

# LAB11: Additional automation capabilities for your Azure Arc-enabled servers

---

## Student Lab Manual

### Table of Contents

Exercise 1 - Run automation runbooks on your Azure Arc-enabled servers using Hybrid runbook workers

**Task 1 - Create Automation account using the Azure Portal**

**Task 2 - Create Automation account using Powershell**

**Task 3 - Add Hybrid Runbook Workers using the Azure Portal**

**Task 4 - Add Hybrid Runbook Workers using Powershell**

**Task 5 - Create and start a runbook**

Exercise 2 - Securely Connect to your Azure Arc-enabled servers using SSH access

**Task 1 - Install prerequisites on client machine**

**Task 2 - Enable SSH service on Arc-enabled servers**

**Task 3 - Connect to Arc-enabled servers**

**Task 4 - Optional: Azure AD/Entra ID based SSH Login**

Exercise 3 - Run PowerShell and Shell scripts on Azure Arc-enabled servers using the Run command

**Task 1 - Check the pre-requisites**

**Task 2 - Use the Run command to execute a simple PowerShell script within an Arc-connected Windows machine**

**Task 3 - Use the Run command to execute a simple Shell command within an Arc-connected Linux machine**

**Task 4 - Direct the output of a Run command to Azure storage blob**

====

## Exercise 1 - Run automation runbooks on your Azure Arc-enabled servers using Hybrid runbook workers

---

### Objective

Onboard Azure Arc-enabled servers as Hybrid runbook workers in Azure Automation.

### Estimated Time to Complete This Lab

20 minutes

### Explanation

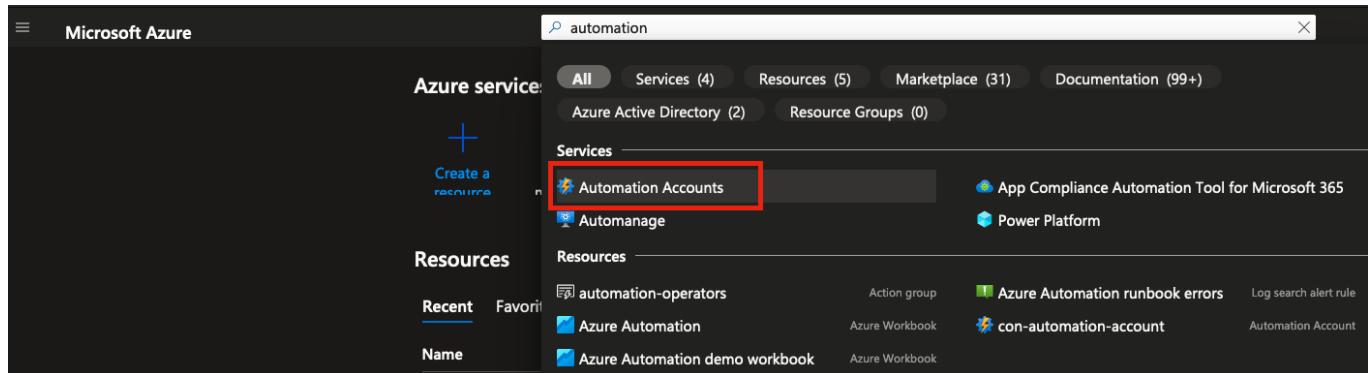
In this module we will onboard Azure Arc-enabled servers as Hybrid runbook workers in Azure Automation. We will then create and start runbooks on the hybrid runbook workers to see how this feature can be leveraged.

====

## Task 1: Create Automation account using the Azure Portal

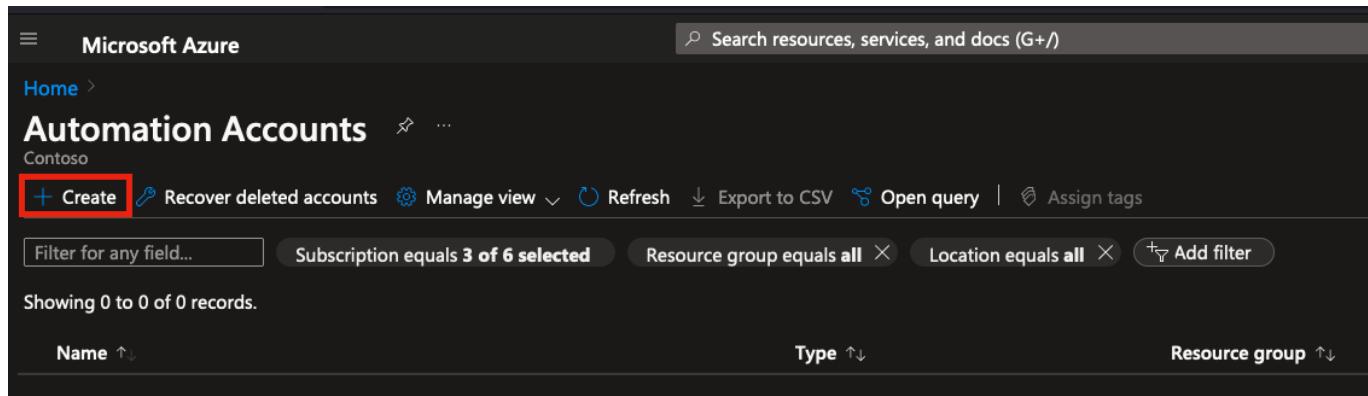
[!alert] You have two options to create an Automation account. You can either follow the steps in this task to use the Azure Portal, or if you prefer to use Powershell then jump to [Task 2](#) and continue from there.

1. [] In the Azure Portal, search for *automation* and navigate to *Automation accounts*



The screenshot shows the Microsoft Azure portal interface. A search bar at the top contains the text "automation". Below the search bar, a navigation bar includes "All", "Services (4)", "Resources (5)", "Marketplace (31)", and "Documentation (99+)". Under the "Services" section, a list of services is shown, with "Automation Accounts" highlighted by a red box. Other listed services include "App Compliance Automation Tool for Microsoft 365", "Power Platform", "Automanage", "automation-operators", "Azure Automation", "Azure Automation demo workbook", "Action group", "Azure Workbook", "Azure Automation runbook errors", "Log search alert rule", "con-automation-account", and "Automation Account".

2. [] Click on "Create"



The screenshot shows the "Automation Accounts" blade in the Microsoft Azure portal. At the top, there is a "Create" button highlighted with a red box. Below it are filter options: "Subscription equals 3 of 6 selected", "Resource group equals all", "Location equals all", and "Add filter". The main area displays a message: "Showing 0 to 0 of 0 records." Below this, there are columns for "Name", "Type", and "Resource group".

3. [] Select the subscription and resource group where you have deployed ArcBox.
4. [] Enter *ArcBox-Automation* as the name for the Automation Account.

5. [] Select the same region as your ArcBox environment is deployed to.

6. [] Click Next

The screenshot shows the 'Create an Automation Account' page in the Microsoft Azure portal. The 'Basics' tab is selected. A red box highlights the following fields:

- Subscription:** A dropdown menu showing a selected subscription.
- Resource group:** A dropdown menu with options "Select a resource group" and "Create new".
- Instance Details:**
  - Automation account name:** An input field with placeholder text "Enter name".
  - Region:** A dropdown menu set to "East US".

7. [] Leave the default settings for *Managed Identities* in place and click Next

The screenshot shows the 'Create an Automation Account' page in the Microsoft Azure portal, with the 'Advanced' tab selected. A red box highlights the "System assigned" checkbox under the "Managed Identities" section. The checkbox is checked by default.

