Welcome student, WorkshopPLUS - Azure Arc

Workshop Type

This workshop contains many labs that are not possible to fully complete during the assigned duration with your instructor. It is important to understand what your target is (i.e. which selection of labs you aim to complete within the assigned duration). The targeted lab selection depends on whether your workshop is a Closed or Open one.

A closed workshop usually means that all attendees are from the same organization and that a scoping discussion has been carried out in advance to decide which labs will be focussed on during the workshop. In contrast, an open workshop means that the attendees are from different organizations with no scoping discussion in advance of the workshop delivery.

Please select the option that applies to you. If you are not sure please ask your instructor: is this an open workshop? openws

For Open Workshops The recommended set of labs to attempt during an open workshop are listed below (The first two labs must be carried out before attempting any other labs).

- Lab01: Lab environment deployment - **must be carried out**
- Lab02: Onboarding Windows and Linux Servers to Azure Arc -**must be carried out**
- Lab03: Securing Azure Arc-enabled servers with Microsoft Defender for Cloud
- Lab04: Governance across Azure Arc-enabled servers using Azure Policy and Azure Machine Configuration
- Lab05: Monitor your Azure Arc-enabled servers using Azure Monitor, Change Tracking and Inventory
- Lab07: Keep your Azure Arc-enabled servers patched using Azure Update Manager
- Lab08: Enrol your Windows Server 2012/R2 machines for Extended Security Updates with Azure Arc

We understand that some of the attendees will be experienced in Azure technologies and with the tools used in this workshop. In that case, these users might be able to finish the recommended labs and still have time to attempt the other labs if they wish to do so. Please note that your instructor might prioritize helping attendees who are attempting the recommended labs so please be patient when asking for help. You can also attempt the labs after the end of the workshop within the time allowed for your environment (consult you instructor if you are not sure).

For Closed Workshops A scoping activity should have been carried out by your instructor with people in your organization to understand your focus areas. This activity should have resulted in an agenda (a minimum set of labs) that you are expected to execute within the workshop. If you do not know this agenda already then consult your instructor. Please follow the suggested labs in your customized agenda first. If you have time after completing those labs, then you can move on to other labs that are of interest to you. Please note that your instructor might prioritize helping attendees who are attempting the labs in the customized agenda, so please be patient when asking for help. You can also attempt the labs after the end of the workshop within the time allowed for your environment (consult you instructor if you are not sure).

Target Audience

This lab guidance is Level 300 primarily intended for IT professionals, system administrators, and cloud architects seeking to modernize their infrastructure and adopt or optimize their hybrid cloud strategy with Azure Arc. Through those labs, you'll:

- Establish basic level of proficiency in Azure Arc Technologies
- Gain hands-on experience in Azure Arc technology
- Get knowledge on applying Azure native services to Arc-enabled assets with:
    - Security and Governance
    - Management
    - Monitoring and Automation.

Foreword

If you have any questions or the need for more clarity on what is happening during the lab, please don't hesitate to talk to the trainer.

Good luck!

Lab Manual

Table of Contents

LAB01: Lab environment deployment

Student Lab Manual

Table of Contents

Exercise 1: Get familiar with your workshop environment

Objective

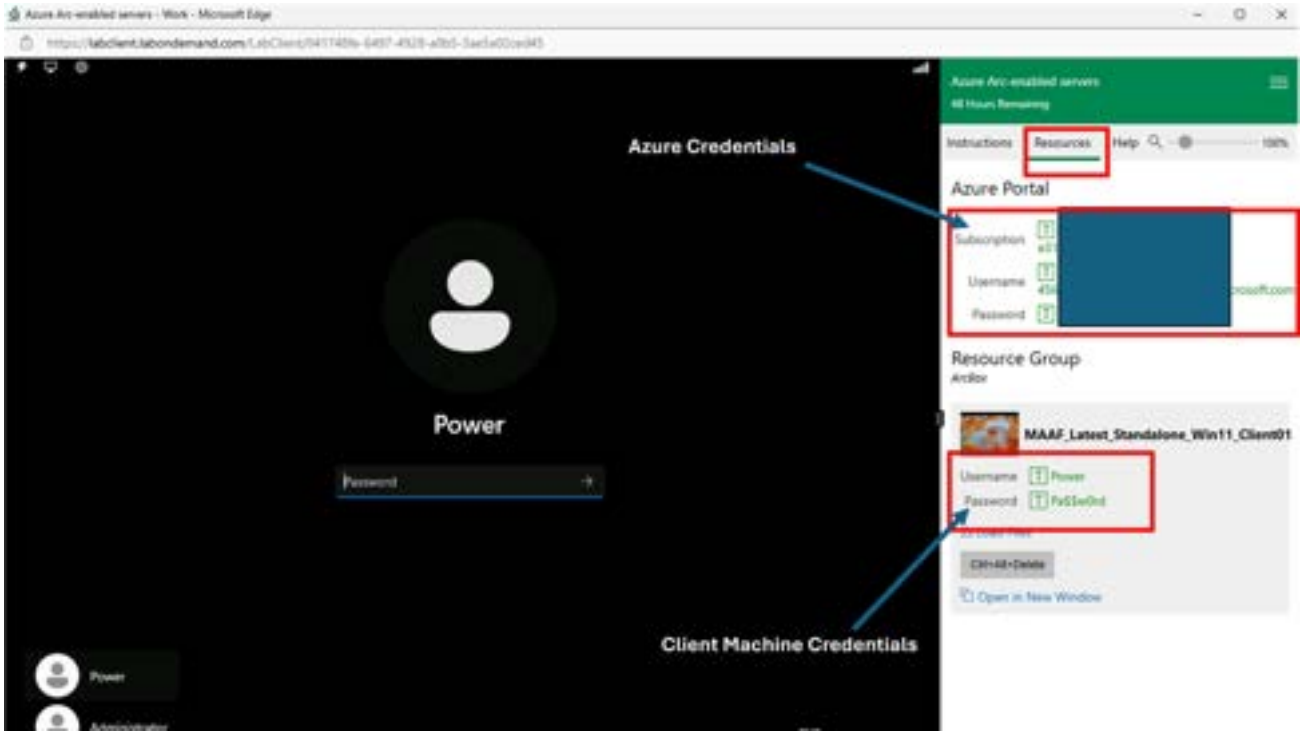
This lab will guide you through the necessary steps to set up your Azure environment.

Estimated Time to Complete This Lab

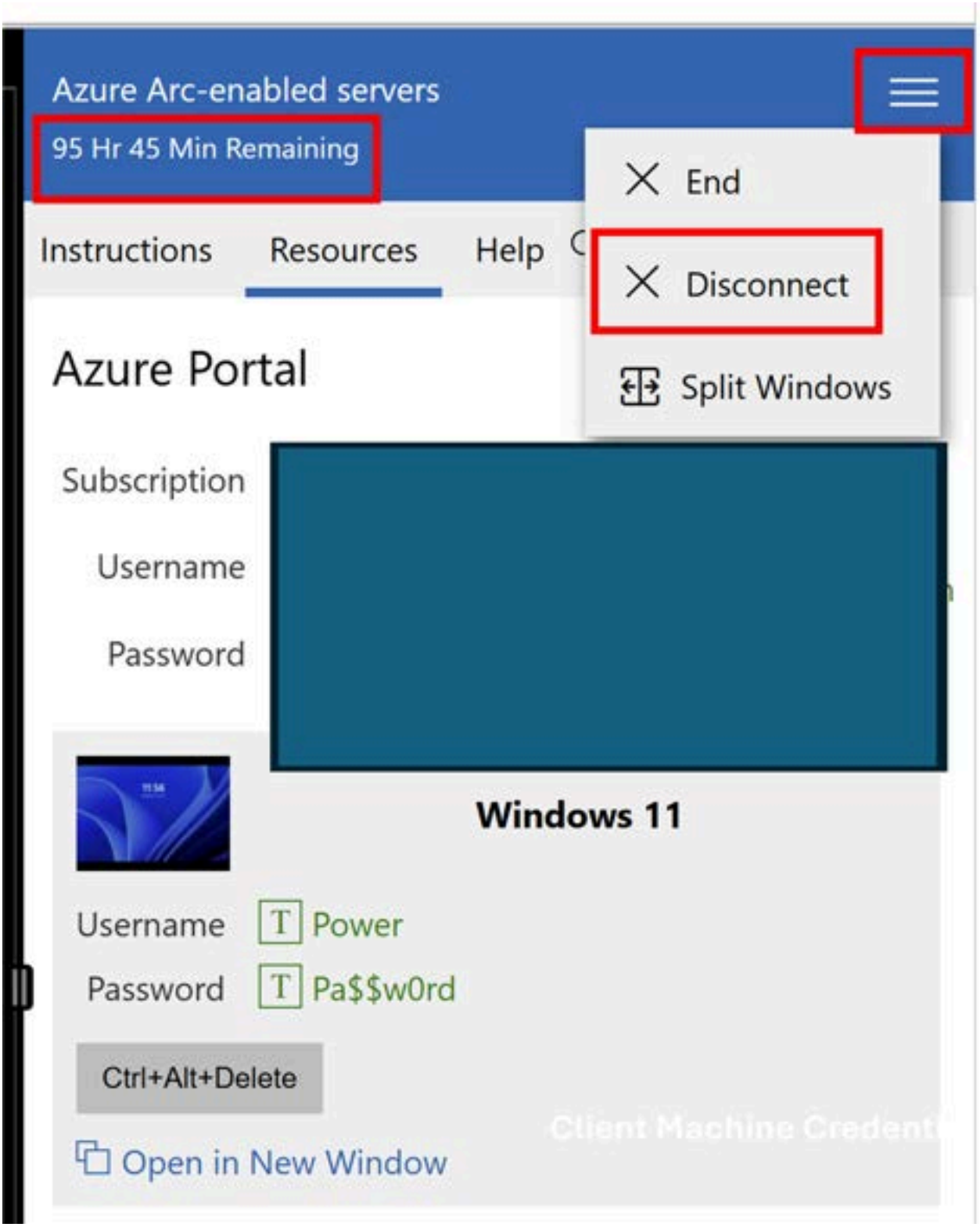
15 minutes

Task 1: View your workshop credentials and log into the workshop machine

This lab is based on a Windows Client machine provisioned for you in the learning platform. At the top of this Instructions window, there is a menu bar with three items: **Instructions**, **Resources** and **Help** as well as a Zoom function. On the *Resources* tab you should see both the local lab machine credentials and also the Azure credentials.



- [ ] Do not use the **END** option as it will destroy your Azure resources and end your lab. Use the **Disconnect** option when you want to take a break from the lab. Your lab timer shows how long you have left before your lab is terminated.



- Notice that on this tab, and throughout the instructions for this lab, you will find **squares with a capital T inside the square**. **This option can be used to have the lab type text for you**. If your cursor is in the password box of the login screen, and if you click on the square box with the T in it, the lab will type the password for you. This typing can be used throughout the lab and the text will appear wherever your cursor is located. **However, please note that this might not work for**

copying into Azure CloudShell and you will need to use Copy and Paste with the mouse right-click.

Now log into the client machine and start with the first task of this lab. Have fun!

Exercise 2 - Lab pre-requisites

Objective

This exercise will walk you through preparing your local development environment to deploy the lab environment using the ArcBox sandbox.

Estimated Time to Complete This Lab

15 minutes

Explanation

ArcBox is a solution that deploys a complete sandbox environment with Windows and Linux machines simulating an on-premises environment. One of the Windows machines will be running a SQL Server. You will Arc-enable those machines and explore the different capabilities provided through Azure Arc. You will deploy ArcBox using the Azure Portal.

Task 1: Prepare the local development environment for lab deployment and log into Azure

Important: Make sure that you carry out the steps in the Windows machine provided by the lab environment and not your local PC.

1. Make sure that you have the x64 version of Azure CLI version 2.66.0 or above. Start PowerShell as administrator and run the following command:

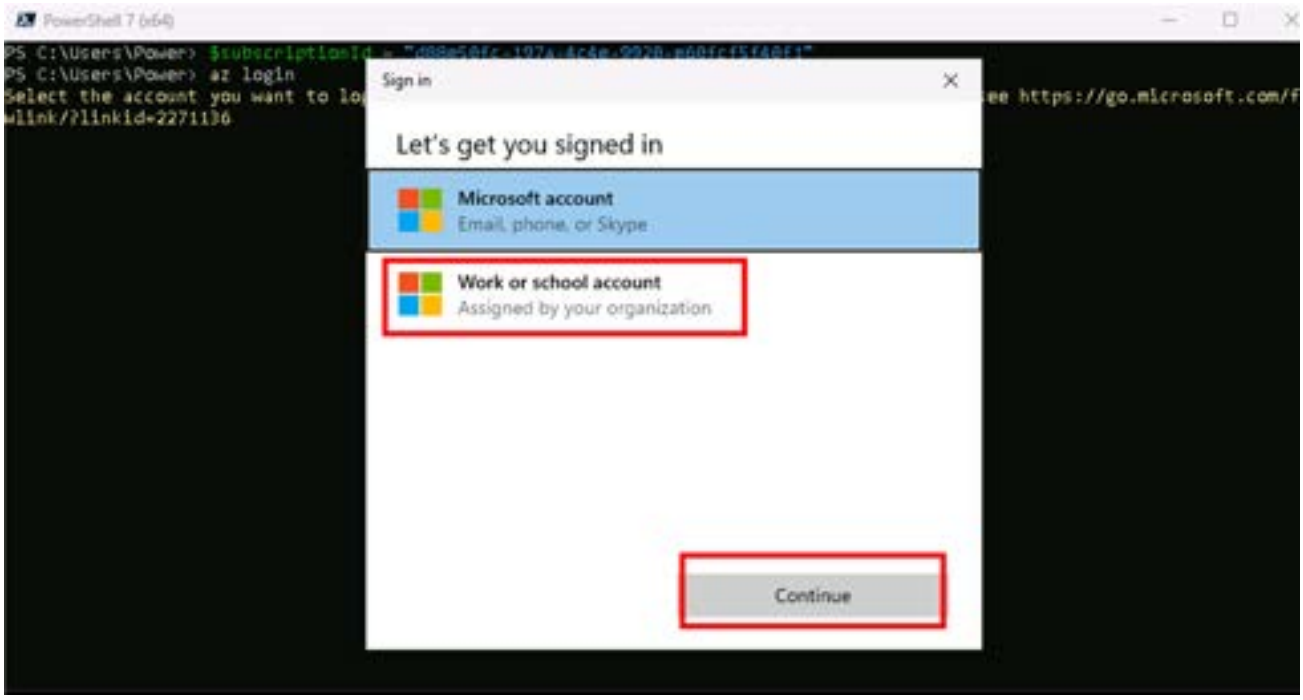
```shell
az --version
```

The output should confirm that you have the Azure CLI installed. If not then you can install it using the following command.

```PowerShell
$ProgressPreference = 'SilentlyContinue'; Invoke-WebRequest -Uri https://aka.ms/
```

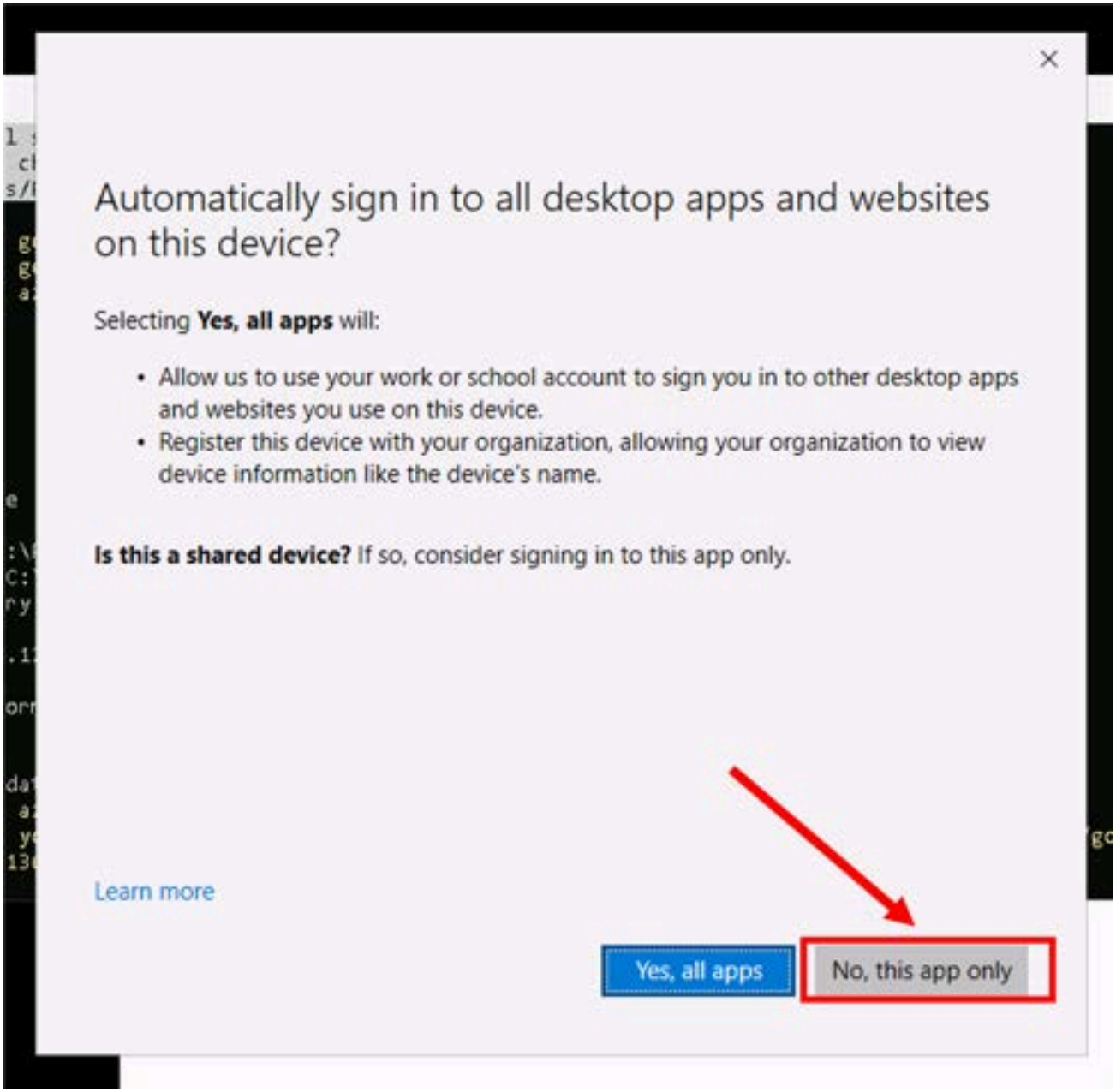
2. Login to AZ CLI using the `az login` command. If you are prompted to choose an account type then select *Work or school account*.

```shell
az login
```



- [ ] **You can use Notepad or Visual Studio Code for the rest of the commands to be able to edit the needed parameters in an easier way than the PowerShell console**

- [ ] 3. If you are prompted to *Autmoatically sign in to all desktop apps and websites on this device* select ***No, this app only***



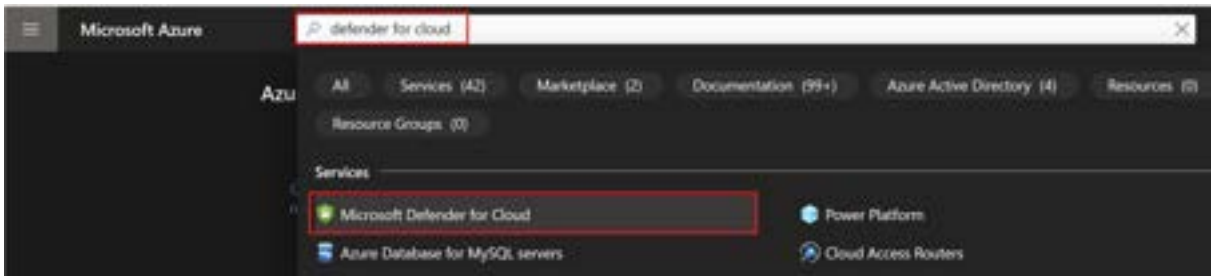
Task 1 has been completed

Task 2: Enable Defender for Servers on your subscription

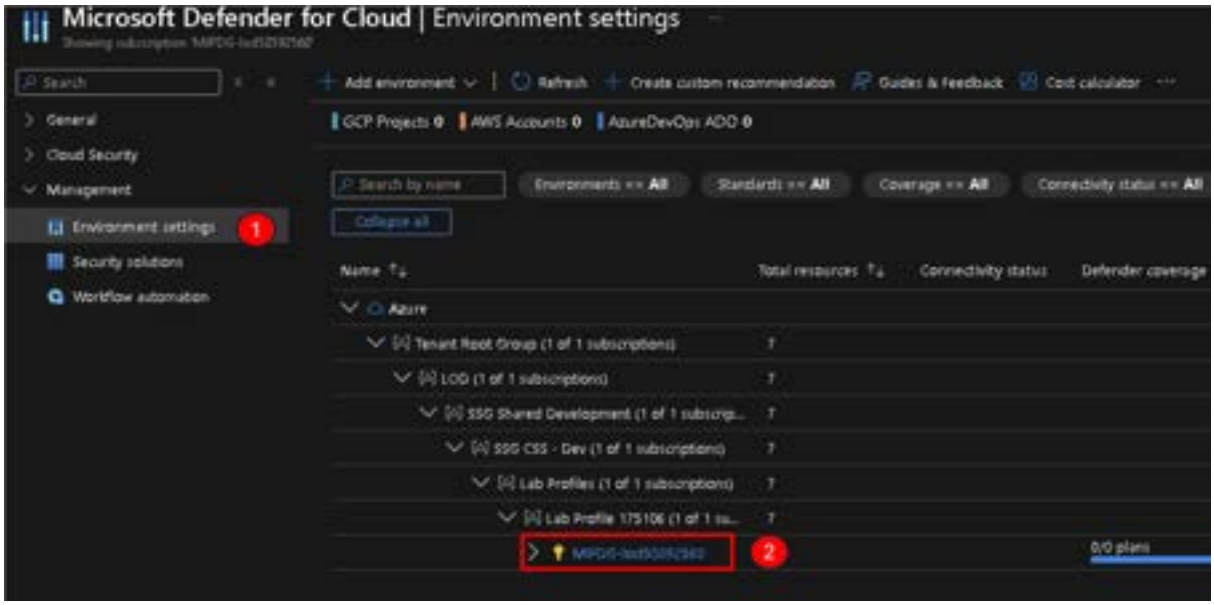
1. Open the EDGE browser and navigate to the Azure portal then log into Azure with lab provided credentials from the **Resources** tab

```shell
https://portal.azure.com
```

2. From the Azure home page, search for defender and select Microsoft Defender for Cloud.

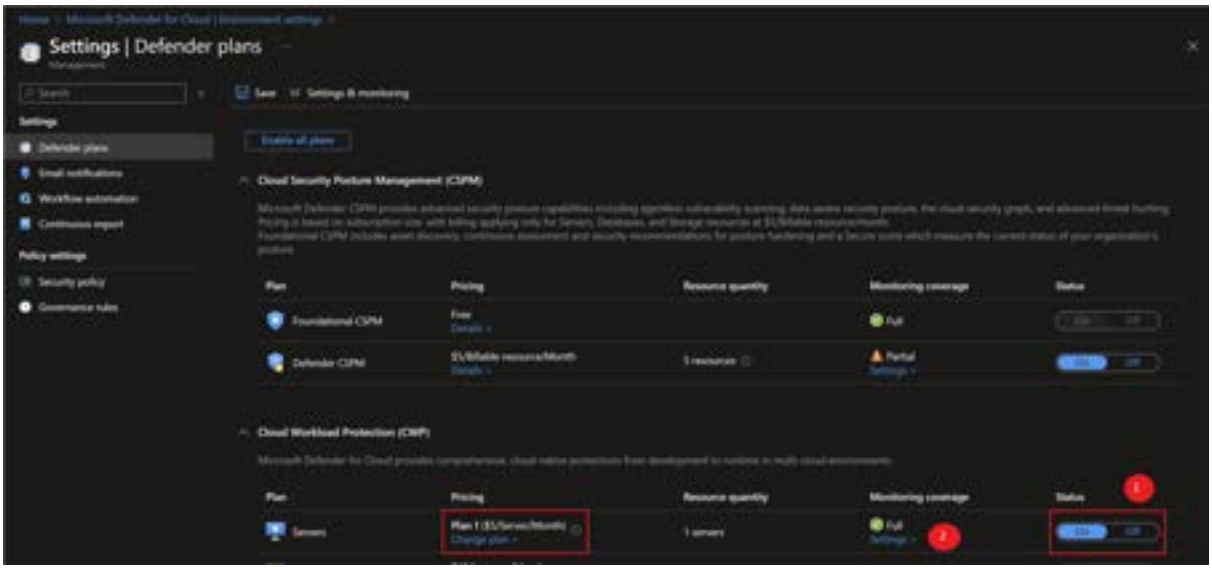


3. If you already have Defender plans setup at your subscription level, you may find that Defender is already turned on for your Arc-enabled servers. However, if Defender is not enabled, select *Environment settings* from the Management section on the left blade.



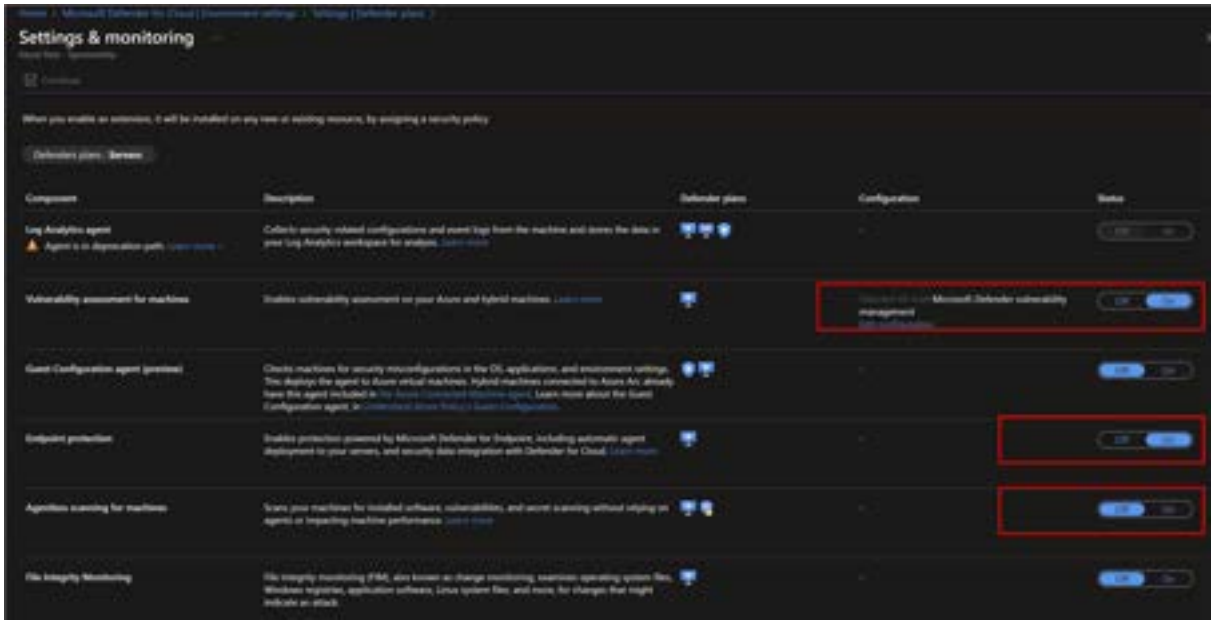
4. Expand the Tenant Root Group, and then select **your subscription**.

5. Enable the plan for servers, you can select either *Plan 1* or *Plan 2* for this exercise

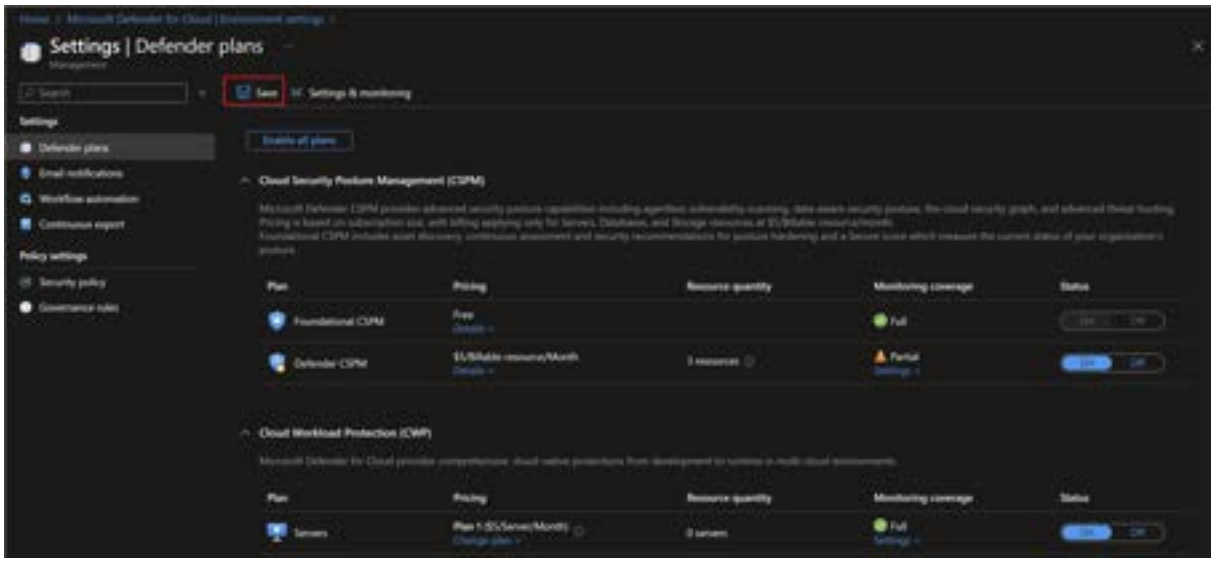


6. Click on the settings option in the *Monitoring coverage* and set the following capabilities as indicated:

    - Vulnerability assessment for machines - **ON**.
    - Endpoint protection - **ON**.
    - Agentless Scanning for machines - **OFF**.



7. Click *Continue* then Click *Save*.



Task 2 has been completed

Exercise 3 - ArcBox deployment

Objective

In this exercise you will deploy the lab using the ArcBox solution which will deploy all the needed Azure resources to your subscription to be able to execute all the upcoming exercises.

Estimated Time to Complete This Lab

45 minutes

Explanation

ArcBox is a solution that deploys a complete sandbox environment with Windows and Linux machines simulating an on-premises environment. One of the Windows machines will be running a SQL Server. You will Arc-enable those machines and explore the different capabilities provided through Azure Arc. You will deploy ArcBox using the Azure Portal.

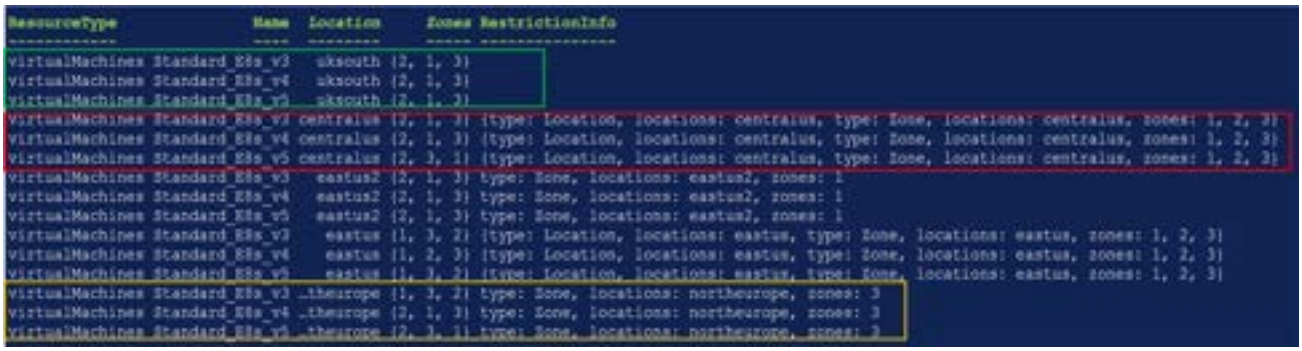
Task 1: Deploy ArcBox using the Azure Portal

1. The specific setup of ArcBox used in this workshop can be deployed in any region that supports the Azure virtual machine SKU required (Standard_E8s_v5, v4 or v3). Occasionally, there might be restrictions on specific subscriptions that can prevent the deployment of a specific SKU even if it is available in a region. To check if there are any restrictions on your subscription you can run the following PowerShell command in **Azure Cloud Shell**. You can change the list of locations in the script if necessary by adding or removing regions. But **do not change the list of VM Skus**. If you prefer to run the command from the lab machine then make sure you login in PowerShell first using *Connect-AzAccount* command.

PowerShell

```
$locations = "uksouth", "northeurope", "eastus", "eastus2", "centralus"
#Do not change the VMSkus below
$VmSkus = "Standard_E8s_v5", "Standard_E8s_v4", "Standard_E8s_v3"
$locations | ForEach-Object -ThrottleLimit 10 -Parallel {
    Get-AzComputeResourceSku -Location $_  | Where-Object {$_.ResourceType -eq "
}
```

The output will show if there are restrictions in any of the listed regions or in specific zones in the region (for a multi-zone region). Choose a region and VM Sku combination that has no restrictions to deploy the lab VM. For example, in the following diagram we can see that *uksouth* is a very good candidate to deploy the lab. On the other hand *centralus* is not because of the full restrictions. The *northeurope* region has restrictions only in zone 3 so it should allow you to deploy.



Make sure that you have selected a region with no restrictions on the required virtual machine sku (as explained above). Failure to do so might cause your deployment to fail. You will need to enter the name of the selected region and VM SKU in the deployment template in the next step.

2. Open the Microsoft Edge browser and paste the following URL.

T

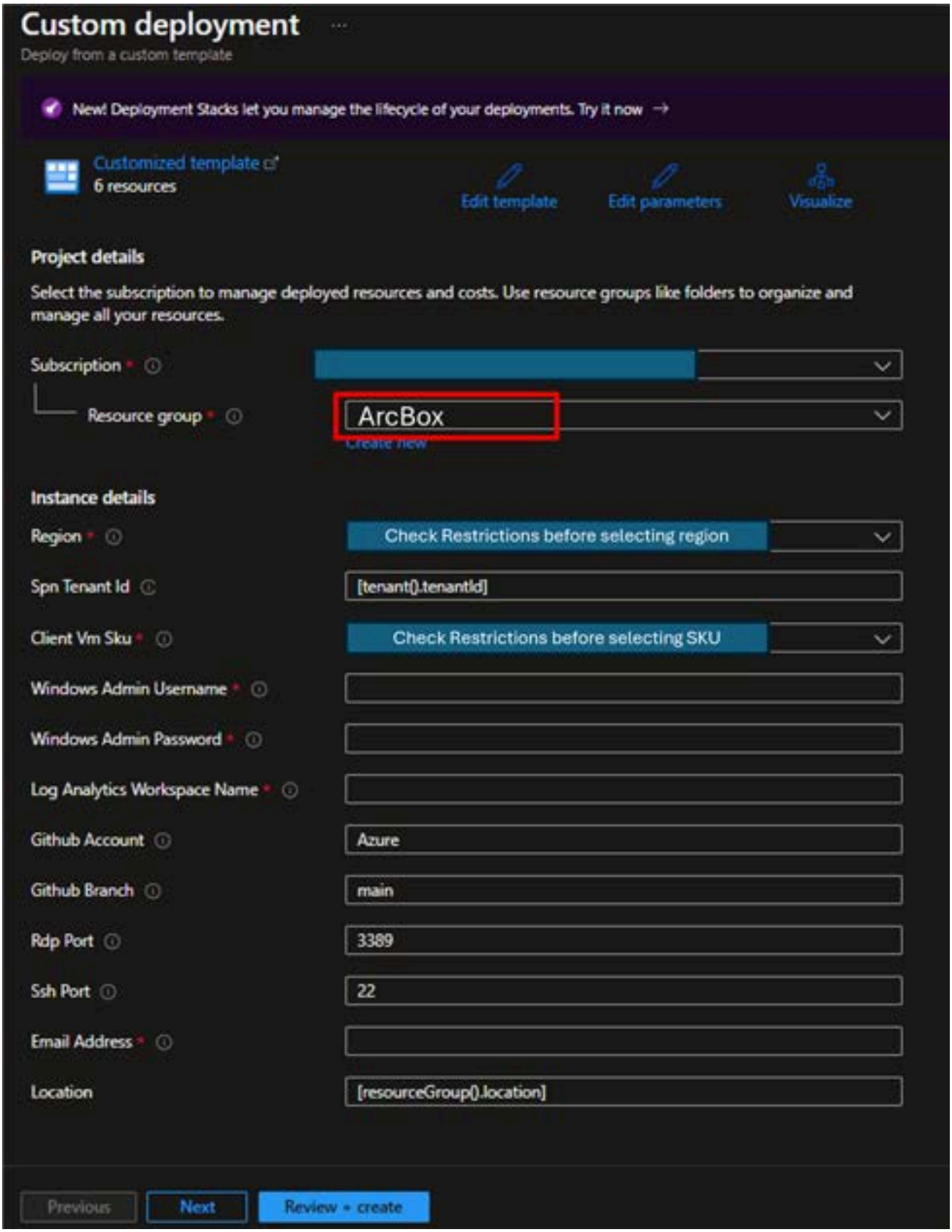
https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzur

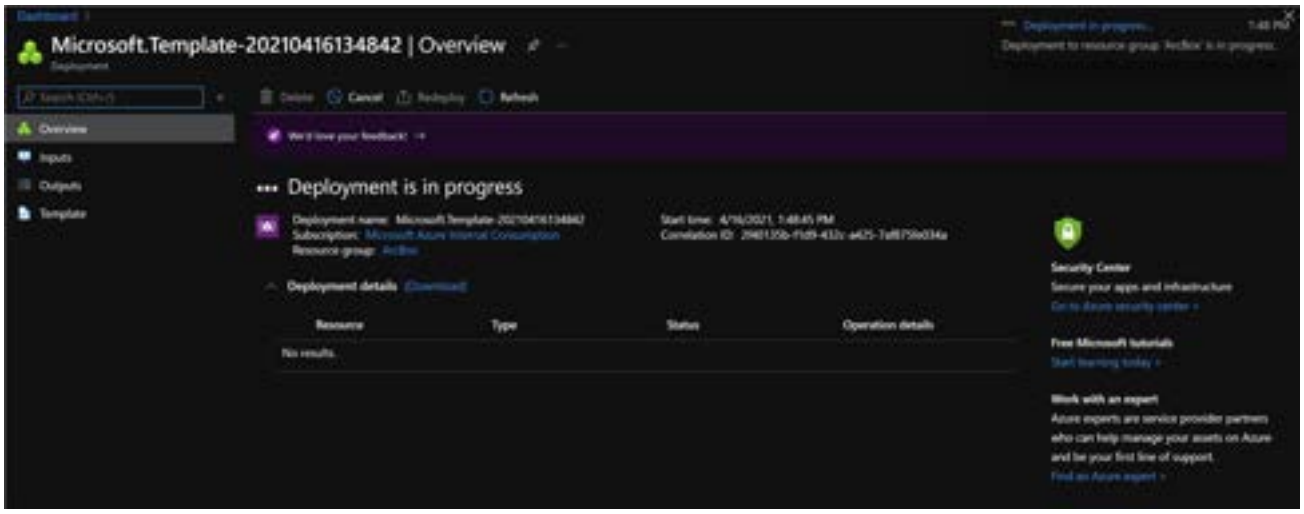
3. Enter values for the the template parameters then click *Review + create*.

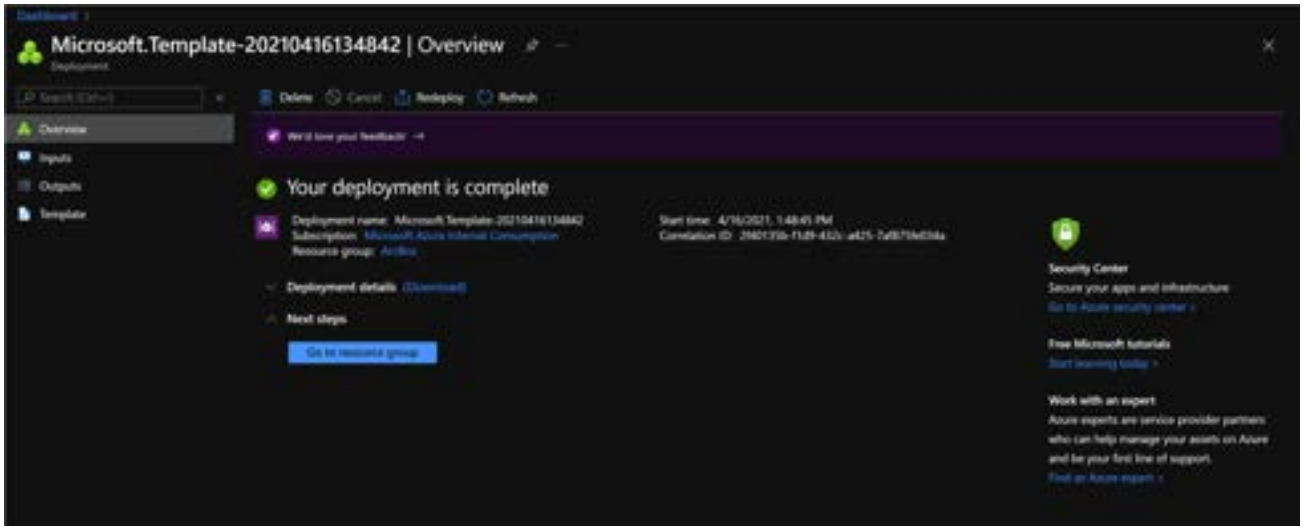
- [ ] **The only acceptable resource group name is *ArcBox*. If you choose any other name then your deployment will fail**.

- [ ] Make sure to note the Windows admin username and password as you will use them in later exercises. In order to avoid using a username and password combination that does not meet the complexity requirements which would lead to the failure of the deployment, we recommend using the username **arcdemo** and the password **Arcboxlabs@12345**.

- [ ] Review the complexity requirements for the Windows virtual machine's password


Custom deployment
Deploy from a custom template
New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →
Customized template
6 resources
Edit template
Edit parameters
Visualize
Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription *
Resource group *
ArcBox
Create new
Instance details
Region *
Check Restrictions before selecting region
Spn Tenant Id
[tenant().tenantId]
Client Vm Sku *
Check Restrictions before selecting SKU
Windows Admin Username *
Windows Admin Password *
Log Analytics Workspace Name *
Github Account
Azure
Github Branch
main
Rdp Port
3389
Ssh Port
22
Email Address *
Location
[resourceGroup().location]
Previous
Next
Review + create





- [ ] The deployment takes around 20 minutes to complete.

- [ ] If you see any failure in the deployment, please check that you have selected a region with no restrictions on the virtual machine sku and consult your workshop instructor. In some cases a transient issue might cause a deployment error. The best action in these cases is to redeploy the template. **Make sure that you delete the whole resource group (not only the resources inside it) and once the deletion is completed you can redeploy the template**.

Task 1 has been completed

Task 2: Connecting to the ArcBox Client virtual machine

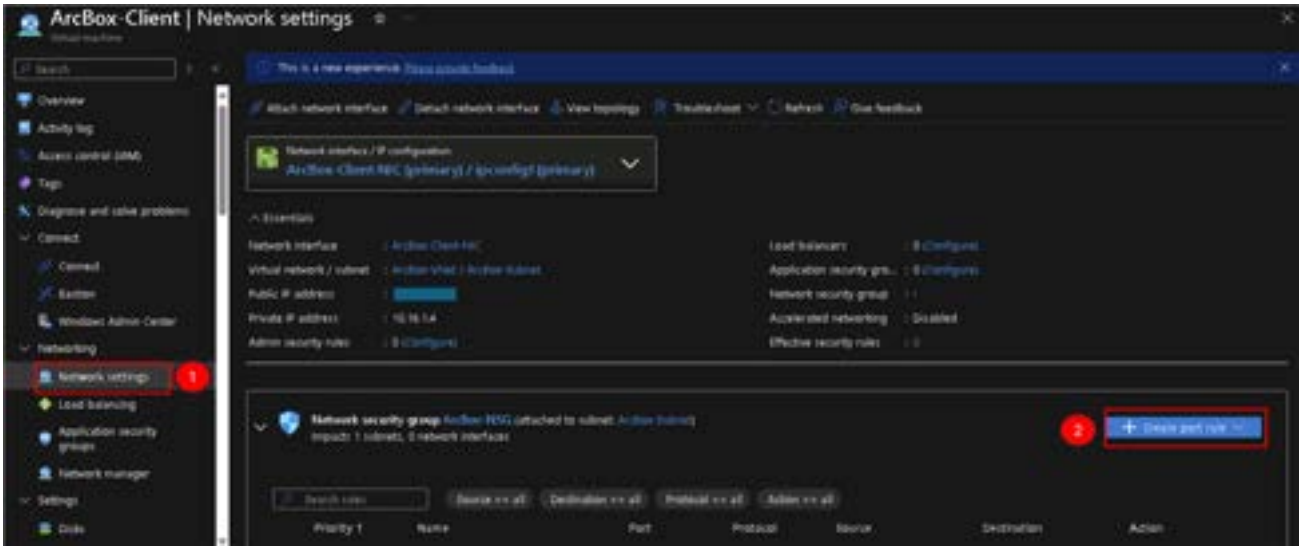
Various options are available to connect to *ArcBox-Client* VM on Azure. However, for this lab you will use the direct RDP method - available after configuring access to port 3389 on the *ArcBox-NSG* Network Security Group.

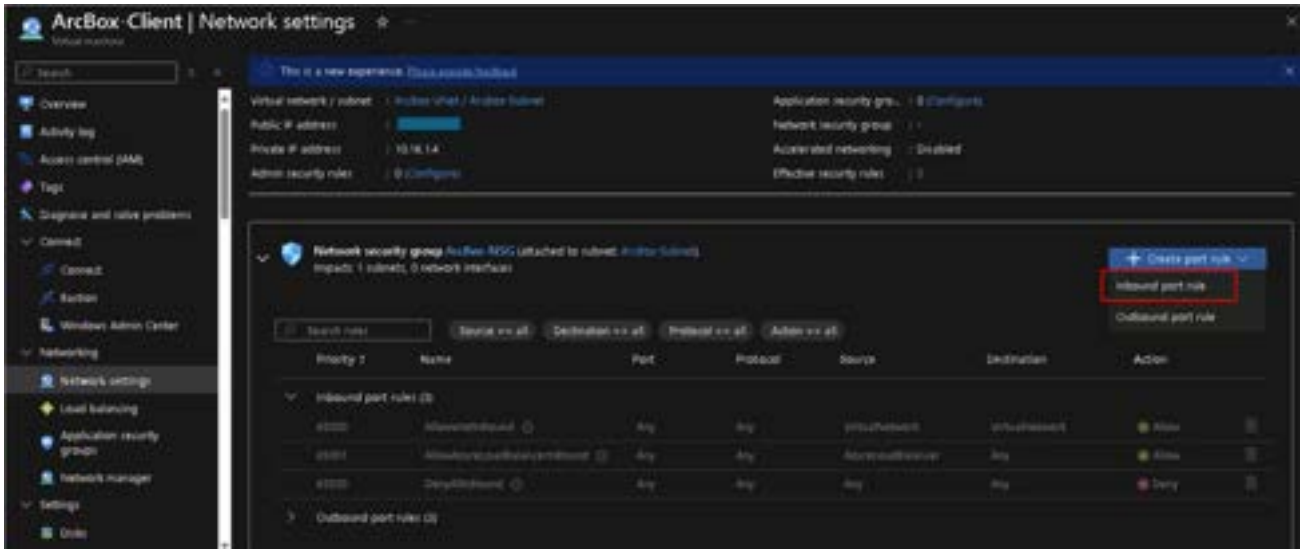
- [ ] **Once you have logged on to the *ArcBox-Client* VM you might see some Powershell scripts in the the process of being executed. DO NOT close any windows or stop any scripts. The set-up automation will close the completed windows automatically**

Connecting directly with RDP

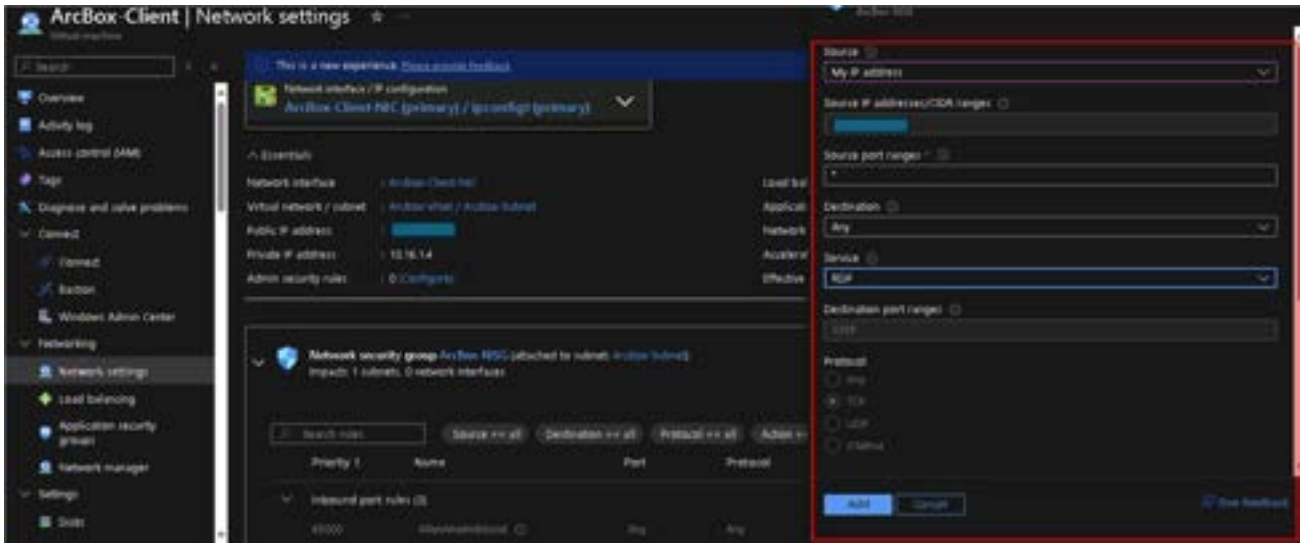
By design, ArcBox does not open port 3389 on the network security group. Therefore, you must create an NSG rule to allow inbound 3389.

- [ ] 1. Open the *Network settings* for the *ArcBox-Client* machine in the Azure portal and click *Create port rule* to add a new inbound rule.

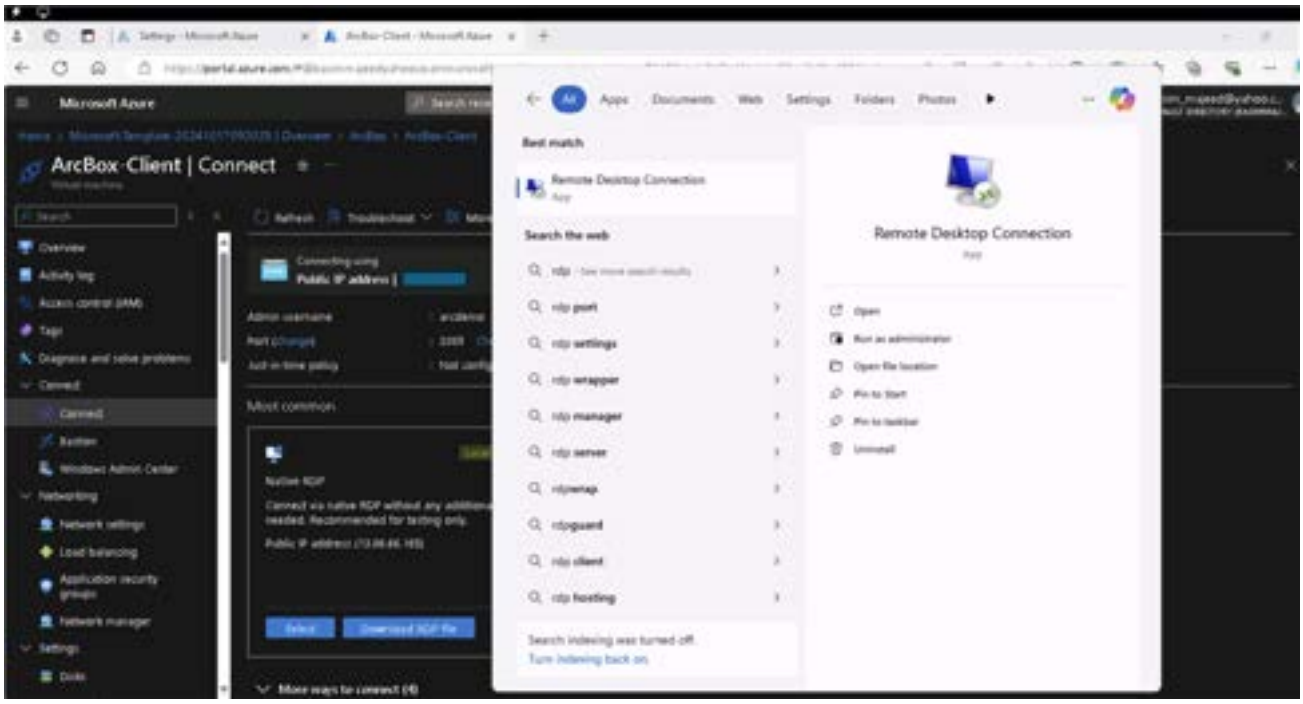




2. Specify the IP address that you will be connecting from (or use *My IP address*) and select RDP as the service. If you need to retrieve your public IP address (not required if you use the *My IP address* setting) then you can do so by accessing https://icanhazip.com or https://whatismyip.com.



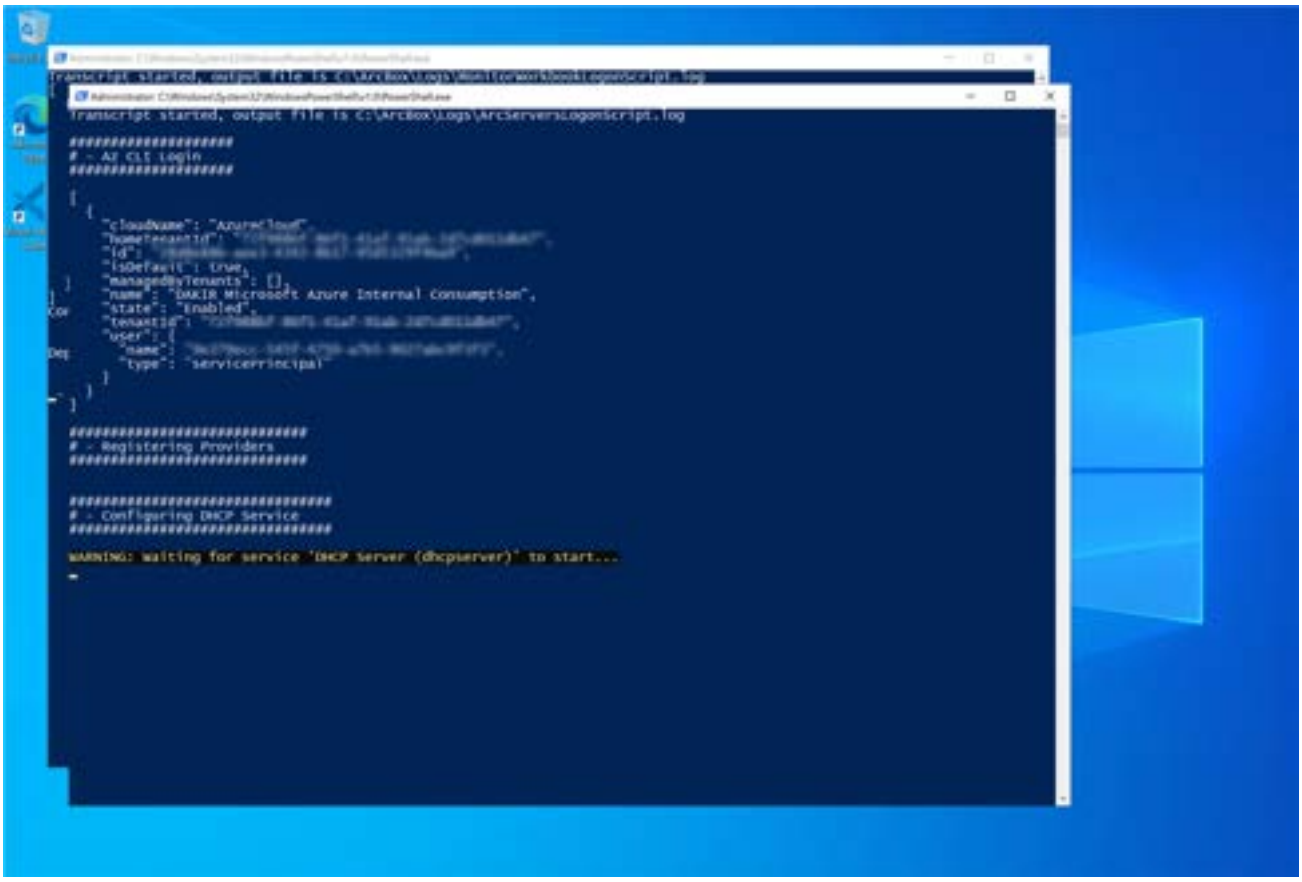
3. Now you can run the Remote Desktop Connection App and connect to the *ArcBox-Client* machine using the user name and password that you have set on the deployment template.



