

Title: Microsoft Azure AD Conditional Access principles and guidance.

Author: Claus Jespersen, Principal Security Consultant in Microsoft AC&AI WE

Twitter: @claus_jespersen

LinkedIn: <https://dk.linkedin.com/in/claus-jespersen-25b0422>

Date: 3'th. October 2023

Contents

Introduction.....	3
Changelog.....	5
CA related components.....	6
General Field Guidance	7
Governance/Roll-out	7
Personas	8
Policy Types	11
CA Principles and recommended best practice.....	15
CA Exclusions	15
Conditional Access Architecture.....	17
Suggested Policies	20
Global Policies (CA001-CA099)	20
Global Base Protection policies	20
Global Attack Surface Reduction policies.....	21
Admins Policies (CA100-CA199)	21
Admins Base Protection policies	23
Admins Identity Protection policies	23
Admins Data and App protection	25
Admins Attack Surface Reduction	26
Internals Policies (CA200-CA299)	26
Internals Base Protection	26
Internals Identity Protection	27
Internals App and Data Protection	28
Internals Attack Surface Reduction	28
Externals Policies (CA300-399)	29
Externals Base Protection.....	29
Externals Identity Protection.....	29
Externals App and Data Protection	30

Externals Attack Surface Reduction.....	31
Guests Policies (CA400-CA499)	31
Guests Base Protection.....	31
Guests Identity Protection.....	31
Guests App and Data Protection	32
Guests Attack Surface Reduction	32
Guests Compliance Protection	33
GuestAdmins Policies (CA500-CA599).....	33
GuestAdmins Base Protection	33
GuestAdmins Identity Protection	34
GuestAdmins App and Data Protection.....	34
GuestAdmins Attack Surface Reduction.....	35
GuestAdmins Compliance Protection.....	35
Microsoft365ServiceAccounts Policies (CA600-CA699)	36
Microsoft365ServiceAccounts Base Protection	36
Microsoft365ServiceAccounts Identity Protection	36
Microsoft365ServiceAccounts Attack Surface Reduction Protection	36
AzureServiceAccounts Policies (CA700-CA799).....	37
AzureServiceAccounts Base Protection.....	37
AzureServiceAccounts Identity Protection.....	37
AzureServiceAccounts Attack Surface Reduction Protection.....	38
CorpServiceAccounts Policies (CA800-CA899)	38
CorpServiceAccounts Base Protection	38
CorpServiceAccounts Attack Surface Reduction Protection	39
WorkloadIdentities Policies (CA900-CA999)	39
WorkloadIdentities Base Protection.....	39
Developers Policies (CA1000-CA1099)	40
Developers Base Protection	40
Developers Identity Protection	44
Developers App and Data Protection	45
Developers Attack Surface Reduction	46
Change Management	46
CA Deployment Model - staged deployment and test	46
CA Security Groups	52

Report-Only mode testing considerations	65
CA Workbooks and insights	68
Conditional Access Overview blade.....	70
CA Automation	71
Export current CA policies	74
Azure AD app registrations.....	77
Azure Subscription configuration	77
Azure DevOps configuration.....	78
References:	91

Introduction

This document describes various principles and recommendations for a Conditional Access framework aligned with modern security principles based on experience from various enterprise customer engagements. The framework is to be considered as a starting point incorporating strong basic protection from the start based on Zero Trust principles.

Please notice that this is not to be considered as official Microsoft guidance but rather a “Notes from the field”. Please also consider looking at formal Microsoft documentation located in various sources on the Internet, including docs.microsoft.com”.

I am in dialog with the product group and our document writers to see if/how we can incorporate some of this guidance in our formal Microsoft CA related documentation. Also stay tuned for more structure to this guidance doc including more architecture/design considerations.

Also my colleague, Alex Filipin has lots of good guidance and a reference for this doc [What is Conditional Access in Azure Active Directory? | Microsoft Docs, Conditional access guidance · AlexFilipin/ConditionalAccess Wiki · GitHub](#).

Some general/official guidance from Microsoft regarding some common Conditional Access policies is located [here](#)

<https://docs.microsoft.com/azure/active-directory/conditional-access/concept-conditional-access-policy-common>

Latest updates include various smaller changes to CA policies for guests, admins and global as well as some more details for the guidance on CA deployment security groups. Also I have included a spreadsheet in the section "Suggested Policies" as a template for all the CA policies to document them in a more separate and readable form.

If you are just starting with Conditional Access, you may find this page useful: [What is Conditional Access in Azure Active Directory? | Microsoft Docs](#)

When applied to a given customer environment, you should expect the suggested policies to be adjusted to fit your specific customers' requirements combined with which licenses are available. Most of the suggested policies are based on E3, with the enforcement of CA policies based on user risk, sign-in risk and device-risk being the exception. So if your customer does not have E5, you will have to leave out these Identity related CA policies.

Some areas that currently are not incorporated into the suggested policies are

- Providing read-only access to EXO/SPO based on CA
- Providing read-only access to other resources using Microsoft Defender for Cloud Apps , MDCA, (previously known as MCAS)
- Protecting data on the fly by using Conditional Access App Control as a session control that forwards requests to MDCA that can apply DLP actions like encrypting the data based on how sensitive the data is (requires MDCA as part of E5). (other than used in the new Developer persona to allow for device code flow i.e.)
- Utilizing some of the new CA features (see [New Azure AD Capabilities for Conditional Access and Azure VMs at RSA 2021 - Microsoft Tech Community](#)) [Introducing Attribute Based Access Control \(ABAC\) in Azure - Microsoft Tech Community](#), like
 - Authentication Context (requires E5) (in Azure PIM, SharePoint Online and MDCA)
 - Filters for devices as a condition (E3)
 - Register or Join devices as a user action with specific grant controls more granular (E3) as opposed to using tenant wise setting in AAD that would require MFA for all identities for device registration
 - Cross-Tenant CA controls as they evolve and being available in public preview
 - Attribute based access (ABAC)

If you are experienced in working with CA, you may want to incorporate some of the above policies from the start, and if not, you may want to wait until you have the suggested policies applied in production and then expand and adjust from there.

Also it is worthwhile to be aware of services that are related to the CA framework for the suggested policies to make sense. Below are some suggestions on CA related areas that will influence the roadmap for your CA architecture and design/implementation.

Changelog

Since the November edition, the following updates have been made

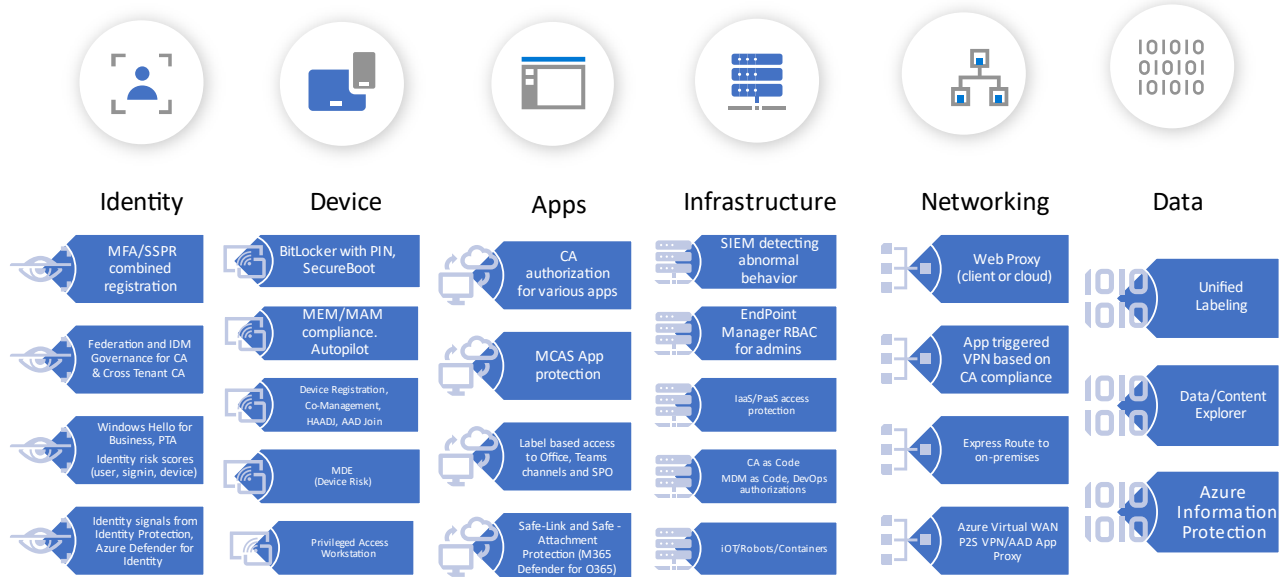
- New sections under change management that provide a few tips on troubleshooting report-only policies as part of a staged deployment.
- New persona introduced, Developers persona covering special developer needs for internals and externals with associated set of suggested CA policies. If you have followed the recommendations of using “All Cloud Apps” everywhere you can, you also know there are some challenges for some scenarios not working out of the box. If so, you want to pay attention to the new base protection policies for the Developers persona as I include a solution to use the ZT/All Cloud Apps approach while allowing for access to device code flow i.e. It incorporates the integration between MCAS and CA in a special way to solve this issue.
- New persona introduced, WorkloadIdentities with associated set of suggested CA policies
- For new personas, I have chosen to keep numbering of existing persona policies and just continue numbering for the new personas. This should make it easier to add new personas to an existing setup for those who have chosen to follow this framework with the suggested numbering.
- I have had requests and even offers about help making the guidance available as GitHub repo. I am open to that, but I want to wait for an internal dialog related to if and how we want to make this guidance part of our official documentation and if so, it will be incorporated in such formal better way of making the information available with the ability to comment, do pull requests i.e.
- A few corrections/changes to existing CA policies
- MCAS wording changed to follow new naming, Microsoft Defender for Cloud Apps (MDCA)
- The use of Microsoft365DSC has been update with information about how to migrate CA policies from one tenant (like a test tenant) to another (like a production tenant).

Last small changes done in October 2023 include

- Change admin policy to require compliant AND MFA in alignment with the Excel file with policies. Also added sign-in frequency every time for risk based policies and intune enrollment policies.

CA related components

Various CA dependencies/relations



Please also consider that for now, Conditional Access is mainly focused on protecting interactive access to resources based on user and device identities. There is an increasing need to also protect access between services when using service principals or managed service identities as well as from a growing number of IoT devices i.e.

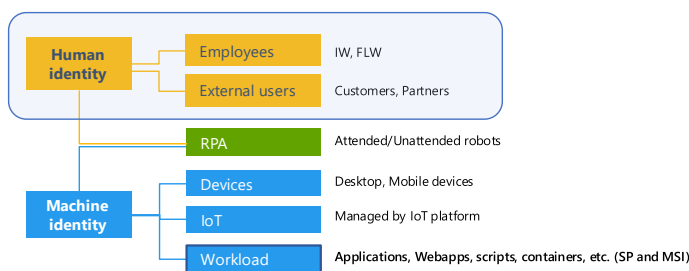
The picture below shows current focus of Conditional Access. Please notice that the non-human access to resources also must be protected. Expect this document to be changed to reflect any such potential changes in the CA policy engine as/if they arrive.

Conditional Access

Protection of various identities

Various identities in Azure AD

CA Protection focuses on human identities



General Field Guidance

Governance/Roll-out

We see many customers who have grown their CA policies over time done by different people without having a consistent naming structure or structure in general. That creates confusing about which policies address what and it is very challenging to figure out what the effects of changing one policy.

So, the recommendation is to start using a consistent naming model for your CA policies

Assure strict naming convention of policies, like <CAnumber>-<Persona/User type>-< Policy Type>-<App>-<Platform>-<Grant>-<OptionalDescription>

Conditional Access Framework

CA Number	Persona	Policy Type	App	Platform	Grant	Description
Component	Description/Examples					
CA Number	Used to quickly identify Policy Type Scope and Order					
Persona	Global, Admins, Internals, Externals, Guests, GuestAdmins, Microsoft365ServiceAccounts, AzureServiceAccounts, CorpServiceAccounts					
Policy Type	BaseProtection, AppProtection, DataProtection, IdentityProtection, AttackSurfaceProtection, Compliance					
App	AllApps, O365 for all O365 services, EXO for Exchange Online i.					
Platform	AnyPlatform, Unknown, Windows, MacOS, iOS, Android					
Grant	Block, ADHJ, Compliant, Unmanaged, where unmanaged is specified in device state condition					
Description	<Placeholder>					

There has been (and still is) many ways of structuring CA policies. One approach is to structure CA policies based on sensitivity of the resource being accessed.

In practice this approach has proven to be very challenging to implement and still protect access to resources for various users. An example would be to define a CA policy that requires known user and known device for access to a sensitive resource that must be accessed by both guests and employees.

As guests come from a managed device, this would not work and you would have to adjust the CA policy to meet both requirements, with typically would result in a policy that only meets lowest denominator (implies less secure).

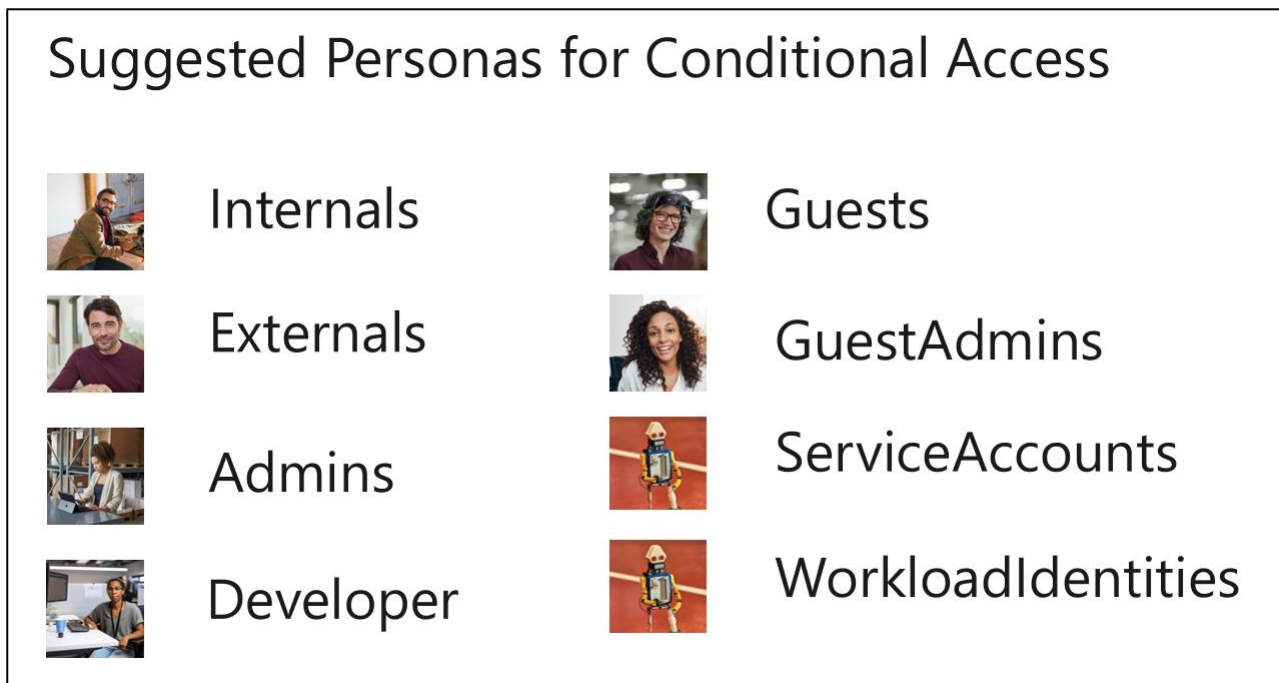
Another approach would be to look at the organization and try to define access policies based on where you are in the organization. However this approach would result in way too many CA policies and seems unmanageable.

A better approach is to structure policies related to common access needs and contain a set of access needs in a persona, representing these needs for various users who have the same needs.

Personas are identity types that share common enterprise attributes, responsibilities, experiences, objectives, and access. We want to emphasize that understanding how enterprise assets and resources are accessed by various personas is integral to developing a comprehensive Zero Trust strategy.

Personas

Some suggested CA personas from Microsoft are shown in the figure below.



Microsoft additionally recommends having separate persona's defined for break glass accounts and for identities that are not part of any persona group. Break glass accounts are excluded from any CA policies and are used in case of emergency where CA may block access for users.

Persona	Description
Global	Global is a persona/placeholder for policies that are general in nature or do not only apply to one persona. So it is used to define policies that apply to all personas or don't apply to one specific persona. The reason for having this persona is to be able to have a model where we can protect all relevant

	<p>scenarios. It should be used to hold policies that apply to all users or policies that enforce protection on scenarios not covered by policies for other personas. An example of a global policy is if you want to have the same policy to block legacy authentication for all users, then you could choose to place it as a global policy as opposed to having a legacy policy per persona that may be different for various personas. Another example is where you want to block a given account or user to specific applications and the user/account is not part of any of the personas. An example would be that if you create a cloud identity in the AAD tenant, then this identity is not part of any of the other personas (given it has not been assigned any Azure AD Roles), but still we may want to hinder this identity in getting access to Office 365 services. Some customers may want to block all access from identities not covered by any persona group, whereas others may want to just enforce MFA.</p>
Admins	<p>We define admins in this context as any non-guest identity (cloud or synced) that have any Azure AD or other Microsoft 365 admin Role (like in MDCA, Exchange, Defender for Endpoints or Compliance). As guests who have such roles are covered in a separate persona, guests are excluded from this persona. Many customers still have separate accounts for sensitive admin roles which this persona is based on. Optimally they use these sensitive accounts from a Privileged Access Workstation (PAW), but often we see that admin accounts are being used on standard workstations where the end-user just switches between accounts on same device/PC. Some customers want to differentiate based on sensitivity of cloud admin roles and assign less sensitive Azure roles to standard end-users (Internals) without using separate accounts for this and rather rely on JIT (Just In Time) elevation. In this case notice that an end-user will be targeted by two sets of CA policies, one for each persona. When using PAWs, you may also want to introduce additional policies that uses device-filters in CA to restrict access for admins to only being allowed when using the PAW.</p>
Developer	<p>The developer persona covers users who have special needs. They are based on AD accounts synced to Azure AD but need special access to services like Azure DevOps, CI/CD pipelines, Device Code Flow, GitHub i.e. The developer</p>

	persona can cover users who are considered Internals as well as Externals, but a person will only be part of one of the personas.
Internals	Internals cover all users who have an AD account synced to Azure AD who are employees of the company and work in a standard end-user role. (Internals who have a developer profile are suggested to be covered by the developer persona)
Externals	This persona holds all external consultants with an AD account synced to Azure AD. (Externals who have a developer profile are suggested to be covered by the developer persona)
Guests	Guests holds all users who have an Azure AD guest account that has been invited into the customer tenant
GuestAdmins	GuestAdmins persona holds all users who have an Azure AD guest account that has any of the mentioned admin roles assigned
Microsoft365ServiceAccounts	This persona covers cloud based (AAD) user-based service accounts used to access Microsoft 365 services where no other solution can cover the need (like using a managed service identity)
AzureServiceAccounts	This persona covers cloud (AAD) user-based service accounts used to access Microsoft Azure (IaaS/PaaS) services, where no other solution can cover the need (like using a managed service identity)
CorpServiceAccounts	This persona covers user-based service accounts originating from on-premises AD used from on-premises or from an IaaS based virtual machine in another (cloud) datacenter, like Azure, synced to Azure AD that accesses any Azure or Microsoft 365 service (should be avoided)
WorkloadIdentities	This persona covers machine identities, like Azure AD service principals and managed identities. CA now supports protecting access to resources from these accounts.

For service accounts, you want to consider.

- Where is the account homed/created (typically AD on-premises or Azure AD)?
- From where the account is used, like Azure PaaS, Azure IaaS VM or from on-premises server?
- What resource is the service account targeting, like Exchange Online, SharePoint online, Azure PowerShell, Azure AD i.e.?

