Microsoft 365 Infrastructure-as-Code

@thomasvochten

About me

Thomas Vochten

Technology Evangelist

#Microsoft365 #Azure #CommunityRocks

@thomasvochten

https://thomasvochten.com

mail@thomasvochten.com

Microsoft® Most Valuable Professional

BIWUG

CRONOS GROEP

Managing Microsoft 365

- Admin Center
- PowerShell

It’s your responsibility to know

- Where to go
- How to do it

Two approaches

Imperative

- Procedural
- Describes the How
- You describe
  the order of commands

Declarative

- Functional
- Describes the What
- You describe
  the desired outcome

Two approaches

Imperative

```
Set-SPOTenant -OneDriveStorageQuota 1048576
```

Declarative

```
ODSettings "ODSettings"
{
    OneDriveStorageQuota = 1048576;
}
```

Benefits of a declarative approach

- Repeatability
- Consistency
- Efficiency
- Testability
- Stability

Buzzword bingo

ARM

Bicep

Terraform

- Infrastructure-as-Code
- Configuration-as-Code

Extend your toolbox: Source Control

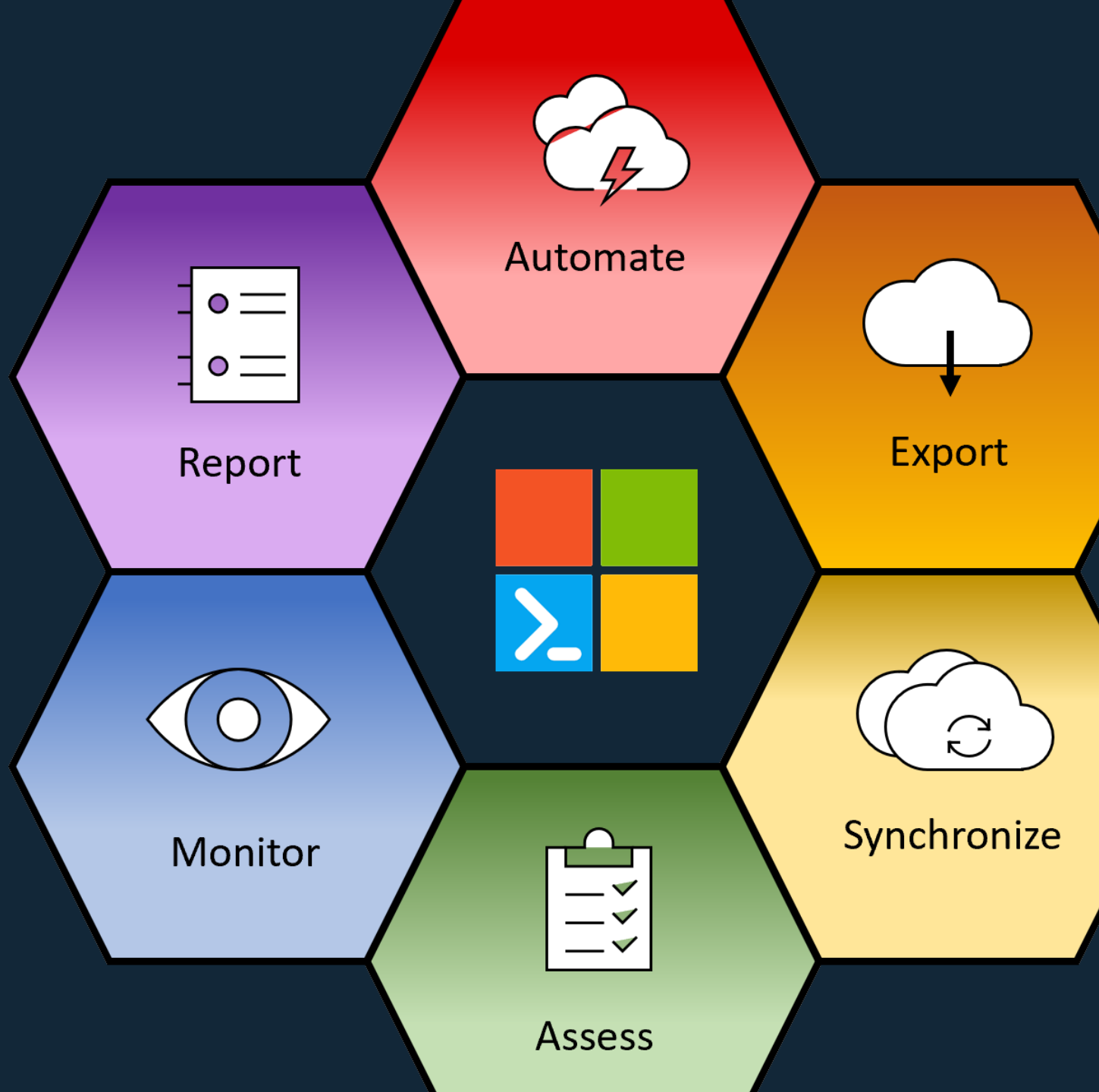
- Version history
- Collaboration
- Branches
- Releases
- Automation

What about Microsoft 365?

- Not built with Configuration-as-Code in mind
- Community or third-party solutions needed
- Always at the mercy of Microsoft API’s
- Limited to Configuration-as-Code

Microsoft 365 DSC

- Open source
- PowerShell module



Desired State Configuration?

- PowerShell-based
- The “make it so” engine
- Simple & scalable
- Build + Push or Pull (MOF files)

```
FirewallRule Rule1
{
    Name = “Rule1”
    Port = 80;
    Direction = “Inbound”
}
FirewallRule Rule2
{
    Name = “Rule2”
    Port = 90;
    Direction = “Inbound”
}
```

Automate / Apply

Start-DSCConfiguration

```
ODSettings "ODSettings"
{
    OneDriveStorageQuota = 1048576;
}
```

Example of repeatable configurations

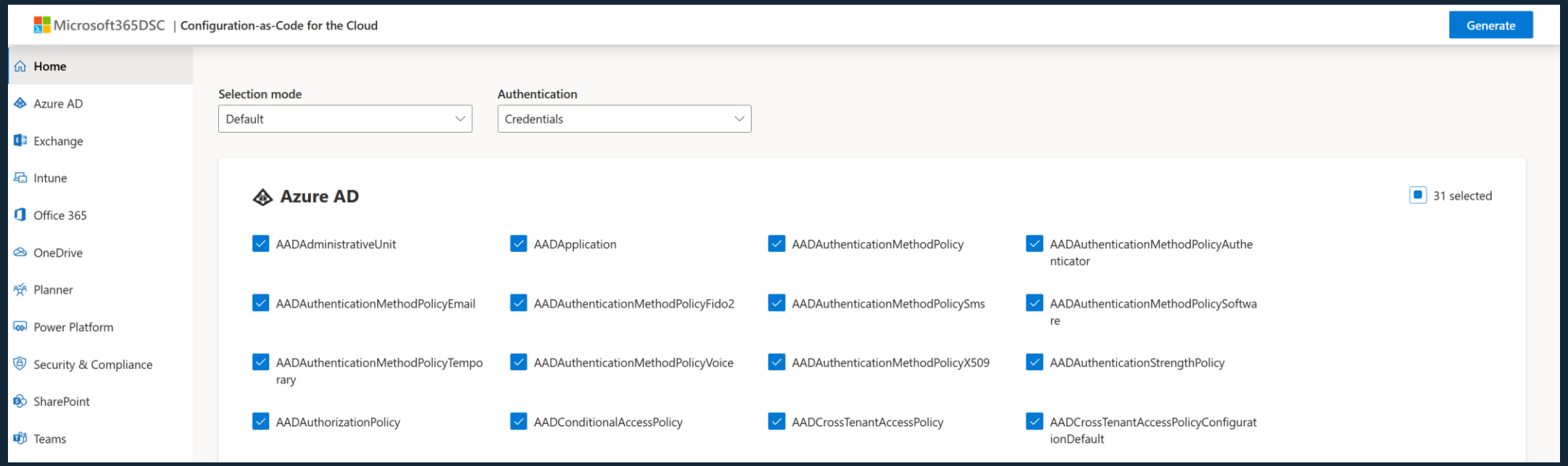
Conditional Access baseline rules to be applied across tenants:

