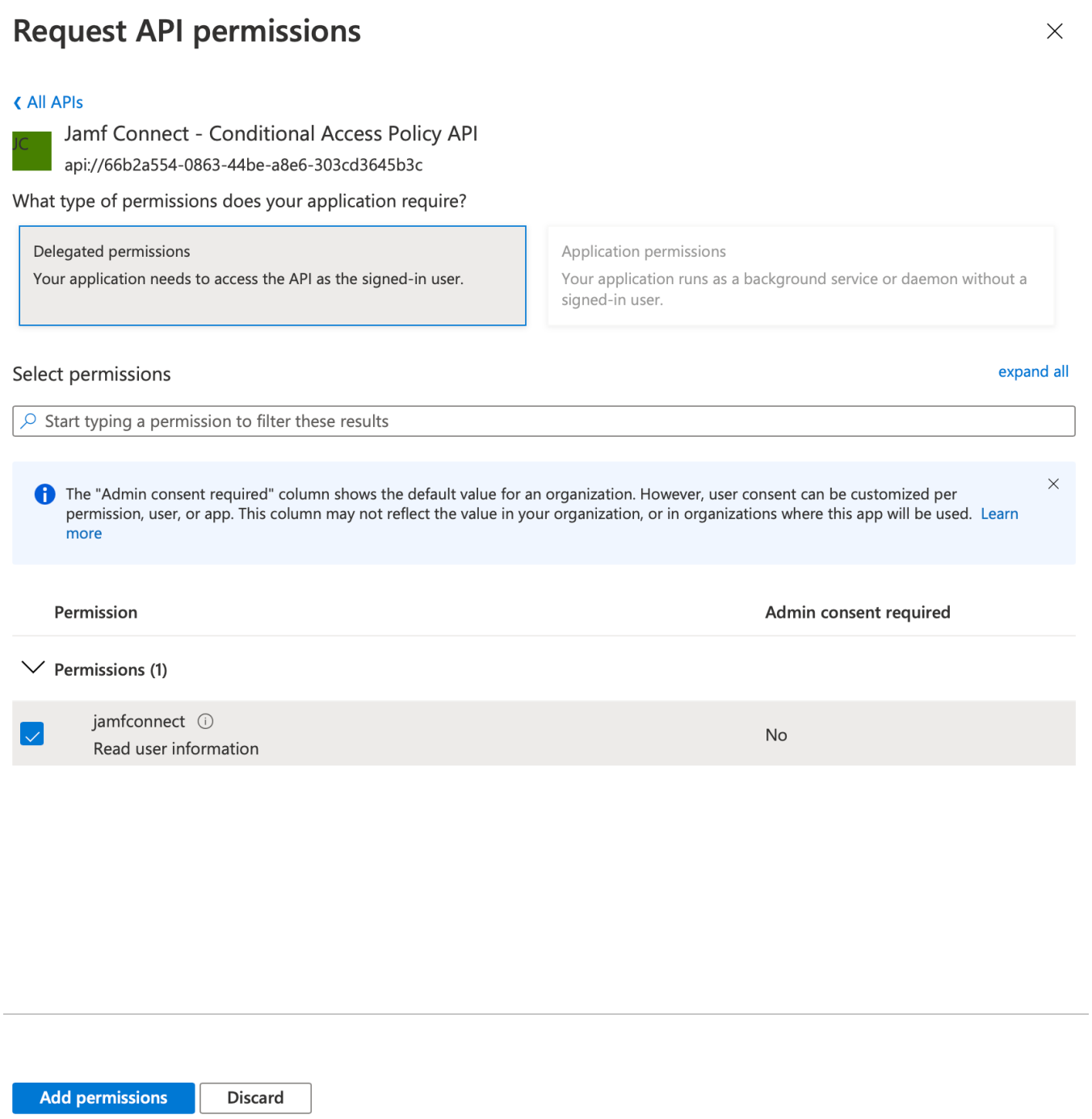



Navigate to API permissions. By default, the Microsoft Graph → User.Read permission is added. Use the “Grant admin consent for [domain] button to grant permission to read the user information on behalf of the user. (The “public” app with the defaults scope of `openid` will still be used by the ROPG process to validate a user’s password, thus user.read permissions are required.)

Next, select “+ Add a permission”. Select the “My APIs” tab. Select the name of the application you created in step 1.