These factors may influence how many personas you want to cover for access from service account to resources protected by Azure AD CA. If the access needs are different for various types of service accounts used and the implied risk does not allow for just excluding them from existing CA policies, then you want to cover them by their own persona.

Even so the current model includes a way of addressing user-based service account, it is worthwhile remembering that it is not recommended to use those as documented in [Introduction to securing Azure Active Directory service accounts | Microsoft Docs](#). The recommendation is rather to use Managed identities or AAD Service Principals.

Protection of access using the latter two is not currently possible in Conditional Access (stay tuned) but are generally considered more secure as they can be used without managing and exposing a static password.

An overview of Azure Service Authentication and Authorization table is described here: [aad-auth-n-z/readme.md at main · jsa2/aad-auth-n-z · GitHub](#) (by Joosua Santasalo)

Policy Types

The suggested policy types with descriptions are shown in the table below.

Policy type	Description
BaseProtection	For each persona, we want to have a base protection that is covered by this policy type. For users on managed devices, this could typically be known user and known device, whereas for external guests, it could be known user and MFA. The base protection is the default policy for all app for users of the given persona. If a given app should have other policy than the default policy, the idea is to exclude that app from the base protection policy and add an explicit policy targeting only that app. An example would be if the base protection for Internals is to require known user and known device for all cloud apps, but you want to allow for Outlook on the Web (OWA) from any device, then you would exclude Exchange Online from the Base Protection policy and add a separate policy for Exchange Online where you require known device OR MFA.
IdentityProtection	On top of the base protection for each persona, we can have CA policies that relate to identity. Examples are: Block Legacy

	Authentication, Require extra MFA for high user or sign-in risk, Require known device for MFA registration i.e.
DataProtection	Type policy type indicates delta policies that protect data as an extra layer on top of the base protection. Examples includes App Protection Policies for iOS and Android where we can protect and encrypt the data on a phone. (App Protection policies also include app protection, so it can be considered both). Other examples include session policies where data is protected using Azure Information protection on the flow if the data being downloaded is considered sensitive data.
AppProtection	This policy type is another addition to the base protection. An example is if/when you want to allow for web access to Exchange Online from any device. In this case you exclude Exchange from the base policy and create a new explicit policy for access to Exchange, where you for example only allow read-only to Exchange Online. Another example of AppProtection policy would be if we require MFA for Endpoint Manager enrollment. We would then exclude "Intune/EM enrollment" from the base policy and add an app protection policy that requires MFA for Endpoint Manager enrollment.
AttackSurfaceReduction	This type of policy is to mitigate against various attacks, like if a user is coming from an unknown platform, then experiences shows that this could be an attempt to try to bypass CA policies where we require a given platform, hence we may want to block requests coming from unknown platforms to mitigate against this. Another example would be to block access to Office 365 services for Azure Administrators or block access to an app for all users if the app is a known to be bad
Compliance	A compliance policy could be used to require a user to see a "Terms Of Use" for guests accessing customer services. In this case you would have an audit record that proves that the guest user has seen your terms that could include how you use the PII data related to their access. This is very important for example for GDPR and other compliance requirements

App type policy

The table below shows details of App type policy

App	Description
AllApps	Indicates that "All Cloud Apps" is being targeted in the CA policy which means that all endpoints are protected for users' access, both those endpoints that support CA as well as those that don't. Using AllApps does have implications of some scenarios that don't work well with this policy. Using AllApps in the base policy is recommended seen from a security point of view as you then have all endpoints protected by the base policy and new apps showing up in Azure AD will also adhere to this policy automatically.
"AppName"	"AppName" is just an example of an app that the policy addresses, it could be "EXO" for Exchange Online (to not make the policy name too long), or SPO for SharePoint Online.

Platform type

The Platform field in a CA policy name is used as specified in the table below

Platform	Description
AnyPlatform	This indicates that the policy should target any platform. This is typically done by selecting "Any Device" (in CA policy both the word platform as well as device are being used).
iOS	Means that the policy targets the Apple iOS platforms
Android	Means that the policy targets the Google Android platforms
WindowsPhone	Means that the policy targets the Windows Phone platforms
macOS	Means that the policy targets the MacOS platforms
iOSAndroid	Means that the policy targets both the iOS and the Android platforms
Unknown	Means that the policy targets platforms not any of the above. This is typically used by including "Any Device" and excluding all the individual platforms

Grant field

The Grant field in a CA policy name is used as specified in the table below

Grant	Description
MFA	Indicate that the policy requires MFA
Compliant	Indicates that the policy requires a compliant device as determined by Endpoint Manager, so the device needs to be managed by Endpoint Manager
CompliantorAADHJ	Indicates that the policy requires a compliant device or Azure AD Hybrid Joined device. A standard company PC that is domain joined is also Azure AD Hybrid Joined. Mobile phones and Windows 10 PCs that are co-managed or Azure AD Joined can be compliant
CompliantandAADHJ	This indicates that the policy requires a compliant AND Azure AD Hybrid Joined Device
MFAorCompliant	Indicates that the policy requires a compliant device OR MFA if it is not
MFAandCompliant	Indicates that the policy requires a compliant device AND MFA to satisfy this policy
MFAorAADHJ	Indicates that the policy requires an Azure AD Hybrid Joined PC or MFA if it is not
MFAandAADHJ	Indicates that the policy requires an Azure AD Hybrid Joined PC and MFA
Unmanaged	This indicates that the policy is targeting devices that are not known by Azure AD. An example of where this could be used would be to allow for access to Exchange Online from any device

Examples:

- CA001-Global-BaseProtection-AllApps-AnyPlatforms-AADJorCompliant
- CA002-Global-IdentityProtection-AllApps-AnyPlatforms-Block-BlockLegacy
- CA100-Admins-BaseProtection-AllApps-AnyPlatforms-Unmanaged

- CA101-Admins-AttackSurfaceReduction-AllApps-Unknown-Block
- CA200-Internals-AppProtection-O365-Windows10-AADHJ
- CA201-Internals-AppProtection-O365-Windows10-Unmanaged
- CA202-Internals-DataProtection-O365-Windows10-Unmanaged-DLPSessioncontrol
- CA203-Internals-AppProtection-O365-iOSAndroid-EMAppProtection
- CA204-Internals-AppProtection-SPO-iOSAndroid
- CA300-Externals-BaseProtection-AllApps-AnyPlatform
- CA301-Guests-Compliance-AllApps-AllDevices-RequireTOU (TOU = Terms Of Use)
- CA401-GuestAdmins-Compliance-AllApps-AllDevices-RequireTOU (TOU = Terms Of Use)

CA Principles and recommended best practice

- Use report-only mode before putting a policy into production.
- Test both positive and negative scenarios
- Use change and revision control on CA policies
- Automate the management of CA policies using tools like Azure DevOps/GitHub or Logic Apps
- Apply Zero Trust principles to Conditional Access
- Limited use of block mode for general access, only if/where needed
- Assure all applications and platform are protected (CA has no implicit "deny all")
- Protect privileged users in all M365 RBAC systems
- Require password change and MFA for high-risk users and sign-ins
- Restrict access from devices with high risk (Intune compliance policy with compliance check in Conditional Access)
- Protect privileged systems (like Azure Mgt. Portal, AWS, GCP)
- Prevent persistent browser sessions for admins and on untrusted devices
- Block legacy authentication
- Restrict access from unknown or unsupported device platforms
- Restrict strong credential registration
- Consider using default session policy that allows sessions to continue working in case of outage given the satisfied the conditions before the outage ([Resilience defaults for Azure AD Conditional Access | Microsoft Docs](#))

CA Exclusions

- Use security groups for exclusions (cloud-based groups are optimal as changes to such groups would be applied quickly as opposed to a group that is synced from on-premises)
- Create an exclusion group for each policy or maybe just per policy type per persona that normally is empty and is used to provide temporary exclusions. Static exclusions are assigned as a direct exclusion assignment in addition to the temporary exclusions.
- Exclude your emergency access accounts from all policies with an emergency access account exclusion group

- Regularly review your exclusion group members (e.g. Access Reviews)
- Exclude your Azure AD Connect service accounts from policies that would prevent it from syncing with a AADC service accounts exclusion group or consider treating this account as a CorpServiceAccounts persona to limit use of this account. It should only have access to AAD to sync users and attributes from AD to AAD and hence it can be limited in what it can be (mis)used for to limit risk of someone misusing this account.

The above recommendations have proved to work well for most organizations when they create policies from scratch. Another consideration is how to test and manage on-going changes to CA policies.

In the past, a ring-based approach has been used by some customers. A ring represents a given number of users and the idea is to first start applying it to a Ring 0 where only includes very few test users and if/when working, deploy the same policy to a ring including more users, like Ring 1, Ring 2 and Ring 3 and so on before applied to all users.

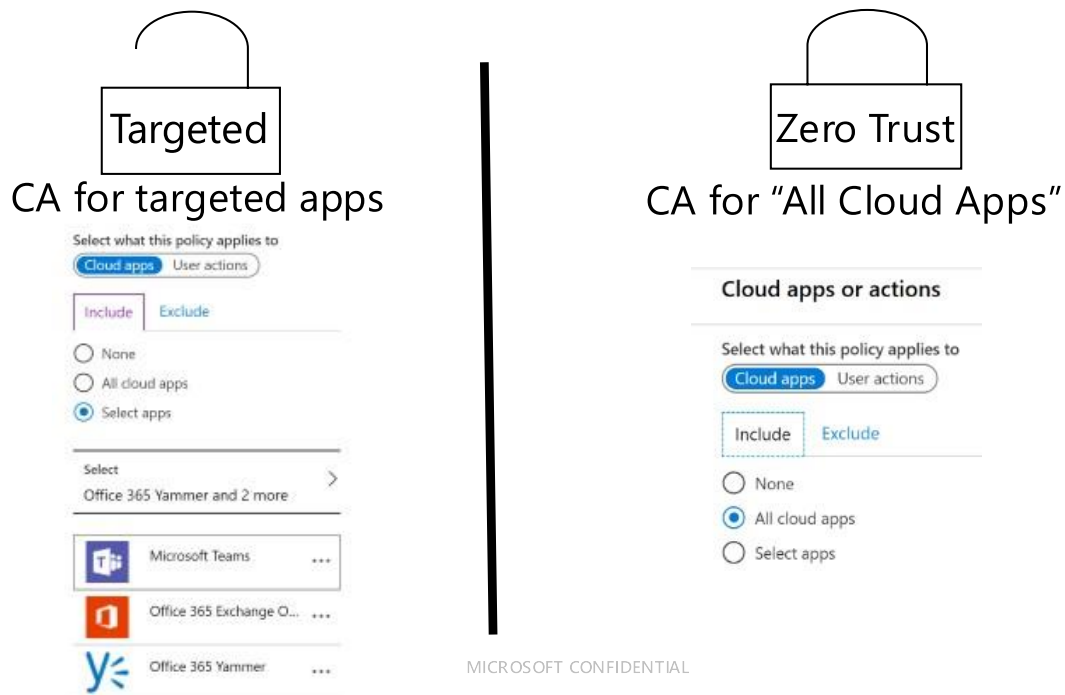
As report-only mode was not available from the start, some customers have chosen to create separate CA policies for each ring. When we consider using a persona-based approach, we would have to duplicate all our CA policies within each persona group reflecting each ring, which would result is a lot of CA policies which increases the complexity in managing the policies.

Instead we suggest a slightly different approach which is documented in a section later in this document (Change management).

Conditional Access Architecture

An important consideration is to choose which architecture the customer wants to pursue. We suggest considering using a Targeted or a Zero Trust CA Architecture. The figure below shows the idea of the two architectures.

Defining CA architecture type



The Zero Trust CA architecture is the one that best fits the principles of Zero Trust (hence the wording). Choosing "All cloud apps" in a CA policy implies that all endpoints are protected by the given grant controls, like known user and known or compliant device. However, the policy applies not only the endpoints/apps that support Conditional Access, but any endpoint that the end-user interacts with.

An example is a device login flow endpoint that is being used in various new PowerShell and Graph tools. Device login flow is a way of allowing login from a device where it is not possible to show a login screen, like on an IOT device. The mechanism is that a device-based login command is executed on the given device and a code is shown to the user. This code is used on another device where the user goes to <https://aka.ms/devicelogin> and specifies the user and password for the user. After login from the other device, the login succeeds on the IOT device in that user context.

The challenge with this login is that it (in nature) does not support device based Conditional Access, which means that no-one can use such tools/commands if we apply a baseline policy for all cloud apps requiring known user and known device. There are other applications that have the same issue with Device Based Conditional Access.

The other architecture, "Targeted architecture", built on the principle that you only target individual apps in CA policies that you want to protect. In this case, endpoints like device-login endpoint are not subjective to the CA policies and hence will continue to work.

The challenge using this architecture is that you may forget to protect all cloud apps. The number of Office 365 and Azure AD apps increase over time as Microsoft or partners release new features or your IT admins integrate various applications with Azure AD.

Access to all such applications will only be protected if you have a mechanism that detects any new app that supports CA and automatically apply a policy to them. Creating and maintaining such a script is a task not to be underestimated. Also notice that it is only supported to have about 250 apps included in one CA policy. In some cases we have found that you can have up to 600 apps in one policy before you may get a technical error about payload being exceeded, but that is not supported.

It often comes as a surprise for many customers that access to portals like security.microsoft.com, portal.office.com and ea.azure.com are not protected in case you use the targeted architecture and don't have "catch-all" rules that enforce MFA on access to all cloud apps. The reason for this is that those portals and many other endpoints are not targetable as a selectable app in a Conditional Access policy.

The same is true for various AAD endpoints that for example graph-based PowerShell modules may be using. So overall you are better protected when choosing the Zero Trust architecture approach based on protecting all cloud apps.

Using the Zero Trust CA architecture will automatically protect any new app, as the "All cloud apps" will target any app, existing or new and is better aligned to a Zero Trust strategy. This is the main reason why this is the suggested architecture.

Also for the issue with some scenarios not working with the Zero Trust architecture approach there are two workarounds that can be used by itself or combined in one potential solution.

Create a separate persona group with policies that allow for access to resources based on either MFA or known/compliant device and limit the members of this persona group as much as possible. Security implications of this need to be discussed and approved by the security department.

Use Microsoft Defender for Cloud Apps, MDCA (previously known as MCAS) together with CA session policies for users with this need and change the base policy for the persons to be send to MDCA for access from unmanaged devices.

MDCA can then be configured with an access policy where access to everything than AAD requires known or compliant device. See details under Developers persona on how this can be used.

SIEM/SoC Alerts

Examples of alerts that can be configured from Azure Sentinel are shown below:

Emergency access accounts

Alert rule name: Emergency access account usage detected

Description: During the last 5 minutes, sign-in log entries for an emergency access account were detected.

SigninLogs

| where UserId == "4a5b13c6-9b00-4235-936f-b95bb060422b"

Conditional access configuration

Alert rule name: Conditional access configuration changed

Description: In the last 5 minutes the conditional access configuration was changed.

AuditLogs

| where ActivityDisplayName == "Add conditional access policy" or ActivityDisplayName == "Update conditional access policy" or ActivityDisplayName == "Delete policy"

Note: Currently there is no dedicated activity for the deletion of conditional access policies, hence the alert is based on the more generic policy object which may cause additional alerts if other policy objects are deleted.

Exclusions

Alert rule name: Accounts added to conditional access exclusion

Description: In the last 5 minutes a significant number of accounts have been added to Conditional Access exclusions.

AuditLogs

| where Category == "GroupManagement" and OperationName == "Add member to group" |
extend Group =
tostring(parse_json(tostring(parse_json(tostring(TargetResources[0].modifiedProperties))[1].newValue))) | where Group startswith "CA_Exclusion"

Note: This alert is based on the display name - an attacker with appropriate privileges could alter the display name to circumvent this alert.

Suggested Policies

Based on latest CA experiences from various customer projects, we suggest to structure CA Policies according to the following areas:

- Global protection (CA001-CA099)
- Admins protection (CA100-CA199)
- Internals user protection (CA200-CA299)
- Externals user protection (CA300-CA399)
- Guests user protection (CA400-CA499)
- GuestAdmins user admins protection (CA500-CA599)
- Microsoft365ServiceAccounts (CA600-CA699)
- AzureServiceAccounts (CA700-CA799)
- CorpServiceAccounts (CA800-CA899)
- WorkloadIdentities (CA900-CA999)
- Developer (CA1000-CA1099)

If you want to document/create the policies suggested, you can use the Excel template attached (for now they are empty for you to fill them in for your own policies). Please find the latest Excel spreadsheet with policies on the GitHub repository.



All conditional
access polices for pers

Global Policies (CA001-CA099)

Global Base Protection policies

We want to limit/restrict access from identities who are not part of any persona group. Some may want to have the grant control being "Require compliant device" or "Require Multifactor Authentication" as opposed to totally block the access. The policy below suggests using block as a catch all policy.

CA001-Global-BaseProtection-AllApps-AnyPlatform-BlockNonPersonas

- Users: Include: All Users
Exclude: CA-BreakglassAccounts, CA-Persona-Admins, CA-Persona-Internals, CA-Persona-Externals, CA-Persona-Guests, CA-Persona-GuestAdmins, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Global-BaseProtection-Exclusions, CA-Persona-Developers, CA-Persona-WorkloadIdentities
- Cloud Apps: All Cloud Apps

- Platform: Any Platform
- Grant: Block

NOTICE! This policy is very strict and has some implications. First remember to update memberships of all personas, especially CA-Persona-Admins as global admins will be blocked by this policy if they are not part of a persona group.

Also it should be noted that some scenarios may work different than expected with this policy in place. A known scenario is when sharing protected content with external who do not have a B2B/Guest account to get access to the protected material and be able to decrypt the content based on guest access CA policies. If you need to be able to share protected content based on Unified Labeling/AIP with externals and you don't want to require the externals to have a B2B Guest account, you can consider excluding "Azure Information Protection" from this policy. See more here:

[Manage sensitivity labels in Office apps - Microsoft 365 Compliance | Microsoft Docs](#)

[FAQs for Azure Information Protection \(AIP\) | Microsoft Docs](#)

Global Attack Surface Reduction policies

If you don't have any apps that you want to block access to for all personas, just leave out the policy below which is just an example policy.

CA002-Global-AttackSurfaceReduction-VariousApps-AnyPlatform-Block

- Users: All Users
Exclude: CA-BreakGlassAccounts, CA-Persona-Global-BaseProtection-Exclusions
- Cloud Apps: <cloud app to be blocked globally if any>
- Platform: Any Platform
- Grant: Block

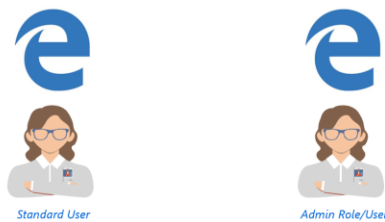
Another AttackSurfaceReduction global policy to consider would be to block any request from any user from a specific country.

Admins Policies (CA100-CA199)

For all administrative roles assigned we require known device as well as known user. And additionally, MFA for admins. This is to mitigate against various attacks for admins and mitigate use-cases where end-users are also using the admin accounts from the same workstation/device as their end-user. See figure below for some common use cases.