- CA Groups
- CA Policies using those groups

```
AADConditionalAccessPolicy bf8ea9dd-154c-4fec-87a0-183bfe577248
{
    ApplicationEnforcedRestrictionsIsEnabled = $False;
    BuiltInControls                          = @("mfa","compliantDevice");
    ClientAppTypes                           = @("all");
    CloudAppSecurityIsEnabled                = $False;
    CloudAppSecurityType                     = "";
    Credential                               = $Credscredential;
    DisplayName                              = "CA100-Admins-BaseProtection-AllApps-AnyPlatform-CompliantorAADHJ";
    Ensure                                   = "Present";
    ExcludeApplications                      = @("d4ebce55-015a-49b5-a083-c84d1797ae8c");
    ExcludeDevices                           = @();
    ExcludeGroups                            = @("CA-BreakGlassAccounts","CA-Persona-Admins-BaseProtection-Exclusions","CA-Persona-Microsoft365ServiceAccounts","CA-
    ExcludeLocations                         = @();
    ExcludePlatforms                         = @();
    ExcludeRoles                             = @();
    ExcludeUsers                             = @();
    GrantControlOperator                     = "OR";
    Id                                       = "99b56601-10cc-4706-8b5b-3479ef3b0f4f";
    IncludeApplications                      = @("All");
    IncludeDevices                           = @();
    IncludeGroups                            = @("CA-Persona-Admins");
    IncludeLocations                         = @();
    IncludePlatforms                         = @("all");
    IncludeRoles                             = @();
    IncludeUserActions                       = @();
    IncludeUsers                             = @();
    PersistentBrowserIsEnabled               = $False;
    PersistentBrowserMode                    = "";
    SignInFrequencyIsEnabled                 = $False;
    SignInFrequencyType                      = "";
    SignInRiskLevels                         = @();
    State                                    = "disabled";
    UserRiskLevels                           = @();
}
```

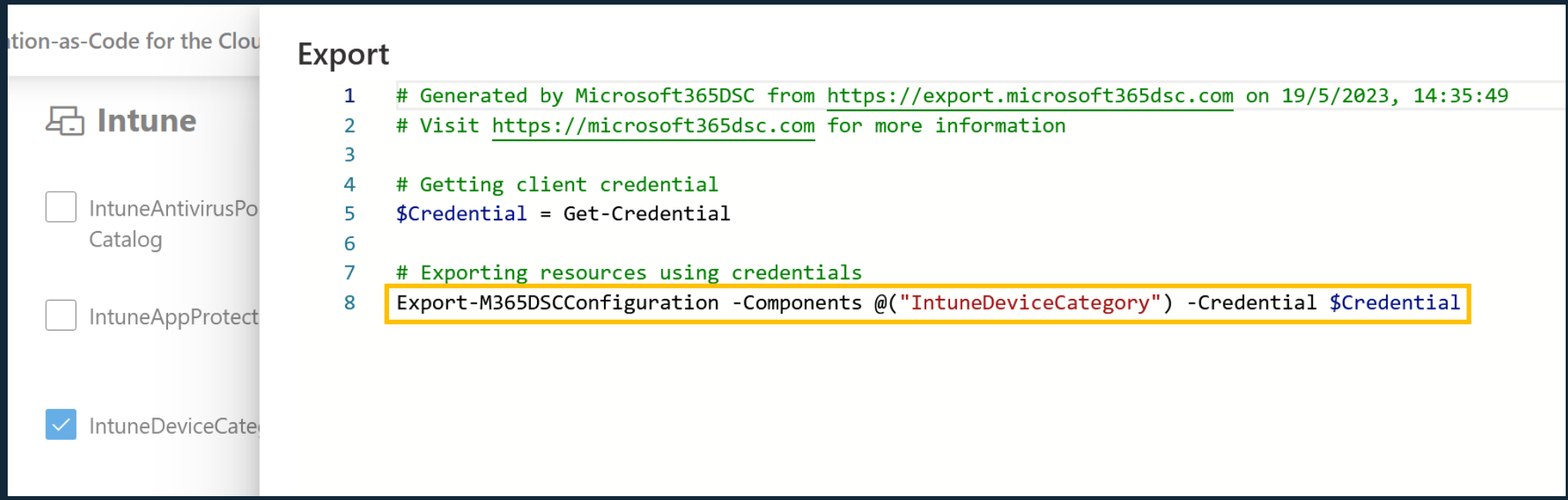
Export

```
Export-M365DSCConfiguration -LaunchWebUI
```



