Erasmus MC
Universitair Medisch Centrum Rotterdam

Tutorial geautomatiseerde certificaatvernieuwing in Azure

Marco Boom
Erasmus MC
30 november 2023

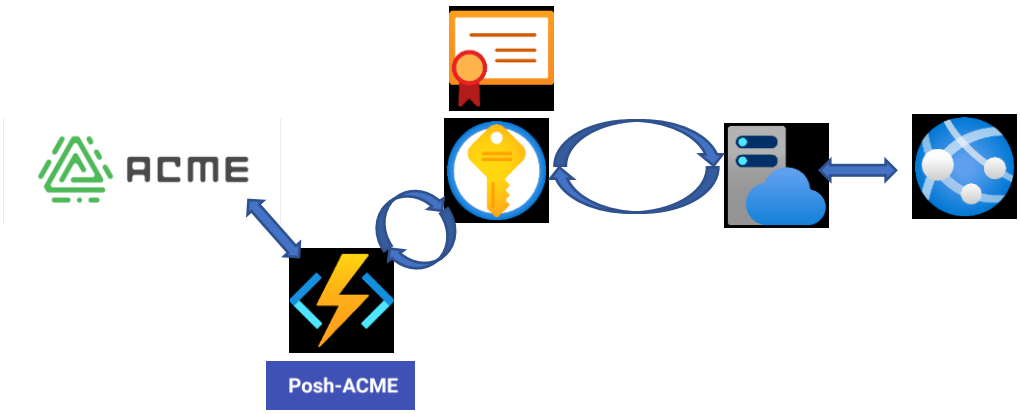
Inhoudsopgave

Inleiding

Dit document beschrijft een manier om geautomatiseerd certificaten te vernieuwen voor diensten in Azure. Deze tutorial richt zich specifiek op een oplossing voor Azure App Services, maar kan ook gebruikt worden voor andere diensten die gebruik maken van een Azure Key Vault. Verder zal in deze tutorial gebruik worden gemaakt van Azure Functions, Azure Storage (Azure Blobs), PosH-ACME en de Sectigo Certificate Manager REST API.

De oplossing ziet er in vogelvlucht als volgt uit:

- In de Certificate Manager wordt een ACME account aangemaakt.
- Certificaten zullen opgeslagen en beheerd worden in een Azure Key Vault.
- Een Azure Function App wordt periodiek uitgevoerd en controleert op en vervangt verlopende certificaten in de Key Vault.
- Een Azure Web App importeert een certificaat uit de Key Vault.



De tutorial begint met het aanmaken van een ACME account bij Sectigo. Daarna maak je een Key Vault en een Function App aan. In de Function App gebruik je PoSH-ACME om certificaten aan te vragen. Tot slot maak je een Web App aan die het certificaat gaat gebruiken. Iedere stap voor de opbouw wordt voornamelijk via command line interfaces doorlopen met af en toe een check-up in grafische gebruikersinterfaces.

Het doel van deze tutorial is vertrouwd raken met geautomatiseerde certificaatvernieuwing in Azure en inzicht te krijgen in geautomatiseerd opspinnen van infrastructuur. Er zijn meerdere manieren mogelijk dan wat in deze tutorial beschreven wordt. Voel je vooral vrij om zelf te experimenteren met verschillende opties en eigen ideeën.

1 ACME account voorbereiden

In dit onderdeel maak je een ACME account aan wat door Azure gebruikt gaat worden voor het aanvragen en vernieuwen van certificaten. Zorg er voor dat je een geregistreerd domeinnaam hebt in Sectigo Certificate Manager.

Voor deze tutorial maak je een subdomeinnaam aan, maar je kan ieder gevalideerd domeinnaam gebruiken. Vervolgens maak je een ACME account aan en koppel je het domeinnaam aan het ACME account.

1.1 Aanmaken subdomein

Voor het aanmaken van een subdomein maak je gebruik van de Certificate Manager REST API. Daarvoor heb je een header nodig die aangeeft welke klant je bent, in welk format je data terug wilt krijgen en met welk account je wilt inloggen. Voor deze tutorial is voor customer uri surfnet gekozen en voor content type application/json.

```
PS >

$Headers_CM = New-Object 'System.Collections.Generic.Dictionary[[String],[String]]'
$Headers_CM.Add('customerUri', 'surfnet')
$Headers_CM.Add('Content-Type', 'application/json')

$CMUser = Read-Host Certificate Manager Username
$CMPassword = Read-Host Certificate Manager Password

$Headers_CM.Add('login', "$CMUser")
$Headers_CM.Add('password', "$CMPassword")
```

Nu kun je met de header API requests versturen naar de Certificate Manager API. Probeer onderstaand commando uit te voeren en bekijk de output die je terug krijgt.

```
PS >

irm 'https://cert-manager.com/api/organization/v1' -Headers $Headers_CM -Method 'GET'
```

> irm is een alias van Invoke-RestMethod. Door een alias te gebruiken kun je een script compacter en vaak beter leesbaarder maken. Als je een alias wilt zoeken voor een veelgebruikt commando kun je het commando Get-Alias gebruiken. Voorbeeld:
>
> ```
> Get-Alias -Definition Invoke-RestMethod
> ```
>
> Als je wilt weten waar een alias voor staat, gebruik dan de alias als parameter: Get-Alias irm.

Voor het aanmaken van een subdomein heb je de organisation id nodig wat aan het subdomein

gekoppeld moet worden.