Select the option for “Delegated permissions” and check the box for “jamfconnect” - the only permission listed in the application. Use the “Add permissions” button to close the window.

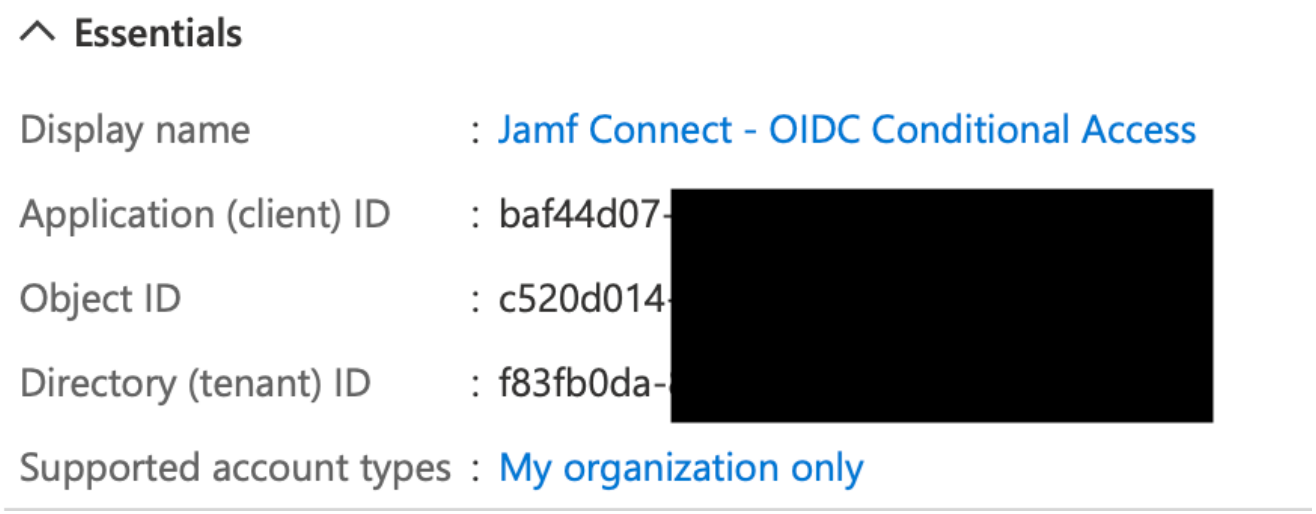



Use the “Grant admin consent for [domain]” to grant permission to access the API on behalf of users.

**Optional**: Use the “App roles” option to add a role for “Administrator” and “Standard”. This will allow you to define users or groups of users directly in Azure who should have administrator rights on macOS client machines. “App roles” is located on the left hand navigation tool bar in the App registration - refer to

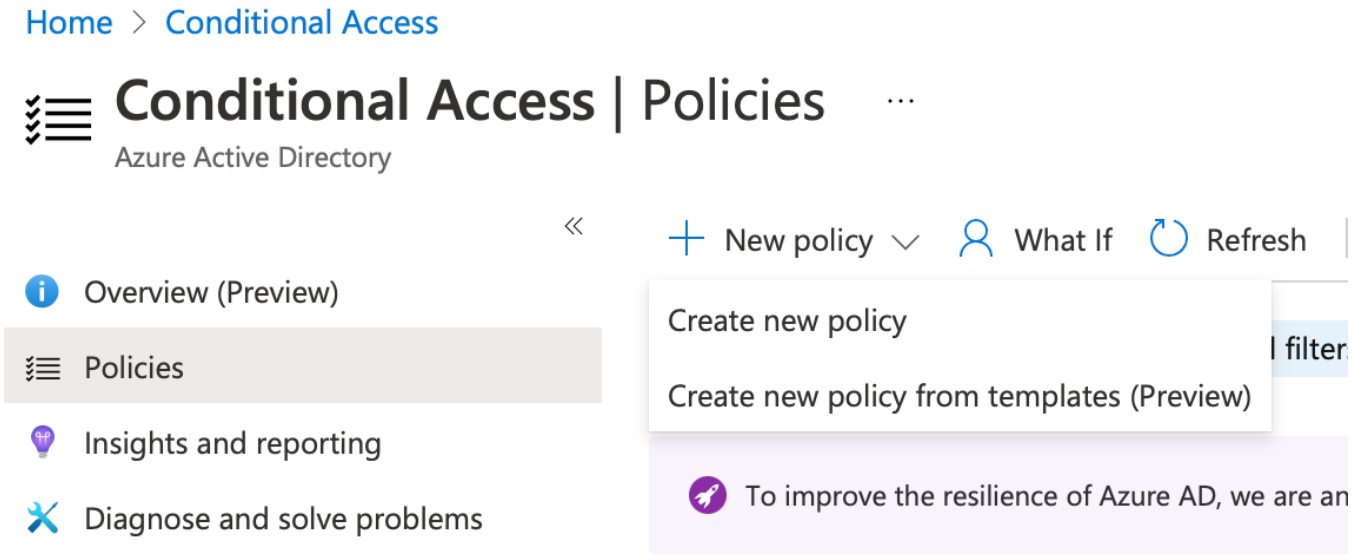
https://docs.jamf.com/jamf-connect/documentation/Login_Window_Preferences.html for more details on the `OIDCAdminAttribute` and `OIDCAdmin` settings for Jamf Connect.