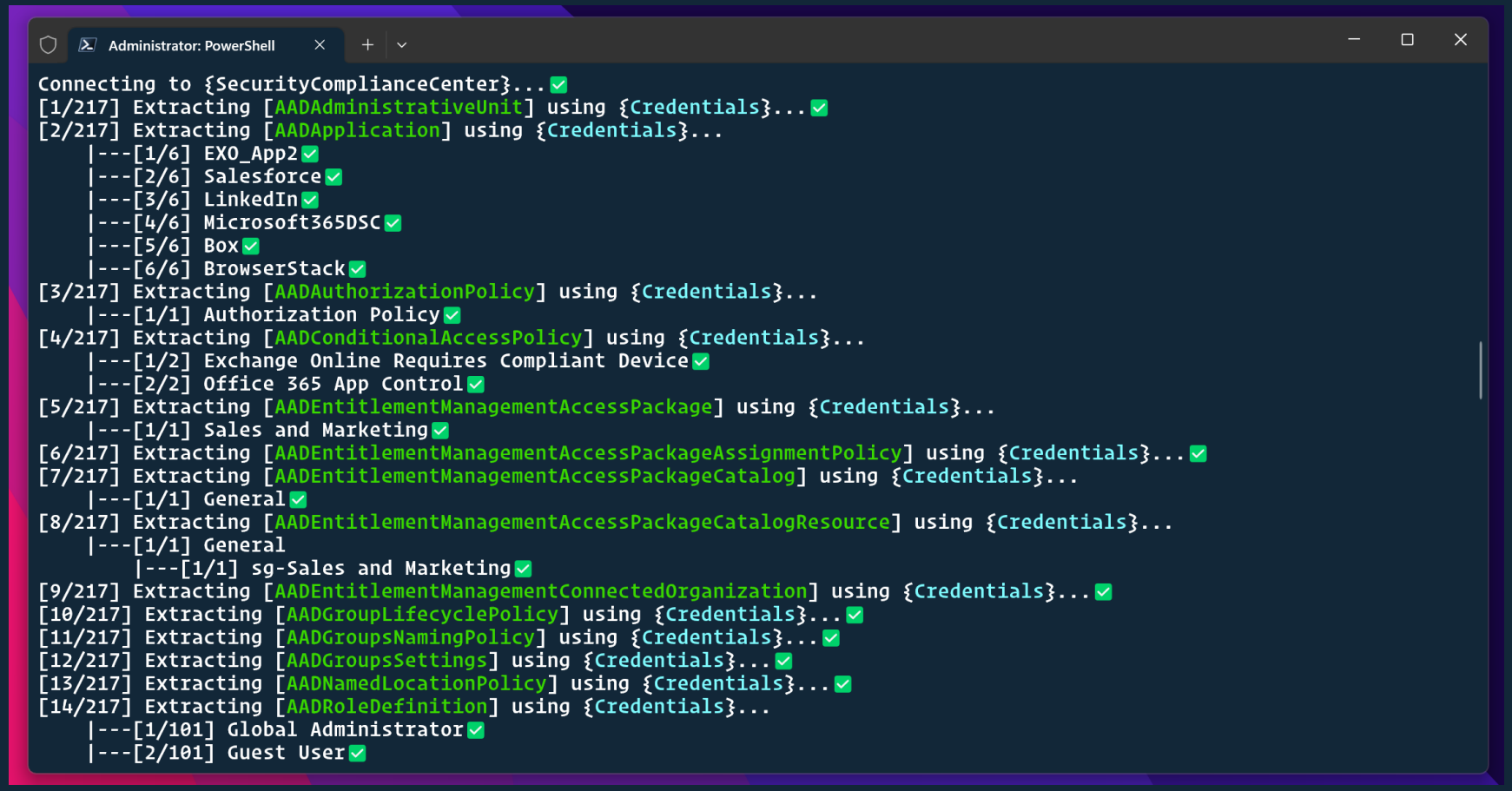
@thomasvochten

Export



@thomasvochten

Export



@thomasvochten

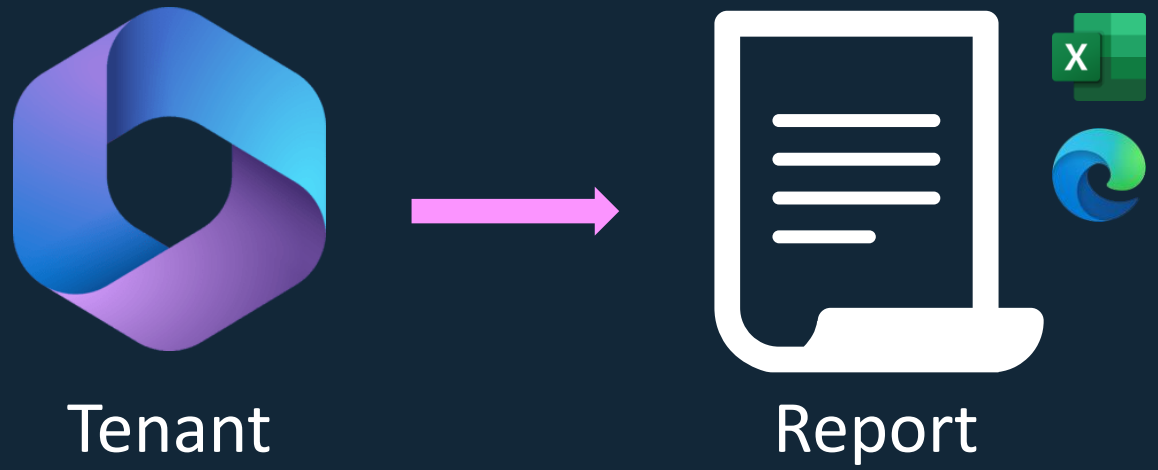
Export

```
TeamsMessagingPolicy ebef2ec9-f207-4a9d-b98b-37965c3dc6fd
{
    AllowGiphy                  = $False;
    AllowImmersiveReader        = $True;
    AllowMemes                  = $True;
    AllowOwnerDeleteMessage     = $False;
    AllowPriorityMessages       = $True;
    AllowRemoveUser             = $True;
    AllowStickers               = $True;
    AllowUrlPreviews            = $True;
    AllowUserChat               = $True;
    AllowUserDeleteMessage      = $True;
    AllowUserEditMessage        = $True;
    AllowUserTranslation        = $True;
    AudioMessageEnabledType     = "ChatsAndChannels";
    ChannelsInChatListEnabledType = "DisabledUserOverride"
    Credential                  = $Credscredential;
    Ensure                      = "Present";
    GiphyRatingType             = "Strict";
    Identity                    = "EduStudent";
    ReadReceiptsEnabledType     = "UserPreference";
}
TeamsMobilityPolicy 575d6b1f-93c7-46da-b745-814c70d6c96f
{
    Credential          = $Credscredential;
    Ensure              = "Present";
    Identity            = "Global";
    IPAudioMobileMode   = "AllNetworks";
    IPVideoMobileMode   = "AllNetworks";
    MobileDialerPreference = "Teams";
}
TeamsMobilityPolicy f70c61b2-4787-4901-b0e9-c065d1359428
{
```

@thomasvochten

Report

```
New-M365DSCReportFromConfiguration
```



Synchronize

```
Export-M365DSCConfiguration
Start-DSCConfiguration
```

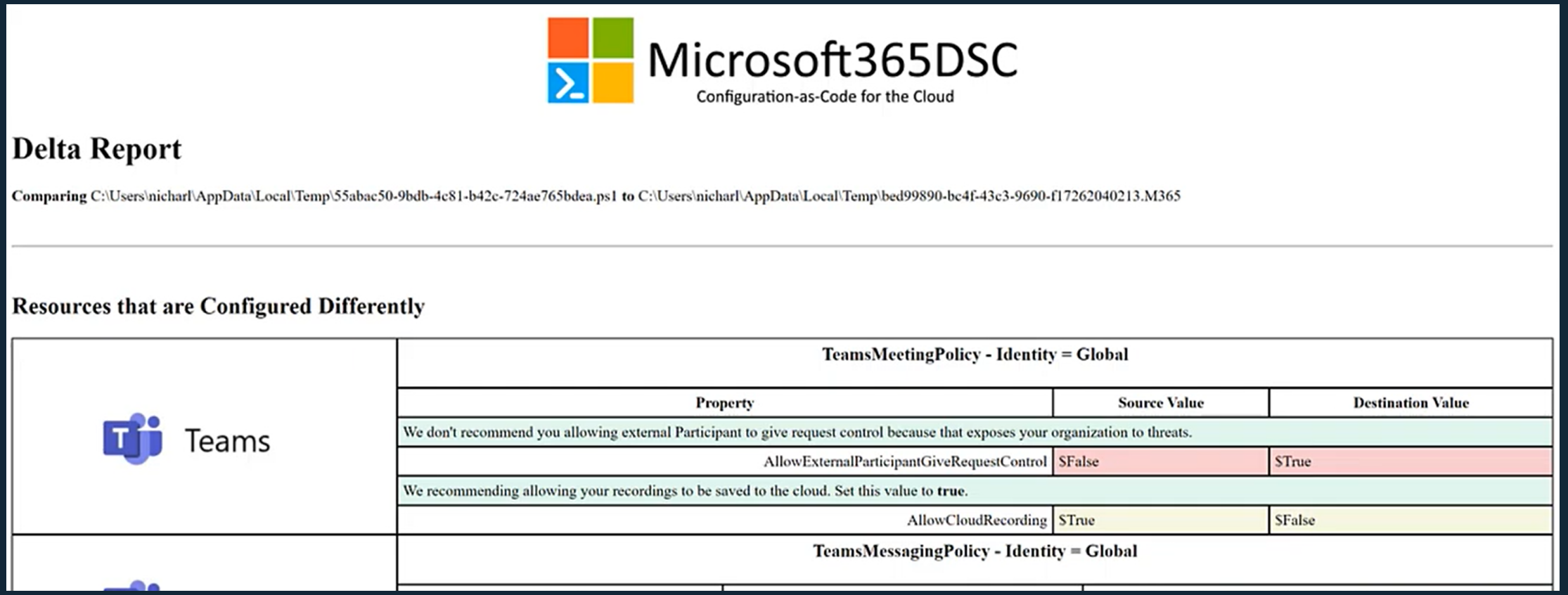
Tenant A

Tenant B

Assess / Assert

```
Assert-M365DSCBlueprint
```

Assess / Assert



Microsoft365DSC

Configuration-as-Code for the Cloud

Delta Report

Comparing C:\Users\nicharl\AppData\Local\Temp\55abac50-9bdb-4c81-b42c-724ae765bdea.ps1 **to** C:\Users\nicharl\AppData\Local\Temp\bed99890-bc4f-43c3-9690-f17262040213.M365

Resources that are Configured Differently

	TeamsMeetingPolicy - Identity = Global		
Teams	**Property**	**Source Value**	**Destination Value**
	We don't recommend you allowing external Participant to give request control because that exposes your organization to threats.		
	AllowExternalParticipantGiveRequestControl	$False	$True
	We recommending allowing your recordings to be saved to the cloud. Set this value to **true**.		
	AllowCloudRecording	$True	$False
	TeamsMessagingPolicy - Identity = Global		

@thomasvochten

Monitor

```
Test-DSCConfiguration
```

Tenant

DSC Engine

Fix / Log / Alert

Monitor

Three configuration modes:

- ApplyOnly
- ApplyAndMonitor
- ApplyAndAutoCorrect

The Challenge: Configuration Drift

- Uncontrolled changes
- Unclear processes

Authentication & Authorization

- AAD Service Principals are preferred
- Regular credentials also work
- Does not support MFA

Workload	PowerShell Module	Credential	Service Principal			Managed Identity
			Certificate Thumbprint	Certificate Path	Application Secret	
*AzureAD**	Microsoft.Graph.Authentication (Connect-MgGraph)	✓	✓	✗	✓	✓
Exchange Online	ExchangeOnlineManagement (Connect-ExchangeOnline)	✓	✓	✓	✗	✓
*Intune**	Microsoft.Graph.Authentication (Connect-MgGraph)	✓	✓	✗	✓	✓
*Office 365**	Microsoft.Graph.Authentication (Connect-MgGraph)	✓	✓	✗	✓	✓
OneDrive	PnP.PowerShell (Connect-PnPOnline)	✓	✓	✓	✓	✓
Power Apps	Microsoft.PowerApps. Administration.PowerShell	✓	✓	✗	✓	✗
*Planner**	Microsoft.Graph.Authentication (Connect-MgGraph)	✓	✓	✗	✓	✓
Security & Compliance Center	ExchangeOnlineManagement (Connect-IPPSSession)	✓	✓	✓	✗	✗
SharePoint Online	PnP.PowerShell (Connect-PnPOnline)	✓	✓	✓	✓	✓
Teams	MicrosoftTeams (Connect-MicrosoftTeams)	✓	✓	✗	✗	✗