PS >

$CMOrganisationID = (irm 'https://cert-manager.com/api/organization/v1' -Headers $Headers_CM -Method 'GET').id

Nu je een header en organisation ID hebt, kun je een subdomeinnaam aanmaken. In het request geef je de naam, een beschrijving en de organisation ID op. Je kiest ook de certificaattypen die voor het domeinnaam gebruikt kunnen worden. Er kan gekozen worden voor SSL, Client of Code Signing certificaten. Voor webverkeer heb je alleen SSL nodig.

PS >

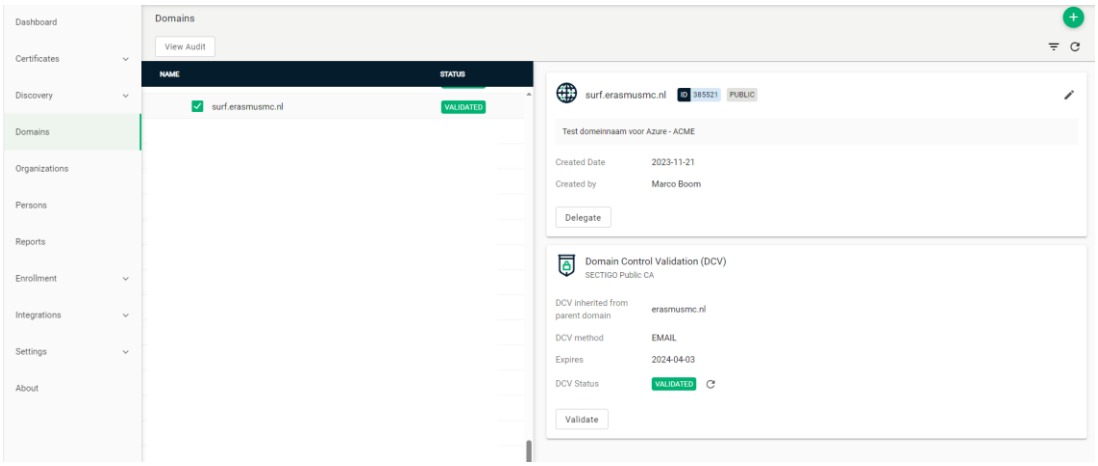
$DomainName = Read-Host domeinnaam
$Description = Read-Host beschrijving
$Body = "{ `"name`": `"$DomainName`", `"description`": `"$Description`", `"active`": true, `"delegations`": [{ `"orgId`": $CMOrganisationID, `"certTypes`": [`"SSL`"] }] }"

Bekijk eventueel de inhoud van de $Body variabele door $Body + Enter uit te voeren. Je krijgt dan een JSON formatted string terug. Voer daarna een POST request uit om het domeinnaam in Certificate Manager aan te maken.

PS >

irm 'https://cert-manager.com/api/domain/v1' -Headers $Headers_CM -Method 'POST' -Body $Body

Als dit commando goed is uitgevoerd krijg je een 201 status code terug. Tevens zul je in Certificate Manager het nieuwe domeinnaam zien.



1.2 Aanmaken ACME account

Voor het aanmaken van een ACME account kun je de headers en organisation ID uit stap 1.2 hergebruiken.

Voor het ACME account moet je een naam, organisation ID en ACME server gebruiken. De ACME server is afhankelijk van het contract dat SURF heeft afgesloten. Standaard wordt Organisation Validation gebruikt en is de CA GEANT. De ACME server die daarbij hoort is https://acme.sectigo.com/v2/GEANTOV.

Voor het volgende commando uit om een ACME account aan te maken.

```
PS >

$AccountName = Read-Host ACME account name

$Body = "{ `"name`": `"$AccountName`", `"acmeServer`":
`"https://acme.sectigo.com/v2/GEANTOV`", `"organizationId`": $CMOrganisationID }"

irm 'https://cert-manager.com/api/acme/v2/account' -Headers $Headers_CM -Method 'POST' -Body
$Body

$ACMEAccountID = (irm "https://cert-
manager.com/api/acme/v2/account?organizationId=$CMOrganisationID&name=$AccountName" -
Headers $Headers_CM -Method 'GET').id | Sort -Desc | Select -First 1
```

Let op! Het is mogelijk om meerdere ACME accounts aan te maken met dezelfde naam. Daarom wordt bij ACMEAccountID gesorteerd op aflopend ID en wordt het meest recente account geselecteerd.

Haal via het volgende commando de acmeServer, macId en macKey op. Deze gegevens zijn nodig voor het toepassen van External Account Binding.

