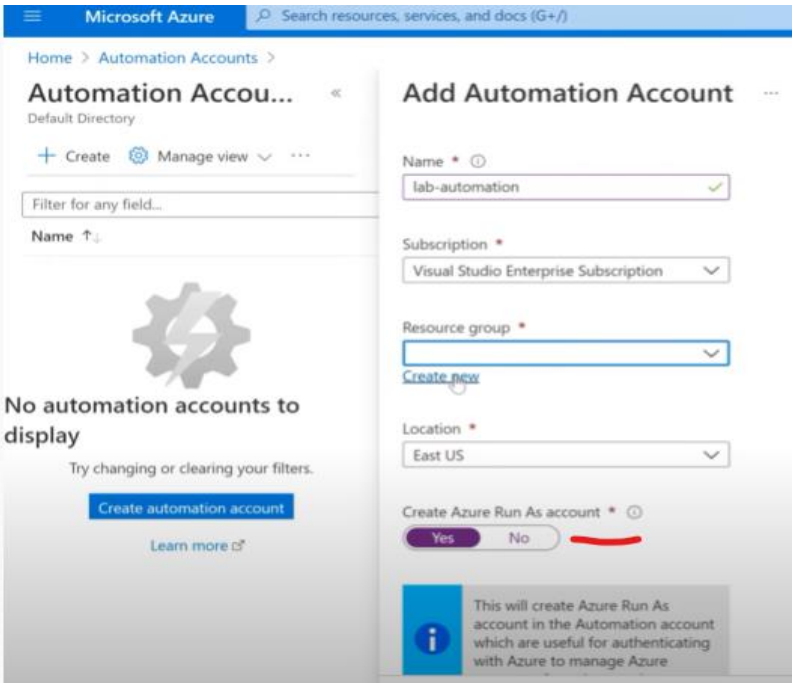
Methods of Authentication for Automation Account

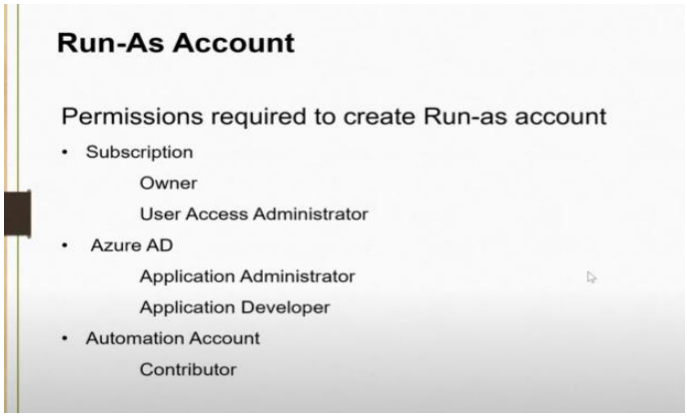
1. Run-As Account

Note: - All the below mentioned resources along with Service Principle are automatically generated if you select "Azure Run As account"