Installation & Prerequisites

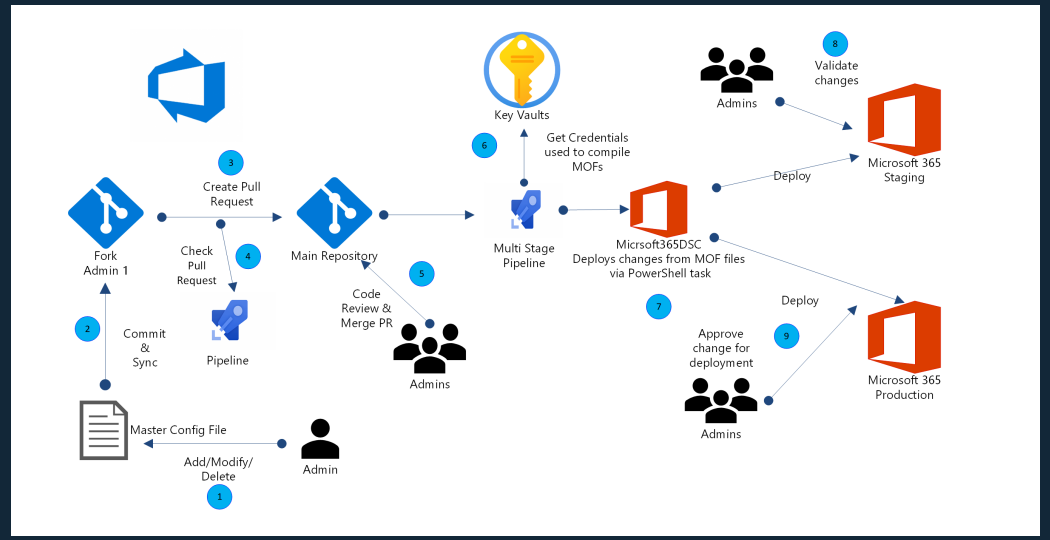
```
Install-Module Microsoft365DSC
Update-M365DSCDependencies
```

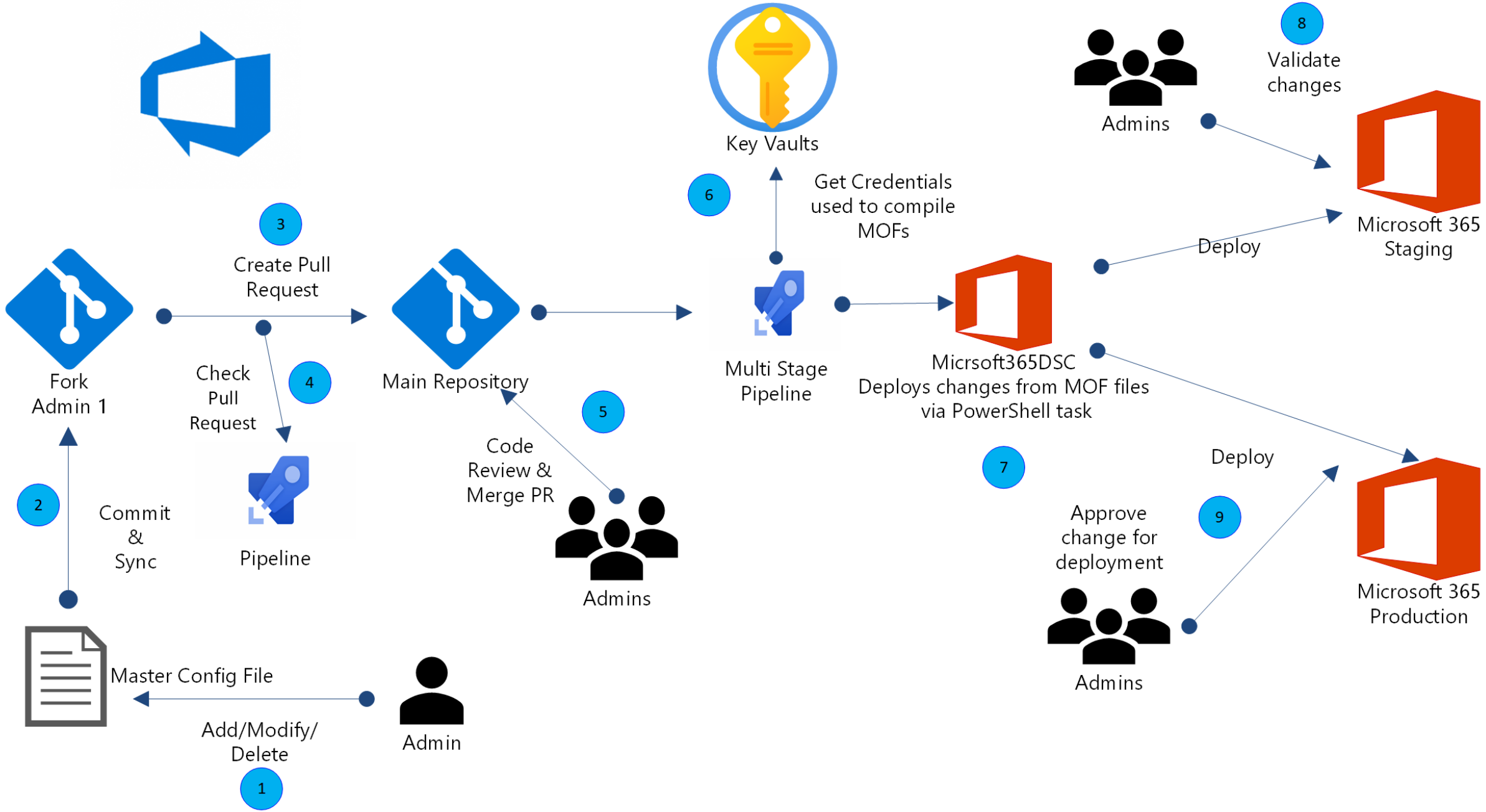
The holy grail: Automation

- Consistency is key
- Controllable process (approvals & quality gates)
- More deliberate change management
- Helps with configuration drift

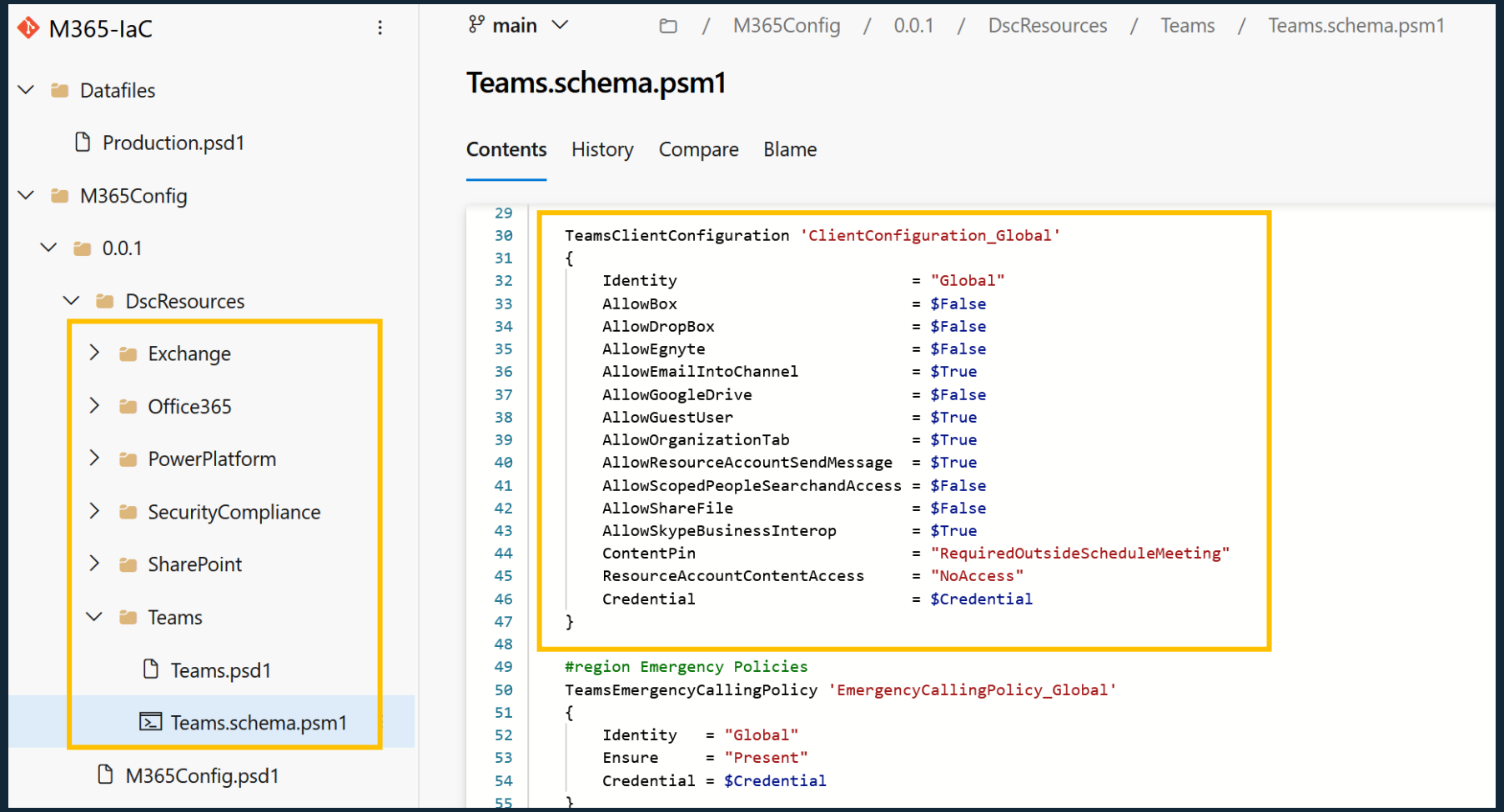
Using Azure DevOps to manage your M365 config