MFA as grant control

- One account used for multiple personas (Internals + Admins)
- Two accounts used on same PC
- Extra MFA for Admin access, enforced in Persona or through PIM
- MFA consistency – consider using Sign-In frequency on guests/admins



Microsoft guidance is not to use admin accounts on a standard PC, but rather using a Privileged Access Workstation (PAW), but in practice we see many customers still using two identities on one PC. Device based conditional access works fine when setup correctly for the primary user of the PC, whereas a second browser session running as admin can't prove device compliance for the admin unless special setup is used. There are a few options

- 1) Configure a separate Edge Profile and sign-in in this profile with the admin account and choose to add the associated work account (see details here: [Require Device Compliance for the non-primary user – 365 by Thijs](#)). This will not work for federated users on Azure AD Hybrid joined PCs. The reason for this is that you can't obtain two Kerberos tickets for different identities from same PC. For this scenario you can choose to migrate them to using PTA/sSSO or rather PHS or you can use a trick, where you manually point your PC to the internal address of the WAP server while registering the Work Account through the Edge sign process. This will use NTLM pass-through as opposed to Kerberos and still be able to mark the device as managed in Azure AD.
- 2) Start a separate cmd session and run: `runas /profile /u:<adminuser> chrome.exe`, which will make it possible to run as admin in this instance of Chrome in parallel to primary user using standard apps and edge i.e. Or as an alternative use Chrome with different profiles just as Edge. (given you have the Windows 10 Account extension installed).

Until lately it has only been Edge and Chrome that have been supporting SSO and CA device authentication with Azure AD, but now Firefox also joins the family of browsers supporting this as shown below where Firefox settings now allow the integration without any extra extension. So this may also be an option for you.



Service accounts with admin privileged should be avoided, but if/when needed they will be excluded from admins Conditional Access following special process for approving this. The separate persona for service accounts will enforce other CA policies for these accounts.

Alternatives to using service accounts like using built-in Azure managed identities or service principal with certificates should be used as opposed to using service accounts wherever possible.

Admins Base Protection policies

CA100-Admins-BaseProtection-AllApps-AnyPlatform-MFAANDCompliant

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts , CA-Persona-Admins-BaseProtection-Exclusions
- Cloud Apps: All Cloud Apps
Exclude: Microsoft Intune Enrollment, CMG-ServerApp
- Platform: Any Platform
- Grant: Require multifactor authentication AND Compliant

Admins Identity Protection policies

CA101-Admins-IdentityProtection-AllApps-AnyPlatform-CombinedRegistration

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-

AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Admins-IdentityProtection-Exclusions

- Cloud Apps/User Actions: Register security information
- Location: Include: Any
Exclude: Citrix Trusted IPs (if relevant for customer)
- Platform: Any Platform
- Grant: Require Compliant or AADHJ

CA102-Admins-IdentityProtection-AllApps-AnyPlatform-MFAandPWDforMediumandHighUserRisk

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Admins-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: User risk: Medium, High
- Grant: Require Multifactor Authentication and Password Change
- Session: Sign-in frequency: Every time

For Password-less authentication for admins, you may want to leave out "Password Change" requirement.

CA103-Admins-IdentityProtection-AllApps-AnyPlatform-MFAforMediumandHighSignInRisk

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts , CA-Persona-Admins-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: Sign-In Risk: Medium, High
- Grant: Require Multifactor Authentication
- Session: Sign-in frequency: Every time

CA104-Admins-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Admins-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

Admins Data and App protection

CA105-Admins-AppProtection-MicrosoftIntuneEnrollment-AnyPlatform-MFA

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Admins-AppProtection-Exclusions
- Cloud Apps: Microsoft Intune Enrollment
- Platform: Any
- Grant: Require multi-factor authentication
- Session: Sign-in frequency: Every time

CA106-Admins-DataandAppProtection-AllApps-iOSorAndroid-ClientAppandAPP

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts , CA-Persona-Admins-DataProtection-Exclusions, CA-Persona-Admins-AppProtection-Exclusions
- Cloud Apps:
Include: All Cloud Apps
Exclude: Microsoft Intune Enrollment
- Platform: iOS and Android
- Grant: Require Approved Client App OR Require App Protection Policy

CA107 Admins-DataProtection-AllApps-AnyPlatform-SessionPolicy

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-

AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Admins-DataProtection-Exclusions

- Cloud Apps: All Cloud Apps
- Platform: Any
- Session:
Sign-in frequency: 4 hours
Persistent browser session: Never persistent

Admins Attack Surface Reduction

CA108-Admins-AttackSurfaceReduction-AllApps-AnyPlatform-BlockUnknownPlatforms

- Users: Include: CA-Persona-Admins
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts, CA-Persona-AzureServiceAccounts, CA-Persona-CorpServiceAccounts, CA-Persona-Admins-AttackSurfaceReduction-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Include: Any platform
Exclude: Android, iOS, Windows, macOS
- Grant: Block access

Internals Policies (CA200-CA299)

The Internals are users from AD synced to Azure AD who are employees.

Internals Base Protection

Require known user and Compliant or Azure AD Hybrid Joined device from any device.

CA200-Internals-BaseProtection-AllApps-AnyPlatform-CompliantorAADHJ

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-BaseProtection-Exclusions
- Cloud Apps: Include: All Cloud Apps
Exclude: Microsoft Intune Enrollment, CMG-ServerApp
- Platforms: Any Platform
- Grant: Compliant or AADHJ

Internals Identity Protection

CA201-Internals-IdentityProtection-AllApps-AnyPlatform-CombinedRegistration

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-IdentityProtection-Exclusions
- Platforms: Any Platform
- Cloud Apps/User Actions: Require security information
- Platform: Any Platform
- Grant: Require Compliant or AADHJ

CA202-Internals-IdentityProtection-AllApps-AnyPlatform-MFAandPWDforHighUserRisk

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: User risk: High
- Grant: Require Multifactor Authentication and Password Change
- Session: Sign-in frequency: Every time

CA203-Internals-IdentityProtection-AllApps-AnyPlatform-MFAforHighSignInRisk

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: Sign-in risk: High
- Grant: Require Multifactor Authentication
- Session: Sign-in frequency: Every time

CA204-Internals-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-IdentityProtection-Exclusions

- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

Internals App and Data Protection

CA205-Internals-AppProtection-MicrosoftIntuneEnrollment-AnyPlatform-MFA

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-AppProtection-Exclusions
- Cloud Apps: Microsoft Intune Enrollment
- Platform: Any
- Grant: Require multi-factor authentication
- Session: Sign-in frequency: Every time

CA206-Internals-DataandAppProtection-AllApps-iOSorAndroid-ClientAppORAPP

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-DataProtection-Exclusions, CA-Persona-Internals-AppProtection-Exclusions
- Cloud Apps:
Include: Office 365
Exclude: Microsoft Intune Enrollment
- Platform: iOS and Android
- Grant: Require Approved Client App or Require App Protection Policy

Internals Attack Surface Reduction

CA207-Internals-AttackSurfaceReduction-AllApps-AnyPlatform-BlockUnknownPlatforms

- Users: Include: CA-Persona-Internals
Exclude: CA-BreakGlassAccounts, CA-Persona-Internals-AttackSurfaceReduction-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Include: Any platform
Exclude: Android, iOS, Windows, macOS

- Grant: Block access

Externals Policies (CA300-399)

The Externals are users in AD synced to Azure AD who are not employees, like a consultant. (A consultant may (also) be a guest account instead of an AD account).

Externals Base Protection

CA300-Externals-BaseProtection-AllApps-AnyPlatform-CompliantorAADHJ

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-BaseProtection-Exclusions
- Cloud Apps: Include: All Cloud Apps
Exclude: Microsoft Intune Enrollment, CMG-ServerApp
- Platforms: Any Platform
- Location: Include: Any
Exclude: Citrix Trusted IPs (if relevant for the customer)
- Grant: Compliant or AADHJ

Externals Identity Protection

CA301-Externals-IdentityProtection-AllApps-AnyPlatform-CombinedRegistration

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-IdentityProtection-Exclusions
- Platforms: Any Platform
- Cloud Apps/User Actions: Require security information
- Platform: Any Platform
- Location: Include: Any
Exclude: Citrix Trusted IPs (if relevant for the customer)
- Grant: Require Compliant or AADHJ

CA302-Externals-IdentityProtection-AllApps-AnyPlatform-MFAandPWDforHighUserRisk

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps

- Platform: Any Platform
- Condition: User risk: High
- Grant: Require Multifactor Authentication and password change
- Session: Sign-in frequency: Every time

CA303-Externals-IdentityProtection-AllApps-AnyPlatform-MFAforHighSignInRisk

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: Sign-in risk: High
- Grant: Require Multifactor Authentication
- Session: Sign-in frequency: Every time

CA304-Externals-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

Externals App and Data Protection

CA305-Externals-AppProtection-MicrosoftIntuneEnrollment-MFA

- Users: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-AppProtection-Exclusions
- Cloud Apps: Microsoft Intune Enrollment
- Platform: Any
- Grant: Require multi-factor authentication
- Session: Sign-in frequency: Every time

CA306-Externals-DataandAppProtection-AllApps-iOSorAndroid-ClientAppORAPP

- Users: Include: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-DataProtection-Exclusions, CA-Persona-Externals-AppProtection-Exclusions
- Cloud Apps:
Include: Office 365
- Platform: iOS and Android
- Grant: Require Approved Client App or Require App Protection Policy

Externals Attack Surface Reduction

CA307-Externals-AttackSurfaceReduction-AllApps-AnyPlatform-BlockUnknownPlatforms

- Users: Include: Include: CA-Persona-Externals
Exclude: CA-BreakGlassAccounts, CA-Persona-Externals-AttackSurfaceReduction-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Include: Any platform
Exclude: Android, iOS, Windows Phone, Windows, macOS
- Grant: Block access

Guests Policies (CA400-CA499)

Guests Base Protection

CA400-Guests-BaseProtection-AllApps-AnyPlatform-MFA

- Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts , CA-Persona-Guests-BaseProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platforms: Any Platform
- Grant: Require Multifactor Authentication

Guests Identity Protection

CA401-Guests-IdentityProtection-AllApps-AnyPlatform-TOU-CombinedRegistration

- Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts, CA-Persona-Guests-IdentityProtection-Exclusions
- Cloud Apps/User Actions: Require security information

- Platform: Any Platform
- Grant: Require Special TOU ?

CA402-Guests-IdentityProtection-AllApps-AnyPlatform-MFAforMediumandHighSignInRisk

- 1) Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts, CA-Persona-Guests-IdentityProtection-Exclusions
- 2) Cloud Apps: All Cloud Apps
- 3) Platform: Any Platform
- 4) Condition: Sign-in risk: Medium, High
- 5) Grant: Require Multifactor Authentication
- 6) Session: Sign-in frequency: Every time

CA403-Guests-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts , CA-Persona-Guests-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

Guests App and Data Protection

Empty for now as Microsoft does not support App Protection Policies and Approved Client App for guest users. Most customers would want to extend these suggested starting policies for guests with some data protection related policies, for example by using MDCA (MCAS) to hinder data leakage.

App restriction policies should also be considered for access to SharePoint and Teams for guest users as it can provide a restricted session with read-only access for guest users.

Guests Attack Surface Reduction

CA404-Guests-AttackSurfaceReduction-AllApps-AnyPlatform-BlockNonGuestAppAccess

- Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts, CA-Persona-Guests-AttackSurfaceReduction-Exclusions

- Cloud Apps: Include: All Cloud Apps
Exclude: Office 365, MyApps (to allow guest user to access myapps.microsoft.com)
- Grant: Block Access

NB! This policy will not work optimally for guests as there are endpoints that gets blocked that can't be excluded as an individual CA app in this policy. (Microsoft Invitation Acceptance Portal)

[Invitation redemption in B2B collaboration - Azure AD | Microsoft Docs](#)

- 1) Instead of using All Cloud Apps, choose individual apps that you want to block access to
- 2) Don't have this policy, just require MFA for all cloud apps for guests
- 3) Give users direct URLs to go to the app as opposed to using myapplications.microsoft.com or myapps.microsoft.com as landing zone.

Guests Compliance Protection

CA405-Guests-ComplianceProtection-AllApps-AnyPlatform-RequireTOU

- Users: Include: CA-Persona-Guests
Exclude: CA-BreakGlassAccounts, CA-Persona-Guests-ComplianceProtection-Exclusions
- Cloud Apps: Include: All Cloud Apps
Exclude: Microsoft Intune Enrollment
- Platform: Any Platform
- Grant: Require Terms Of Use

GuestAdmins Policies (CA500-CA599)

GuestAdmins Base Protection

Notice that GuestAdmins are excluded from the other set of policies for Admins.

CA500-GuestAdmins-BaseProtection-AllApps-AnyPlatform-MFA

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-BreakGlassAccounts, CA-Persona-GuestAdmins-BaseProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platforms: Any Platform

- Grant: Require Multifactor Authentication

GuestAdmins Identity Protection

CA501-GuestAdmins-IdentityProtection-AllApps-AnyPlatform-CombinedRegistration

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-BreakGlassAccounts, CA-Persona-GuestAdmins-IdentityProtection-Exclusions
- Cloud Apps/User Actions: Require security information
- Platform: Any Platform
- Grant: Require Special TOU ?

CA502-GuestAdmins-IdentityProtection-AllApps-AnyPlatform-MFforMediumandHighSignInRisk

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-BreakGlassAccounts , CA-Persona-GuestAdmins-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: Sign-in risk: Medium, High
- Grant: Require Multifactor Authentication
- Session: Sign-in frequency: Every time

CA503-GuestAdmins-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-BreakGlassAccounts, CA-Persona-GuestAdmins-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

GuestAdmins App and Data Protection

Empty for now as Microsoft does not support App Protection Policies and Approved Client App for guest users. Most customers would want to extend these suggested starting policies

for guests with some data protection related policies, for example by using MDCA (MCAS) to hinder data leakage.

App restriction policies should also be considered for access to SharePoint and Teams for guest users as it can provide a restricted session with read-only access for guest users.

GuestAdmins Attack Surface Reduction

CA504-GuestAdmins-AttackSurfaceReduction-AllApps-AnyPlatform-BlockNonO365andAzureAccess

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-Persona-GuestAdmins-AttackSurfaceReduction-Exclusions
- Cloud Apps: Include: All Cloud Apps
Exclude: Office 365, Microsoft Azure Management, MyApps (to allow guest user to access myapps.microsoft.com)
- Grant: Block Access

NB! This policy will not work optimally for guests as there are endpoints that gets blocked that can't be excluded as an individual CA app in this policy. (Microsoft Invitation Acceptance Portal)

[Invitation redemption in B2B collaboration - Azure AD | Microsoft Docs](#)

- 1) Instead of using All Cloud Apps, choose individual apps that you want to block access to
- 2) Don't have this policy, just require MFA for all cloud apps for guests
- 3) Give users direct URLs to go to the app as opposed to using myapplications.microsoft.com or myapps.microsoft.com as landing zone.

GuestAdmins Compliance Protection

CA505-GuestAdmins-ComplianceProtection-AnyPlatform-RequireTOU

- Users: Include: CA-Persona-GuestAdmins
Exclude: CA-BreakGlassAccounts, CA-Persona-GuestAdmins-Compliance-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Grant: Require Terms Of Use

Microsoft365ServiceAccounts Policies (CA600-CA699)

Microsoft365ServiceAccounts Base Protection

CA600-Microsoft365ServiceAccounts-BaseProtection-AllApps-AnyPlatform-BlockUntrustedLocations

- Users: Include: CA-Persona-Microsoft365ServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps: All Cloud Apps
- Platforms: Any Platform
- Locations:
Include: Any
Exclude: All Trusted Locations (or Azure/O365 location if possible)
- Grant: Block Access

Sometimes, you may not be able to control the location from where these accounts are used, which means that you may have to adjust policies accordingly.

Microsoft365ServiceAccounts Identity Protection

CA601-Microsoft365ServiceAccounts-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Microsoft365ServiceAccounts
Exclude: CA-BreakGlassAccounts, CA-Persona-Microsoft365ServiceAccounts-IdentityProtection-Exclusions
- Cloud Apps:
Include: All Cloud Apps
- Platforms: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block Access

Microsoft365ServiceAccounts Attack Surface Reduction Protection

CA602-Microsoft365ServiceAccounts-AttackSurfaceReduction-O365-AnyPlatform-BlockNonO365

- Users: Include: CA-Persona-Microsoft365ServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps:
Include: All Cloud Apps
Exclude: Office 365
- Platforms: Any Platform
- Locations: Any
Grant: Block Access

NB! This policy will not work optimally for all scenarios as there are endpoints that gets blocked that can't be excluded as an individual CA app in this policy. In that case you may want to block access to specific/individual apps instead of using "All Cloud Apps"

AzureServiceAccounts Policies (CA700-CA799)

AzureServiceAccounts Base Protection

CA700-AzureServiceAccounts-BaseProtection-AllApps-AnyPlatform-BlockUntrustedLocations

- Users: Include: CA-Persona-AzureServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps: All Cloud App
- Platforms: Any Platform
- Locations:
Include: Any
Exclude: All Trusted Locations (or Azure Location if possible)
- Grant: Block Access

AzureServiceAccounts Identity Protection

CA701-AzureServiceAccounts-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-AzureServiceAccounts
Exclude: CA-BreakGlassAccounts, CA-Persona-AzureServiceAccounts-IdentityProtection-Exclusions

- Cloud Apps:
Include: All Cloud Apps
- Platforms: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block Access

AzureServiceAccounts Attack Surface Reduction Protection

CA702-AzureServiceAccounts-AttackSurfaceReduction-AllApps-AnyPlatform-BlockNonAzure

- Users: Include: CA-Persona-AzureServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps:
Include: All Cloud Apps
Exclude: Microsoft Azure Management
- Platforms: Any Platform
- Grant: Block Access

NB! This policy will not work optimally for all scenarios as there are endpoints that gets blocked that can't be excluded as an individual CA app in this policy. In that case you may want to block access to specific/individual apps instead of using "All Cloud Apps"

CorpServiceAccounts Policies (CA800-CA899)

CorpServiceAccounts Base Protection

CA800-CorpServiceAccounts-BaseProtection-AllApps-AnyPlatform-BlockUntrustedLocations

- Users: Include: CA-Persona-CorpServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps: All Cloud App
- Platforms: Any Platform
- Locations:
Include: Any
Exclude: All Trusted Locations (Corp or Azure vNET if possible)

- Grant: Block Access

CA801-CorpServiceAccounts-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-AzureServiceAccounts
Exclude: CA-BreakGlassAccounts, CA-Persona-CorpServiceAccounts-IdentityProtection-Exclusions
- Cloud Apps:
Include: All Cloud Apps
- Platforms: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block Access

CorpServiceAccounts Attack Surface Reduction Protection

CA802-CorpServiceAccounts-AttackSurfaceReduction-AllApps-AnyPlatform-BlockNonO365andAzure

- Users: Include: CA-Persona-CorpServiceAccounts
Exclude: CA-BreakGlassAccounts
- Cloud Apps:
Include: All Cloud Apps
Exclude: Microsoft Azure Management, Office 365
- Platforms: Any Platform
- Grant: Block Access

NB! This policy will not work optimally for all scenarios as there are endpoints that gets blocked that can't be excluded as an individual CA app in this policy. In that case you may want to block access to specific/individual apps instead of using "All Cloud Apps"

WorkloadIdentities Policies (CA900-CA999)

WorkloadIdentities Base Protection

CA900-WorkloadIdentities-BaseProtection-AllApps-AnyPlatform-BlockUntrustedLocations

- Workload identities:
Include: None (Select All owned service principals or individual workload identities)
Exclude:
- Cloud Apps: All Cloud App
- Platforms: Any Platform
- Locations:
Include: Any
Exclude: All Trusted Locations (Corp or Azure vNET if possible)
- Grant: Block Access

Developers Policies (CA1000-CA1099)

The Developers are users from AD synced to Azure AD who need special access to services like Azure DevOps, containers, Linux i.e. where for example OAuth device code flow is needed and being blocked by base policies for Internals or Externals.

Members can be both Internals and Externals, but those members are then moved from these personas over into the Developer Persona.