8. [] Leave the default settings for *Connectivity configuration* in place and click Next

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Automation Accounts >

## Create an Automation Account

Basics Advanced Networking Tags Review + Create

### Network connectivity

You can connect to your automation account either through public IP addresses for public access or through a private endpoint for private access. [Learn more](#)

Connectivity configuration  Public access  Private access

**i** When you select public access, traffic from all public networks can access this Automation account resource. Select private access if you want to restrict access to automation endpoints only from authorized virtual networks required for secure applications or environments.

9. [] Optionally, add any tags you may want to add to the resource. Click Next

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Automation Accounts >

## Create an Automation Account

Basics Advanced Networking Tags Review + Create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

**i** Note: Automation accounts are limited to 15 tags.

Name ⓘ	Value ⓘ
	:

10. [] Click Create

The screenshot shows the 'Create an Automation Account' page in the Microsoft Azure portal. At the top, there's a green banner with a checkmark icon and the text 'Validation passed'. Below this, there are tabs for 'Basics', 'Advanced', 'Networking', 'Tags', and 'Review + Create', with 'Review + Create' being the active tab. The 'Basics' section contains the following configuration:

Name	ArcBox-Automation
Subscription	Demo
Resource group	arcbox-demo-rg
Region	East US

The 'Advanced' section shows:

System assigned identity	Yes
User assigned identity	None

The 'Networking' section shows:

Network connectivity	Public access
----------------------	---------------

The 'Tags' section shows '(none)'. At the bottom, there are buttons for 'Create' (which is highlighted with a red box), 'Previous', and 'Next'.

## Task 1 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 2: Create Automation account using Powershell

[!alert] This task is an alternative to, and only needed if you have not executed [task 1](#). You can jump straight to [task 3](#) if you have already provisioned an Automation account using the Azure portal.

1. [] From the *ArcBox-Client* machine start PowerShell 7.

2. [] Customize the parameter values in the following Powershell script to reflect your environment for the resource group name and location. Paste the code in the PowerShell window and press Enter.

```
New-AzAutomationAccount -Location "<Your chosen Location>" -Name "ArcBox-Automation" -ResourceGroupName "<Your Resource Group Name>" -AssignSystemIdentity
```

The output should look similar to this:

```
New-AzAutomationAccount @AutomationAccountParams

SubscriptionId      : [REDACTED]
ResourceGroupName   : jan-arcbox-01-rg
AutomationAccountName : ArcBox-Automation
Location           : East US
State              : Ok
Plan               : Basic
CreationTime       : 9/6/2023 7:53:30 PM +00:00
LastModifiedTime   : 9/6/2023 7:53:30 PM +00:00
LastModifiedBy     :
Tags               : {[Project, jumpstart_arcbox]}
Identity           : Microsoft.Azure.Management.Automation.Models.Identity
Encryption         : Microsoft.Azure.Management.Automation.Models.EncryptionProperties
PublicNetworkAccess :
```

### Task 2 has been completed

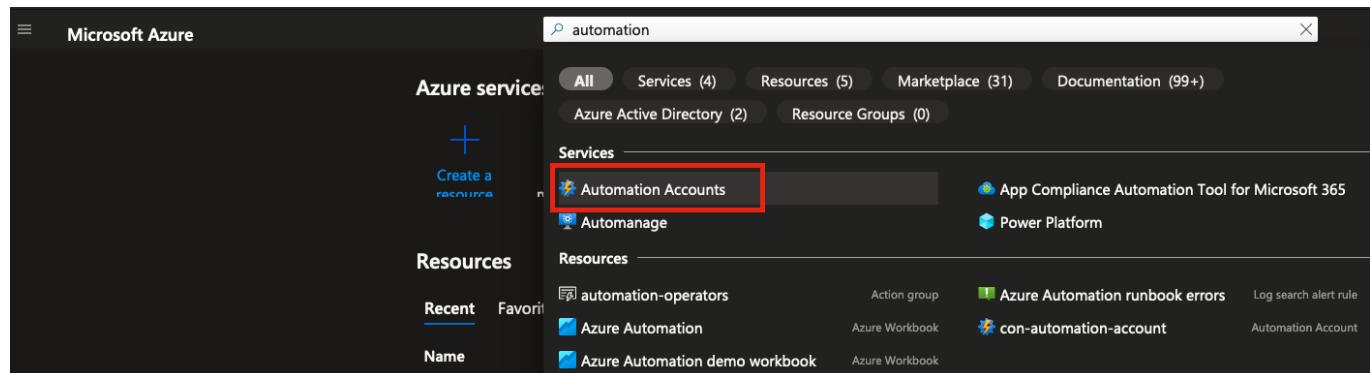
Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 3 - Add Hybrid Runbook Workers using the Azure Portal

[!alert] You have two options to add hybrid runbook workers. You can either follow the steps in this task to use the Azure Portal, or if you prefer to use Powershell then jump to Task [task 4](#) and continue from there.

1. [] In the Azure Portal, search for *automation* and navigate to *Automation accounts*



2. [] Navigate to the ArcBox-Automation account you created previously

3. [] Select *Hybrid worker groups*:

The screenshot shows the Azure portal interface for the 'ArcBox-Automation' automation account. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Process Automation, Runbooks, Jobs, and Hybrid worker groups. The 'Hybrid worker groups' link is highlighted with a red box. The main content area displays essential information about the resource group, including its name ('jan-arcbox-01-rg'), location ('East US'), subscription ('Azure Arc Jumpstart Subscription'), and tags ('Project : jumpstart\_arcbox'). A note at the top encourages using the Azure Automation Extension for Visual Studio Code.

4. [] Click *Create hybrid worker group*:

The screenshot shows the 'Hybrid worker groups' page within the 'ArcBox-Automation' account. The left sidebar has a red box around the 'Hybrid worker groups' link. The main area shows a note about the retirement of Agent-based User Hybrid Workers and the creation of User hybrid worker groups. It includes a search bar and a table with columns for 'Group name' and 'Created On'. A message at the bottom states 'No hybrid worker groups found.'

5. [] Type *windows-workers* as the name of the new Hybrid worker group, leave the default value for *Use Hybrid Worker Credentials* and click Next

The screenshot shows the 'Create Hybrid worker group' wizard on the 'Basics' tab. The 'Name' field is highlighted with a red box. Below it, there are options for 'Use Hybrid Worker Credentials' (with 'Custom' and 'Default' buttons) and a note about completing the Basics tab before proceeding to Review + Create.

6. [] Click *Add machines*:

Home > ArcBox-Automation | Hybrid worker groups >

## Create Hybrid worker group

Hybrid worker group

\* Basics   Hybrid workers   Review + Create

Select Azure Virtual Machines, Arc-enabled servers or Arc-enabled VMware VM (preview) to host the automation jobs. If you don't select any machines, an empty hybrid worker group will be created. You can still add or remove machines later.

**Add machines** Remove machines

Machine Name	Subscription Id	Resource Group	Operating System	Location
No Machines Selected				

7. [] Select *ArcBox-Win2K22* and click *Add*:

Home > Automation Accounts > AutomationAccounts | Hybrid worker groups > Create Hybrid worker group >

### Add machines as hybrid worker

Selected machines would be added to the hybrid worker group and hybrid worker extension would be installed to connect to Automation service. If a machine is not visible or not added to the group, read about the [prerequisites](#) and [troubleshooting steps](#).

Subscription : Visual Studio Enterprise Subsc...   Resource Group : No filter   Location : No filter   Type : All

Search to filter items...