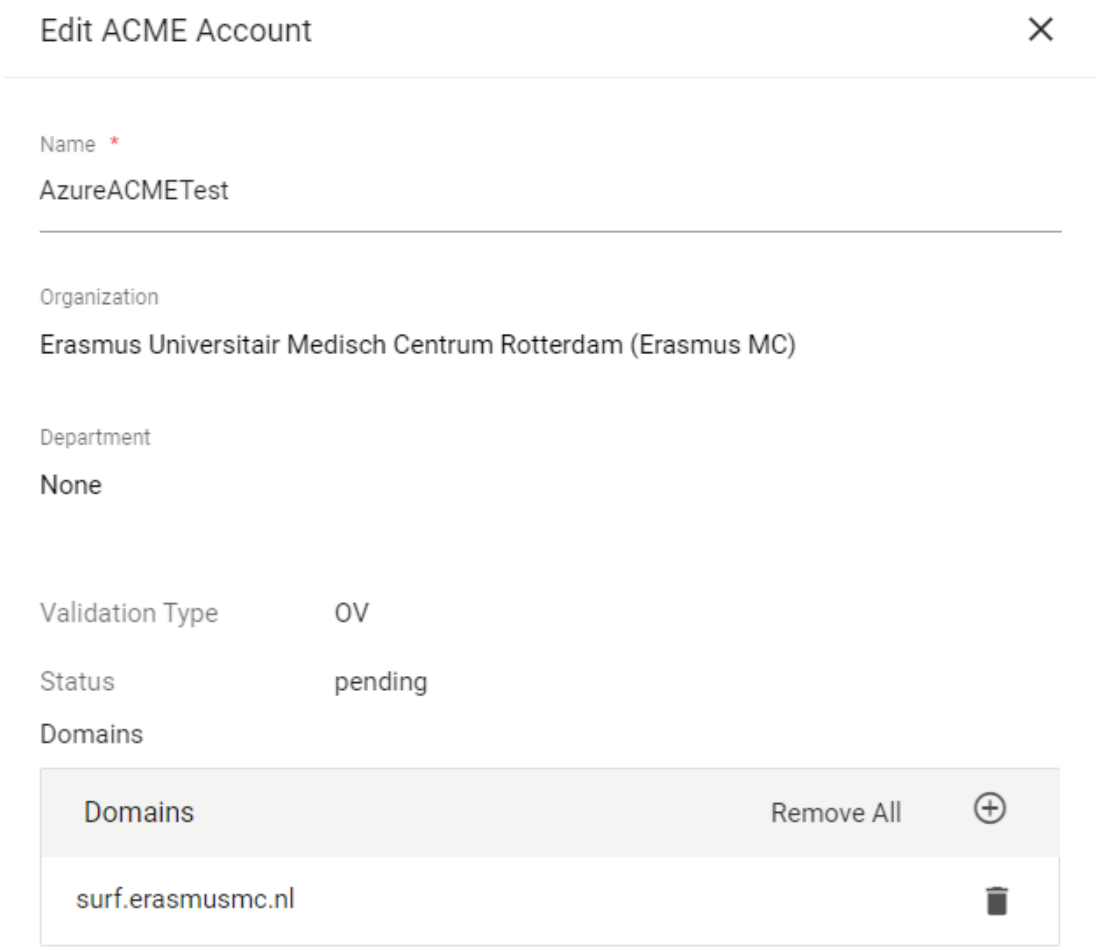
```
PS >

$ACMEAccount = irm "https://cert-manager.com/api/acme/v2/account/$ACMEAccountID" -Headers
$Headers_CM -Method 'GET'
```

Koppel tot slot alle domeinnamen waar je via het ACME account certificaten voor wilt aanvragen.

```
PS >

$Body = "{ `"domains`": [ {`"name`": `"$DomainName`"} ] }"

irm "https://cert-manager.com/api/acme/v2/account/$ACMEAccountID/domain" -Headers
$Headers_CM -Method 'POST' -Body $Body
```

Via de Certificate Manager web interface kun je nu je nieuw aangemaakt ACME account

bekijken. Ga in het menu naar Enrollment -> ACME. Kies voor https://acme.sectigo.com/v2/GEANTOV en klik op Accounts. Kies je nieuwe ACME account en klik op Edit. Je krijgt dan onderstaand scherm te zien inclusief de door jou gekoppelde domeinnamen.



2 Azure Key Vault

In dit onderdeel maak je een Azure Key Vault aan. In de Key Vault zullen certificaten en ACME account gegevens worden opgeslagen. Voordat je de Key Vault en andere resources zelf kan maken heb je eerst de Azure PowerShell module en een resource groep nodig.

2.1 Azure PowerShell installeren

Controleer of de verouderde AzureRM module geïnstalleerd is. Zo ja, verwijder dan deze module.

```
PS > Get-Module -Name AzureRM -ListAvailable
```

Aangezien de Azure PowerShell module (ook wel bekend als Az) heel groot is (de module bestaat uit ongeveer 180 submodules) en niet alle submodules nodig zijn voor deze tutorial, hoef je alleen de submodules te installeren die in deze tutorial gebruikt worden.

Installeer Azure PowerShell en maak verbinding met Azure. **Let op!** Het installeren van Azure PowerShell kan ongeveer 20 minuten duren. De module Az.Websites wordt alleen voor App Service gebruikt.

```
PS >

Install-Module Az.Accounts, Az.Functions, Az.KeyVault, Az.Resources, Az.Storage, Az.Websites -Force
-Scope CurrentUser

Connect-AzAccount
```

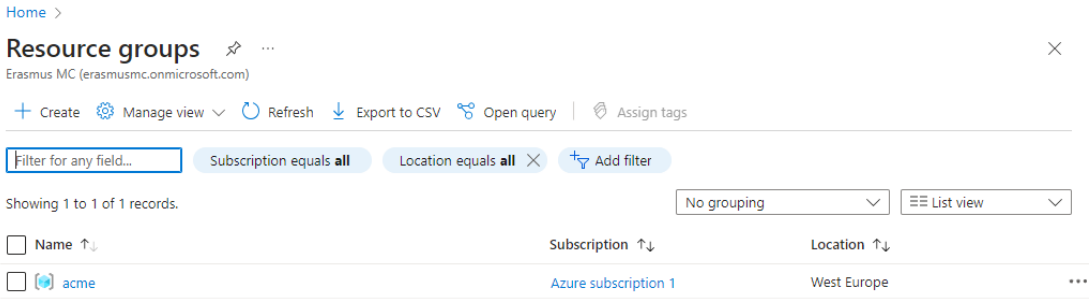
2.2 Azure Resource groep aanmaken

“Een resourcegroep is een container met verwante resources voor een Azure-oplossing. De resourcegroep kan alle resources voor de oplossing bevatten of enkel de resources die u als groep wilt beheren.” – Microsoft