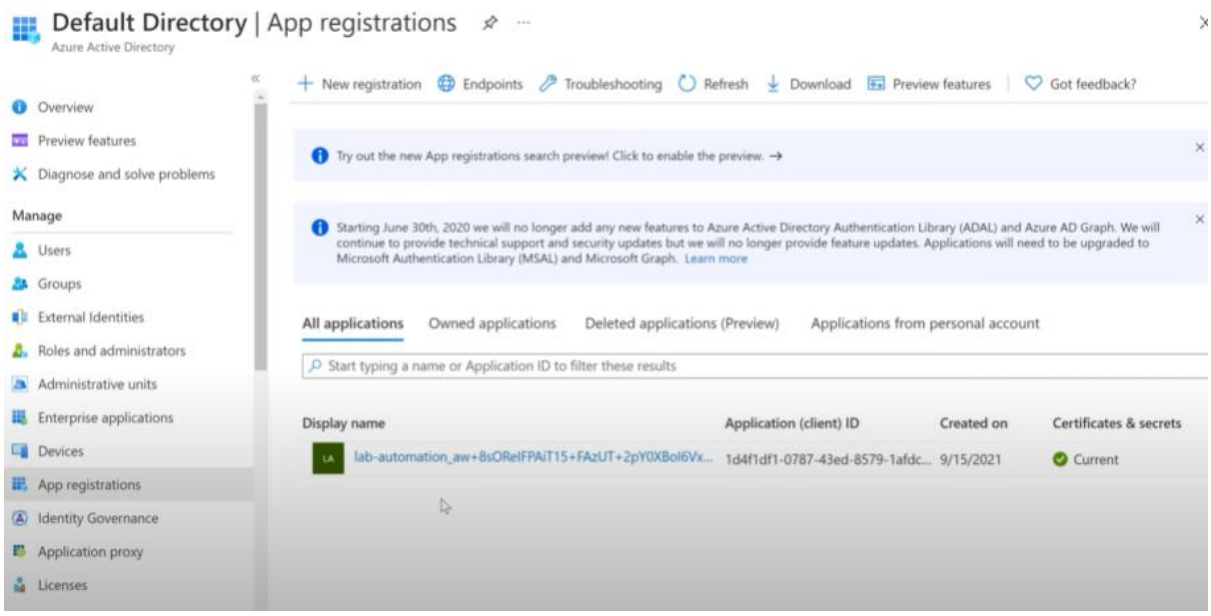
Note: - When you create an Automation account, the option to create a Run As account is no longer available. However, you can create a Run As account in your Automation account from the Azure portal after the creation.

For an automation account to have all permissions to automate services, one must have the following permissions:

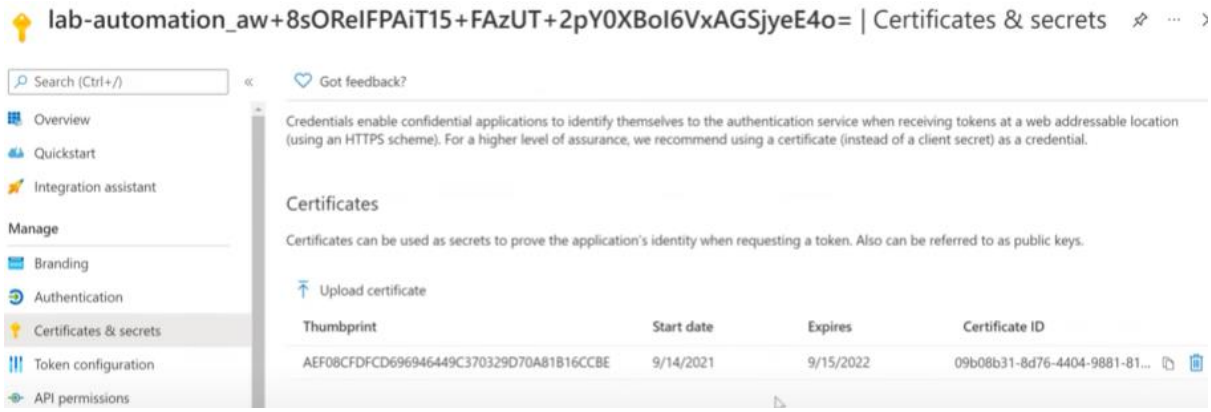


To verify:

Go to **Azure Active Directory** > **App Registrations** > There you shall see a new service principal created.