Name	Subscription	Resource group	Operating System	Location	Status
<input type="checkbox"/> ArcBox-Client	[REDACTED]	arbbox-levelup	Windows	centralus	VM running
<input type="checkbox"/> arcbox-ubuntu-02	[REDACTED]	Arbbox	Linux	centralus	Connected
<input checked="" type="checkbox"/> ArcBox-Win2K22	[REDACTED]	Arbbox	Windows	centralus	Connected
<input type="checkbox"/> Arcbox-Win2K25	[REDACTED]	Arbbox	Windows	centralus	Connected

Add   Cancel   0 machines selected

8. [] Click *Review + Create*:

Home > ArcBox-Automation | Hybrid worker groups >

## Create Hybrid worker group

Hybrid worker group

\* Basics   Hybrid workers   **Review + Create**

Select Azure Virtual Machines, Arc-enabled servers or Arc-enabled VMware VM (preview) to host the automation jobs. If you don't select any machines, an empty hybrid worker group will be created. You can still add or remove machines later.

**Add machines** Remove machines

Machine Name	Subscription Id	Resource Group	Operating System	Location
<input type="checkbox"/> ArcBox-Win2K22	16471a83-9151-456e-bbb1-463027bed604	jan-arcbox-01-rg	Windows	northeurope

**Review + Create**   Previous   Next

9. [] Click *Create*:

Home > ArcBox-Automation | Hybrid worker groups >

## Create Hybrid worker group

Hybrid worker group

\* Basics   Hybrid workers   Review + Create

Selected machines would be added to the hybrid worker group and hybrid worker extension would be installed to connect to Automation service.

Name	windows-workers
Use Hybrid Worker Credentials	NA
Hybrid Workers count	1

**Create**   Previous   Next

10. [] Wait for the following activities to be finished:

The screenshot shows a list of activity log notifications in a dark-themed interface. There are four entries:

- HybridWorkerExtension installation status**: Extension installation status for the machines which were successfully added as hybrid workers. Success: 1 Failure: 0. A minute ago.
- HybridWorkerExtension installation succeeded**: Machine name: 'ArcBox-Win2K22'. A minute ago.
- Hybrid worker addition status: 'windows-workers'**: Success: 1 Failure: 0. 3 minutes ago.
- Created hybrid worker group successfully**: Group name: 'windows-workers'. 3 minutes ago.

11. [] Repeat the above steps to create an additional Hybrid worker group called *linux-workers* where you select to onboard the machine *ArcBox-Ubuntu01* to the group.
12. [] After completing this task you should have the following Hybrid worker groups:

Group name	Created On
linux-workers	06/09/2023, 10:29 pm
windows-workers	06/09/2023, 10:23 pm

### Task 3 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 4 - Add Hybrid Runbook Workers using Powershell

[!alert] This task is an alternative to [task 3](#), and only needed if you have not executed task 3. You can jump straight to [task 5](#) if you have already added the hybrid runbook workers from the Azure portal.

1. [] From the *ArcBox-Client* machine start PowerShell 7.
2. [] Copy the following Powershell script into a text editor and customize the parameter values to reflect your environment for the resource group name if needed (**repeat for all occurrences!**). Paste the code in the Cloud Shell PowerShell window and press Enter.

```
# Retrieve service URL for Automation account (used when registering Arc-enabled
Servers as Hybrid Runbook Workers)
$AutomationAccount = Get-AzResource -ResourceGroupName "ArcBox" -Name "ArcBox-
Automation"
$AutomationAccountInfo = Invoke-AzRestMethod -SubscriptionId
$AutomationAccount.SubscriptionId -ResourceGroupName
$AutomationAccount.ResourceGroupName -ResourceProviderName Microsoft.Automation -
ResourceType automationAccounts -Name $AutomationAccount.Name -ApiVersion 2021-06-
22 -Method GET
$AutomationHybridServiceUrl = ($AutomationAccountInfo.Content | ConvertFrom-
Json).Properties.automationHybridServiceUrl

# Create the Linux Hybrid Worker Group
```

```
New-AzAutomationHybridRunbookWorkerGroup -Name linux-workers -ResourceGroupName
ArcBox -AutomationAccountName ArcBox-Automation

#Get the Arc-enabled Linux VM and add to the Hybrid Worker to the group
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-
Ubuntu-01
New-AzAutomationHybridRunbookWorker -ResourceGroupName ArcBox -
AutomationAccountName ArcBox-Automation -HybridRunbookWorkerGroupName linux-
workers -VMResourceId $ArcResource.Id -Name (New-Guid).Guid
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-
Ubuntu-01
$settings = @{
    "AutomationAccountURL" = $AutomationHybridServiceUrl
}
New-AzConnectedMachineExtension -ResourceGroupName $ArcResource.ResourceGroupName
-Location $ArcResource.Location -MachineName $ArcResource.Name -Name
"HybridWorkerExtension" -Publisher "Microsoft.Azure.Automation.HybridWorker" -
ExtensionType HybridWorkerForLinux -TypeHandlerVersion 1.1 -Setting $settings -
EnableAutomaticUpgrade

# Create the Windows Hybrid Worker Group
New-AzAutomationHybridRunbookWorkerGroup -Name windows-workers -ResourceGroupName
ArcBox -AutomationAccountName ArcBox-Automation

#Get the Arc-enabled Windows VM and add to the Hybrid Worker to the group
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name ArcBox-
Win2K22
New-AzAutomationHybridRunbookWorker -ResourceGroupName ArcBox -
AutomationAccountName ArcBox-Automation -HybridRunbookWorkerGroupName windows-
workers -VMResourceId $ArcResource.Id -Name (New-Guid).Guid
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-
Win2K22
$settings = @{
    "AutomationAccountURL" = $AutomationHybridServiceUrl
}
New-AzConnectedMachineExtension -ResourceGroupName $ArcResource.ResourceGroupName
-Location $ArcResource.Location -MachineName $ArcResource.Name -Name
"HybridWorkerExtension" -Publisher "Microsoft.Azure.Automation.HybridWorker" -
ExtensionType HybridWorkerForWindows -TypeHandlerVersion 1.1 -Setting $settings -
EnableAutomaticUpgrade
```

3. [] It will take few minutes to execute the script.

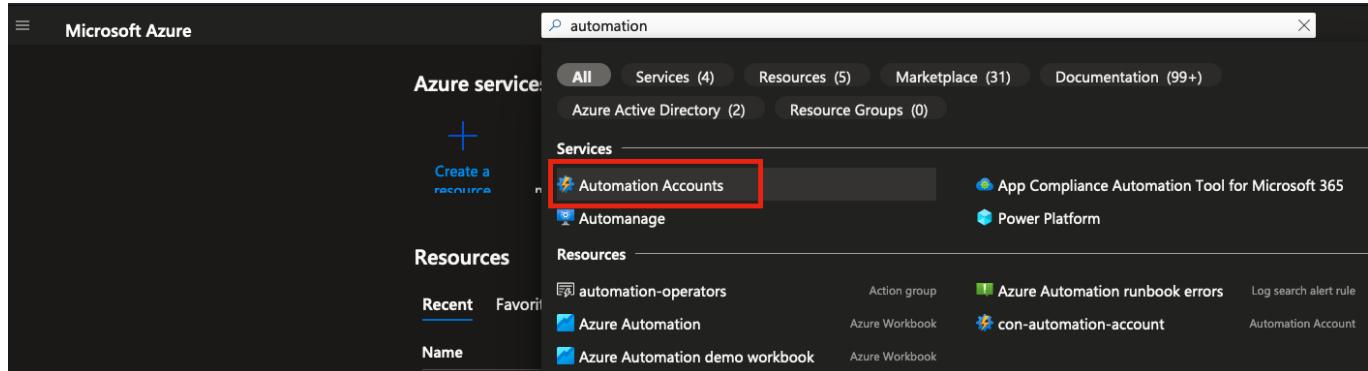
#### Task 4 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