Task 2 has been completed

Task 3: Post-deployment automation

1. Once you log into the *ArcBox-Client* VM, multiple automated scripts might be open and running. These scripts usually take 10-20 minutes to finish (**Do not close any windows and do not stop any running scripts!**), and once completed, the script windows will close automatically. At this point, the deployment is complete.



2. Deployment is complete when you see the following screen background! Let's begin exploring the features of Azure Arc-enabled servers with the other labs in this workshop.

Task 3 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB02: Onboarding Windows and Linux Servers to Azure Arc

Student Lab Manual

Table of Contents

Exercise 1 - Examine the existing Arc-connected machines

Objective

The deployment process that you have walked through in Lab01 should have set up a number of VMs running on Hyper-V in the ArcBox-Client machine. Two of these machines have been connected to Azure Arc for you by the set script. In this exercise you will verify that these two machines are indeed Arc-enabled and you will identify the other machines that you will Arc-enable.

Estimated Time to Complete This Lab

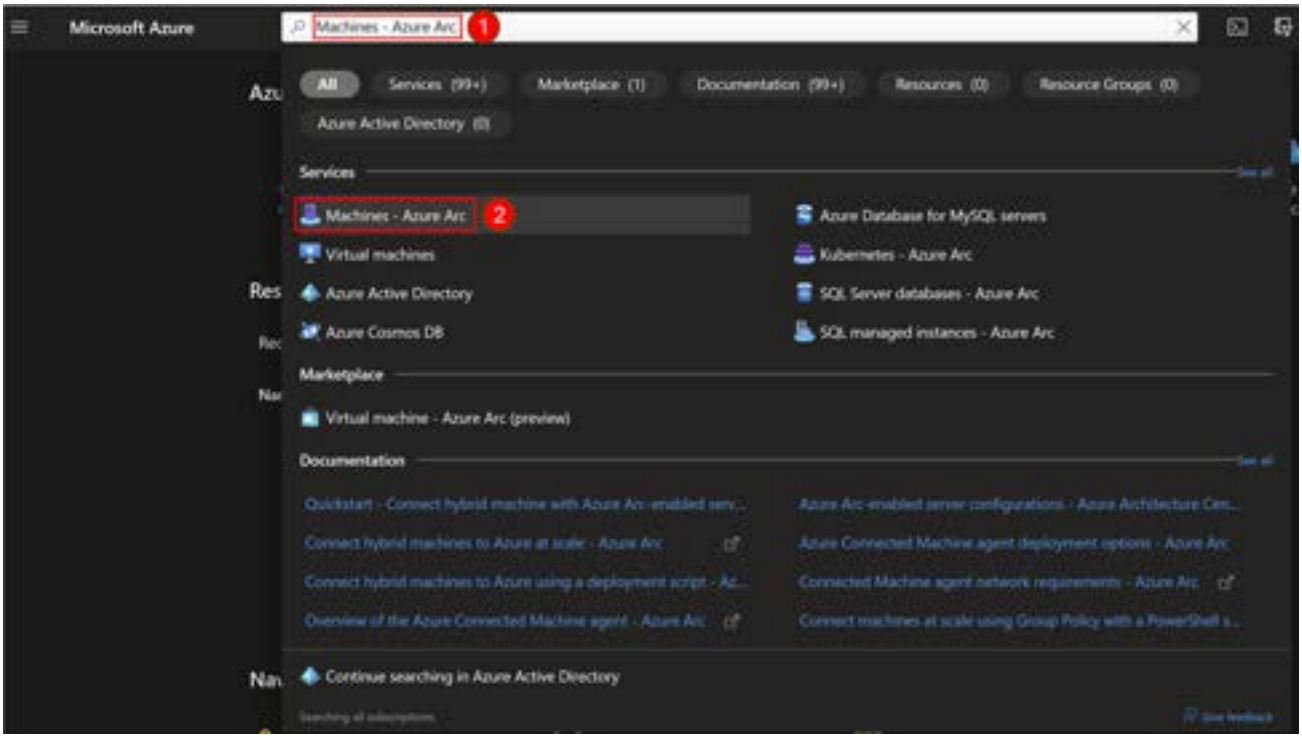
10 minutes

Explanation

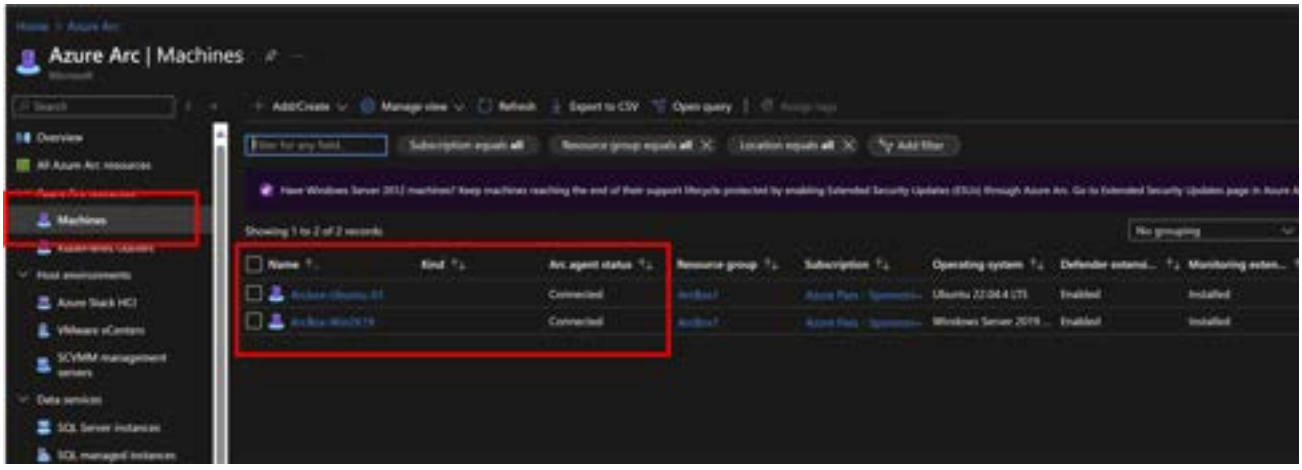
The Arc-connected machines can be viewed on the Azure portal.

Task 1: Use the Azure portal to examine you Arc-enabled machines inventory

1. Enter "Machines - Azure Arc" in the top search bar in the Azure portal and select it from the displayed services.



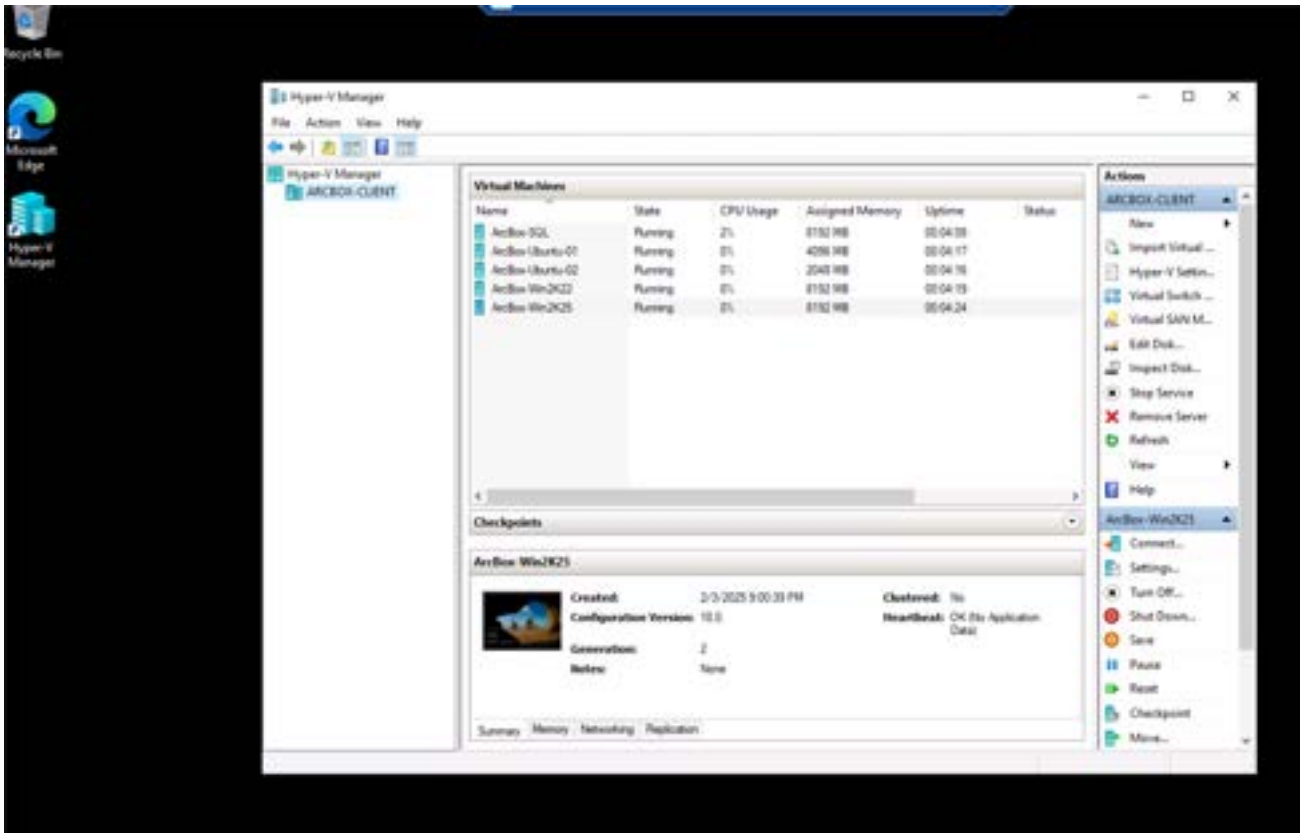
2. You should see the machines that are connected to Arc already: *Arcbox-Ubuntu-01* and *ArcBox-Win2K25*.



Task 1 has been completed

Task 2: Examine the virtual machines that you will Arc-enable

1. You want to connect 2 more machines running as VMs in the ArcBox-Client. You can see these (ArcBox-Win2K22 and ArcBox-Ubuntu-02) by running the Hyper-V Manager in the ArcBox-Client (after you have connected to it with RDP as explained earlier in Lab01).



Task 2 has been completed

Exercise 2 - Onboard a Windows and a Linux machine to Azure Arc

Objective

In this exercise you will onboard the Windows machine ArcBox-Win2K22 and the Linux machine ArcBox-Ubuntu-02 to Azure Arc using the Service Principal onboarding method.

Estimated Time to Complete This Lab

60 minutes

Explanation

Connecting machines in your hybrid environment directly with Azure can be accomplished using different methods, depending on your requirements and the tools you prefer to use. One method to connect the machines to Azure Ar is to use a Microsoft Entra ID service principal. This service principal is a special limited management identity that has only the minimum permission necessary to connect machines to Azure. This method is safer than using a higher privileged account like a Subscription Contributor and follows access control security best practices. The service principal is used only during onboarding; it is not used for any other purpose.

The Azure Connected Machine agent for Linux and Windows communicates outbound securely to Azure Arc over TCP port 443. By default, the agent uses the default route to the internet to reach Azure services. You can optionally configure the agent to use a proxy server if your network requires it. There are a number of URLs that must be available in order to install and use the Connected Machine agent and onboard the machine to Azure Arc. The proxy must be configured to allow all these URLs including ones containing wildcards. However, the number of URLs can be consolidated using the Azure Arc Gateway service which reduces the number of URLs to a small set of endpoints that need to be allowed by the proxy.

In Task 1 of this exercise you will learn how to onboard a Windows machine to Azure Arc using direct internet access (without a proxy).

You will then learn how to onboard a Linux machine to Azure Arc using either the direct internet access (without a proxy) in task 2 or using a proxy together with the Azure Arc Gateway in task 3.

- ☐ **For onboarding the Linux machine to Azure Arc, it is important to follow either Task 2 or Task 3.**

Documentation

Azure Connected Machine agent deployment options.

Connect hybrid machines to Azure at scale.

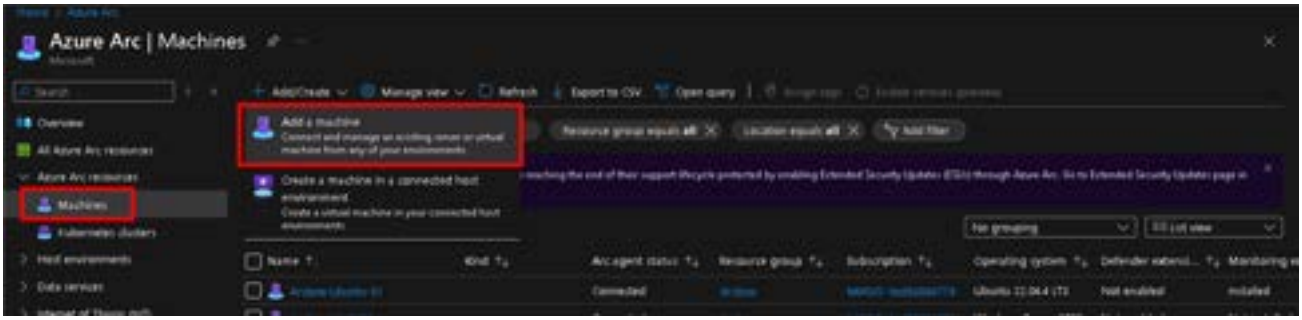
Connected Machine agent network requirements

Simplify network configuration requirements with Azure Arc gateway

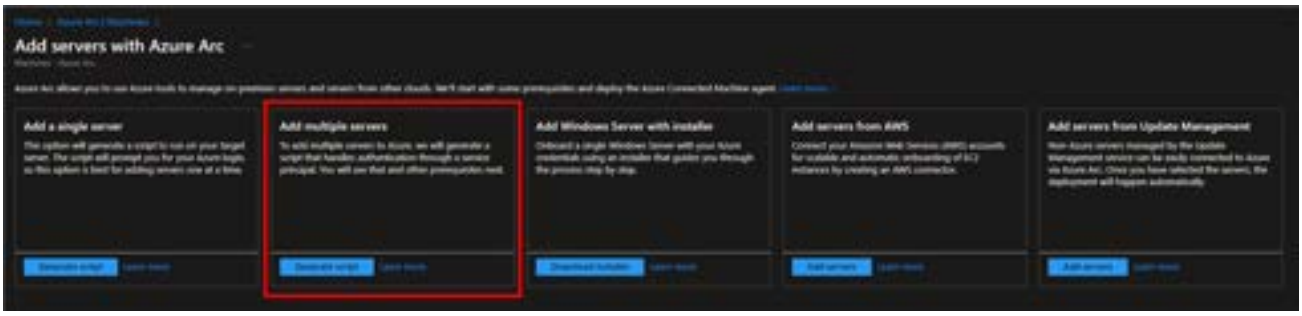
Task 1: Generate a script to automate the download and installation of the Azure Arc connected machine agent for a Windows machine, and to connect the machine to Azure Arc

---

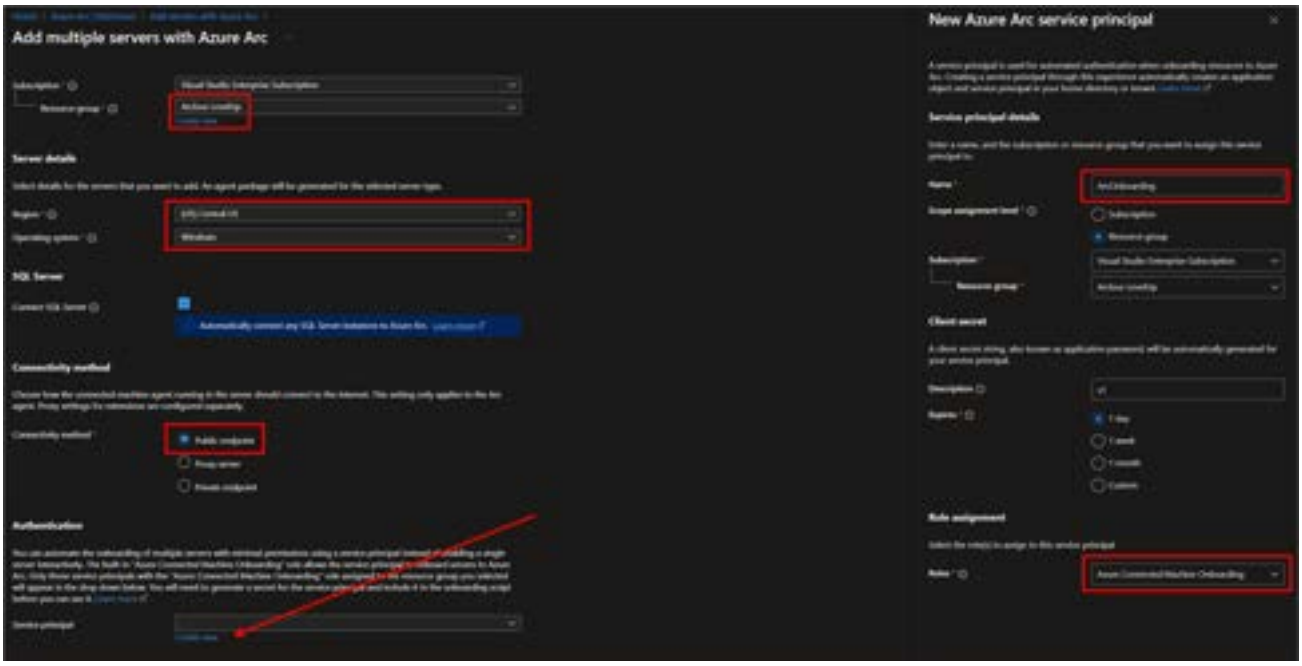
- [ ] 1. From the Azure portal go to the "Machines - Azure Arc" page and select "Add/Create" at the upper left, then select "Add a machine".



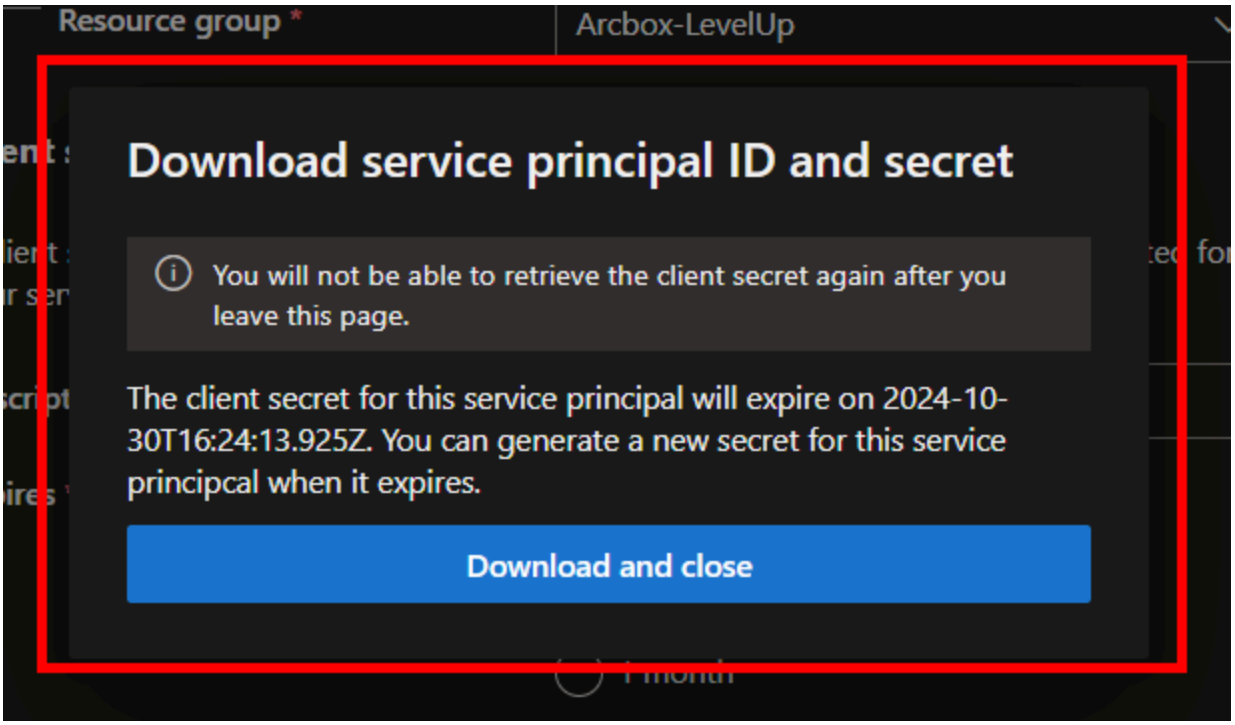
- [ ] 2. In the next screen, go to "Add multiple severs" and click on "Generate script".



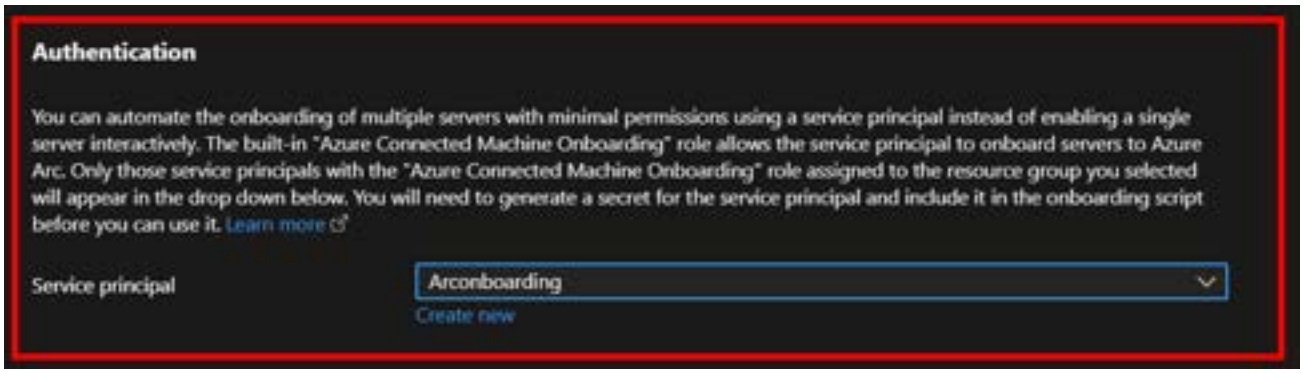
- [ ] 3. Fill in the Resource Group, Region, Operating System (Windows), keep Connectivity as "Public endpoint". Keep the *Connect SQL Server* option ticked. In the Authentication section click on the "Create new" link under the "Service Principal" dropdown and then enter a name (ex:ArcOnboarding) and choose "Azure Connected Machine Onboarding" as the role and click "Create".



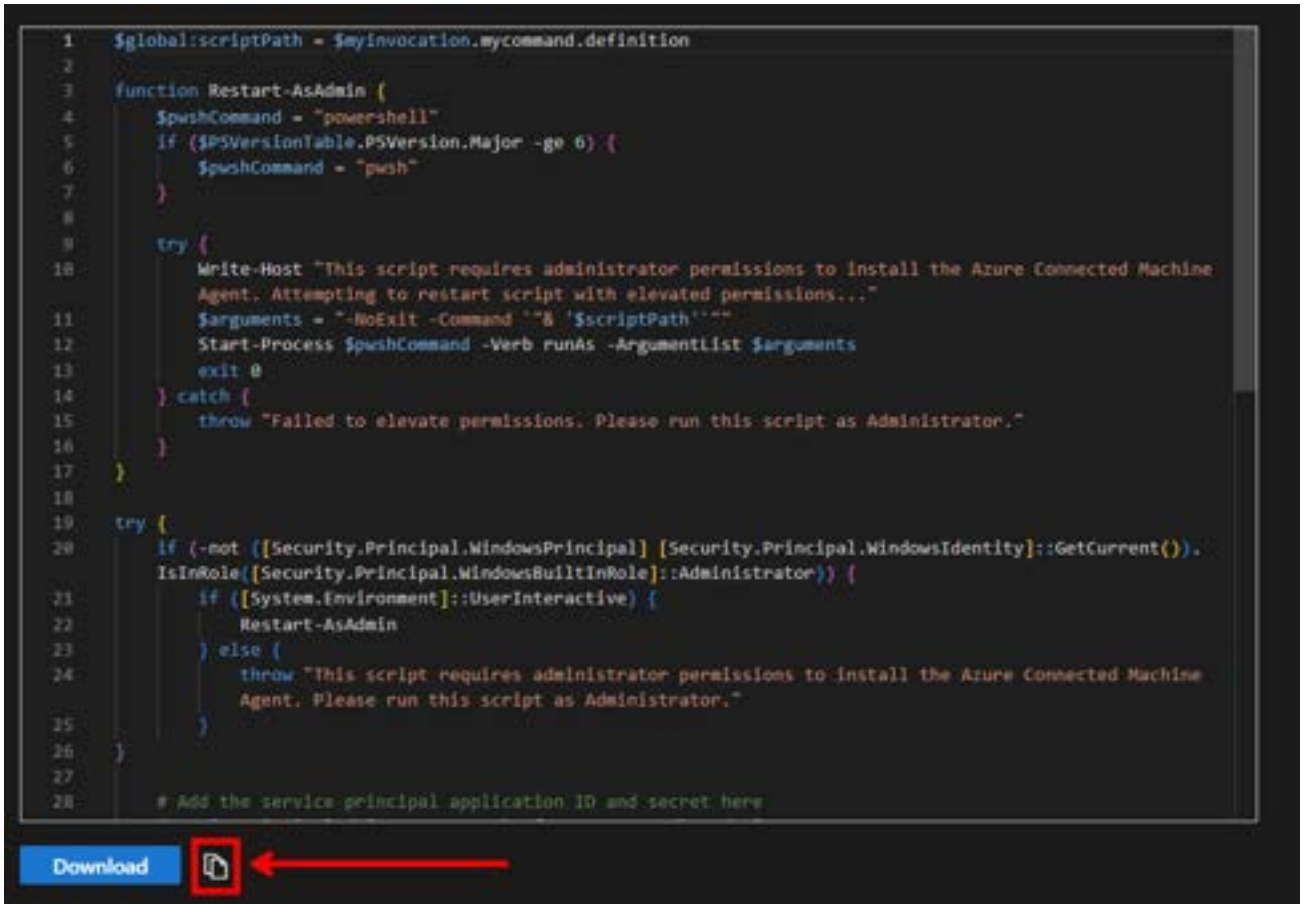
- [ ] 4. Click on "Download and close" to save your new service principal ID and secret as a text file (you will use this in a future step).



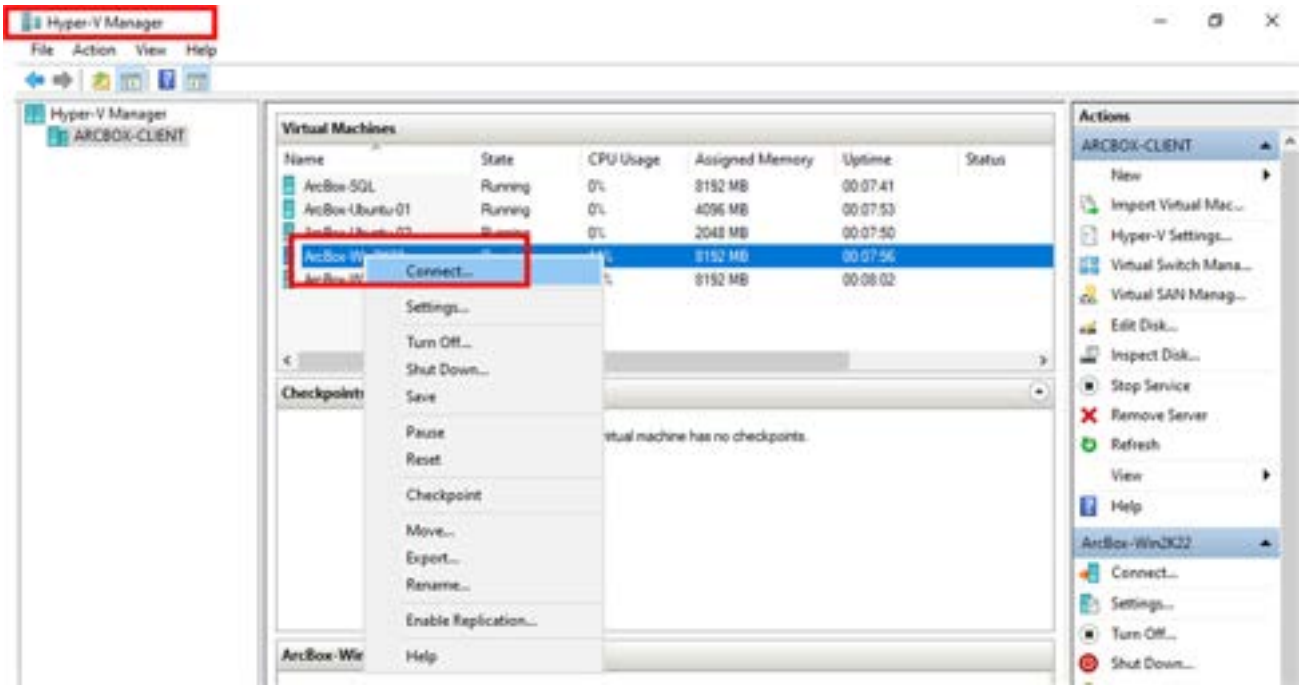
- [ ] 5. Select your newly created "ArcOnboarding" service principal from the dropdown menu.



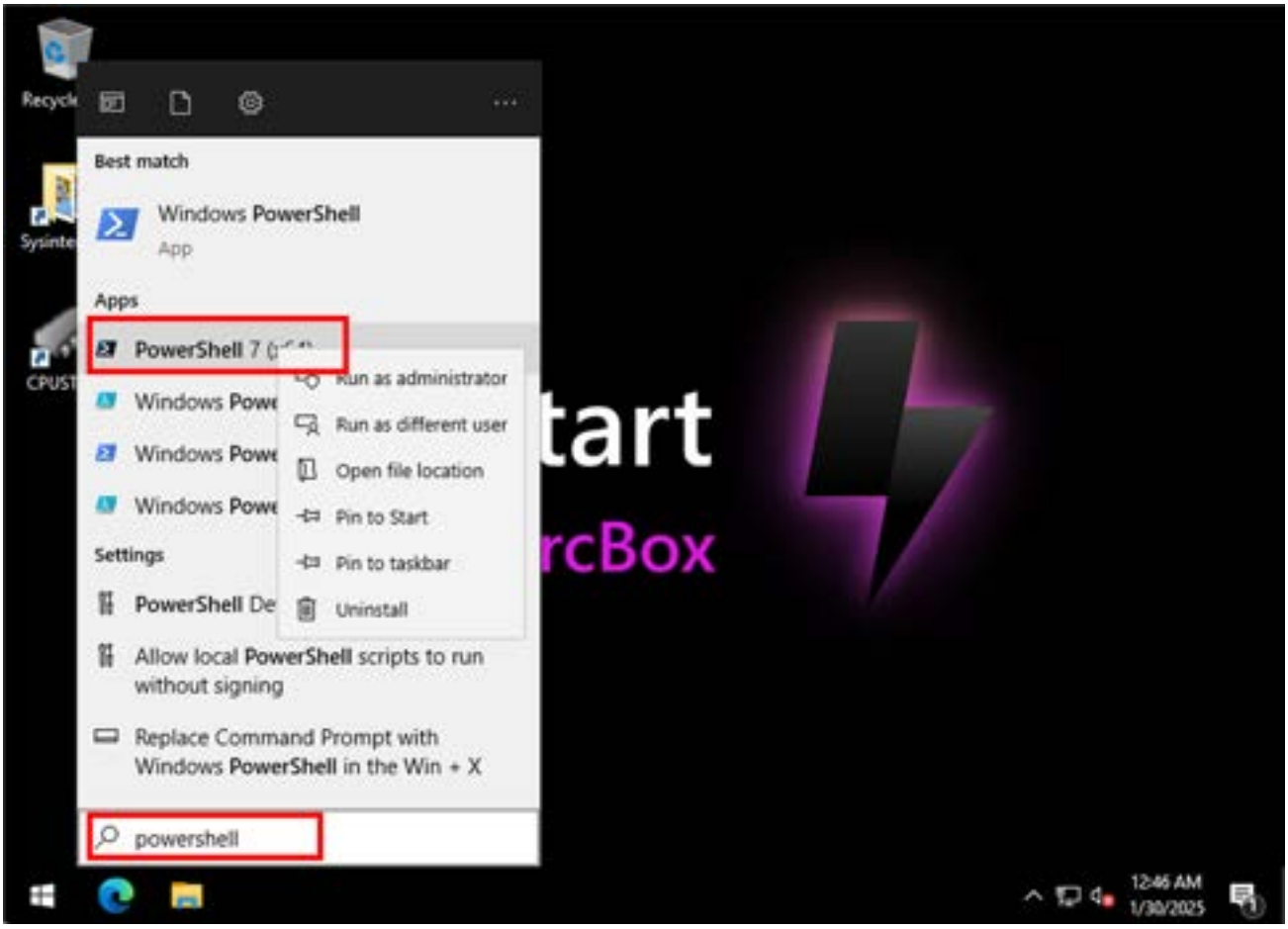
- [ ] 6. Then click on "Download and run script" and scroll down till you see the "Download" button, and next to it click on the icon that looks like two pieces of paper stacked on each other to copy your onboarding script.

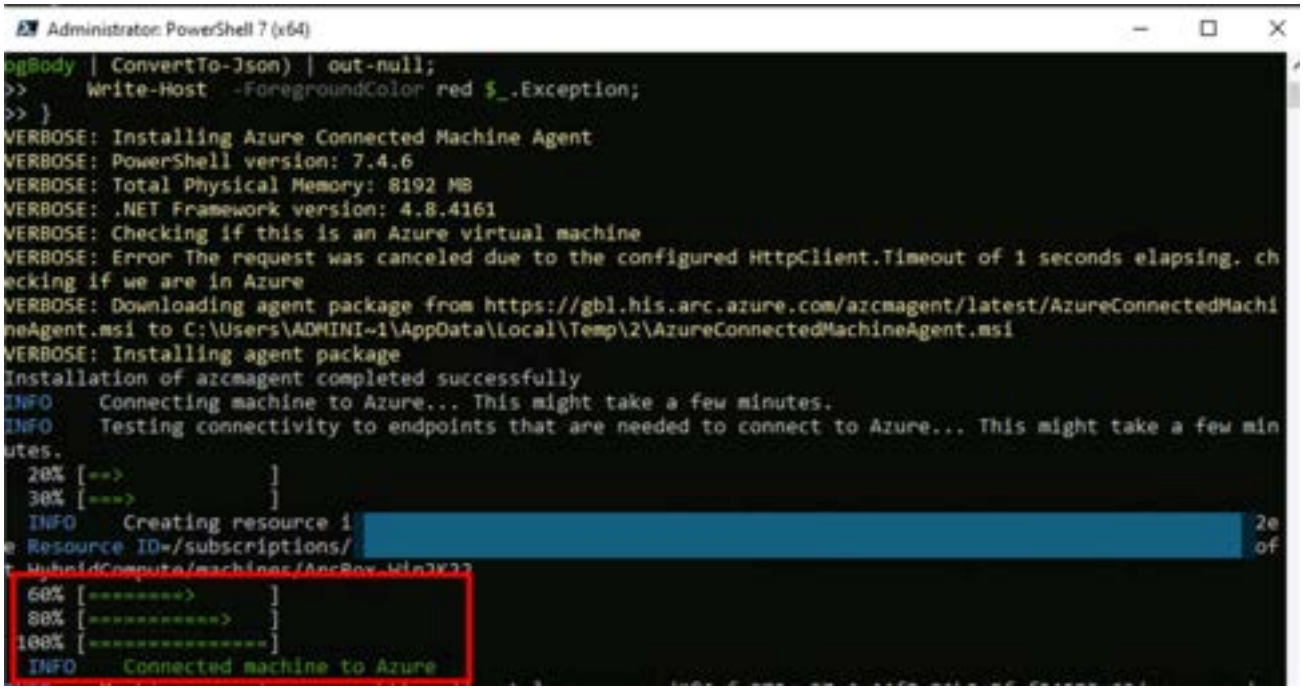


- [ ] 7. Go to the ArcBox-Client machine via RDP and from Hyper-V manager right-click on the *ArcBox-Win2K22* VM and click "Connect" (Administrator default password is JS123!!). Next you will need to open an editor (Visual Studio Code or Notepad for example) and paste the content of the onboarding script.

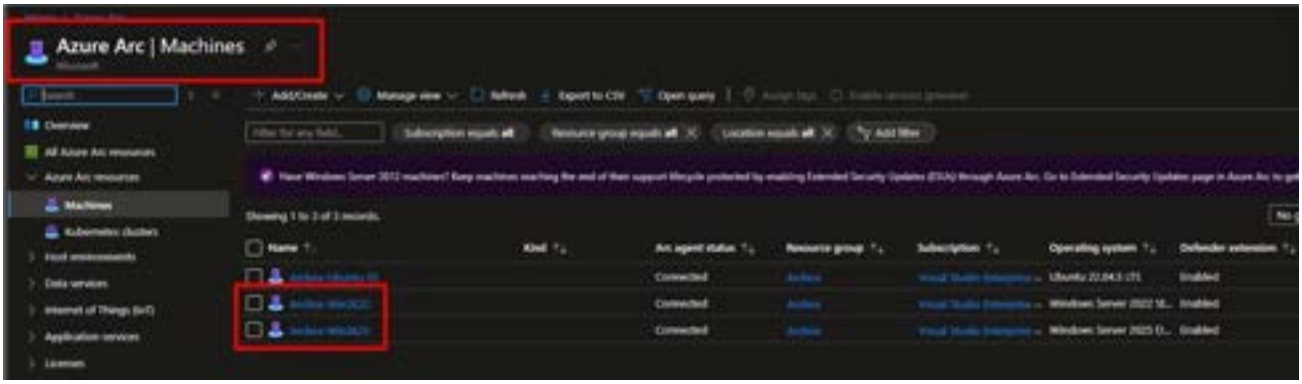


☐ 8. Fill in the Service Principal Id, if it is not already populated, and the secret from the text file that was downloaded above into the script where you see "ENTER SECRET HERE". Start Powershell 7 and paste the script (after you have added the secret). This will execute the script.





9. On successful completion a message is displayed to confirm the machine is connected to Azure Arc. We can also verify that our Windows machines are all now connected in the Azure portal (Machines - Azure Arc).

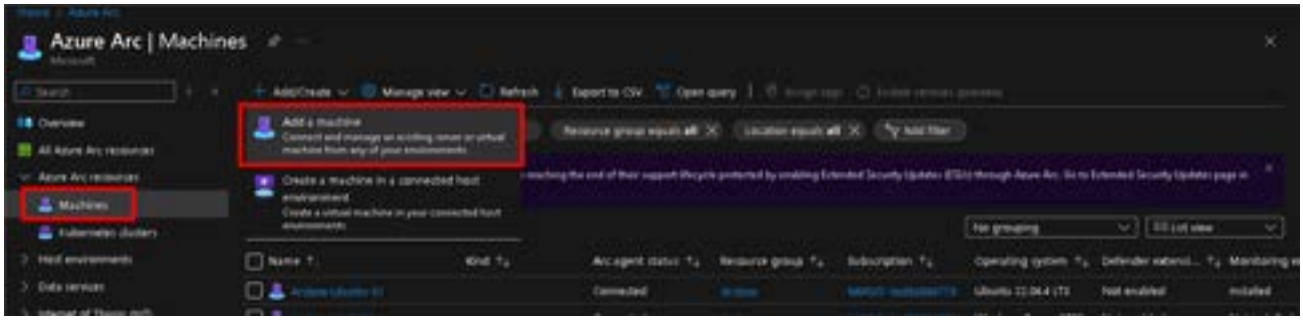


10. **Optional**: If you intend to attempt the Arc-enabled SQL server labs within this workshop, then you might want to run the same script inside the *ArcBox-SQL* VM to onboard this server ready for the coming labs, provided that you had selected the SQL onboarding option ticked when you created the script in the portal.

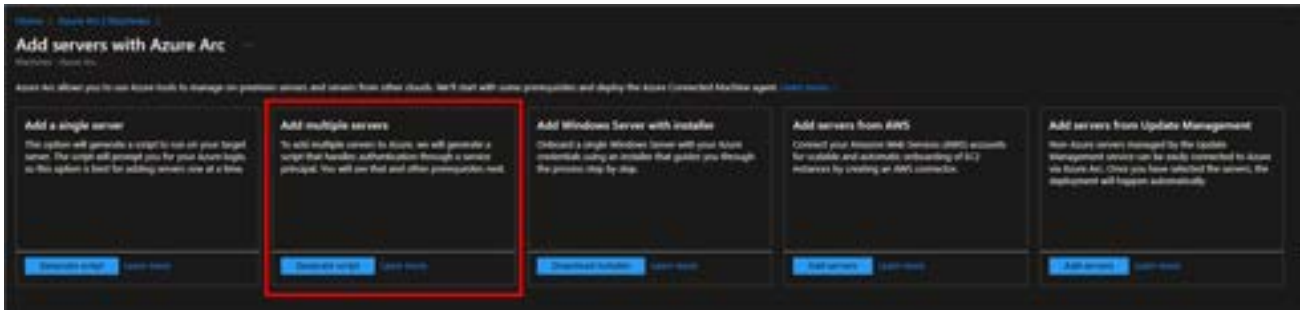
Task 1 has been completed

Task 2: Connect a Linux machine to Azure Arc using the direct internet access (without a proxy)

1. From the Azure portal go to the "Machines - Azure Arc" page and select "Add/Create" at the upper left, then select "Add a machine".



2. In the next screen, go to "Add multiple severs" and click on "Generate script".

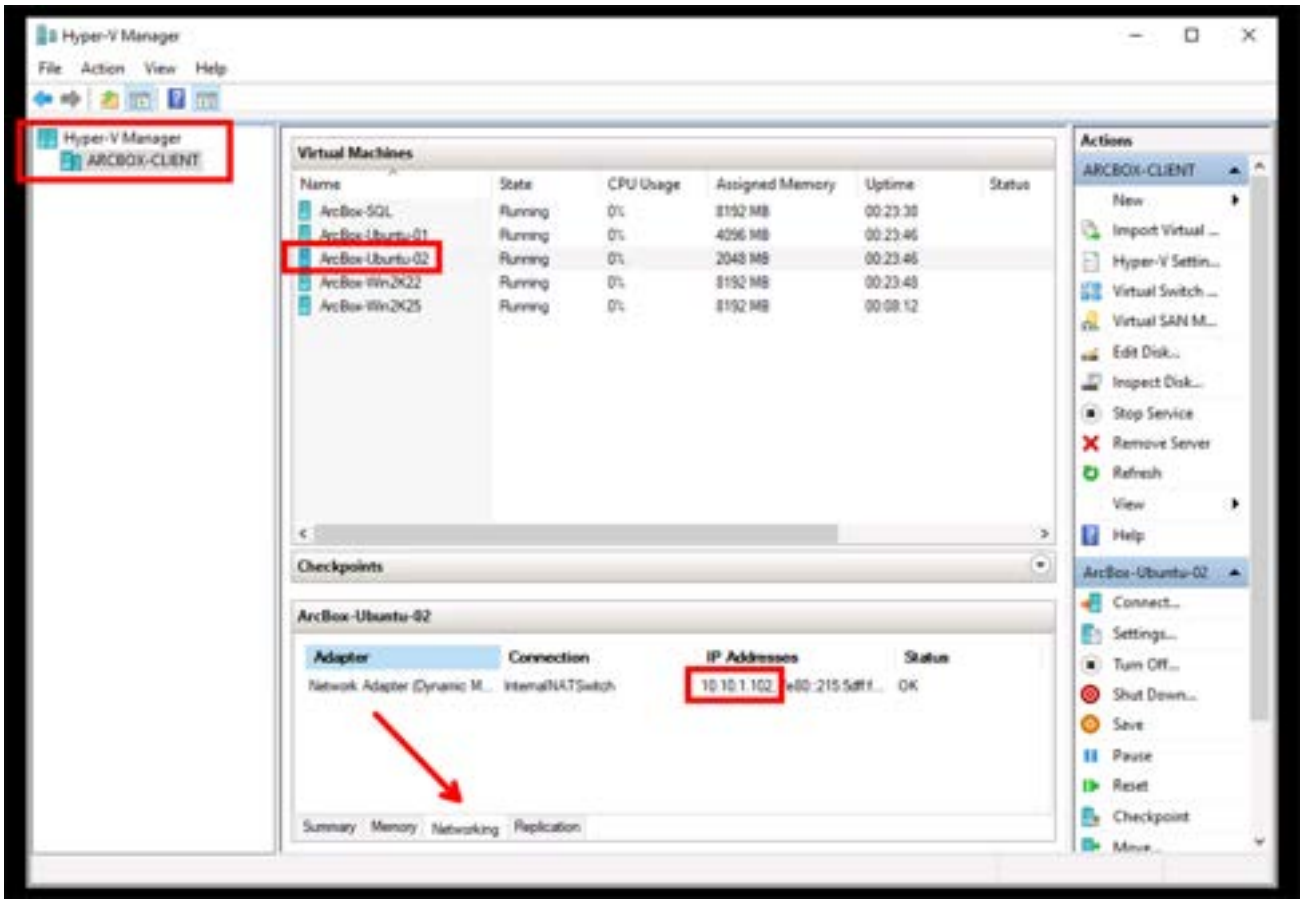


3. Fill in the required details but this time choose Linux for the Operating System box and choose the service principal in the "Authentication" dropdown that you created in the previous exercise. Then download the script to your local machine (or you can copy the content into the clipboard).

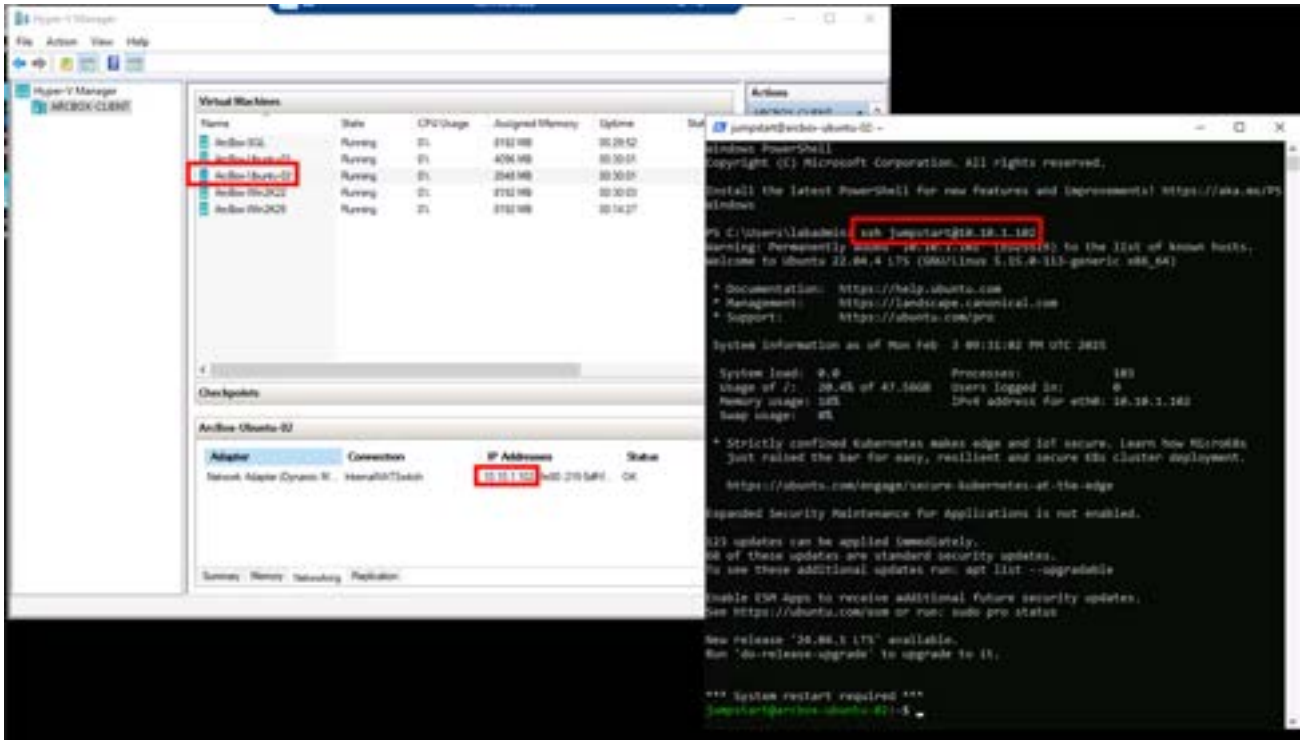
4. Make sure that the Service Principal Id is filled in, then add the client secret to the script using your editor. Also add the following 3 lines to the script **just below the last *export* statement** (to allow onboarding of Azure linux machines):

```shell
sudo ufw --force enable
sudo ufw deny out from any to 169.254.169.254
sudo ufw default allow incoming
```

5. Connect the ArcBox-Client machine, and from the "Networking" tab on Hyper-v Manager find the IP address of the Linux machine.



☐ 6. SSH into the ArcBox-Ubuntu-02 machine using "PowerShell". If you are prompted for a password then use *JS123!!*:



```PowerShell
```

```
ssh jumpstart@<Enter IP Address of of the Linux machine>
```

- [ ] 7. Create an empty onboarding script file using the nano editor, and paste the script content from your local machine. Once you are in the editor, navigate using the arrow keys and not the mouse.

```shell
nano onboardingscript.sh
```

- [ ] 8. Save the file (Ctrl-O then Enter) and exit (Ctrl-X). Now you can run the script:

```shell
sudo bash ./onboardingscript.sh
```

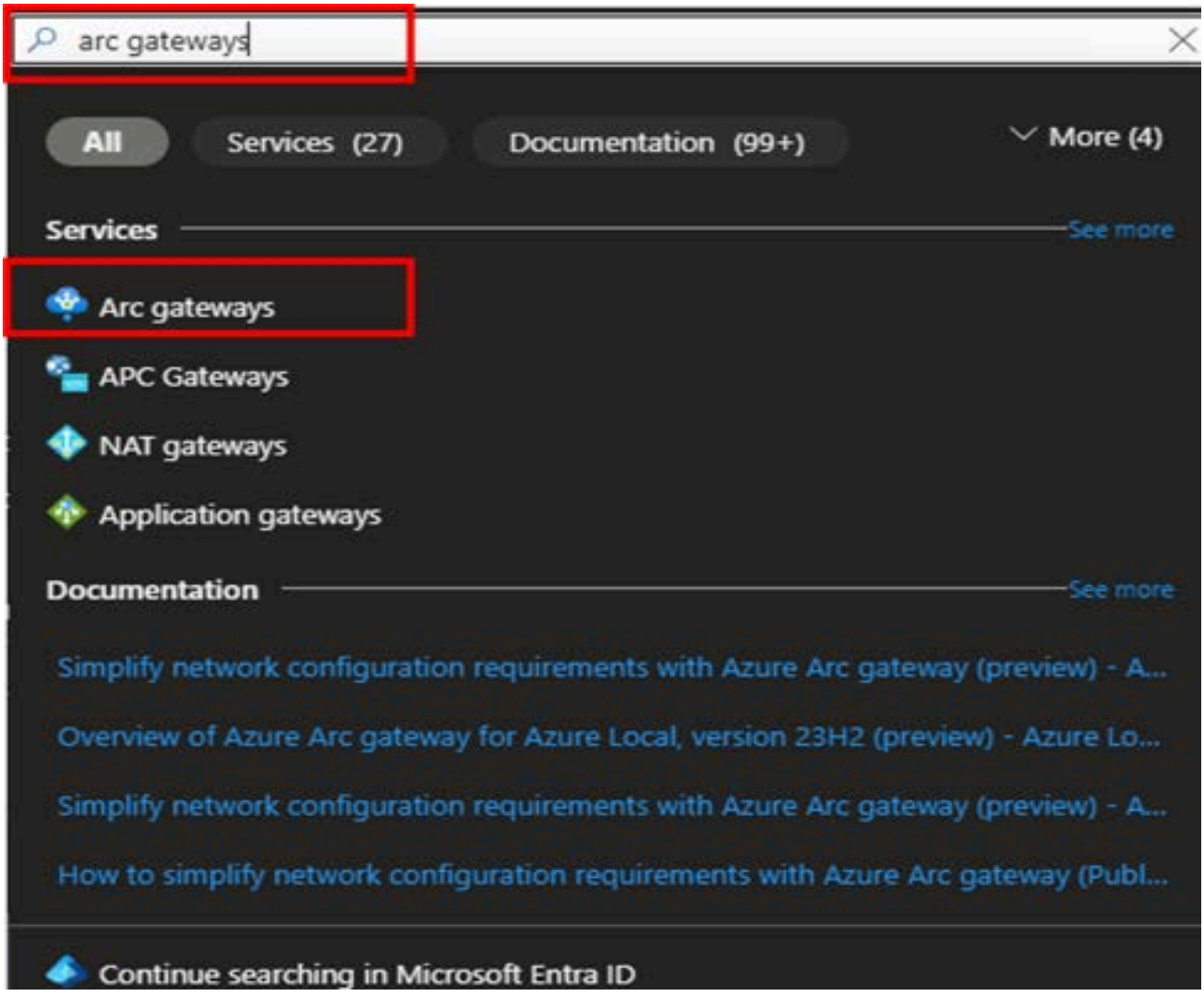
- [ ] 9. Wait for the script to finish successfully. A message should confirm that the machine is now Arc-connected. You can also verify that our Linux machine is connected in the Azure portal (Machines - Azure Arc).