Developers Base Protection

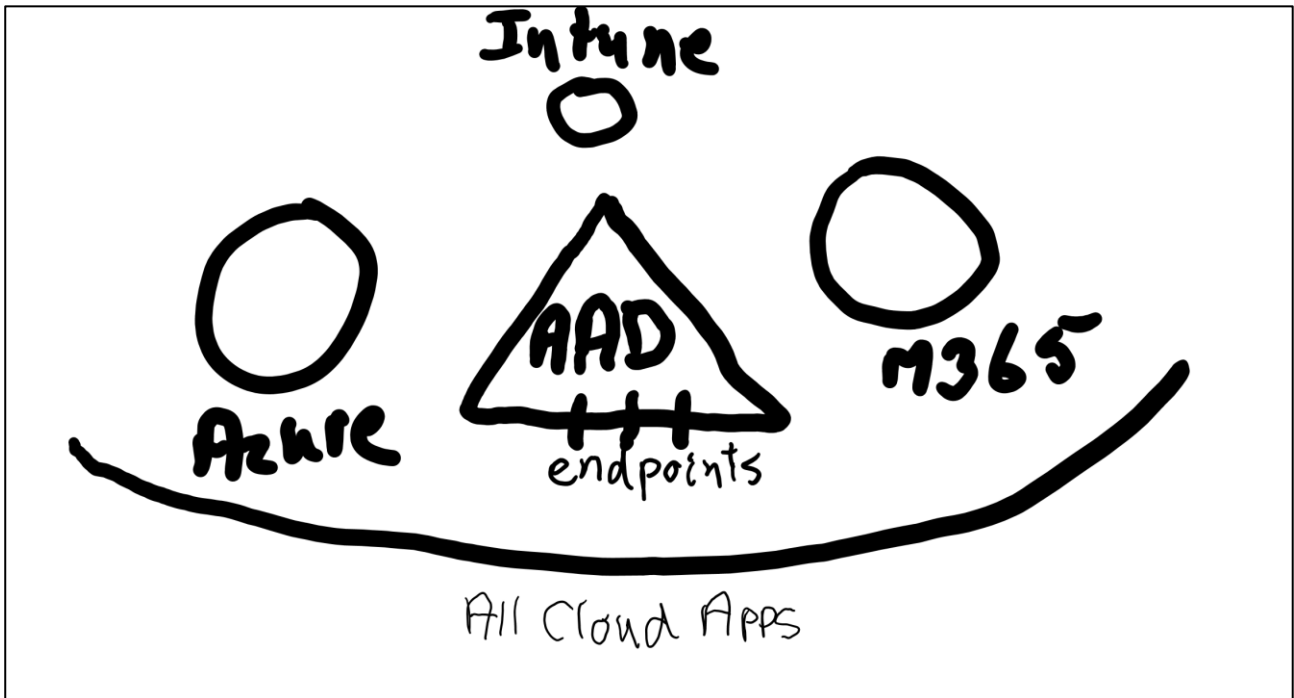
Based on input from customers, they typically have some employees who can't do their work optimally when the CA policies from the Internals are enforced. An example is a developer who needs to manage Azure using "az" from a Linux VM or use a PowerShell cmdlet that only supports device code flow as opposed to standard login based on modern authentication.

Also such people may need access to DevOps CI/CD pipelines from a development tool like Visual Studio Code based on PAT keys (Personal Access Token) and separate policies for access to DevOps overall.

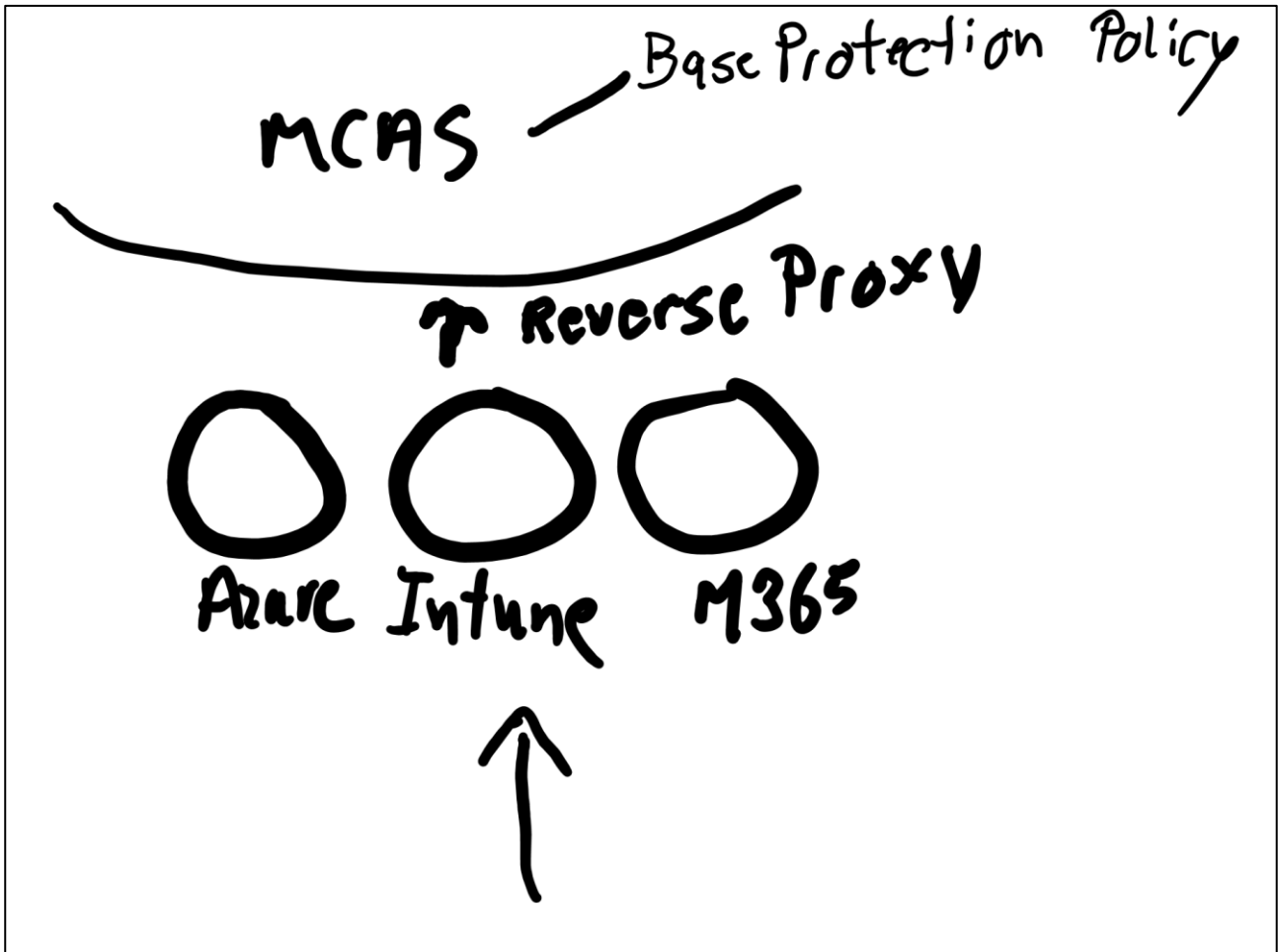
To accommodate for such requirements we suggest this new persona called "Developers" and move people from the Internals or Externals persona group over into this Developers persona group. Also we suggest the base protection for developers to be changed from requiring "Known user" and Compliant or Azure AD Hybrid Joined device from any device to a new policy that sends the requests to MDCA (MCAS) as a session policy.

The reason for using the session app control and MDCA (MCAS) integration is that CA in itself does not allow us to define a policy that can satisfy the requirements why adhering to our Zero Trust principles and be easy to maintain at the same time. Using "All Cloud Apps" together with Complaint or HAADJ grant controls will break functionality for such developers.

The figure below shows how using All Cloud Apps enforces base protection on all apps and endpoints, including some endpoints in AAD that don't support device based Conditional Access.



If we don't enforce any grant controls but rather use App Control as a session control, the protection shields changes as seen in the figure below



The main thing to understand when we do this is that non-interactive access requests and access to the AAD endpoints, like device code flow are not being forwarded by Conditional Access to MDCA (MCAS) and won't be blocked as we don't have any grant control in this policy.

Hence it allows for device code flow to be allowed for while enforcing known user and known/compliant device for access to any other resource, given we have the associated MCAS policy defined correctly.

We do need CA to cover MFA from unmanaged devices in this case as we else would leave an open whole to be able to access endpoints without MFA.

CA1000-Developers-BaseProtection-AllApps-AnyPlatform-ForwardToDefenderforCloudApps

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-BaseProtection-Exclusions
- Cloud Apps: Include: All Cloud Apps
Exclude: Microsoft Intune Enrollment, CMG-ServerApp
- Platforms: Any Platform

- Grant: None
- Session: Use Conditional Access App Control->Use Custom Policy

CA1001-Developers-BaseProtection-AllApps-AnyPlatform-MFAfromUnamagedDevices

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-BaseProtection-Exclusions
- Cloud Apps: Include: All Cloud Apps
- Platforms: Any Platform
- Conditions:
Filter for devices:

trustType Not equals Azure AD joined AND
trustType Not equals Hybrid Azure AD joined
- Grant: Require multi-factor authentication

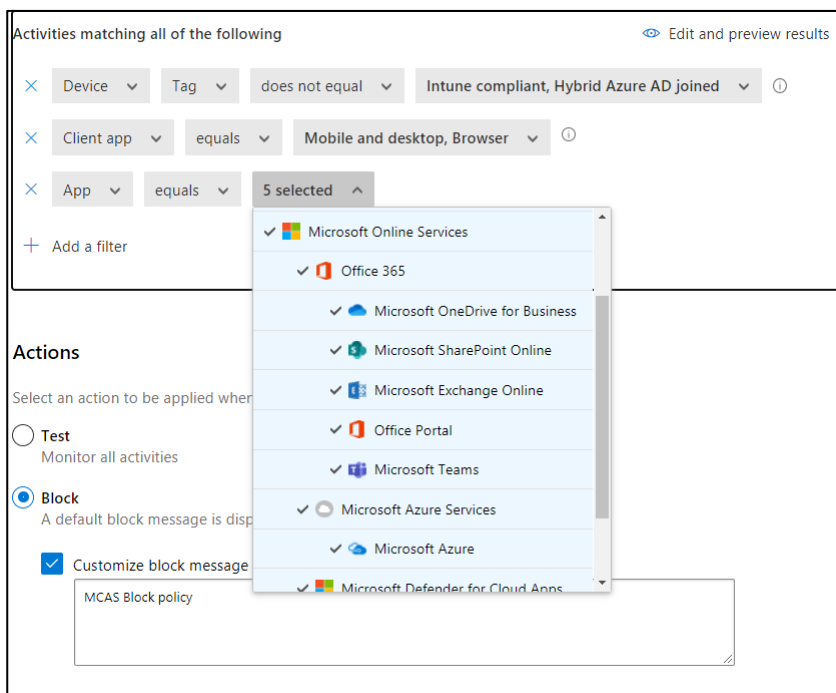
The Defender for Cloud Apps policy is shown below

Policy Name: DefenderforCloudAppsCASessionControl

Category: Access control

Filters:

- User Name from group equals CA-Persona-Developers
- Device Tag does not equal Intune compliant, Hybrid Azure AD Joined
- Actions: Block
- Client app equals Mobile and desktop, Browser
- App equals <select all apps available", see picture below



Customize block message: optional

Selecting "client app" is important, as if you don't do that then only web based access is protected. Also when you specify "Client app", you do have to include at least one app, and as there is no "All Apps", we select what is available. This means that there may be endpoints not protected if they are not covered by these apps. We already know that access to the non-interactive endpoints are not protected, which is what we based this solution on, but other apps would preferably have to be covered.

Developers Identity Protection

CA1002-Developers-IdentityProtection-AllApps-AnyPlatform-CombinedRegistration

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-IdentityProtection-Exclusions
- Platforms: Any Platform
- Cloud Apps/User Actions: Require security information
- Platform: Any Platform
- Grant: Require Compliant or AADHJ

CA1003-Developers-IdentityProtection-AllApps-AnyPlatform-MFAandPWDforHighUserRisk

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-IdentityProtection-Exclusions

- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: User risk: High
- Grant: Require Multifactor Authentication and Password Change
- Session: Sign-in frequency: Every time

CA1004-Developers-IdentityProtection-AllApps-AnyPlatform-MFAforHighSignInRisk

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Condition: Sign-in risk: High
- Grant: Require Multifactor Authentication
- Session: Sign-in frequency: Every time

CA1005-Developers-IdentityProtection-AllApps-AnyPlatform-BlockLegacyAuth

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-IdentityProtection-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Any Platform
- Client Apps: Exchange Active Sync clients, Other Clients
- Grant: Block

Developers App and Data Protection

CA1006-Developers-AppProtection-MicrosoftIntuneEnrollment-AnyPlatform-MFA

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-AppProtection-Exclusions
- Cloud Apps: Microsoft Intune Enrollment
- Platform: Any
- Grant: Require multi-factor authentication

- Session: Sign-in frequency: Every time

CA1007-Developers-DataandAppProtection-AllApps-iOSorAndroid-ClientAppORAPP

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-DataProtection-Exclusions, CA-Persona-Developers-AppProtection-Exclusions
- Cloud Apps:
Include: Office 365
Exclude: Microsoft Intune Enrollment
- Platform: iOS and Android
- Grant: Require Approved Client App or Require App Protection Policy

Developers Attack Surface Reduction

CA1008-Developers-AttackSurfaceReduction-AllApps-AnyPlatform-BlockUnknownPlatforms

- Users: Include: CA-Persona-Developers
Exclude: CA-BreakGlassAccounts, CA-Persona-Developers-AttackSurfaceReduction-Exclusions
- Cloud Apps: All Cloud Apps
- Platform: Include: Any platform
Exclude: Android, iOS, Windows, macOS
- Grant: Block access

Change Management

It is important to consider how to do change management and tests for existing and new CA policies.

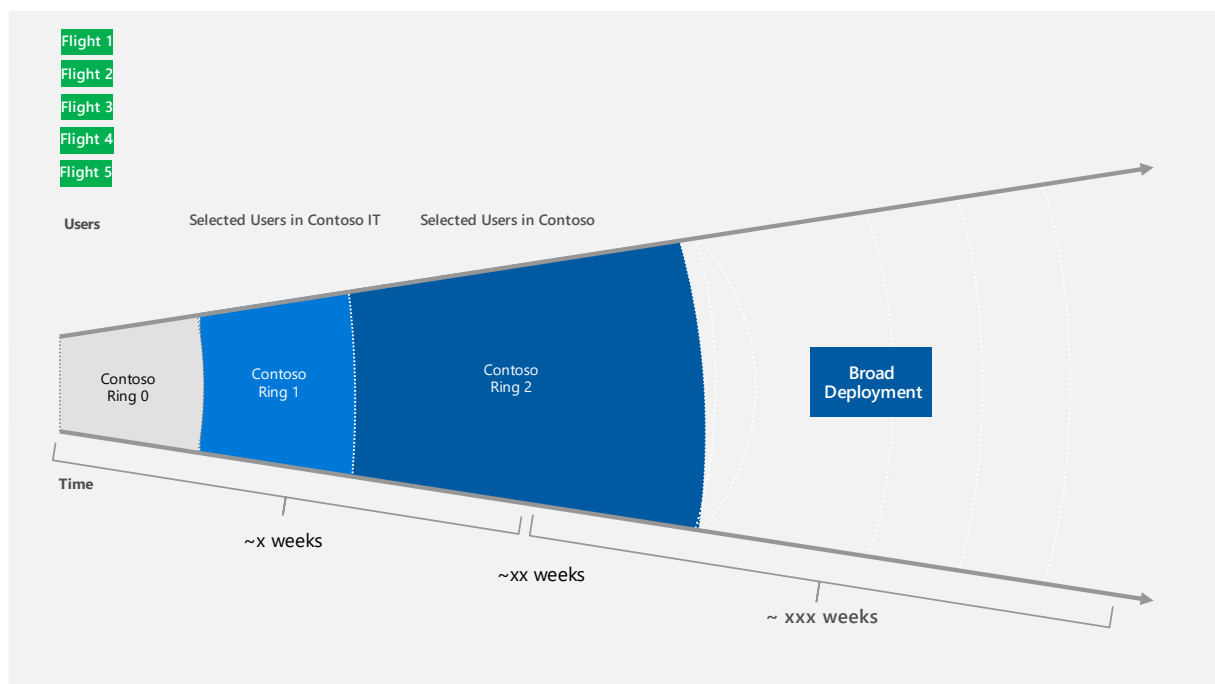
CA Deployment Model - staged deployment and test

When new policies or changes to existing policies are introduced, we want to be able to deploy them in a staged approach meaning that we apply new policies to a limited set of users and test and based on feedback (gate approval) we proceed to deploy the policy to a larger set of users, ending up applying it to all users when all gates have been approved.

Some companies may prefer to manage these test users in an ad-hoc manner and include relevant users in a test group and just expand membership in that group until fully tested and

applied to all users. However you should expect ongoing changes to the CA policies, and we find it useful to have a more structured approach on how to introduce changes. This also involves more easy and structured communication and collaboration with the stakeholders about exact status on various features for individual personas.

A ring-based approach has been used by some customers in the past to manage such changes and testing of new policies in a staged approach and being able to introduce new CA protections as features referred to as flights. Such a model that has been used in the past based on flights and rings is shown below.



A few examples of what a flight could be are

- Support for mobiles
- Support for Windows servers
- Support for remote access to on-premises on Windows

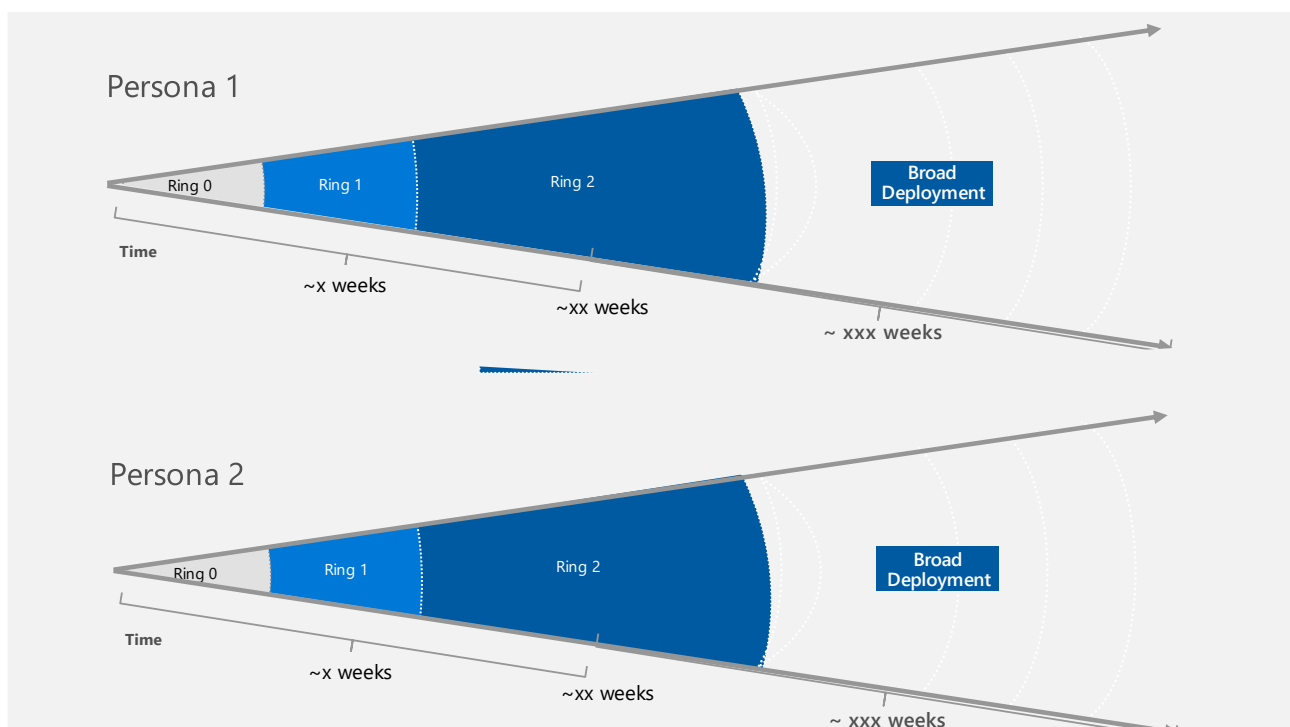
This model has mainly been used when introducing CA for employees with a focus on flights based on device/platform for the employees.