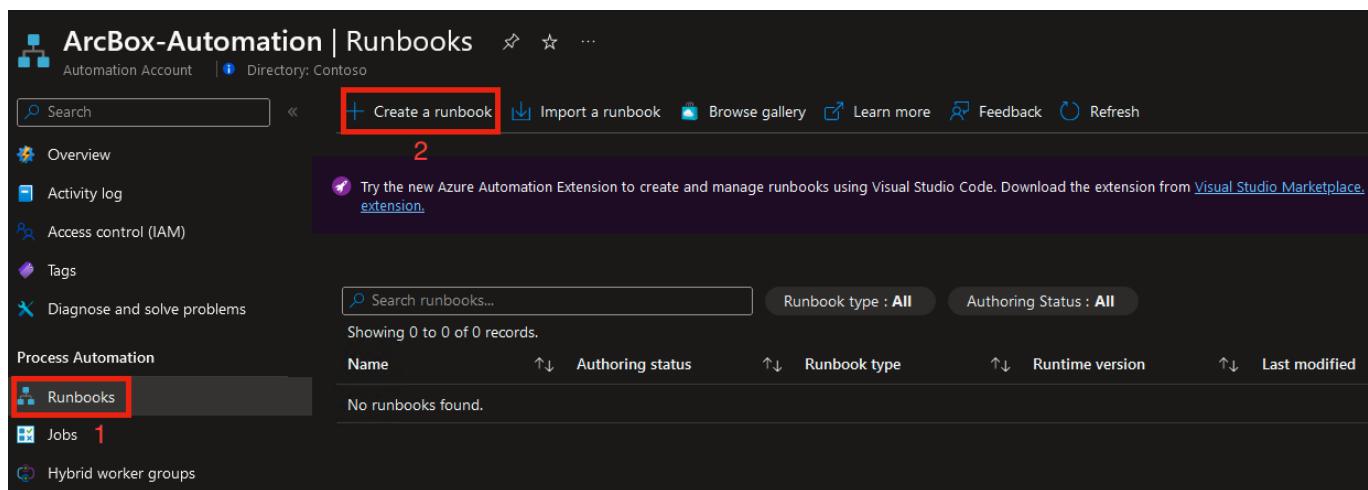
## Task 5 - Create and start a runbook

1. [] In the Azure Portal, search for *automation* and navigate to *Automation accounts*



The screenshot shows the Microsoft Azure portal interface. A search bar at the top contains the text "automation". Below the search bar, a navigation bar includes tabs for "All", "Services (4)", "Resources (5)", "Marketplace (31)", and "Documentation (99+)". Under the "Services" tab, a list of services is shown, with "Automation Accounts" highlighted by a red box. Other listed services include "App Compliance Automation Tool for Microsoft 365", "Power Platform", "Automanage", and "Azure Automation". On the left side, there are sections for "Create a resource" and "Resources" (Recent and Name). The main content area displays several automation-related resources.

2. [] Navigate to the *ArcBox-Automation* account you created previously. Select *Runbooks* and click *Create a runbook*



The screenshot shows the "ArcBox-Automation | Runbooks" page. The top navigation bar includes a search bar, a "Create a runbook" button (highlighted with a red box), and links for "Import a runbook", "Browse gallery", "Learn more", "Feedback", and "Refresh". The left sidebar lists "Overview", "Activity log", "Access control (IAM)", "Tags", "Diagnose and solve problems", "Process Automation" (with "Runbooks" highlighted with a red box), "Jobs" (1 item), and "Hybrid worker groups". The main content area displays a table for runbooks, showing 0 records found. The table has columns for "Name", "Authoring status", "Runbook type", "Runtime version", and "Last modified".

3. [] Enter the following values then click *Create*

- Name: Start-DiskClean
- Runbook type: PowerShell
- Runtime version: 7.2
- Description: Invoke disk cleanup

Home > Automation Accounts > ArcBox-Automation | Runbooks >

## Create a runbook

Name \* ⓘ Start-DiskClean ✓

Runbook type \* ⓘ PowerShell ✓

Runtime version \* ⓘ 7.2 (preview) ✓

Description  
Invoke disk cleanup

**i** During runbook execution, PowerShell modules targeting 7.2 runtime version will be used. Please make sure the required PowerShell modules are present in 7.2 runtime version.

**Create** **Cancel**

4. [] After provisioning, the runbook editor will open the newly created runbook:

Home > Automation Accounts > ArcBox-Automation | Runbooks > Start-DiskClean (ArcBox-Automation/Start-DiskClean) >

## Edit PowerShell Runbook

Start-DiskClean

Save Publish Revert to published Test pane Edit in VS Code Feedback

> CMDLETS  
> RUNBOOKS  
> ASSETS

5. [] Paste the following script into the editor pane then Save:

```
if ($IsWindows) {  
  
    Write-Output 'Free disk space before cleanup action'  
  
    Get-Volume -DriveLetter C | Out-String  
  
    Write-Output "Windows Update component store cleanup"  
    Dism.exe /online /Cleanup-Image /StartComponentCleanup /ResetBase  
  
    $SystemTemp = "$env:SystemRoot\Temp"  
    Write-Output "Empty the system temporary folder: $SystemTemp"  
    Get-ChildItem -Path $SystemTemp -Recurse | Remove-Item -Force -Recurse  
  
    Write-Output 'Free disk space after cleanup action'
```

```

Get-Volume -DriveLetter C | Out-String

} elseif ($IsLinux) {

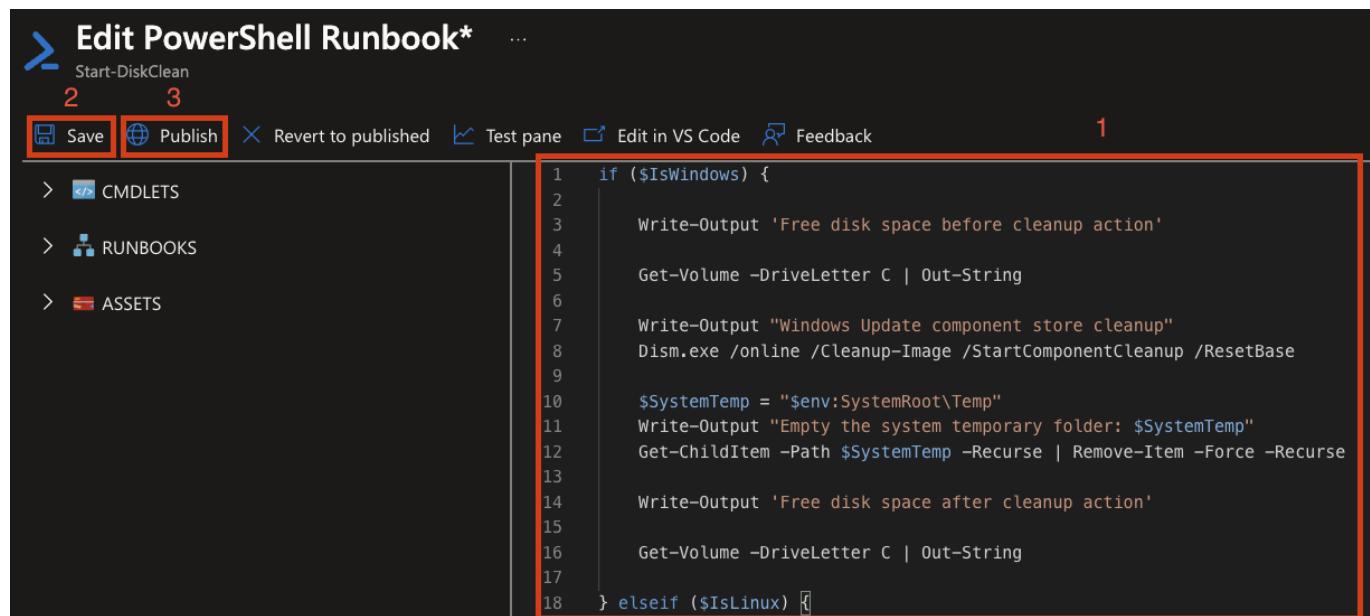
    Write-Output 'Free disk space before cleanup action'
    df -h -m
    # Specify the directory where your log files are located
    $logDir = '/var/log'
    # Define the number of days to retain log files
    $daysToKeep = 7
    # Get the current date
    $currentDate = Get-Date
    # Calculate the date threshold for log file deletion
    $thresholdDate = $currentDate.AddDays(-$daysToKeep)
    # List log files in the specified directory that are older than the threshold
    $filesToDelete = Get-ChildItem -Path $logDir -File | Where-Object {
        $_.LastWriteTime -lt $thresholdDate }

    # Delete the old log files
    foreach ($file in $filesToDelete) {
        Remove-Item -Path $file.FullName -Force
    }
    Write-Output 'Free disk space after cleanup action'
    df -h -m

}