Task 2 has been completed

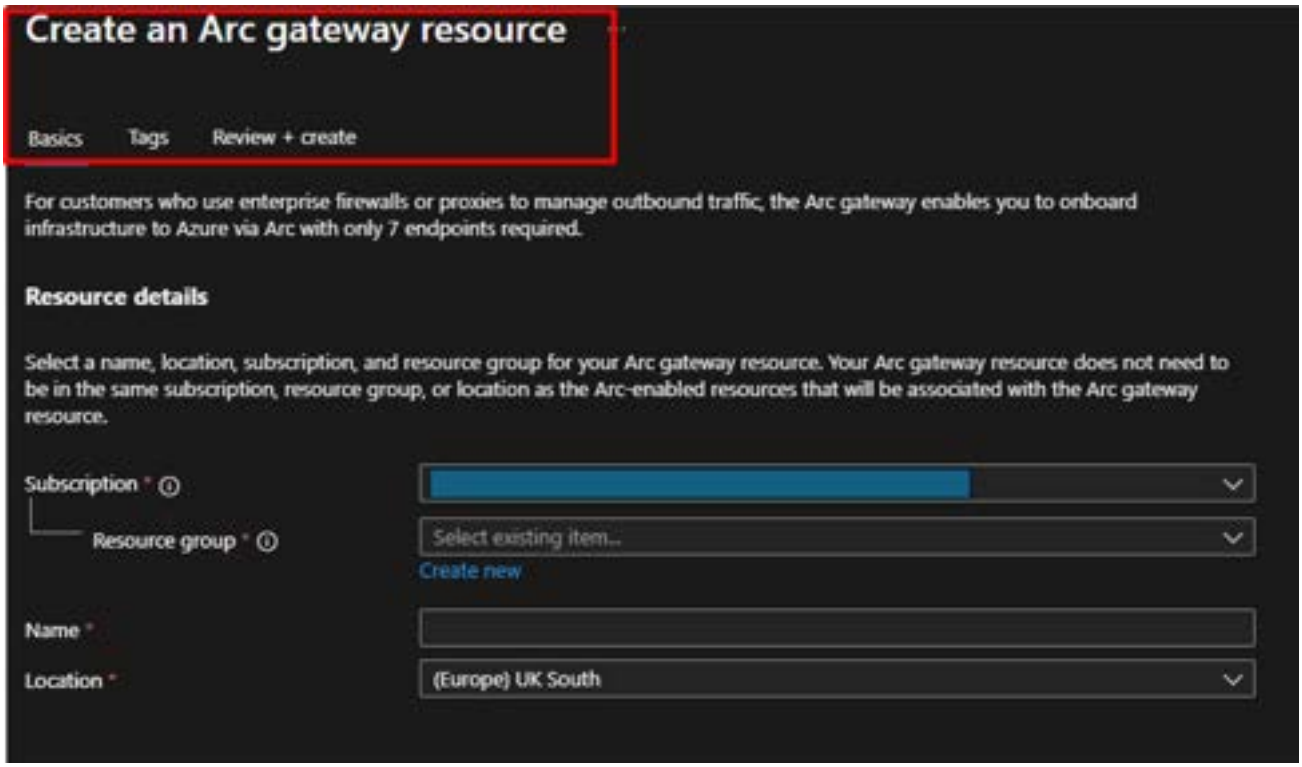
Task 3: Connect a linux machine to Azure Arc using a proxy and Arc Gateway

- On the Ubuntu02 machine, run the following command - sudo apt purge azcmagent
- Delete Ubuntu02 machine from the Azure Arc - Machines by navigating to Azure Arc | Machines/arcbox-ubuntu-02, and select the delte option.

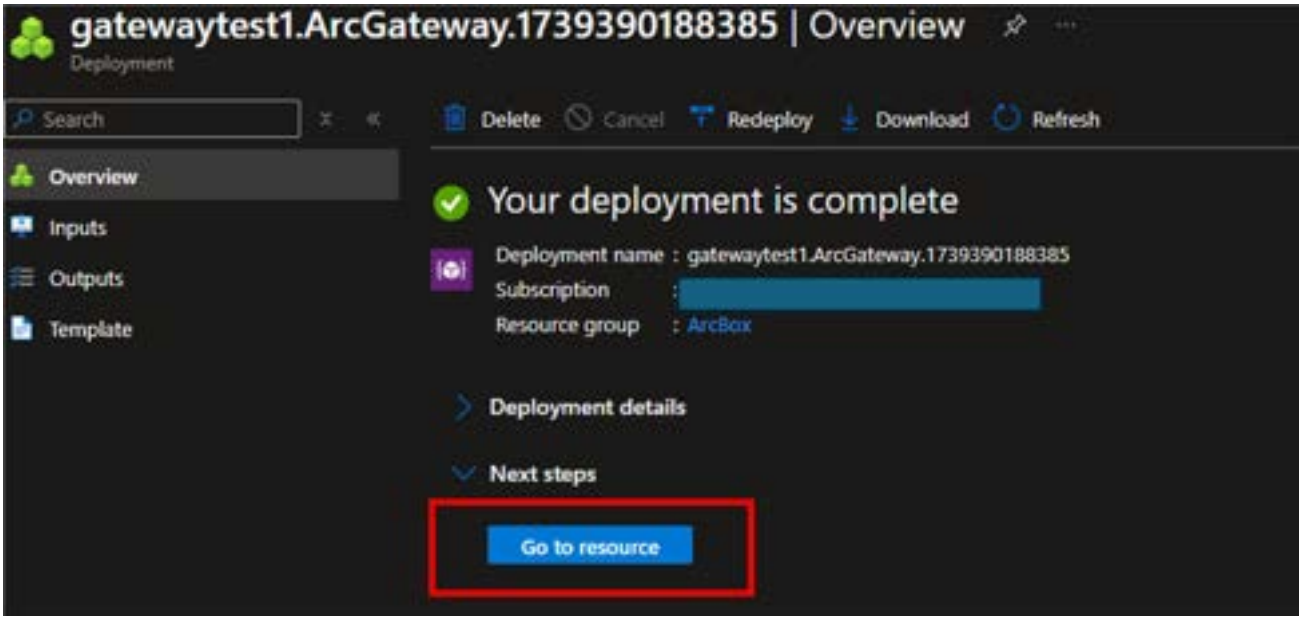
1. From the Azure portal search for *Arc gateways* and select from the search results, then click on the *Create* button.



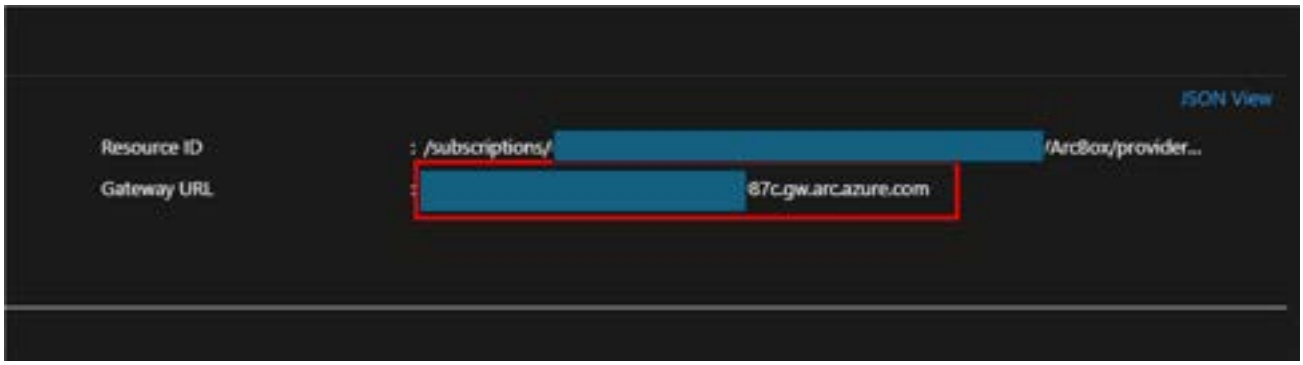
2. Fill in the details of the resource group name, choose a name for your Arc Gateway and choose the region of your lab resources. Then move to the *Review + create* tab and create the gateway.



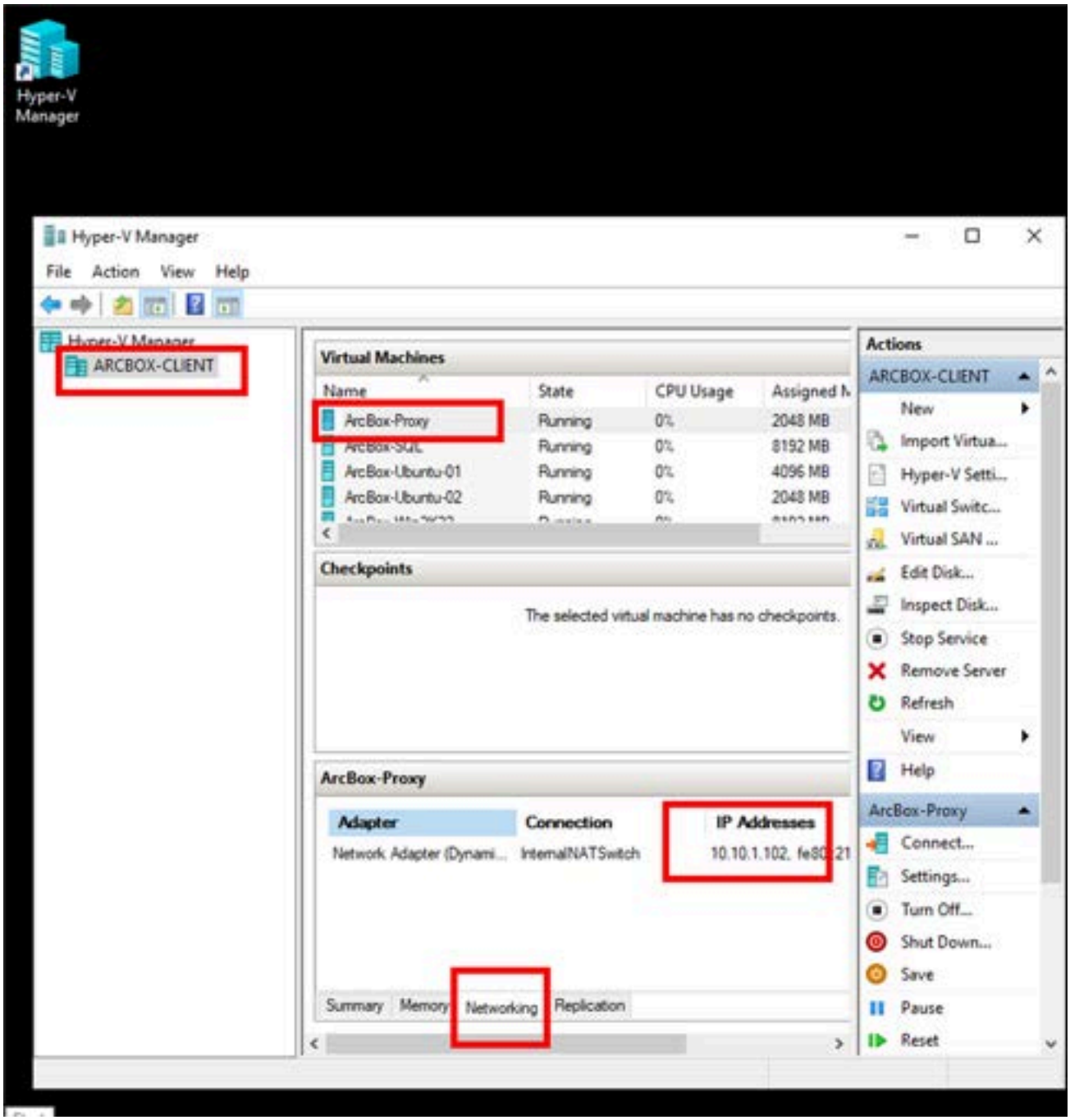
☐ 3. The deployment will take few minutes and when it is done choose *Go to resource*.



☐ 4. Find the Gateway URL and make a note of it to be used later.



5. From the *ArcBox-Client* machine open *Hype-V Manager*, click the *ArcBox-Client* server, click the *ArcBox-Proxy* under Virtual Machines. From the *Networking* tab find the IPV4 address of the proxy. Make a note of this IP address.

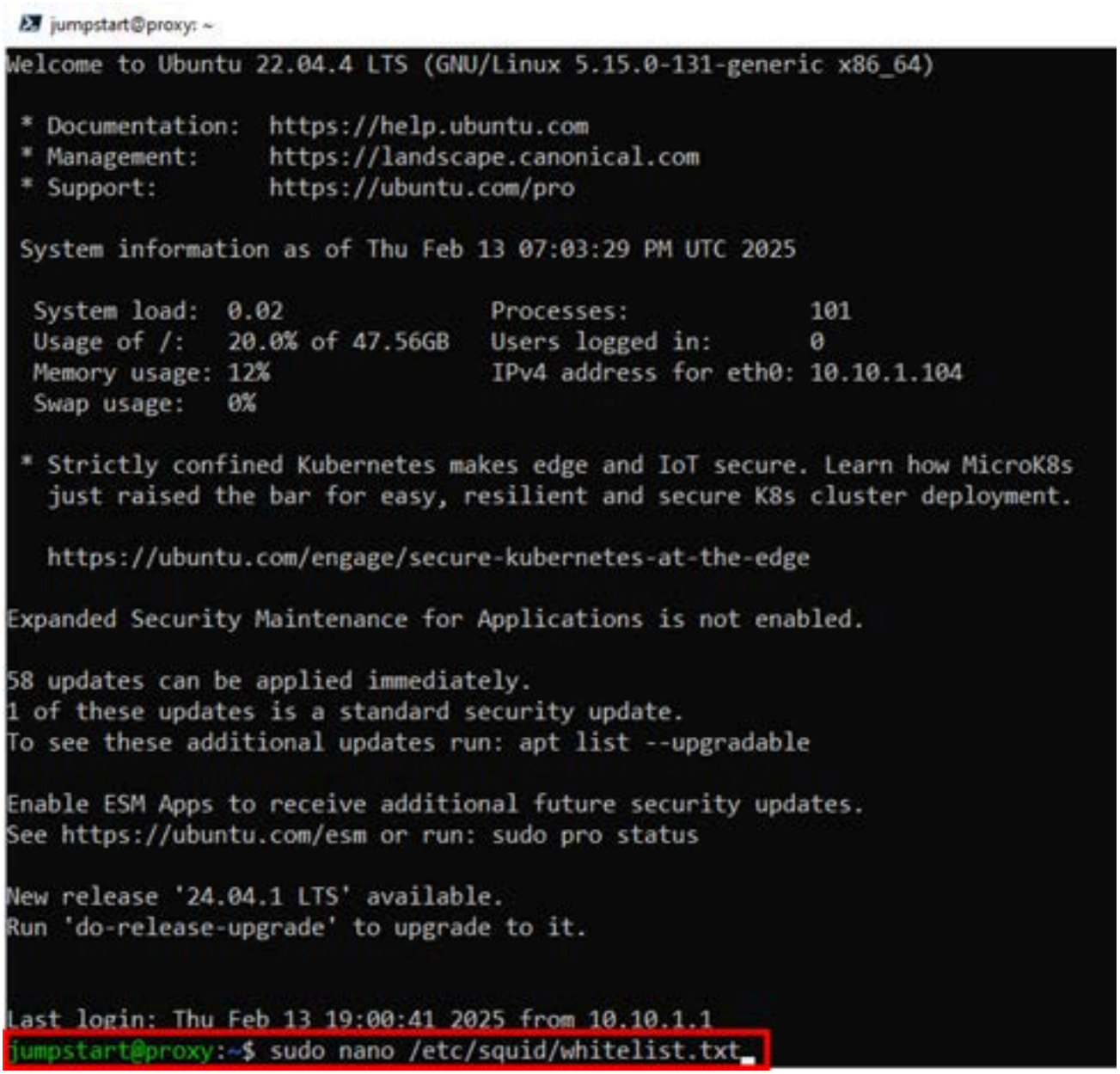


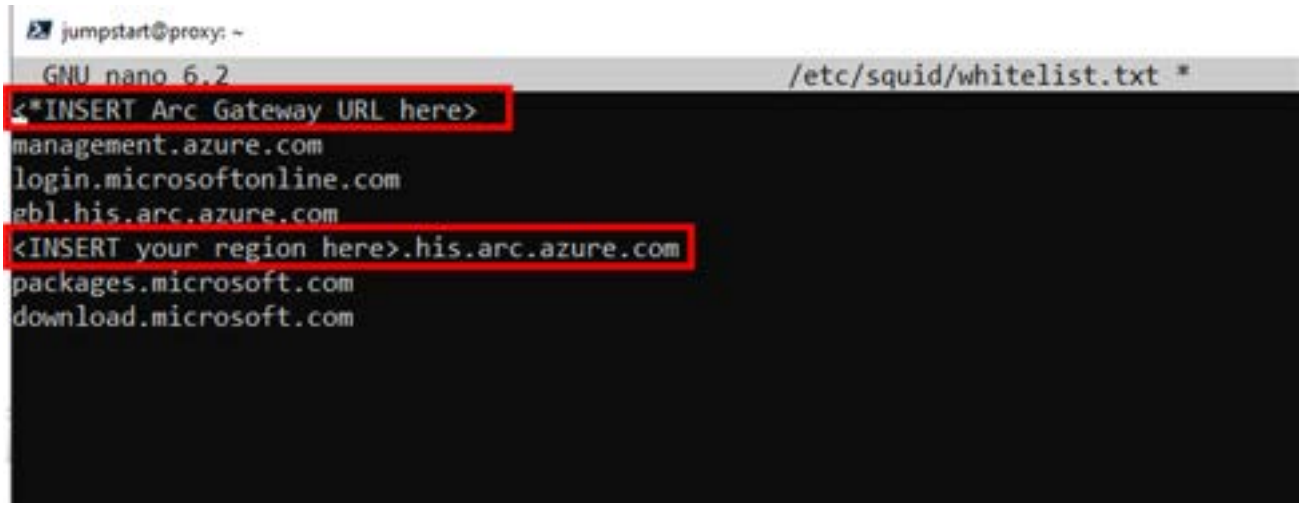
6. From the *ArcBox-Client* machine use Powershell 7 to ssh into the proxy server *ArcBox-Proxy* using the IP address you found in the previous steps.

```PowerShell
ssh jumpstart@<Enter IP Address of of the proxy machine>
```

7. Once you have logged into the proxy machine edit the *whitelist.txt* file which contains the list of links allowed by the proxy. These should be similar to the links stated in Arc Gateway documentation. If any of the links is missing then add it to the whitelist. Once you are in the editor, navigate using the arrow keys and not the mouse. Make sure that you fill in the Arc gateway URL that you have from the earlier steps, and also the region of your deployment.

```PowerShell
sudo nano /etc/squid/whitelist.txt
```





☐ 8. Save the file (Ctrl-O then Enter) and exit (Ctrl-X).

9. In order for the new whitelist to be active, restart the Squid proxy service.

```
PowerShell
sudo systemctl restart squid
```

10. **Optional**: If you want to have a look at how the Squid proxy server configuration uses the whitelist then open the file /etc/squid/squid.conf and examine the settings, including the port used *3128* which is the default Squid proxy port.

11. From the *ArcBox-Client* machine open *Hype-V Manager*, click the *ArcBox-Client* server, click the *ArcBox-Ubuntu-02* under Virtual Machines. From the *Networking* tab find the IPV4 address of the machine. Make a note of this IP address. Then ssh into it from Powershell 7.

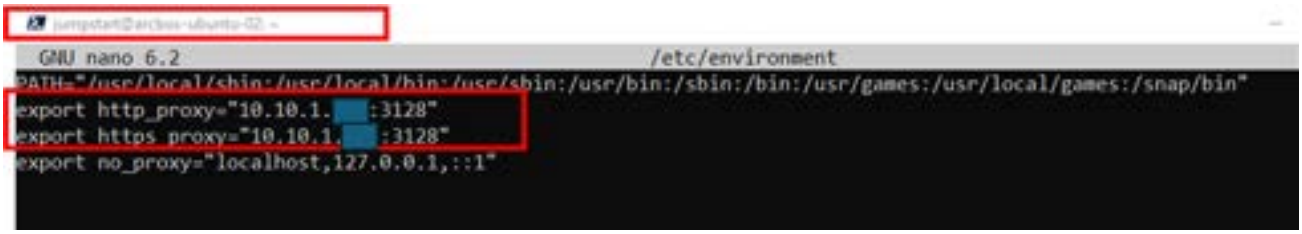
```
PowerShell
ssh jumpstart@<Enter IP Address of of the ArcBox-Ubuntu-02 machine>
```

12. Configure the proxy settings on the ArcBox-Ubuntu-02 machine.

```
PowerShell
sudo nano /etc/environment
```

13. Add the following 3 lines to the file making sure that you use the IP address of the **proxy server**! After you have done, Save the file (Ctrl-O then Enter) and exit (Ctrl-X).

```
PowerShell
export http_proxy="<proxy IP Address>:3128"
export https_proxy="<proxy IP Address>:3128"
export no_proxy="localhost,127.0.0.1,::1"
```



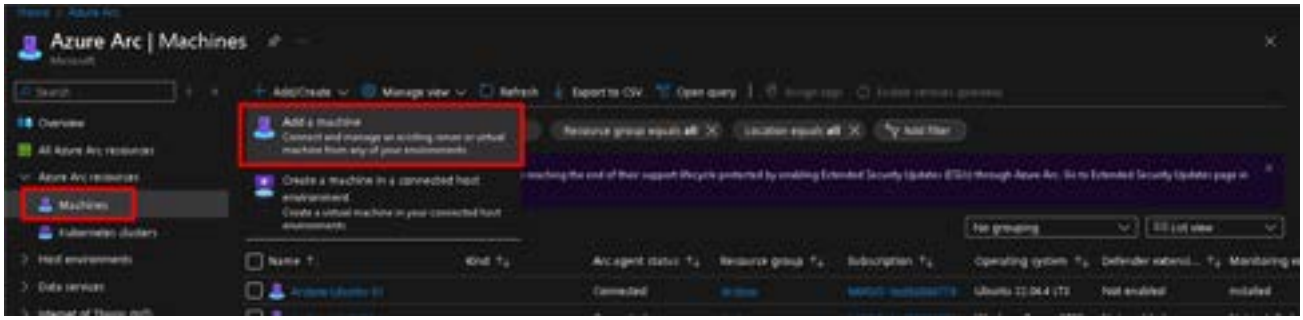
14. Type *EXIT* then re-login again to the *ArcBox-Ubuntu-02* for the changes to take effect.

```
PowerShell
ssh jumpstart@<Enter IP Address of of the ArcBox-Ubuntu-02 machine>
```

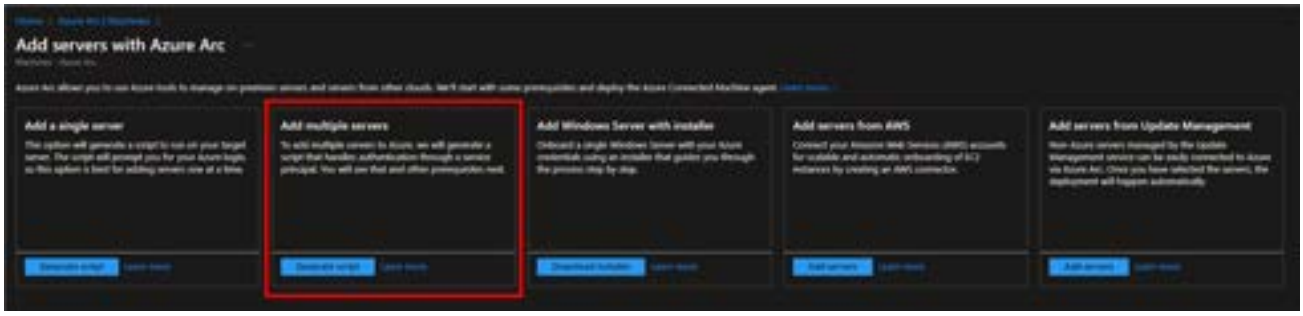
15. Optional: Test that your proxy is blocking internet access

```
jumpstart@arcbox-ubuntu-02: ~
jumpstart@arcbox-ubuntu-02:~$ curl -I http://bing.com
HTTP/1.1 403 Forbidden
Server: squid/5.9
Mime-Version: 1.0
Date: Thu, 13 Feb 2025 13:40:33 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 3485
X-Squid-Error: ERR_ACCESS_DENIED 0
Vary: Accept-Language
Content-Language: en
X-Cache: MISS from proxy
X-Cache-Lookup: NONE from proxy:3128
Via: 1.1 proxy (squid/5.9)
Connection: keep-alive

jumpstart@arcbox-ubuntu-02:~$
```

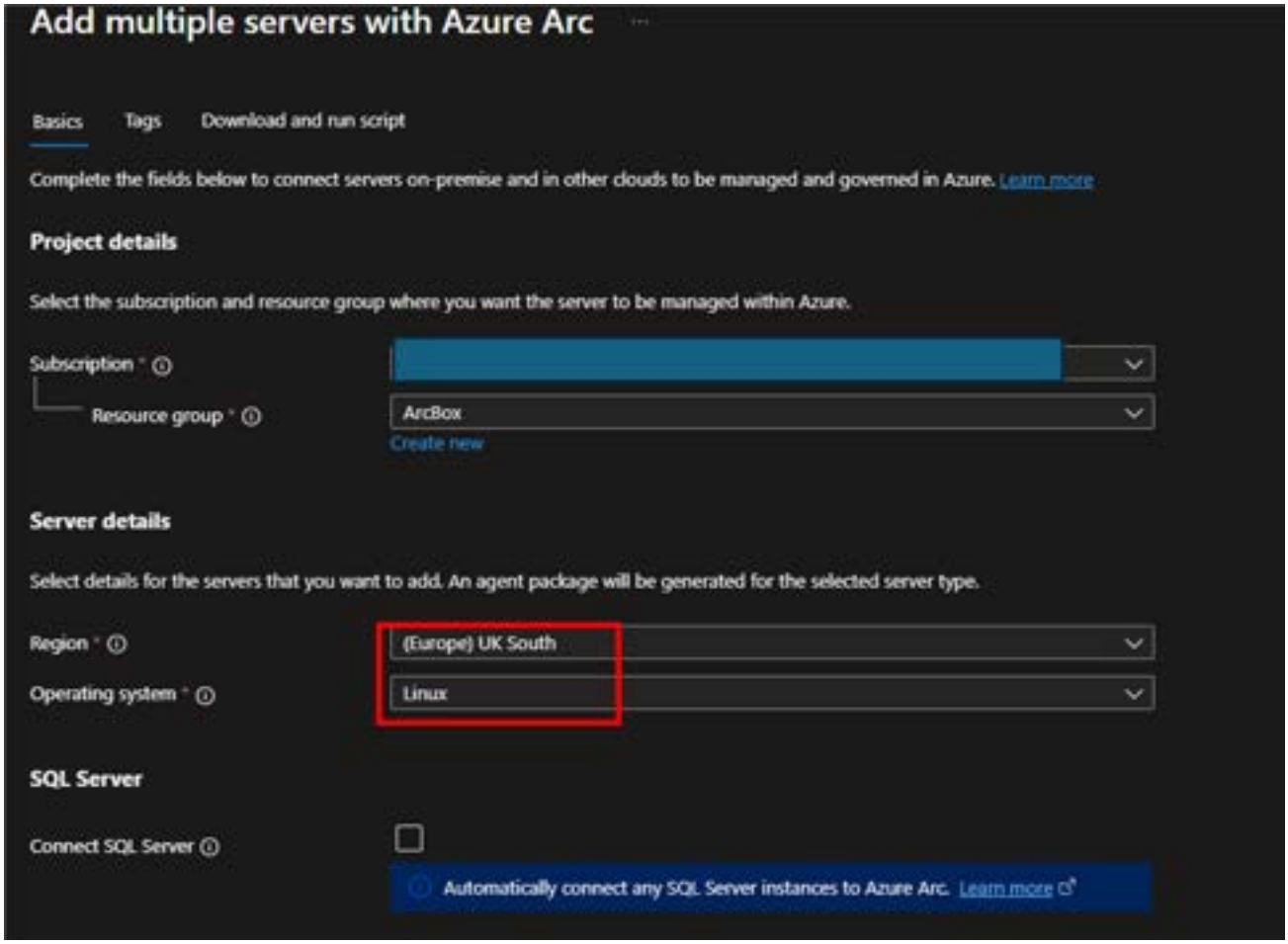
- [ ] 16. From the Azure portal go to the "Machines - Azure Arc" page and select "Add/Create" at the upper left, then select "Add a machine".

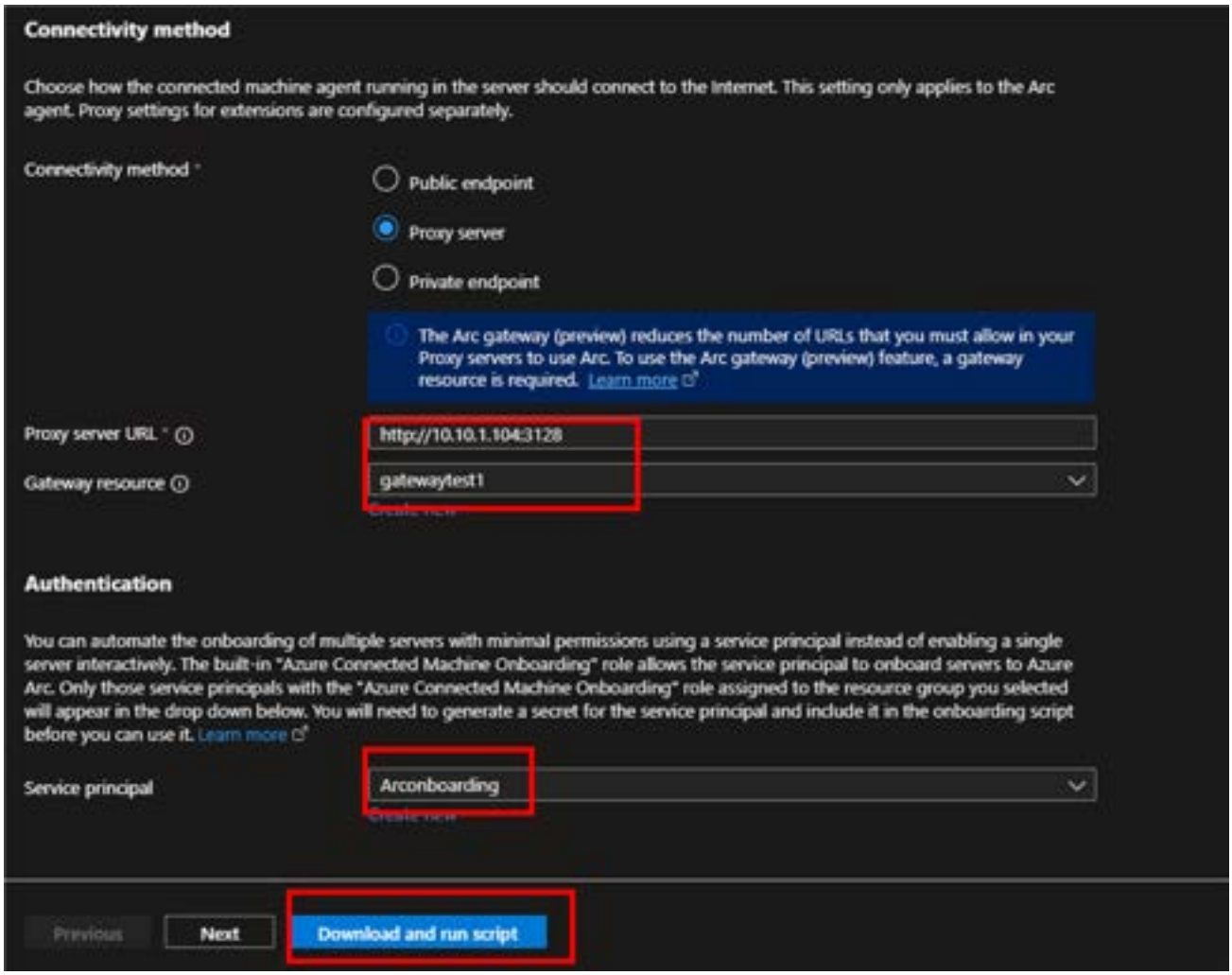


- [ ] 17. In the next screen, go to "Add multiple severs" and click on "Generate script".

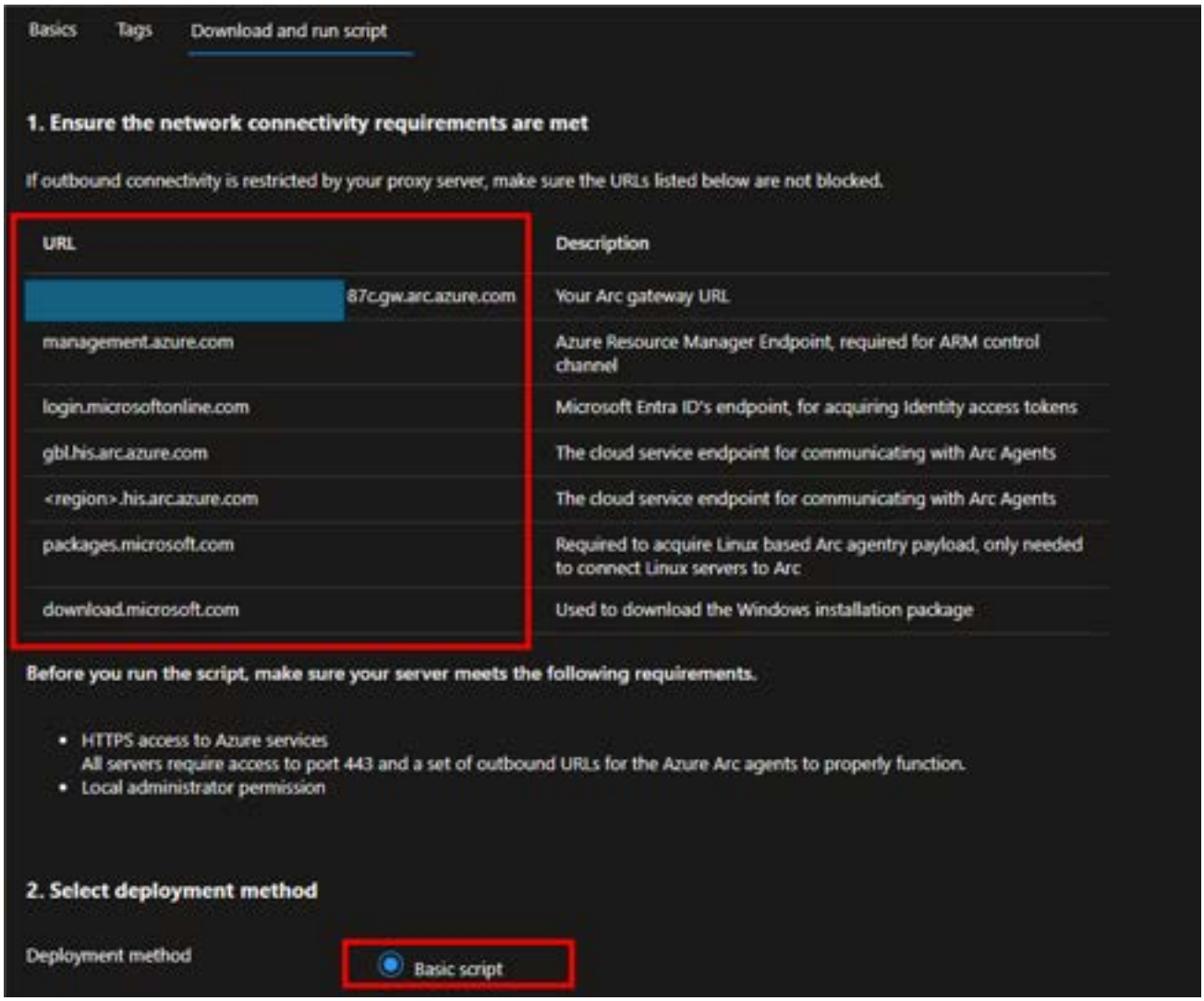


- [ ] 18. Fill in the required details and choose Linux for the Operating System box. Set the connectivity method to *Proxy server*. Enter the URL of the proxy server and name of the Arc gateway as shown in the screenshots but make sure that they match with your own settings (the IP address of the proxy server that you observed earlier). Also Choose the service principal in the "Authentication" dropdown that you created in the previous exercise.

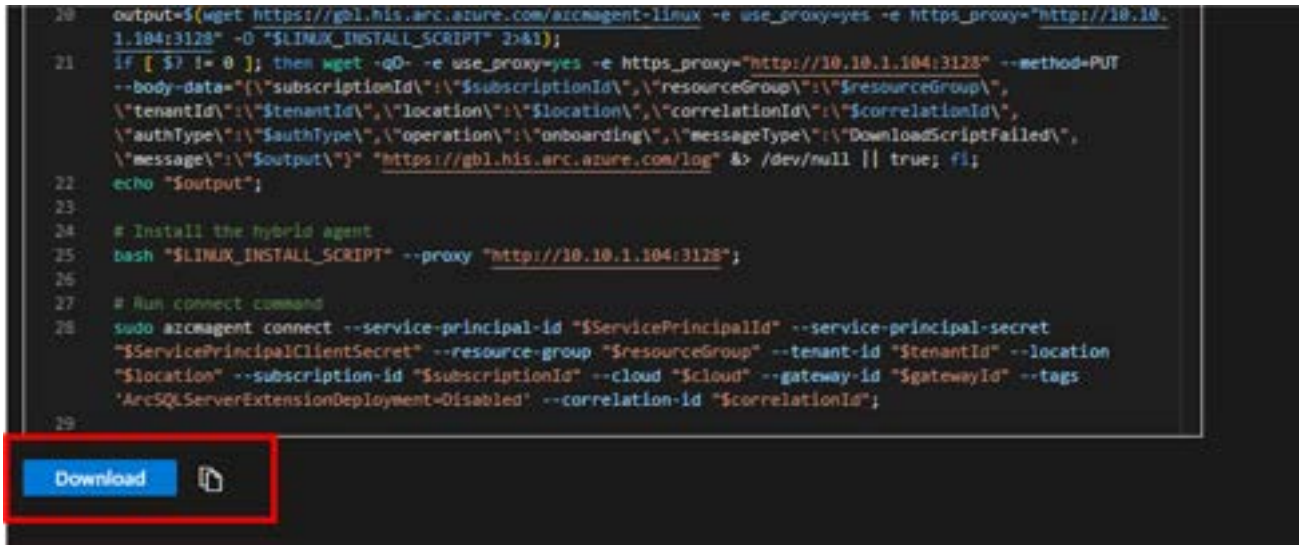




☐ 19. On the *Download and run script* tab notice the reminder to ensure that the required endpoints are not blocked by the proxy server. These were the links that you added to the proxy whitelist earlier. Choose the *Basic script* option.



☐ 20. Copy the script or download it. You should see that your proxy IP address has been configured in the script.



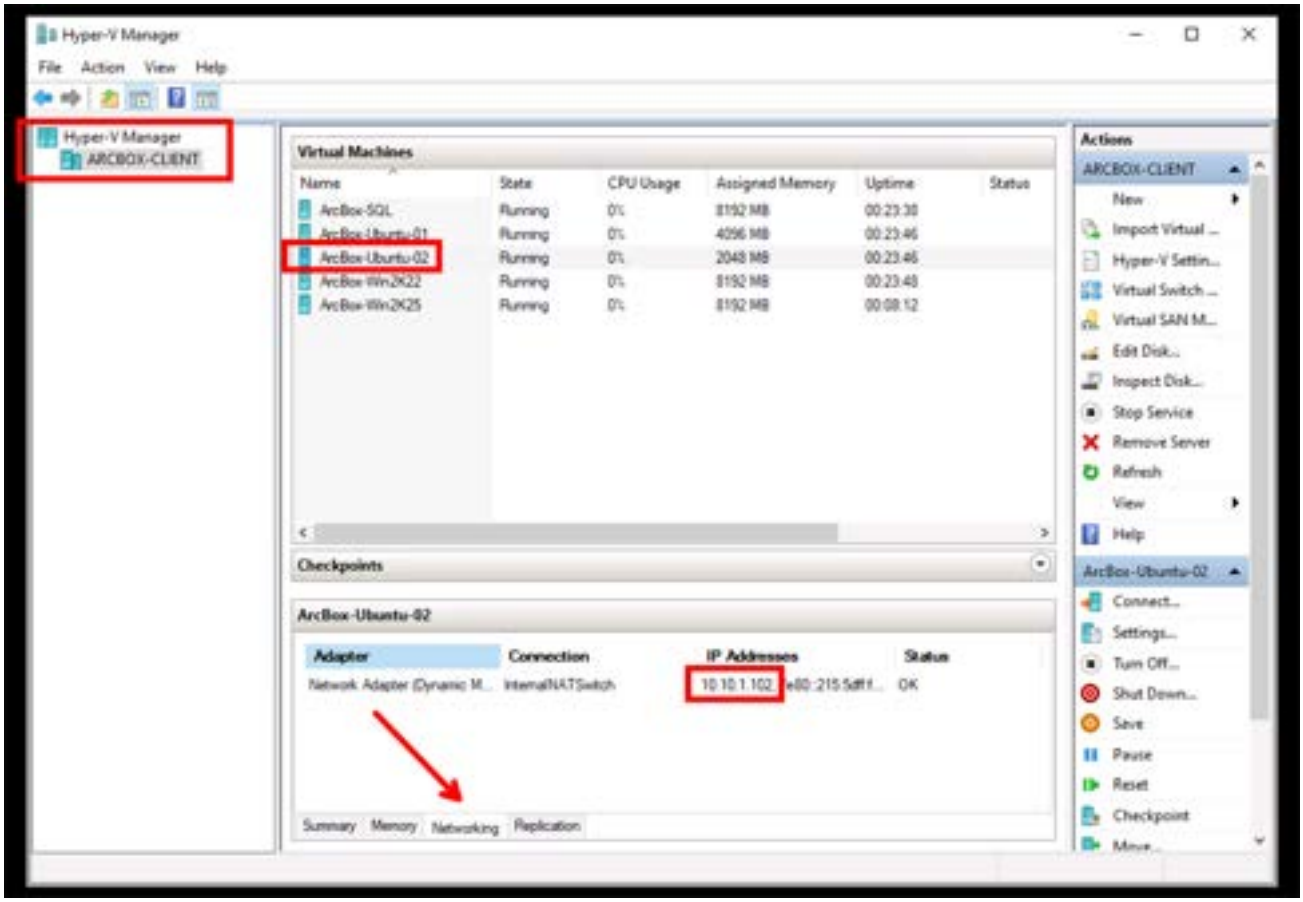
☐ 21. In your local editor, paste the script and make sure that the Service Principal Id and the client secret are filled in. Also add the following 3 lines to the script **just below the last *export***

statement (to allow onboarding of Azure linux machines):

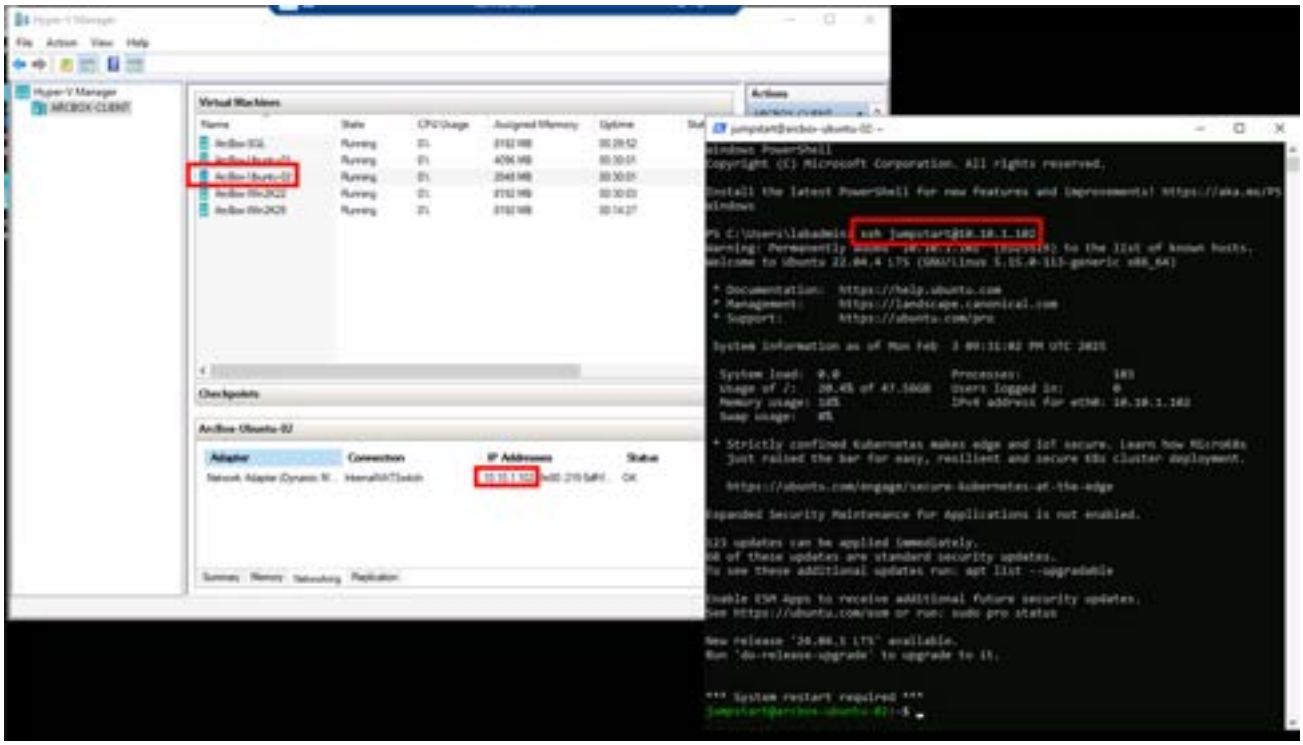
```shell
sudo ufw --force enable
sudo ufw deny out from any to 169.254.169.254
sudo ufw default allow incoming
```

22. Connect the ArcBox-Client machine. If you are still connected to the *ArcBox-Ubuntu-02* machine with ssh session then ignore the following 2 steps.

23. From the "Networking" tab on Hyper-v Manager find the IP address of the Linux machine *ArcBox-Ubuntu-02*.



24. SSH into the *ArcBox-Ubuntu-02* machine using "PowerShell". If you are prompted for a password then use *JS123!!*:



PowerShell

```
ssh jumpstart@<Enter IP Address of of the Linux machine>
```

- [ ] 25. Create an empty onboarding script file using the nano editor, and paste (Right-click) the script content from your local machine. Once you are in the editor, navigate using the arrow keys and not the mouse.

shell

```
nano onboardingscript.sh
```

- [ ] 26. Save the file (Ctrl-O then Enter) and exit (Ctrl-X). Now you can run the script:

shell

```
sudo bash ./onboardingscript.sh
```

- [ ] 27. Wait for the script to finish successfully. A message should confirm that the machine is now Arc-connected. You can also verify that our Linux machine is connected in the Azure portal (Machines - Azure Arc).

- [ ] 28. You can confirm that the *ArcBox-Ubuntu-02* is associated with your Arc gateway by navigating to the Azure portal page of the Arc gateway and examining the *Associated resources*.

Task 2 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB03: Securing Azure Arc-enabled servers with Microsoft Defender for Cloud

---

Student Lab Manual

Table of Contents

Exercise 1 - Detect threats on your servers using alerts

Objective

This exercise will walk you through simulating malicious activity on your machines and examine how Defender for Servers can alert you on those threats.

Estimated Time to Complete This Lab

45 minutes

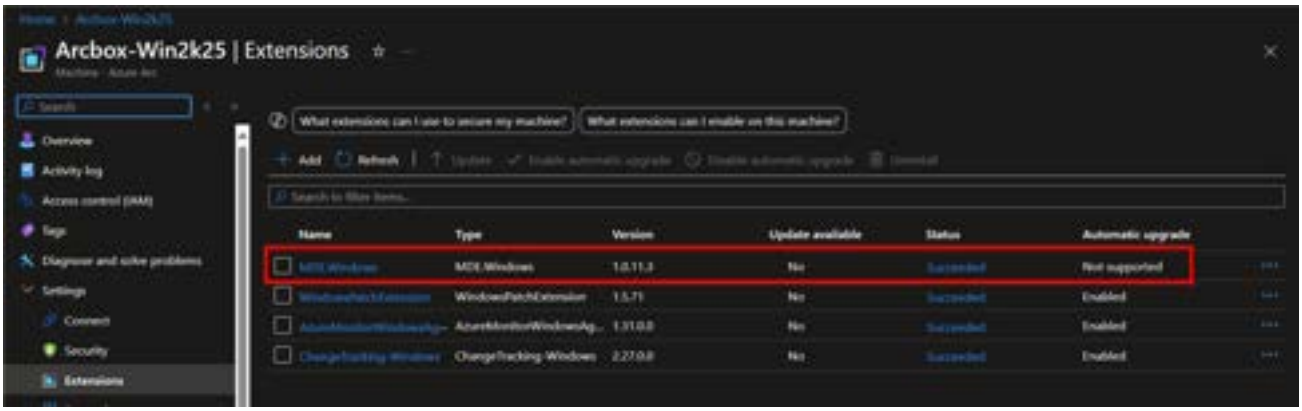
Explanation

In this exercise, you will learn how to leverage Defender for Servers to detect threats and malicious activities.

Task 1: Simulate malicious activities

1. To simulate a malicious activity on the *ArcBox-Win2K25* servers, rdp into the *ArcBox-Client* VM.

Before simulating the alert, make sure that the 'MDE.Windows' is installed on the Arc-enabled server. You can do this by checking the Extensions installed on the Arc enabled server *ArcBox-Win2k25* from the Azure portal.



If the 'MDE.Windows' extension is missing then either the *Microsoft Defender for Cloud* configuration has not been set correctly as explained in Lab01 **OR** the configuration of the *Microsoft Defender for Cloud* has not taken effect yet. Go back to Lab01 and check that you have set the Defender for Cloud correctly in the step *Enable the Defender for Servers plan* and have saved the configuration.

2. **If the 'MDE.Windows' extension is missing and not in the process of being created as seen in the Azure Portal** then you can manually install it on the *ArcBox-Win2k25* by running the following Powershell script either in the *ArcBox-Client* machine or in Powershell from the Azure Portal cloud shell, **making sure that the Resource Group name is correct**. Otherwise if the extension is installed then move to the next step.

If you decide to run the manual installation of the *MDE.Windows* extension from the *ArcBox-Client* machine, then you will need to login from PowerShell using the *Connect-AzAccount* command and choosing the *Work or School* option. Additionally you will need to run the command *Install-Module Az.ConnectedMachine -Force* before continuing.

PowerShell

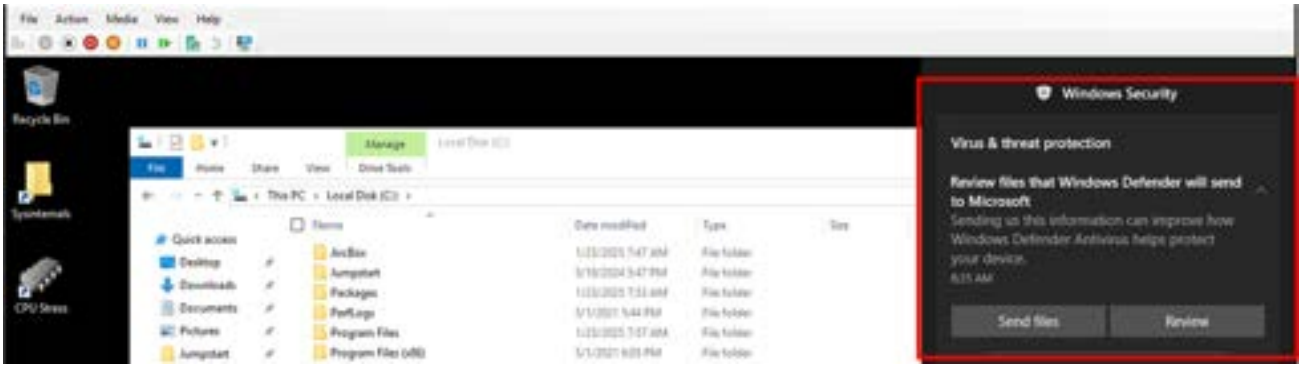
```
$vm = Get-AzConnectedMachine -ResourceGroupName "<Insert Resource Group Name>" -
$mdePackage = Invoke-AzRestMethod -Uri https://management.azure.com/subscription

$protectedSetting = @{
    "defenderForEndpointOnboardingScript" = ($mdePackage.content | ConvertFrom-J
}
```

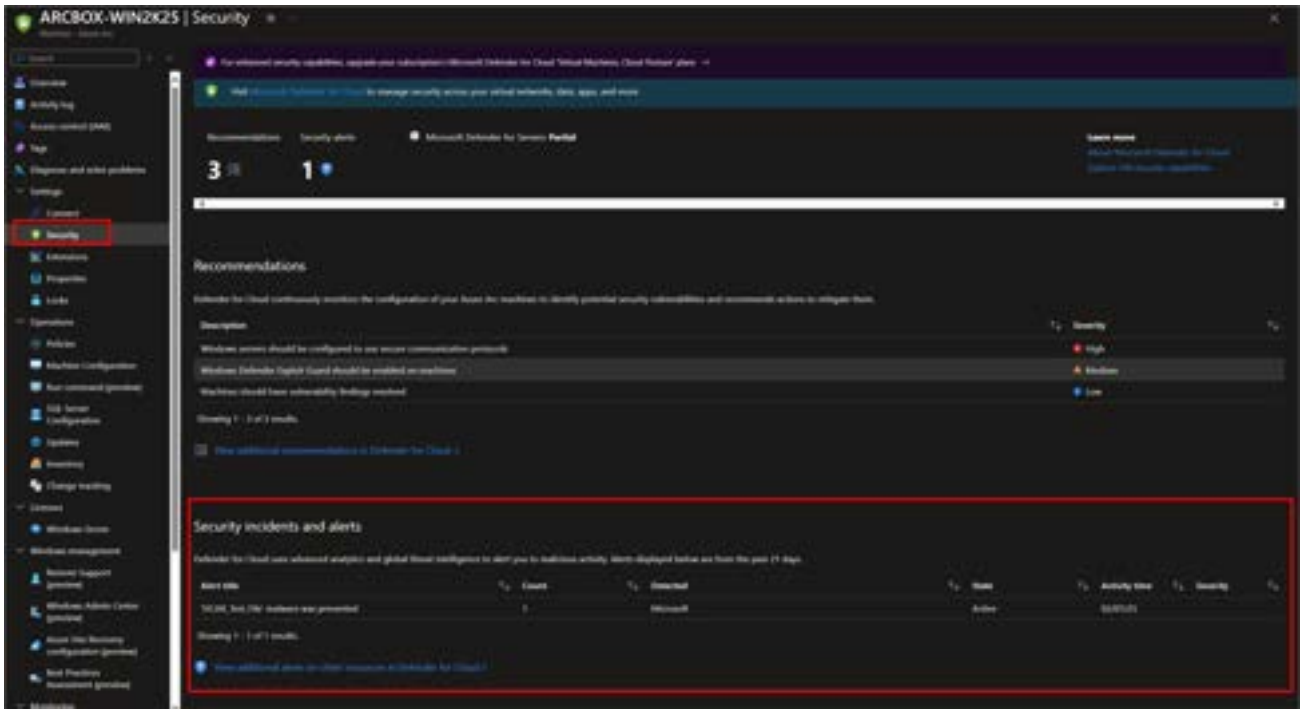
```
$Setting = @{
    "azureResourceId" = $vm.Id
    "vNextEnabled" = $true
}
New-AzConnectedMachineExtension -Name 'MDE.Windows' -ExtensionType 'MDE.Windows'
```

- [ ] 3. Wait for the installation of the MDE.Windows extension to be successful, then on the *ArcBox-Client* VM, go to Hyper-V manager and logon to the *ArcBox-Win2k25* VM using *JS123!!* as the Administrator password. Create an empty text file in the C: drive. Use the browser to open this site and find the malware test string. Copy the test string to the empty text file you created and save it as test.txt on the C: drive.

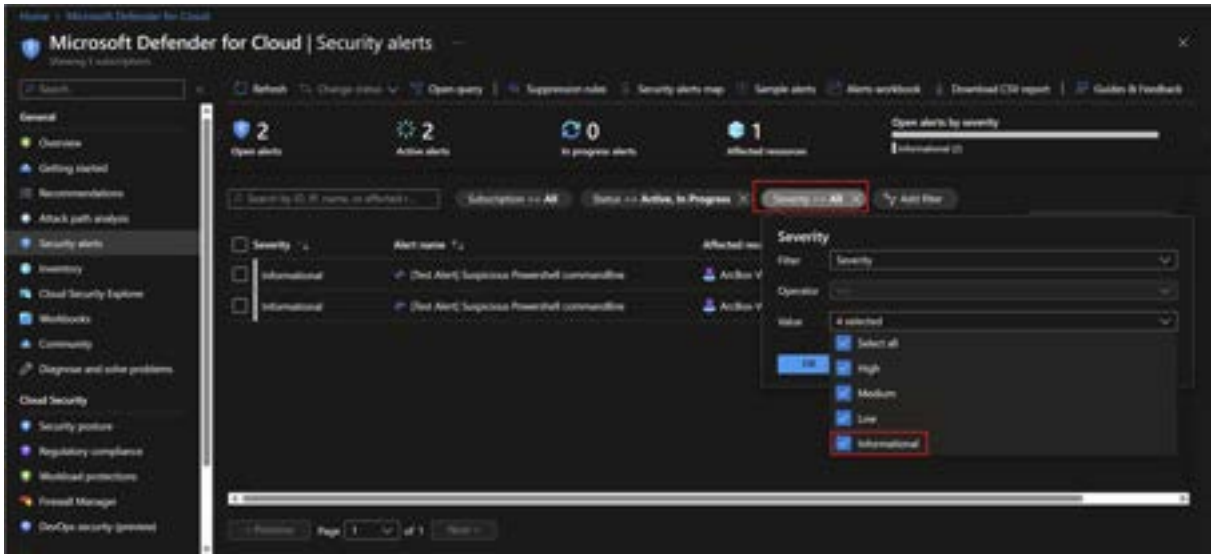
- [ ] 4. The local Anti-Malware software should detect this simulated threat on the ArcBox-Win2k25 VM. Navigate to the Security tab of the *ArcBox-Win2k25* Arc-enabled server in the portal

- [ ] **It might take up to 20 minutes or more for the alert to show up in the portal, you can move to the next exercise or lab and come back later to see the results if you prefer. This way, you can make better use of your time in this workshop**.