Navigate to Overview.  Record the Application (client) ID and the Directory (tenant) ID for later use with Jamf Connect Configuration.




Navigate to Azure Active Directory → Enterprise Applications.  Find the Jamf Connect - OIDC Endpoint application you created and assign users and roles to the application.

H3: **Step Three: Create an Azure Conditional Access policy**
Navigate to portal.azure.com → Azure Conditional Access.  Create a new policy.




Name the policy as desired.  The sample will name the policy “Jamf Connect - Require Multifactor Authentication”

Home > Conditional Access >

# New ...

Conditional Access policy

Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies.
Learn more

Name *

Jamf Connect - Require Multifactor Auth...

## Assignments

Users or workload identities

0 users or workload identities selected

Cloud apps or actions

No cloud apps, actions, or authentication contexts selected

Conditions

0 conditions selected

## Access controls

Grant

0 controls selected

Session

0 controls selected

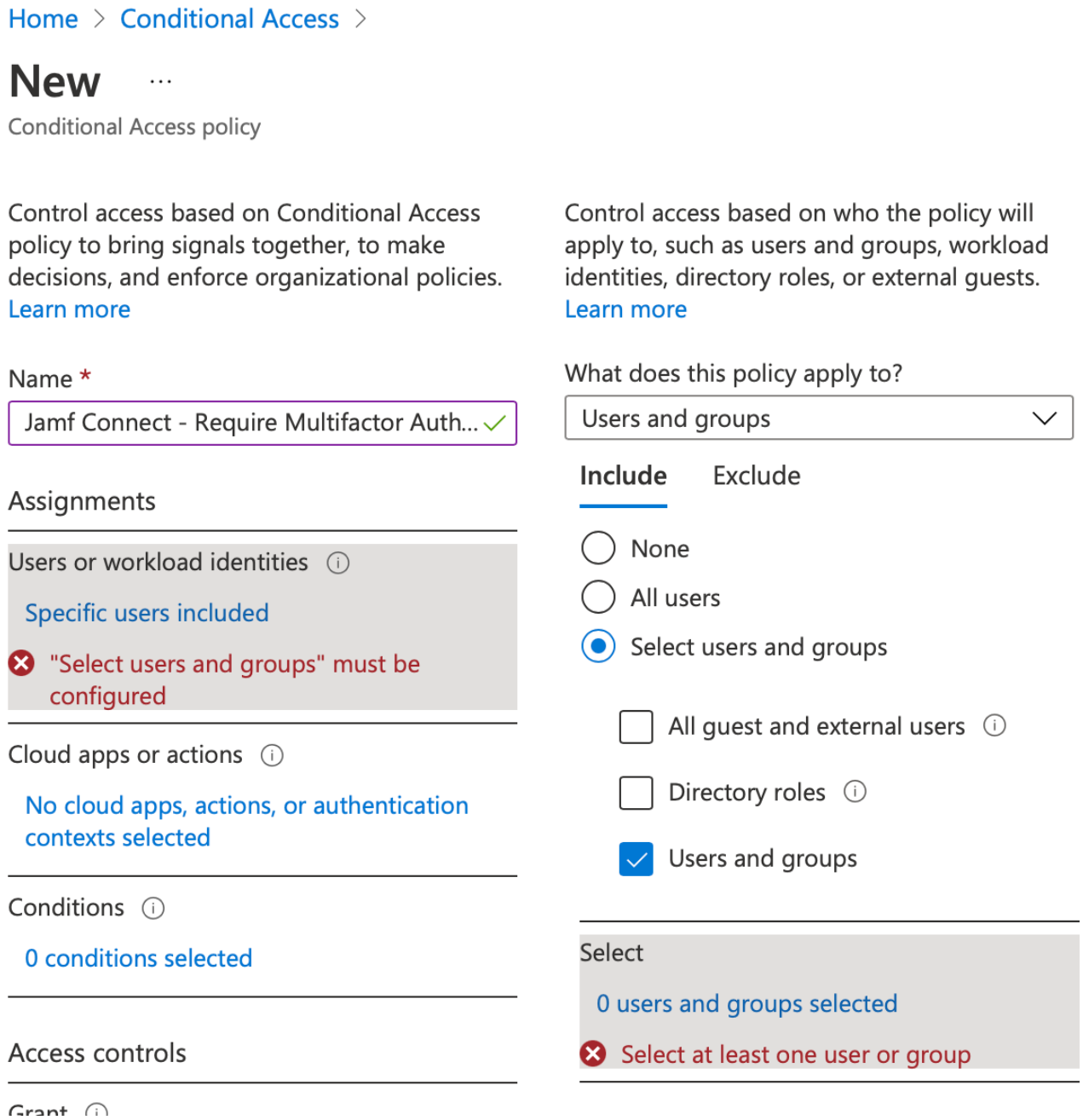
Enable policy

Report-only On Off

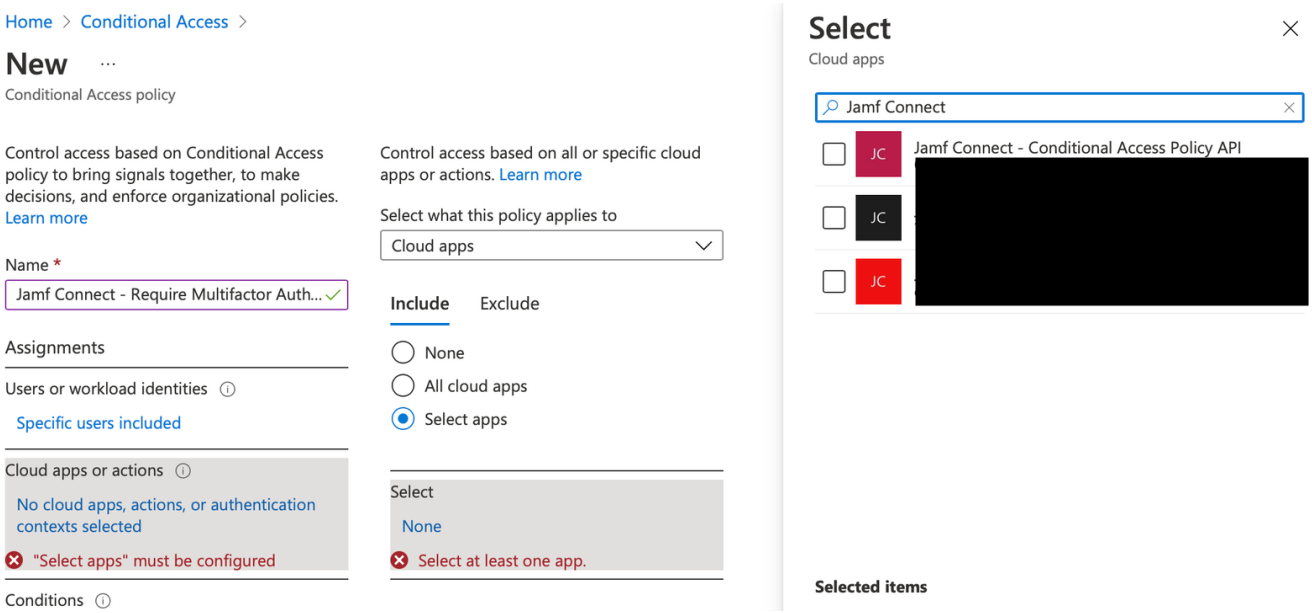
Do not block yourself out! This policy impacts the Azure portal and other clients that do not support CAE today.