```

## 6. [] Click Publish

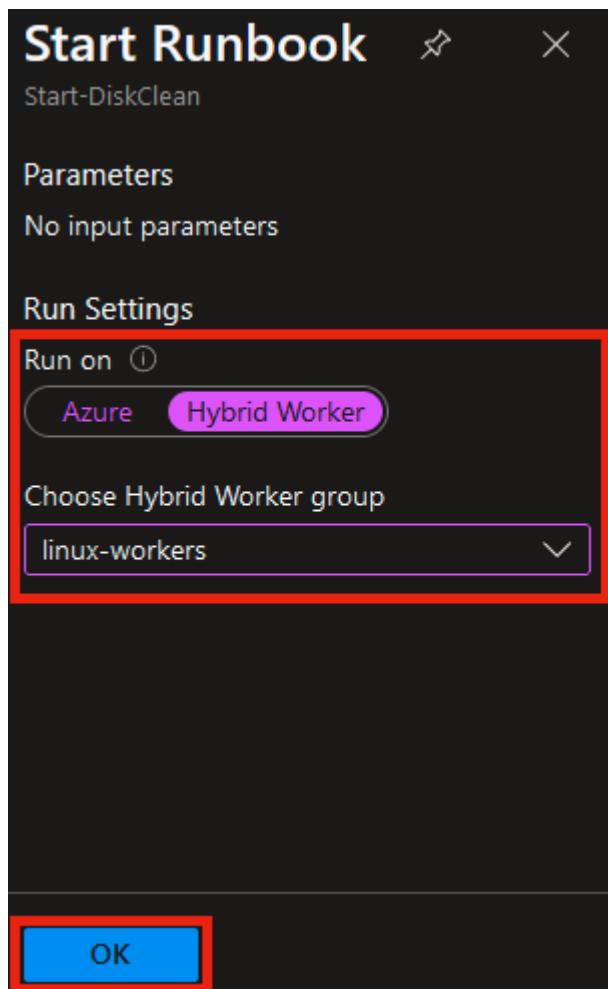


## 7. [] Click Start

**Note:** You may need to click Refresh for the Start button to become active

The screenshot shows the Azure portal interface for a runbook named "Start-DiskClean". The top navigation bar includes "Home", "Automation Accounts", "ArcBox-Automation | Runbooks", and the specific runbook name. Below the title, there are several tabs: "Runbook" (selected), "View", "Edit", "Link to schedule", "Add webhook", "Delete", "Export", "Feedback", and "Refresh". A search bar is at the top left. On the left, a sidebar lists "Overview", "Activity log", "Tags", "Diagnose and solve problems", "Resources" (Jobs, Schedules, Webhooks), and "Runbook settings" (Properties, Description). The main content area is titled "Essentials" and displays resource group ("arcbox-demo-rg"), account ("ArcBox-Automation"), location ("East US"), subscription ("Demo"), and tags ("Project : jumpstart\_arcbox"). To the right, detailed information is shown: Subscription ID, Status, Runbook type, Runtime version, and Last modified. Below this, a section for "Recent Jobs" shows a single entry: "Status" (Created), "Last updated" (9/9/2023, 7:58 AM), and a note "No jobs found."

8. [] Select *Hybrid Worker* and select *linux-workers* under *Choose Hybrid Worker group*. Click *OK*



9. [] Click on the *Output* tab and wait for the job to finish. You should notice that the amount of free space has increased after the cleanup action has completed.

Home > ArcBox-Automation | Runbooks > Start-DiskClean (ArcBox-Automation/Start-DiskClean) >

## Start-DiskClean 09/09/2023, 08:51

 Job

▷ Resume  Stop || Suspend  Refresh

^ Essentials

Id : fa7388d2-652f-4d18-b7d1-8e6b7fecb215	Created : 09/09/2023, 08:51:33
Status : Completed	Last Update : 09/09/2023, 08:52:13
Ran on : linux-workers	Runbook : <a href="#">Start-DiskClean</a>
Ran As : User	Source snapshot : <a href="#">View source snapshot</a>

Input **Output** Errors Warnings All Logs Exception

```
Free disk space before cleanup action
Filesystem      1M-blocks  Used Available Use% Mounted on
tmpfs           392       2     391   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv  48701 11559  34951 25% /
tmpfs           1956      1     1956  1% /dev/shm
tmpfs           5        0      5   0% /run/lock
/dev/sda2        974     253     654  28% /boot
/dev/sda1        511      7     505  2% /boot/efi
tmpfs           392       1     392   1% /run/user/1000

Free disk space after cleanup action
Filesystem      1M-blocks  Used Available Use% Mounted on
tmpfs           392       2     391   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv  48701 11555  34955 25% /
tmpfs           1956      1     1956  1% /dev/shm
tmpfs           5        0      5   0% /run/lock
/dev/sda2        974     253     654  28% /boot
/dev/sda1        511      7     505  2% /boot/efi
tmpfs           392       1     392   1% /run/user/1000
```

[!hint] The provided runbook is a starting point for cleaning a single directory. Additional logic and directories may be added as required for specific scenarios. For example, it is also possible to add logic to connect to other machines in order to perform cleanup actions on those.

10. [] Next, you will be running the same runbook on a Windows machine. Navigate back to the runbook overview page for *Start-DiskClean* and click *Start*.