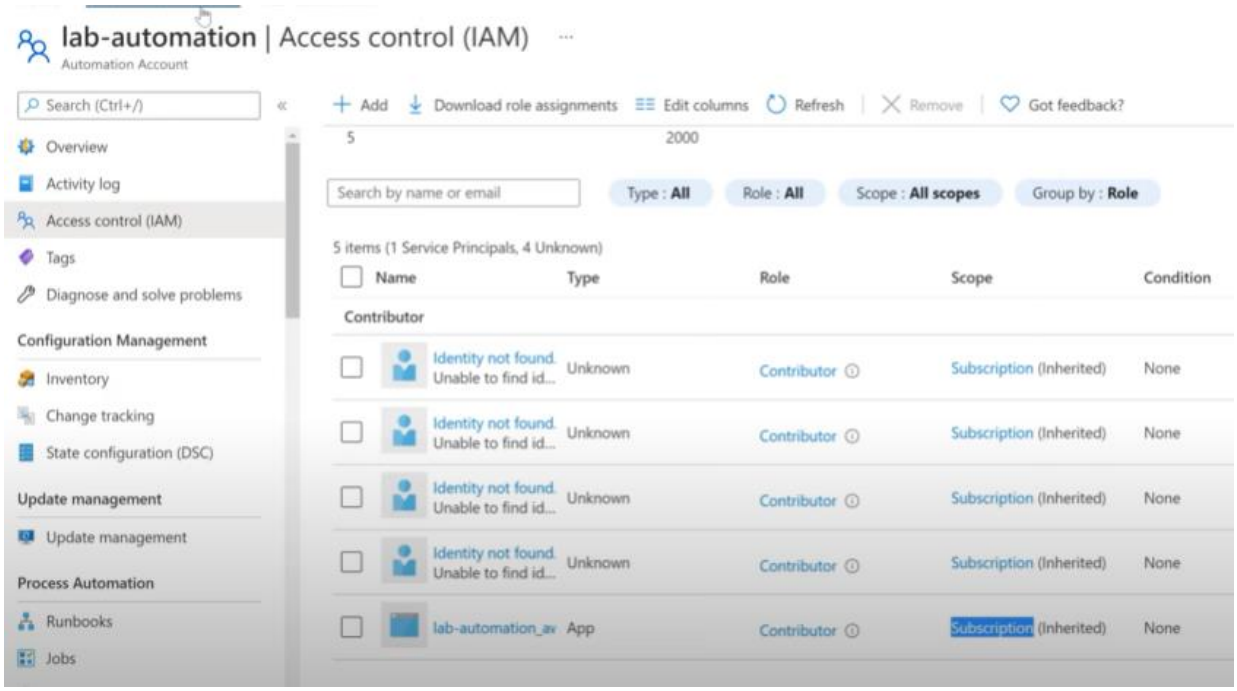


Select the created principal and go to **Certificates and Secrets**, there a self-signed certificate should also be created.



Note: - The automation account should stop working when the certificate expires.

Navigate back to Automation Account and click on **Access Control (IAM)** to check access.

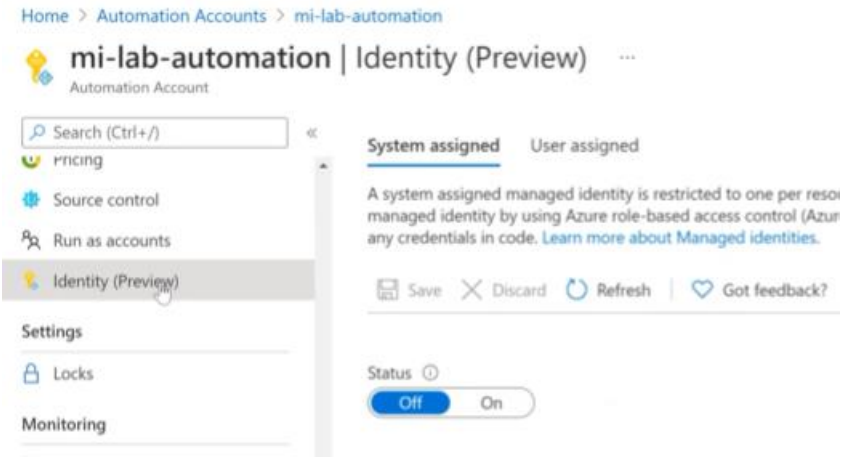


lab-automation is the service principle created in Azure Active Directory.