Now that we shift into focusing on personas, we will have most "flights" covered within a persona group and go a bit away from the using the flights model. Still we want to continue the ring-based approach to deploy policies in a controlled way with low risk on affecting production, applying mitigating CA related security controls in a staged rollout.

Before report-only mode was available in Conditional Access, there was a need to create multiple CA policies representing different CA policies for each ring.

This results in many CA policies to be created. Now that we have report-only mode available we propose not to duplicate all the CA policies for each persona for each ring for deployment purposes, rather we suggest managing rings as static Azure AD security groups. This implies to do the staged roll-out of a CA policy by changing assignment of the same policy to gradually include more rings as opposed to do it based on different CA policies having to be created.

The suggested approach is depicted in the figure below where we introduce a given change (feature) in a ring-based approach on a per persona basis. Depending on the number of users in each persona group and the sensitivity to changes for a given persona, you may want to introduce more rings. It is a balance between controlling risk and overhead in manageability and time needed for testing.



The time between each ring may vary, the figure indicates weeks between ring deployments, however you may sometimes only have hours or days, depending on the criticality of the change and complexity involved.

We may now introduce a change to existing policies by use of "feature branches" in Azure DevOps, having a new branch covering each new feature or change. More about this when we discuss automation of CA policies using Azure DevOps.

The table below shows the new ring-based approach for staged roll-out for the Internals persona based on three rings being used.

Ring/AAD security group	Members	Description
-------------------------	---------	-------------

CA-Persona-Internals-Ring0	Only a few 1-5 users who are part of developing CA Policies, like CA Administrators.	The CA administrators are expected to use their own end-user identity to be included in Ring 0, but only for Internals. Admins may need to create separate test accounts for other personas to test out policies for these personas without having to shift rings.
CA-Persona-Internals-Ring1	Some few users in IT who are not part of developing CA policies, 5-10 users	Only assign the new CA policy to this group after Ring 0 deployment and experience have been approved.
CA-Persona-Internals-Ring2	A mix of IT and standard end users	AAD dynamic security group with random criteria's that result in about 1%
CA-Persona-Internals-Ring3	A mix of IT and standard end users	AAD dynamic security group with random criteria's that result in about 5-10%
All Internals CA-Persona-Internals	All members of CA-Persona-Internals	The standard production group representing this persona.

We assume that all CA policies are running in an enabled state for all personas.
The process of introducing a new CA policy is the following

- Create a new CA Policy within the persona group that it is meant for (follow CA naming standard for the persona). Follow the naming standard with the type of policy being used i.e. Put it into reporting-only mode to not affect production before tested. If you wish to minimize risk of end-user implications when using report-only mode, you can choose to apply the policy in report-only mode to the CA-Persona-Internals-Ring0 first, then adding Ring1, Ring2 and Ring3 and finally set it to CA-Persona-Internals.
- Let the policy run for a day or two, - based on potential failures in CA workbooks and sign-in logs assuring they are understood before proceeding and verify if end-users have complained about new login prompts i.e. especially on mobile devices as report-only can result in unexpected prompts in a few use-cases.
- Potentially adjust policy and continue running it in report-only mode for a few extra days and verify that issues have been solved or fully understood before enabling the policy for the first ring.
- Assign the policy to CA-Persona-Internals-Ring0 and enable it

- Test and verify that everything is working as expected over a few days
- Additionally assign the policy to CA-Persona-Internals-Ring1 (so that both Ring0 and Ring1 groups are assigned)
- Test and verify that everything is working as expected over a few days
- Additionally assign the policy to CA-Persona-Internals-Ring2 (so that both Ring0, Ring1 and Ring2 groups are assigned)
- Test and verify that everything is working as expected over a few days
- Additionally assign the policy to CA-Persona-Internals-Ring3 (so that both Ring0, Ring1 and Ring2 groups are assigned)
- Test and verify that everything is working as expected over a few days
- Assign the policy to CA-Persona-Internals. The new policy is now running in full production.

The time or number of days you test and verify may vary depending on how urgent the new policy is needed for the business and weight risks of breaking stuff not fully tested up against how urgent the business needs it, or security demands it to be changed quickly, maybe part of incident response.

If you need to change an existing CA policy as opposed to creating a new, the suggested deployment process is slightly different. We want to be sure that we don't lower the security controls while testing new policies, so we don't just want to disable an existing policy. The suggested procedure for changing existing policies is shown below

- Keep the existing/original policy running while testing the change in a duplicated policy in report-only mode as described below.
- Duplicate existing/original policy and rename it to the existing name with "staged-deployment" or "test" being added – try to be consistent about which wording is added to existing policy to indicate that it is a staged rollout. Duplicate is not yet supported in the portal in which case you need to create a new policy with same content as the existing one and following the naming. For changes done via CI/CD it should be easy to duplicate the CA Policy in a policy configuration file, like json file.
- Assign the duplicated policy to CA-Persona-Internals-Ring0 and put it in report-only mode as we want to test what will happen if we apply this policy.
- Verify workbooks and sign-in logs and follow up on any failures for the new test policy and adjust accordingly
- Add CA-Persona-Internals-Ring1, CA-Persona-Internals-Ring2, and CA-Persona-Internals-Ring3 subsequently while verifying workbooks and sign-in logs and potentially end-users complaints.
- If everything looks ok, we now want to enable/enforce it in production. Start by excluding CA-Persona-Internals-Ring0 from the original policy
- Change the duplicated policy to target only CA-Persona-Internals-Ring0, change it from report-only to enable.
- Verify and follow up on any failures for the new test policy and adjust accordingly
- Change original policy to also exclude CA-Persona-Internals-Ring1 in the assignment of users/groups

- Assign CA-Persona-Internal-Ring1 as additional include to the test policy to CA-Persona-Internals
- Verify and follow up on any failures for the new test policy and adjust accordingly
- Change original policy to also exclude CA-Persona-Internals-Ring2 in the assignment of users/groups
- Assign CA-Persona-Internal-Ring2 as additional include to the test policy to CA-Persona-Internals
- Verify and follow up on any failures for the new test policy and adjust accordingly
- Change original policy to also exclude CA-Persona-Internals-Ring3 in the assignment of users/groups
- Assign CA-Persona-Internal-Ring3 as additional include to the test policy to CA-Persona-Internals
- Verify and follow up on any failures for the new test policy and adjust accordingly
- Assign the test policy to CA-Persona-Internals
- Change original policy from enabled to off
- Verify and follow up on any failures for the new test policy and adjust accordingly
- If everything is working correctly, delete the original policy
- Change name of the test policy to not include -test in the name, so that it now has the same name as the original policy but running with the new changes.

The above model makes it possible to introduce new policies as well as change existing in a controlled manner without introducing lots of new CA policies. The security groups in AAD for the rings will be created once and used ongoingly. Users can choose to volunteer into a ring, just like today at Microsoft where users can choose to be part of Insider rings, like Insider-Slow or Insider-Fast. A user should only be part of one Ring at any point in time. Relevant IdM processes need to be created for request/approval workflows for membership and changes to a ring. Also the need for extra collaboration to end-users and stakeholders should not be overlooked.

We have seen some customers who have multiple MDM systems and maybe also have multiple authentication mechanisms being used for their employees, like ADFS with CBA, PTA/sSSO/PHS. In this case, you may want to create separate personas reflecting these needs as they may need different CA policies to work. A user considered under Internals may be using AirWatch and ADFS/CBA as opposed to another user in Internals using MEM/Intune on a cloud PC with WHfB.

Overall for these Internals, the combined set of policies are now a set of CA and ADFS rules, so it may be easier to manage these scenarios by having Internals-MEM as well as Internals-AirWatch separated out in two Internals related persona groups and have rings-based approach within these persona groups.

This is far from optimal as we want to have as few personas (and CA policies) as possible, so strive at consolidating federations to Azure AD and make any 3party MDM systems report

compliance to MEM, so that we can cover access in the same consistent way independent on how they are managed and authenticate. This is possible for AirWatch and JAMF today as examples.

The above deployment model is based on manually managing the policies directly in the portal. Going forward, it is recommended to automate policies and as part of that introduce formal change and version control based on Azure CI/CD.

Using Microsoft365DSC for this is an option to consider. Also as part of this there should be an approval workflow and the ability to quickly roll back in case of unexpected errors that disrupt the business.

CA Security Groups

We are using various security groups as part of the suggested persona-based CA framework. It is important to have well defined (automated) processes for populating the groups with the right members. The considerations on governance for group types and creation and keeping them up to date are a bit different depending on where the groups are being used.

The table below shows some recommendations on how to govern the groups.

Group	Type	How to populate group members
CA-BreakGlassAccount		
CA-BreakGlassAccount	AAD security group	Manually enter two or three accounts in this group. Ref: Manage emergency access admin accounts - Azure AD Microsoft Docs
CA-Persona-Internals		
CA-Persona-Internals	AAD security group as we can then include both on-premises as well as cloud only based identities.	Use AAD dynamic security Group based on attribute that only employees have (like manager or specific characters in UPN).
CA-Persona-Internals-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month

CA-Persona-Internals-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Internals-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Internals-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Internals-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Internals-Ring0	AAD security group	Manually managed by Conditional Access administrators as there are only a few users in this ring.
CA-Persona-Internals-Ring1	AAD security group	Manually managed by Conditional Access administrators as there are only a few users in this ring.
CA-Persona-Internals-Ring2	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 1%

CA-Persona-Internals-Ring3	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 5-10%
CA-Persona-Developers		
CA-Persona-Developers	AAD security group as we can then include both on-premises as well as cloud only based identities.	Use AAD dynamic security Group based on attribute that only developers have (like manager or specific characters in UPN).
CA-Persona-Developers-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Developers-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Developers-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Developers-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month

CA-Persona-Developers-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Developers-Ring0	AAD security group	Manually managed by Conditional Access administrators as there are only a few users in this ring.
CA-Persona-Developers-Ring1	AAD security group	Manually managed by Conditional Access administrators as there are only a few users in this ring.
CA-Persona-Developers-Ring2	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 1%
CA-Persona-Developers-Ring3	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 5-10%
CA-Persona-Externals		
CA-Persona-Externals	AAD security group as we can then include both on-premises as well as cloud only based identities.	Use AAD dynamic security Group based on attribute that only Externals have (like specific characters in UPN)
CA-Persona-Externals - BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Externals-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month

CA-Persona-Externals-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Externals-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Externals-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Externals-Ring0	AAD security group	Manually managed by Conditional Access administrators as we only have a few users in this ring.
CA-Persona-Externals-Ring1	AAD security group	Manually managed by Conditional Access administrators as we only have a few users in this ring.
CA-Persona-Externals-Ring2	AAD security group	Maybe use an existing group representing internal IT or something like that. It could be an AD on-premises group that is synced to AAD as member of this group
CA-Persona-Externals-Ring3	AAD security group	Maybe use an existing group representing people from various parts of the organization. It could be existing AD on-premises

		groups that are synced to AAD as members of this group. For example. Finance and HR.
Guests		
CA-Persona-Guests	Built-in or AAD Dynamic Security Groups	You may want to use built-in "All guest and external users" as opposed to a dynamic security group as you may want to have an invited user being able to access resources quicker than the time population of a dynamic group membership takes, which is typical about 30 minutes but can be as high as 24 hours (SLA)
CA-Persona-Guests-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM

		Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-Compliance-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Guests-Ring0	AAD security group	Manually managed as there are only few guest users in ring 0
CA-Persona-Guests-Ring1	AAD security group	Manually managed as there are only few guest users in ring 1
CA-Persona-Guests-Ring2	AAD dynamic security group	Dynamic security group with Randomly about 1 %
CA-Persona-Guests-Ring3	AAD dynamic security group	Randomly about 5-10% of existing guest users
GuestAdmins		
CA-Persona-GuestAdmins	AAD security group	
CA-Persona-GuestAdmins-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month

CA-Persona-GuestAdmins-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-GuestAdmins-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-GuestAdmins-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-GuestAdmins-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-GuestAdmins-Compliance-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-GuestAdmins-Ring0	AAD security group	Manually managed as there are only few guest users in ring 0

CA-Persona-GuestAdmins-Ring1	AAD security group	Manually managed as there are only few guest users in ring 1
CA-Persona-GuestAdmins-Ring2	AAD dynamic security group	Dynamic security group with Randomly about 1 %
CA-Persona-GuestAdmins-Ring3	AAD dynamic security group	Randomly about 5-10% of existing guest users
CA-Persona-Microsoft365ServiceAccounts		
CA-Persona-Microsoft365ServiceAccounts	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Microsoft365ServiceAccounts-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Microsoft365ServiceAccounts-AttackServiceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Microsoft365ServiceAccounts - Ring0	AAD security group	Manually managed
CA-Persona-Microsoft365ServiceAccounts - Ring1	AAD dynamic security group	AAD dynamic group with random criteria's that result in about 5-10% (given there are many such accounts, else manage it manually as a static AAD security group)
CA-Persona-AzureServiceAccounts		

CA-Persona-AzureServiceAccounts	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-AzureServiceAccounts-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-AzureServiceAccounts-AttackServiceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-AzureServiceAccounts-Ring0	AAD security group	Manually managed
CA-Persona-AzureServiceAccounts-Ring1	AAD dynamic security group	AAD dynamic group with random criteria's that result in about 5-10% (given there are many such accounts, else manage it manually as a static AAD security group)
CA-Persona-CorpServiceAccounts		
CA-Persona-CorpServiceAccounts	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-CorpAccounts-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM

		Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-CorpServiceAccounts-AttackServiceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-CorpServiceAccounts-Ring0	AAD security group	Manually managed
CA-Persona-CorpServiceAccounts-Ring1	AAD dynamic security group	AAD dynamic group with random criteria's that result in about 5-10% (given there are many such accounts, else manage it manually as a static AAD security group)
CA-Persona-Admins		
CA-Persona-Admins	Preference is AAD security group. Only use AD based security group synced to AAD if your IDM tools are not integrated with AAD	Create an Azure FunctionApp that ongoingly add all Azure AD built-in admin roles as well as any other user that has an admin role in other workloads, like MEM, MCAS, EXO Compliance, MDE i.e.
CA-Persona-Admins-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Admins-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM

		Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Admins-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Admins-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Admins-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-Admins-Ring0	AAD Security Group	Managed manually
CA-Persona-Admins-Ring1	AAD dynamic security group	AAD dynamic group with random criteria's that result in about 5-10% (given there are many such accounts, else manage it manually as a static AAD security group)
CA-Persona-Admins-Ring2	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 1%
CA-Persona-Admins-Ring3	AAD dynamic security group	AAD dynamic security group with random criteria's that result in about 5-10%
CA-Persona-WorkloadIdentities		
CA-Persona-WorkloadIdentities	Preference is AAD security group. Only use AD based	Request access: AAD Identity Governance or 3party IDM


	security group synced to AAD if your IDM tools are not integrated with AAD	Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-BaseProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-IdentityProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-DataProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-AppProtection-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-AttackSurfaceReduction-Exclusions	AAD security group	Request access: AAD Identity Governance or 3party IDM Approval: AAD Identity Governance approvals or 3party IDM

		Access Reviews: AAD Identity Governance or 3party IDM: Each month
CA-Persona-WorkloadIdentities-Ring0	AAD Security Group	Managed manually
CA-Persona-WorkloadIdentities-Ring1	AAD dynamic security group	AAD dynamic group with random criteria's that result in about 5-10% (given there are many such accounts, else manage it manually as a static AAD security group)

Report-Only mode testing considerations

Microsoft suggests testing new CA policies using Report-Only mode. See more here: [What is Conditional Access report-only mode? - Azure Active Directory | Microsoft Docs](#)

Some customers have raised a concern that even so the intention with using Report-Only mode is to test implications of CA policies without affecting users, there is a notice from Microsoft that mentions that on some devices for some scenarios, the end-users may be prompted to select a certificate which is intrusive.

 **Warning**

Policies in report-only mode that require compliant devices may prompt users on Mac, iOS, and Android to select a device certificate during policy evaluation, even though device compliance is not enforced. These prompts may repeat until the device is made compliant. To prevent end users from receiving prompts during sign-in, exclude device platforms Mac, iOS and Android from report-only policies that perform device compliance checks. Note that report-only mode is not applicable for Conditional Access policies with "User Actions" scope.

The challenge related to following the recommendation about excluding iOS, Android and Mac from the report-only CA policies where we require compliant or hybrid-joined device is that we don't get full insight into all implications for users for all scenarios on all platforms.

We have also seen that if a customer uses report-only mode for mobiles, like iOS for App Protection and approved client app CA policies, the report-only may report failure, even so it will work as expected when you switch it to enable. So overall using report-only mode is a very nice feature but be prepared that it does not always show the correct result when you enable the policy. Hence real tests to a limited set of users are an important step when using report-only mode as part of the deployment mode.

Single Sign On is typically obtained by using one of the following components: - Browser/web components (Embedded web view or System Web View) - Token Cache - Brokers (like Company Portal on Android and MS Authenticator App on iOS or Enterprise SSO Extension)

The proof of known device that satisfy device based Conditional Access policies is typically done by using one of the following three mechanisms.

- PRT (Primary Refresh Token) (Supported by apps using ADAL/MSAL libraries)
- PKeyAuth (Microsoft http extension to prove possession of device private key, only supported by some 1party Microsoft applications)
- Client-TLS (std. SSL/TLS protocols)

Applications that are using Microsoft ADAL or MSAL libraries to do authentications (such as all Microsoft first party applications and an increasing number of other applications on the market) will typically be able to check/prove device using PKeyAuth or PRT silently, whereas a fallback to doing the check using Client-TLS typically will result in an intrusive certificate prompt depending on application and OS version/device.

A pre-req. of being able to get and use PRT between multiple applications on a device is to have a broker like Authenticator App (iOS) or Company Portal (Android) and the device is registered in Azure AD. Currently no broker exists for Mac which means that certificate prompts should be expected more often for Mac devices.

The table below provides information about when we can expect the certificate prompt on end-users devices when using report-only mode for policies with device/compliance check. The information below has not been verified by Microsoft officially, so it is work in progress as Microsoft currently does not have such public information.

The essence of the table is that very few prompts (if any) are expected using report-only on devices that are already registered in Azure AD or managed by EM/Intune or having a broker like Authenticator App, Company Portal or and SSO extension from Microsoft (for iOS and MAC) or JAMF (for Mac).

Platform	Application	Components installed	Expected End-User Experience
Android	Application using ADAL/MSAL(Like MS Office) or Browser	Authenticator app and managed by EM/Intune	No Prompt
Android	Application using ADAL/MSAL(Like MS Office)	Company Portal or Authenticator App	No prompt

Android	Application using ADAL/MSAL(Like MS Office)	Company Portal	No Prompt
Android	Application using ADAL/MSAL(Like MS Office)	None	May prompt
Android	Edge web browser	Any	No Prompt
Android	App using system web browser	Company Portal with browser access enabled	No prompt
Android	App using system web browser	No Company Portal or Company portal without browser access enabled	May prompt
Android	App using embedded Web browser	Any	Expect prompt
iOS	Application using ADAL/MSAL(Like MS Office) or Browser	Authenticator app and managed by EM/Intune	No Prompt
iOS	Application using ADAL/MSAL(Like MS Office)	Authenticator App	No prompt
iOS	Application using ADAL/MSAL(Like MS Office)	Authenticator App, Intune/EM Enrolled	No Prompt

iOS	Application using ADAL/MSAL(Like MS Office)	None	May prompt
iOS	Edge web browser	Any	No Prompt
iOS	App using system web browser (Safari)	Any	Expect prompt
Mac	Apps using ADAL/MSAL or Browser	Managed by EM/Intune using Company Portal	No Prompt
Mac	Apps using ADAL/MSAL or Browser	JAMF Connect Login/AAD integration	No prompt expected?
Mac	Apps using ADAL/MSAL or Browser	Microsoft AAD SSO Extension for app	No prompt expected?
Mac	Apps using ADAL/MSAL (Like MS Office)	With or without Company Portal	Expect prompt
Mac	Browser (Safari or Chrome)	With or without Company Portal	Expect Prompt
Mac	Browser (Microsoft Edge)	With or without Company Portal	??