Create

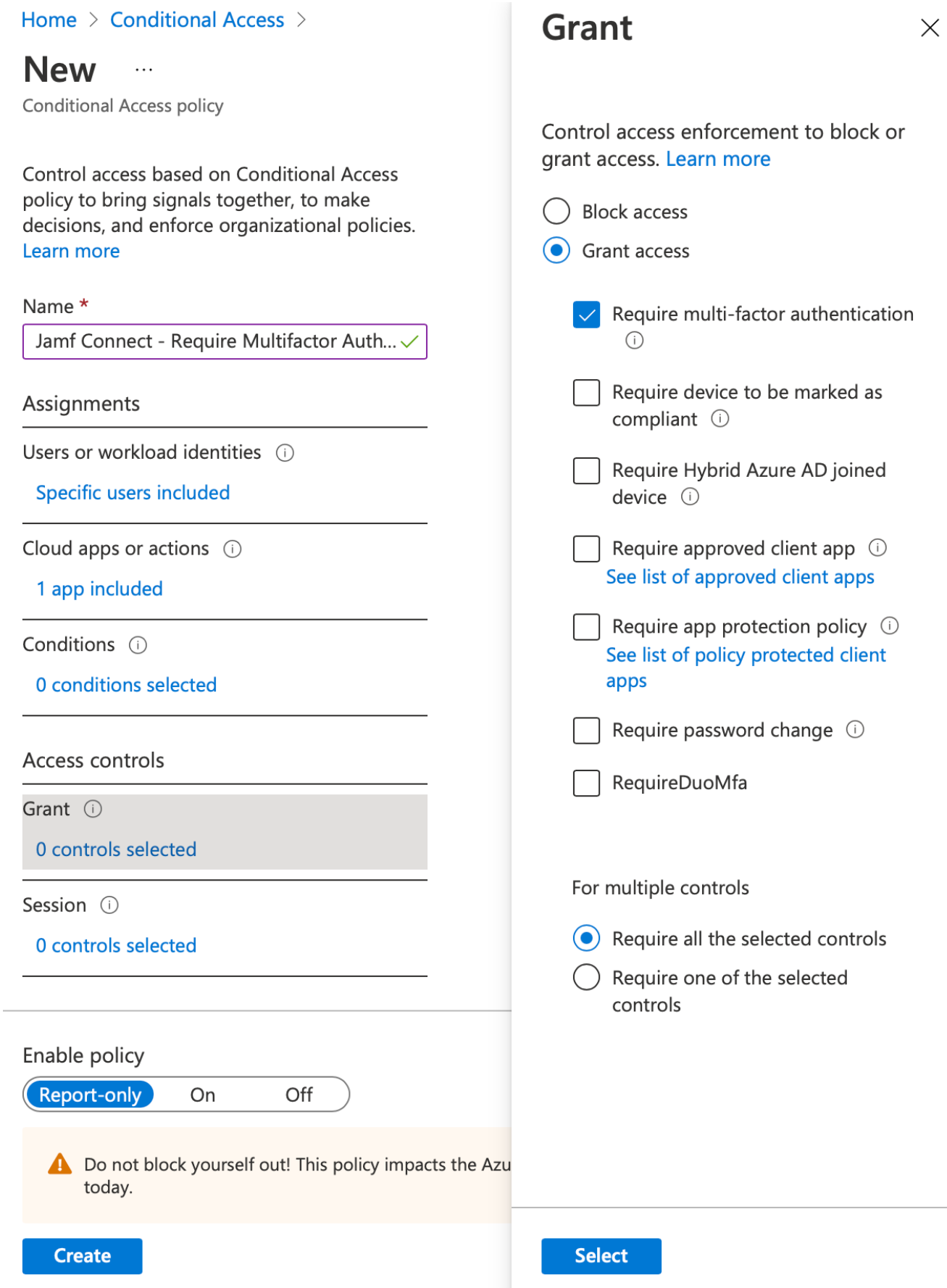
Select “Users or workload identities”. Select a test user to test your conditional access policy before applying to all users.




Select “Cloud apps or actions”. Select the Jamf Connect - Conditional Access Policy API you created in step one.




Select “Grant”. Check the option for “Require multi-factor authentication”. Set Enable policy to “On” and “Create” to save the policy.