2. Managed Identity

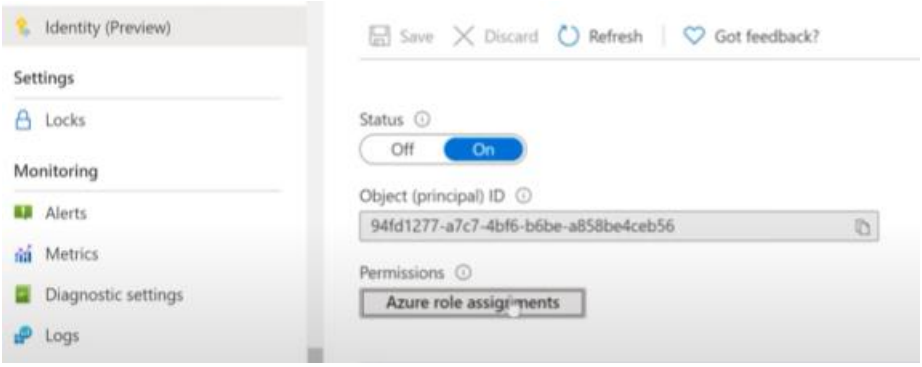
In this authentication method, you do not select the “Run as Account” and use the **Identity** feature.

Go to your Automation Account and find Identity.



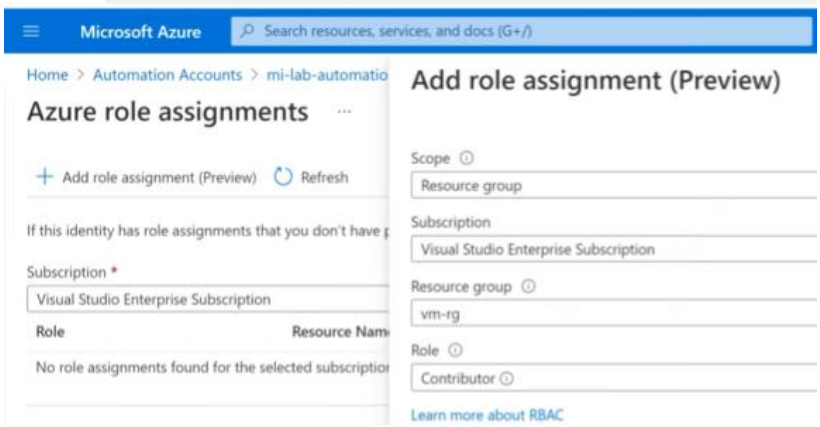
Set the Status to ON and save.

This creates a service principal in the back end and will allow Microsoft to manage the service principal for you.



After turning it on, go to the Azure role assignments and assign the role to the service which you want to use for automation.

In this example, Contributor access to a resource group is given.



Note: - If you go back to the Active Directory App Registrations, you won't see the service principal there because it is managed by Microsoft.

To start working on the runbook, Use the following line of code in case of Identity Management Authentication to set a connection.

```
 Param
        (
            [Parameter (Mandatory= $true)]
            [string]$VmResourceGroup,
            [Parameter (Mandatory= $true)]
            [string]$VmName
        )

        Connect-AzAccount -Identity
        Set-AzContext -SubscriptionId "xxxx-xxxxxx-xxxxxx-xxxxx"
```

For Run As Account, use the following;

```
Param
(
    [Parameter (Mandatory= $true)]
    [string]$VmResourceGroup,
    [Parameter (Mandatory= $true)]
    [string]$VmName
)

$connectionName = "AzureRunAsConnection"
try
{
    $servicePrincipalConnection = Get-AutomationConnection -Name $connectionName
    Connect-AzAccount `
        -ServicePrincipal `
        -TenantId $servicePrincipalConnection.TenantId `
        -ApplicationId $servicePrincipalConnection.ApplicationId `
        -CertificateThumbprint $servicePrincipalConnection.CertificateThumbprint
}
catch {
    if (!$servicePrincipalConnection)
    {
        $ErrorMessage = "Connection $connectionName not found."
        throw $ErrorMessage
    } else{
        Write-Error -Message $_.Exception
        throw $_.Exception
    }
}
```