5. You can also see the alerts from the *Defender for Cloud* portal, in the *Security alerts* pane. **If you don't see the alerts, make sure to select the Information severity in the filters.**.



Task 1 has been completed

Exercise 2 - Configure vulnerability assessment on your machines

Objective

This exercise will walk you through enabling a vulnerability assessment solution on your machines.

Estimated Time to Complete This Lab

20 minutes

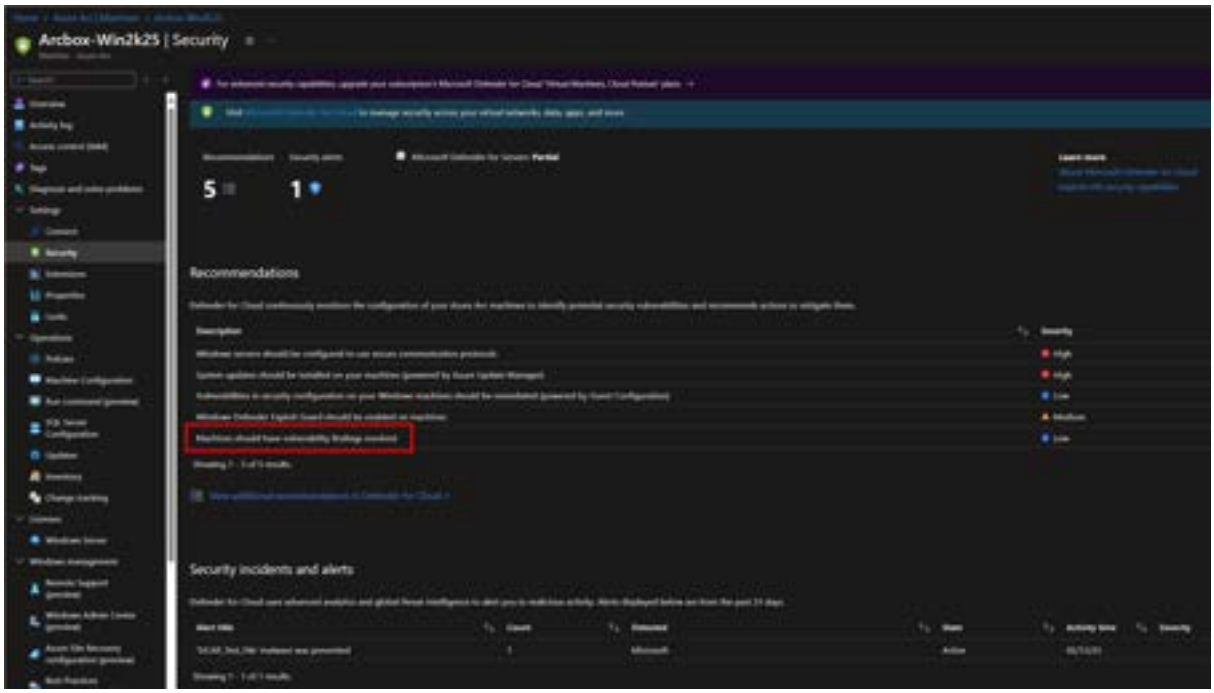
Explanation

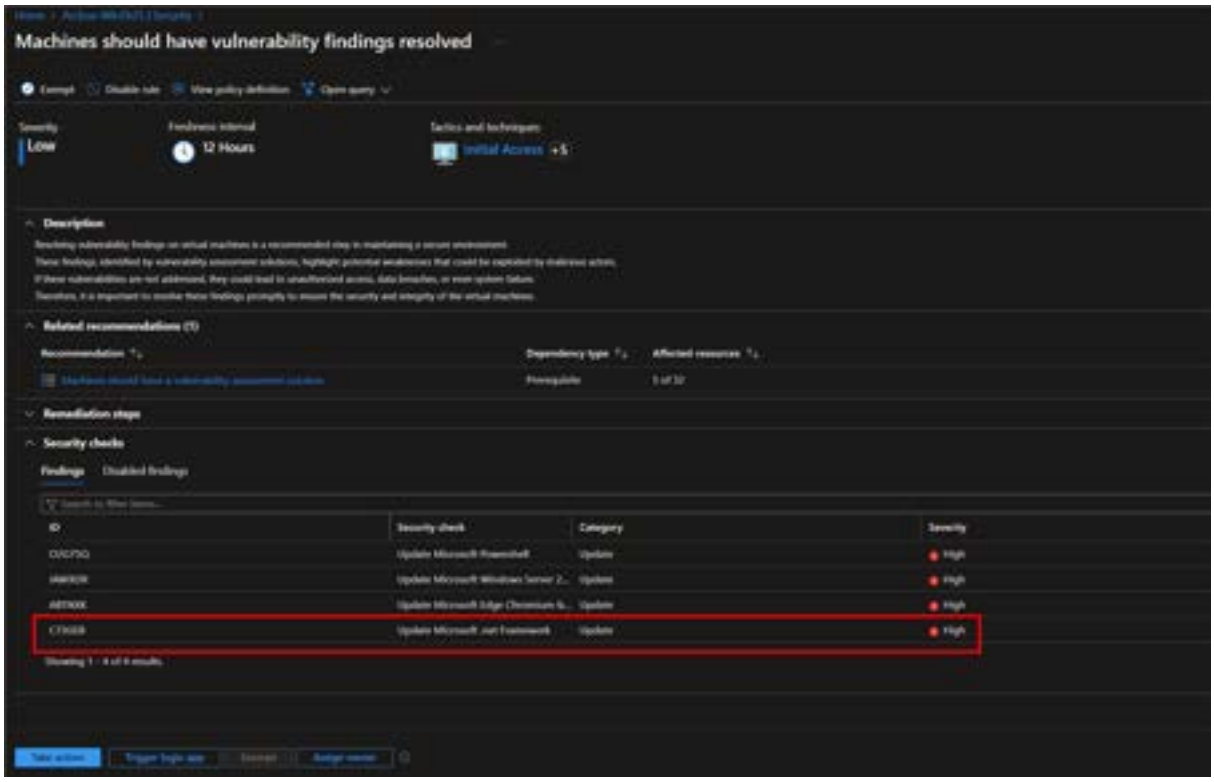
In this exercise, you will learn how to configure a vulnerability assessment on your machines and view detected vulnerabilities.

Task 1: Configure vulnerability assessment on your machines

1. After about 20-30 minutes from setting up Defender plans for servers (done in Lab01), you should start seeing recommendations for the Arc-enabled machines in the "Security" blade.

2. You should find the recommendation *Machines should have vulnerability findings resolved* if the vulnerability assessment has been enabled automatically on the subscription. Click on the recommendation.

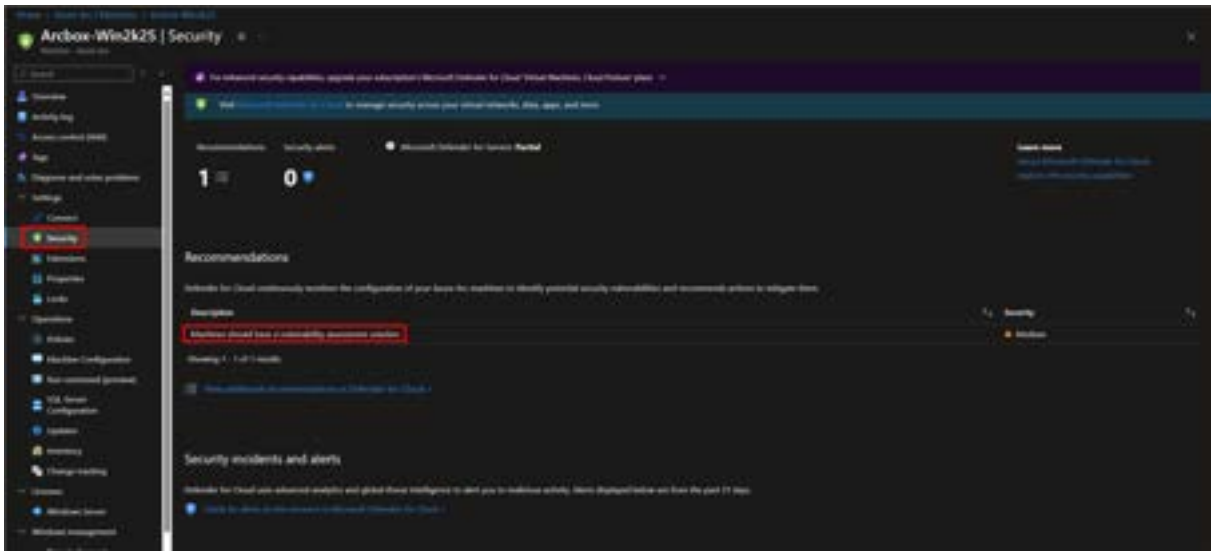




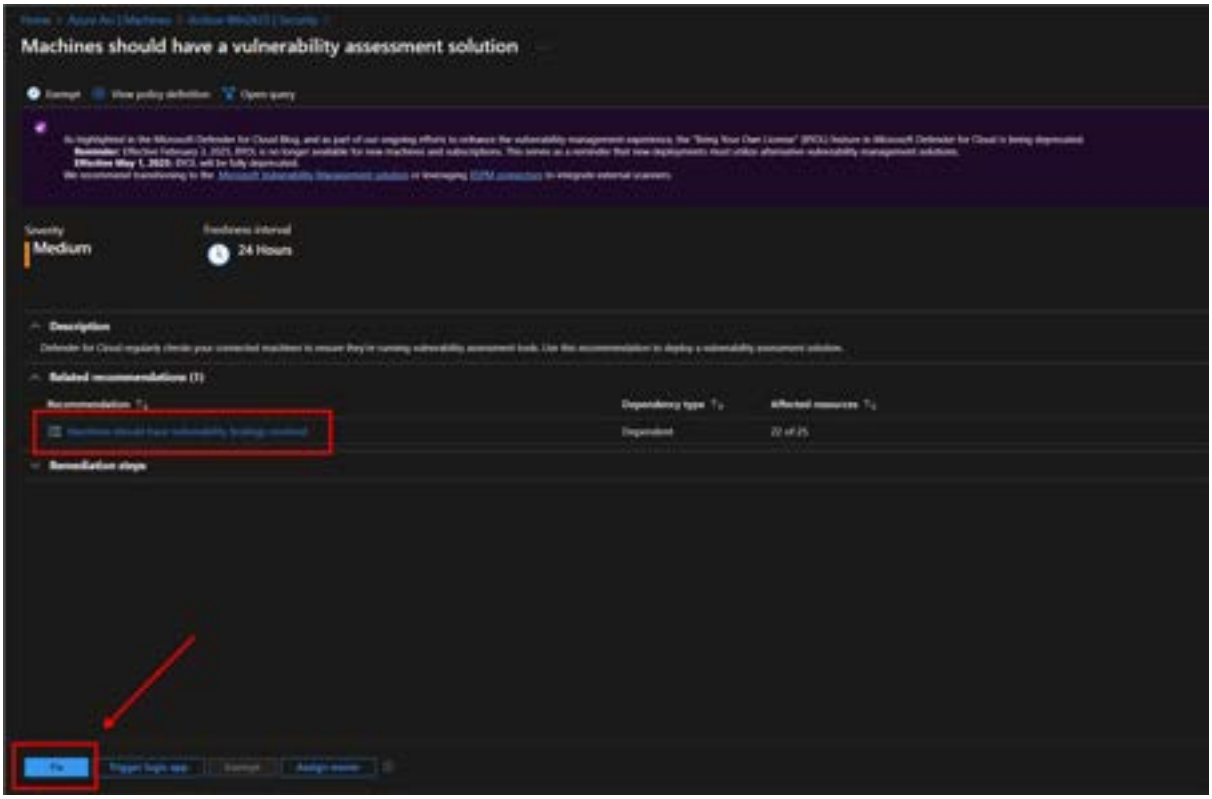
.

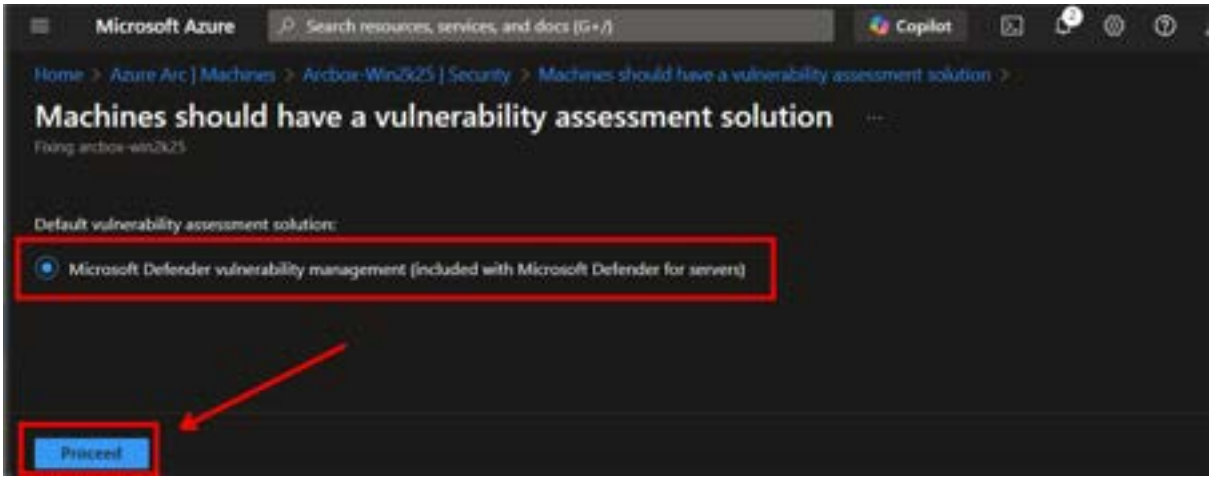
This means that Vulnerability Assessment is working on the server and you can end this lab at this point.

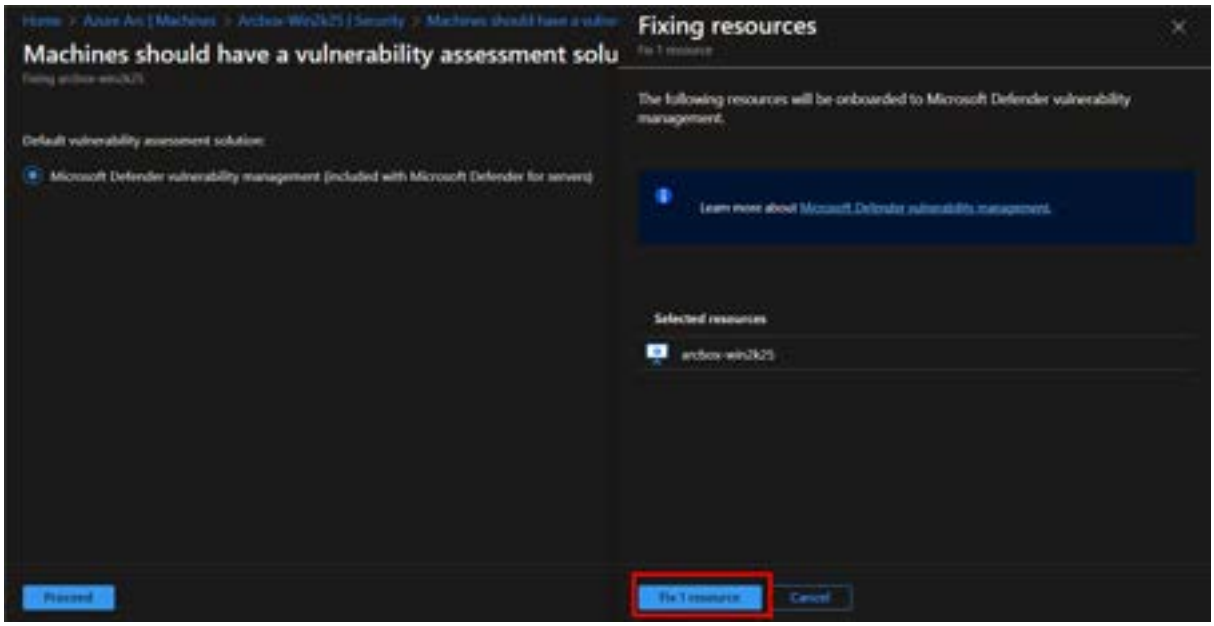
☐ 3. If you do not see this recommendation, click on the *Machines should have a vulnerability assessment solution*

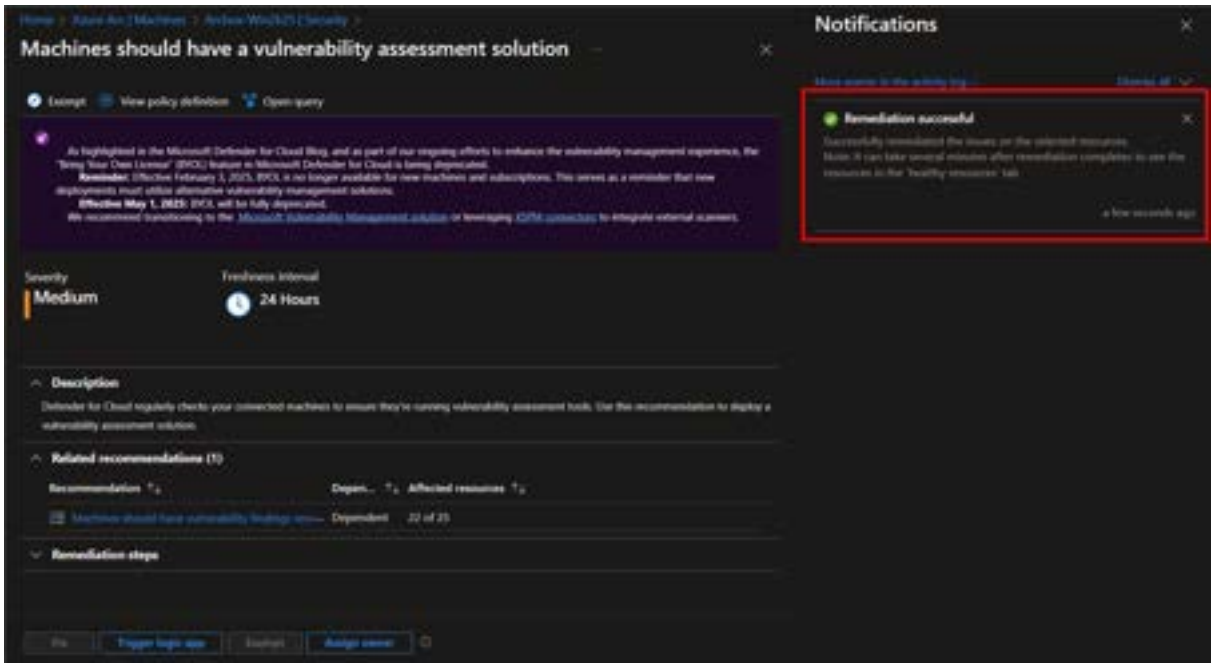


☐ 4. Click on the "Machines should have a vulnerability assessment solution" recommendation and click "Fix"


Machines should have a vulnerability assessment solution
Medium
24 Hours
Description
Related recommendations (1)
Remediation steps
Fix


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Azure Arc | Machines > Arcbox-Win2k25 | Security > Machines should have a vulnerability assessment solution >
Machines should have a vulnerability assessment solution
Fixing: arcbox-win2k25
Default vulnerability assessment solution:
Microsoft Defender vulnerability management (included with Microsoft Defender for servers)
Proceed





Task 1 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB04: Governance across Azure Arc-enabled servers using Azure Policy and Azure Machine Configuration

---

Student Lab Manual

Table of Contents

Exercise 1 - Enforce governance across your Azure Arc-enabled servers using Azure Policy

---

Objective

In this module you will use Azure Policy to Audit Arc-enabled Linux servers that have a certain application installed.

Estimated Time to Complete This Lab

30 minutes

Explanation

In this exercise, you will learn how to use Azure Policy to audit Arc-enabled Linux machines that have a specific application installed.

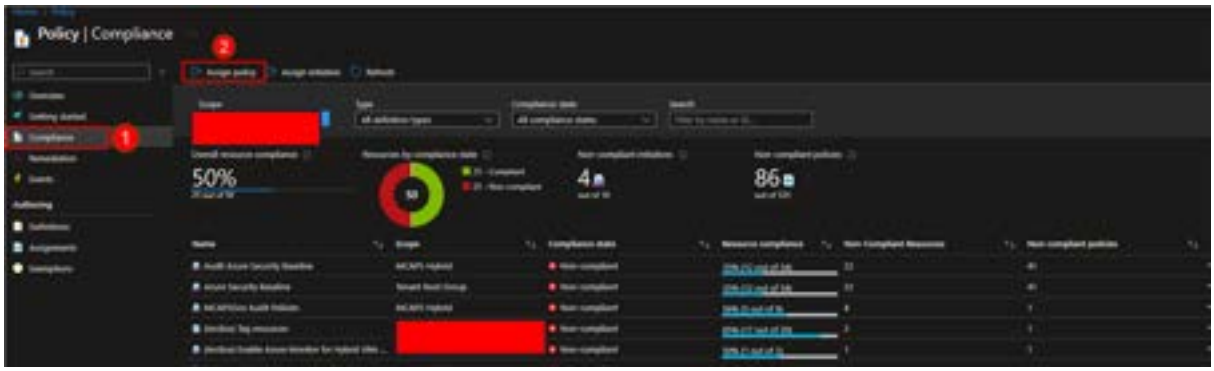
Task 1: Assign a built-in Azure Policy to the Arc resource group

☐ Azure policy can be assigned at Management Group, Subscription or Resource Group scope.

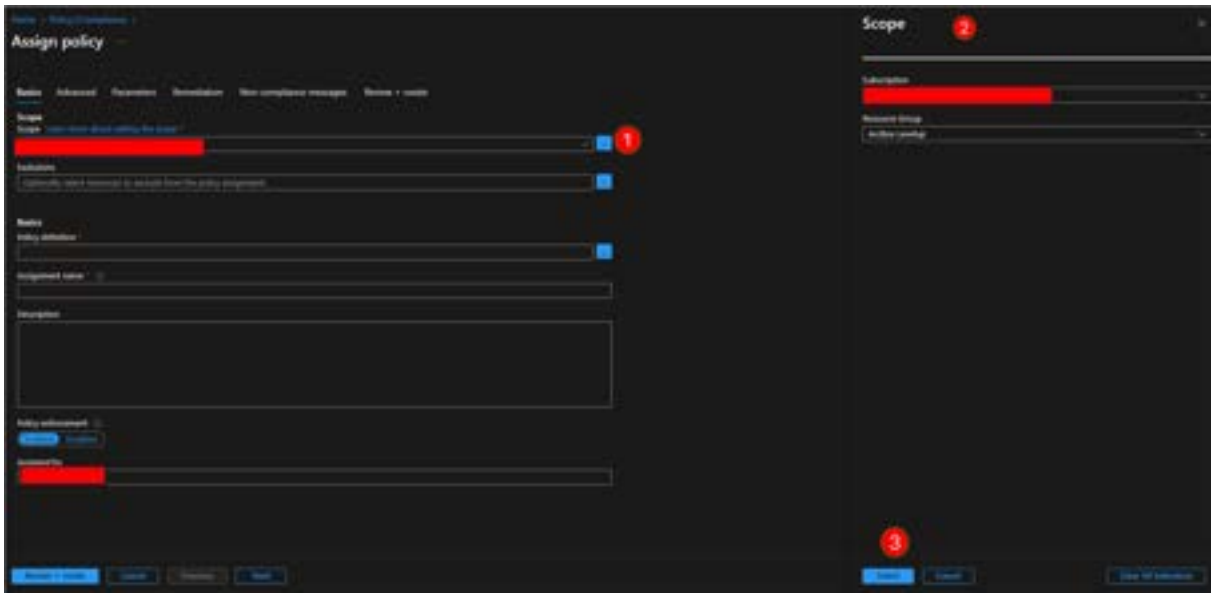
- In this scenario we will use the Resource Group scope.
- We will show two ways to accomplish this first task. **First we will use the Azure portal** (but if you prefer to use Powershell then skip to that section at the end of this first task)

☐ 1. In the Azure Portal search for the "Policy" resource and navigate to it.

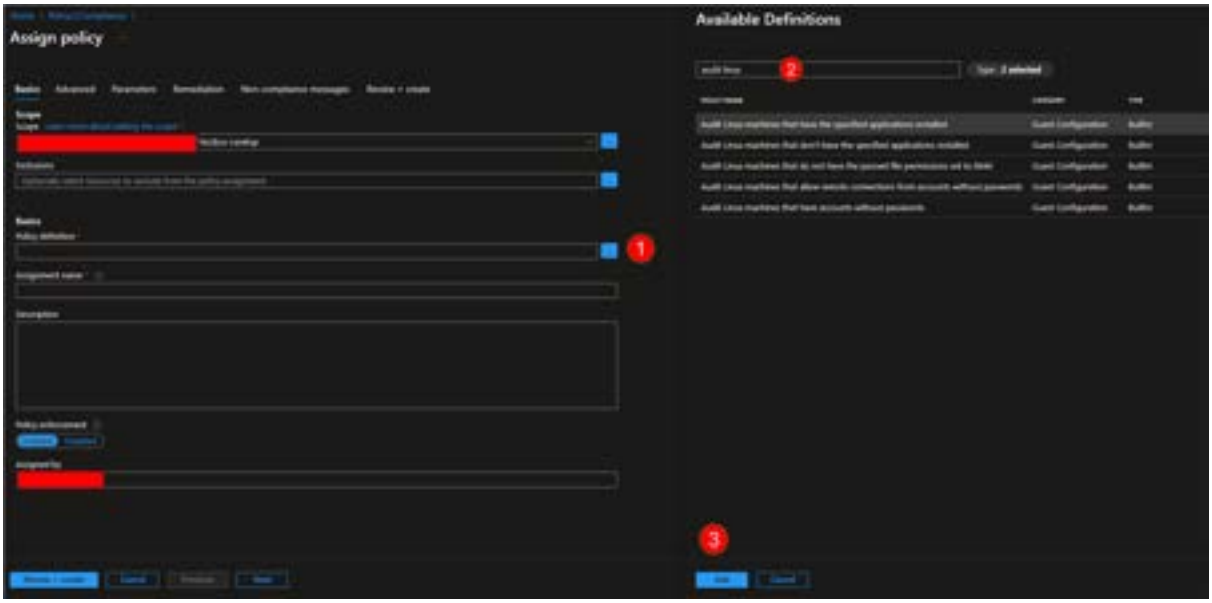
☐ 2. Click on "Compliance" in the left mene then click **"Assign policy"**.



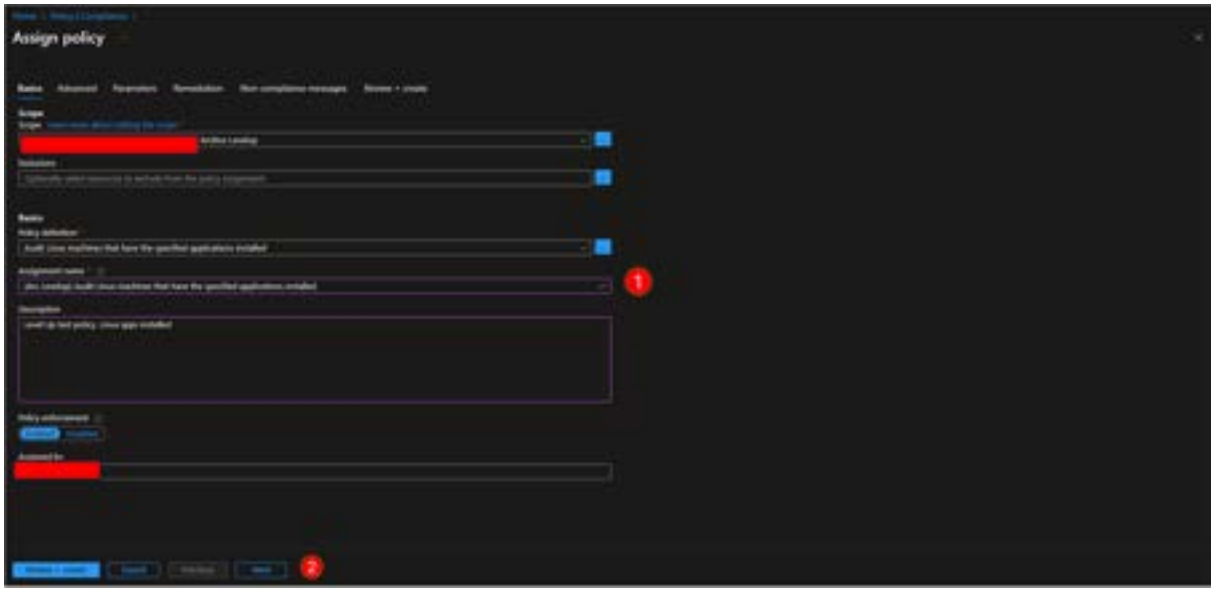
☐ 3. Set the scope of the policy assignment to the Subscription and the Resource Group as shown below:



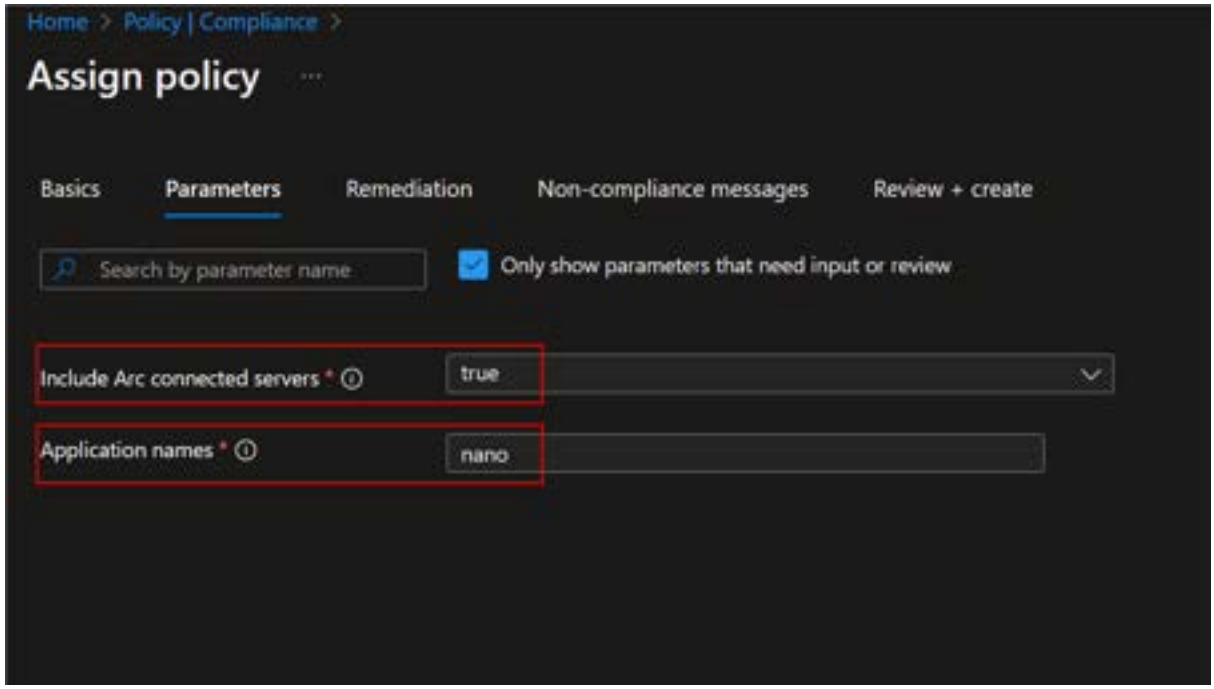
☐ 4. Click on the ellipsis next to "Policy definition". This opens the "Available Definitions" panel, where you can start searching for **"Audit Linux machines that have the specified applications installed"** policy which belongs to the "Guest Configuration" category.
Select this policy as shown below:



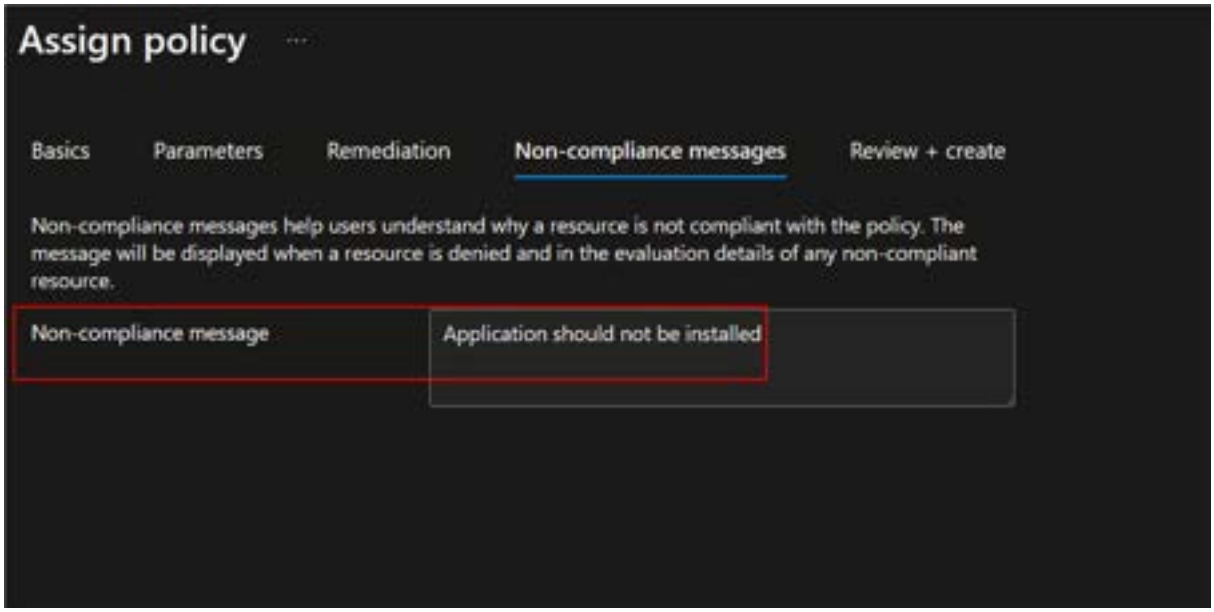
5. Modify the "Assignment name" so that it would be easy to identify our policy in the compliance list later as shown below, then click "Next" twice to reach the "Parameters" tab.



6. On the "Parameters" Screen, set the **"Include Arc connected servers"** to **"true"** and then set the name/s of the application/s you want to audit the Linux servers for. If you have more than one application then include them in a semicolon separated list enclosed in single quotes e.g. 'App1; App2; App3'.



☐ 7. Move to the "Non-compliance message" tab to add a message of your choice.



☐ 8. Next move to the "Review + create" tab and click **"Create"** to assign the policy.

☐ If you want to use **Powershell as an alternative method** to assign the policy, then the following procedure accomplishes the same as the portal method explained above.

- Run the following powershell commands making sure you use the correct resource group name

```PowerShell
  '{
     "IncludeArcMachines":{
```

```
      "value":"true"
     },
    "ApplicationName": {
       "value":"nano"
     }
}' > .\params.json
$ResourceGroup = Get-AzResourceGroup -Name 'ArcBox'
$Policy = Get-AzPolicyDefinition -BuiltIn | Where-Object {$_.DisplayName -eq '
New-AzPolicyAssignment -Name '(Arc Workshop) Audit Linux machines with nano in
```

Task 1 has been completed

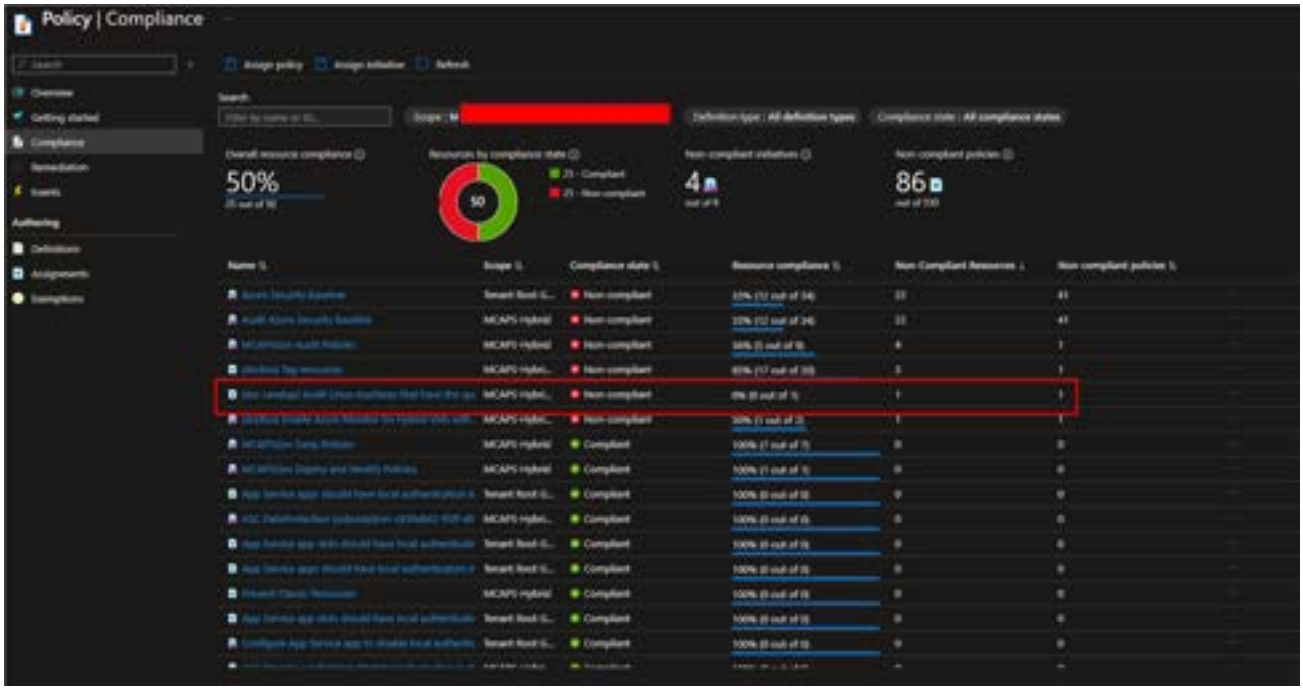
Click **Next** for the next task or **go back to the main table of content**

Task 2: Examine the policy compliance

- [ ] The creation of the assignment and for it to take effect and get evaluated might take some time. You can keep refreshing the "Compliance" list until you can see an indication that there is at least one resource which is non-compliant with the policy we created (this depends on how many Arc-connected Linux servers with the specified applications we have).

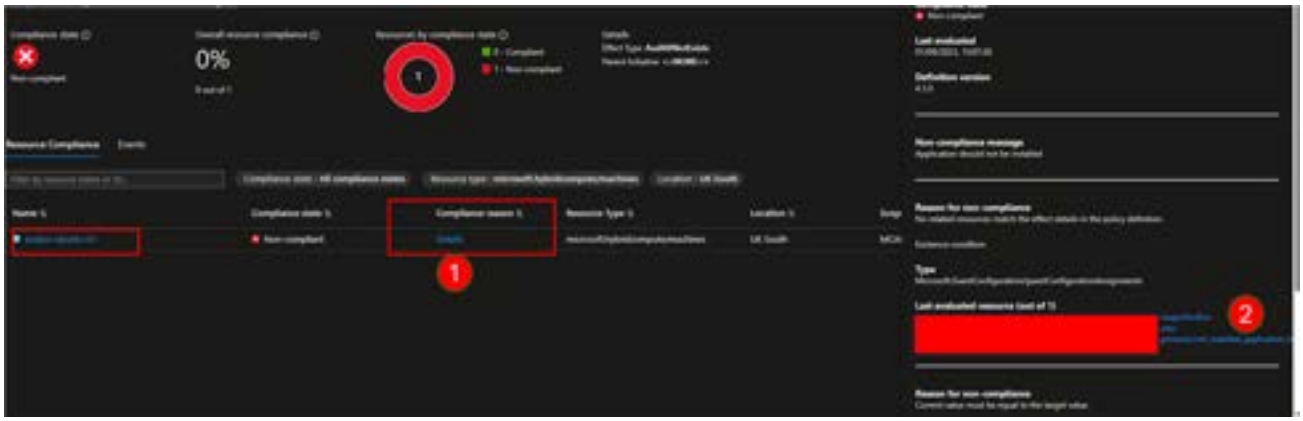
- [ ] If this does not happen in a reasonable time then you can go to **task 3** where there is another view that might faster to show the compliance indication**.

- [ ] We can also attempt to use PowerShell force a policy scan (see note at end of this task) which **might** improve the speed to populate the compliance dashboard.

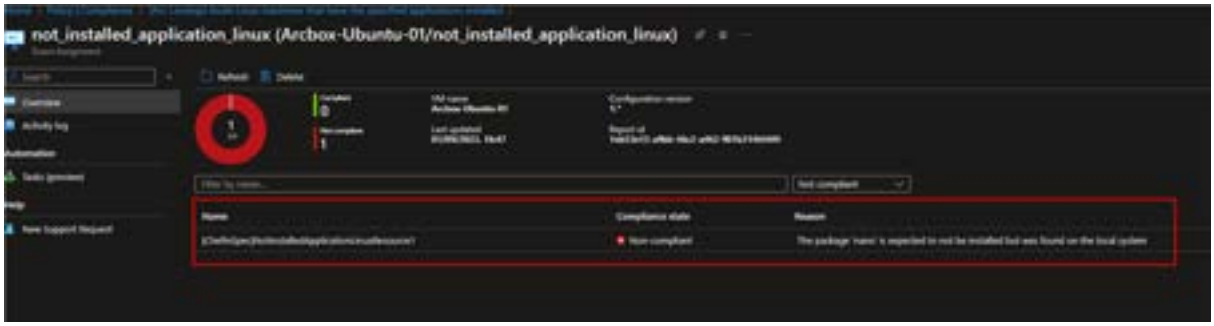


- [ ] 1. Click on the policy from the name column and this will take you to a more detailed view of the specific policy compliance as shown below. You can then click on "Details" which will open another panel on the right hand side.

- [ ] If the **"Details"** link is not ready yet then you will need to wait for it, or try **task 3** for another way of looking at the compliance of specific servers, which might be faster to populate.



2. Click on the link below "Last evaluated resource ...". This will open the "Guest Assignment" screen showing exactly why that specific server is not compliant with the policy.



- The steps above helps you identify non-compliant resources and then you can act on resolving the non-compliance reasons.

(Optional): As mentioned at the beginning of this task, **to force a policy scan** we can use the Start-AzPolicyComplianceScan PowerShell command.

For example the following PowerShell commands will focus the scan on our resource group, run the scan as a job and wait for it to complete in the background. Make sure you use the correct resource group name:

```PowerShell
$job = Start-AzPolicyComplianceScan  -ResourceGroupName "ArcBox" -AsJob
$job | Wait-Job
```

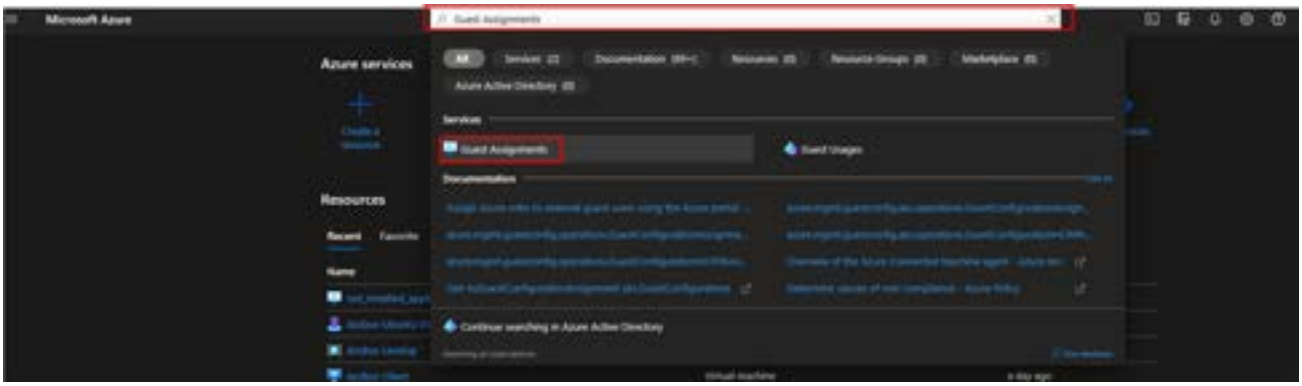
Task 2 has been completed

Click **Next** for the next task or **Go back to the main table of content**

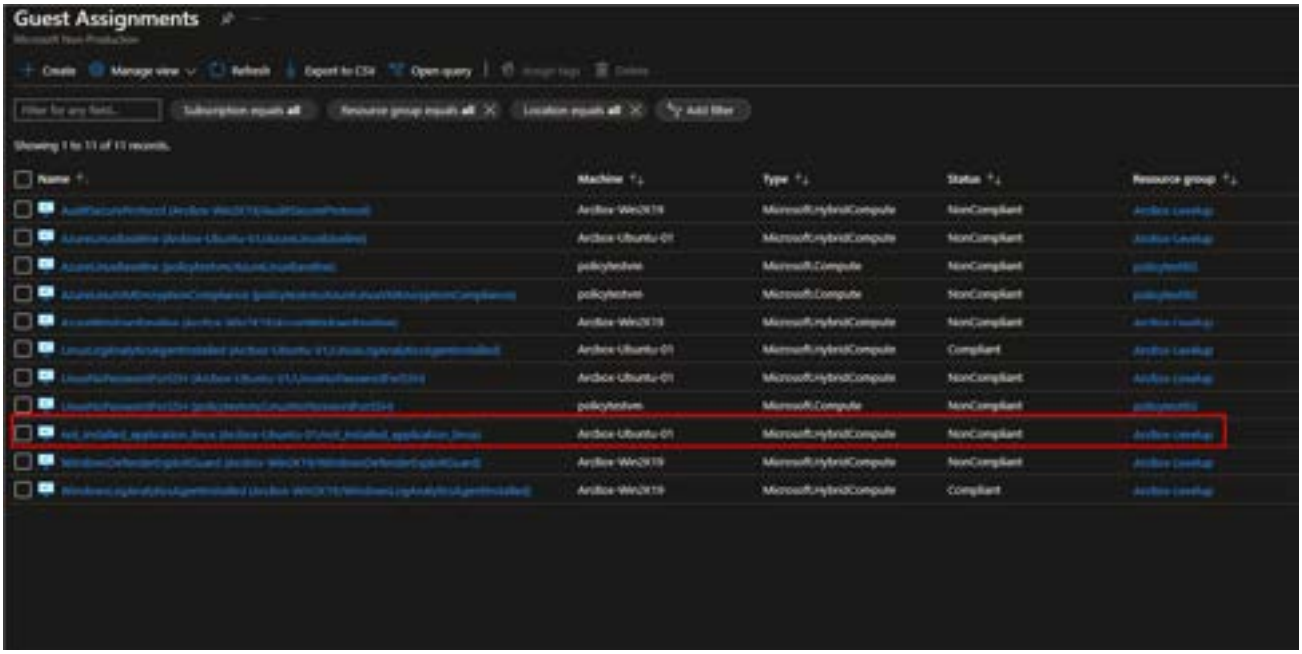
Task 3: Using the Guest Assignments views directly

As mentioned in **task 2**, the policy compliance dashboard can sometimes take a long time before it is updated with the accurate compliance details.

1. We can use a direct route to view the **"Guest Assignments"** for each resource by searching for "Guest Assignments" from the Azure portal and selecting it.



2. You can now look at the compliance of the individual resources and identify the ones that are affected by our policy assignment.



3. Click on the identified policy/resource combination and this will take you to the screen that we saw earlier at the end of **task 2**, showing the details of the compliance/non-compliance.

Task 3 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 2 - Configure your Azure Arc-enabled servers using Azure Machine Configuration

Objective

In this module, you will learn to create and assign a custom Machine Configuration to an Azure Arc-enabled Windows servers to create a local user and control installed roles and features.

Estimated Time to Complete This Lab

30 minutes

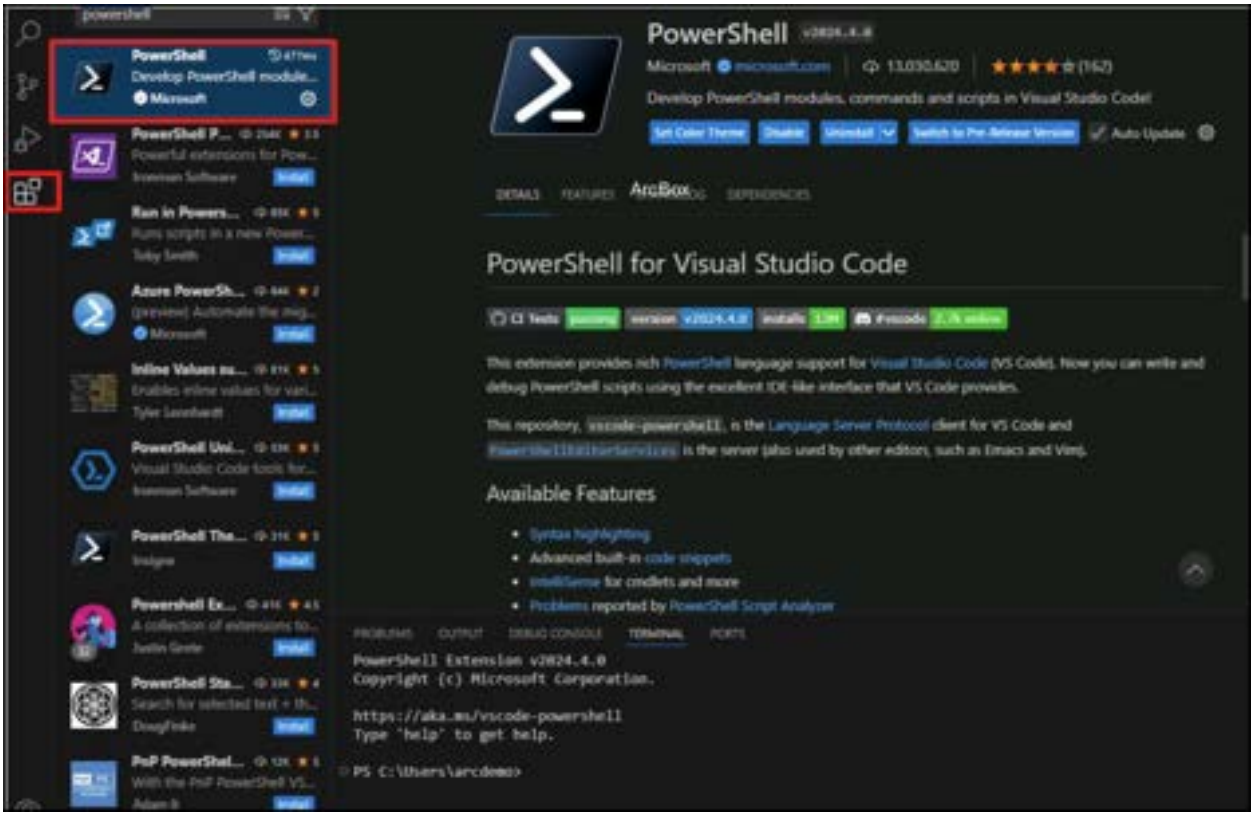
Explanation

In this exercise, you will learn how to use Azure Machine Configuration to configure an Arc-enabled machine declaratively by creating a local user and installing a Windows Server role.

Task 1: Create Machine Configuration - custom configurations for Windows

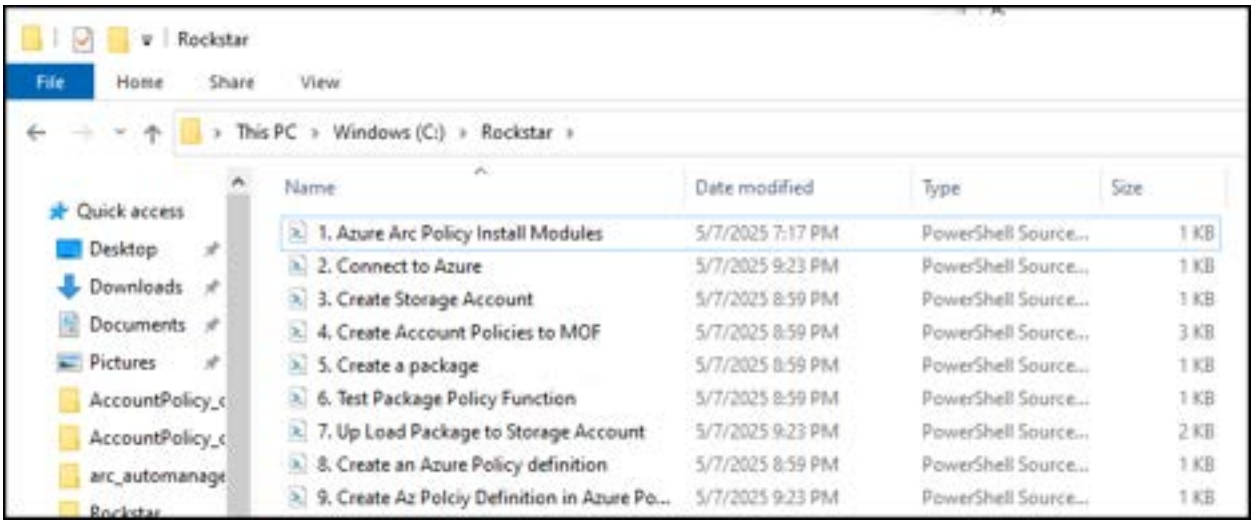
We will be using the **ArcBox Client** virtual machine for the configuration authoring.

1. RDP into the *ArcBox-Client* VM
2. Open **Visual Studio Code** from the desktop shortcut. Check if the Microsoft PowerShell Plugin is installed, if not then install it.