Home > Conditional Access >
New
Conditional Access policy
Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. Learn more
Name *
Jamf Connect - Require Multifactor Auth...
Assignments
Users or workload identities
Specific users included
Cloud apps or actions
1 app included
Conditions
0 conditions selected
Access controls
Grant
0 controls selected
Session
0 controls selected
Enable policy
Report-only
On
Off
Do not block yourself out! This policy impacts the Azu
today.
Create
Grant
Control access enforcement to block or grant access. Learn more
Block access
Grant access
Require multi-factor authentication
Require device to be marked as compliant
Require Hybrid Azure AD joined device
Require approved client app
See list of approved client apps
Require app protection policy
See list of policy protected client apps
Require password change
RequireDuoMfa
For multiple controls
Require all the selected controls
Require one of the selected controls
Select

# Step Four: Remove any Conditional Access policies applied to All cloud apps

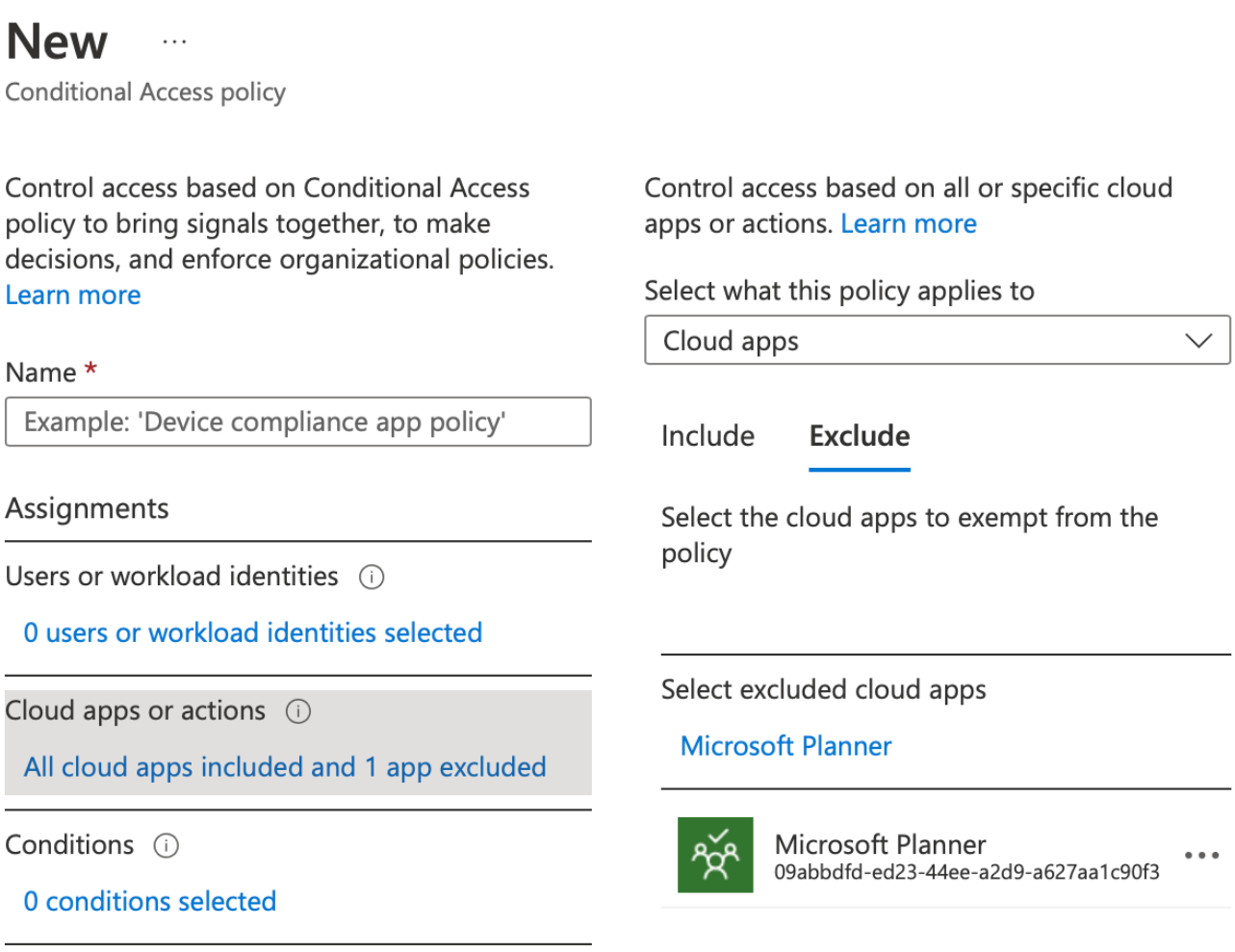
## Preferred method

Navigate to portal.azure.com → Azure Conditional Access. Examine any application applied to the scope of “All cloud apps”. Either set “Enable policy” to “Off” for any application that has a Grant of “Require multi-factor authentication” or modify the “Cloud apps or actions” to specifically list resources that should have MFA applied.