Home > Automation Accounts > ArcBox-Automation | Runbooks >

## Start-DiskClean (ArcBox-Automation/Start-DiskClean)

 Runbook | Directory: Contoso

 Start  Edit  Add webhook  Delete  Export  Refresh

Search

Overview

Resource group : [arbox-demo-rg](#) Subscription ID : 87b8def0-f5cf-402e-a8db-10e0ee958565

Activity log Status : Published

Tags Runbook type : PowerShell

Diagnose and solve problems Runtime version : 7.2 (preview)

Recent Jobs Last modified : 9/9/2023, 7:58 AM

Resources

Jobs Tags (edit) : Project : jumpstart\_arbox

Schedules

Webhooks

Runbook settings

Properties

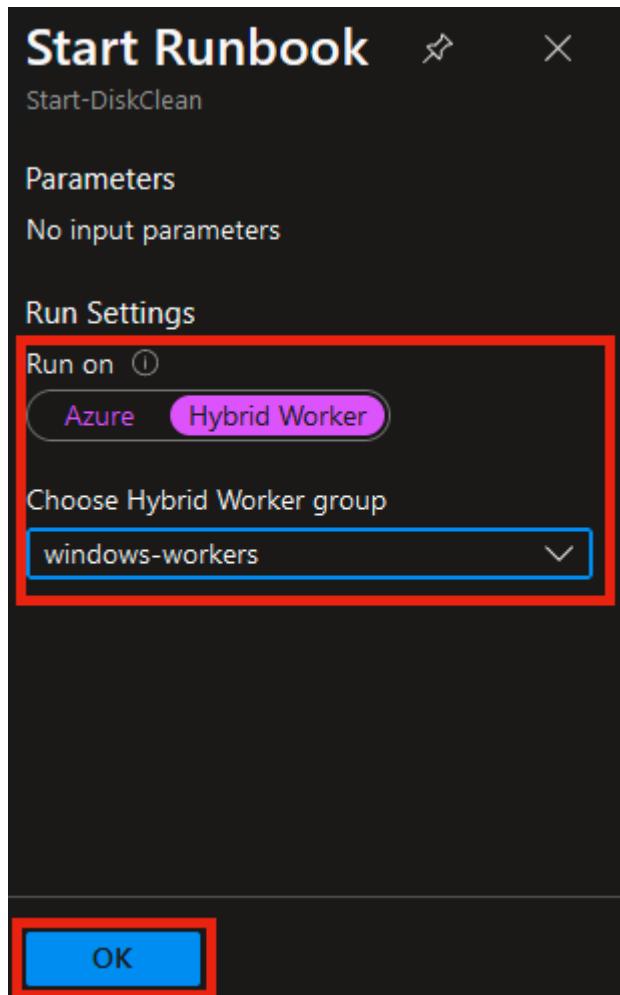
Description

Recent Jobs

Status Created Last updated

No jobs found.

11. [] Select *Hybrid Worker* and select *windows-workers* under *Choose Hybrid Worker group*, then Click *OK*.



12. [] Click on the *Output* tab and wait for the job to finish.

- The cleanup action may run for a few minutes, so feel free to continue and revisit the job output later.
- When completed, you should notice that the amount of free space has increased after the cleanup action has completed.

 Start-DiskClean 11/09/2023, 22:44

Job

▷ Resume  Stop || Suspend  Refresh

Essentials

Id : 7d578905-66da-44ea-99cb-74fa7e3a8f38	Created : 11/09/2023, 22:44:53
Status : Completed	Last Update : 11/09/2023, 22:54:35
Ran on : windows-workers	Runbook : <a href="#">Start-DiskClean</a>
Ran As : User	Source snapshot : <a href="#">View source snapshot</a>

Input **Output** Errors Warnings All Logs Exception

```
Free disk space before cleanup action
```

DriveLetter	FriendlyName	FileSystemType	DriveType	HealthStatus	OperationalStatus	SizeRemaining	Size
C		NTFS	Fixed	Healthy	OK	81.04 GB	99.37 GB

```
Windows Update component store cleanup
```

```
Deployment Image Servicing and Management tool
Version: 10.0.20348.681
```

```
Image Version: 10.0.20348.1787
```

```
[===== 10.0% ]  
[===== 19.0% ]  
[===== 20.0% ]  
[===== 20.0% ]
```

```
Free disk space after cleanup action
```

DriveLetter	FriendlyName	FileSystemType	DriveType	HealthStatus	OperationalStatus	SizeRemaining	Size
C		NTFS	Fixed	Healthy	OK	82.42 GB	99.37 GB

## Task 5 has been completed

---

Click **Next** for the next exercise or [Go back to the main table of content](#)

====

# Exercise 2 - Securely Connect to your Azure Arc-enabled servers using SSH access

---

## Objective

Enable SSH based connections to Arc-enabled servers without requiring a public IP address or additional open ports.

## Estimated Time to Complete This Lab

20 minutes

## Explanation

This feature enables SSH based connections to Arc-enabled servers without requiring a public IP address or additional open ports. In this exercise, you will learn how to enable and configure this functionality. At the end, you will interactively explore how to access to Arc-enabled Windows and Linux machines.

====

## Task 1: Install prerequisites on client machine

---

It is possible to leverage both Azure CLI and Azure PowerShell to connect to Arc-enabled servers. Choose the one to use based on your own preferences.

1. [] RDP into the *ArcBox-Client* VM.
2. [] Open PowerShell and install either the Azure CLI extension or the Azure PowerShell modules based on your preference of tooling.

### Azure CLI

```
az extension add --name ssh
```

or

### Azure PowerShell

```
Install-Module -Name Az.Ssh -Scope CurrentUser -Repository PSGallery  
Install-Module -Name Az.Ssh.ArcProxy -Scope CurrentUser -Repository PSGallery
```

[!hint] We recommend that you install the tools on the ArcBox Client virtual machine, but you may also choose to use your local machine if you want to verify that the Arc-enabled servers is reachable from any internet-connected machine after performing the tasks in this module.

### Task 1 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 2 - Enable SSH service on Arc-enabled servers

---

[!hint] We will use two Arc-enabled servers running in ArcBox for this module:

- *ArcBox-Win2K22*
- *ArcBox-Ubuntu-01*

1. [] RDP into the *ArcBox-Client* VM
2. [] Open Hyper-V Manager
3. [] Right click *ArcBox-Win2K22* and select Connect twice
4. [] Login to the operating system using username Administrator and the password you used when deploying ArcBox, by default this is **JS123!!**
5. [] Open PowerShell and install OpenSSH for Windows by running the following:

```
# Install the OpenSSH Server
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

# Start the sshd service
Start-Service sshd

# Configure the service to start automatically
Set-Service -Name sshd -StartupType 'Automatic'

# Confirm the Windows Firewall is configured to allow SSH. The rule should be
# created automatically by setup. Run the following to verify:
if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction
SilentlyContinue | Select-Object Name, Enabled)) {
    Write-Output "Firewall Rule 'OpenSSH-Server-In-TCP' does not exist, creating
it..."
    New-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -DisplayName "OpenSSH Server
(sshd)" -Enabled True -Direction Inbound -Protocol TCP -Action Allow -LocalPort 22
} else {
    Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and
exists."
}
```

6. [] Close the connection to *ArcBox-Win2K22*
7. [] Right click *ArcBox-Ubuntu-01* in Hyper-V Manager and select Connect
8. [] Login to the operating system using username **jumpstart** and the password **JS123!!**
9. [] Run the command `systemctl status ssh` to verify that the SSH service is active and running
10. [] Close the connection to *ArcBox-Ubuntu-01*

### Task 2 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 3 - Connect to Arc-enabled servers

- From the *ArcBox-Client* VM, open a PowerShell session and use the below commands to connect to **ArcBox-Ubuntu-01** using SSH:

## Azure CLI

```
$serverName = "ArcBox-Ubuntu-01"  
$localUser = "jumpstart"  
$resourceGroup = "ArcBox"  
az ssh arc --resource-group $resourceGroup --name $serverName --local-user  
$localUser
```

or

## Azure PowerShell

```
$serverName = "ArcBox-Ubuntu-01"  
$localUser = "jumpstart"  
$resourceGroup = "ArcBox"  
Enter-AzVM -ResourceGroupName $resourceGroup -Name $serverName -LocalUser  
$localUser
```

- The first time you connect to an Arc-enabled server using SSH, you will see the following prompt:

Port 22 is not allowed for SSH connections in this resource. Would you like to update the current Service Configuration in the endpoint to allow connections to port 22? If you would like to update the Service Configuration to allow connections to a different port, please provide the -Port parameter or manually set up the Service Configuration. (y/n)