Je kunt zelf kiezen welke resource groepen je gebruikt en hoe je ze indeelt. In deze tutorial wordt 1 resource groep gebruikt met de naam acme. Maak met onderstaand commando een nieuwe resource groep aan. Met de locatie kun je kiezen in welke Azure-regio je resources aangemaakt moeten worden. Een overzicht van regio’s vind je op https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/#geographies. De regio West-Europa is gevestigd in Nederland.

```
PS > New-AzResourceGroup -Name 'acme' -Location 'West Europe'
```

Je ziet na het uitvoeren van dit commando de resource groep in Azure verschijnen. In deze tutorial gebruik je deze resource groep voor alle nieuwe resources de je gaat aanmaken.



2.3 Key Vault aanmaken

Maak nu een Key Vault aan. Kies een naam, bijvoorbeeld ACMEVault, stel de resource groep en locatie in en schakel RBAC Authorization in. Momenteel gebruikt Key Vault standaard het verouderde Key Vault-toegangsbeleid. Inmiddels is RBAC het aanbevolen autorisatiesysteem voor Key Vaults. Daarom is het advies om voor nieuwe vaults RBAC in te schakelen.

Stel vervolgens met een Role Assignment je eigen account in als Key Vault Administrator.

Sla tot slot het HMAC Key ID en HMAC Key Secret op in de vault. De HMAC Key ID sla je op in een Key Vault Secret met de naam MacId. De HMAC Key Secret sla je op in een Key Vault Secret met de naam MacKey.

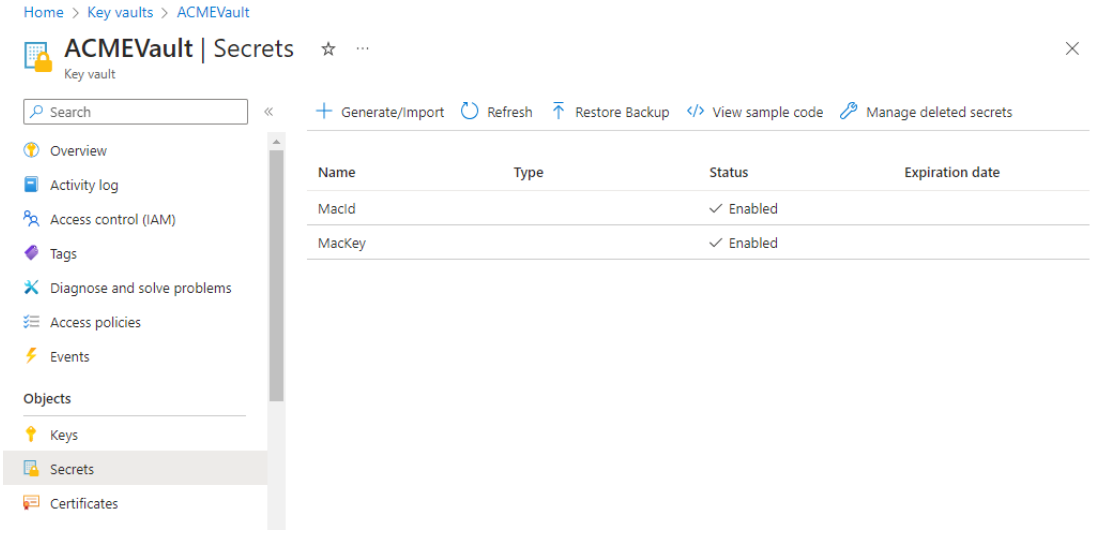
PS >

$KeyVault = New-AzKeyVault -Name 'ACMEVault' -ResourceGroupName 'acme' -Location 'West Europe' -EnableRbacAuthorization

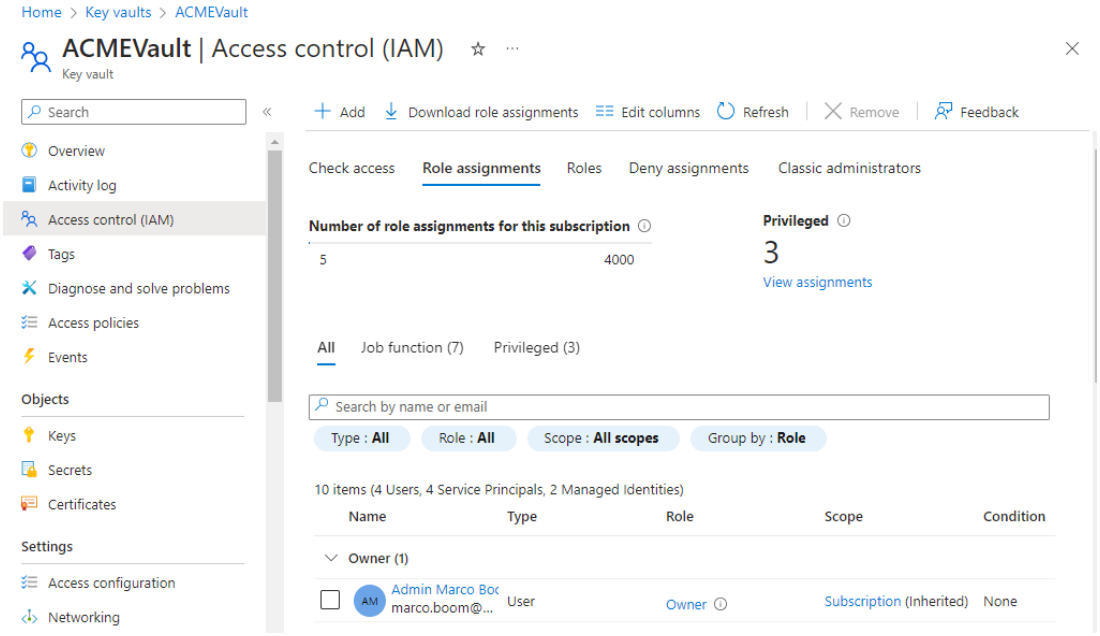
New-AzRoleAssignment -SignInName (Get-AzContext).Account.Id -RoleDefinitionName 'Key Vault Administrator' -Scope $KeyVault.ResourceId