Applying a policy to require MFA for “All cloud apps” will cause the ROPG application in the next step to inaccurately show failed logins in the Azure sign-in logs.

## Alternative method

**WARNING: Contains undocumented behavior, subject to change by Microsoft at any time.** If you wish to keep “All cloud apps” as a definition, but you still want the policy to not be applied to the `openid` scope, create an unused Enterprise app registration for an unused SAML application, and then use that bogus app registration as an “Exclude” to the “All cloud apps” policy:




This is undocumented behavior, but the application of an exclusion to the policy will break how the policy is applied to `openid` scope. As this is undocumented, the behavior may be unexpected and may change by Microsoft without notice.

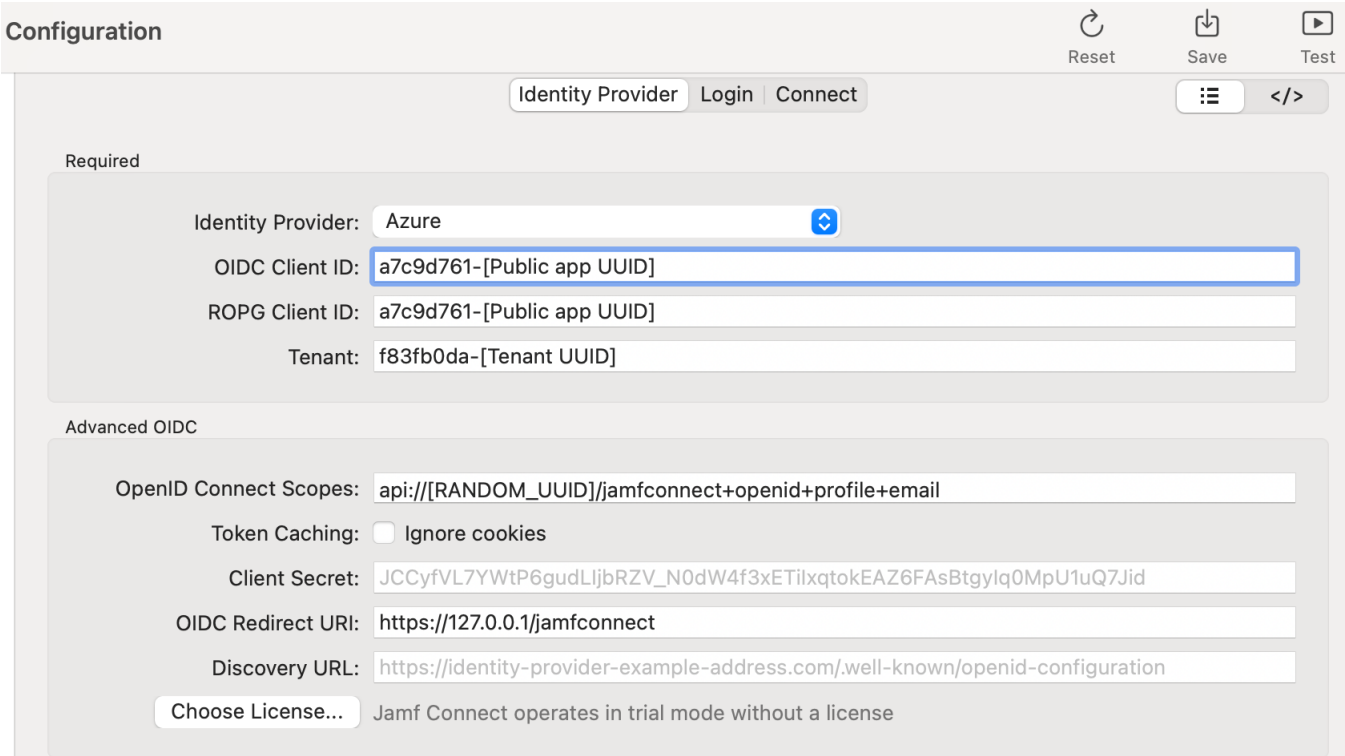
# Step Five: Create a Jamf Connect Configuration Profile

Use the Jamf Connect Configuration app included in the Jamf Connect software distribution disk image which you can download from account.jamf.com with your Jamf Nation credentials.