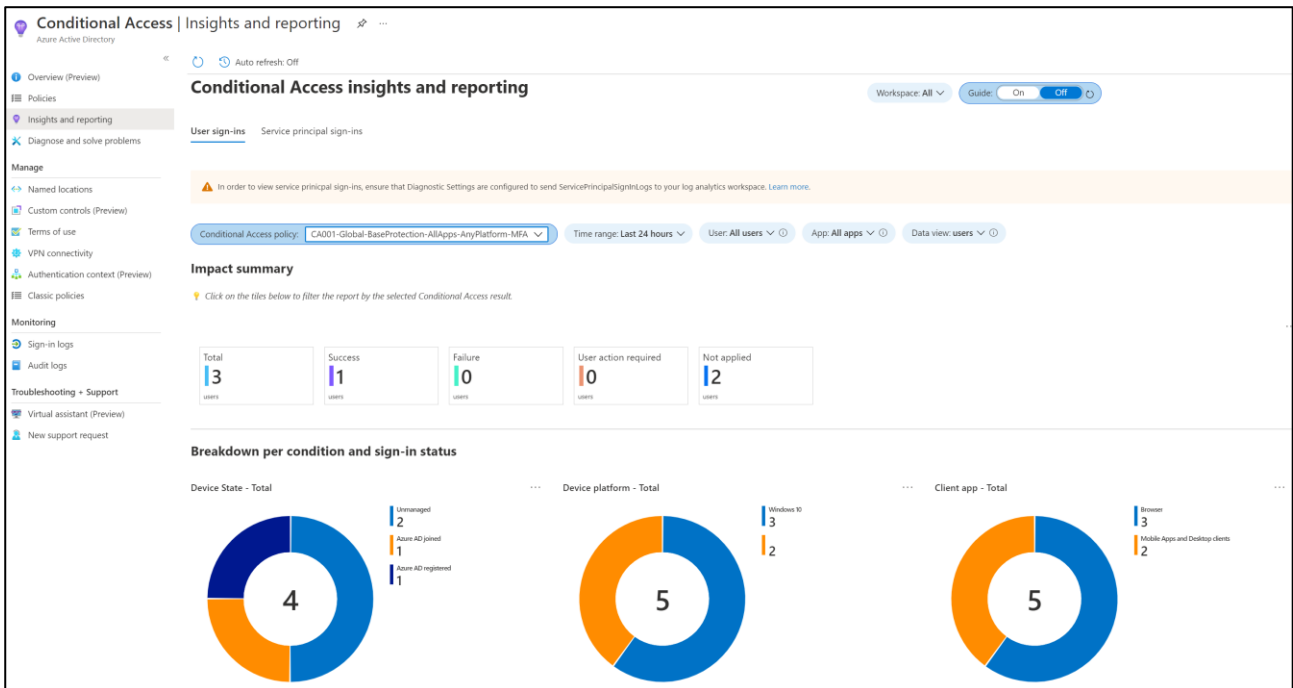
CA Workbooks and insights

In the Azure AD portal under Conditional Access there are some great options available

- Conditional Access Overview blade
- Conditional Access Insights workbook

Conditional Access Insights and reporting workbook

The insights and reporting workbook can be enabled by enabling AAD sign-in logs to be sent to an Azure AD Log Analytics workspace. It provides great overview of the impact of CA policies being applied, either in report-only mode or fully enabled. – see below.



Many customers have found this useful. However it also sometimes seems to indicate that a given CA policy in report-only mode shows more failures than what would really be a problem. One reason for this is that the KQL queries evaluate the CA status for all users and not only for the group of users it is assigned to. This means that you may see a failure related to a guest user that a policy CA200 targeted for Internals. This is not an issue per se but does make it more time consuming using the CA workbook with the ambition to have zero failures for a report-only mode policy before enabling it. Typically you would look at all errors in the workbook and switch to the actual AAD Sign-In log to evaluate the details to see if this is just noise related to another user or a real issue for a user in the given Persona group that this policy is applied to.

You can choose to take a copy of the workbook and adjust it to your needs, or you can choose to make your own workbook with your preferred KQL queries. An example of a useful KQL query that shows failures for a CA200 policy meant for Internals is shown below

SigninLogs

```
extend OS = DeviceDetail.operatingSystem
extend devID = DeviceDetail.deviceId
extend trustType = DeviceDetail.trustType
extend isManaged = DeviceDetail.isManaged
extend isCompliant = DeviceDetail.isCompliant
where OS has "Windows"
where UserType has "Member"
mvexpand ConditionalAccessPolicies
where ConditionalAccessPolicies["result"] has "reportOnlyFailure"
```

```
| where ConditionalAccessPolicies["displayName"] contains "CA200"
| project TimeGenerated, Identity, UserPrincipalName, trustType, AzureADApplication
= AppDisplayName, ClientApplication = ClientAppUsed, ClientBrowser =
DeviceDetail.browser, ClientOperatingSystem = DeviceDetail.operatingSystem,
ClientIPAddress = IPAddress , ClientUserAgent = UserAgent ,
ConditionalAccessPolicyName = ConditionalAccessPolicies["displayName"],
ConditionalAccessPolicyID = ConditionalAccessPolicies["id"],
isManaged,ConditionalAccessStatus, ConditionalAccessPolicies["result"]
```

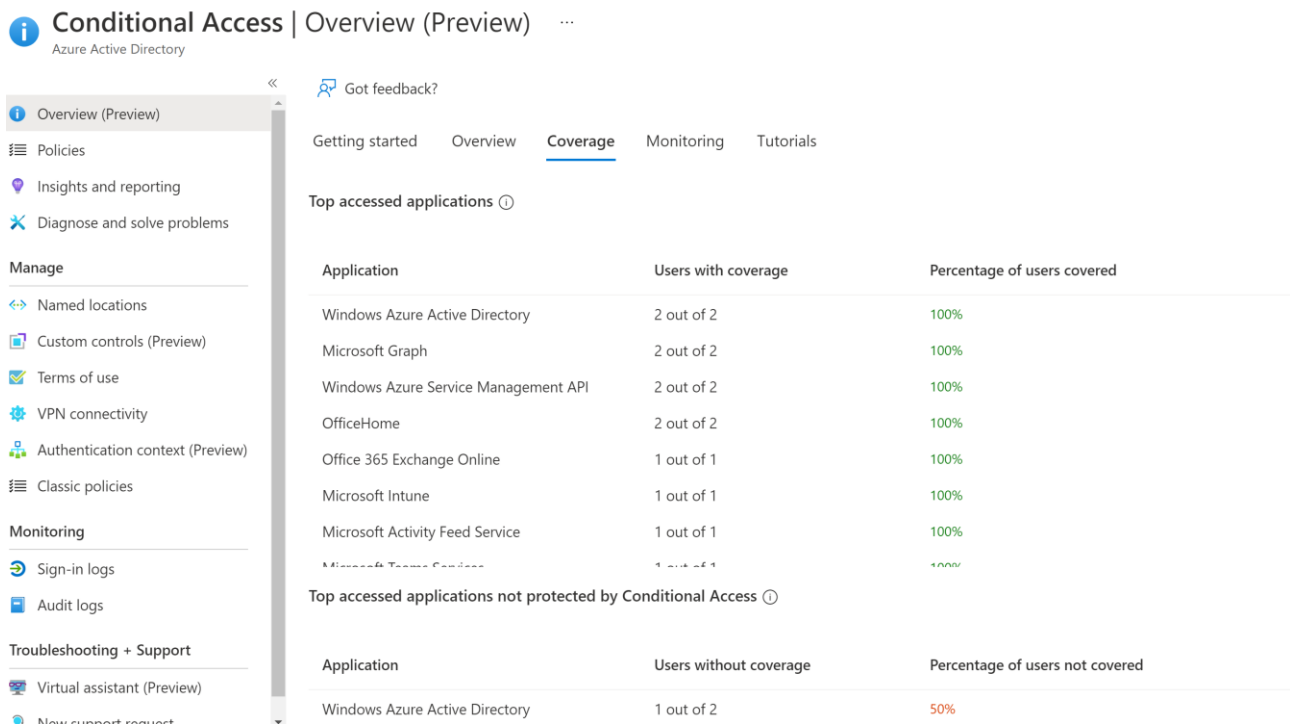
This KQL query lists report-only failures related to the CA200 base policy for member users from a Windows OS.

Or if you want to see all CA failures related to CA200, you can use the following (notice the report-only condition is commented out)

```
SigninLogs
| extend OS = DeviceDetail.operatingSystem
| extend devID = DeviceDetail.deviceId
| extend trustType = DeviceDetail.trustType
| extend isManaged = DeviceDetail.isManaged
| extend isCompliant = DeviceDetail.isCompliant
| where OS has "Windows"
| where UserType has "Member"
| mvexpand ConditionalAccessPolicies
//| where ConditionalAccessPolicies["result"] has "reportOnlyFailure"
| where ConditionalAccessStatus has "failure"
| where ConditionalAccessPolicies["displayName"] contains "CA200"
| project TimeGenerated, Identity, UserPrincipalName, trustType, AzureADApplication
= AppDisplayName, ClientApplication = ClientAppUsed, ClientBrowser =
DeviceDetail.browser, ClientOperatingSystem = DeviceDetail.operatingSystem,
ClientIPAddress = IPAddress , ClientUserAgent = UserAgent ,
ConditionalAccessPolicyName = ConditionalAccessPolicies["displayName"],
ConditionalAccessPolicyID = ConditionalAccessPolicies["id"],
isManaged,ConditionalAccessStatus, ConditionalAccessPolicies["result"]
```

Conditional Access Overview blade

The overview blade is very informative, like the coverage shown below.



You would strive at getting to 100% coverage for users for applications accessed and 0% of users not covered for applications access overall.

CA Automation

We suggest using Azure DevOps CI/CD pipelines to manage changes to CA Policies as opposed to doing it manually in the Azure Portal. This reduces the risk of human mistakes when configuring CA policies and makes it possible to include approval workflows for changes as well as ability to revert to a previous set of policies.

Below is some guidance on how to setup Azure DevOps CI/CD for automating CA policies based on Microsoft365DSC. It is based on the existing guidance from the community site: <https://microsoft365dsc.com> . Currently the guidance does not include how to also automate the creation of the Azure resources, in this draft they are created manually.

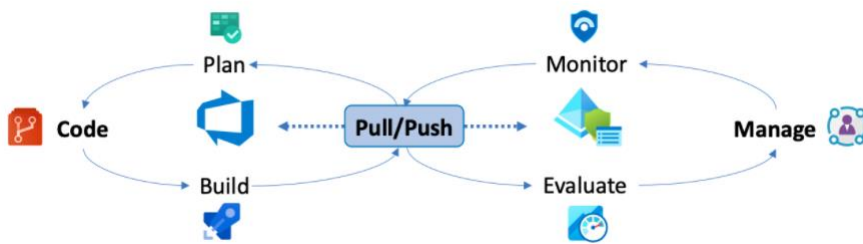
[Managing Microsoft 365 in true DevOps style with Microsoft365Dsc and Azure DevOps](#)

Other guidance is available on ideas on why/what and how to automate Conditional Access here:

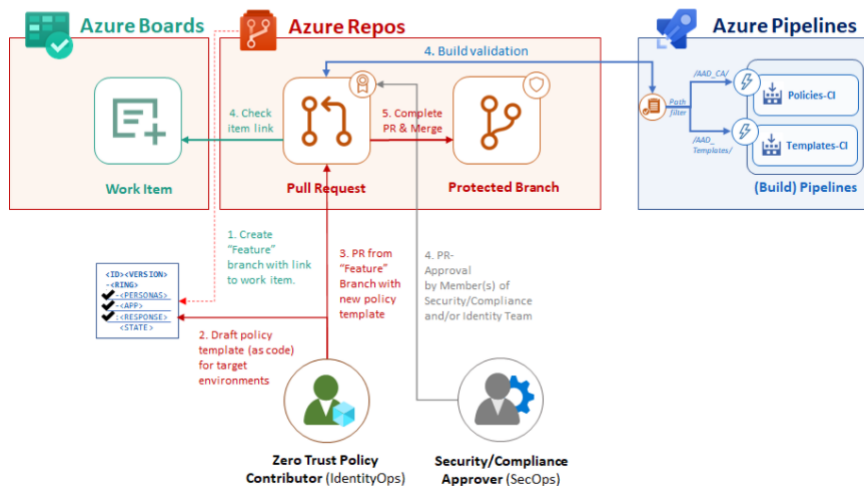
[Conditional Access APIs and PowerShell - Azure Active Directory | Microsoft Docs](#)

[Cloud-Architekt.net | Overview of Azure AD \(Conditional Access\) automation](#)

[Cloud-Architekt.net | AADOps: Operationalization of Azure AD Conditional Access](#)



Code: Create policy-as-code template in repo



Source: [Cloud-Architekt.net | AADOps: Operationalization of Azure AD Conditional Access](https://cloud-architekt.net/AADOps/Operationalization%20of%20Azure%20AD%20Conditional%20Access)

As part of the CA guidance in this document it is chosen to use the Microsoft365DSC solution as a starting point with some suggested Zero Trust related changes.

For example, the Microsoft365 covers most Microsoft365 services and because not all these services can be managed using PowerShell or Microsoft Graph based on certificate authentication, but rather only using a password. The current documented Microsoft365DSC automation solution is based on the following

- Self-Hosted DevOps pipeline agent
- Azure DevOps using yaml with separate build and release pipelines
- Standard user/password-based account with Global Administrator role assigned
- Service account with local administrator rights for DevOps agent

PS! The Microsoft365DSC automation white paper has been updated per primo October and now includes guidance on how to authenticate using certificates and graph, - however it is still based on using a self-hosted agent. The table below shows for which scenarios the certificate authentication can be used (source: microsoft365dsc.com white paper)

Workload	PowerShell Module	Credential	Certificate Thumbprint	Certificate Path	Application Secret*
AzureAD	AzureADPreview (Connect-AzureAD)	✓	✓	✗	✗
Exchange Online	ExchangeOnlineManagement (Connect-ExchangeOnline)	✓	✓	✓	✗
Intune	Microsoft.Graph.Intune (Connect-MSGraph)	✓	✗	✗	✗
Office 365	AzureADPreview (Connect-AzureAD)	✓	✓	✗	✗
OneDrive	SharePointPnPPowerShellOnline (Connect-PnPOnline)	✓	✓	✓	✗
PowerApps	Microsoft.PowerApps.Administration.PowerShell	✓	✓	✗	✗
Planner	Microsoft.Graph.Authentication (Connect-Graph)	✗	✓	✗	✗
Security & Compliance Center	ExchangeOnlineManagement (Connect-IPSSession)	✓	✗	✗	✗
Skype for Business Online	MicrosoftTeams (New-CSOnlineSession)	✓	✗	✗	✗
SharePoint Online	SharePointPnPPowerShellOnline (Connect-PnPOnline)	✓	✓	✓	✗
Teams	MicrosoftTeams (New-CSOnlineSession)	✓	✓	✗	✗
* Application secret is possible with the OneDrive, PowerApps and SharePoint cmdlets, but has not been implemented for Microsoft365DSC yet.					

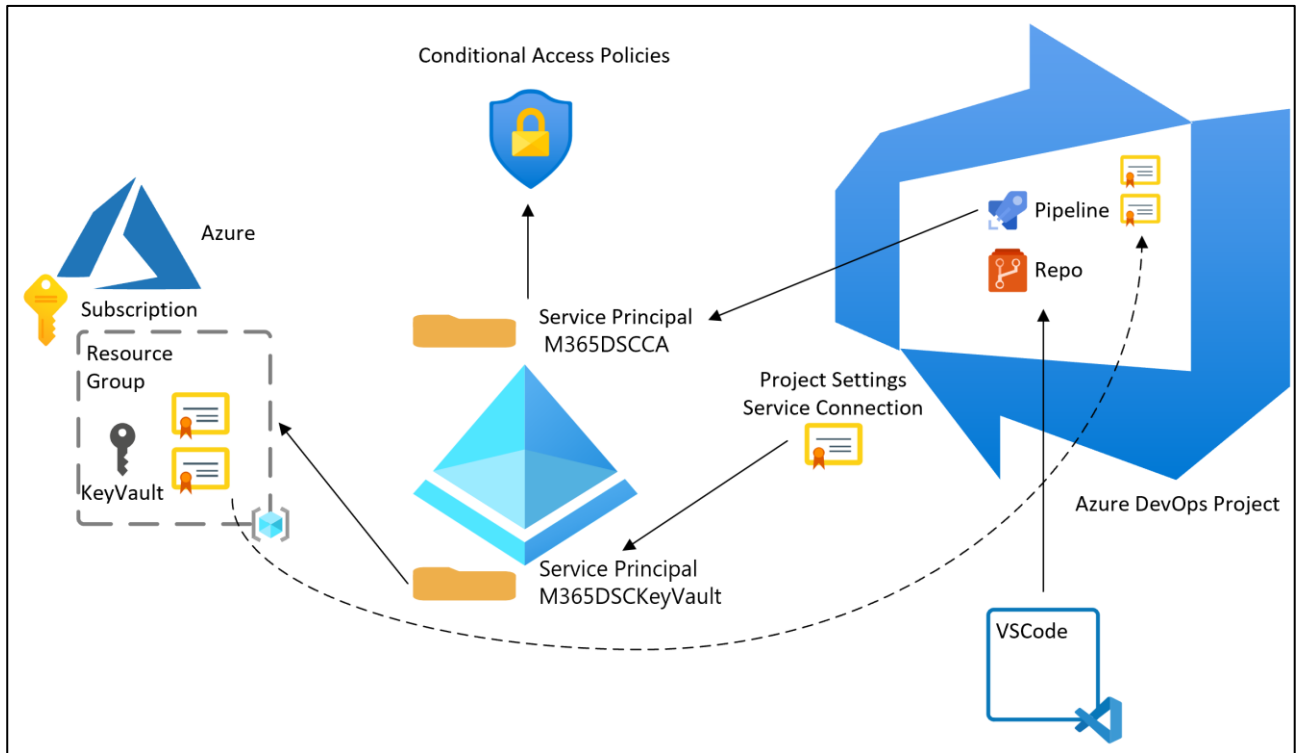
For Conditional Access the modules involved in the automation solution do support certificate-based authentication, which leads us to the following suggested approach.

- Microsoft hosted DevOps agent
- Azure DevOps using yaml with one multistage pipeline as opposed to using separate DevOps release pipeline
- Self-signed certificate with public/private key for DevOps access/permissions to Azure subscription, stored in Azure DevOps service connection setup. The service connection with the associated Azure AD service principal (M365DSCKeyVault) setup manually using applications permissions. Public key of the certificate is configured to be allowed access with permissions to the KeyVault resource.
- Self-signed certificate with public/private key stored in Azure KeyVault for signing the DSC configuration
- Self-signed certificate with public/private key stored in Azure KeyVault used to access Conditional Access through a registered service principal (M365DSCCA). Public key of this certificate is configured to allow using the app based on application permissions.