- Maintain your config in git
- Create a Pull Request to integrate your changes
- Config gets deployed to one or more tenants




Key Vaults
Get Credentials used to compile MOFs
6
Admins
8
Validate changes
Microsoft 365 Staging
Deploy
3
Create Pull Request
Fork
Admin 1
Main Repository
Multi Stage Pipeline
Micrsoft365DSC
Deploys changes from MOF files via PowerShell task
Check Pull Request
4
5
Code Review & Merge PR
Admins
Pipeline
7
Deploy
Approve change for deployment
9
Admins
Microsoft 365 Production
2
Commit & Sync
Master Config File
Admin
Add/Modify/ Delete
1

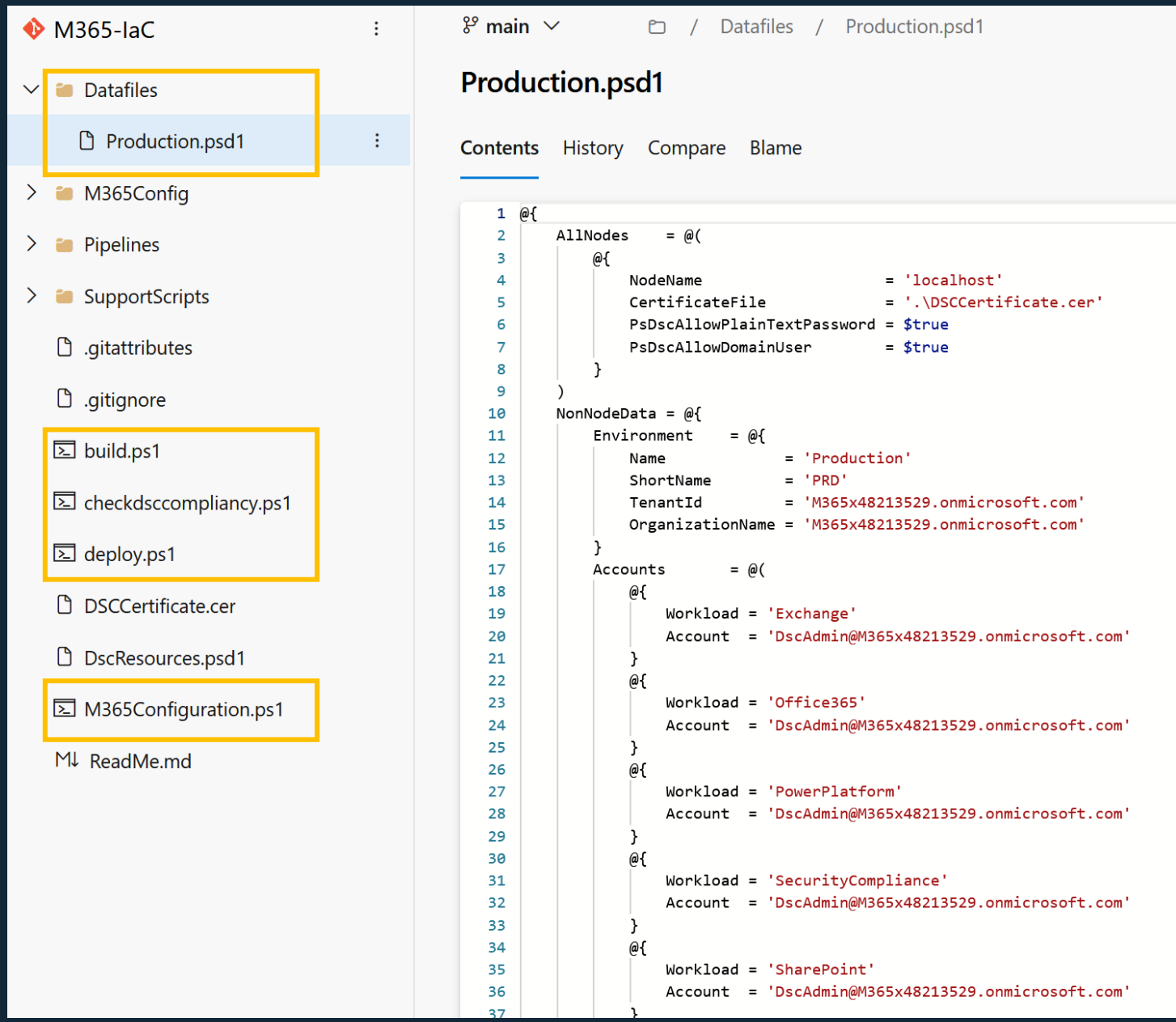
Settings


M365-IaC
Datafiles
Production.psd1
M365Config
0.0.1
DscResources
Exchange
Office365
PowerPlatform
SecurityCompliance
SharePoint
Teams
Teams.psd1
Teams.schema.psm1
M365Config.psd1
main
/ M365Config / 0.0.1 / DscResources / Teams / Teams.schema.psm1
Teams.schema.psm1
Contents History Compare Blame
TeamsClientConfiguration 'ClientConfiguration_Global'
{
Identity = "Global"
AllowBox = $False
AllowDropBox = $False
AllowEgnyte = $False
AllowEmailIntoChannel = $True
AllowGoogleDrive = $False
AllowGuestUser = $True
AllowOrganizationTab = $True
AllowResourceAccountSendMessage = $True
AllowScopedPeopleSearchandAccess = $False
AllowShareFile = $False
AllowSkypeBusinessInterop = $True
ContentPin = "RequiredOutsideScheduleMeeting"
ResourceAccountContentAccess = "NoAccess"
Credential = $Credential
}
#region Emergency Policies
TeamsEmergencyCallingPolicy 'EmergencyCallingPolicy_Global'
{
Identity = "Global"
Ensure = "Present"
Credential = $Credential