On the Identity Provider tab, set:

- Identity Provider: Azure

- OIDC Client ID: The application ID of the PUBLIC application you created in Step Two
- ROPG Client ID: The same application ID
- Scopes: Combine the scope you saved in Step One with `+openid+profile+email` to look similar to: `api://[RANDOM UUID STRING]/jamfconnect+openid+profile+email`
- Tenant: Enter the UUID of the tenant of your Azure instance. This can be found under the "Overview" tab of either of the App registrations made in Step One or Step Two.
- OIDC Redirect URI: (optional) Set to `https://127.0.0.1/jamfconnect`




**OPTIONAL**: If you want to define a role for users to be made administrators on a macOS client device, on the Login tab, set:

- User Creation → Admin Roles: The value of the administrator App role you created in Step Two
- User Creation → Admin Attribute: `roles`

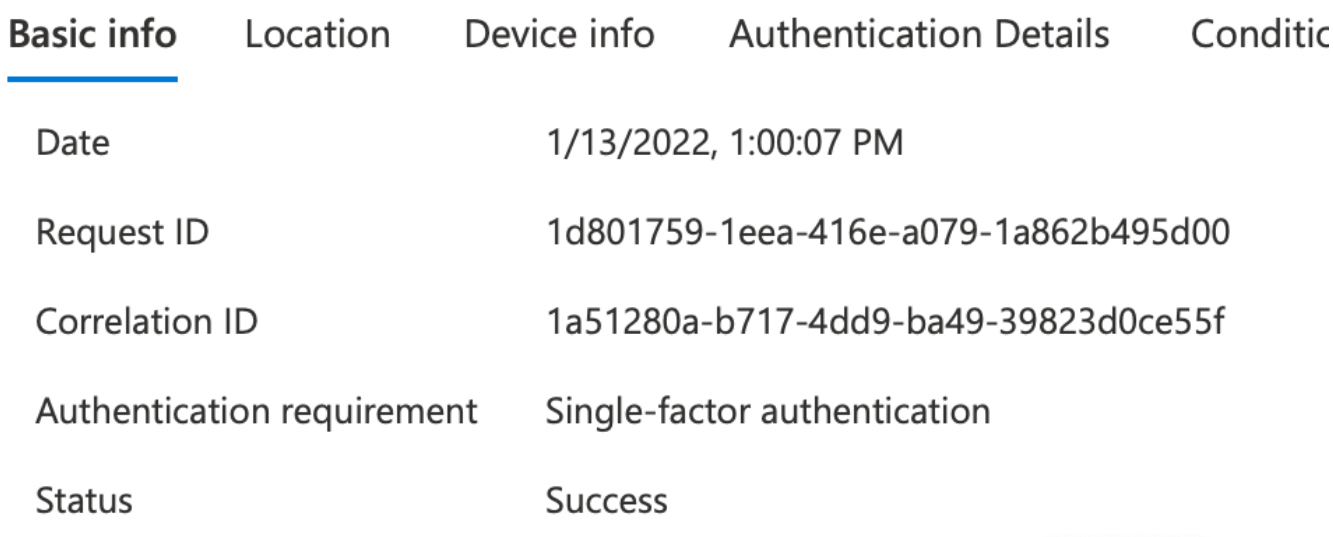
On the Connect tab, set:

- Authentication

- ROPG Client ID: This should auto populate from your entry on the Identity Provider screen
- ROPG Tenant: The UUID of the Azure tenant
- ROPG Scopes: Set value to `openid+email+profile`

Test your configuration with the test user via OIDC. Make sure MFA was required.

Test your configuration with the test user via ROPG. Validate in the Azure portal under Azure Active Directory → Sign-in logs that the authentication was successful. Look for the Authentication Requirement to be “Single-factor authentication”. The Basic tab will show something like:




The Conditional Access tab should show that no policy was applied to the login.

**Note**: Administrators will still see a single MFA failure immediately after a user logs in using the Jamf Connect login window. As of Jamf Connect 2.14, the login window will use the scopes defined in the `OIDCScopes` key which will trigger the need for MFA for the ROPG process. This behavior is expected and harmless.

**Copyright and Trademarks**


Jamf has made all efforts to ensure that this guide is accurate.

Jamf
100 Washington Ave S Suite 1100
Minneapolis, MN 55401-2155
(612) 605-6625