$MacId = ConvertTo-SecureString $ACMEAccount.macId -AsPlainText –Force
$MacKey = ConvertTo-SecureString $ACMEAccount.macKey -AsPlainText –Force
Set-AzKeyVaultSecret -VaultName 'ACMEVault' -Name 'MacId' -SecretValue $MacId
Set-AzKeyVaultSecret -VaultName 'ACMEVault' -Name 'MacKey' -SecretValue $MacKey

Na het uitvoeren van deze commando’s kun je de nieuwe Key Vault met de ACME secrets in het Azure portal terugvinden.



Via het tabblad Access Control (IAM) -> Role assignments kun je alle toegewezen RBAC toewijzingen bekijken.



3 Azure Function App

In dit onderdeel maak je een Azure Function App aan. In de Function App zal een PowerShell script geïmplementeerd worden dat met behulp van Posh-ACME certificaten kan vernieuwen. Met behulp van een timer trigger wordt de functie periodiek uitgevoerd. Op deze manier kunnen certificaten geautomatiseerd periodiek vernieuwd worden.

3.1 Function App aanmaken

Voor het aanmaken van een Function App is Storage nodig. De functie zal deze storage gebruiken voor bijvoorbeeld tijdelijke bestanden en configuratiebestanden. Registreer Microsoft.Storage als een Resource Provider. Maak vervolgens een Storage Account aan.

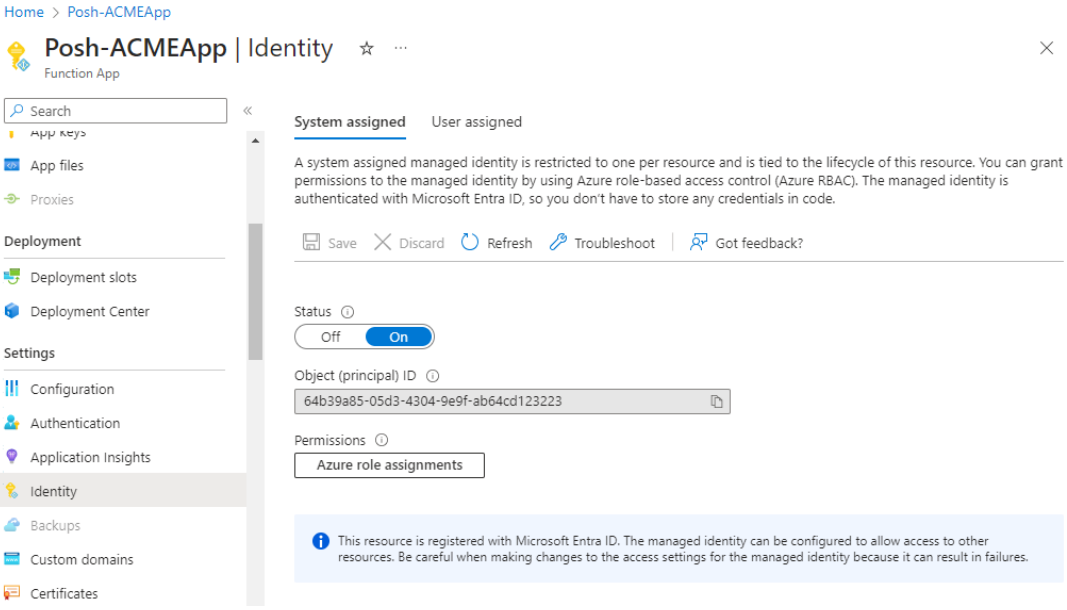
Voor deze tutorial maak je gebruik van de SKU Standard_LRS. De SKU bepaalt welke dienstverlening geleverd wordt met betrekking tot o.a. redundancy. Standard_LRS is Standard Locally Redundant Storage. Bekijk een overzicht van alle SKU's op https://learn.microsoft.com/en-us/rest/api/storagerp/srp_sku_types.

Maak vervolgens een Function App aan met de naam Posh-ACMEApp en runtime PowerShell. Schakel ook de System Assigned Identity in. Hiermee wordt een security principal voor de app aangemaakt zodat de app rechten kan krijgen op andere resources.