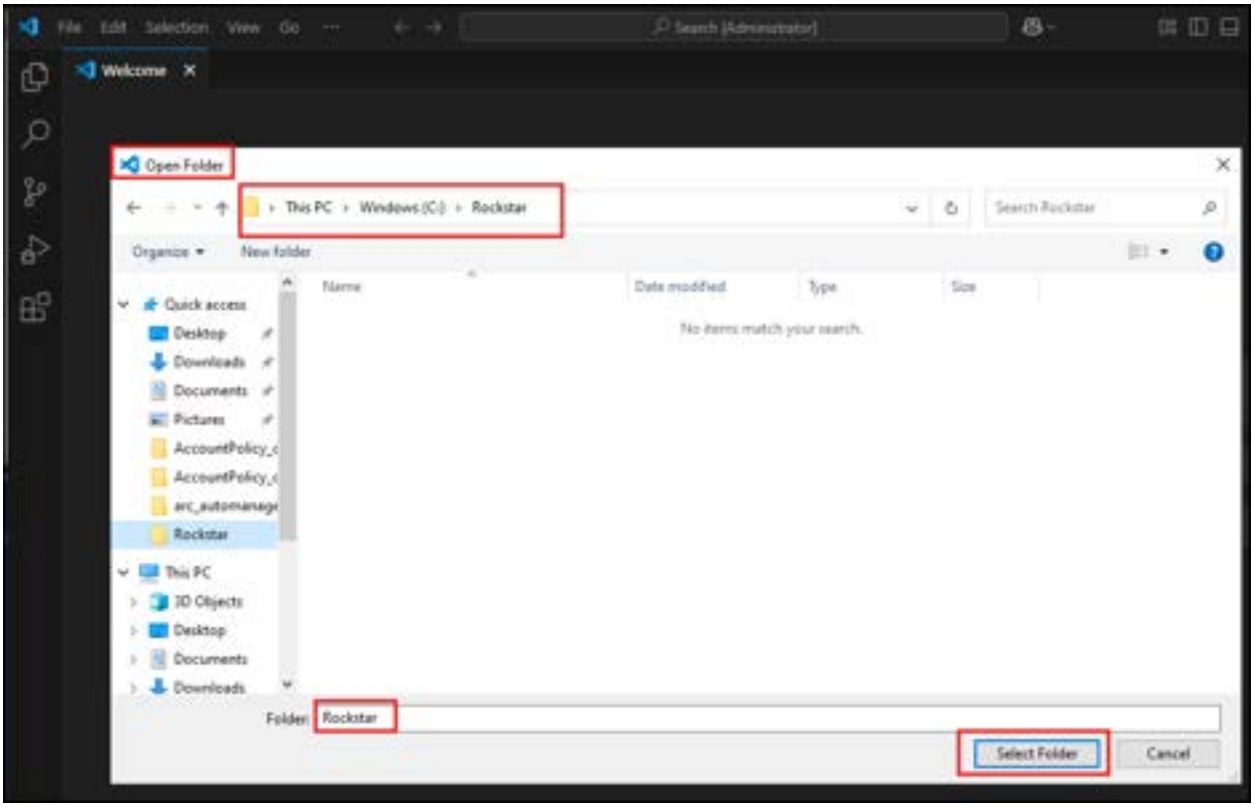


3. Copy Rockstar.zip from https://github.com/microsoft/Azure_Arc_Workshop/raw/main/scripts/Rockstar.zip
4. Unzip contents to C:\Rockstar\

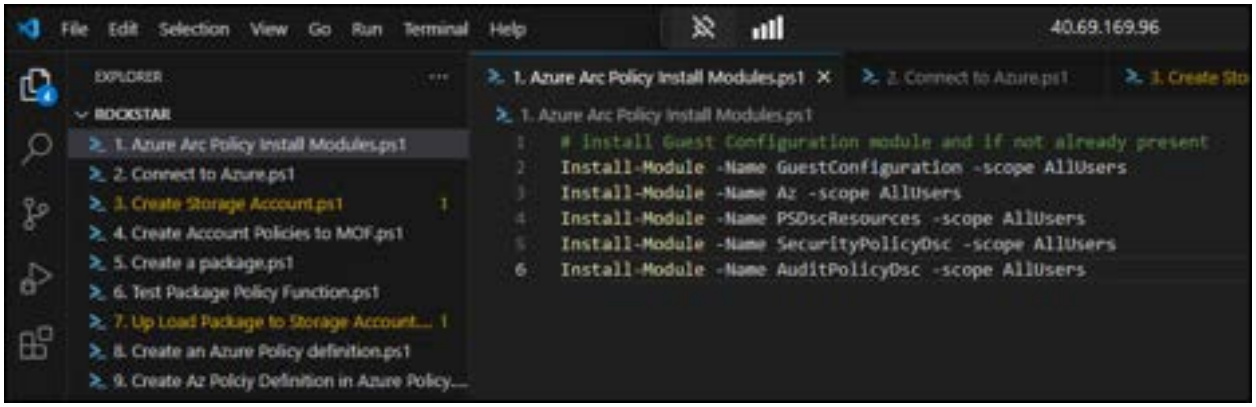
Custom configuration for Windows



5. In **Visual Studio** go to **File**, **Open Folder** and select **Rockstar.**



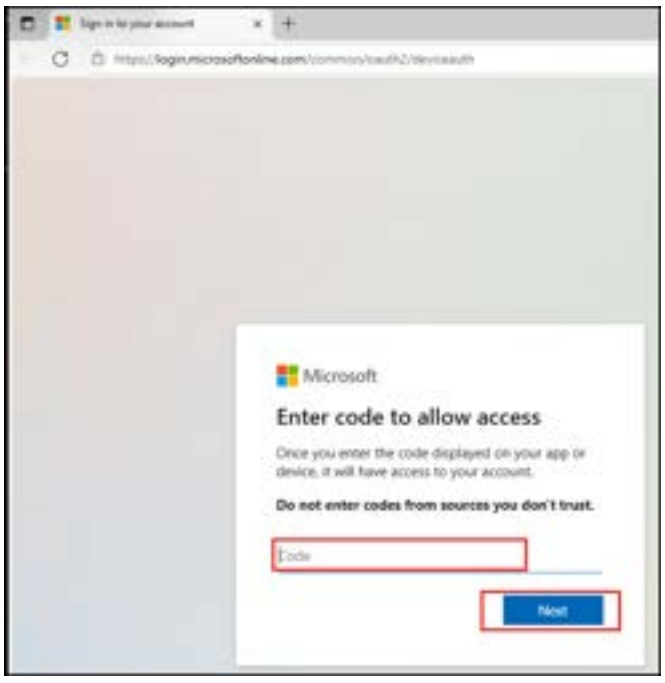
6. Start with code snippet **1** and run each in order until you finish with **9**.



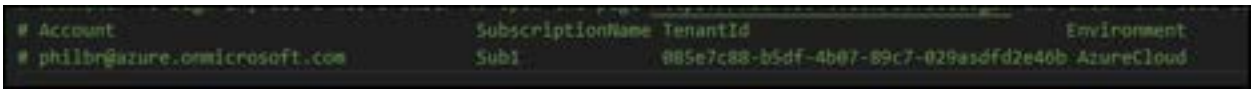
[!knowledge] To run each additional code snippet you paste in VSCode, highlight the code you need to run and press **F8**.

[!knowledge] You can also run the rest of the code snippets in PowerShell directly without using VScode if you prefer.

7. The code snippet **1** is to install the required PowerShell modules.
8. The code snippet **2** is to **Connect using your own account**. You will need to sign in using the web browser. i.e To sign in, use a web browser to open the page **https://microsoft.com/devicelogin** and enter the code sample **dfkjdakfh** to authenticate
9. Connect to the Browser on **Arcbox-Client**. Enter Code and Click **Next.**

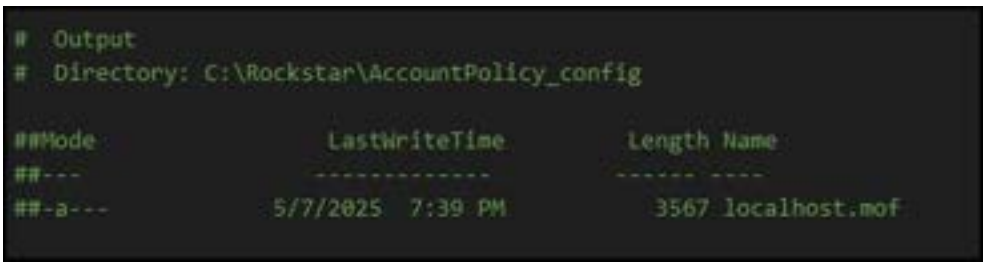


Note output:

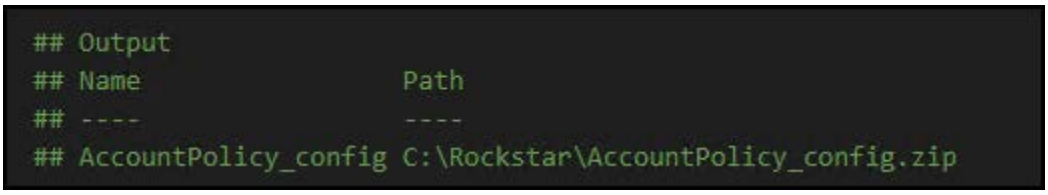


10. Code snippet **3** creates the **Storage Account**

11. Code snippet **4** creates the **MOF** File. This will check for compliance and do the remediation.



12. Code snippet **5** creates a package zip file to be uploaded to the storage account.



13. Code snippet **6** tests package zip file against the local machine.

14. Code snippet **7** Creates the AzStorageContainer, set the con and uploads the zip file.

15. Code snippet **8** Creates an Azure Policy definition.

16. Code snippet 9 Creates Az Policy Definition in Azure Policy.

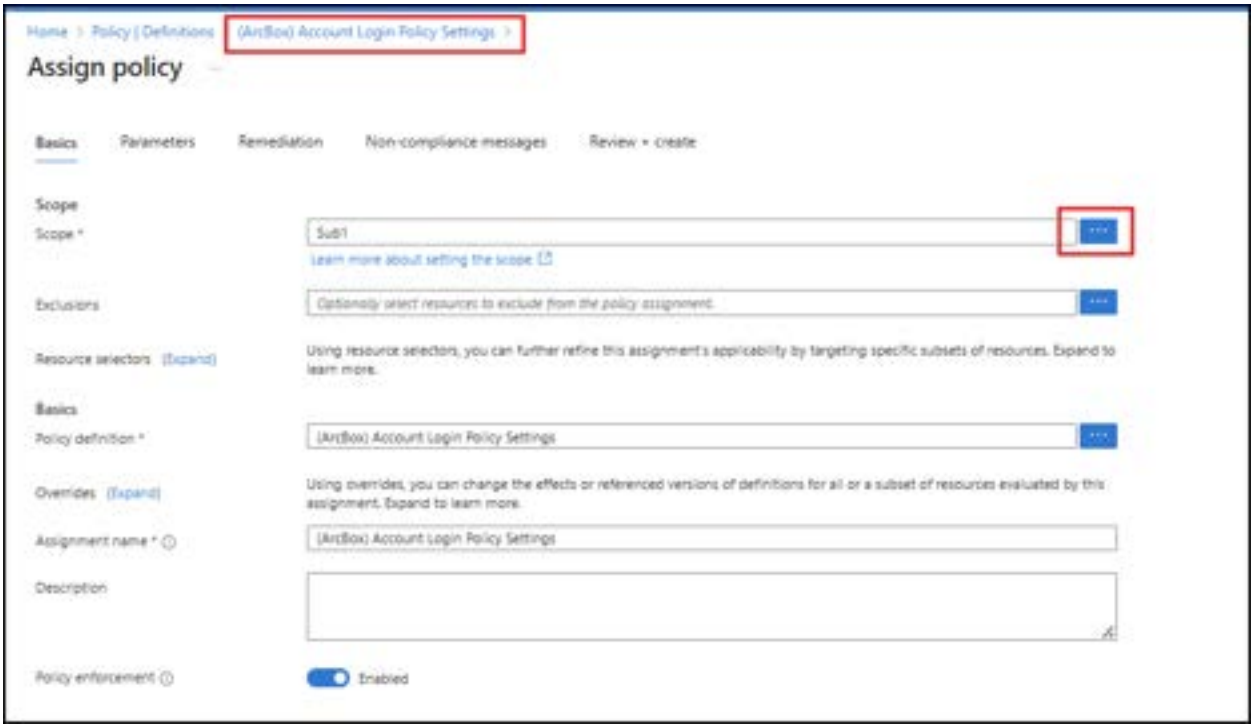
Assign Azure Policy

1. On the **Arcbox-Client** go to **https://portal.azure.com** go to **Policy**. Under **Authoring** and **Definitions**. Search **(ArcBox).**

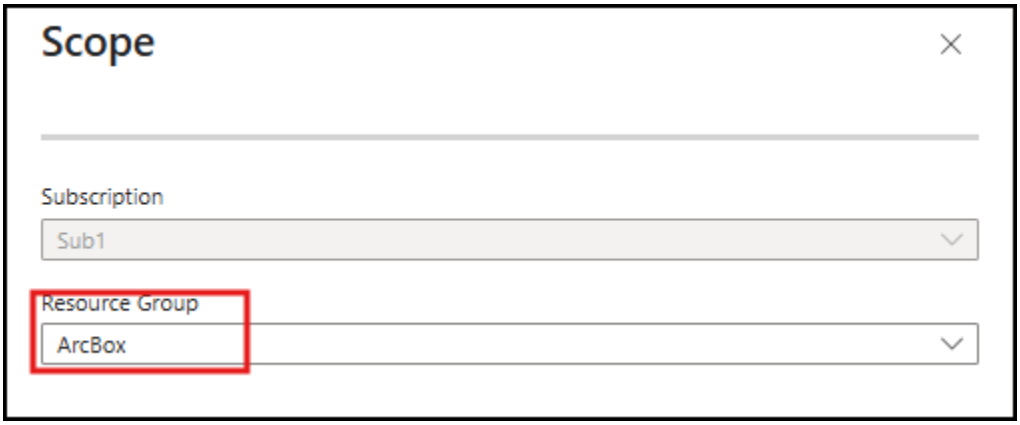
2. Click on **(ArcBox) Account Login Policy Settings.**

3. Click on **Assign Policy.**

4. Click on the **ellipse** to assign the **Resource Group.**

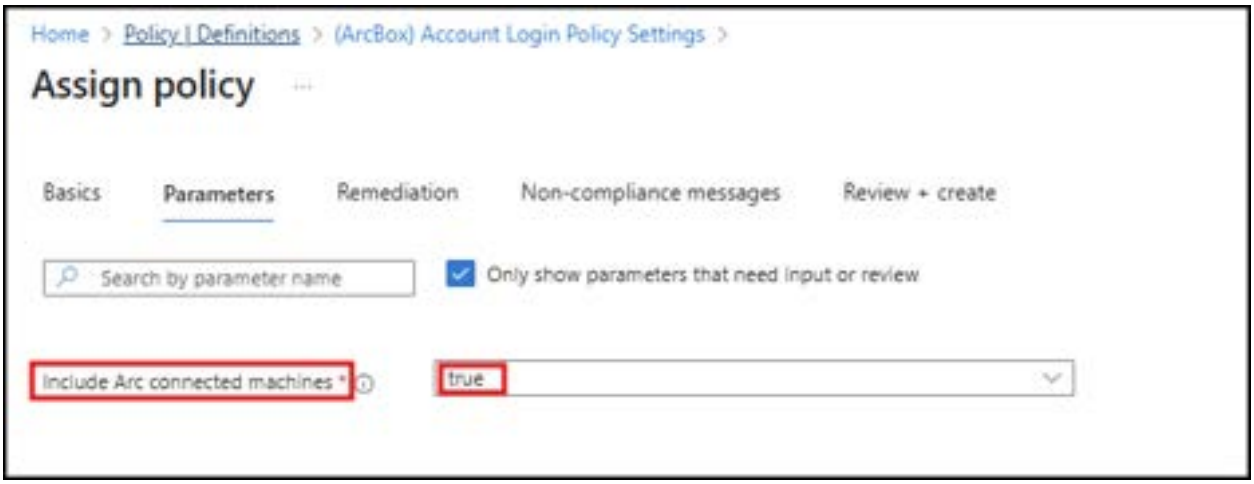


5. Select the **Resource Group Arcbox**. Do not forget to click on **Select** on the bottom of the windows



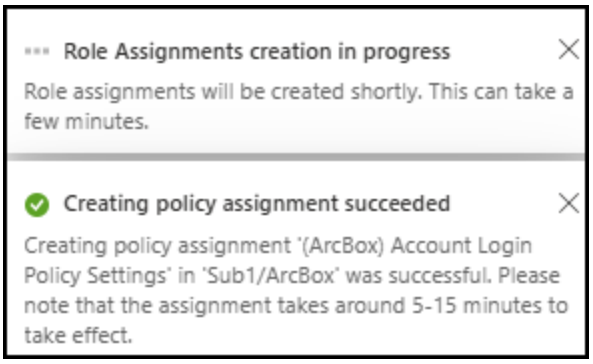
6. Select **Parameters**

7. Make **sure Include Arc connected machines** is set to **true.**



8. Click on **Review and Create.**

9. Click on **Create** on the Bottom of the window. The **Policy is Assigned**.



Check Azure Policy

1. To check policy compliance, in the Azure Portal, navigate to *Policy* -> **Compliance**

2. Set the scope to the resource group your instance of ArcBox is deployed to.

3. Filter for *(Arcbox)*

[!alert] **It may take 30 minutes or more for the policy remediation to be completed. You can move to the next exercise or lab and come back later to see the results if you prefer. This way, you can make better use of your time in this workshop**.

4. To get a Machine Configuration status for a specific machine, navigate to ***Machines – Azure Arc*** in the Azure Portal.
5. Click on ArcBox-Win2K25 -> **Machine Configuration.**

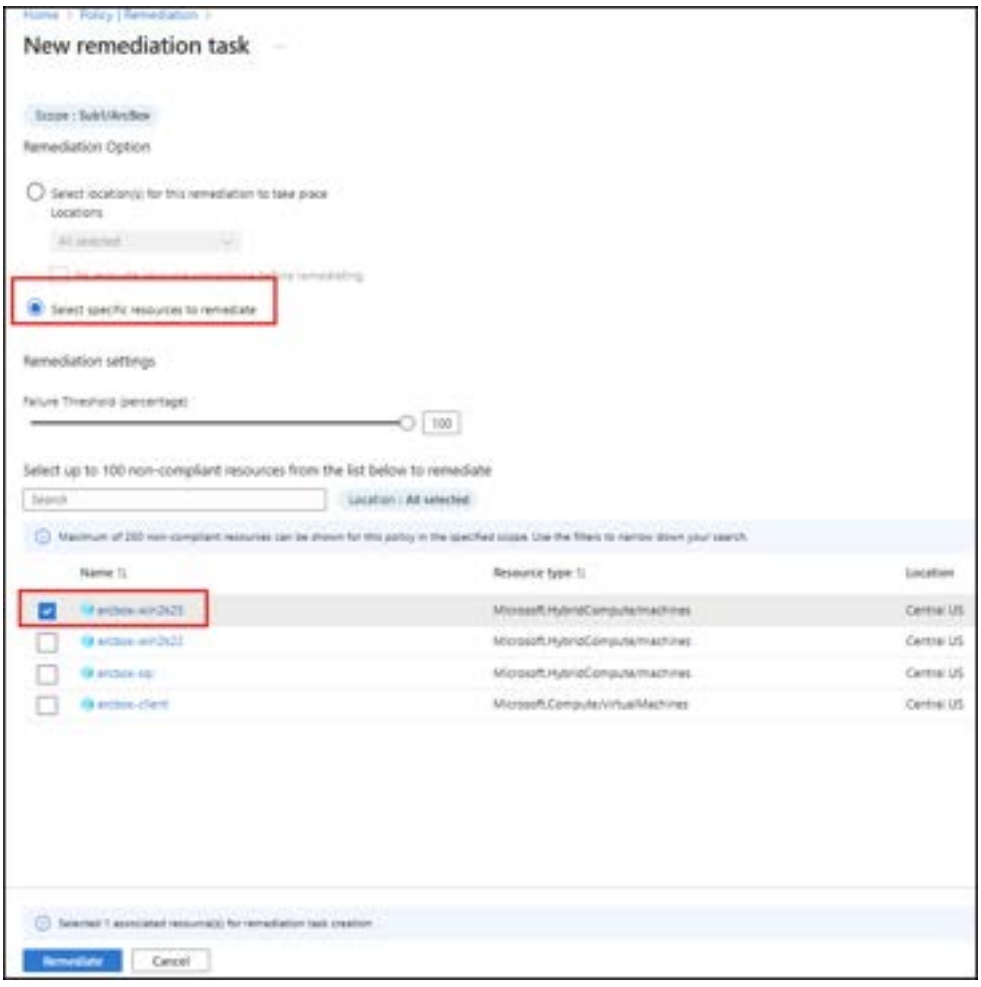
- If the status for *ArcBox-Win2K25/AccountPolicy_Config*.. is **not Compliant** or **Pending**, wait a few more minutes and click *Refresh*

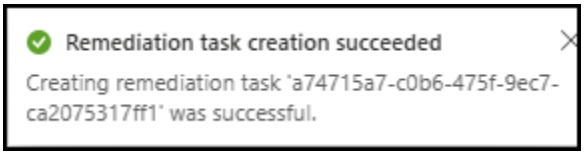
6. Click on *ArcBox-Win2K25/ AccountPolicy_Config*.. to get a per-resource view of the compliance state in the assigned configuration

Remediate The Azure Policy

1. To Remediate the Policy for a specific machine, navigate to ***Policy*** in the Azure Portal. Select **Remediation.** Filter for *(**Arcbox**).* *Select* ***(ArcBox) Account Login Policy Settings.***

2. **Select Resources to Remediate** and choose **Arcbox-win2k25.** Then Click on **Remediate**





[!knowledge]The **Azure PowerShell modules** are used for:

- Publishing the package to Azure storage
- Creating a policy definition
- Publishing the policy
- Connecting to the Azure Arc-enabled servers

[!knowledge]The **GuestConfiguration module** automates the process of creating custom content including:

- Creating a machine configuration content artifact (.zip)
- Validating the package meets requirements
- Installing the machine configuration agent locally for testing
- Validating the package can be used to audit settings in a machine
- Validating the package can be used to configure settings in a machine

[!knowledge] Version 3 of **Desired State Configuration module** is removing the dependency on MOF.
Initially, there is only support for DSC Resources written as PowerShell classes.
Due to using MOF-based DSC resources for the Windows demo-configuration, we are using version 2.0.5.

LAB05: Monitor your Azure Arc-enabled servers using Azure Monitor, Change Tracking and Inventory

In this lab, you will learn how to deploy the Azure Monitor agent to your Arc-enabled Windows and Linux machines, how to deploy the dependency agent to your Arc-enabled Windows machines, how to enable the *VM Insights* solution to start monitoring your machines using Azure Monitor, how to run queries on the Log analytics workspace and how to configure alerts. In addition, you will learn how to use the Change Tracking and Inventory features to track changes in your machine.

Student Lab Manual

Table of Contents

Exercise 1: Deploy Azure Monitor Agent (AMA) to your Arc-enabled machine using Azure Policy and define the Data Collection Rules

---

Objective

Use Azure Policy to enforce the installation of the Azure Monitor Agent (AMA) to your Arc-enabled machine.

Estimated Time to Complete This Lab

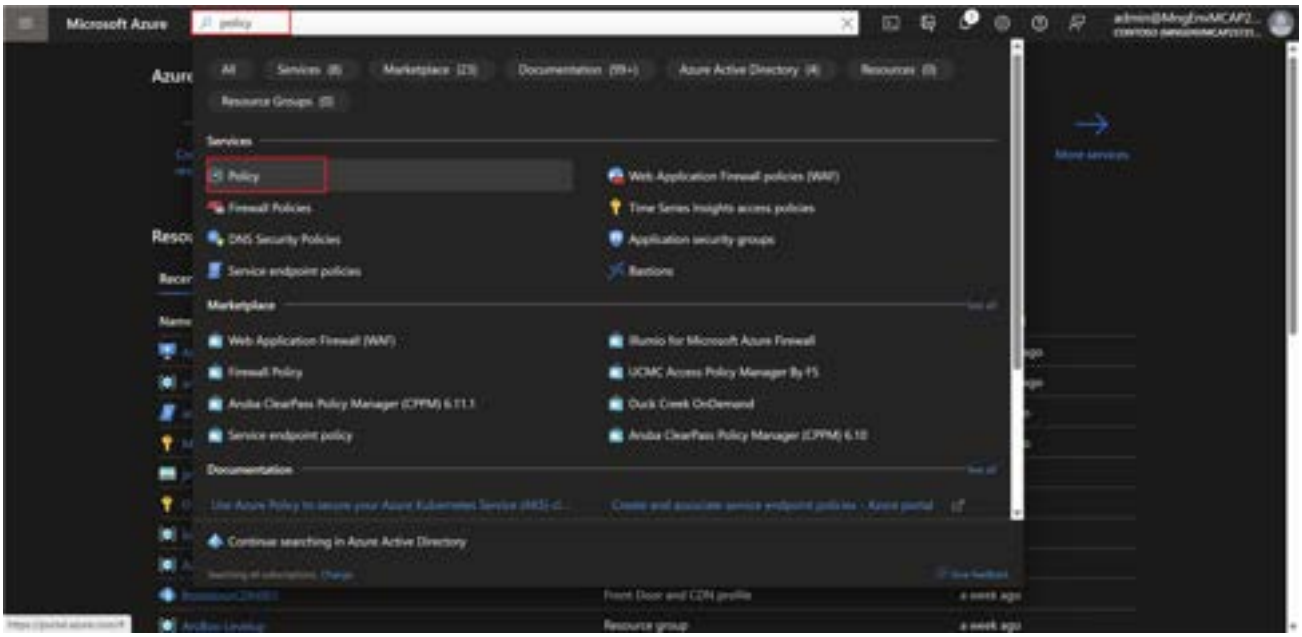
45 minutes

Explanation

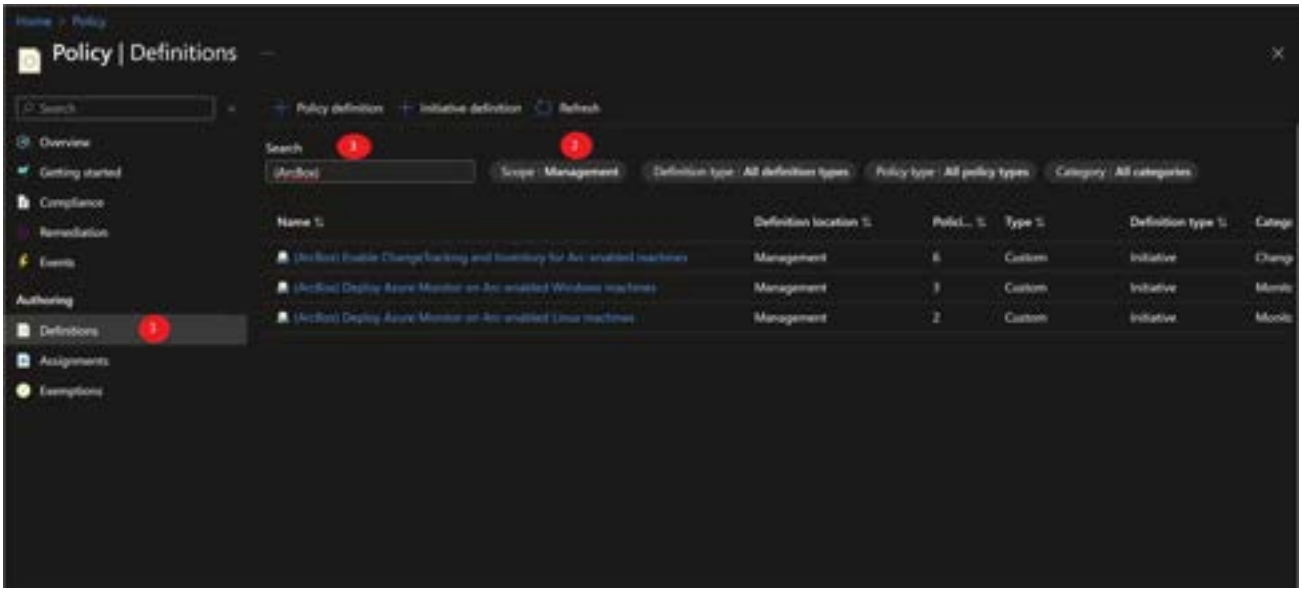
Azure Policy lets you set and enforce requirements for all new resources you create and resources you modify. VM insights policy initiatives, which are predefined sets of policies created for VM insights, install the agents required for VM insights and enable monitoring on all new virtual machines in your Azure environment.

Task 1: Deploy the Azure Monitor Agent

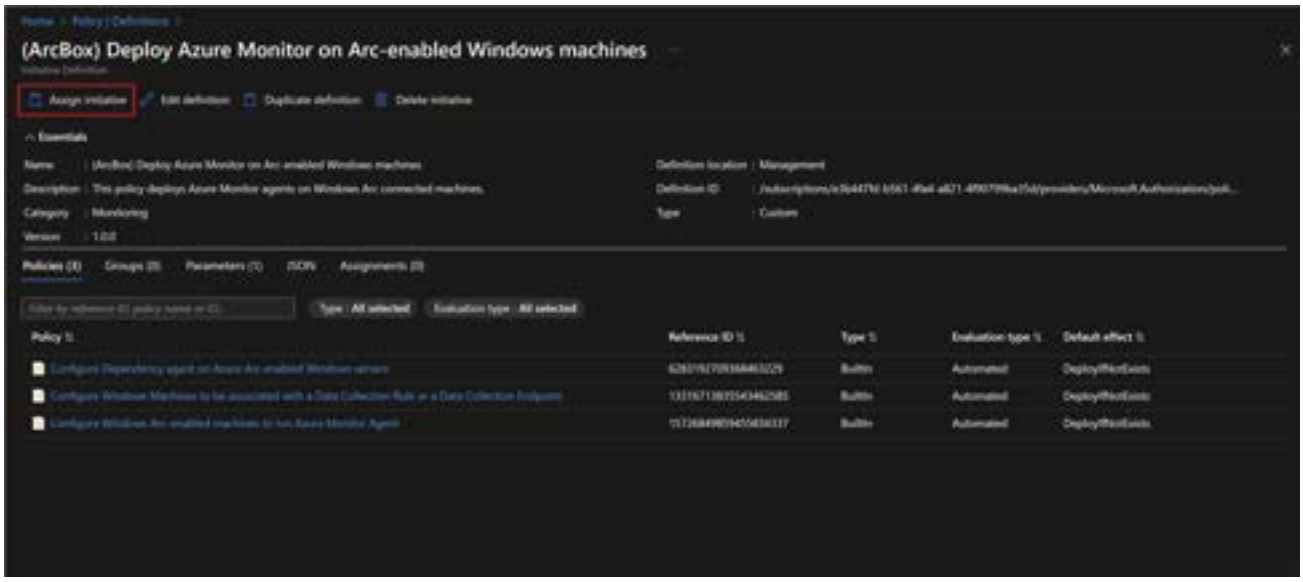
1. In the Azure portal, search for *Policy*.



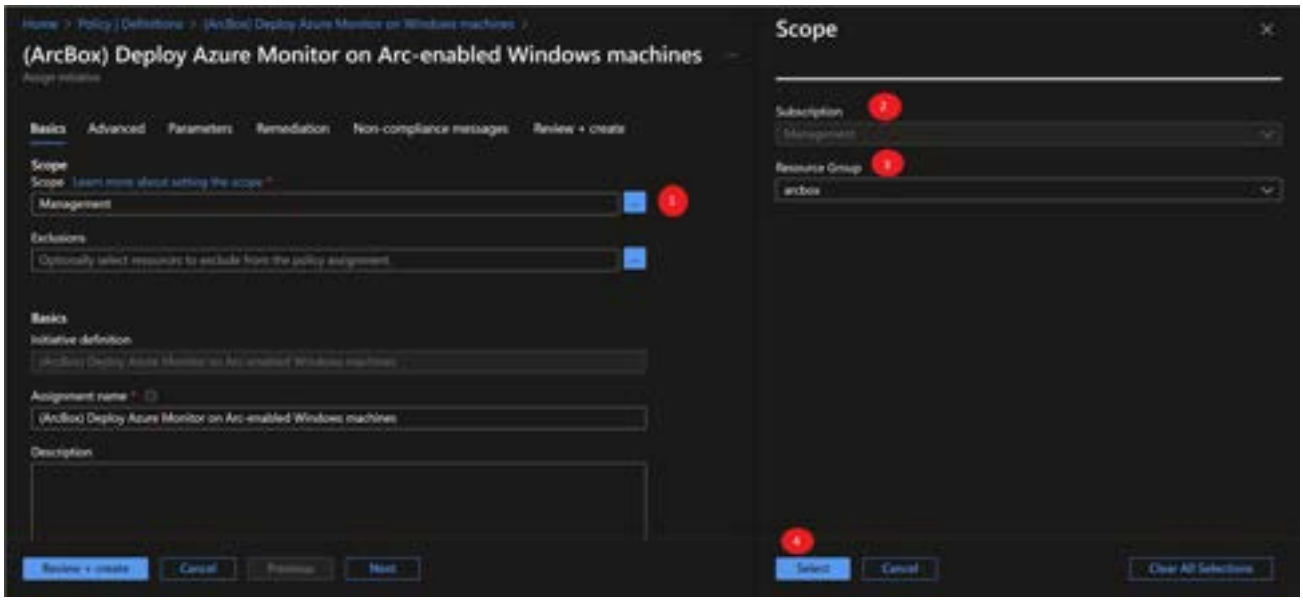
2. Click on "Definitions" and search for the *(ArcBox) Deploy Azure Monitor on Arc-enabled Windows machines* policy. This is a predefined *Initiative* containing a group of policies to deploy the Azure Monitor Agent.



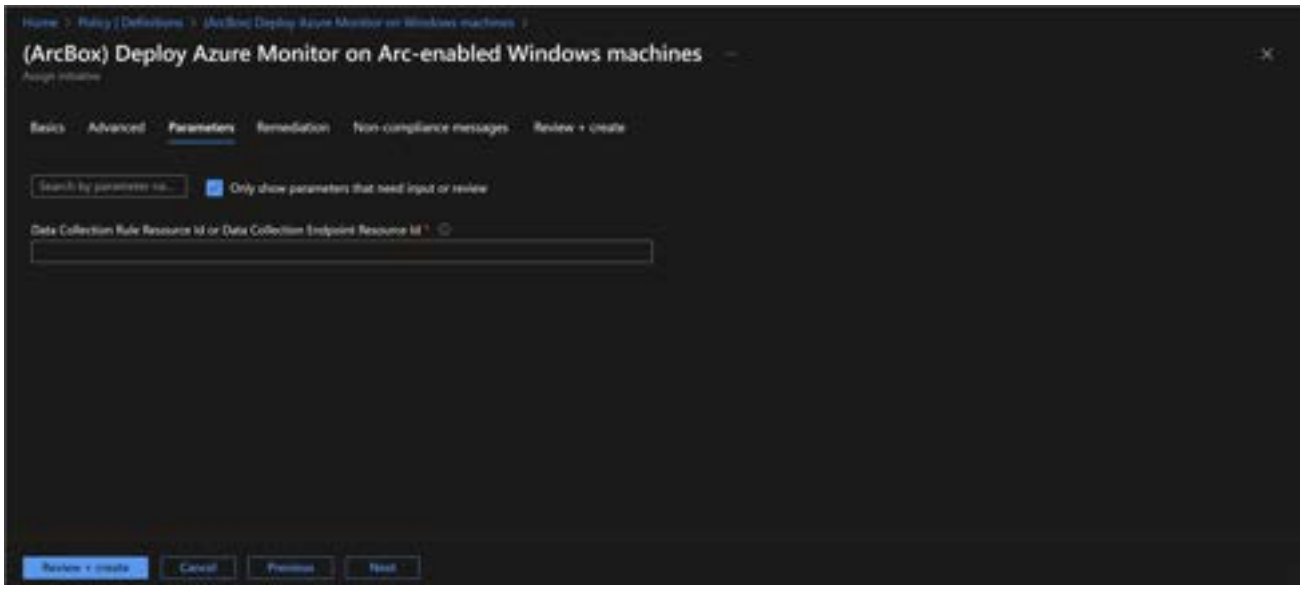
3. Click "Assign Initiative".



- [ ] 4. Select the right scope (management group, subscription and resource group) for the resource group where you deployed *ArcBox*.



- [ ] 5. After validating the scope, navigate to the parameters tab.

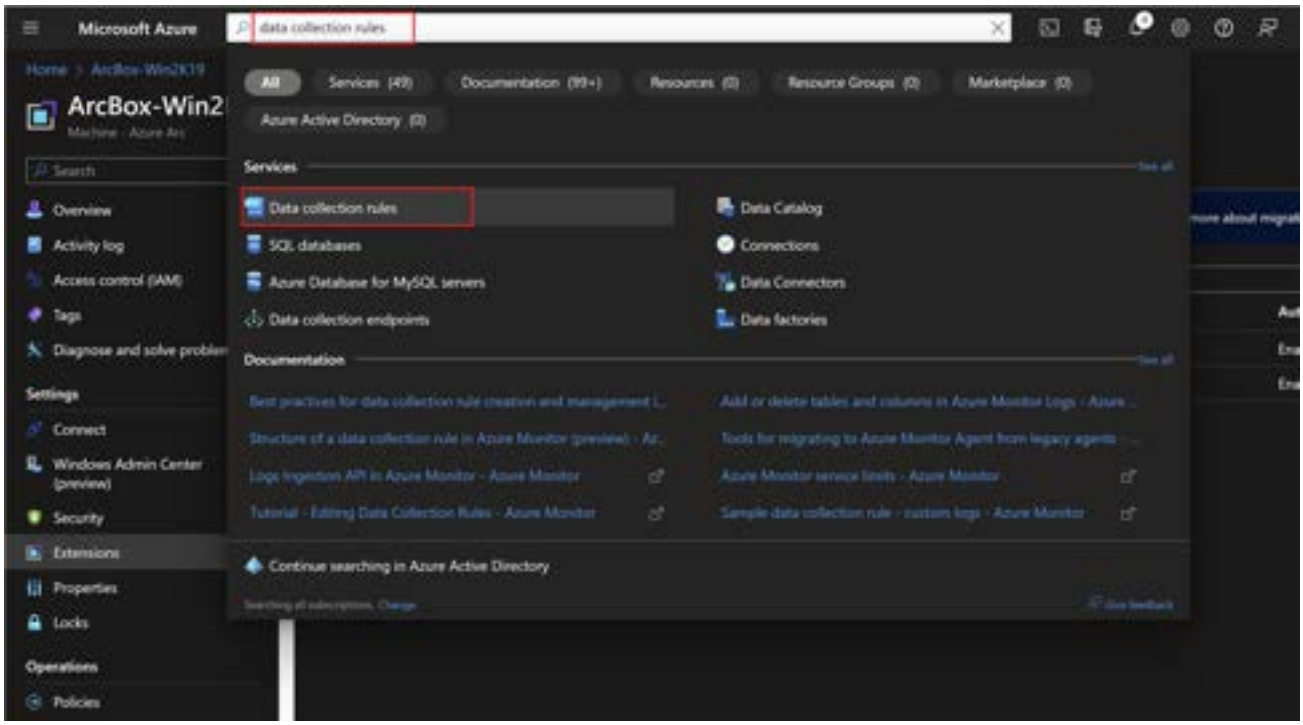


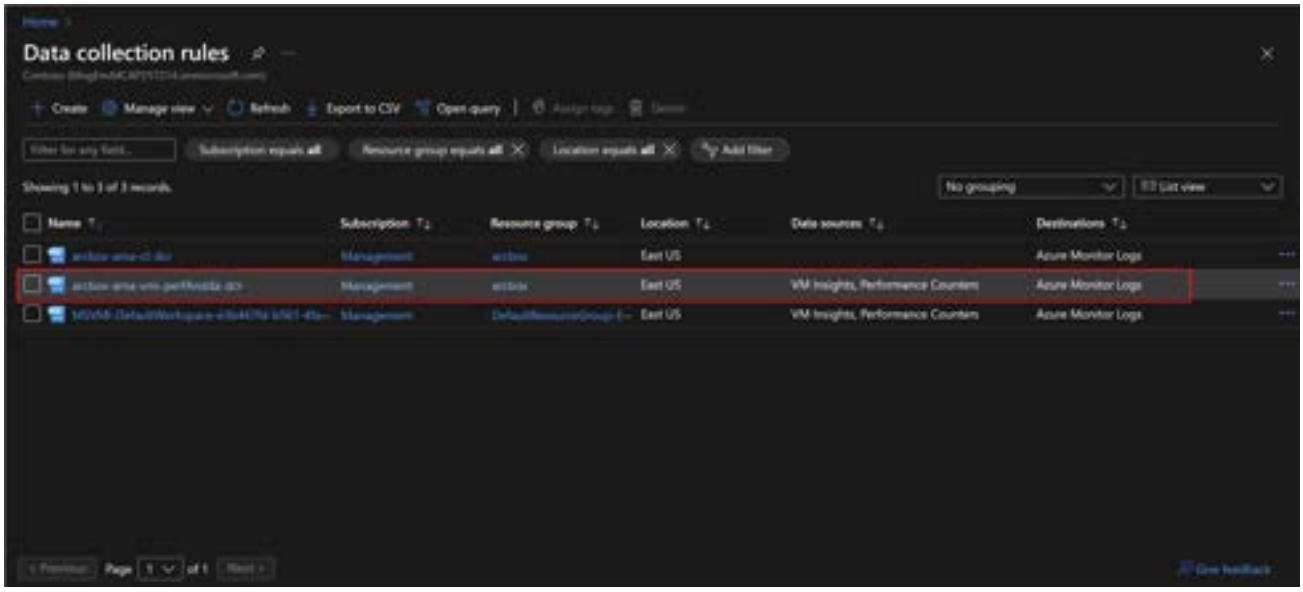
- [ ] 6. To get the "Data Collection Rule" resource Id, run the following command in PowerShell (making sure you enter the correct name of the resource group), copy the result into the field then click *create*.

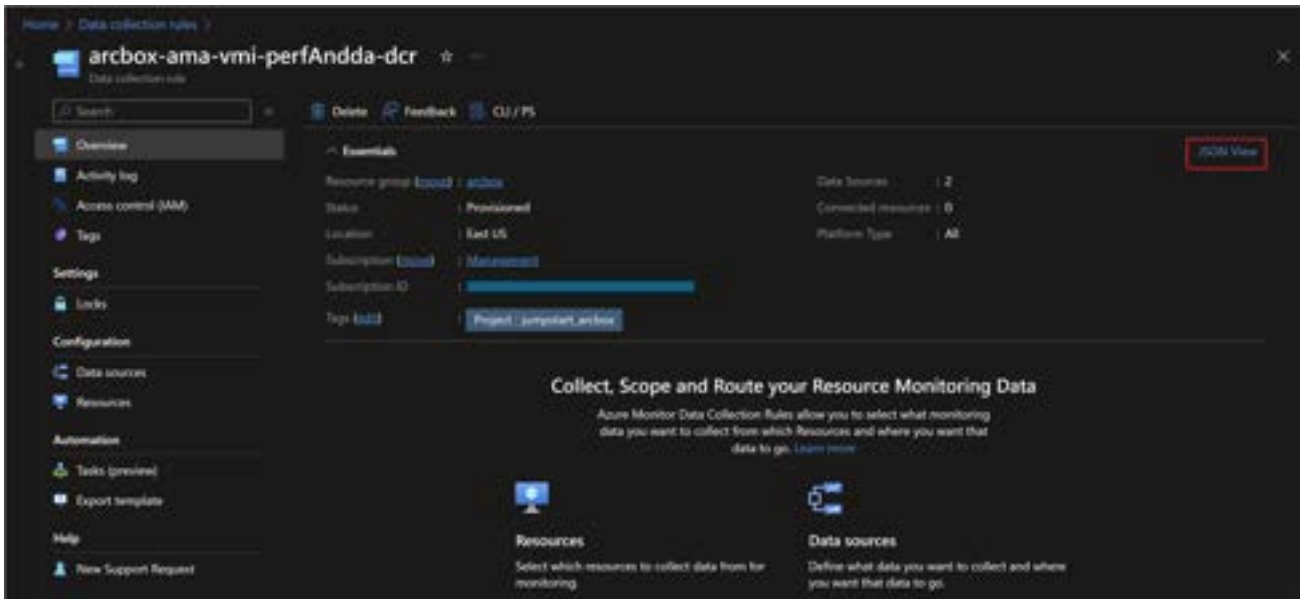
PowerShell

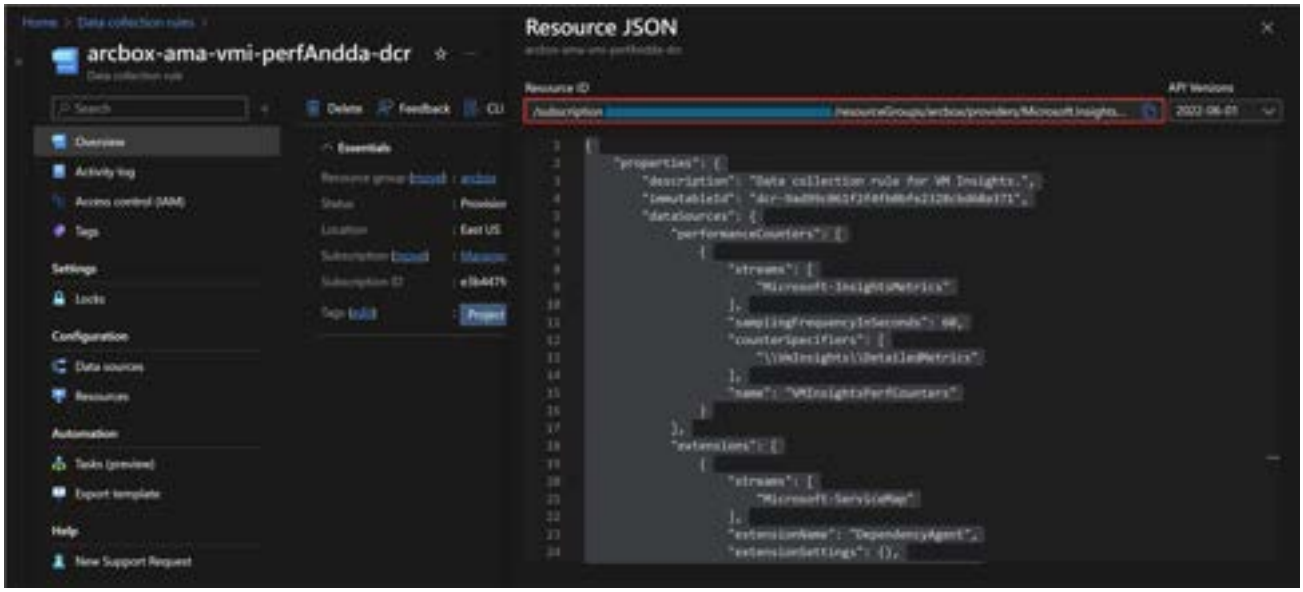
```
az resource show --name "arcbox-ama-vmi-perfAndda-dcr" `
                 --resource-group "<resource group name>" `
                 --resource-type Microsoft.Insights/dataCollectionRules `
                 --query id `
                 --output tsv
```

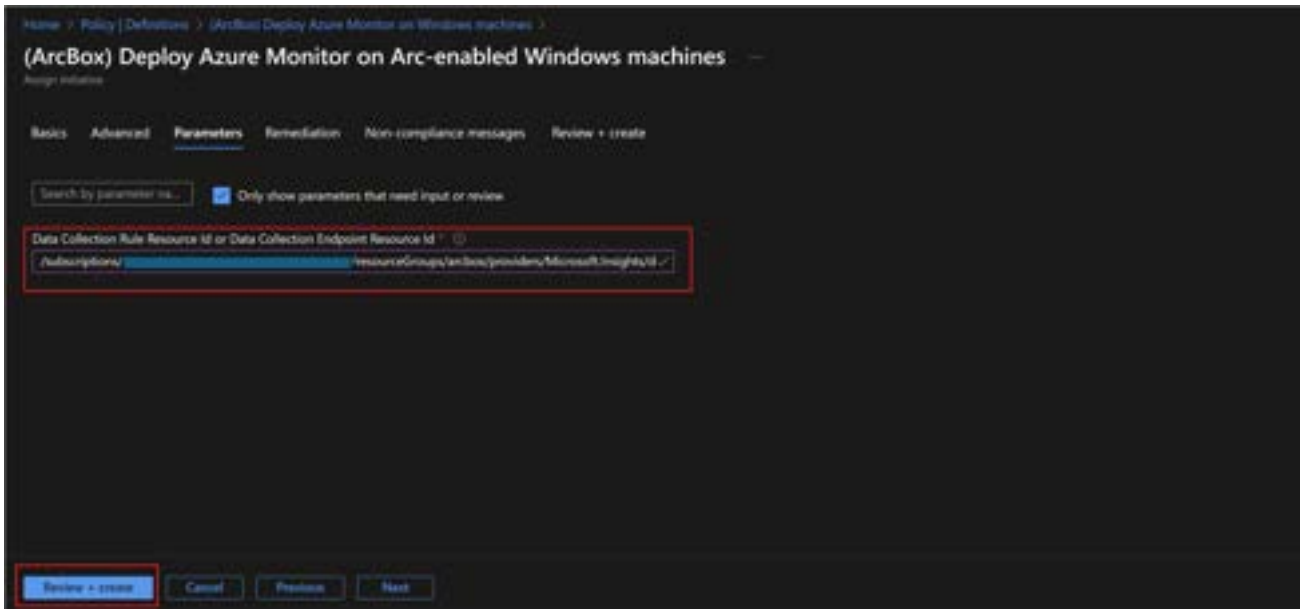
i Optionally you can also find the "Data Collection Rule" resource Id from the Azure portal. Search for the *arcbox-ama-vmi-perfAndda-dcr* data collection rule.


Microsoft Azure
data collection rules
All
Services (49)
Documentation (99+)
Resources (0)
Resource Groups (0)
Marketplace (0)
Azure Active Directory (0)
Services
Data collection rules
SQL databases
Azure Database for MySQL servers
Data collection endpoints
Data Catalog
Connections
Data Connectors
Data factories
Documentation
Overview
Activity log
Access control (IAM)
Tags
Settings
Connect
Windows Admin Center (preview)
Security
Extensions
Properties
Locks
Operations
Policies
Continue searching in Azure Active Directory


Data collection rules
Create
Manage view
Refresh
Export to CSV
Open query
Name
Subscription
Resource group
Location
Data sources
Destinations
East US
Azure Monitor Logs
VM Insights, Performance Counters
Azure Monitor Logs
East US
VM Insights, Performance Counters
Azure Monitor Logs

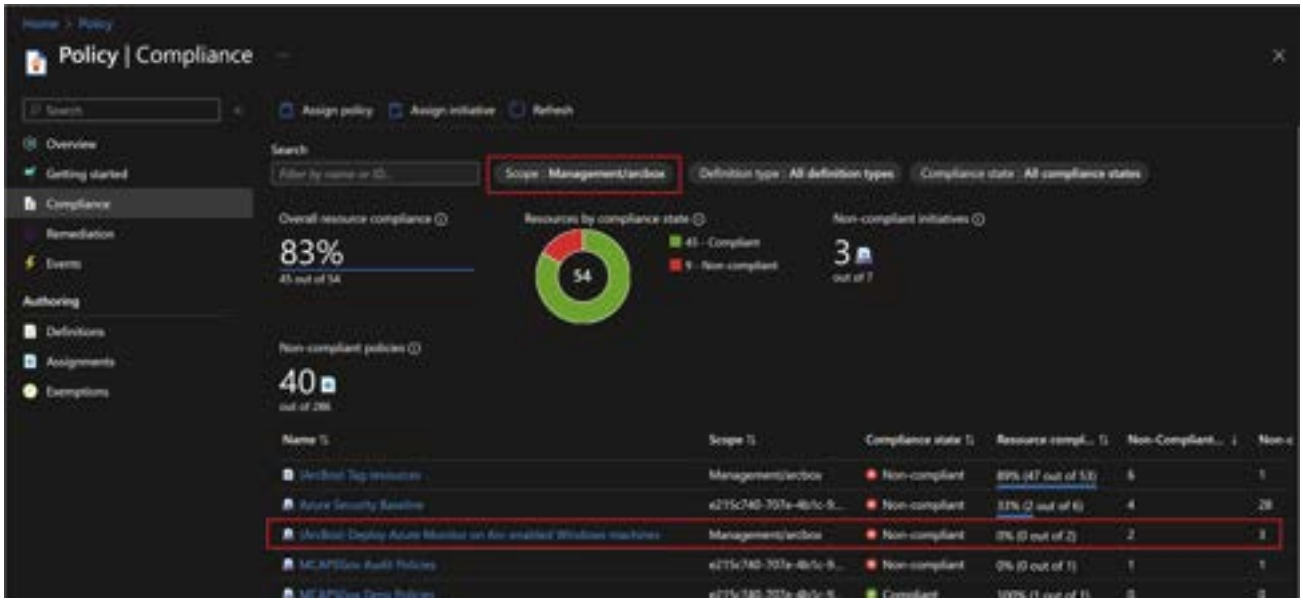

arcbox-ama-vmi-perfAndda-dcr
Delete
Feedback
CLI / PS
Overview
Activity log
Access control (IAM)
Tags
Settings
Locks
Configuration
Data sources
Resources
Automation
Tasks (preview)
Export template
Help
New Support Request
Essentials
JSON View
Collect, Scope and Route your Resource Monitoring Data
Azure Monitor Data Collection Rules allow you to select what monitoring data you want to collect from which Resources and where you want that data to go.
Resources
Select which resources to collect data from for monitoring
Data sources
Define what data you want to collect and where you want that data to go.

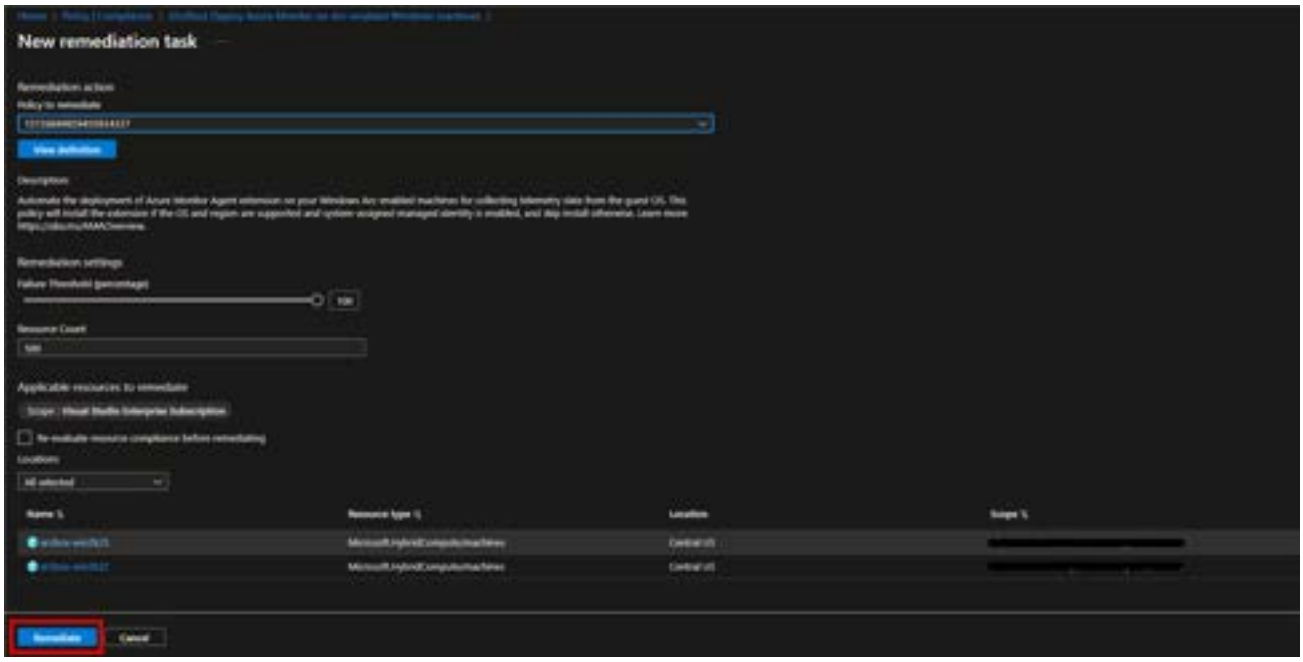

arcbox-ama-vmi-perfAndda-dcr
Resource JSON
Resource ID
API Version
2022-06-01
{
"properties": {
"description": "Data collection rule for VM Insights.",
"dataSources": {
"performanceCounters": [
{
"streams": [
"Microsoft-InsightsMetrics"
],
"samplingFrequencyInSeconds": 60,
"counterSpecifiers": [
"\\VmInsights\\DetailedMetrics"
],
"name": "VMInsightsPerfCounters"
}
],
"extensions": [
{
"streams": [
"Microsoft-ServiceMap"
],
"extensionName": "DependencyAgent",
"extensionSettings": {},


(ArcBox) Deploy Azure Monitor on Arc-enabled Windows machines
Basics
Advanced
Parameters
Remediation
Non-compliance messages
Review + create
Only show parameters that need input or review
Data Collection Rule Resource Id or Data Collection Endpoint Resource Id
Review + create
Cancel
Previous
Next

NOTE:The policy will take 5-15 minutes to assess the current resources.