Configuration

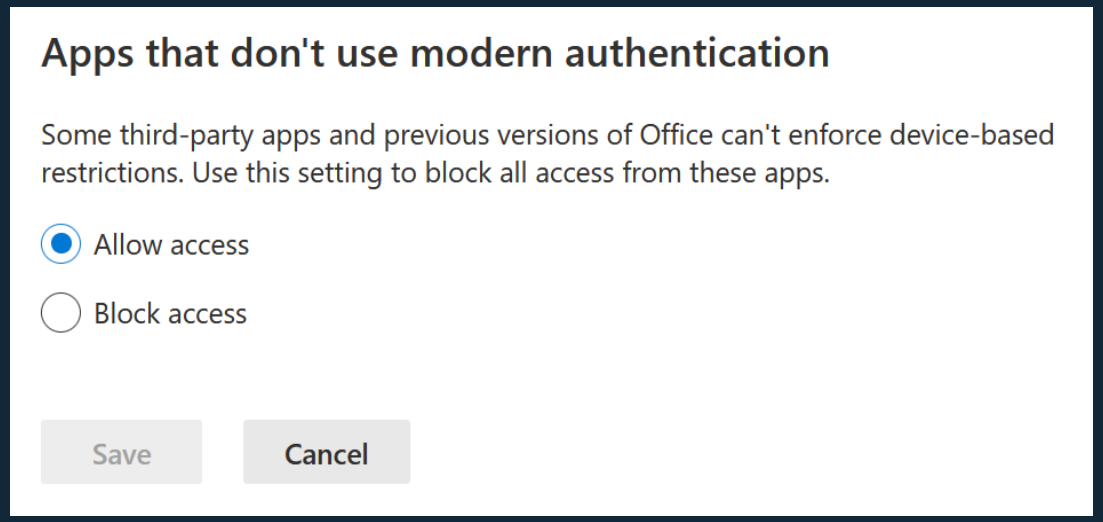
PowerShell to configure

- Environments/tenants
- Pipeline execution scripts
- Configuration scripts

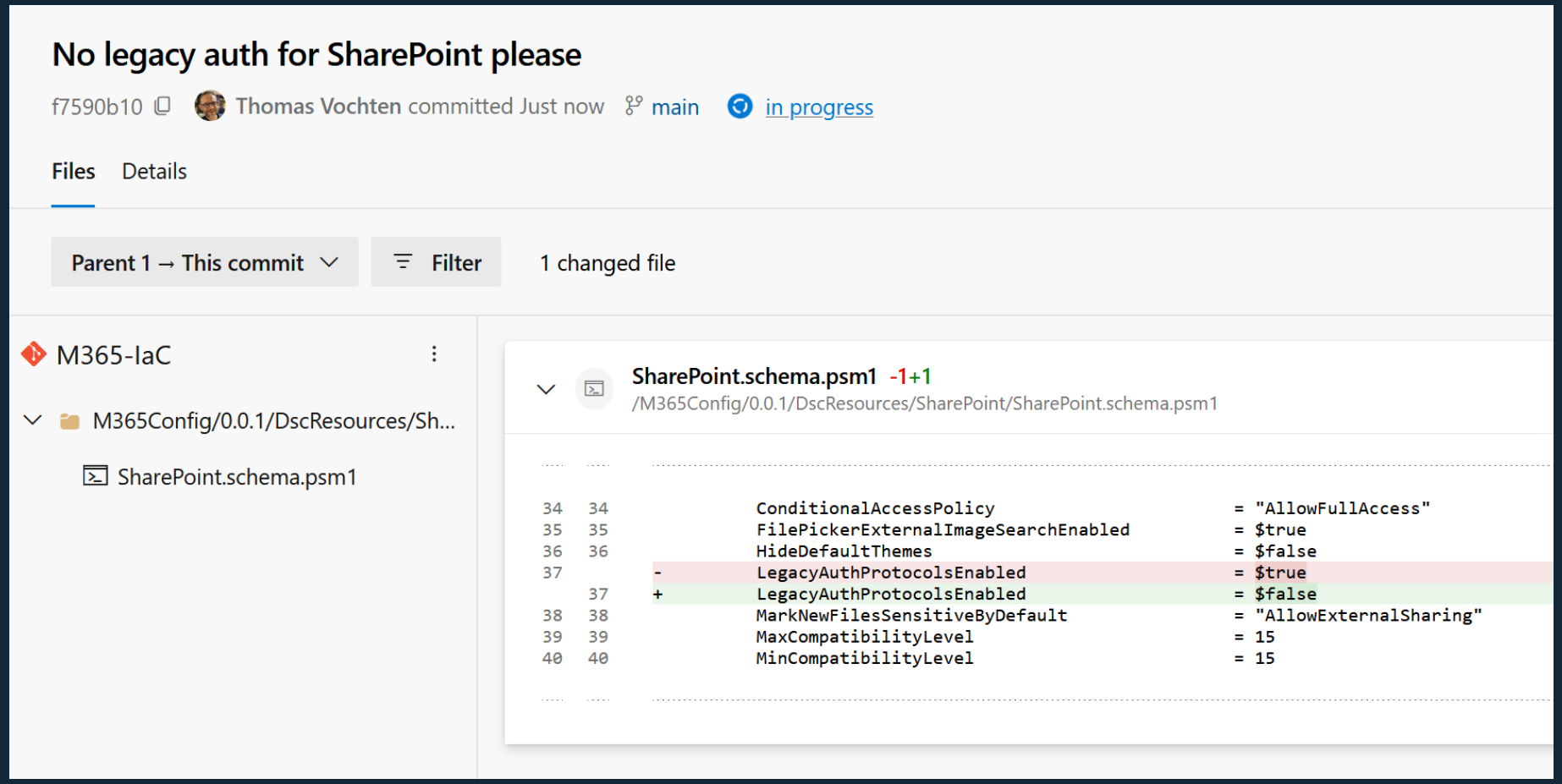


Applying

Before



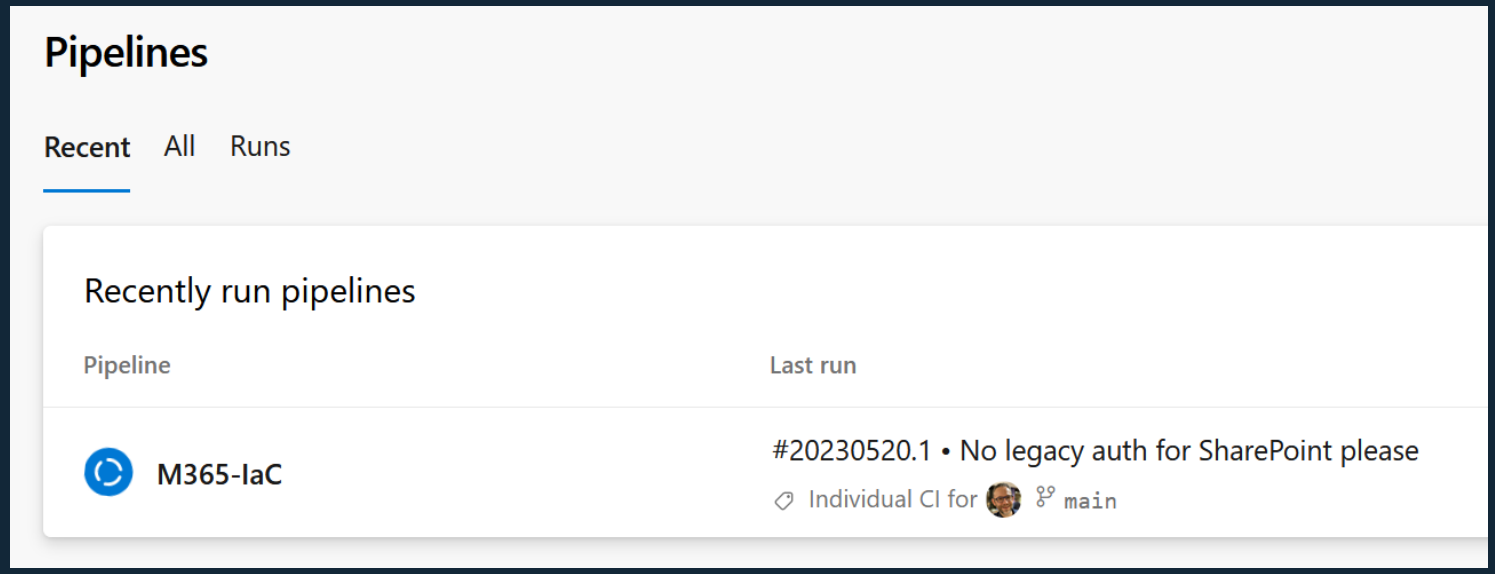
Applying



@thomasvochten

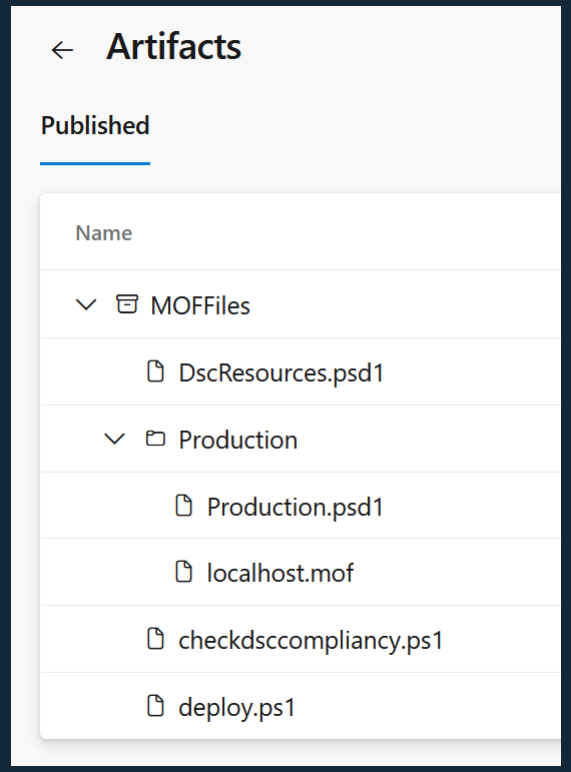
Applying

Change has been detected and build pipeline starts



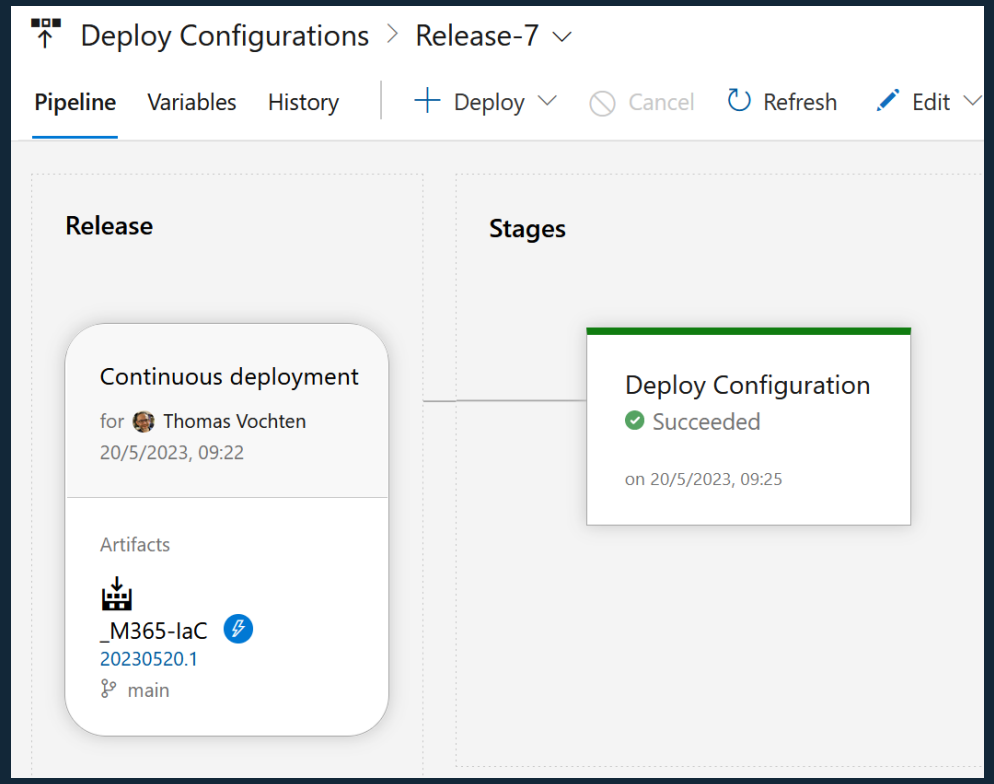
Applying

Build generates the DSC MOF files



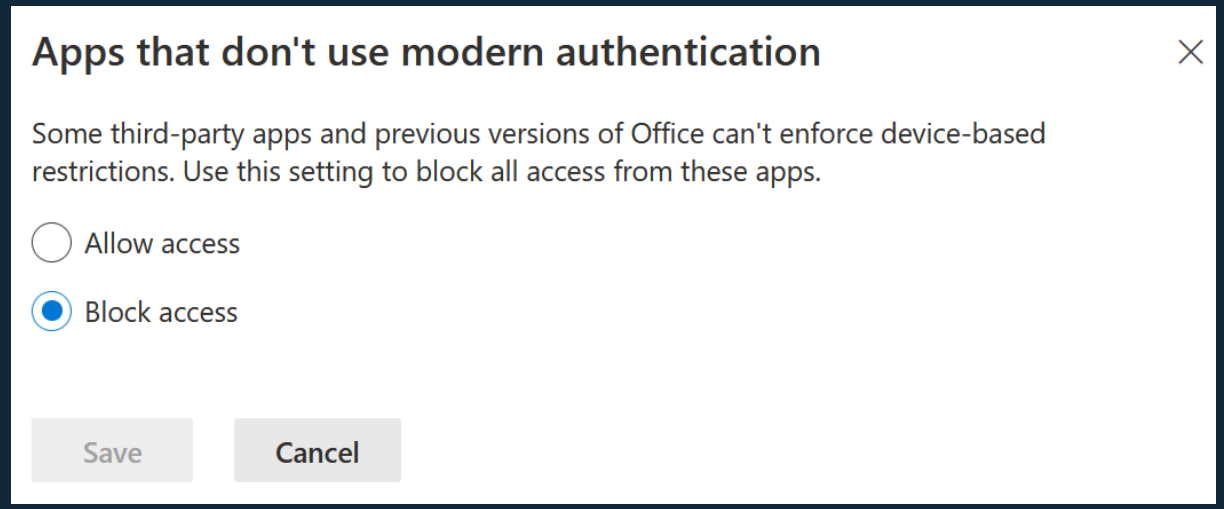
Applying

Release pipeline kicks off and applies the change



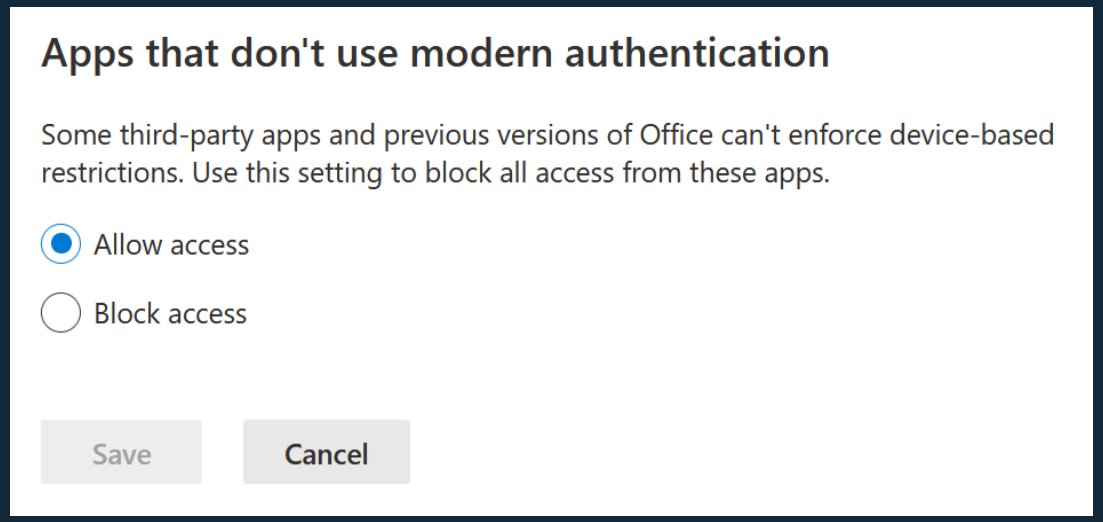