Stel de Function App vervolgens in als Key Vault Certificates Officer en Key Vault Secrets User op de ACME Key Vault zodat de app rechten krijgt om certificaten in de vault te beheren en de HMAC secrets in de vault kan uitlezen.

```
PS >

Register-AzResourceProvider -ProviderNamespace Microsoft.Storage

New-AzStorageAccount -ResourceGroupName 'acme' -Name 'poshacmestore' -SkuName
'Standard_LRS' -Location 'West Europe' -AllowBlobPublicAccess $false

$PoshACMEApp = New-AzFunctionApp -Name 'Posh-ACMEApp' -ResourceGroupName 'acme' -
StorageAccount 'poshacmestore' -Runtime PowerShell -FunctionsVersion 4 -Location 'West Europe' -
IdentityType SystemAssigned

New-AzRoleAssignment -ObjectId $PoshACMEApp.IdentityPrincipalId -RoleDefinitionName 'Key Vault
Certificates Officer' -Scope $KeyVault.ResourceId

New-AzRoleAssignment -ObjectId $PoshACMEApp.IdentityPrincipalId -RoleDefinitionName 'Key Vault
Secrets User' -Scope $KeyVault.ResourceId
```

Nadat je de commando's hebt uitgevoerd, kun je de Function App terugvinden in het Azure portal. Onder het tabblad Identity zie je dat de System assigned identity ingeschakeld is.



3.2 Lokaal Function project aanmaken

Voor het gebruiken van een Azure Function App, moet je een lokaal Function project aanmaken. Hier heb je de Azure Functions Core Tools voor nodig. Download de Azure Functions Core Tools msi via https://github.com/Azure/azure-functions-core-tools/blob/v4.x/README.md#windows. Installeer de tools via een PowerShell venster met verhoogde rechten:

```
PS > msiexec.exe /package ".\func-cli-x64.msi" /qn /norestart
```

Maak een function project en een function aan in een PowerShell venster met standaardrechten. Het project heet LocalAcmeFunctionProj en de functie heet ReqCert. Kies eventueel eigen namen naar keus.

```
PS >

func init LocalAcmeFunctionProj --powershell

cd LocalAcmeFunctionProj

func new --name ReqCert --template 'Timer trigger'
```

Voeg de PowerShell modules Posh-ACME en Az.KeyVault als requirement toe aan het Function project.

PS >

```
$Content = Get-Content .\requirements.psd1
$InsertPosition = ($Content | Select-String '}').LineNumber – 2
$Content[$InsertPosition] += "`n    'Posh-ACME' = '4.*'`n    'Az.KeyVault' = '5.*'"
$Content | Set-Content .\requirements.psd1
```

Stel de Function Trigger in op 0 0 0 * * 0 (Cron taal voor 00:00 op zondag, elke week)
PS >

```
$Trigger = Get-Content '.\ReqCert\function.json' | ConvertFrom-Json
$Trigger.bindings[0].schedule = '0 0 0 * * 0 '
$Trigger | ConvertTo-Json | Set-Content '.\ReqCert\function.json'
```

Definieer vervolgens de inhoud van de Function App. De uitvoerbare code van de functie wordt geplaatst in .\ReqCert\run.ps1. De functie doorloopt de volgende stappen:

1. De functie controleert of de ACME server al ingesteld is. Zo niet, dan wordt de ACME server geconfigureerd.
2. De functie controleert of lokaal een ACME client account ingesteld is bij de gekozen ACME server. Indien er lokaal geen ACME client account bestaat, zal de functie het account lokaal aanmaken. Hierbij wordt External Account Binding gebruikt om de client te koppelen aan het eerder aangemaakte ACME account. Er moet een e-mailadres voor de client gekozen worden waar notificaties over certificaten naartoe gestuurd kunnen worden.
3. De functie haalt het huidige certificaat met de naam SURFCert, indien het bestaat, op.
4. Indien het certificaat met de naam SURFCert nog niet bestaat of het certificaat gaat binnen 30 dagen verlopen, maakt de functie een nieuw certificaat aan of vervangt het huidige certificaat.
    a. Er wordt een nieuwe Certificate Policy gemaakt. In de policy wordt gedefinieerd dat het certificaat wordt uitgegeven door een externe CA met de optie -Issuer 'Unknown'. Tevens wordt gedefinieerd wat het Subject moet worden en welke cryptoalgoritmen gebruikt moeten worden.
    b. Daarna wordt een nieuw certificaatobject aangemaakt met de naam SURFCert. Indien er al een certificaatobject met die naam bestond, wordt een nieuwe versie aan het bestaande object toegevoegd.
    c. De CSR van het nieuwe certificaat object wordt in een tijdelijk bestand opgeslagen en door New-PACertificate gebruikt om een nieuw certificaat bij de CA aan te vragen.
    d. Als de aanvraag gelukt is wordt de nieuwe Full Chain geïmporteerd in de vault. Indien het (oude) certificaat door andere diensten vanuit de vault is geïmporteerd zullen die andere diensten het certificaat automatisch binnen 24 uur bijwerken.