A discussion point is also using KeyVault to store the two certificates as opposed to using DevOps itself, just as the certificate used for the service connection is kept within the DevOps project. You can argue, that if you have access to DevOps, you can (indirectly) get access to the certificates in KeyVault as well, so why not just store them as secret variables in DevOps.

In general it is considered better to use the KeyVault approach. A threat model would be the right next step to determine what is considered to have the appropriate level of protection. It also depends on roles and permission in the organization to DevOps.

The above components and how the interrelate is shown in the figure below



It has been considered how to use a Managed Identity to change the CA policies through code in Azure DevOps. However, to use System defined Managed Identities from a DevOps Agent, we would need to use a self-hosted DevOps agent running in a Virtual Machine with Managed Identity enabled which implies other security implications as we would need to manage and continuously secure such a virtual machine that would run that agent.

Another consideration is whether to use just one certificate and one service principal as opposed to three certificates and two service principals. Currently it is based on the latter to separate access and permissions, so that the certificate stored in DevOps does not have permissions to change the CA policies directly.

[Export current CA policies](#)

To get started with the automation, you may want to start by exporting existing CA policies from your Azure AD tenant as opposed to building the Microsoft365DSC config file from ground up. The Azure AD Module has a cmdlet that can export CA policies

```
install-module azuread
$Policies = get-azureadmsconditionalaccesspolicy
$Policies.toJson() | out-file capolicieswithpolicyid.json
```

To save the policies without id: for easy import with graph using the following command

```
get-content .\capolicieswithpolicyid.json | select-string -pattern '"id"' -notmatch | out-file capolicieswithoutpolicyid.json
```

Another tool is available that includes some nice functions that can convert GUIDs to Display Names and vice versa to be able to also migrate policies between tenants

<https://github.com/Fortigi/ConditionalAccess>

The Microsoft365DSC has an export command that you can use as documented here: [Extracting Configuration from an Existing Microsoft 365 Tenant - Microsoft365DSC - Your Cloud Configuration](#)

An example of the export command focused on Conditional Access using certificate based service principal to export is shown below

```
Export-M365DSCConfiguration -quiet -Components @("AADConditionalAccessPolicy") -  
ConfigurationName "ConfigureConditionalAccess" -path "C:\temp" -filename  
"Microsoft365DSCCA.ps1" -ApplicationId <appid of serviceprincipal> -CertificateThumbprint  
<thumbprint of your cert to access service principal> -TenantId "<domain>.onmicrosoft.com"
```

You can adjust/edit the extracted .ps1 file and push it to the Azure DevOps repo as it will be referenced in the pipelines, and this is our main source for all the CA policies. In the example pipelines, the name Microsoft365DSCCA.ps1 is used

Another simple way of exporting CA policies from one tenant to another based on a credential that has the right permissions to do so is described below. This export is done interactive and must adhere to relevant existing CA policies in place both in the source as well as in the test tenant. You may want to use this method to get to know the Microsoft365dsc module before fully automating it using the certificate-based approach.

The described process below assumes you have all policies and CA related groups configured in a test tenant. Instead of having to create all CA groups and policies in a source tenant, you can also just use the two resources exported from a test tenant (I can provide such a file for you). .

If you choose to do that, you should skip step 6-15 as you would just put the files in the two folders and apply them to the target tenant.

- 1) Use Windows PowerShell or Windows Terminal with PowerShell (Windows PowerShell Core 7.2 does not work with both export/import described here)
- 2) Launch a PS session as local admin
- 3) Winrm quickconfig
- 4) Install-module Microsoft.graph
- 5) Install-module microsoft365dsc
- 6) Optionally if you get an error about maximumfunctioncount, you can increase this:
\$MaximumFunctionCount = 8192, \$MaximumVariableCount = 8192
- 7) Select-mgprofile -name beta
- 8) Import-module microsoft365dsc

- 9) \$credential = get-credential (specify credential for source tenant with permissions that can be granted as specified below)
- 10) Connect-graph -scopes Application.Read.All, Group.ReadWrite.All, Directory.Read.All, Policy.Read.All, Policy.Read.ConditionalAccess, Policy.ReadWrite.ConditionalAccess, RoleManagement.Read.All, RoleManagement.Read.Directory, User.Read.All -tenantid <sourcetenant> (to consent to graph permissions, read is enough in the source tenant, but read/write is needed for target tenant)
- 11) Create two local directories, like AADConditionalAccessPolicy and AADGroup
- 12) Shift to the AADGroup directory
- 13) \$credential = get-credential , and specify the credentials for the source tenant
- 14) Export-M365DSCConfiguration -Components @("AADGroup") -credential \$credential
- 15) Review and adjust the content of the M365TenantConfig.ps1 script and remove any group that you don't want to have in the target tenant. For example search for groups with CA-Persona and remove any group that does not start with this.
- 16) Execute the script M365TenantConfig.ps1, specify the credential of the target tenant. This creates a compiled mof file that can be applied in the target tenant. Be careful about the mof file. It includes the password for the credential uses, and you want to remove this password from the file after having used it.
- 17) Shift to the AADConditionalAccessPolicy directory
- 18) Export-M365DSCConfiguration -Components @("AADConditionalAccessPolicy") -credential \$credential
- 19) Execute the script M365TenantConfig.ps1, specify the credential of the target tenant. This creates a compiled mof file that can be applied in the target tenant. Be careful about the mof file. It includes the password for the credential uses, and you want to remove this password from the file after having used it.
- 20) Adjust (temporarily) CA policies in target tenant to not require MFA for the used user as the start-dscconfiguration command does not support MFA.
- 21) Connect-graph -scopes Application.Read.All, Group.ReadWrite.All, Directory.Read.All, Policy.Read.All, Policy.Read.ConditionalAccess, Policy.ReadWrite.ConditionalAccess, RoleManagement.Read.All, RoleManagement.Read.Directory, User.Read.All -tenantid <targettenant> (to consent to graph permissions, read is enough in the source tenant, but read/write is needed for target tenant)
- 22) Shift to the AADGroup folder as we want to create groups in the target tenant before creating the CA policies
- 23) Start-dscconfiguration -wait -force M365TenantConfig. You should now have the groups created in the target tenant. Remember to remove the password from the mof file.

24) Shift to the AADConditionalAccessPolicy folder

25) Start-dscconfiguration -wait -force M365TenantConfig. You should now have the CA policies created in the target tenant. Remember to remove the password from the mof file. Remember to remove the password from the mof file.

The configuration is described in more details in the sections below. Please notice that the guidance included below for the automation is to be used for a proof of concept and as a suggested starting point.

The code and scripts need more quality assurance before putting into production. Also you may want to include more details on the various branches and approval workflows for submitting changes to the code i.e.

Azure AD app registrations

We create two service principals as app registrations in Azure AD with the following settings

Service Principal Setting	Service Principal to access KeyVault	Service Principal to read and write Conditional Access policies
Name	M365DSCKeyVault	M365DSCCA
Application(Client) ID	Be sure to note this as it must be used to assign KeyVault permissions to this Client ID	Update documentation after created. This will be used as a parameter in the DevOps build pipeline.
Owners	Be sure to add owners, following your organization governance procedures	Be sure to add owners, following your organization governance procedures
Authentication	Accounts in this organizational directory only	Accounts in this organizational directory only
Microsoft Graph Application Permissions	Default (User.Read)	Application.Read.All Directory.Read.all Group.Read.All Policy.ReadWrite.ConditionalAccess RoleManagement.Read.All RoleManagement.Read.Directory User.Read.All
Certificates & Secrets	Public Key of service connection cert (see section on how to create this self-signed cert)	Public key of m365dscacert

Azure Subscription configuration

Create a new resource group for Microsoft365DSC automation to limit access to these resources based on least privileged Zero Trust principle.

Create a KeyVault resource within the resource group with the settings as specified below

KeyVault object	Setting
Name	Microsoft365DSCkv

Certificate	M365dscsigningcert Name:m365dscsigningcert Self-Signed Certificate Subject: CN=m365dscsigningcert Validity Period (Months): 12 Content Type: PKCS#12 LifeTime: Automatically review Percentage Lifetime: 80 Advanced Policy Configuration: Default Enable cert
Certificate	M365dscacert Name:m365dscacert Subject: CN=m365dscacert Validity Period (Months): 12 Content Type: PKCS#12 LifeTime: Automatically review Percentage Lifetime: 80 Advanced Policy Configuration: Default Enable cert
Access policies	Permission Model: Vault access policy Enable access to: Azure Resource Manager for template deployment Access Policy Applications: M365DSCkeyvault service principal Secret Permission: Get, List Certificate Permissions: Get, List
Networking	All networks (we connect from Azure DevOps pipeline agent)

It is recommended to enable Azure Defender for KeyVault if you have license for it.

[Azure DevOps configuration](#)

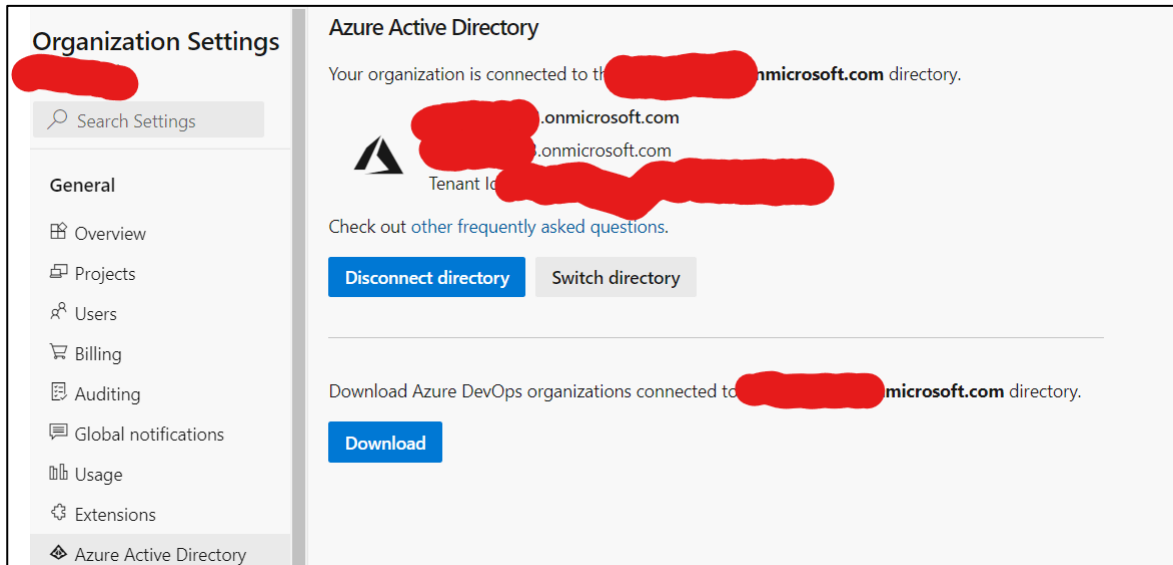
The suggested Azure DevOps configuration is described below. You may want to adjust to align with your internal policies and procedures.

[Azure DevOps Organization settings](#)

We configure a separate project in Azure DevOps, typically in a new or in an existing Azure DevOps organization depending on security requirements and roles and responsibilities. As access to the projects implies indirect access to change CA policies, the permissions must be tightly controlled.

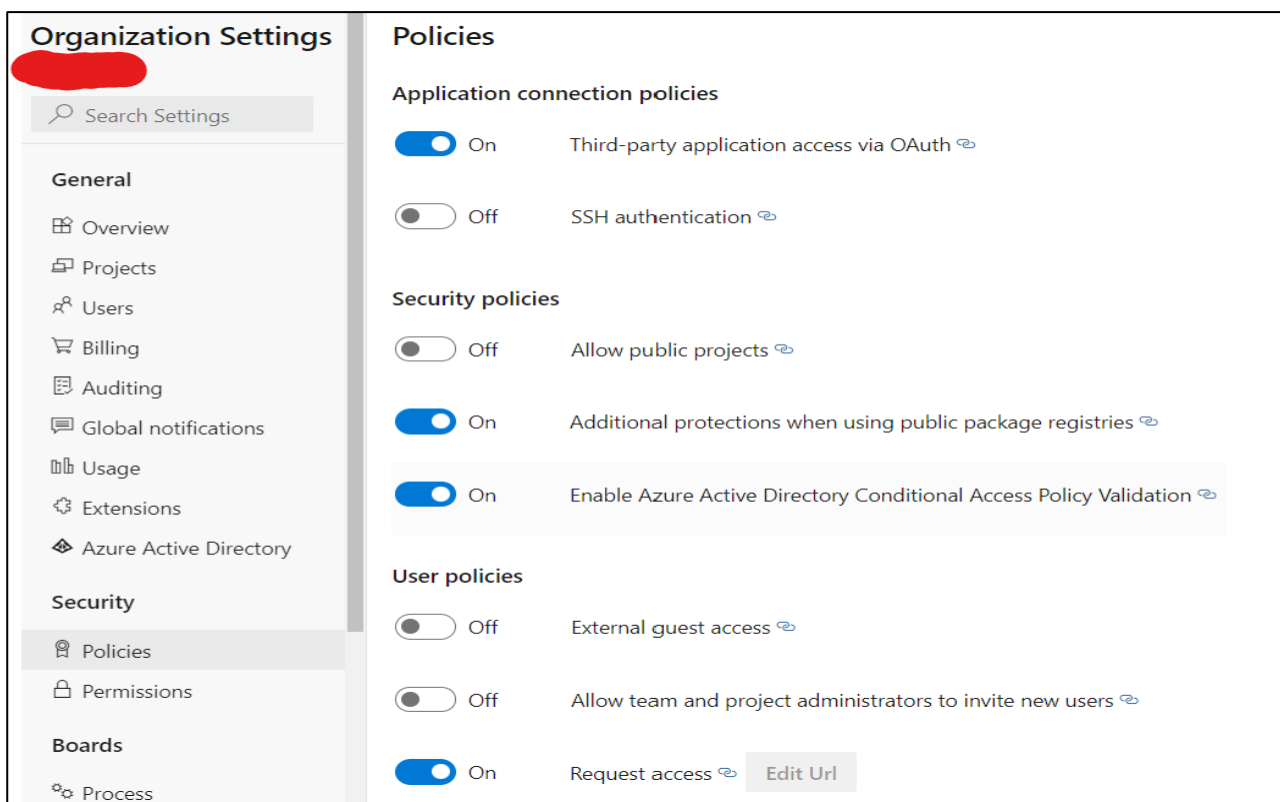
[DevOps Azure AD Integration](#)

Be sure to create the project in an organization that is integrated with Azure AD to be able to control access using Conditional Access.



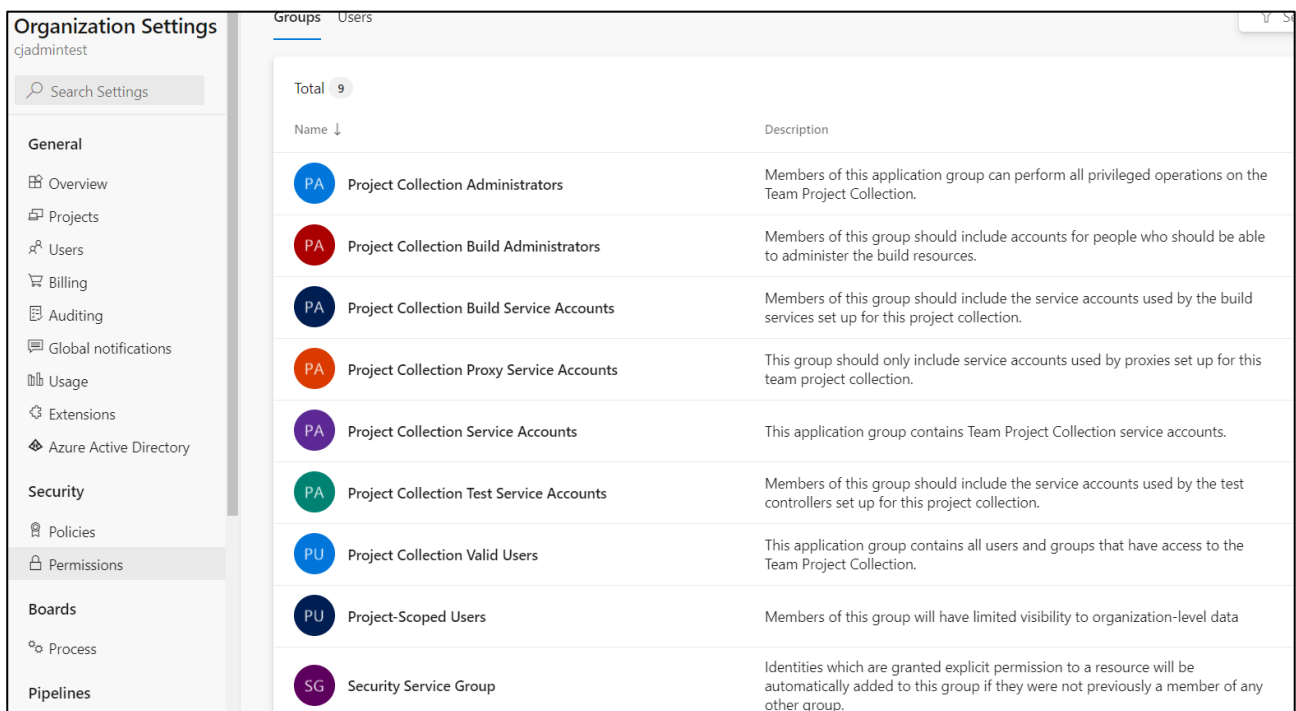
Organization security policies and permissions

Also assure that the security policies and permissions are configured rather strict, using the figure below as a suggested setup that needs to be discussed and aligned with the organization std. policies for use of Azure DevOps. Notice that Conditional Access is being enabled for the organization.



NB! May 2022 this is outdated. Conditional Access enablement is not available as a configuration item on the organization level, it is rather directly integrated with AAD CA as a targetable app when integrated with AAD.

If the permissions for the organization are a bit too open for this sensitive project to be hosted in it, consider creating a separate organization. But else adjust the roles shown below to your needs.



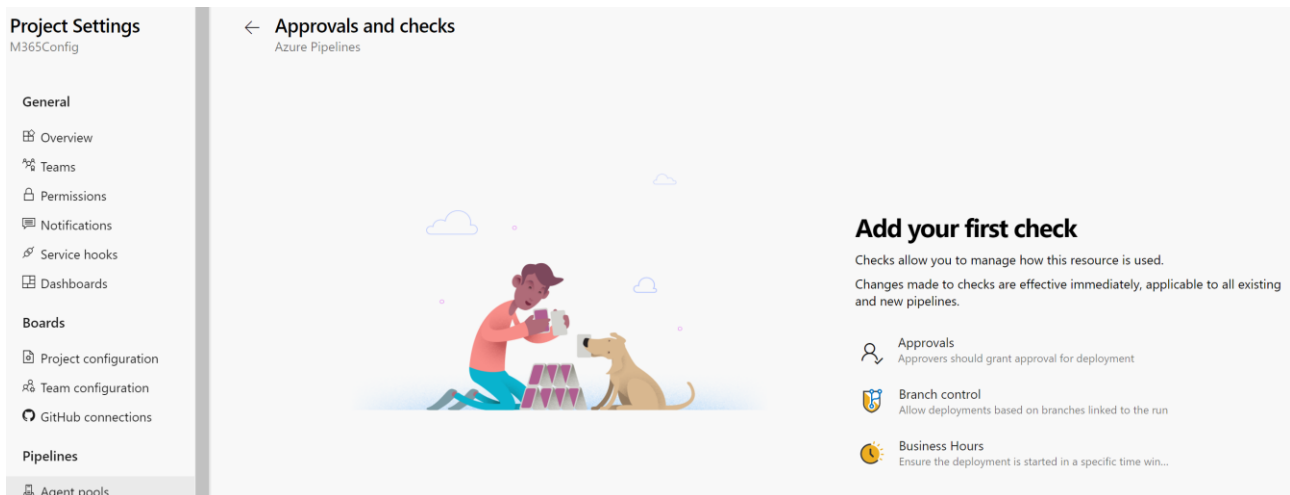
The screenshot shows the 'Organization Settings' interface for 'cjadmintest'. The left sidebar contains navigation links for General, Security, Boards, and Pipelines. The 'Groups' tab is active, displaying a table of application groups. The table has columns for 'Name' and 'Description'. There are 9 groups listed in total.