Applying

After



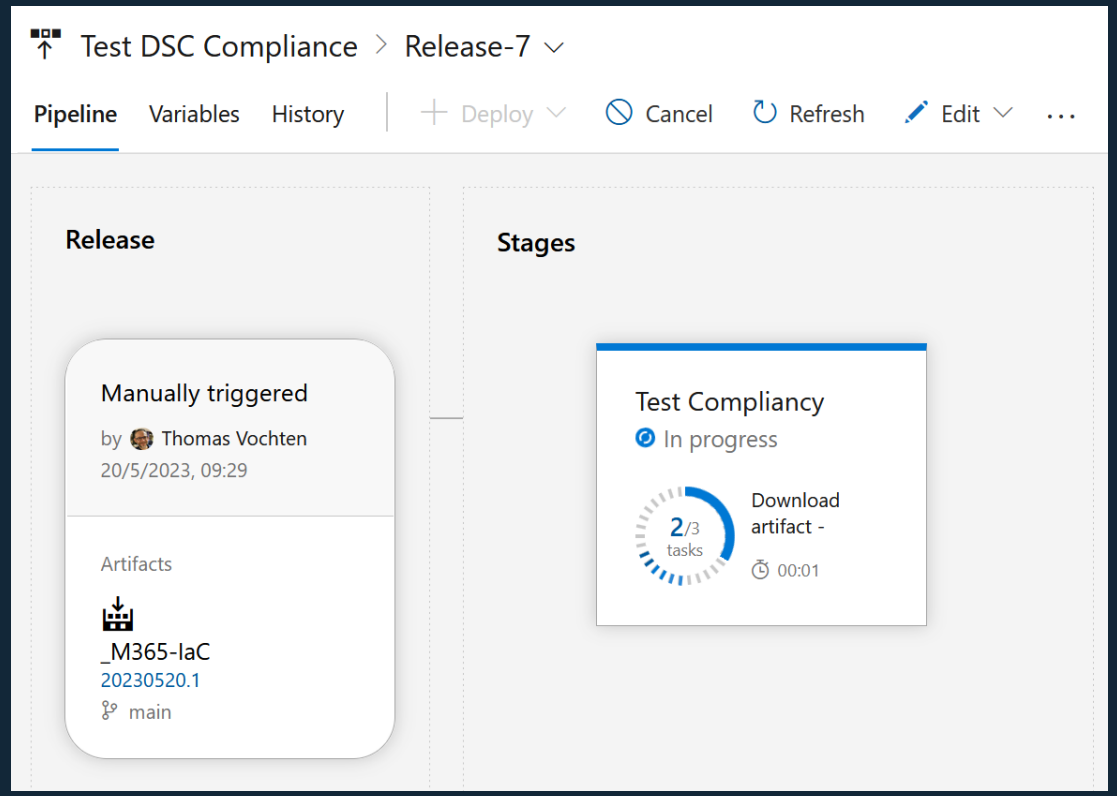
Auditing

Wait a minute…



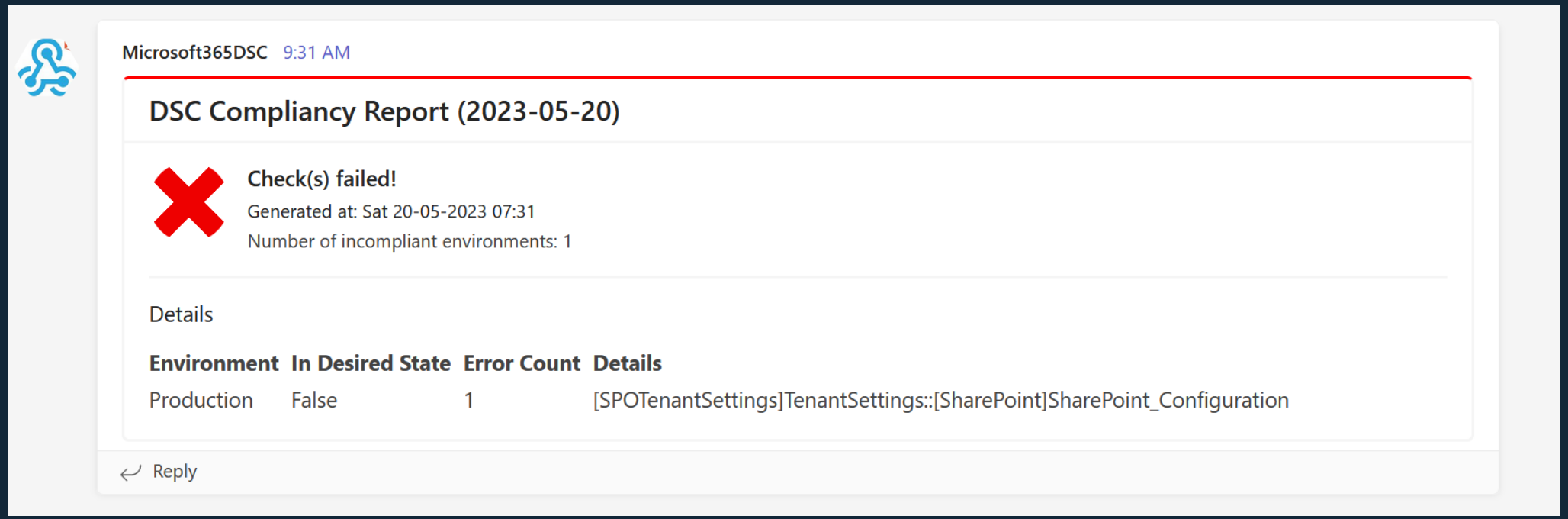
Auditing

Compliancy job is running (scheduled)



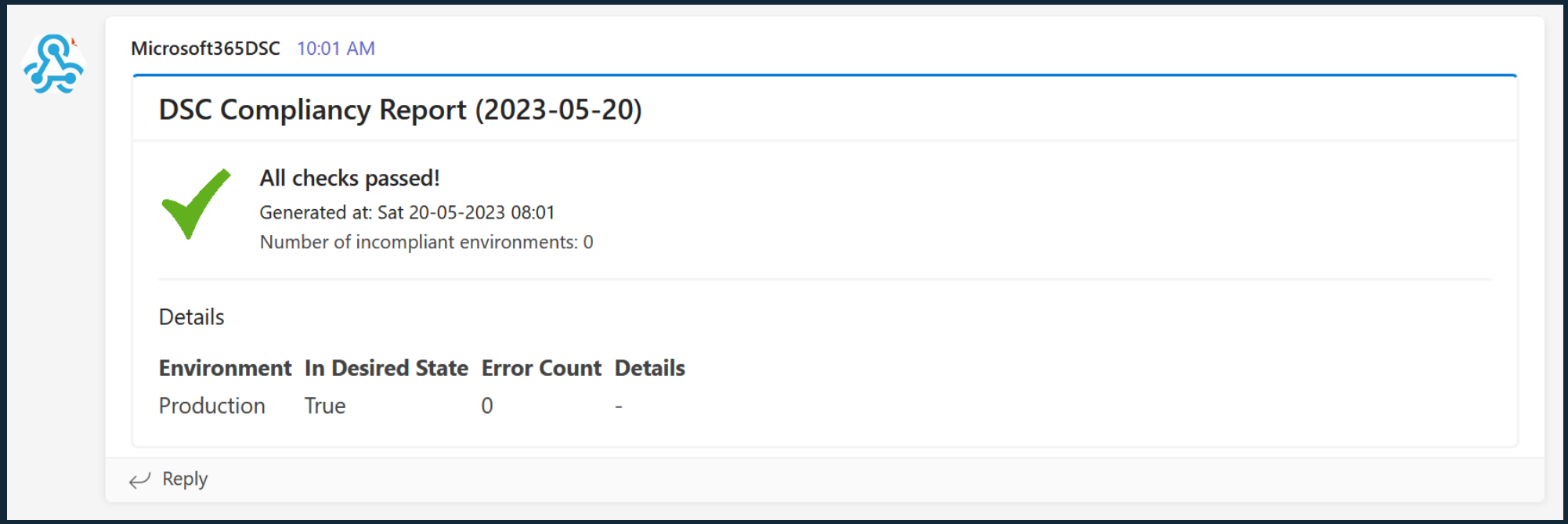
@thomasvochten

Auditing



@thomasvochten

Auditing



@thomasvochten

Solution Components

- Git repository
- Azure KeyVault
- Azure DevOps Build & Release Pipelines
- Azure DevOps (Self) Hosted Agents
- PowerShell
- Microsoft 365 DSC

But… why?

- Enforce governance policies & avoiding configuration drift
- When you need strict change management
- If you want to maintain the same baseline across tenants
- Development / Test / Acceptance / Production
- …

Key takeaways & advice

- Just because you can, doesn’t mean you should
- Get familiar with PowerShell DSC if you aren’t already
- Testing is key. Be careful with (auto)applying changes
- Using proper source control is a must
- Automation is cool, but optional (and complicated)

Key resources

- https://microsoft365dsc.com
- https://github.com/microsoft/microsoft365DSC
- https://microsoft365dsc.com/about/community-resources
- https://github.com/microsoft/ConditionalAccessforZeroTrustResources

Thank you

@thomasvochten

https://thomasvochten.com

mail@thomasvochten.com