Kopieer onderstaand blok code in zijn geheel naar je PowerShell venster en voer het uit.

```
PS >

$Email = Read-Host E-mailadres voor notificaties
$VaultName = 'ACMEVault'
$CertName = 'SURFCert'

$Function = @"
param(`$Timer)

if ((Get-PAServer -DirectoryUrl "$($ACMEAccount.acmeServer)").name.Count -lt 1) {
   Set-PAServer -DirectoryUrl "$($ACMEAccount.acmeServer)"
}

if ((Get-PAAccount -List | % { `$_.location.Contains("$($ACMEAccount.acmeServer)") }).Count -lt 1) {
   `$MacId = Get-AzKeyVaultSecret -VaultName "$VaultName" -Name 'MacId' -AsPlainText
   `$MacKey = Get-AzKeyVaultSecret -VaultName "$VaultName" -Name 'MacKey' -AsPlainText
   New-PAAccount -Contact "$Email" -ExtAcctKID `$MacId -ExtAcctHMACKey `$MacKey
}

`$CurrentCert = Get-AzKeyVaultCertificate -VaultName "$VaultName" -Name "$CertName"

if (`$CurrentCert.Count -lt 1 -or (`$CurrentCert.Expires - (Get-Date)).Days -lt 30) {
   `$Policy = New-AzKeyVaultCertificatePolicy -IssuerName 'Unknown' -SubjectName
"CN=$DomainName" -SecretContentType 'application/x-pkcs12' -KeyType 'EC' -Curve 'P-384'
   `$CertReq = Add-AzKeyVaultCertificate -VaultName "$VaultName" -Name "$CertName" -
CertificatePolicy `$Policy
   `$CSRFile = New-TemporaryFile
   "-----BEGIN NEW CERTIFICATE REQUEST-----``n" + `$CertReq.CertificateSigningRequest + "``n-----
END NEW CERTIFICATE REQUEST-----" | Set-Content `$CSRFile.VersionInfo.FileName
   `$Cert = New-PACertificate -CSRPath `$CSRFile.VersionInfo.FileName
   Import-AzKeyVaultCertificate -VaultName "$VaultName" -Name "$CertName" -FilePath
`$Cert.FullChainFile
}
"@

$Function | Set-Content '.\ReqCert\run.ps1'
```

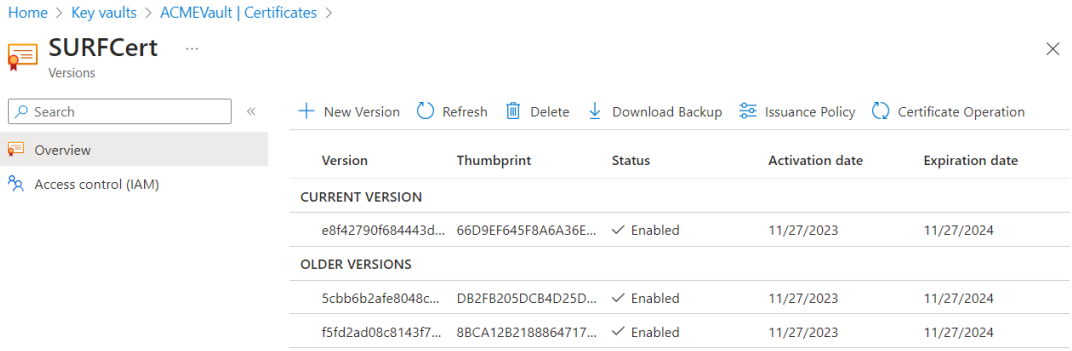
Test de functie en publiceer de Function App in Azure. Bij het testen kan een eerste certificaat al in de Key Vault geplaatst worden.

PS >

powershell.exe -file

func azure functionapp publish 'Posh-ACMEApp'

Indien de test geslaagd is, zie je een nieuw certificaat in het Azure Portal. Indien er meerdere versies van het certificaat bestaan, zullen die ook zichtbaar zijn.



4 Azure Web App

In dit onderdeel maak je een Web App aan. Deze Web App is een voorbeeld van een dienst dat een certificaat gebruikt. Je kunt behalve een Web App ook andere diensten configureren om certificaten in de Key Vault te gebruiken.

4.1 Azure Web App aanmaken

Voor het maken van een Web App heb je eerst een App Service Plan nodig. Dit plan definieert met welke resources en hoeveel resources de app uitgevoerd mag worden. Vervolgens maak je de Web App aan.

Daarna wijs je de rol Key Vault Secrets Users toe Azure App Service, zodat App Service in staat is om het certificaat uit de Key Vault te importeren naar de Web App.

PS >

New-AzAppServicePlan -ResourceGroupName 'acme' -Name 'SURFWebAppService' -Location 'West Europe' -Tier 'Standard S1' -NumberofWorkers 1 -WorkerSize 'ExtraSmall'

$WebApp = New-AzWebApp -ResourceGroupName 'acme' -Name 'SURFWebApp' -Location 'West Europe' -AppServicePlan 'SURFWebAppService'

$AzureAppServiceId = (Get-AzADServicePrincipal -DisplayName 'Microsoft Azure App Service').Id

New-AzRoleAssignment -ObjectId $AzureAppServiceId -RoleDefinitionName 'Key Vault Secrets User' -Scope $KeyVault.ResourceId

$Cert = Import-AzWebAppKeyVaultCertificate -ResourceGroupName 'acme' -WebAppName 'SURFWebApp' -KeyVaultName 'ACMEVault' -CertName 'SURFCert'

$WebApp.HostNames
$WebApp.CustomDomainVerificationId

Maak een DNS CNAME record aan dat verwijst naar het oorspronkelijke domeinnaam van de webapp. Het oorspronkelijke domeinnaam wordt met het HostNames commando zichtbaar gemaakt. In het voorbeeld ziet het CNAME record er als volgt uit: surf.erasmusmc.nl CNAME surfwebapp.azurewebsites.net.

Maak vervolgens een DNS TXT record aan met de Custom Domain Verification ID. Het TXT record moet de naam asuid hebben en een subdomeinnaam zijn van het custom domeinnaam. In het voorbeeld ziet het TXT record er als volgt uit: asuid.surf.erasmusmc.nl TXT 44D45676AB4C3A462F7DD8971710306C4D5D1C6901264A5B79456402230B2165.