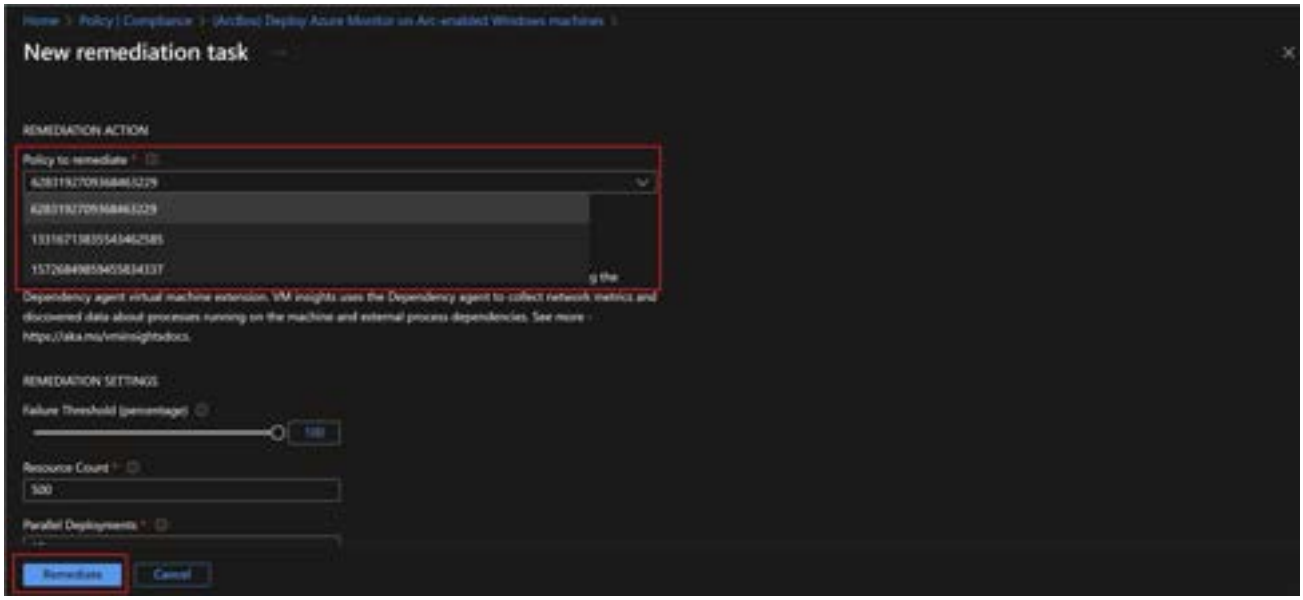
- [ ] 8. After the policy has reported compliance, create a remediation task to remediate existing machines.



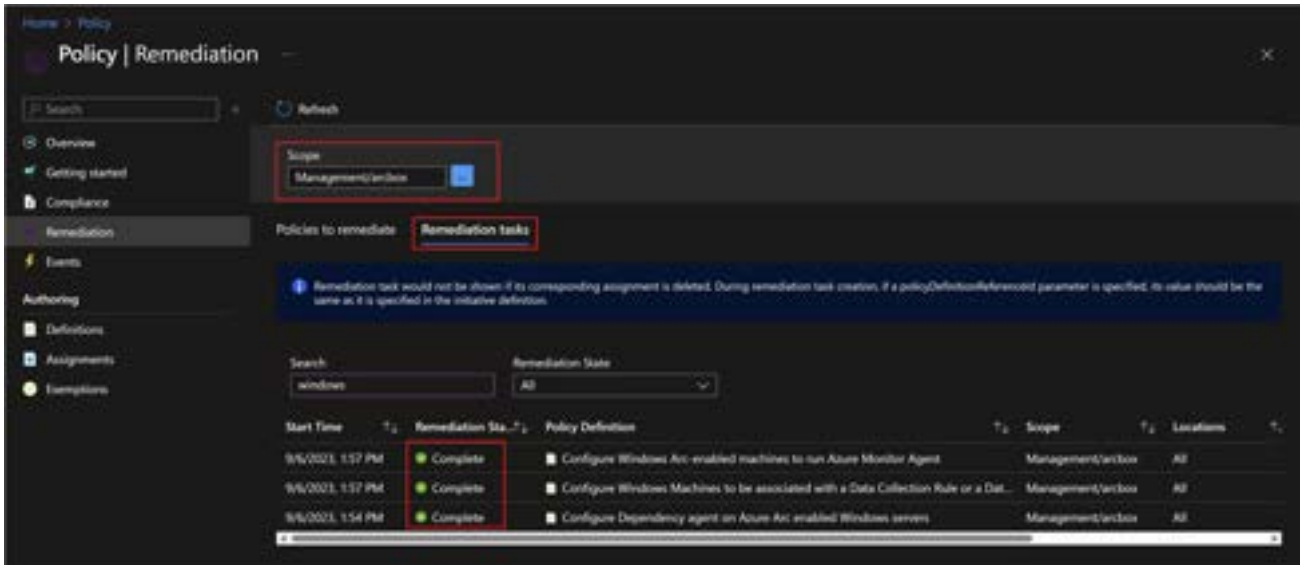


NOTE: When creating the remediation task, make sure to select the same region where you deployed ArcBox

☐ 9. Create one remediation task per policy definition in the initiative.

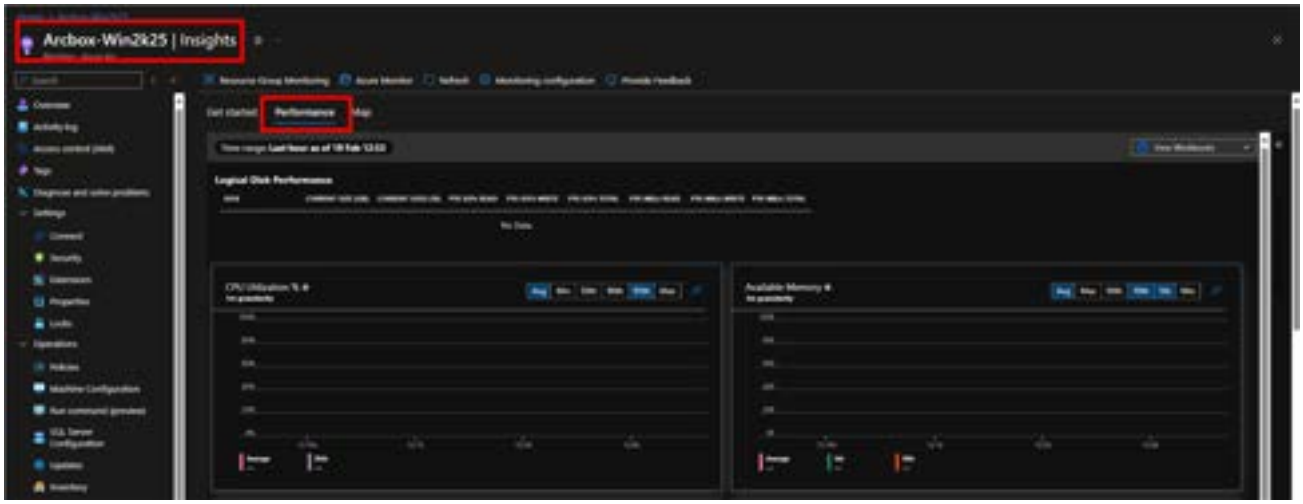


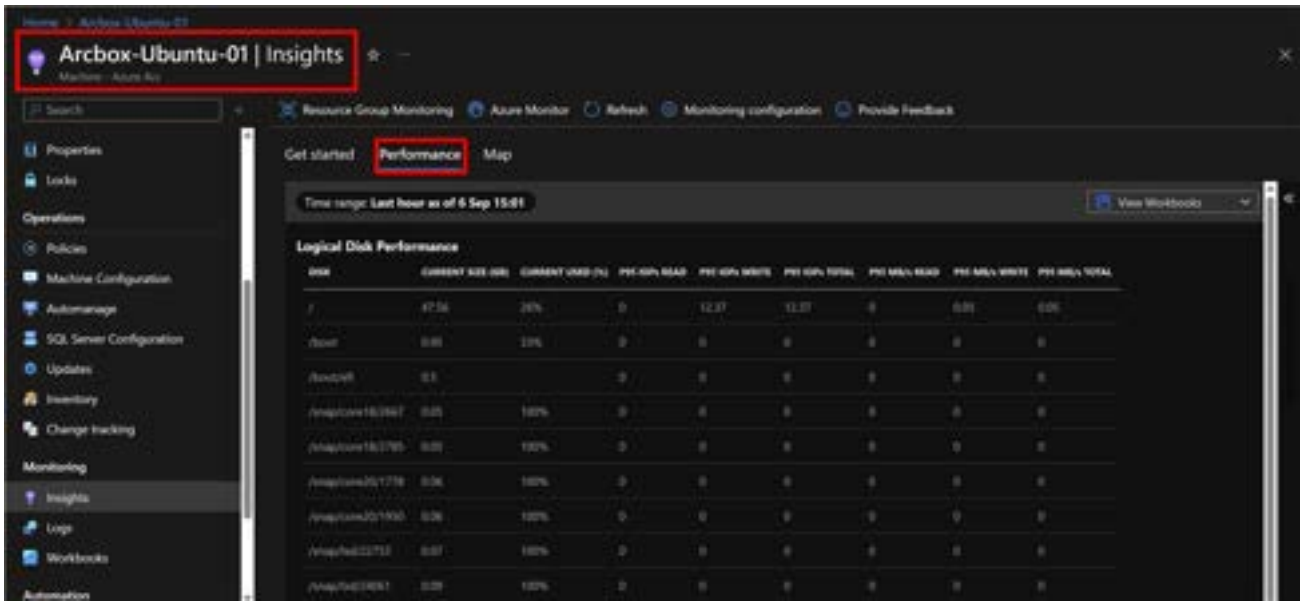
☐ 10. After all remediation tasks have completed. You should see the Azure Monitor agent extension and the dependency agent extension deployed to the Arc-enabled machines.



- [ ] 11. Repeat the same steps in *Task 2* to assign the Linux policy for data collection *(ArcBox) Deploy Azure Monitor on Arc-enabled Linux machines*.

- [ ] 12. After configuring the agents and VM insights using Azure Policy, it will take 10-25 minutes for the insights data to start showing up. **However, for the purposes of this workshop, the agent was pre-installed on the *ArcBox-Win2k25* and *Arcbox-Ubuntu-01* Arc-enabled machines so that you can see the expected results without having to wait until the remediation tasks have completed**. Head over to the these two machines to see the data collected by VM insights.





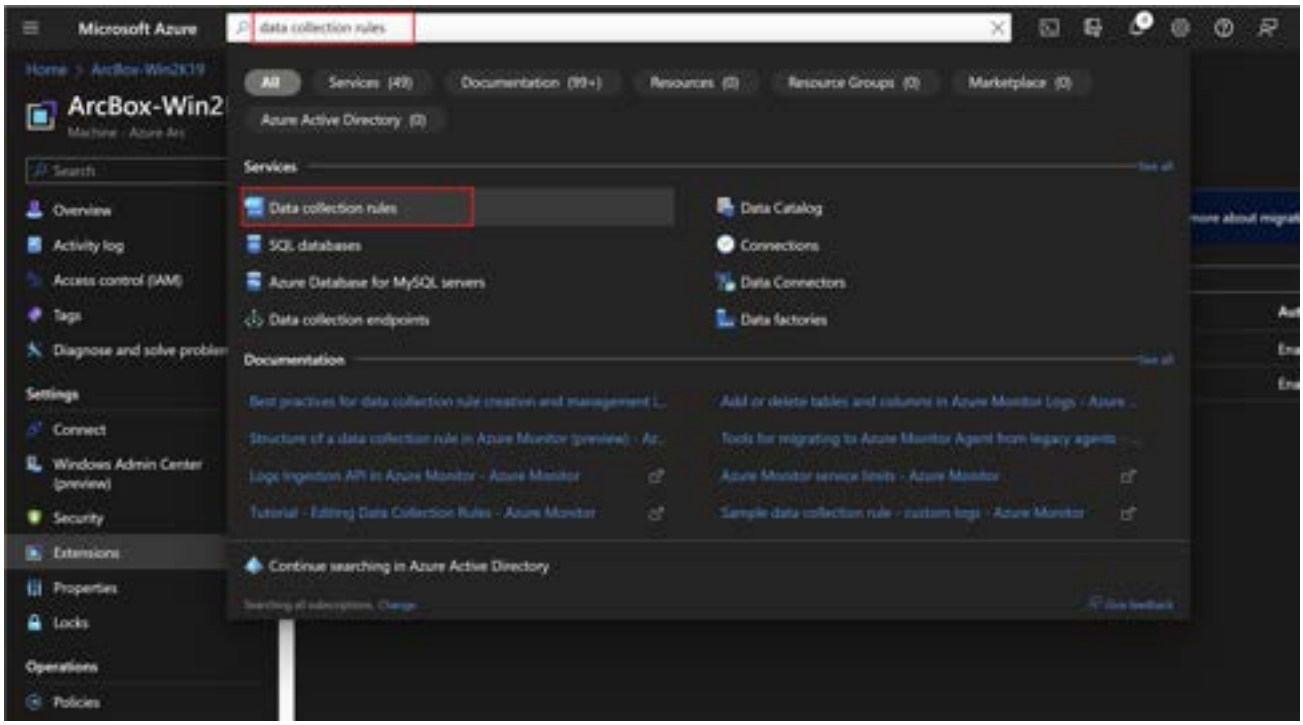
Task 1 has been completed

Click **Next** for the next task or **Go back to the main table of content**

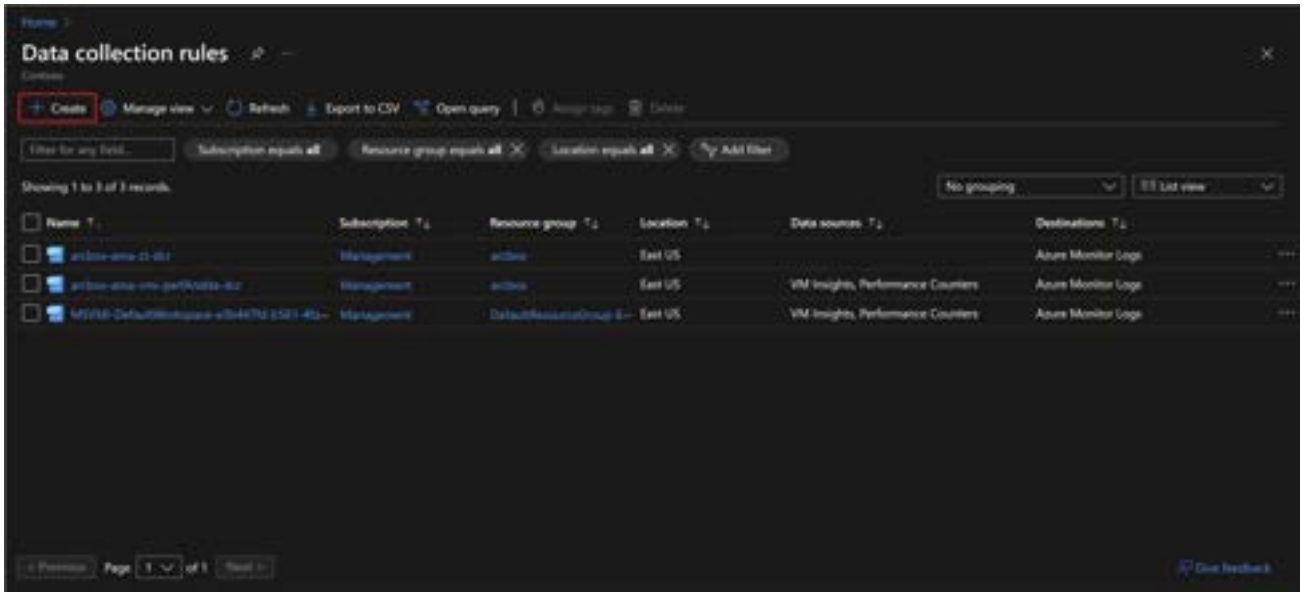
Task 2: Configure data collection for logs and metrics

As part of the ArcBox automation, some alerts and workbooks have been created to demonstrate the different monitoring operations you can perform after onboarding the Arc-enabled machines. You will now configure some data collection rules to start sending the needed metrics and logs to the Log Analytics workspace.

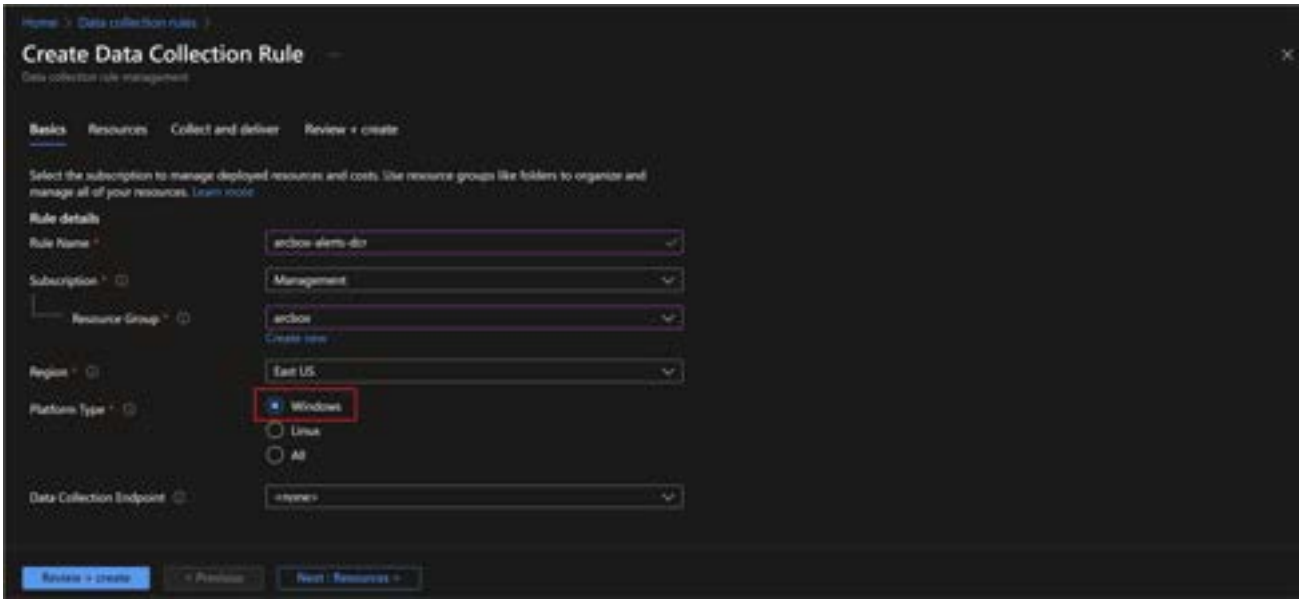
☐ 1. In the Azure portal, search for *Data Collection rules*.



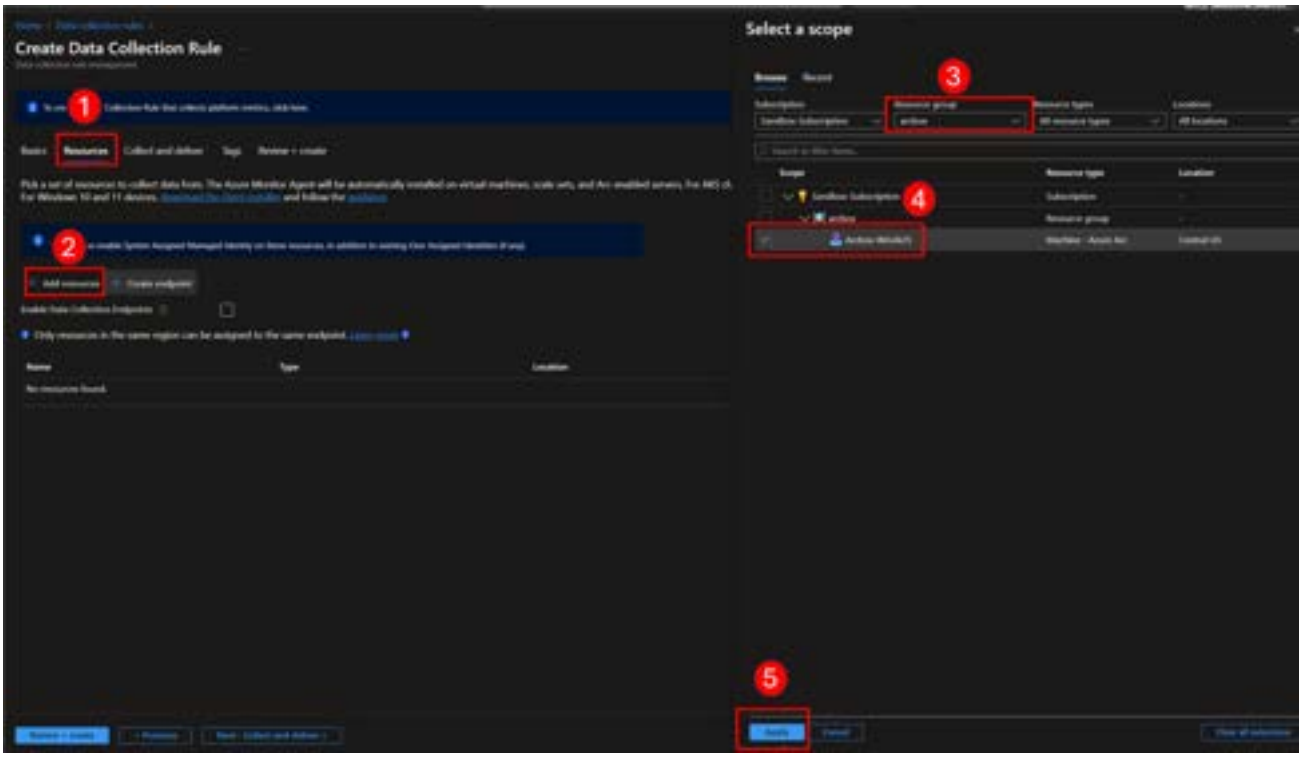
☐ 2. Create a new data collection rule.



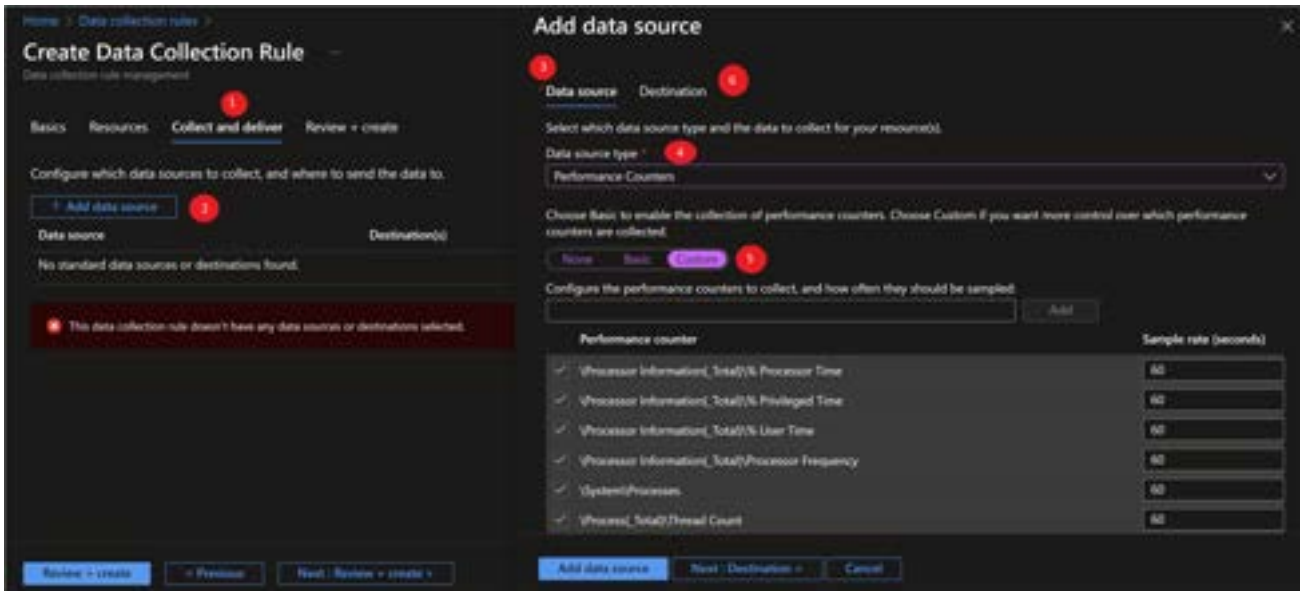
- [ ] 3. Provide a name and select the same resource group where ArcBox is deployed. Make sure to select Windows as the operating system.



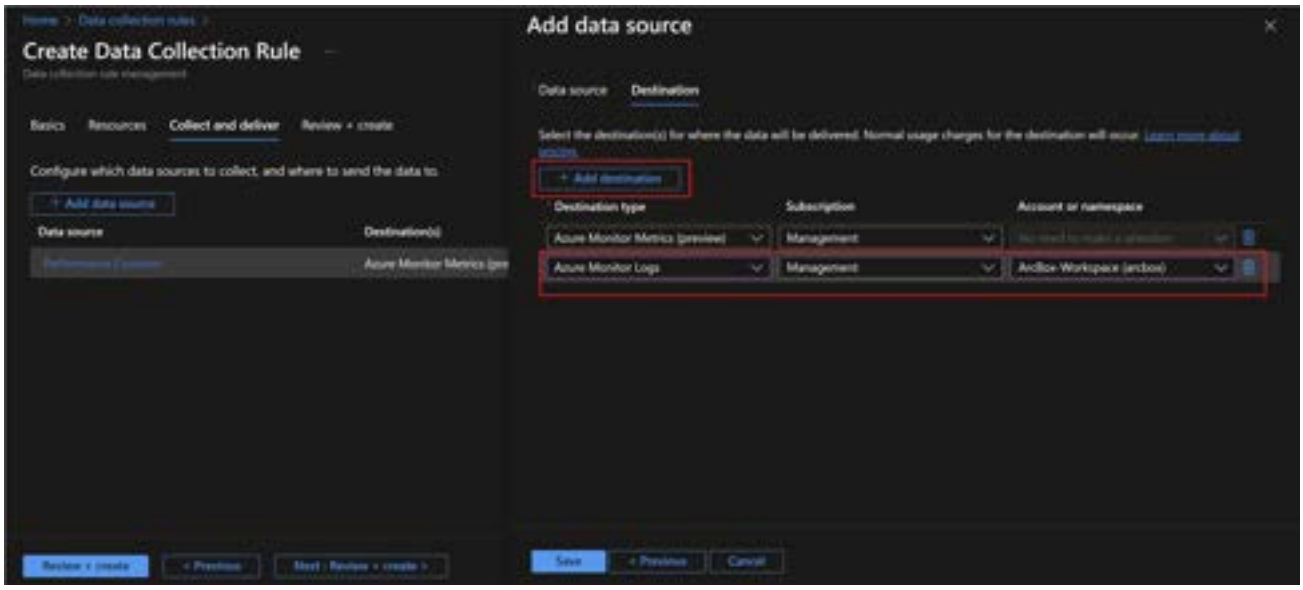
- [ ] 4. In the "Resources" tab, select the right resource group and the Arc-enabled servers onboarded.



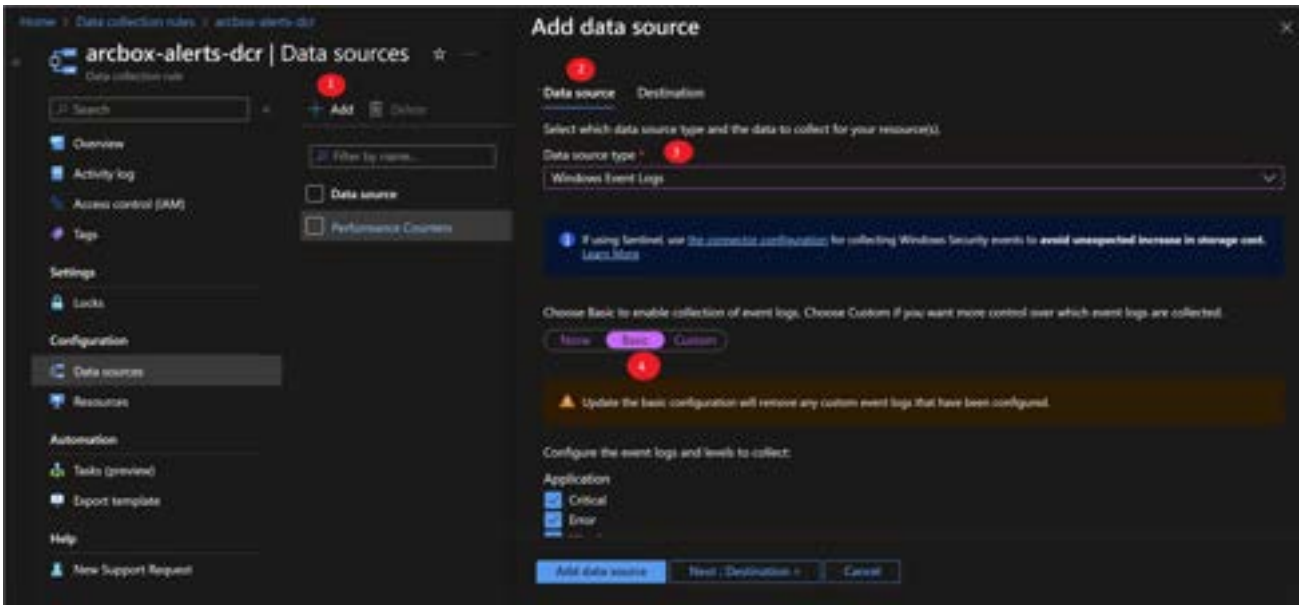
- [ ] 5. Add a new "Performance Counters" data source, and make sure to select all the custom counters.



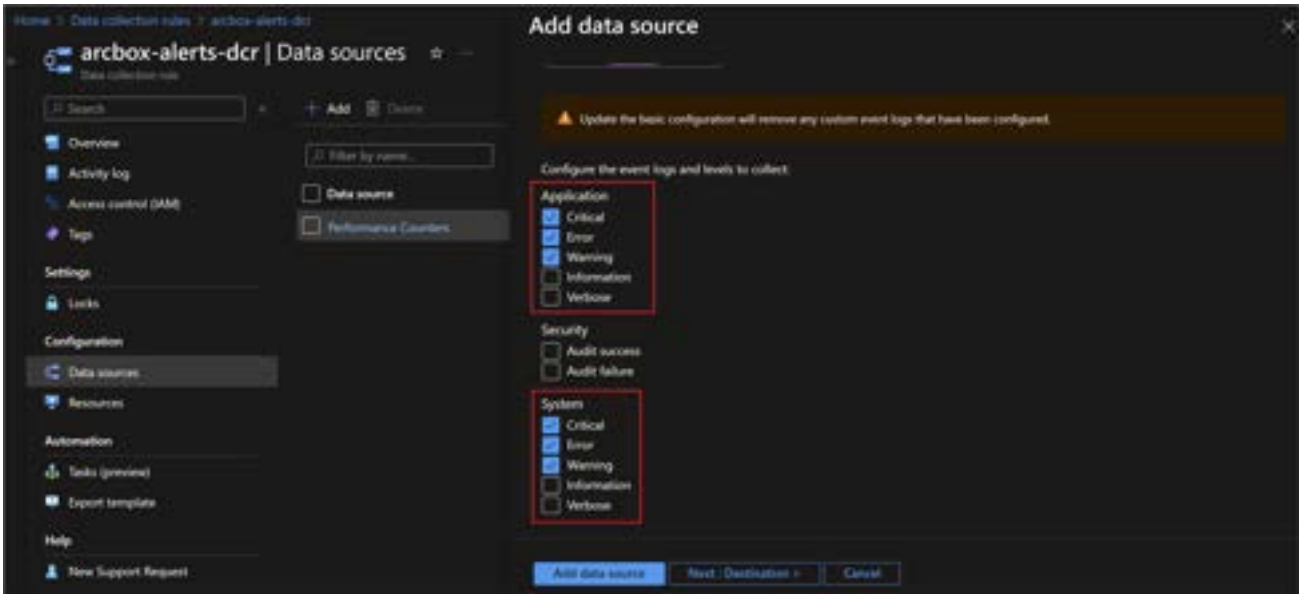
☐ 6. Add a new "Azure Monitor Logs" destination and select the log analytics workspace deployed in the ArcBox resource group and save.



☐ 7. Add a new "Windows Event logs" data source.

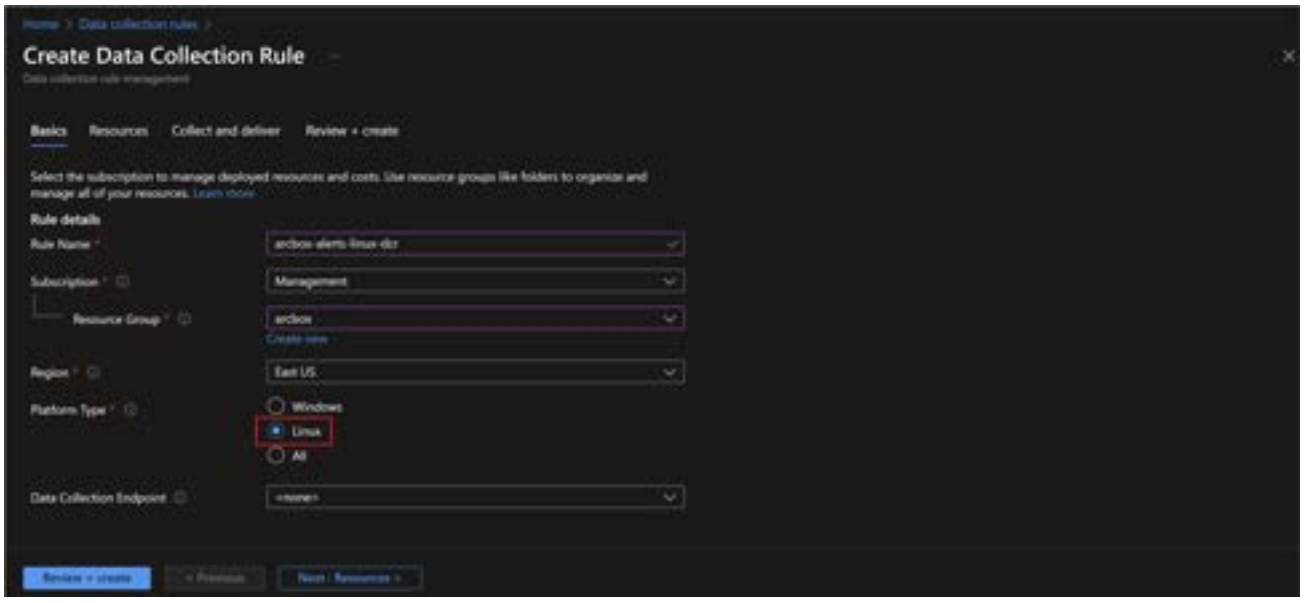


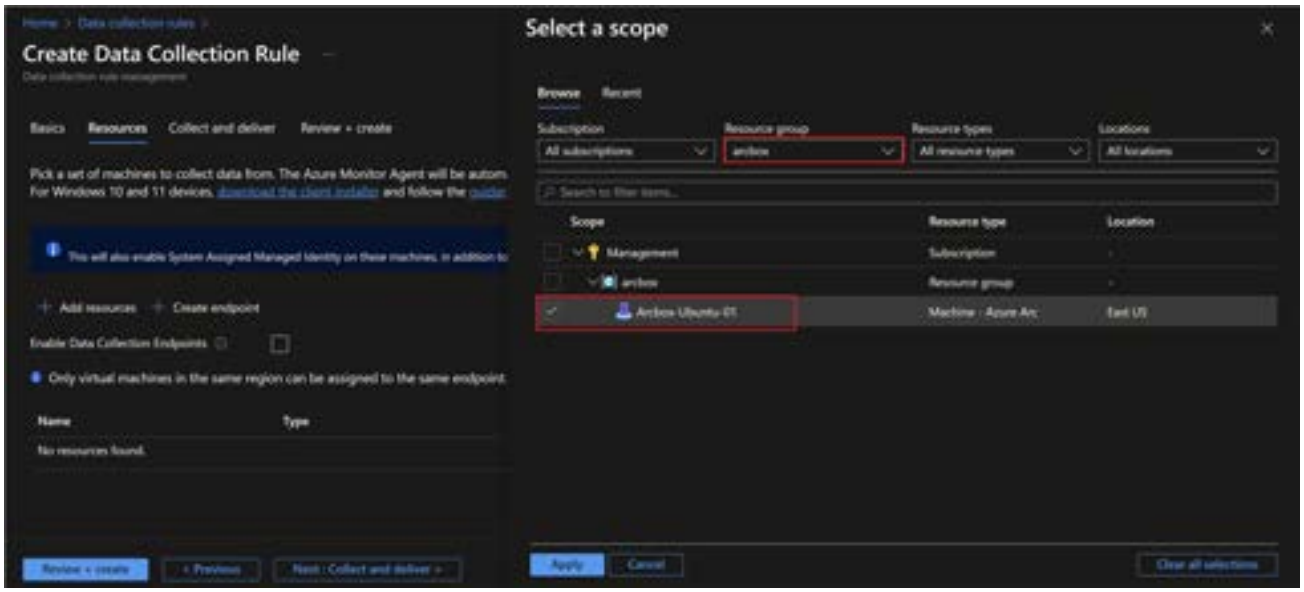
- [ ] 8. - Select *Critial*, *Error* and *Warning* events in the Application and System logs and add the data source.

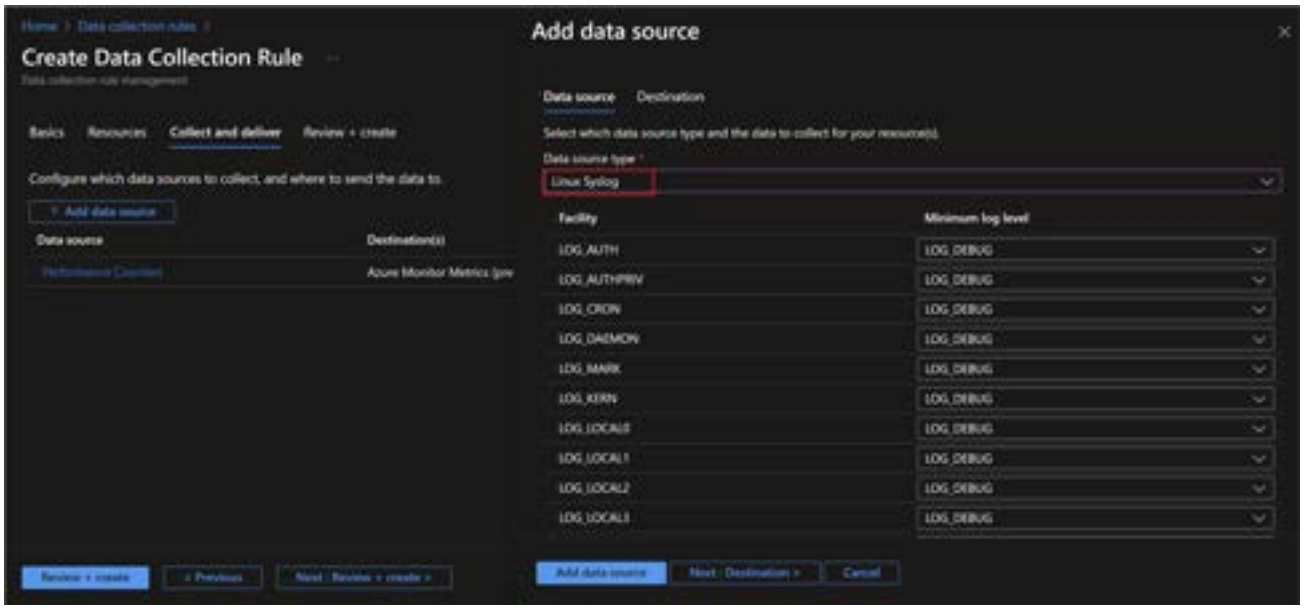


- [ ] 9. Save and create the data collection rule.

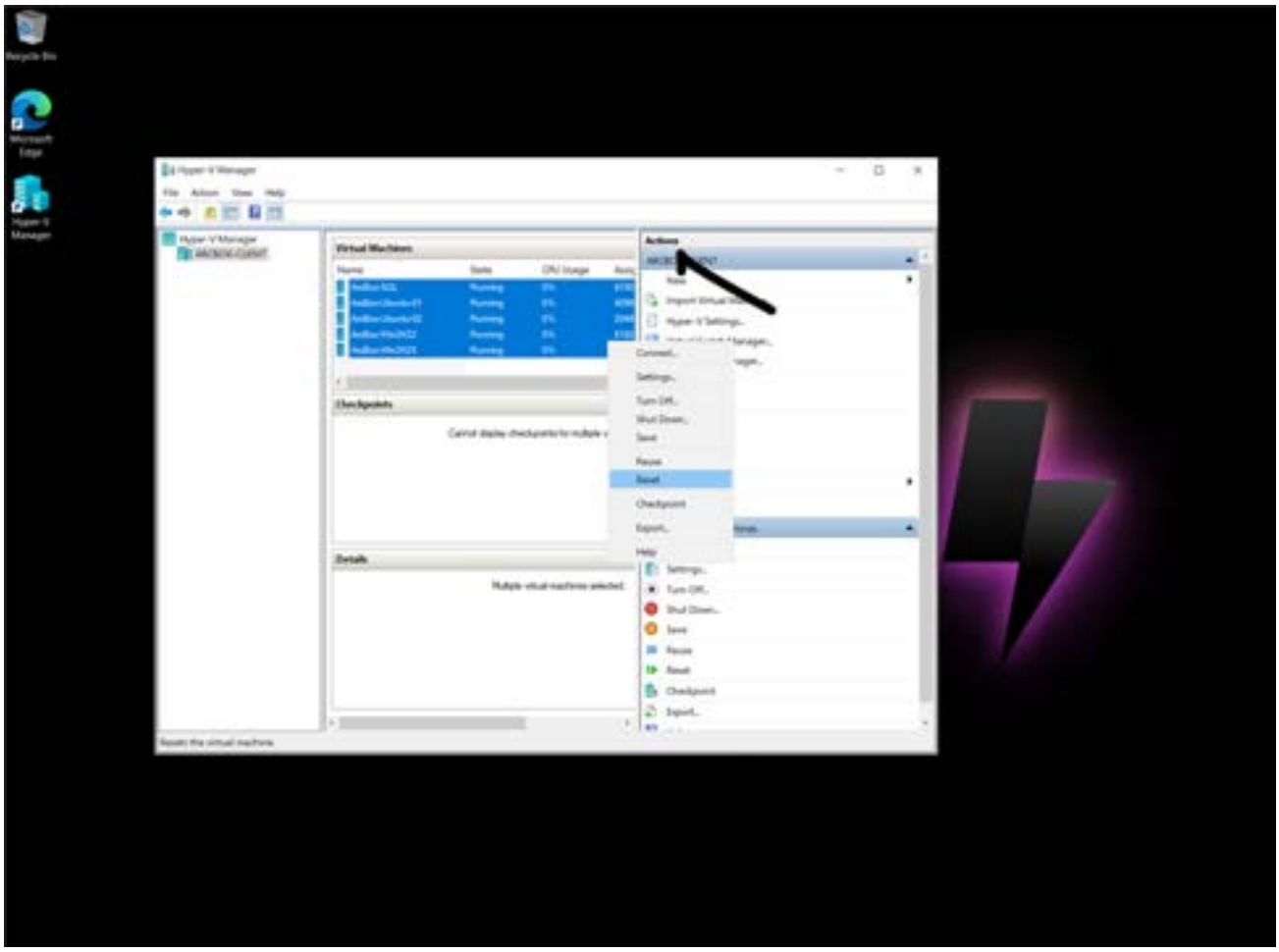
- [ ] 10. Repeat the previous steps to create another Linux data collection rule.


Create Data Collection Rule
Basics
Resources
Collect and deliver
Review + create
Rule details
Rule Name *
arcbox-alerts-linux-dcr
Subscription *
Management
Resource Group *
arcbox
Region *
East US
Platform Type *
Windows
Linux
All
Data Collection Endpoint
<none>


Create Data Collection Rule
Basics
Resources
Collect and deliver
Review + create
Add resources
Create endpoint
Enable Data Collection Endpoints
Only virtual machines in the same region can be assigned to the same endpoint.
Name
Type
No resources found.
Select a scope
Browse
Recent
Subscription
All subscriptions
Resource group
arcbox
Resource types
All resource types
Locations
All locations
Scope
Resource type
Location
Management
Subscription
arcbox
Resource group
Arcbox-Ubuntu-01
Machine - Azure Arc
East US
Apply
Cancel
Clear all selections


Create Data Collection Rule
Basics
Resources
Collect and deliver
Review + create
Configure which data sources to collect, and where to send the data to.
Add data source
Data source
Destination(s)
Add data source
Data source
Destination
Select which data source type and the data to collect for your resource(s).
Data source type *
Linux Syslog
Facility
Minimum log level
LOG_AUTH
LOG_AUTHPRIV
LOG_CRON
LOG_DAEMON
LOG_MARK
LOG_KERN
LOG_LOCAL0
LOG_LOCAL1
LOG_LOCAL2
LOG_LOCAL3
LOG_DEBUG
Add data source
Next : Destination >
Cancel

11. After waiting for 5-10 minutes for the data collection rule to start collecting data, restart the servers in the Hyper-V manager on the *ArcBox-Client* VM to trigger some new events.



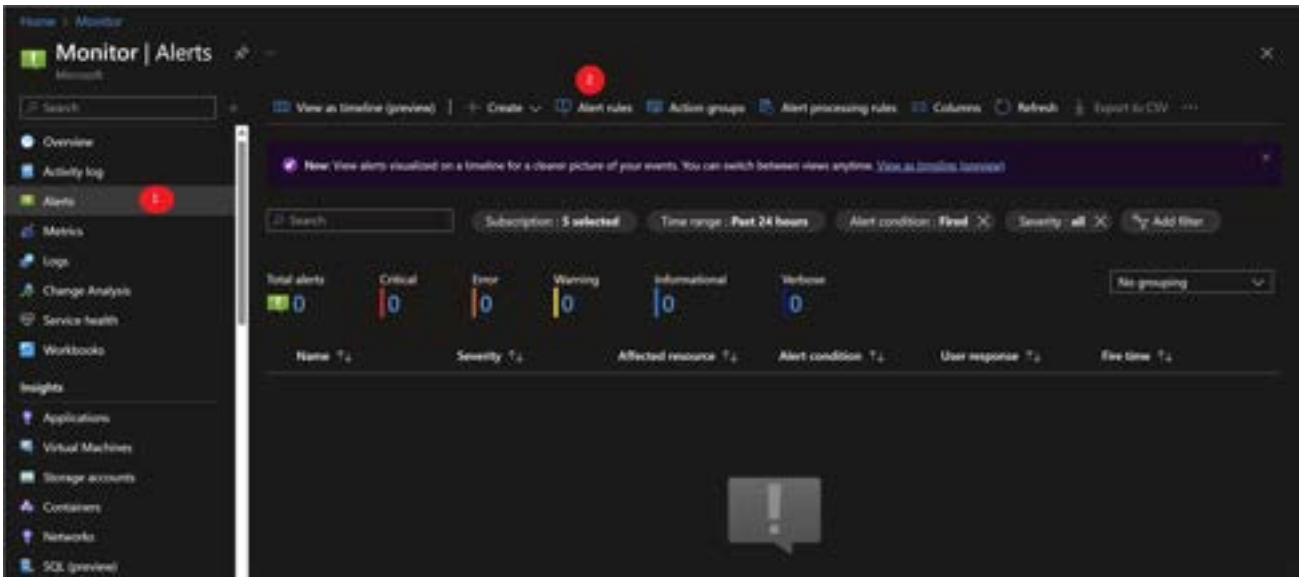
Task 2 has been completed

Click **Next** for the next task or **Go back to the main table of content**

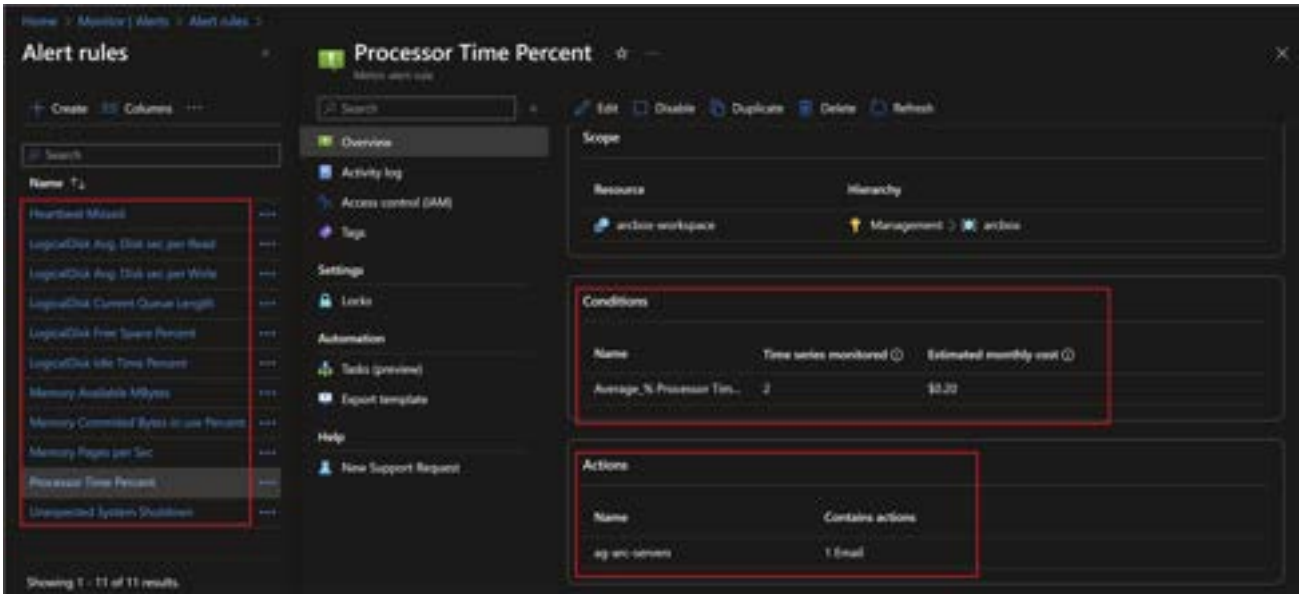
Task 3: View alerts and visualizations

NOTE: It might take some time for all visualizations to load properly

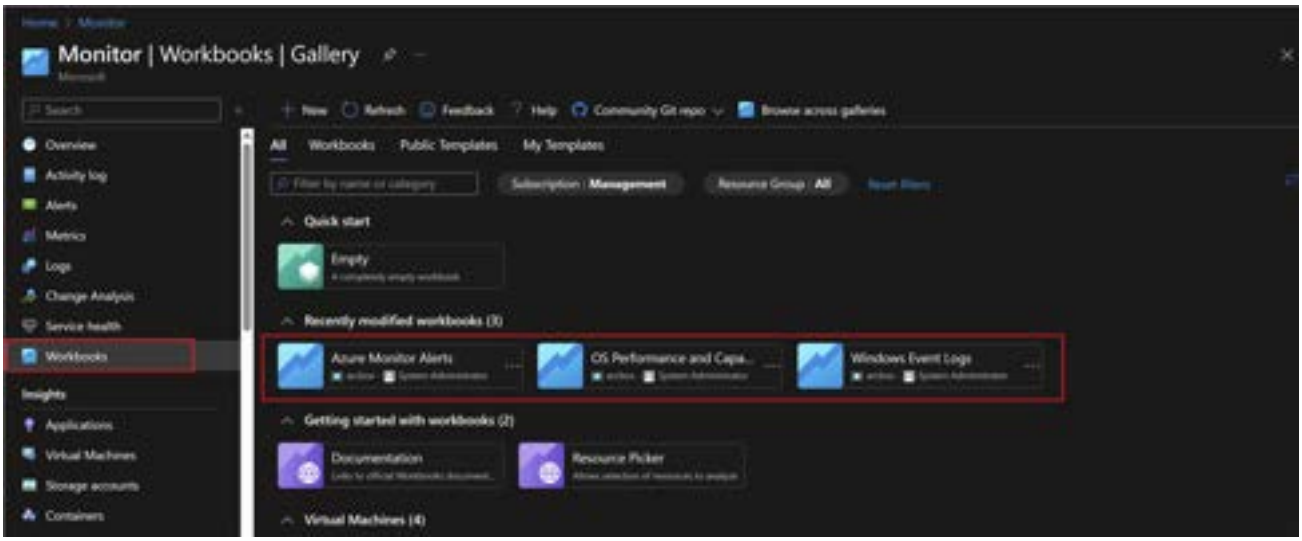
☐ 1. In Azure Monitor, click on *Alerts*. and select *Alert rules*

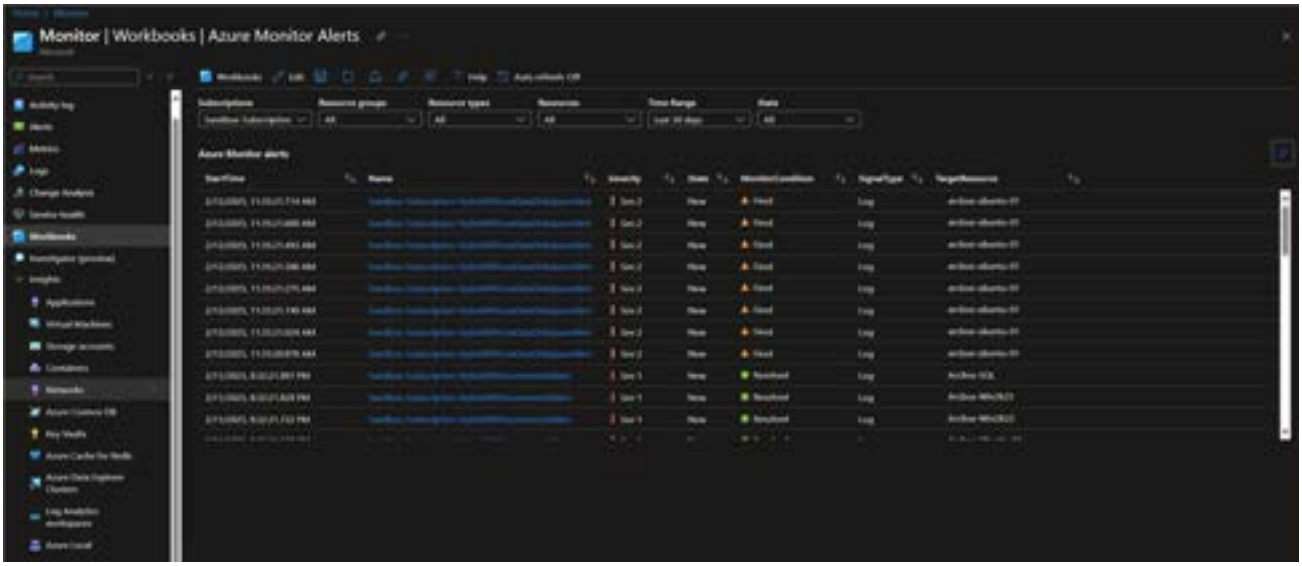


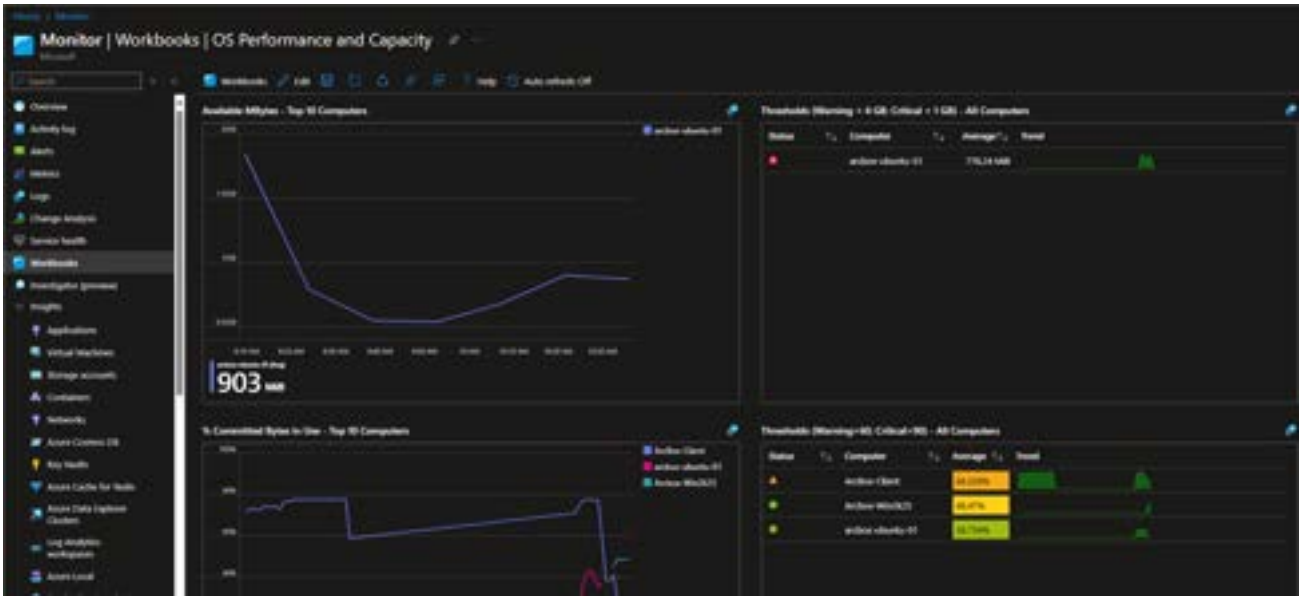
☐ 2. Explore the alert rules crated for you.

