

## 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"

The screenshot shows the 'Add Automation Account' page in the Microsoft Azure portal. On the left, there's a sidebar with 'Automation Accounts' and a 'No automation accounts to display' message. The main form on the right has the following fields:

- Name:** lab-automation
- Subscription:** Visual Studio Enterprise Subscription
- Resource group:** (empty, with a 'Create new' link below it)
- Location:** East US
- Create Azure Run As account:** Yes (selected), No

A blue information box at the bottom states: "This will create Azure Run As account in the Automation account which are useful for authenticating with Azure to manage Azure".

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:

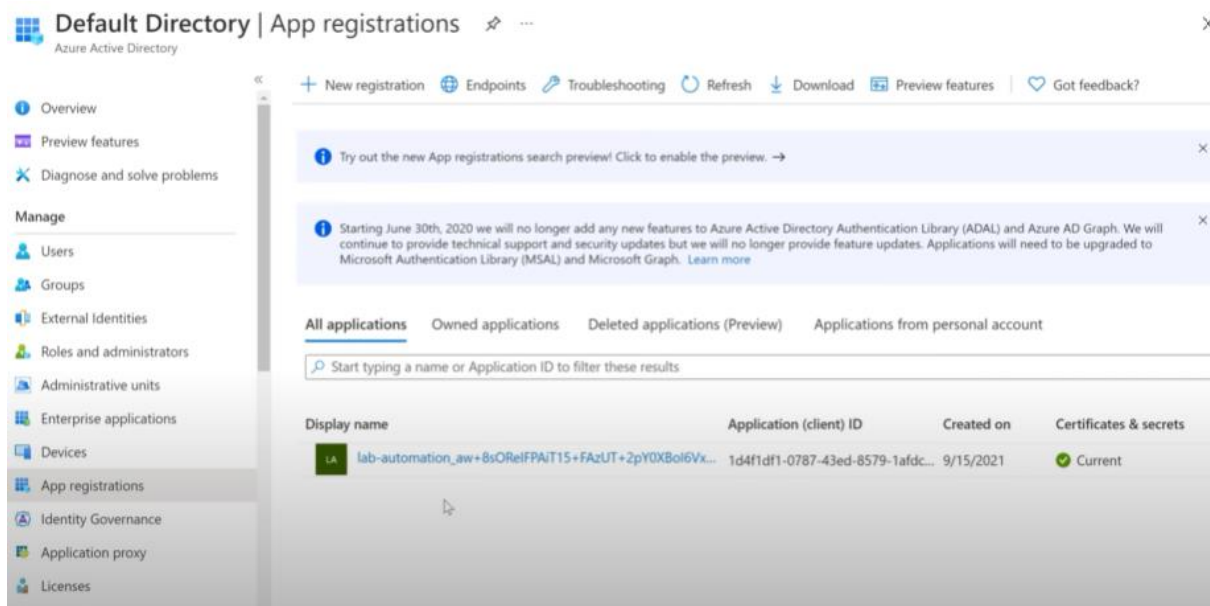
**Run-As Account**

Permissions required to create Run-as account

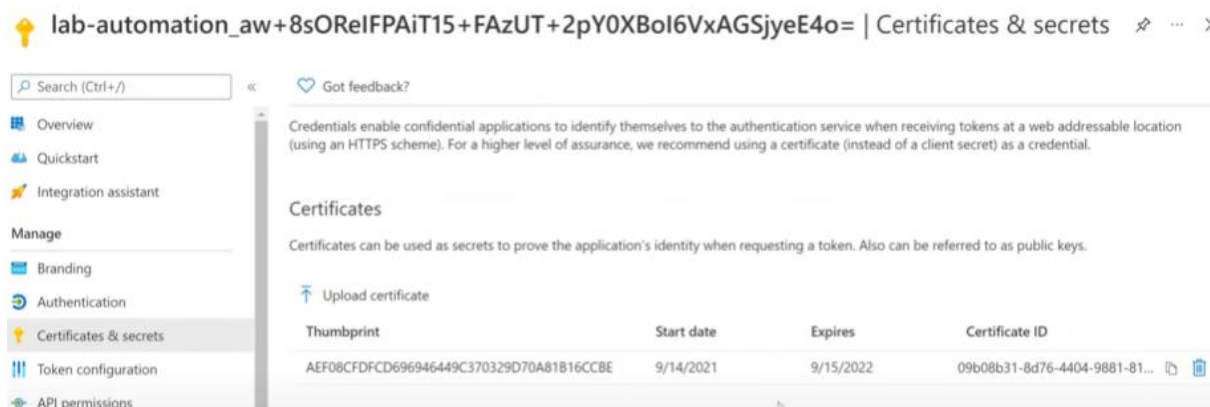
- Subscription
  - Owner
  - User Access Administrator
- Azure AD
  - Application Administrator
  - Application Developer
- Automation Account
  - Contributor

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.

The screenshot shows the 'Access control (IAM)' page for the 'lab-automation' Automation Account. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration Management (Inventory, Change tracking, State configuration (DSC)), Update management (Update management), Process Automation (Runbooks, Jobs), and a search bar. The main content area shows a list of 5 items (1 Service Principal, 4 Unknown) with columns for Name, Type, Role, Scope, and Condition. The 'lab-automation\_av' App is listed as a Contributor with a Subscription (Inherited) scope.