- It is possible to pre-configure this setting on the Arc-enabled servers by following the steps in the section *Enable functionality on your Arc-enabled server* in the [documentation](#). However, for this exercise, type **yes** and press Enter to proceed.

```
PS C:\Users\arcdemo> az ssh arc --resource-group $Env:resourceGroup --name $serverName --local-user $localUser  
The authenticity of host 'arcbox-ubuntu-01 (<no hostip for proxy command>)' can't be established.  
ECDSA key fingerprint is SHA256:tmUJUPFfw5dsjgXPZ2KRmanuo24aVLuHZ2GkHEmusuc.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'arcbox-ubuntu-01' (ECDSA) to the list of known hosts.  
arcdemo@arcbox-ubuntu-01's password: -
```

```
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-137-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Mon 23 Jan 2023 07:40:10 AM UTC

 System load: 0.61          Processes:           143
 Usage of /: 42.3% of 47.56GB  Users logged in:      0
 Memory usage: 26%          IPv4 address for eth0: 10.10.1.104
 Swap usage: 0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

15 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
```

```
Last login: Fri Jan 20 21:06:23 2023 from 127.0.0.1
arcdemo@arcbox-ubuntu-01:~$ -
```

4. [] Following the previous method, connect to *ArcBox-Win2K22* via SSH.

## Azure CLI

```
$serverName = "ArcBox-Win2K22"
$localUser = "Administrator"
$resourceGroup = "ArcBox"
az ssh arc --resource-group $resourceGroup --name $serverName --local-user
$localUser
```

or

## Azure PowerShell

```
$serverName = "ArcBox-Win2K22"
$localUser = "Administrator"
$resourceGroup = "ArcBox"
Enter-AzVM -ResourceGroupName $resourceGroup -Name $serverName -LocalUser
$localUser
```

```
PS C:\Users\arcdemo> az ssh arc --resource-group $Env:resourceGroup --name $serverName --local-user $localUser
The authenticity of host 'arcbox-win2k22 (<no hostip for proxy command>)' can't be established.
ECDSA key fingerprint is SHA256:RCu06BLhtQ8lgr50FXW6h2BXRT2xocNWbDZEq2K8IzU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'arcbox-win2k22' (ECDSA) to the list of known hosts.
Administrator@arcbox-win2k22's password: -
```

```
Microsoft Windows [Version 10.0.20348.1366]
(c) Microsoft Corporation. All rights reserved.

administrator@ARCBOX-WIN2K22 C:\Users\Administrator>
```

5. [] In addition to SSH, you can also connect to the Azure Arc-enabled servers, Windows Server virtual machines using **Remote Desktop** tunneled via SSH.

## Azure CLI

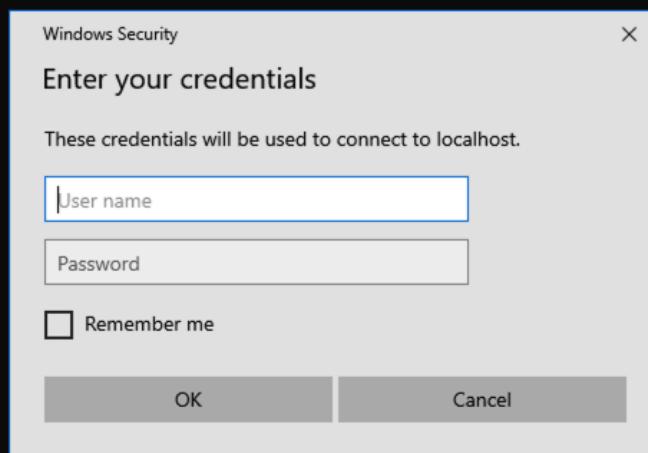
```
$serverName = "ArcBox-Win2K22"
$localUser = "Administrator"
$resourceGroup = "ArcBox"
az ssh arc --resource-group $resourceGroup --name $serverName --local-user
$localUser --rdp
```

or

## Azure PowerShell

```
$serverName = "ArcBox-Win2K22"
$localUser = "Administrator"
$resourceGroup = "ArcBox"
Enter-AzVM -ResourceGroupName $resourceGroup -Name $serverName -LocalUser
$localUser -Rdp
```

```
PS C:\Users\arcdemo> az ssh arc --resource-group $Env:resourceGroup --name $serverName --local-user $localUser --rdp
RDP feature is in preview.
The authenticity of host 'arcbox-win2k22 (<no hostip for proxy command>)' can't be established.
ECDSA key fingerprint is SHA256:RCU06BLhtQ8lgr50FXW6h2BXRT2xocNLwbDZEq2K8IzU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Administrator@arcbox-win2k22's password:
Launching Remote Desktop Connection
To close this session, close the Remote Desktop Connection window.
```



**Task 3 has been completed**

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 4 - Optional: Azure Entra ID based SSH Login

1. [] The *Azure AD based SSH Login – Azure Arc VM extension* can be added from the extensions menu of the Arc server in the Azure portal. The Azure AD login extension can also be installed locally via a package manager via: `apt-get install aadsshlogin` or the following command:

```
$serverName = "ArcBox-Ubuntu-01"
$resourceGroup = "ArcBox"
az connectedmachine extension create --machine-name $serverName --resource-group
$resourceGroup --publisher Microsoft.Azure.ActiveDirectory --name AADSSHLogin --
--type AADSSHLoginForLinux --location $env:azureLocation
```

2. [] Configure role assignments for the Arc-enabled server *ArcBox-Ubuntu-01* using the Azure portal. Two Azure roles are used to authorize VM login:

- **Virtual Machine Administrator Login:** Users who have this role assigned can log in to an Azure virtual machine with administrator privileges.
- **Virtual Machine User Login:** Users who have this role assigned can log in to an Azure virtual machine with regular user privileges.

3. [] After assigning one of the two roles to your personal Azure Entra ID user account, run the following commands on the *ArcBox-Client* to connect to *ArcBox-Ubuntu-01* using SSH and Entra ID-based authentication (you can use either Azure CLI or PowerShell) :

### Azure CLI

```
# Log out from the Service Principal context
az logout
# Log in using your personal account
az login
$serverName = "ArcBox-Ubuntu-01"
$resourceGroup = "ArcBox"
az ssh arc --resource-group $resourceGroup --name $serverName
```

### Azure PowerShell

```
# Log out from the Service Principal context
Disconnect-AzAccount
# Log in using your personal account
Connect-AzAccount
```

```
$serverName = "ArcBox-Ubuntu-01"  
$resourceGroup = "ArcBox"  
Enter-AzVM -ResourceGroupName $resourceGroup -Name $serverName
```

4. [] You should now be connected and authenticated using your Azure Entra ID account. Verify this by running:

```
whoami
```

5. [] Exit the ssh session by typing *exit*.

### **Exercise 2 has been completed**

====

## **Exercise 3 - Run PowerShell and Shell scripts on Azure Arc-enabled servers using the Run command**

### **Objective**

Run PowerShell and Shell commands on Arc-enabled Windows and Linux servers.

### **Estimated Time to Complete This Lab**

30 minutes

### **Explanation**

The Run command feature uses the Connected Machine agent to remotely run PowerShell scripts within an Azure Arc-connected Windows machine and Shell scripts within an Azure Arc-connected Linux machine. This capability is useful in all scenarios where you want to run a script within an Arc-connected machine. It allows you to troubleshoot and remediate a machine that doesn't have the RDP or SSH port open because of improper network or administrative user configuration.