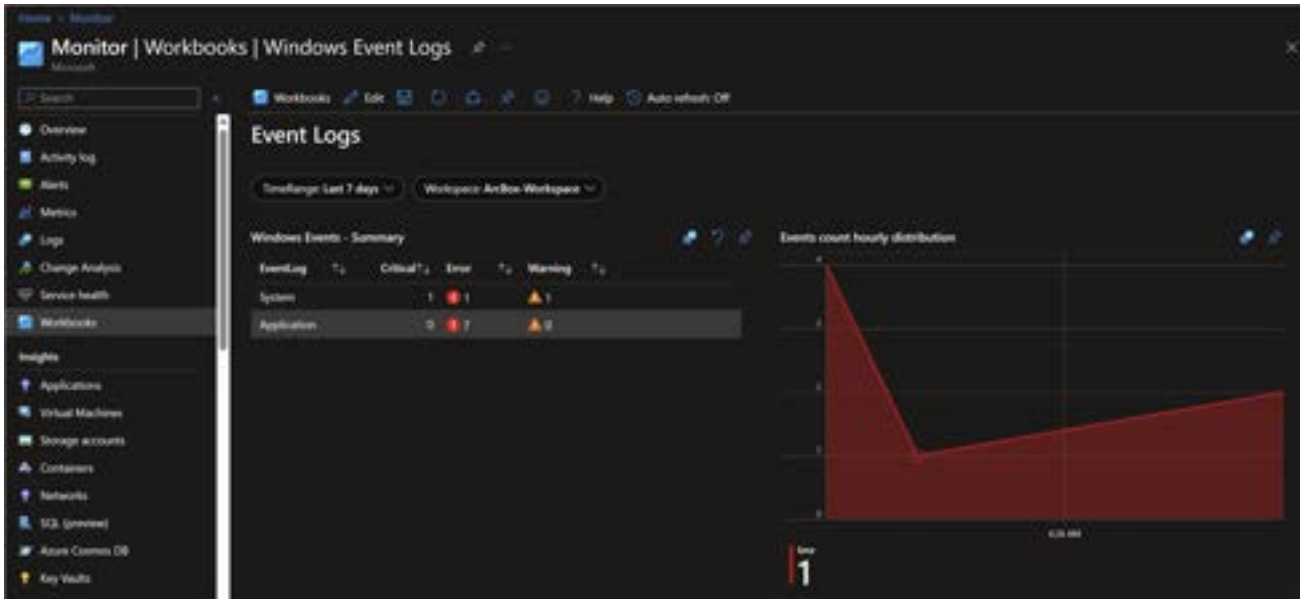


☐ 3. Go back to Azure Monitor and click on *Workbooks*. There are three workbooks deployed for you.


Monitor | Workbooks | Gallery
Overview
Activity log
Alerts
Metrics
Logs
Change Analysis
Service health
Workbooks
Insights
Applications
Virtual Machines
Storage accounts
Containers
New
Refresh
Feedback
Help
Community Git repo
Browse across galleries
All
Workbooks
Public Templates
My Templates
Quick start
Empty
Recently modified workbooks (3)
Azure Monitor Alerts
OS Performance and Capa...
Windows Event Logs
Getting started with workbooks (2)
Documentation
Resource Picker
Virtual Machines (4)


Monitor | Workbooks | Azure Monitor Alerts


Monitor | Workbooks | OS Performance and Capacity
903



Task 3 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 2: Monitor changes to your Azure Arc-enabled servers using Change Tracking and Inventory

Objective

Tracks changes in Your Azure Arc-enabled machines to help you pinpoint operational and environmental issues.

Estimated Time to Complete This Lab

30 minutes

Explanation

Change Tracking and Inventory is a built-in Azure service, provided by Azure Automation. The new version uses the Azure Monitor Agent AMA as opposed to the Log Analytics Agent. You will be using the new version in this exercise.

Task 1: Prerequisites

The following are required for this task:

- [ ] 1. Ensure that the servers are already on-boarded to Azure Arc.
- [ ] 2. Ensure that the Azure Monitor agent (AMA) is already deployed on every Arc-enabled server (This was done in Exercise 1 of this lab).
- [ ] 3. Note the Current Limitations as listed in https://learn.microsoft.com/azure/automation/change-tracking/overview-monitoring-agent?tabs=win-az-vm#current-limitations.

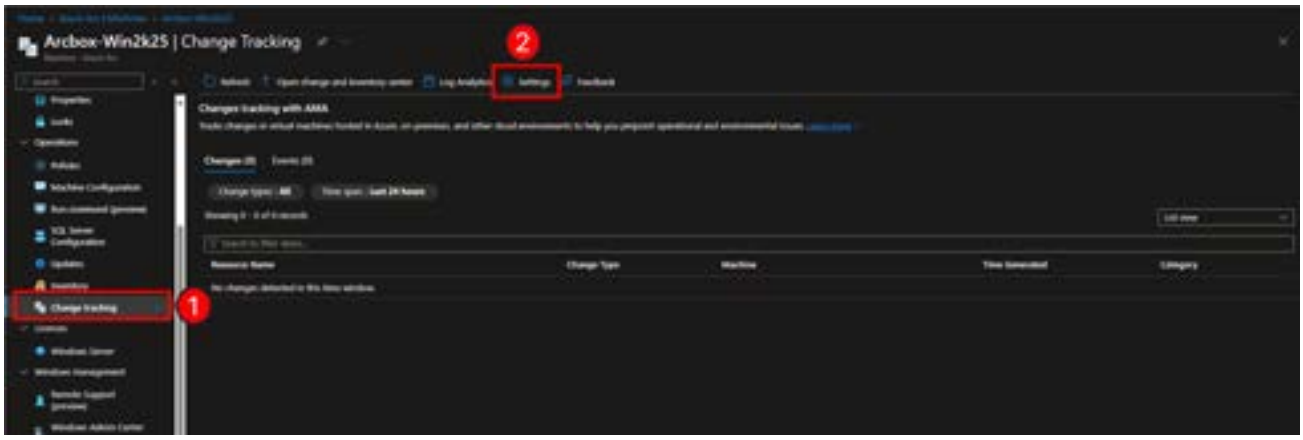
Task 1 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Task 2: Enable Change Tracking and Inventory

1. To enable these features you would need to set up a Data Collection Rule that would collect the right events and data for Change Tracking and Inventory and create an Azure policy to onboard your Arc-enabled machines to Change Tracking. **For the purposes of this workshop** - these tasks have all been done for you, so you do not need to do them manually. Follow the link here to learn how to do these yourself in future.

2. Verify that Change Tracking and Inventory is now enabled on the *ArcBox-Win2K25* Arc enabled server in the Azure Portal.



Task 2 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Task 3: Track changes in Windows services

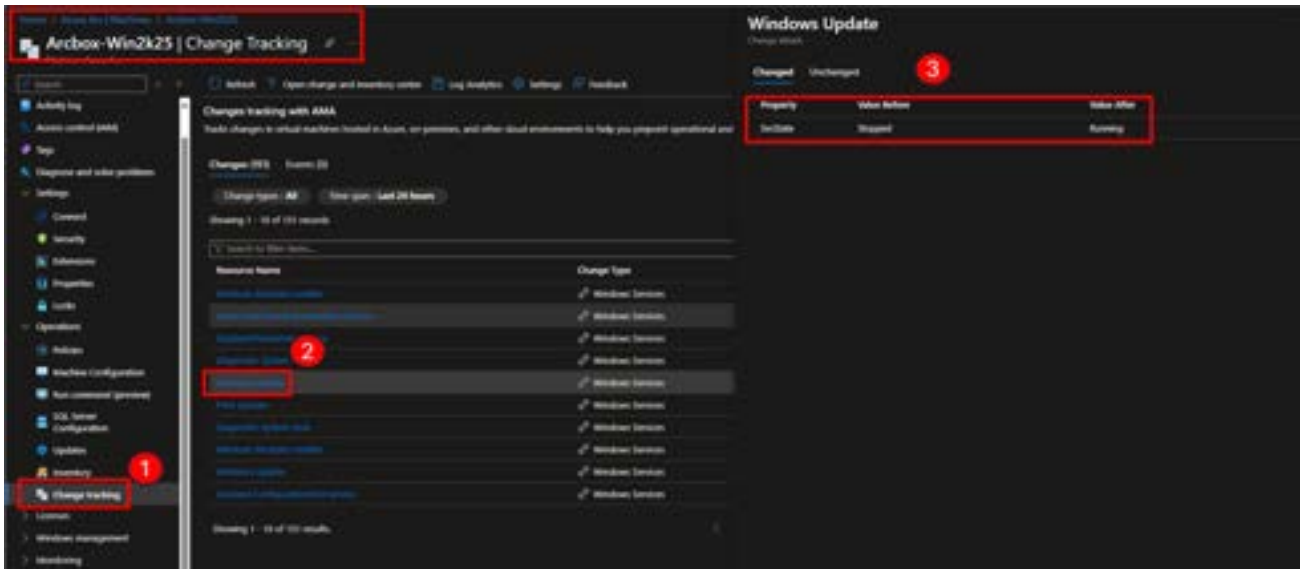
1. From the "Change tracking" settings select "Windows Services" and change the "Collection Frequency" to 10 minutes.

2. Go to the ArcBox-Client machine via RDP and from Hyper-V manager right-click on *ArcBox-Win2K25* VM then click "Connect" (Administrator default password is JS123!!). Try stopping the "Print Spooler" and the "Windows Update" services using an Administrator PowerShell session (or from the Services desktop application).

```PowerShell
Stop-Service spooler
Stop-Service wuauserv
```

3. The service changes will eventually show up in the "Change tracking" page for the Arc-enabled machine. (By default Windows services status are updated every 30 minutes but you changed that to 10 minutes earlier to speed up the result for this task).

It may take 30 minutes or more for the changes to show up in the portal, you can move to the next task/exercise or lab and come back later to see the results if you prefer. This way, you can make better use of your time in this workshop.



- [ ] 4. You can restart the stopped services on the server if you wish and change tracking will show the outcome in the portal after some time.

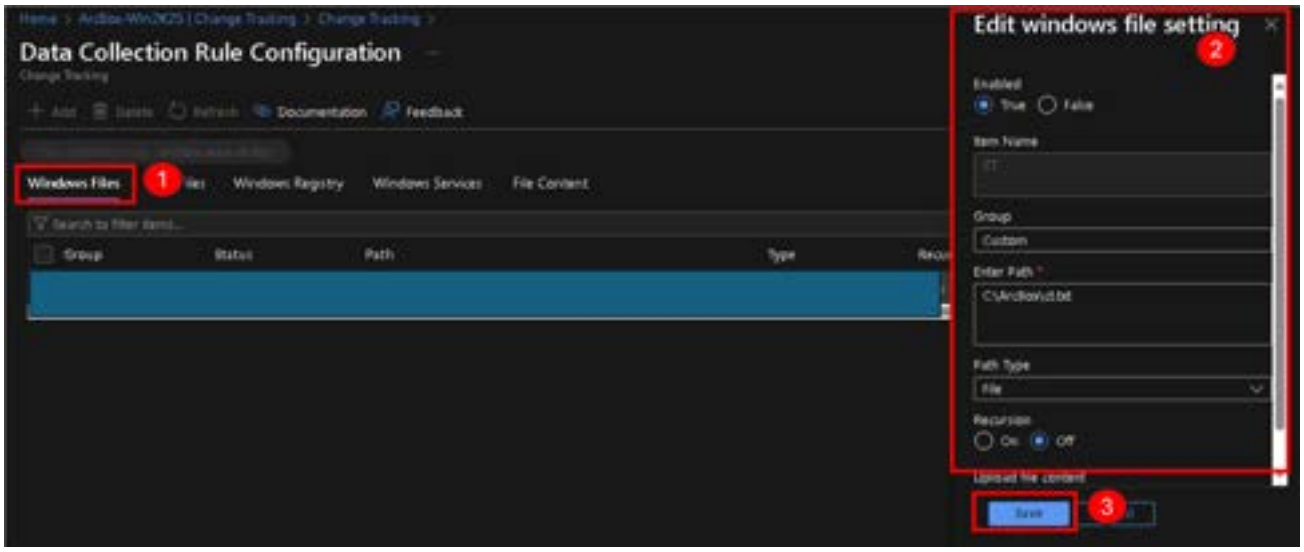
```PowerShell
Start-Service spooler
Start-Service wuauserv
```

Task 3 has been completed

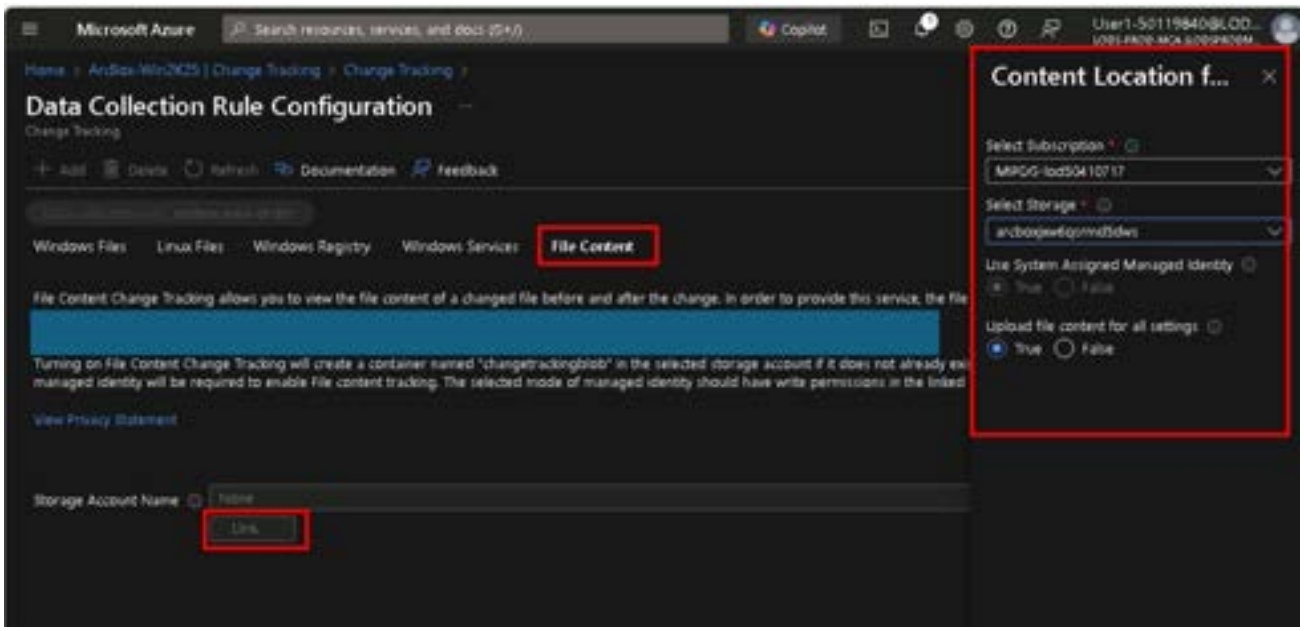
Click **Next** for the next exercise or **Go back to the main table of content**

Task 4: Track file changes

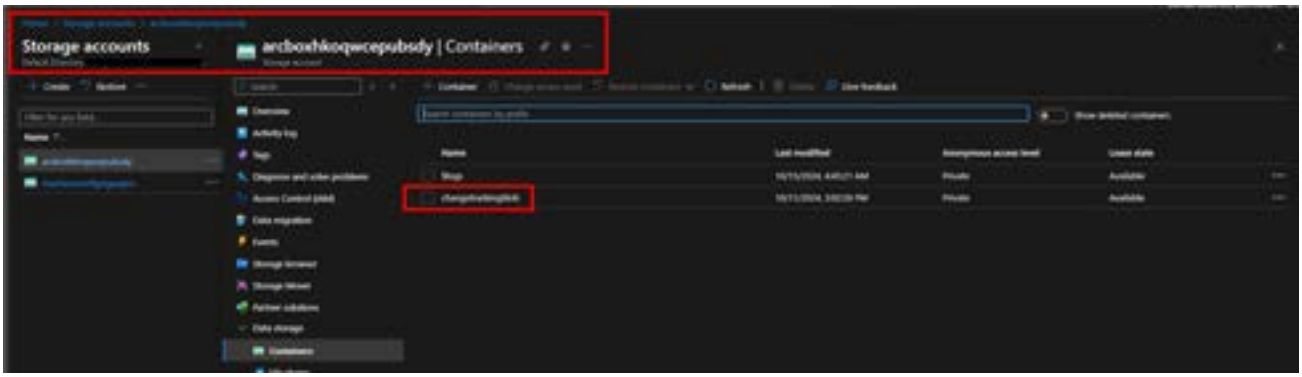
1. Navigate to *ArcBox-Win2K25* Arc-enabled Windows machines on the Azure Portal and select "Change tracking" then select "Settings" then select "Windows Files". You should see the "Add windows file setting" screen on the right hand side. Configure these settings to track the changes to the file "C:\ArcBox\ct.txt" and to upload the file content to storage.



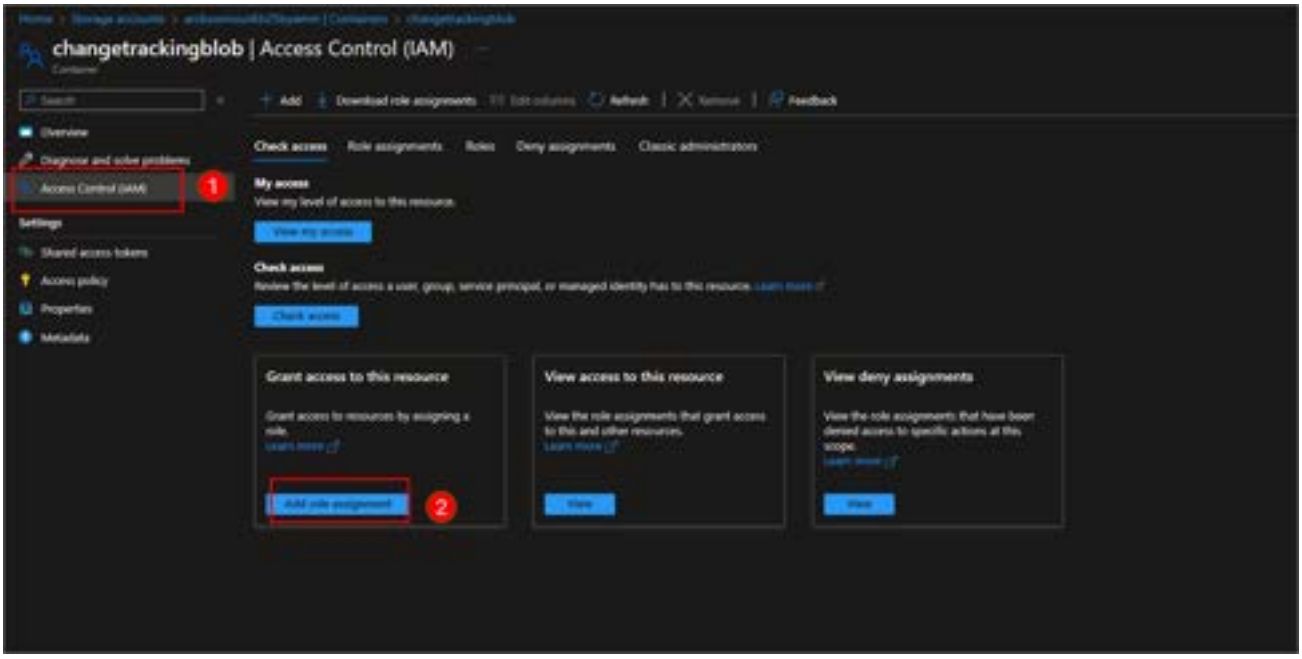
2. Set the file location where changed files will be uploaded. You should have a storage account deployed in the resource group of this lab. Click the *Link* button and set the parameters.



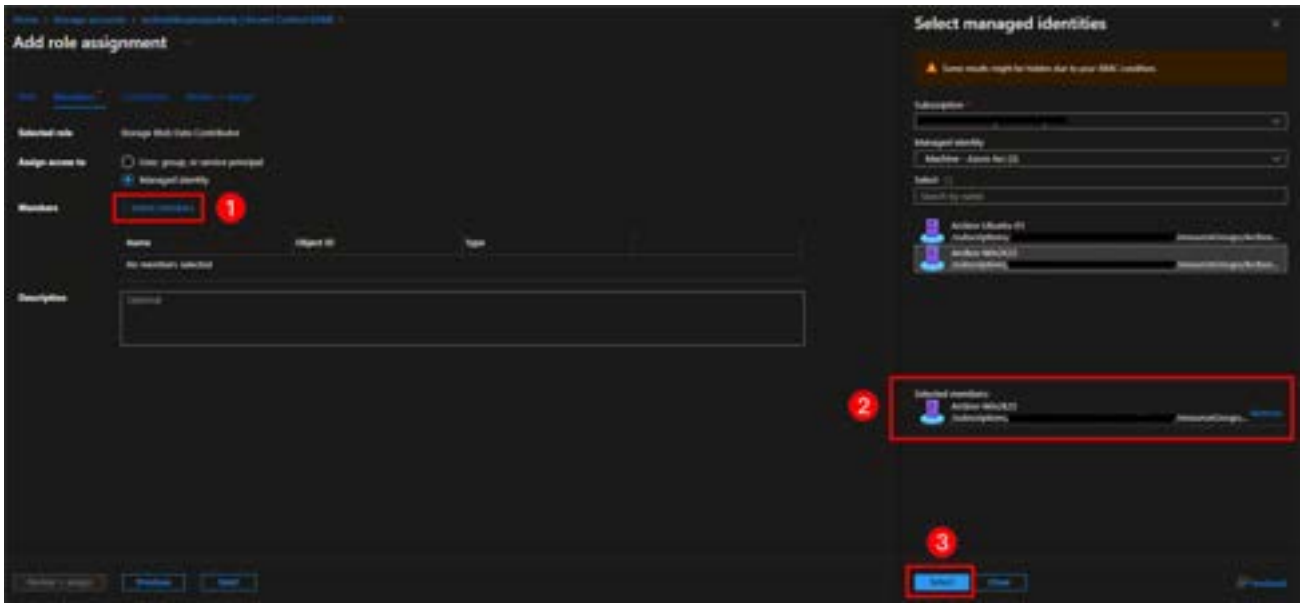
3. Navigate to the storage account. Click on "Containers" and you should see a container created automatically for you by Azure Change Tracking.



4. Click on the "changetrackingblob" container, and in the next page select "Access Control (IAM)", then on "Add role assignment".



5. Select the Storage Blob Data Contributor role then assign the role to the Windows Arc enabled machines managed identity.



☐ 6. Modify the *C:\ArcBox\ct.txt* file on the Arc-enabled machine.

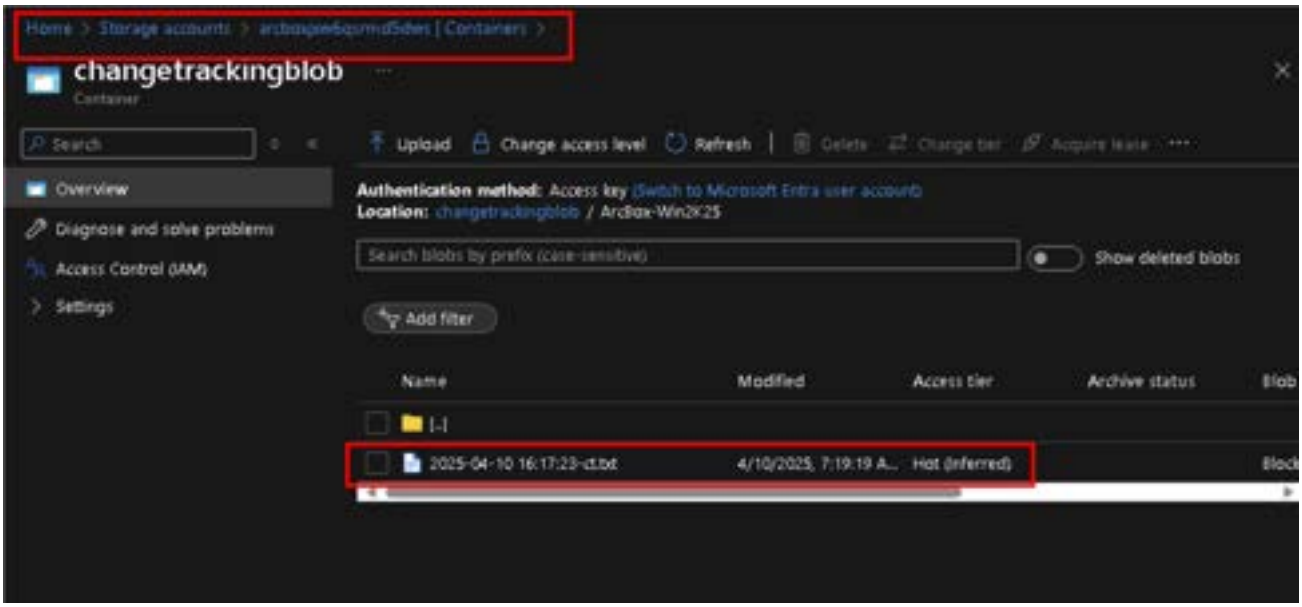
NOTE: To modify the file, open *Notepad* as Administrator, select File>Open, and then browse to C:\ArcBox\ct.txt

☐ 7. Add a line like this from an administrative notepad and save the file.

```shell
Change 1
```

☐ 8. Eventually, the file changes will show up in the change tracking page of the machine (it might take some time to show so move on to other tasks and come back to check later). The file changed content will also be uploaded to the "changetrackingblob" storage container.

☐ **It may take some for the changes to show up in the portal, you can move to the next task/exercise or lab and come back later to see the results if you prefer. This way, you can make better use of your time in this workshop**.



Task 4 has been completed

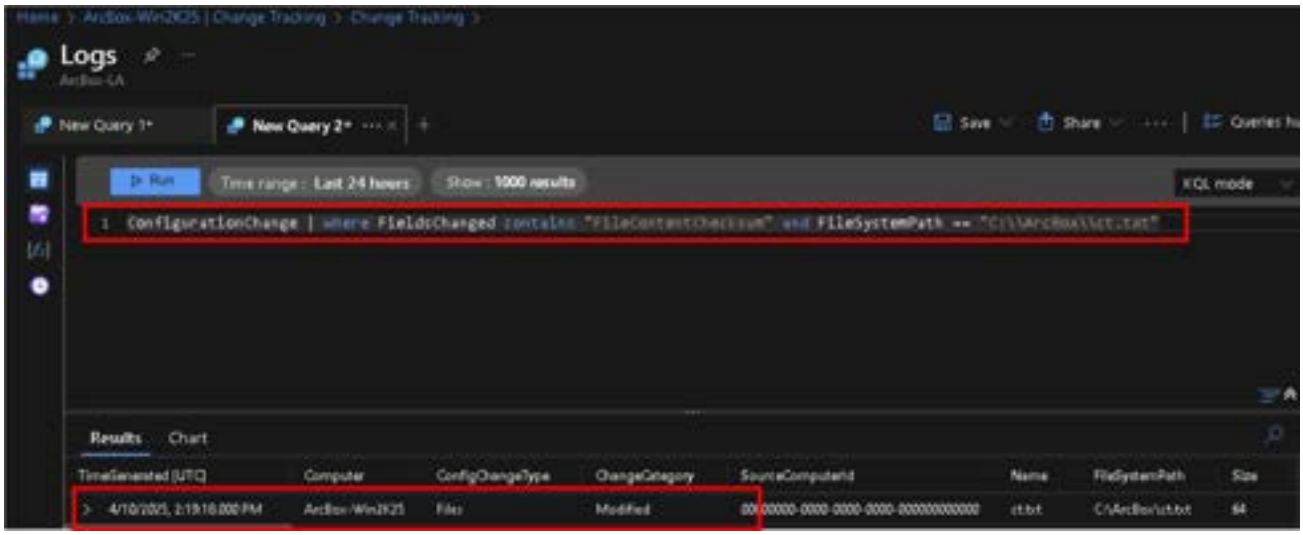
Click **Next** for the next exercise or **Go back to the main table of content**

Task 5: Query in Log Analytics

- [ ] 1. On the Change tracking page from your Arc-enabled machine, select *Log Analytics*.

- [ ] 2. In the Logs search, look for content changes to the *ct.txt* file by entering and running the following query. The result should show information about the changes.

```shell
ConfigurationChange | where FieldsChanged contains "FileContentChecksum" and Fil
```



- [ ] 4. (Optional) In Log Analytics, alerts are always created based on log analytics query result. If you want to be alerted when someone changes a file on any one of your server, then you can configure an alert by referring to this tutorial.

Task 5 has been completed

Click **Next** for the next lab or **Go back to the main table of content**

LAB06: Gain security insights from your Arc-enabled servers using Microsoft Sentinel

---

Student Lab Manual

Table of Contents

Exercise 1 - Configure data collection on Sentinel

Objective

This exercise will walk you through how to enable data collection on Sentinel for Windows security events.

Estimated Time to Complete This Lab

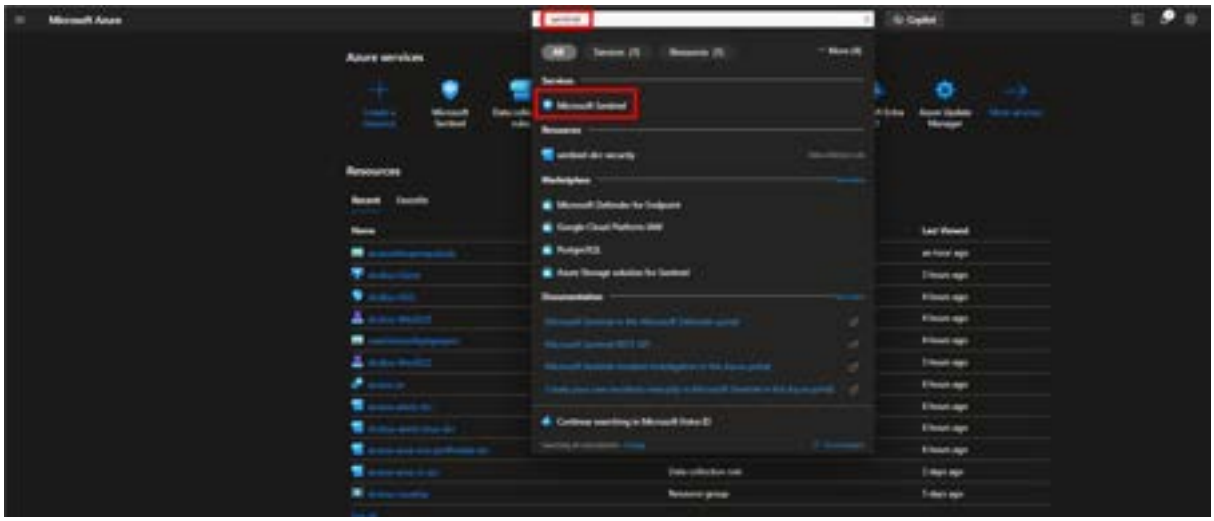
20 minutes

Explanation

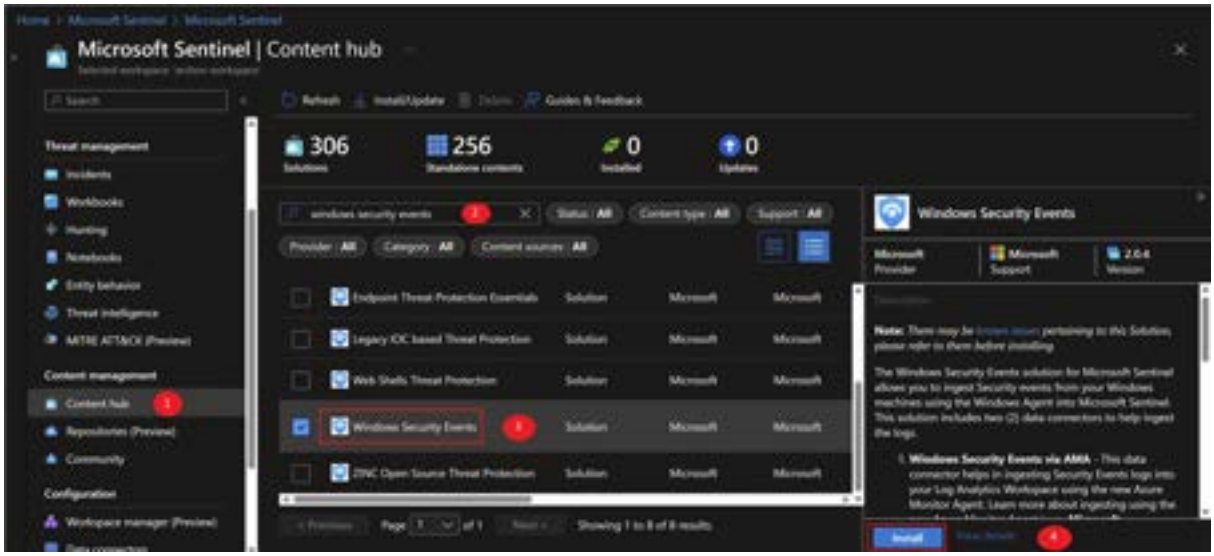
In this exercise, you will learn how to enable data collection on Sentinel for Windows security events, simulate failed logins and visualize them in Sentinel.

Task 1: Configure data collection on Sentinel

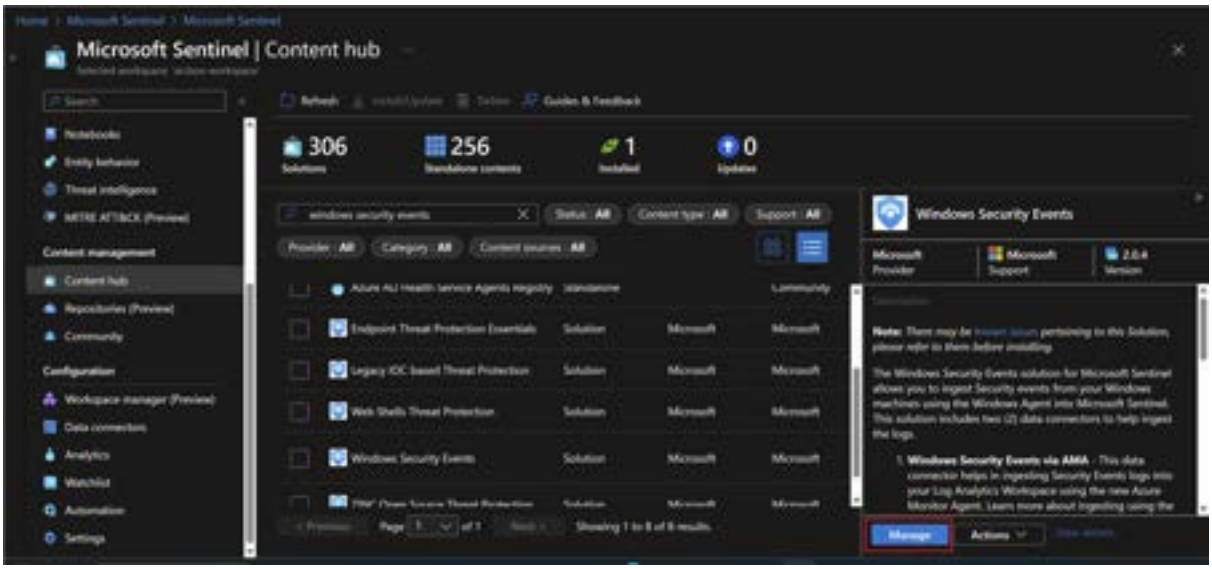
1. In the Azure Portal, search for *Sentinel*



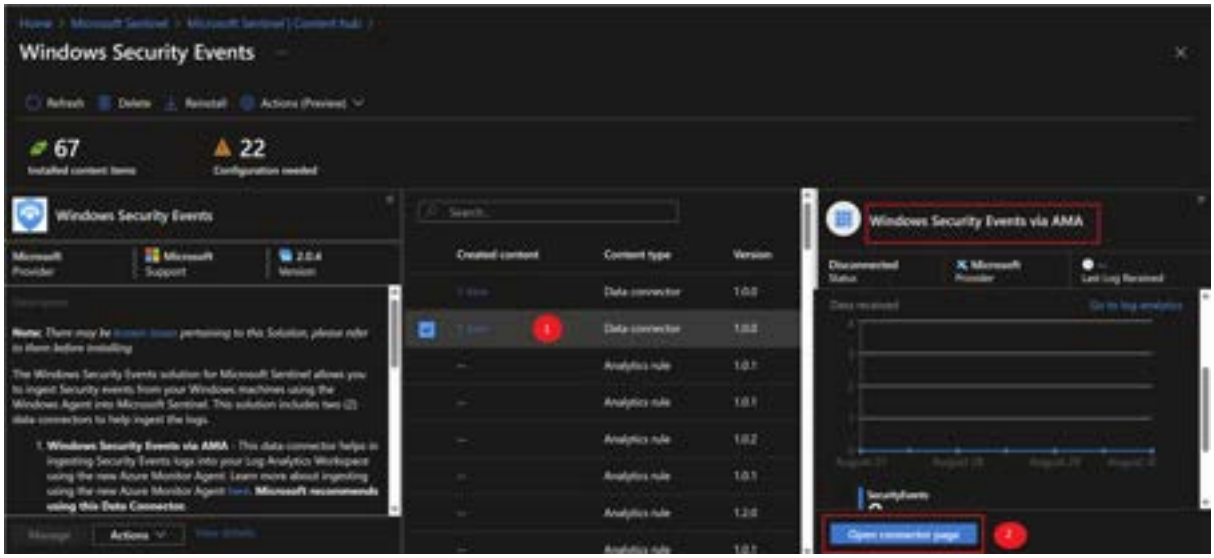
2. Click on "Content Hub" and search for "Windows Security Events" and install it.



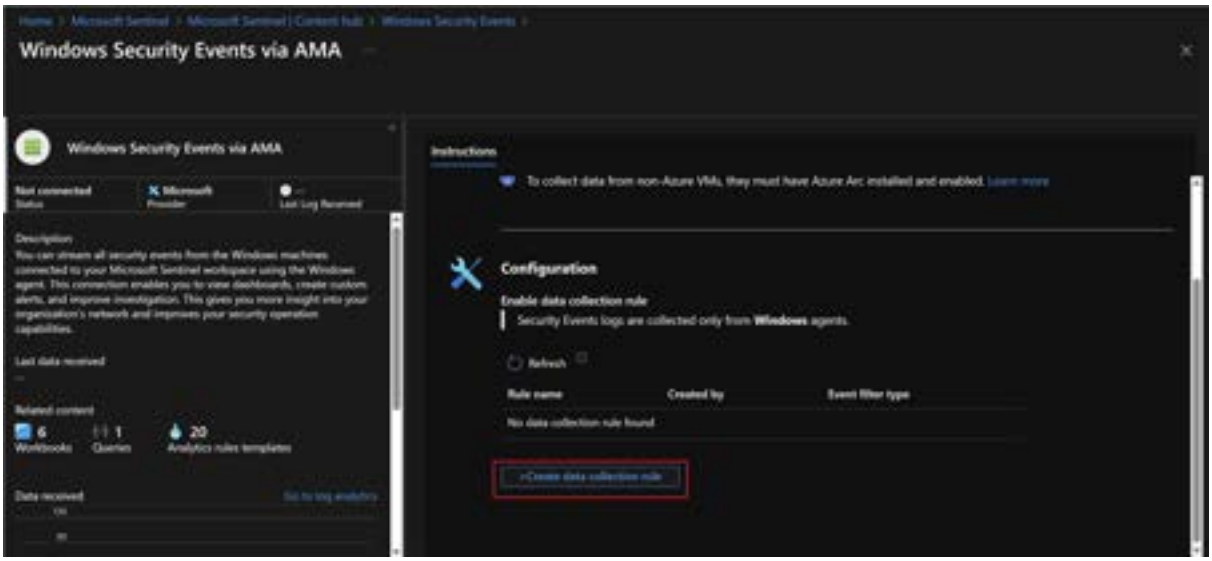
3. After installation, click on "Manage" to configure the collector.



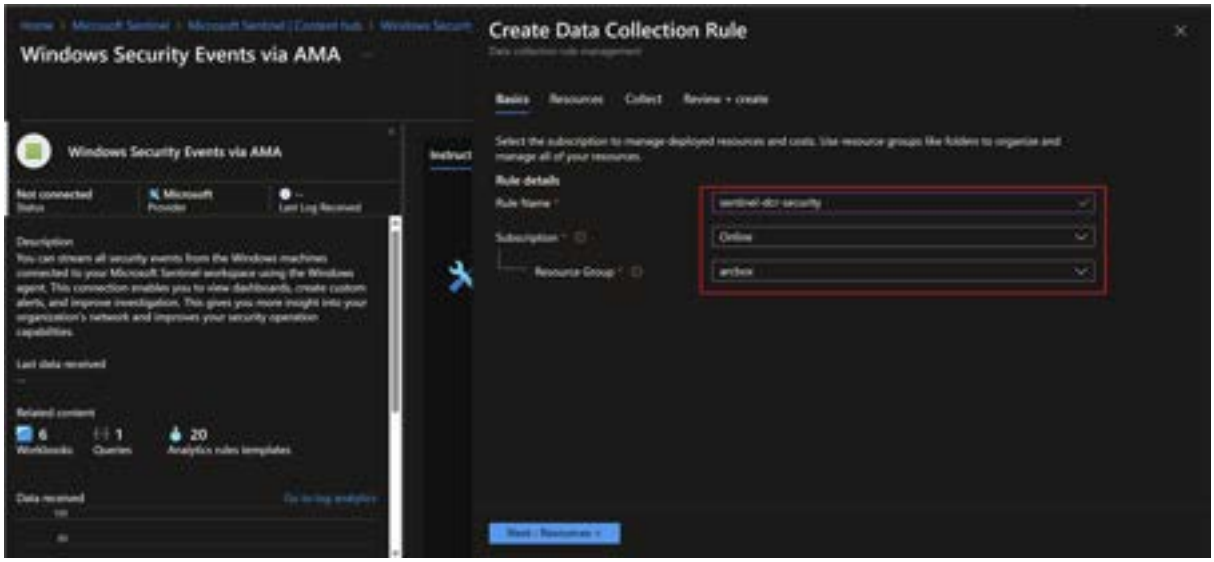
4. Select the data connector and make sure you've selected the *Windows Security Events via AMA* and click "open connector page".



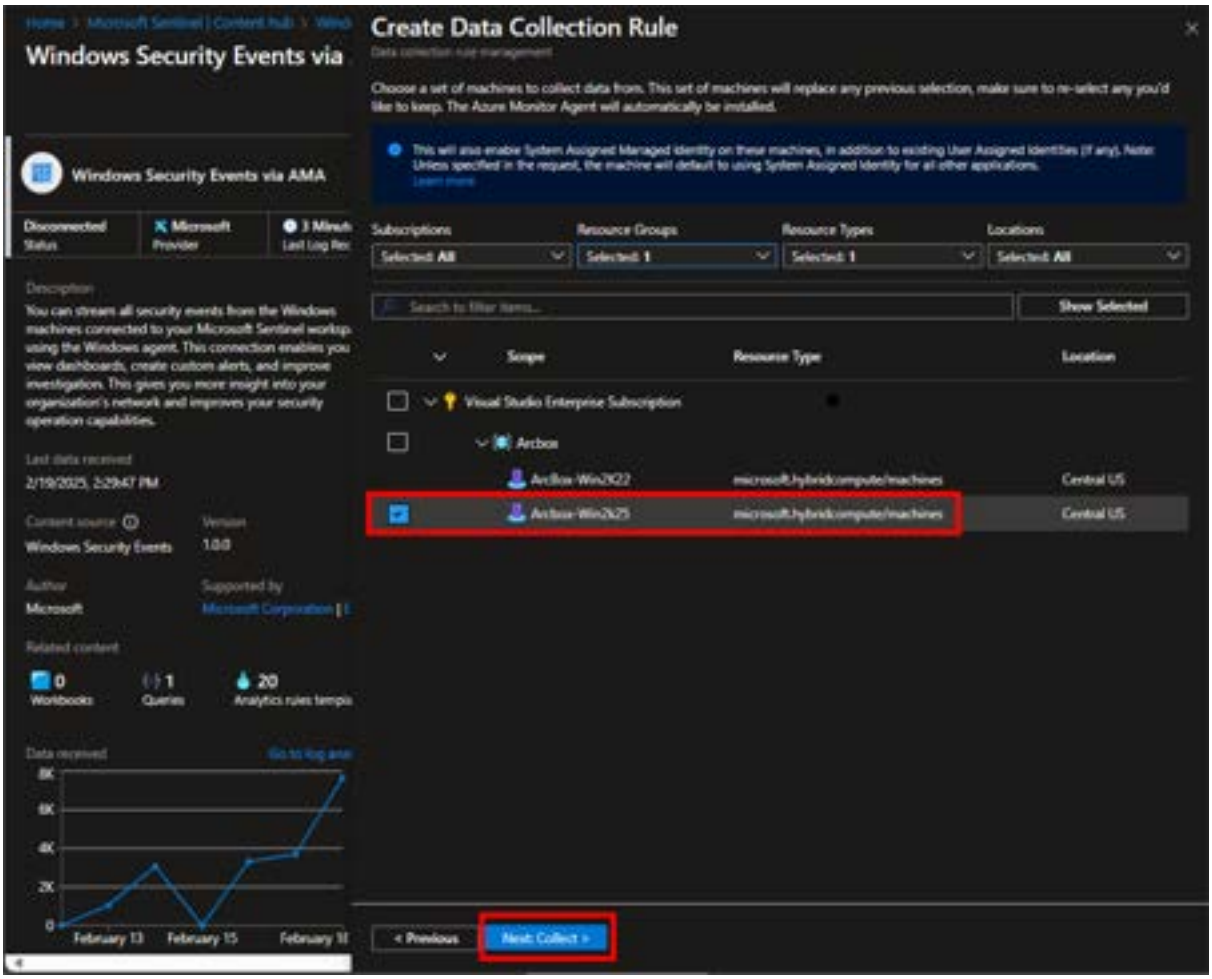
5. Create a new data collection rule.



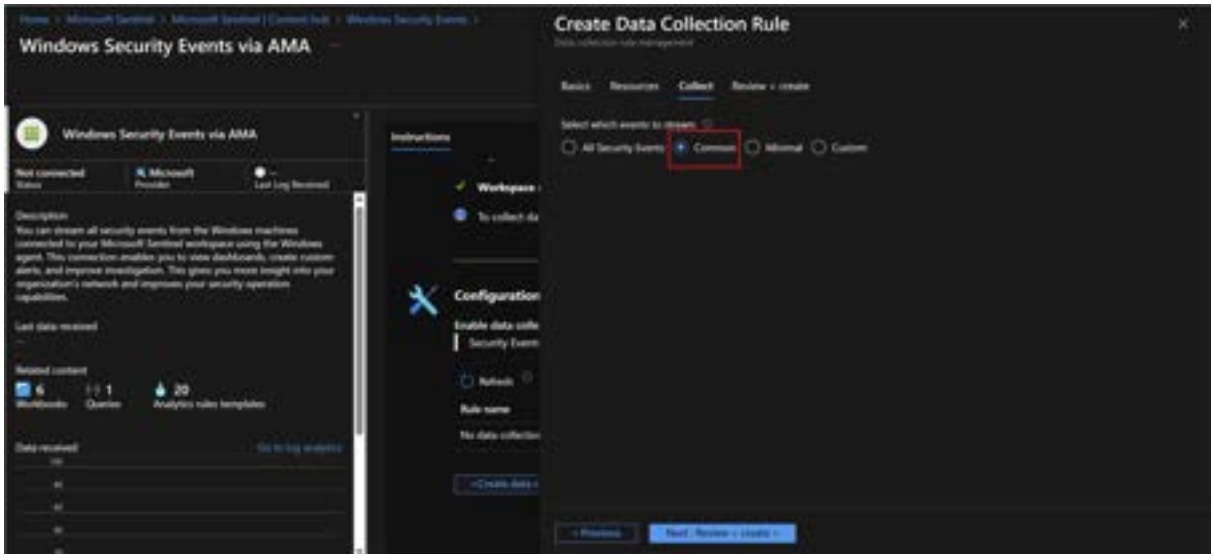
- [ ] 6. Provide a name for the data collection rule and select the same resource group where you've deployed this level-up lab.

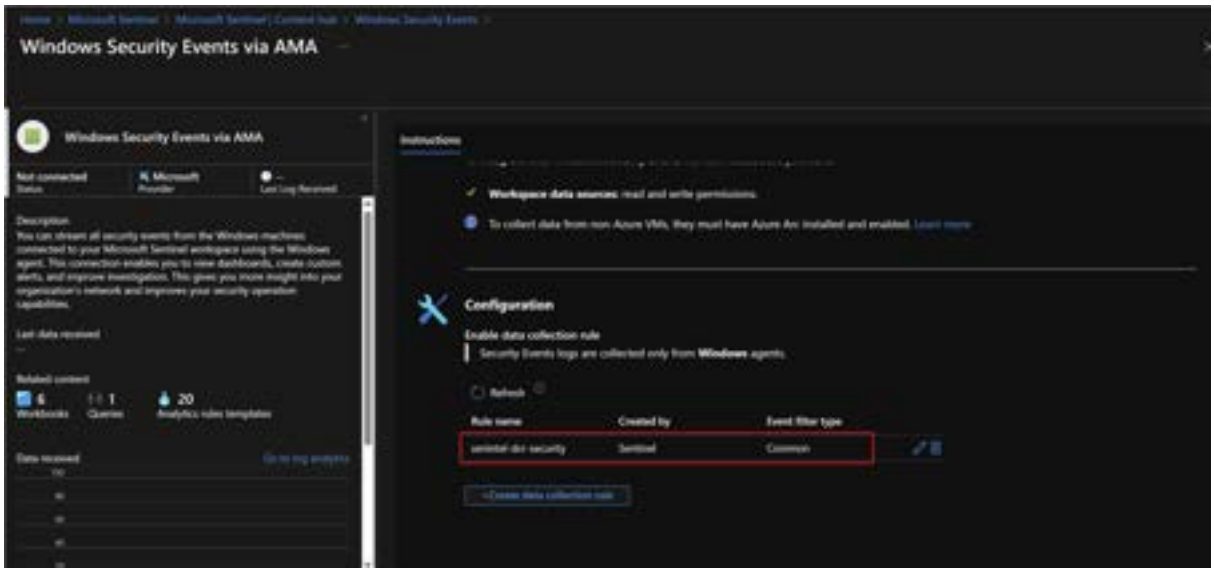


- [ ] 7. Select one or multiple Windows Arc-enabled machines.



8. Select the "Common" event type and create the data collection rule.





Task 1 has been completed

Exercise 2 - Simulating and viewing security events

Objective

This exercise will walk you through how to view Windows security events in Sentinel.

Estimated Time to Complete This Lab

20 minutes

Explanation

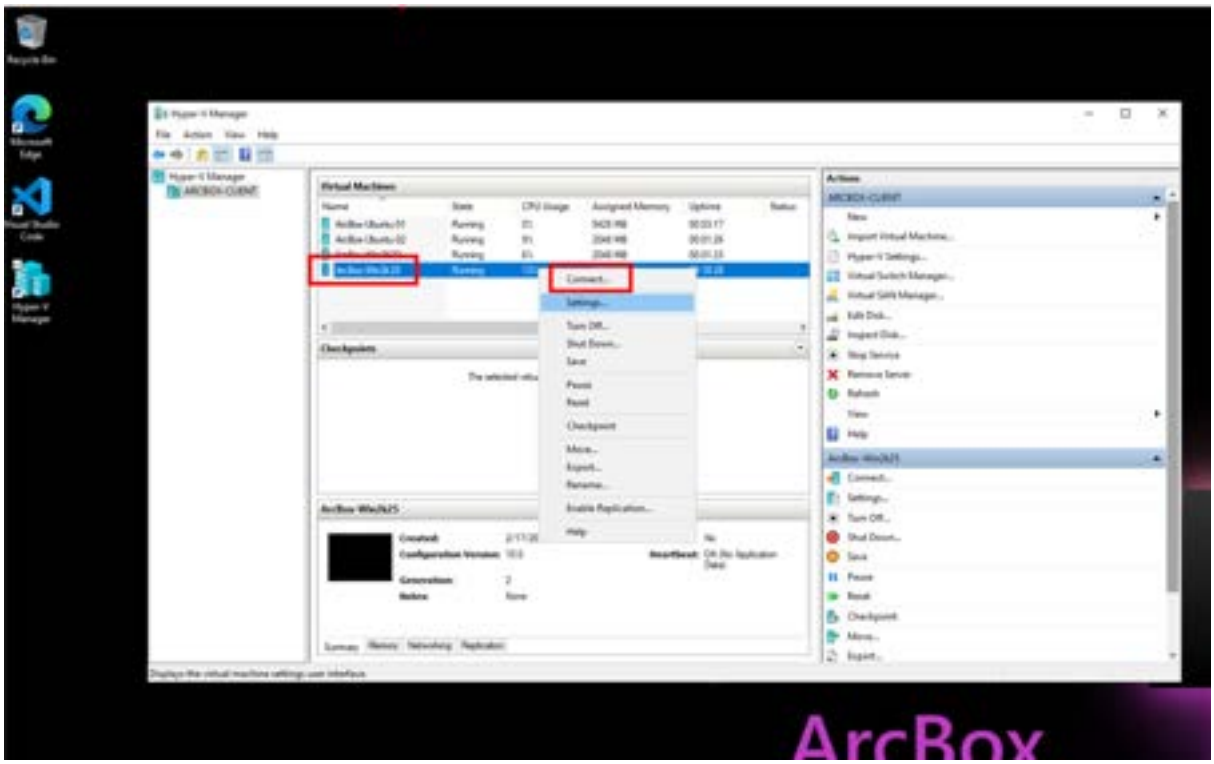
In this exercise, you will learn how to leverage Sentinel to view failed logins using Windows security events data collection.

Task 1: Simulating-and viewing security events

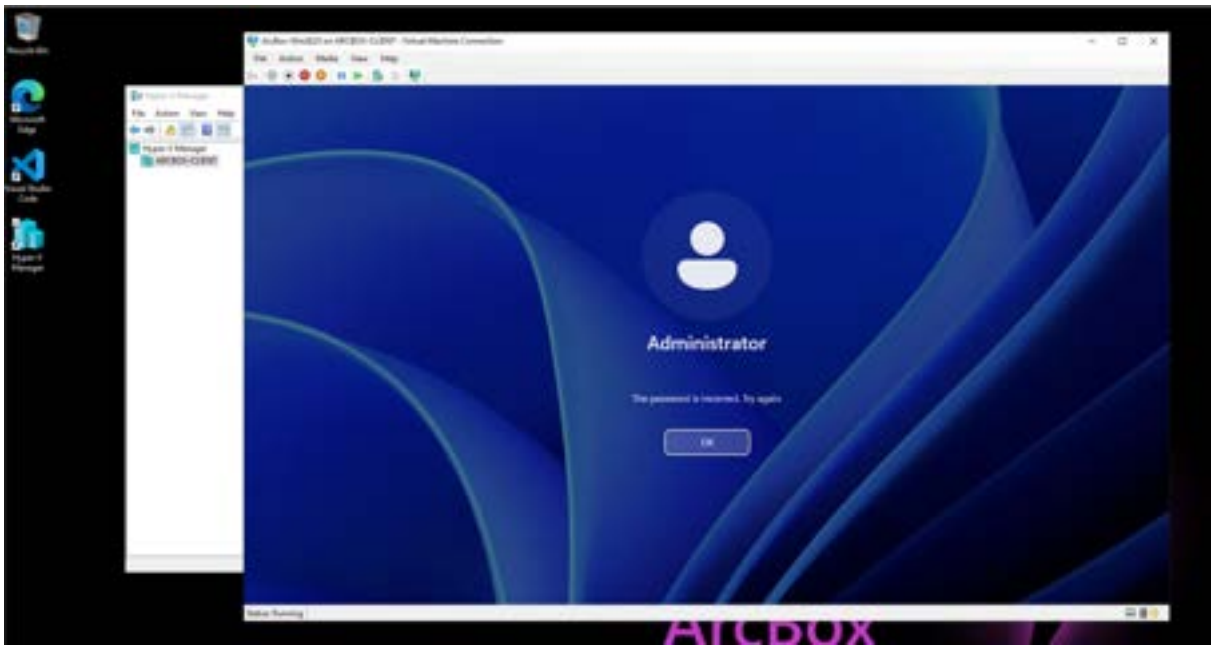
1. After configuring Sentinel, now we need to simulate some failed login attempts on one or more Windows Arc-enabled machines.