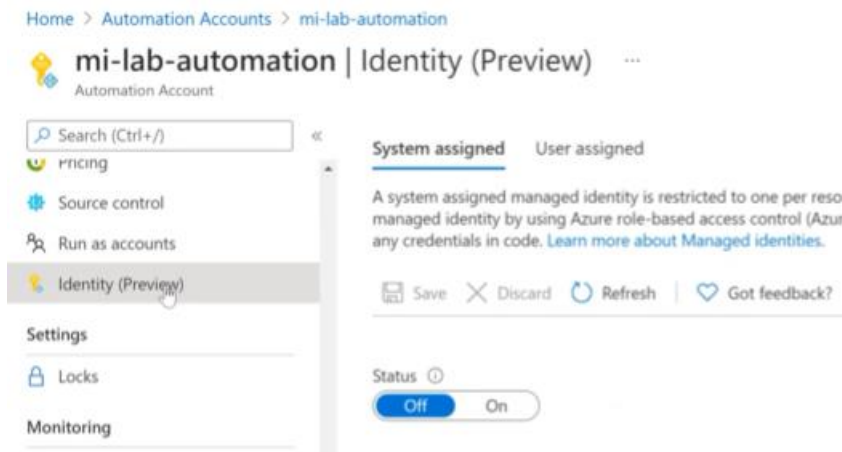
Name	Type	Role	Scope	Condition
Identity not found. Unable to find id...	Unknown	Contributor	Subscription (Inherited)	None
Identity not found. Unable to find id...	Unknown	Contributor	Subscription (Inherited)	None
Identity not found. Unable to find id...	Unknown	Contributor	Subscription (Inherited)	None
Identity not found. Unable to find id...	Unknown	Contributor	Subscription (Inherited)	None
lab-automation_av	App	Contributor	Subscription (Inherited)	None

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

## 2. Managed Identity

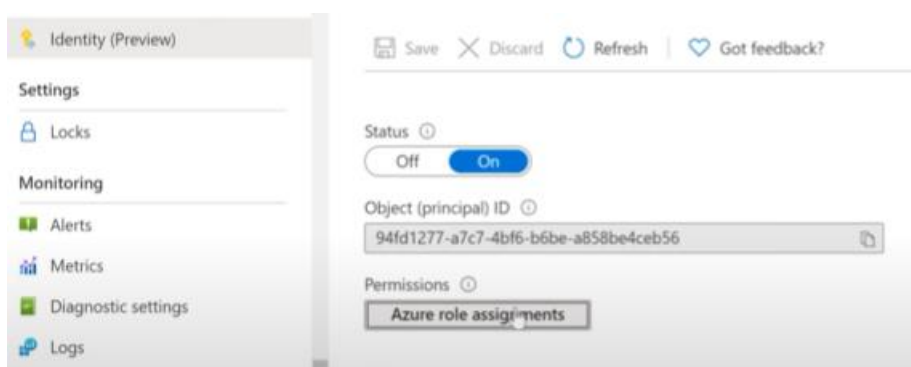
In this authentication method, you do not select the “Run as Account” and use the **Identity** feature.

Go to your Automation Account and find Identity.



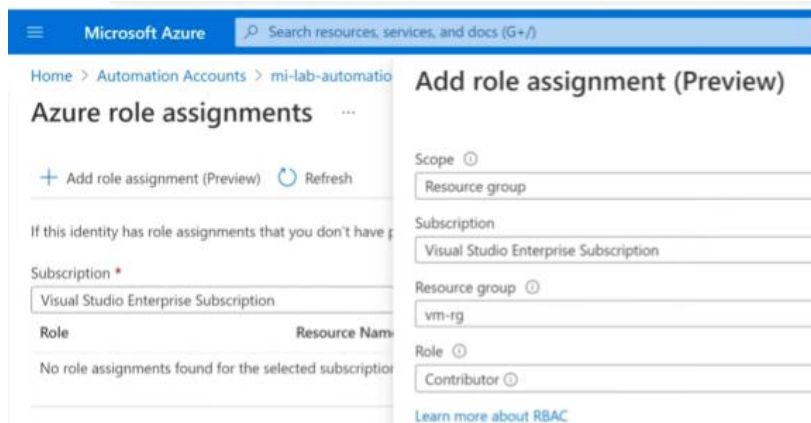
Set the Status to ON and save.

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.



Microsoft Azure Search resources, services, and docs (G+)

Home > Automation Accounts > mi-lab-automation

### Azure role assignments

+ Add role assignment (Preview) Refresh

If this identity has role assignments that you don't have permission to view, click here.

Subscription \*

Visual Studio Enterprise Subscription

Role	Resource Name
No role assignments found for the selected subscription.	

### Add role assignment (Preview)

Scope ⓘ

Resource group

Subscription

Visual Studio Enterprise Subscription

Resource group ⓘ

vm-rg

Role ⓘ

Contributor ⓘ

[Learn more about RBAC](#)

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
    }
}
```