Name	Description
Project Collection Administrators	Members of this application group can perform all privileged operations on the Team Project Collection.
Project Collection Build Administrators	Members of this group should include accounts for people who should be able to administer the build resources.
Project Collection Build Service Accounts	Members of this group should include the service accounts used by the build services set up for this project collection.
Project Collection Proxy Service Accounts	This group should only include service accounts used by proxies set up for this team project collection.
Project Collection Service Accounts	This application group contains Team Project Collection service accounts.
Project Collection Test Service Accounts	Members of this group should include the service accounts used by the test controllers set up for this project collection.
Project Collection Valid Users	This application group contains all users and groups that have access to the Team Project Collection.
Project-Scoped Users	Members of this group will have limited visibility to organization-level data
Security Service Group	Identities which are granted explicit permission to a resource will be automatically added to this group if they were not previously a member of any other group.

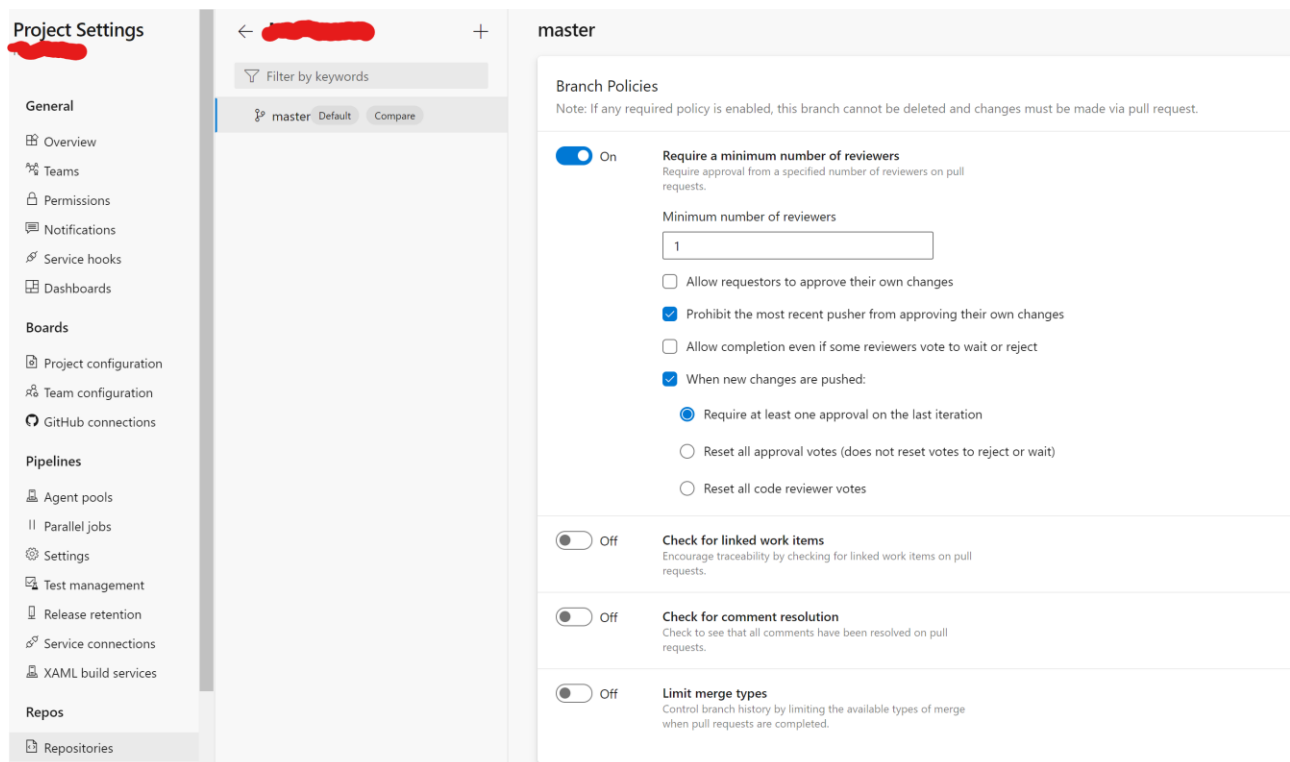
Agent Pools

Assure that Azure Pipelines (this is the Microsoft hosted/managed agent pool) is added to the Agent pools on the organization level to be used in the project we create. As agent pools are configured per organization, this may be a reason why you may want to have a separate organization for your project.

You typically want to add approvals and checks as shown below



And as an example if you just have one master branch, you may want to require reviewers as shown below



This means that changes to the master branch needs to be done based on Pull Requests and merging from a different branch to the master branch.

Service Connection

Create a service connection, call it something like Microsoft365DSC and specify details as shown below. You will have to choose service principal (manual). You want to be sure that you choose certificate as credentials

Edit service connection ✕
Azure Resource Manager using service principal (manual)

Scope Level

- ☒ Subscription
- ☐ Management Group
- ☐ Machine Learning Workspace

Subscription Id
Subscription Id from the publish settings file

Subscription Name
Subscription Name from the publish settings file

Authentication

Service Principal Id
Client id for connecting to the endpoint. Refer to [Azure Service Principal link](#) on how to create Azure Service Principal.

Credential

- ☐ Service principal key
- ☒ Certificate

Certificate
Provide PEM file content. Include both certificate and private key content. Ignore this field if the authentication type is spnKey.

Tenant ID
Tenant id for connecting to the endpoint. Refer to [Azure Service Principal link](#) on how to create Azure Service Principal.

[Learn more](#) [Troubleshoot](#) Cancel Verify and save ▼

You now create a self-signed certificate that will be used to access the Azure KeyVault as a resource in an Azure Subscription. This can be done by running the script below.

Remember to give the service principal read access to the Azure subscription used.

```
openssl req -newkey rsa:4096 -nodes -keyout "service-principal.key" -out "service-principal.csr"
```

```
openssl x509 -signkey "service-principal.key" -in "service-principal.csr" -req -days 365 -out "service-principal.crt"
```

```
openssl pkcs12 -export -out "service-principal.pfx" -inkey "service-principal.key" -in "service-principal.crt"
```

```
openssl pkcs12 -in "service-principal.pfx" -out "service-principal.pem" -nodes
```

edit .pem file so that it only includes Bag Attributes like shown below

Bag Attributes

W

-----BEGIN CERTIFICATE-----

AA
AA

...

AAAAAAAAAAAAAAAAAAAAAAAAAAAA

-----END CERTIFICATE-----

Bag Attributes

-----BEGIN PRIVATE KEY-----

AA
AA

...

AAAAAAAAAAAAAAAAAAAAAAAAAAAA

-----END PRIVATE KEY-----

The public key of this certificate should be uploaded/configured to the Azure AD service principal M365DSCKeyVault used to access the KeyVault. Delete the .pem file immediately after you finished the above configuration, as it is a very sensitive file with private key included.

We will create the other two certificates directly in KeyVault after we have created the KeyVault in the Azure Subscription.

Azure DevOps Variable Group

We create a variable group in the DevOps project, under the pipeline library. It is used at runtime to store the values of secrets stored in the KeyVault.

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Releases

Library

Task groups

Deployment groups

Library

Boards

Work items

Boards

Backlogs

Sprints

Queries

Delivery Plans

+ Variable group

Security

Help

(x)

New variable group

Create groups of variables that you can share across multiple pipelines.

+ Variable group

[Learn more about variable groups.](#)

Library > Kvcavargroup*

Variable group

Save

Clone

Security

Pipeline permissions

Approve

Properties

Variable group name

Kvcavargroup

Description

Kvcavargroup

☒

Link secrets from an Azure key vault as variables

?

Azure subscription *

| Manage

↻

ⓘ This setting is required.

Key vault name *

| Manage

↻

Invalid Value

ⓘ This setting is required.

The variable group has the following settings.

Parameter	Value
Value	Kvcavargroup
Azure Subscription	<Your Azure Subscription used for this project>
KeyVaultName	M365DSCKeyVault
Variables:	SecretName: m365dscacert ContentType: application/x-pkcs12
	SecretName: m365dscsigningcert ContentType: application/x-pkcs12

Now create a pipeline in the project but before doing that, be sure to initialize the repo for the project.

Azure DevOps YAML multistage pipeline

The YAML multistage pipeline is shown below

```
trigger:
- master

pool:
  vmImage: windows-latest

variables:
  - group: kvcavargroup

stages:
- stage: Build
  jobs:
  - job: Build

    steps:

    - task: PowerShell@2
      inputs:
        targetType: 'filePath'
        filePath: 'prep.ps1'
        arguments: -m365dscacert $(m365dscacert) -m365dscsigningcert $(m365dscsigningcert)
        errorActionPreference: 'stop'

    - task: PowerShell@2
      inputs:
        targetType: 'filePath'
        filePath: 'build.ps1'
        arguments: -m365dscacert $(m365dscacert)
        errorActionPreference: 'stop'

    - task: PublishPipelineArtifact@1
      inputs:
        targetPath: $(Build.ArtifactStagingDirectory)
        artifactName: 'MofFile'
        artifactType: 'pipeline'

- stage: Deploy
  dependsOn: Build

  jobs:
  - job: Deploy
    steps:

    - task: DownloadPipelineArtifact@2
      inputs:
        artifactName: 'MofFile'
        downloadPath: $(System.ArtifactsDirectory)

    - task: PowerShell@2
      inputs:
        targetType: 'filePath'
        filePath: 'release.ps1'
        arguments: -m365dscacert $(m365dscacert) -m365dscsigningcert $(m365dscsigningcert)
        errorActionPreference: 'stop'
```

Azure DevOps Build pipeline

We use the following script with content below to prepare the build pipeline

```
[CmdletBinding()]
param (
    [Parameter(Mandatory = $True)]
    [string]$m365dscacert,
    [string]$m365dscsigningcert
)

Install-Module -Name Microsoft365Dsc -force -AllowClobber
Install-m365dscdevbranch
Import-Module Microsoft365Dsc
$location = Get-Location

$m365configcertStringBase64 = $m365dscacert
$m365configcertByteArray = [System.Convert]::FromBase64String($m365configcertStringBase64)
[System.IO.File]::WriteAllBytes("$location/m365dscacert.pfx", $m365configcertByteArray)
$m365configcert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($m365configcertByteArray)
$m365configcertThumbprint = $m365configcert.Thumbprint
Import-PfxCertificate -FilePath $location/m365dscacert.pfx -CertStoreLocation Cert:\LocalMachine\My
$m365configcert.Dispose()
$m365configcert = $null
Remove-Item "$location/m365dscacert.pfx"

$dsccertStringBase64 = $m365dscsigningcert
$dsccertByteArray = [System.Convert]::FromBase64String($dsccertStringBase64)
[System.IO.File]::WriteAllBytes("$location/m365dscsigningcert.pfx", $dsccertByteArray)
$dsccert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($dsccertByteArray)
$dsccertThumbprint = $dsccert.Thumbprint
Import-PfxCertificate -FilePath $location/m365dscsigningcert.pfx -CertStoreLocation Cert:\LocalMachine\My
$dsccert.Dispose()
$dsccert = $null
Remove-Item "$location/m365dscsigningcert.pfx"

winrm quickconfig -force

Configuration ConfigureLCM
{
    param (
        [Parameter(Mandatory = $True)]
        [string]$dsccertThumbprint
    )

    Import-DscResource -ModuleName PsDesiredStateConfiguration
    node localhost
    {
        LocalConfigurationManager
        {
            {
                CertificateId = $dsccertThumbprint
            }
        }
    }
}
ConfigureLCM -dsccertThumbprint $dsccertThumbprint

set-DscLocalConfigurationManager -Path $location\ConfigureLcm\ -force -verbose
```

Build.ps1 holds the build pipeline with the content showed below.

```
[CmdletBinding()]
param (
    [Parameter(Mandatory = $True)]
    [string]$m365dscacert
)

$m365configcertStringBase64 = $m365dscacert
$m365configcertByteArray = [System.Convert]::FromBase64String($m365configcertStringBase64)
$m365configcert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($m365configcertByteArray)
$m365configcertThumbprint = $m365configcert.Thumbprint

. .\Microsoft365DSCCA.ps1 -CertificateThumbprint $m365configcertThumbprint -TenantId <yourtenantid> -
ApplicationId <M365DSCCA appid>

ConfigureConditionalAccess -CertificateThumbprint $m365configcertThumbprint -TenantId <yourtenantid> -
ApplicationId <M365DSCCA appid> -OutputPath $env:Build_ArtifactStagingDirectory
```

Azure DevOps Release pipeline

```
[CmdletBinding()]
param (
    [Parameter(Mandatory = $True)]
    [string]$m365dscacert,
    [string]$m365dscsigningcert
)

Install-Module -Name Microsoft365Dsc -force -AllowClobber
Install-m365dscdevbranch
Import-Module Microsoft365Dsc
$location = Get-Location

$m365configcertStringBase64 = $m365dscacert
$m365configcertByteArray = [System.Convert]::FromBase64String($m365configcertStringBase64)
[System.IO.File]::WriteAllBytes("$location/m365dscacert.pfx", $m365configcertByteArray)
$m365configcert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($m365configcertByteArray)
$m365configcertThumbprint = $m365configcert.Thumbprint
Import-PfxCertificate -FilePath $location/m365dscacert.pfx -
CertStoreLocation Cert:\LocalMachine\My
$m365configcert.Dispose()
$m365configcert = $null
Remove-Item "$location/m365dscacert.pfx"

$dsccertStringBase64 = $m365dscsigningcert
$dsccertByteArray = [System.Convert]::FromBase64String($dsccertStringBase64)
[System.IO.File]::WriteAllBytes("$location/m365dscsigningcert.pfx", $dsccertByteArray)
$dsccert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($dsccertByteArray)
$dsccertThumbprint = $dsccert.Thumbprint
Import-PfxCertificate -FilePath $location/m365dscsigningcert.pfx -
CertStoreLocation Cert:\LocalMachine\My
$dsccert.Dispose()
$dsccert = $null
Remove-Item "$location/m365dscsigningcert.pfx"

winrm quickconfig -force

Configuration ConfigureLCM
{
    param (
        [Parameter(Mandatory = $True)]
        [string]$dsccertThumbprint
    )

    Import-DscResource -ModuleName PsDesiredStateConfiguration
    node localhost
    {
        LocalConfigurationManager
        {
            {
                CertificateId = $dsccertThumbprint
            }
        }
    }
}
ConfigureLCM -dsccertThumbprint $dsccertThumbprint
```

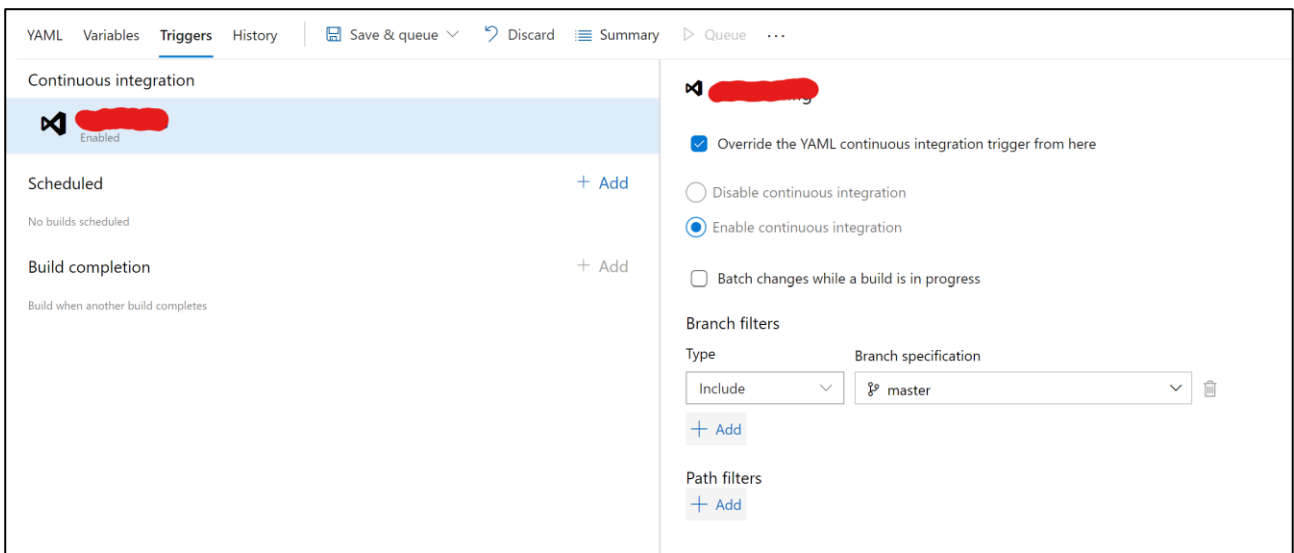

Continuous integration and deployment – triggers

We want to have continuous integration and continuous deployment enabled, meaning that each time a change to the CA policies has been committed/approved, we want to automatically deploy the new CA policies. Also even so there are no changes, we want to utilize the desired state configuration to check if the running set of policies have been changed and if so, align with the approved policies in the repository.

The way we do this is described below.

Continuous Integration

We can have continuous integration enabled either directly configured/enabled in the yaml file (through the line: (triggers: - master) or by overwriting it as shown below as part of the pipeline triggers configuration



Continuous Delivery

Continuous Delivery can be enabled in the yaml file using the format

Schedules: (like each four hours)

- cron: `"* * 4 * * *"`
displayName: schedule 4 hours build/release
always: true
branches:
 include:
 - master

Cron syntax:

mm HH DD MM DW

\ \ \ \ \ Days of week

\ \ \ \ Months

\ \ \ Days

\ \ Hours

\ Minutes


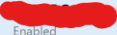
It can also be configured through the portal as a trigger in the pipeline, but seems easier to just do in the pipeline as you would have to have multiple triggers to have it running each four hours.

When triggers and schedules are configured only in the yaml file, it is important to have the following setting in the GUI: (which means that we don't want to override the YAML configuration)


☐ Override the YAML continuous integration trigger from here


YAMLVariablesTriggersHistorySave & queueDiscardSummaryQueue...


Continuous integration


 
Enabled


Scheduled+ Add


 0:00
Mon through Sun

 4:00
Mon through Fri

 8:00
Mon through Fri



 12:00
Mon through Fri

 16:00
Mon through Fri

 20:00
Mon through Fri

Build completion+ Add

Build when another build completes

☐ Override the YAML continuous integration trigger from here

References:

[Best practices for Azure AD application registration configuration - Microsoft identity platform | Microsoft Docs](#)

[What is Conditional Access in Azure Active Directory? | Microsoft Docs, Conditional access guidance · AlexFilipin/ConditionalAccess Wiki · GitHub](#)

[Common Conditional Access policies - Azure Active Directory | Microsoft Docs](#)

<https://microsoft365dsc.com>

<https://microsoft365dsc.com/Pages/Resources/Whitepapers/Managing Microsoft 365 with Microsoft365Dsc and Azure DevOps.pdf>

<https://microsoft365dsc.com/Pages/Resources/Whitepapers/Managing Microsoft 365 with Microsoft365Dsc and Azure DevOps.pdf>

[Managing Microsoft 365 in true DevOps style with Microsoft365Dsc and Azure DevOps](#)

[Conditional Access APIs and PowerShell - Azure Active Directory | Microsoft Docs](#)

[Cloud-Architekt.net | Overview of Azure AD \(Conditional Access\) automation](#)

[Cloud-Architekt.net | AADOps: Operationalization of Azure AD Conditional Access](#)