Voeg daarna het domeinnaam aan de app toe en maak een SSLBinding tussen de app en het certificaat.

PS >

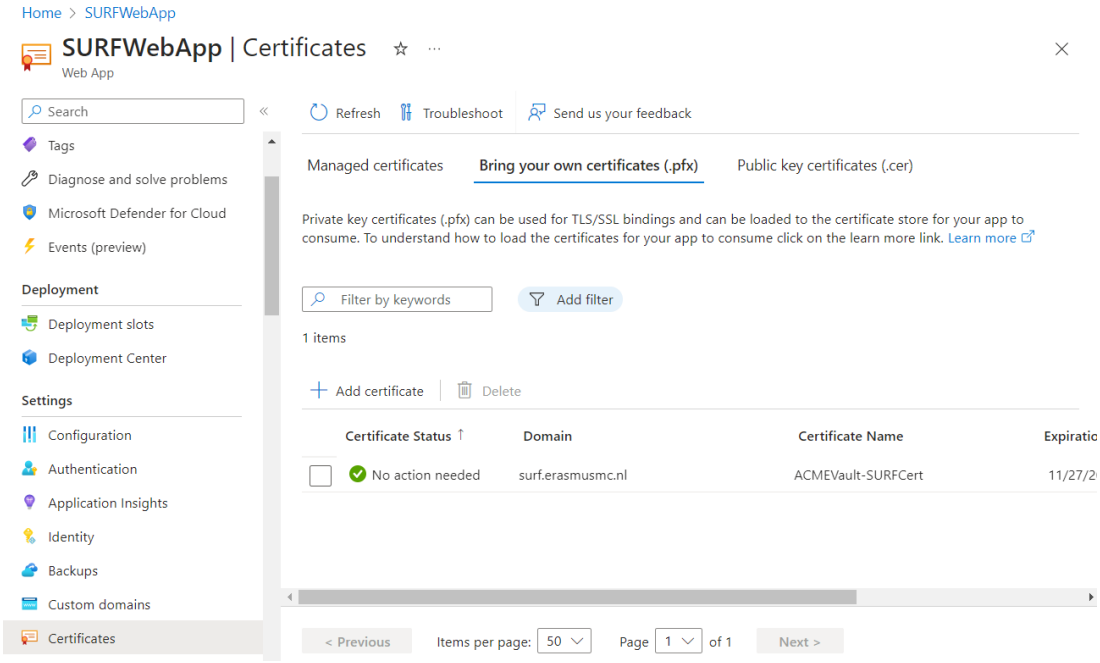
Set-AzWebApp -Name 'SURFWebApp' -ResourceGroupName 'acme' -HostNames (@($DomainName) + $WebApp.Hostnames)

New-AzWebAppSSLBinding -ResourceGroupName 'acme' -WebAppName 'SURFWebApp' -Thumbprint $Cert.Thumbprint -Name $DomainName

Upload vervolgens een bestand via FTP naar de WebApp.

```
PS >

$PubProfile = [xml] (Get-AzWebAppPublishingProfile -Name 'SURFWebApp' -ResourceGroupName
'acme' -OutputFile $null)

$username =
$PubProfile.SelectNodes('//publishProfile[@publishMethod="FTP"]/@userName').value
$password = $PubProfile.SelectNodes('//publishProfile[@publishMethod="FTP"]/@userPWD').value
$url = $PubProfile.SelectNodes('//publishProfile[@publishMethod="FTP"]/@publishUrl').value
$url = $url -replace 'ftps://', 'ftp://'

$filePath = Read-Host File Path
$file = Get-Item -Path $filePath
$uri = New-Object System.Uri("$url/$($file.Name)")

$request = [System.Net.FtpWebRequest] ([System.Net.WebRequest]::Create($uri))
$request.Method = [System.Net.WebRequestMethods+Ftp]::UploadFile
$request.Credentials = New-Object System.Net.NetworkCredential($username,$password)
$request.EnableSsl = $true;

$fileBytes = [System.IO.File]::ReadAllBytes($file.VersionInfo.FileName)
$request.ContentLength = $fileBytes.Length;
$requestStream = $request.GetRequestStream()

try {
   $requestStream.Write($fileBytes, 0, $fileBytes.Length)
}
finally {
   $requestStream.Dispose()
}

try {
   $response = [System.Net.FtpWebResponse] ($request.GetResponse())
   Write-Host "Status: $($response.StatusDescription)"
}
finally {
   if ($null -ne $response) {
      $response.Close()
   }
}
```

Na het uitvoeren van deze commando's heb je een nieuwe App Service Plan en Web App resource beschikbaar in het Azure portal. Bij de Web App kun je ook het geïmporteerde certificaat terugvinden.


Home > SURFWebApp
SURFWebApp | Certificates
Web App
Search
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Events (preview)
Deployment
Deployment slots
Deployment Center
Settings
Configuration
Authentication
Application Insights
Identity
Backups
Custom domains
Certificates
Refresh
Troubleshoot
Send us your feedback
Managed certificates
Bring your own certificates (.pfx)
Public key certificates (.cer)
Private key certificates (.pfx) can be used for TLS/SSL bindings and can be loaded to the certificate store for your app to consume. To understand how to load the certificates for your app to consume click on the learn more link. Learn more
Filter by keywords
Add filter
1 items
Add certificate
Delete
Certificate Status
Domain
Certificate Name
Expiratio
No action needed
surf.erasmusmc.nl
ACMEVault-SURFCert
11/27/2
< Previous
Items per page: 50
Page 1 of 1
Next >