**[!alert] Use PowerShell 7.0 on the ArcBox-Client to run the commands in the rest of this Exercise. Or alternatively you can use Azure Cloud Shell but you might need to deal with some Powershell module compatibility settings.**

====

## **Task 1: Check the pre-requisites**

1. [] Check the version of the Azure CLI extension "connectedmachine" on your ArcBox-Client machine using the following command:

```
az extension list --query "[?name=='connectedmachine'].version"
```

2. [] If the "connectedmachine" extension does not exist, or if the version of the Azure CLI extension is older than "1.0.0" then remove the old version (if it exists) and install the new one:

```
#Remove the old version of the extension  
az extension remove --name connectedmachine
```

```
#Install latest version of the extension  
az extension add --name connectedmachine --allow-preview True
```

3. [] Check if the installed version of module Az.ConnectedMachine is 1.0.0 or higher. Use the following PowerShell command to check the installed version:

```
Get-Module -Name Az.ConnectedMachine
```

4. [] If you need to install the latest version then use the following PowerShell command:

```
Install-Module -Name Az.ConnectedMachine -Force
```

### Task 1 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 2: Use the Run command to execute a simple PowerShell script within an Arc-connected Windows machine

1. [] Run the following Azure CLI command after adding the appropriate resource group, name of the Arc-connected machine, a name identifying the command (**without spaces**), and the location of your Arc-connected machine:

```
az connectedmachine run-command create --resource-group <Resource Group Name> --  
machine-name <Machine Name> --run-command-name <Identifying Name of command> --  
script "Write-Host 'Hello World'" --location <Location>
```

For example:

```
az connectedmachine run-command create --resource-group ArcBox --machine-name  
ArcBox-Win2K22 --run-command-name HelloWorld --script "Write-Host 'Hello World'" -  
-location uksouth
```

[!alert] If you receive an error message about the command not being recognized or that a preview version is needed then go back to the **Check the pre-requisites** step and follow the instructions to first remove the *connectedmachine* extension, then to install the extension with **--allow-preview True** option.

After a couple of minutes the result is returned. If the execution is successful then you will see the following within the longer returned JSON string:

```
"executionState": "Succeeded",  
"exitCode": 0,  
"output": "Hello World",
```

2. [] You can also execute the script using a PowerShell command. Add the appropriate variable values to the following command then run it:

```
New-AzConnectedMachineRunCommand -ResourceGroupName "<Resource Group Name>" -  
Location "<Location>" -SourceScript "Write-Host 'Hello World'" -RunCommandName "  
<Identifying Name of command>" -MachineName "<Machine Name>"
```

The successful execution of the PowerShell command will show the following output:

```
InstanceViewExecutionMessage      : RunCommand script execution completed  
InstanceViewState                 : Succeeded  
InstanceViewExitCode              : 0  
InstanceViewOutput                : Hello World
```

### Task 2 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 3: Use the Run command to execute a simple Shell command within an Arc-connected Linux machine

1. [] Run the following Azure CLI command after adding the appropriate parameters:

```
az connectedmachine run-command create --resource-group <Resource Group Name> --  
machine-name <Machine Name> --run-command-name <Identifying Name of command> --  
script "ifconfig" --location <Location>
```

Or in PowerShell:

```
New-AzConnectedMachineRunCommand -ResourceGroupName "<Resource Group Name>" -  
Location "<Location>" -SourceScript "ifconfig" -RunCommandName "<Identifying Name  
of command>" -MachineName "<Machine Name>"
```

If the execution is successful then you should have an output that includes the result of the ifconfig command as a string. Notice that the line breaks are indicated by the "\n" string:

```
"output": "eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500\n      inet 10.10.1.102 netmask 255.255.255.0  
broadcast 10.10.1.255\n      inet6 fe80::215:5dff:fe01:403 prefixlen 64 scopeid 0x20<link>\n      ether 00:15:5d:01:  
04:03 txqueuelen 1000 (Ethernet)\n      RX packets 27507 bytes 26632258 (26.6 MB)\n      RX errors 0 dropped 0 ove  
rruns 0 frame 0\n      TX packets 13246 bytes 4237820 (4.2 MB)\n      TX errors 0 dropped 0 overruns 0 carrier 0 c  
ollisions 0\n      nlo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536\n      inet 127.0.0.1 netmask 255.0.0.0\n      inet6 ::1  
prefixlen 128 scopeid 0x10<host>\n      loop txqueuelen 1000 (Local Loopback)\n      RX packets 10160 bytes 1743552  
(1.7 MB)\n      RX errors 0 dropped 0 overruns 0 frame 0\n      TX packets 10160 bytes 1743552 (1.7 MB)\n      T  
X errors 0 dropped 0 overruns 0 carrier 0 collisions 0\n      ",  
},
```

### Task 3 has been completed

Click **Next** for the next task or [go back to the main table of content](#)

====

## Task 4: Direct the output of a Run command to Azure storage blob

---

1. [] Create a storage account (if you do not have one) using the following command after filling in the required parameters:

```
az storage account create --name <Storage account name> --resource-group <Resource  
Group Name> --location <Location> --sku Standard_LRS --kind storageV2 --allow-  
blob-public-access false
```

2. [] Create a storage container to which you will direct the output of the run command:

```
az storage container create --name <container name> --account-name <Storage  
account name> --auth-mode login
```

3. [] Create a blob SAS URI with the following permissions: Read, Write, Create, delete, and append. You will need an end date for the validity of the SAS token, for example one day from the current date. Also, to be able to use the SAS URI in our run command you will need to remove any double quotes from the beginning and the end of the generated URI.

```
$end=(Get-date).AddDays(1) | Get-Date -UFormat '+%Y-%m-%dT%H:%MZ'  
$sasuri = $(az storage blob generate-sas --account-name <storage account name> --  
container-name <storage container name> --name <name of blob for command output  
destination - it will be created if it doesn't exist> --permissions acdrw --expiry  
$end --full-uri)
```

4. [] Execute the following run command which runs a PowerShell script within the Arc-enabled Windows machine. The run command directs the output to the append blob:

```
az connectedmachine run-command create --resource-group "<Resource Group Name>" --  
location "<Location>" --script "Get-Process | Sort-Object CPU -desc | Select-  
Object -first 5" --name "<Identifying Name of command>" --machine-name "<Machine  
Name>" --output-blob-uri $sasuri
```

Or in PowerShell:

```
New-AzConnectedMachineRunCommand -ResourceGroupName "<Resource Group Name>" -  
Location "<Location>" -SourceScript "Get-Process | Sort-Object CPU -desc | Select-  
Object -first 5" -RunCommandName "<Identifying Name of command>" -MachineName "  
<Machine Name>" -OutputBlobUri $sasuri
```

5. [] Examine the storage container in the Azure portal or using the Azure storage explorer. Look for the output of the command in the blob specified by the SAS URI used in the run command. The output should be the top five processes for CPU usage in the machine.

The screenshot shows the Azure Storage Explorer interface. On the left, a sidebar lists container settings like 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', 'Shared access tokens', 'Access policy', 'Properties', and 'Metadata'. The main area displays two blobs: 'outputblob' and 'outputblob5'. The 'outputblob5' blob's properties are shown in detail, including its type as 'Append blob' and size of '841 B'. A 'File Activity' section shows metrics for handles, NPM(K), PM(K), WS(K), CPU(s), ID, and SI ProcessName. A separate window titled 'outputblobs' is overlaid, showing a table of file activity data.

	Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	ID	SI	ProcessName
1	868	106	267060	212660	151.33	3316	0	MsMpEng
2	394	27	51096	66948	48.70	2000	0	gc_service
3	424	21	26948	25996	43.80	1512	0	himds
4	886	43	82228	115608	35.08	3388	0	MsSense
5	363	14	12496	17632	13.09	1580	0	svchost

## Task 4 has been completed

## Exercise 2 has been completed

---

Click **Next** for the next lab or [Go back to the main table of content](#)