2. Connect to *ArcBox-Client* VM, and open the *Hyper-v manager*.

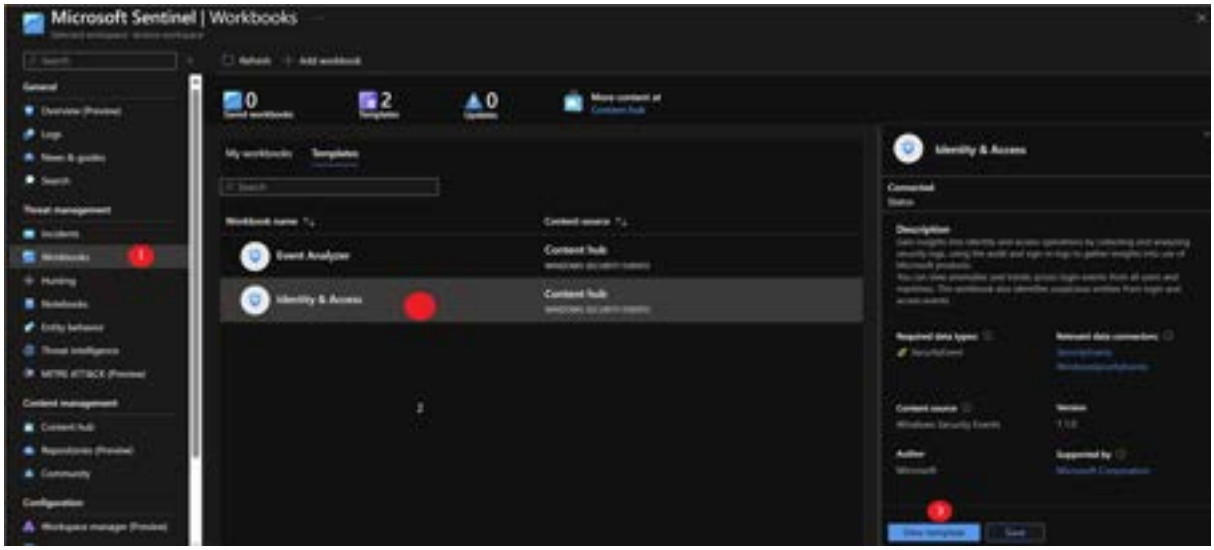
3. Right-click one of the Windows machines and connect to it.



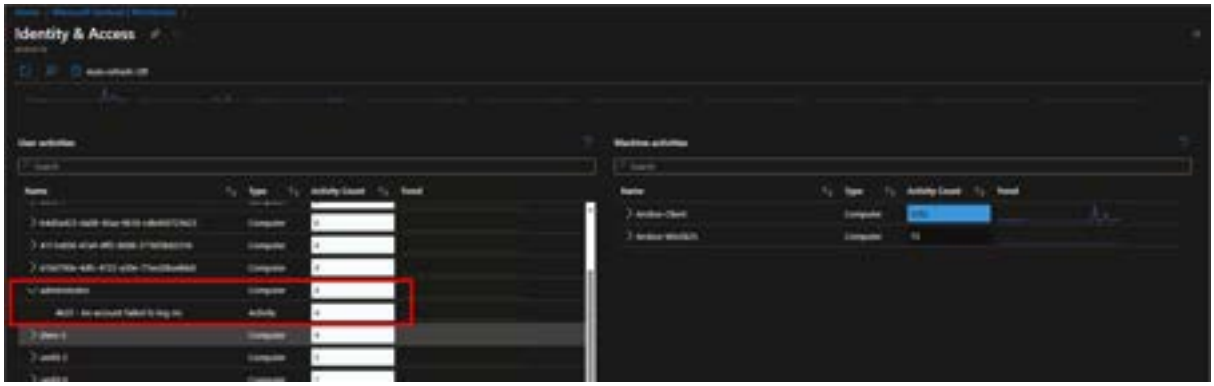
4. Simulate some failed login attempts by trying to login multiple times using an incorrect password.



5. After waiting for about 10-15 minutes for data to start getting ingested into the log analytics workspace, navigate to "Workbooks" and select the "Identity & Access" workbook.



6. Once data is being ingested, you will start seeing the failed login attempts in the workbook.



Task 1 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB07: Keep your Azure Arc-enabled servers patched using Azure Update Manager

Student Lab Manual

Table of Contents

Lab overview

Azure Update Manager is the service that unifies all VMs running in Azure together with Azure Arc, putting all update tasks in 1 common area for all supported Linux and Windows versions.

In this Lab, you will setup Azure update manager and learn how to enable it to efficiently manage all updates for your machines, regardless of where they are. You will also see some of the default reports using workbooks to monitor your Azure update manager environment.

Exercise 1 - Enable Azure update manager for the Arc-enabled servers.

Objective

In this exercise, the goal is to see all the Arc-enabled servers in the Azure update manager portal and to be able to create an automatic recurring task to refresh the status of your machines.

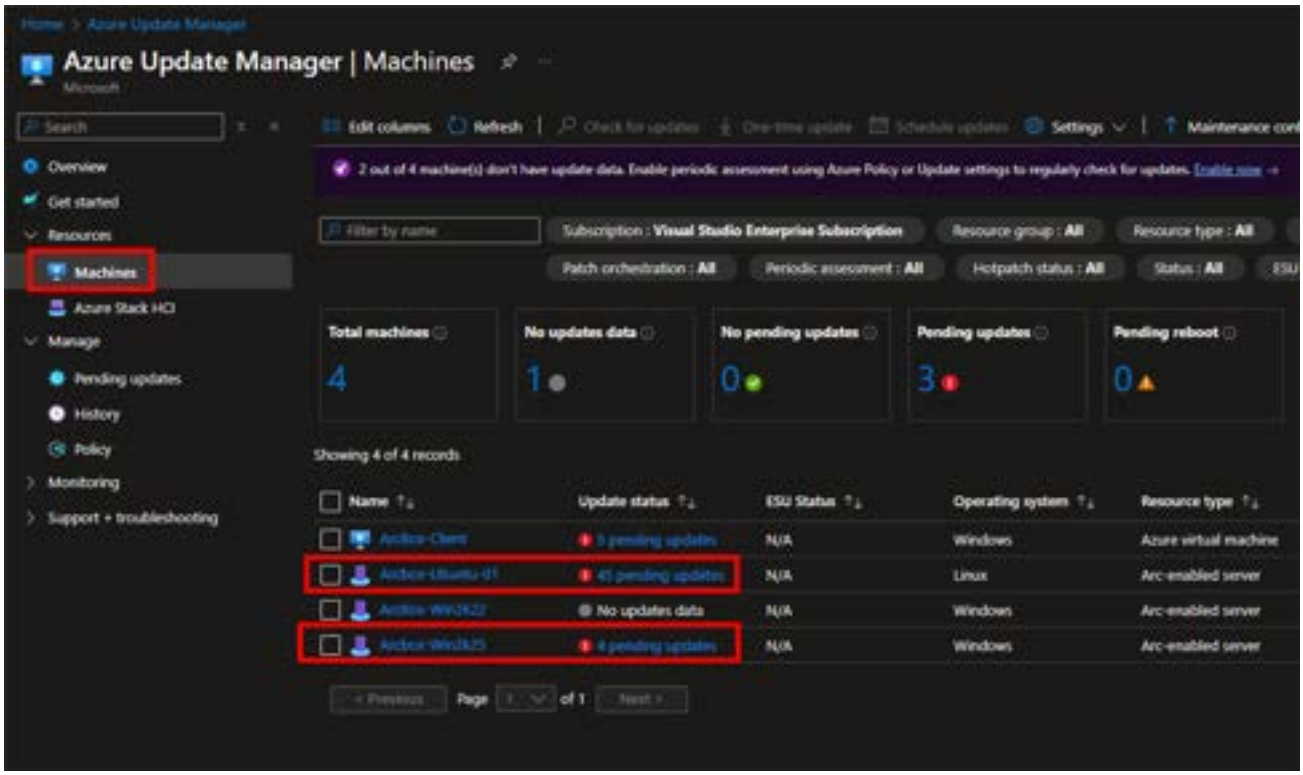
Estimated Time to complete this lab

20 minutes

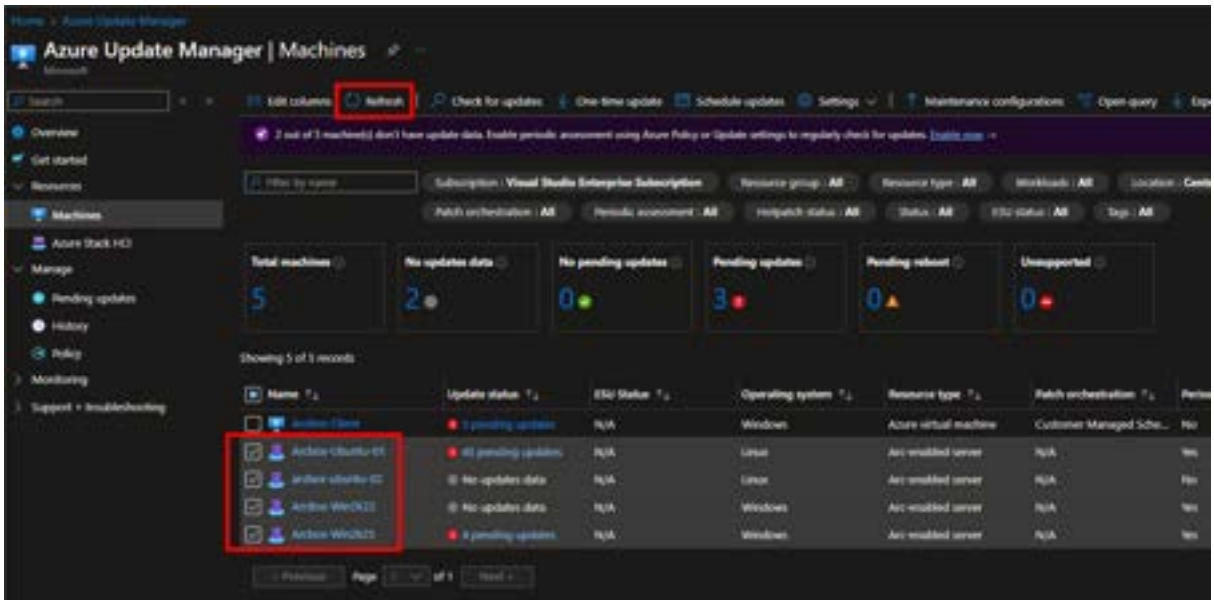
Task 1: Use the Azure portal and search for Azure Update manager

1. Find the Azure Update Manager in the Azure portal then click on "Machines" in the left blade to view all your Azure machines.

Note that all Azure VMs and Arc Server VMs are already visible in this Azure Update Manager service. In addition, the required extensions are installed on the *ArcBox-Win2k25* and *Arcbox-Ubuntu-01* Arc-enabled machines so that you can perform the tasks in this lab faster should you want to do that. As such these two machines might already show that they require some updates before carrying out the assessment in the following steps.



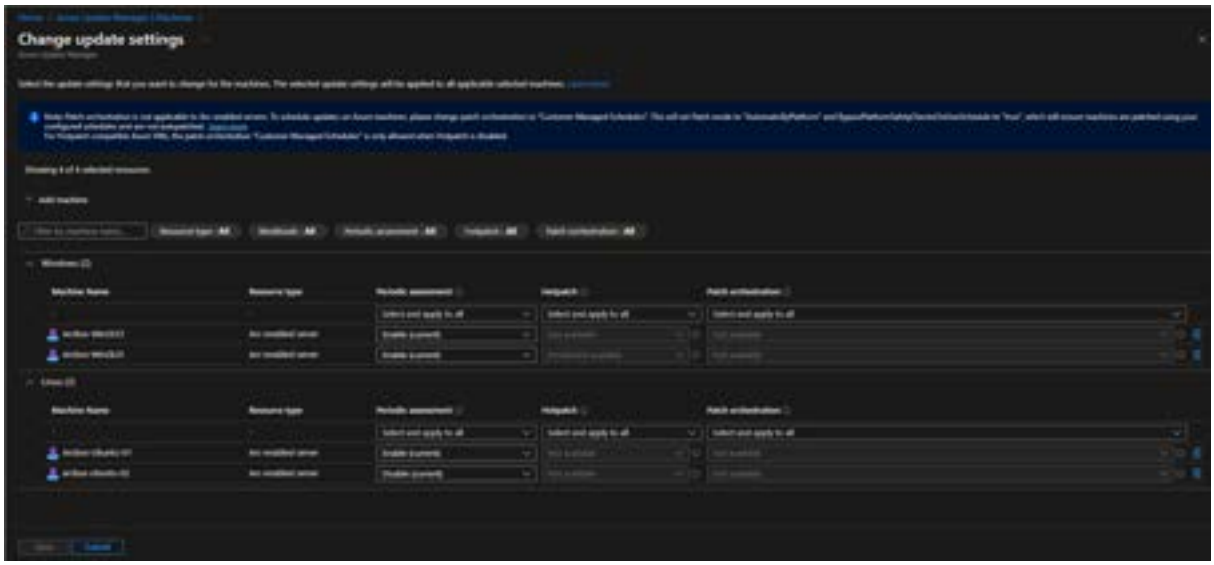
2. Select the Arc-enabled machines and click on the refresh button to refresh the current status of the selected VMs.



3. **Optional** - you can enable automatic recurring task for at scale refresh once every 24 hours.

   select the Arc-enabled servers, click on settings then choose update settings, and set the periodic assessment drop down to enable.

   Note that the rest of the VMs will automatically be enabled.



Task 1 has been completed

Exercise 2 - Apply machine updates for the Arc-enabled servers using Azure update manager

Objective

In this exercise, the goal is to apply machine updates for your Azure VMs including Azure Arc-enabled VMs using maintenance configuration and one time updates.

Estimated Time to complete this lab

30 minutes

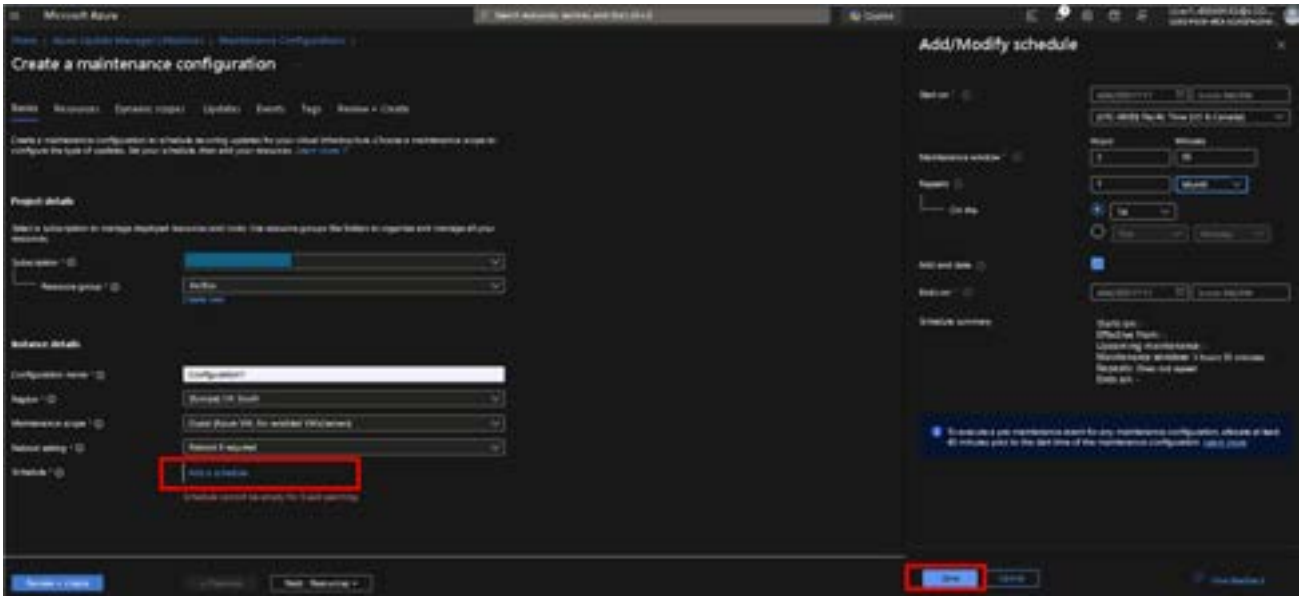
Task 1: Create a maintenance configuration from Azure update manager.

---

1. Click on *Maintenance configuration* under the "Manage" blade on the left as shown below, then select *Create maintenance configuration*.

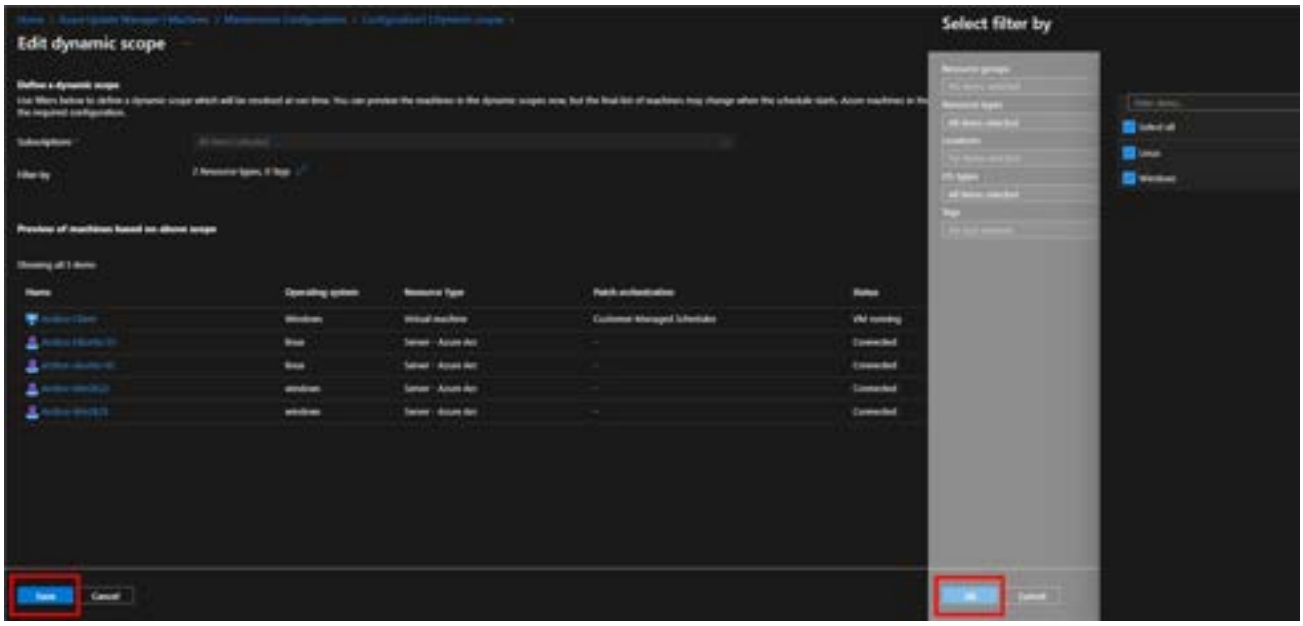
2. Fill out the basics tab as shown below, make sure you choose a region based on your location. Leave the rest as default.

Note For countries in Asia and Europe, it is a good idea to use second Tuesday + 1 day to coincide with Patch Tuesday. Do not use "Second Wednesday of the month".

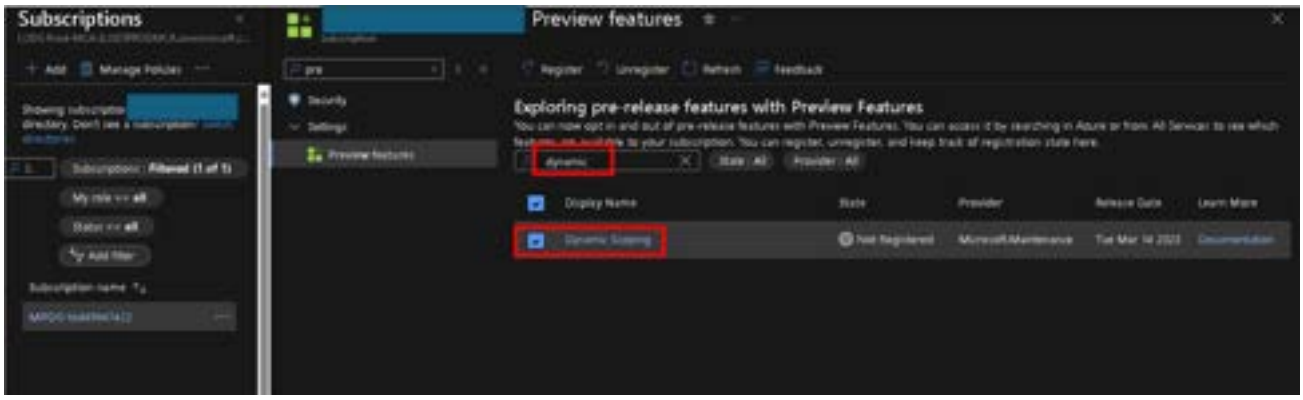


3. **Optional** click on dynamic scopes > click on the subscriptions where your Arc-enabled machines are > click on "filter by" option and choose how machines are added to this maintenance configuration (by OS, location, resource group)

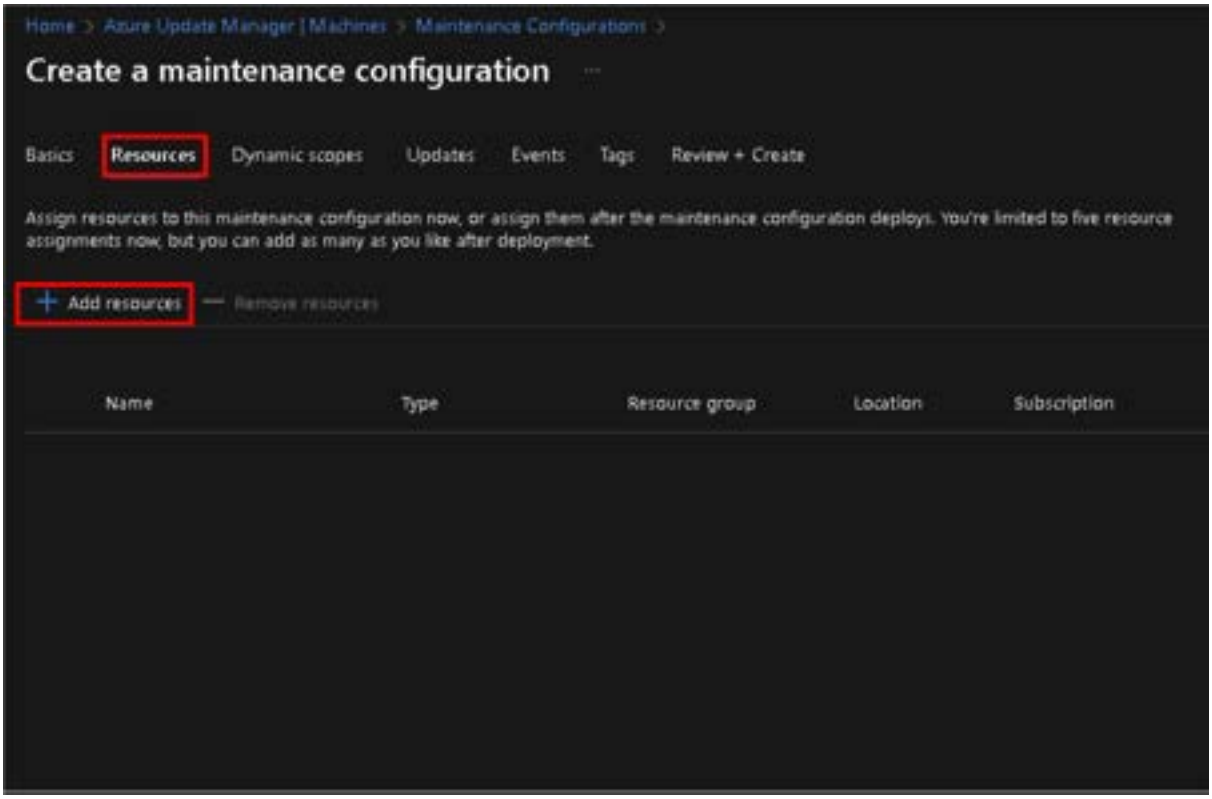
In this guide, we filtered by the OS type as shown below.



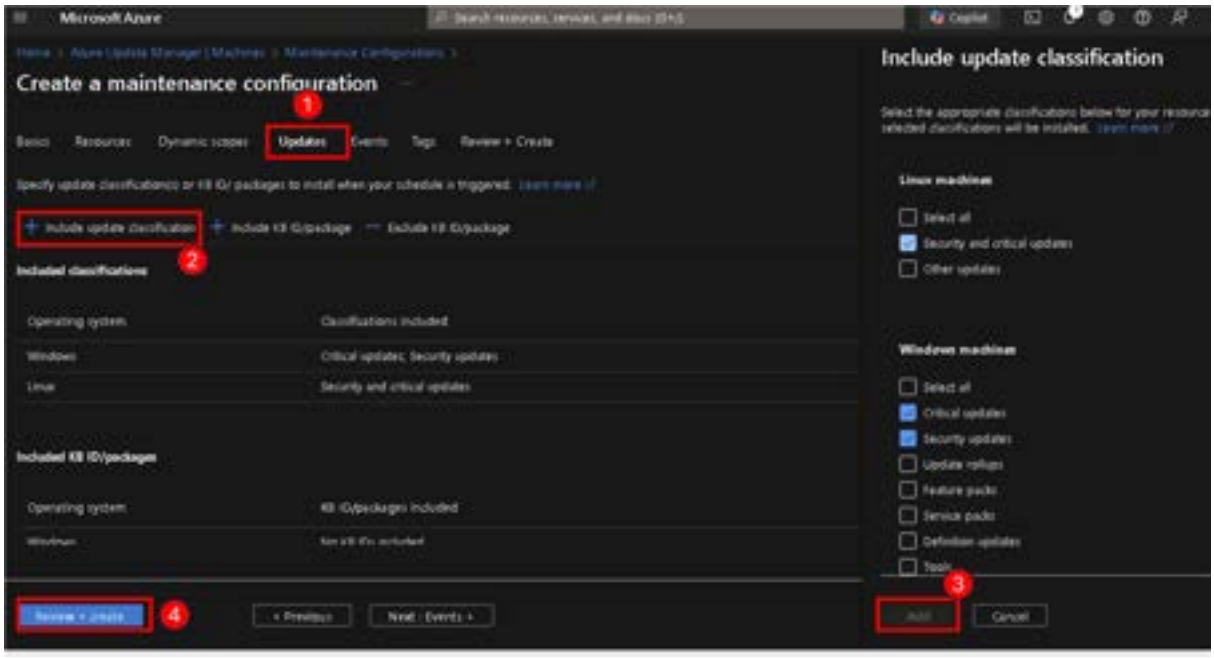
Note: If you want to enable dynamic scopes, you will need to enable the "Dynamic Scoping" preview feature in the subscription.



☐ 4. Click on the *Resources* tab to choose machines specifically instead of dynamically.



5. Click on the updates tab to choose what type of updates will be installed by this config as shown below.

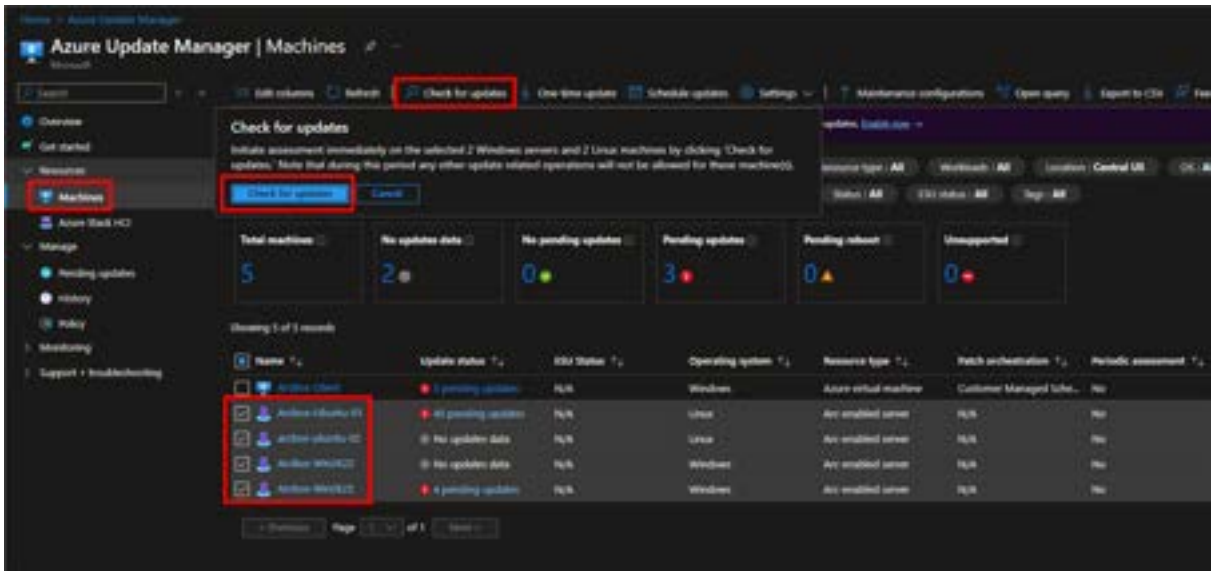


Task 1 has been completed

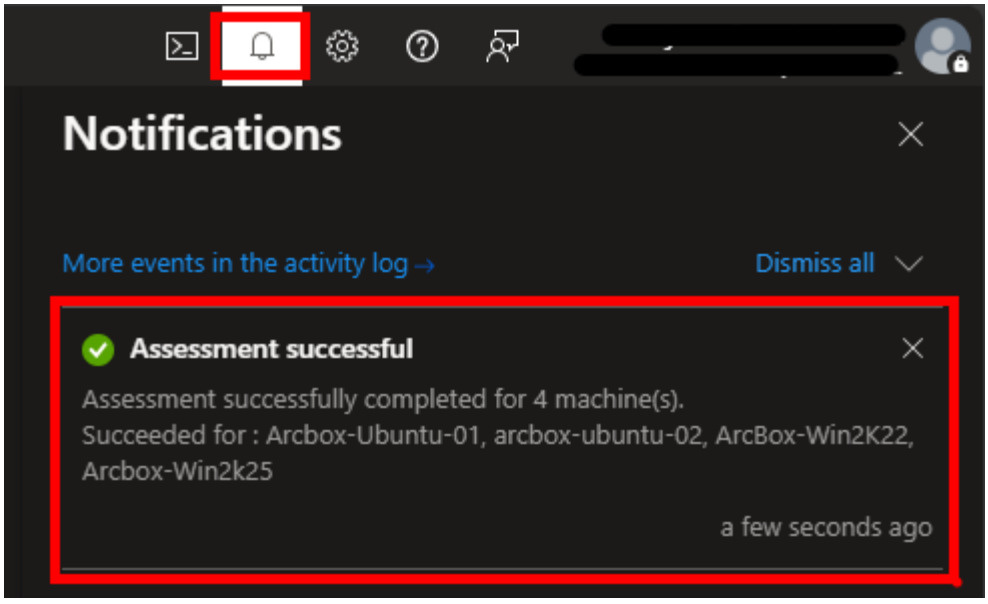
Task 2: Apply one-time updates from Azure update manager.

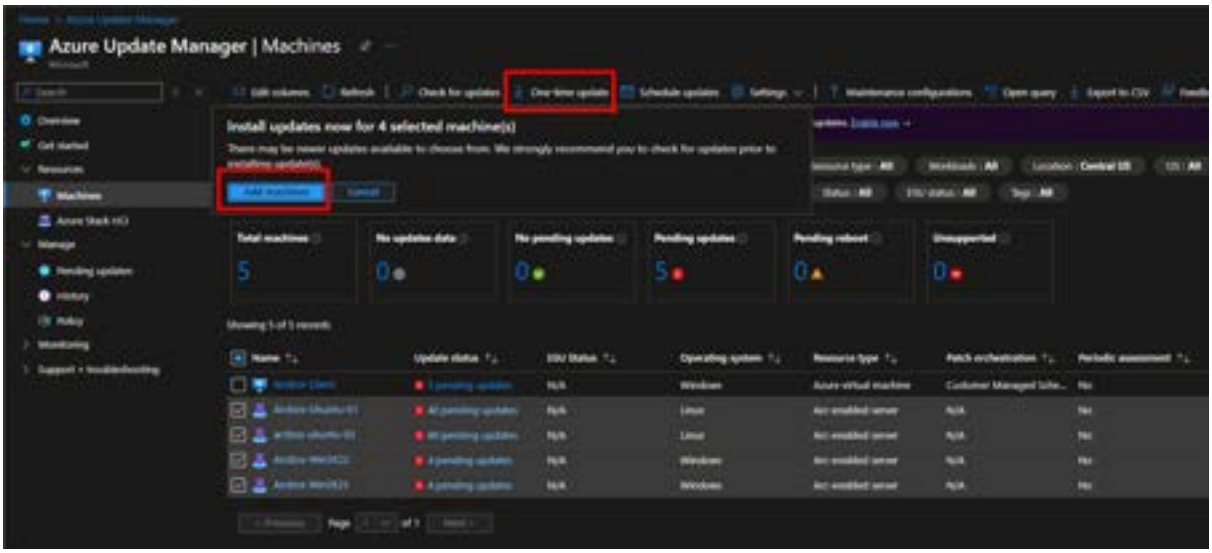
Instead of using maintenance configs with specific recurring cycles, you can also setup one-time updates (immediately!). Start by forcing an immediate refresh.

- [ ] 1. Select your Arc-enabled machines, select "Check for updates" from the top menu

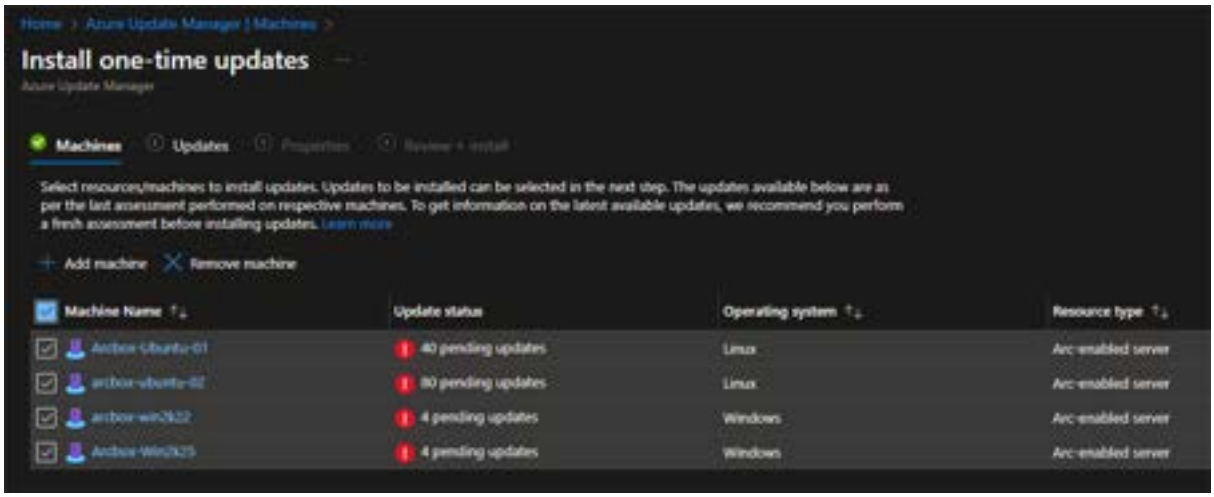


- [ ] 2. Wait for the assessment to finish then select from the top as shown below.



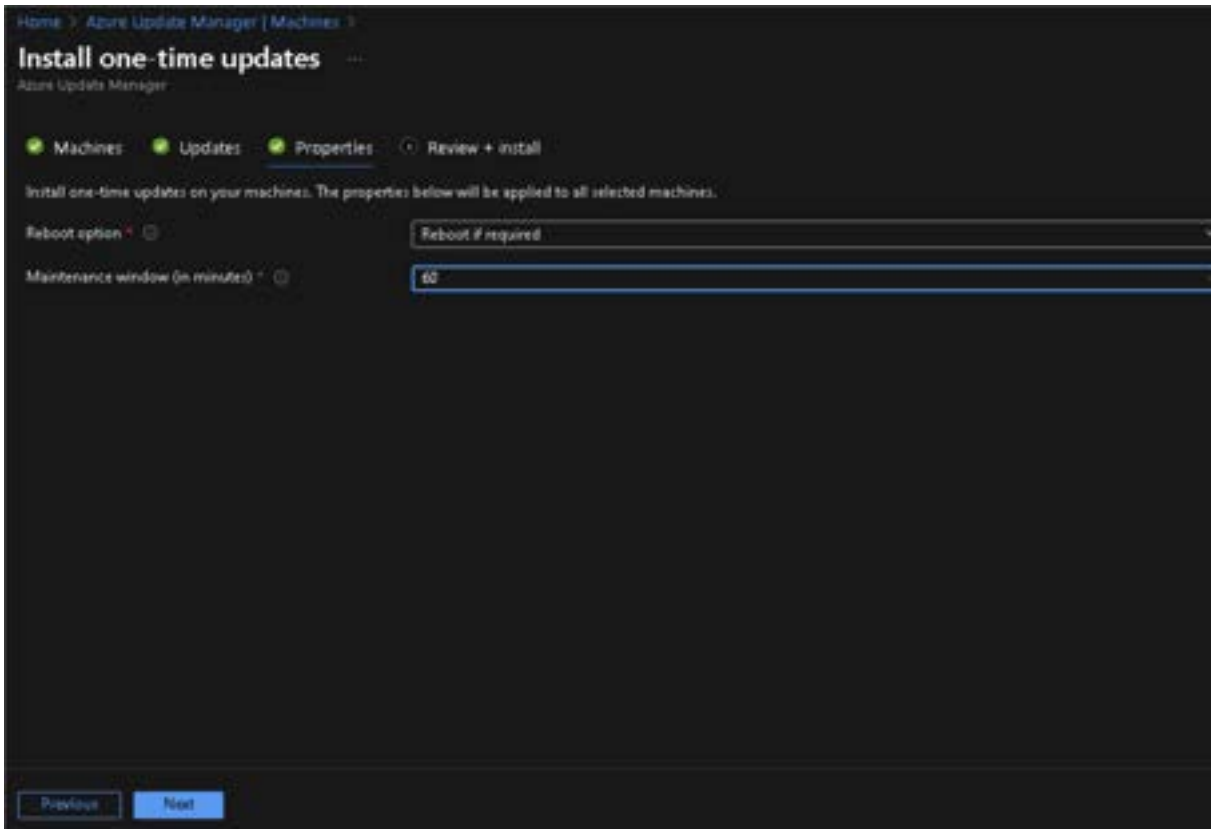


3. Confirm your machines selection from the machines tab.



4. Click on the updates tab and select updates of your choice to apply to your machines.

5. Click on the properties tab and select "reboot if required" and "60 minutes" for the Maintenance window.



- [ ] 6. Wait for a few hours and a few reboots - this can take repeated forcing for machines that have not been updated for a long time.

Task 2 has been completed

Exercise 3 - Monitor Azure update manager using the update reports.

Objective

In this exercise, the goal is to view different reports in Azure update manager that relate to your Azure Arc-enabled servers and how to navigate through them.

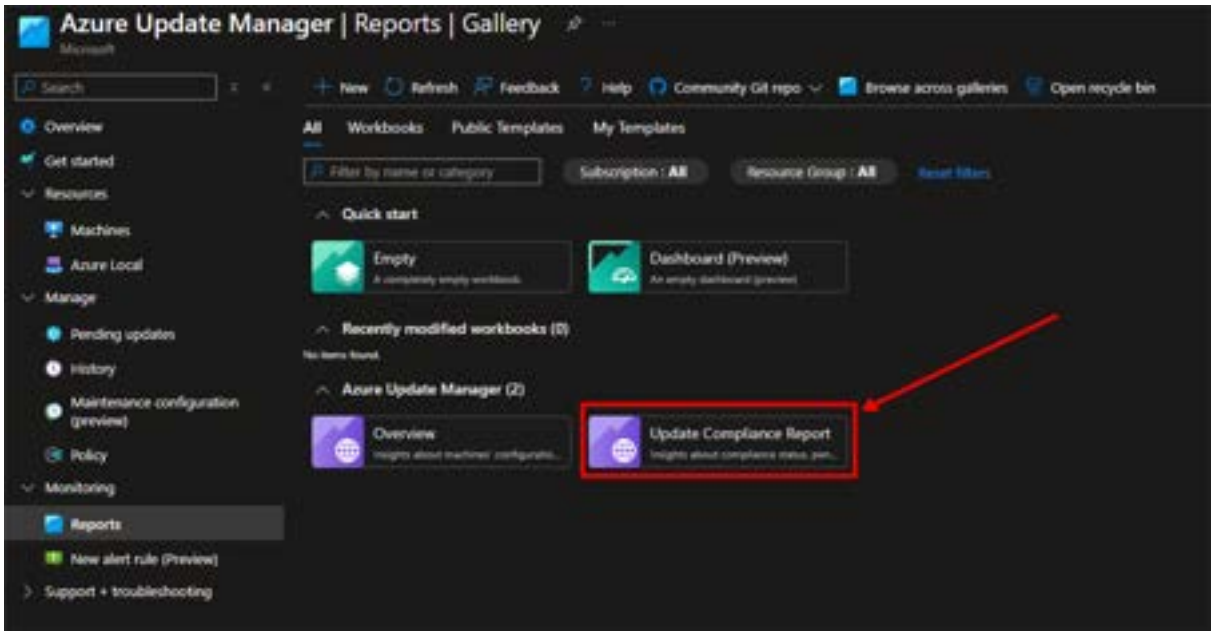
Estimated Time to complete this lab

20 minutes

Task 1: Deploy the Azure update manager Overview workbook.

Under the Monitoring part of the Update Manager, there is a default workbook, which is an overview of the Azure Update Manager. There are a few views in there that show the total number of machines connected, history of runs, and the status.

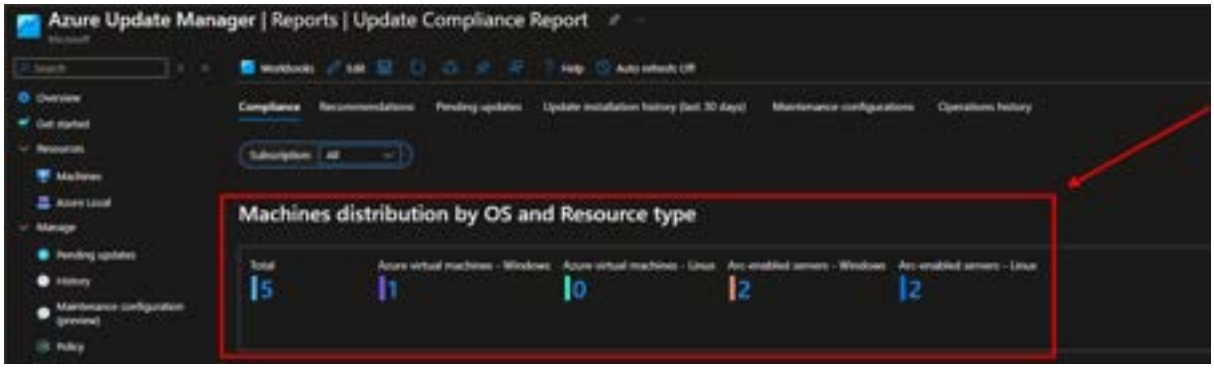
☐ 1. In Azure Update Manager, click on *Reports* under Monitoring. Select *Overview*.



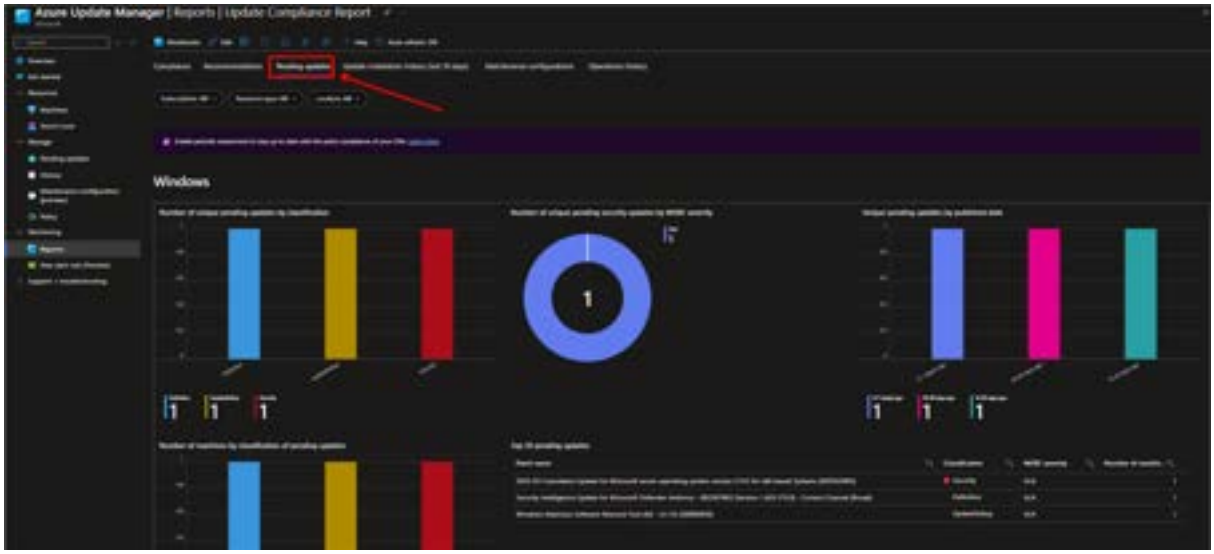
☐ 2. Select the subscription that has your Azure Arc-enabled servers.

Note that there are a number of views in there that show the total number of machines connected, history of runs, and the status.

☐ 3. Expand the "Machines overall status & configurations" view of currently connected machines, split by Azure and Azure Arc VMs, and Windows and Linux numbers. Notice the View of manual vs periodic assessments and manual vs automatically updated



4. Expand the "Updates Data Overview" view and look at the updates by classification



Please expand the rest of the views "Schedules/maintenance configurations" and "History of installation runs" to visualize the updates running in Azure Update manager.

Task 1 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB08: Enroll your Windows Server 2012/ R2 machines for Extended Security Updates with Azure Arc

Student Lab Manual

Table of Contents

Exercise 1 - Enable Extended Security Updates (ESU) license

Objective

In this module, you will learn how to enable Extended Security Updates (ESU) for Azure Arc-enabled Windows Server 2012 R2 through the Azure portal.

Estimated Time to Complete This Lab

30 minutes

Explanation

Once Windows Server 2012 and 2012 R2 are registered as Azure Arc-enabled servers, you can enroll them for ESU via the Azure portal, connect through Azure Arc, and you'll be billed monthly via your Azure subscription.

Documentation

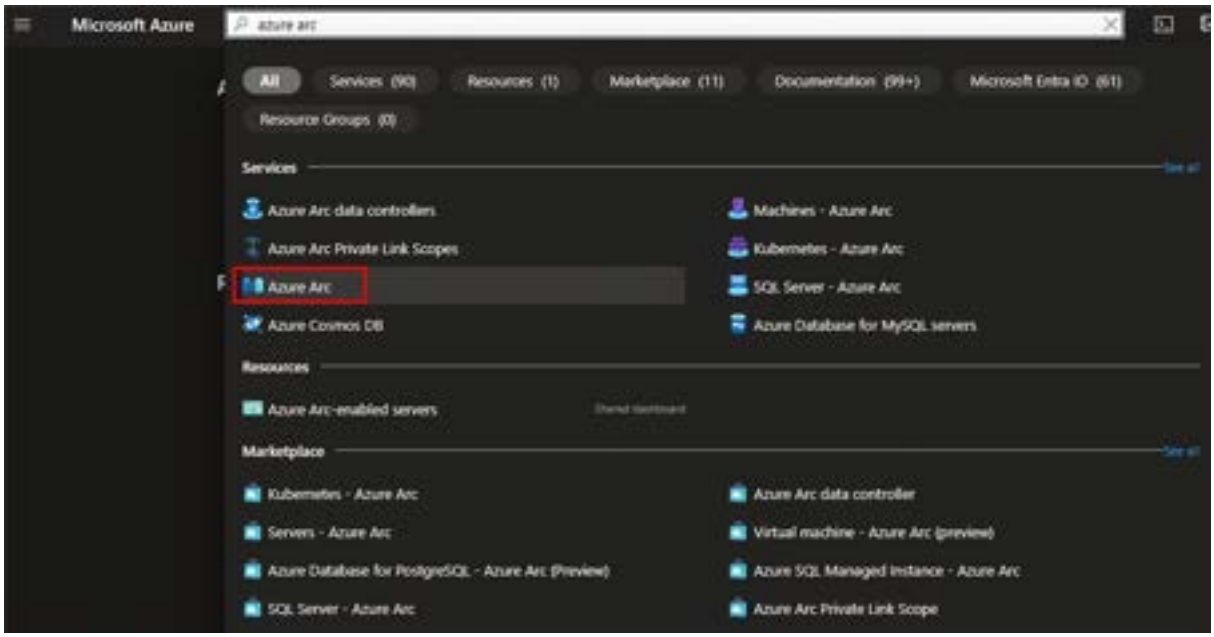
Deliver Extended Security Updates for Windows Server 2012

License provisioning guidelines for Extended Security Updates for Windows Server 2012

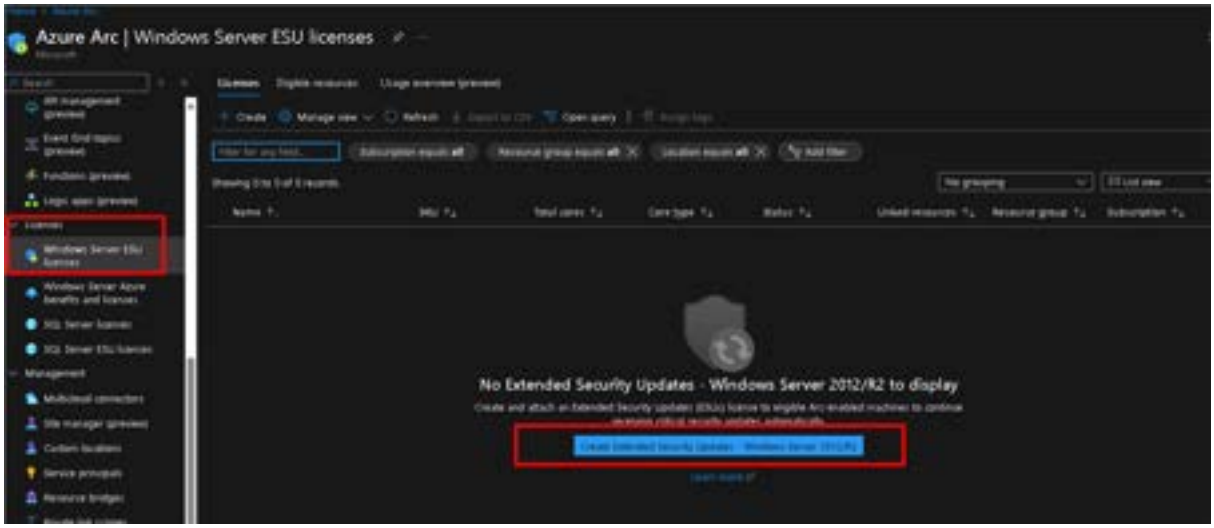
```
>[!knowledge] To continue receiving security updates after extended support has
```

Task 1: Create a ESU license

1. Navigate to the Azure Arc page

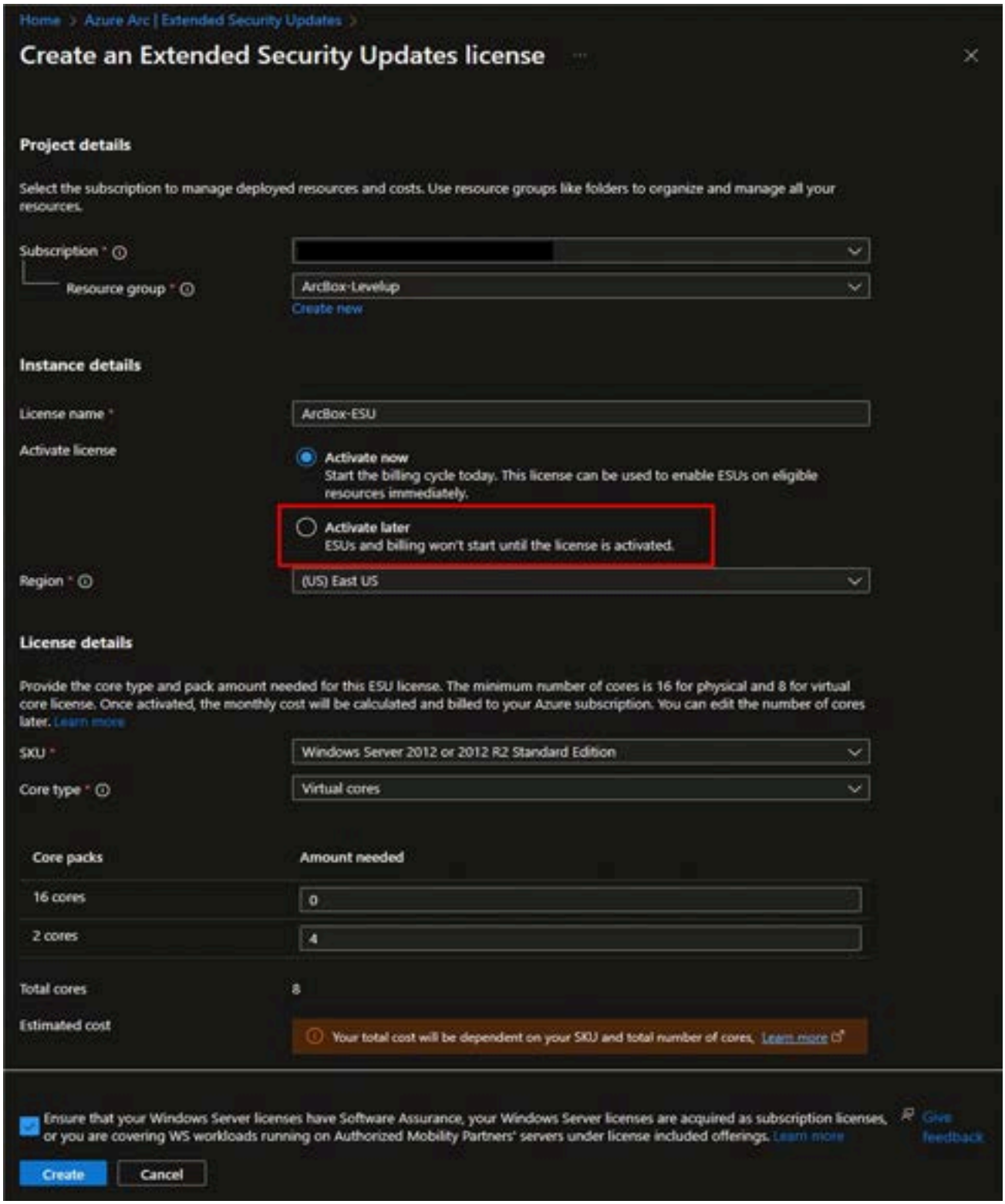


2. Select *Windows Server ESU licenses* in the left pane. Then click on *Create Extended Security Updates - Windows Server 2012/R2*



3. Provision Windows Server 2012 and 2012 R2 Extended Security Update licenses from Azure Arc.

Make sure to select "Activate later" for the license settings to avoid getting billed on the Azure Subscription



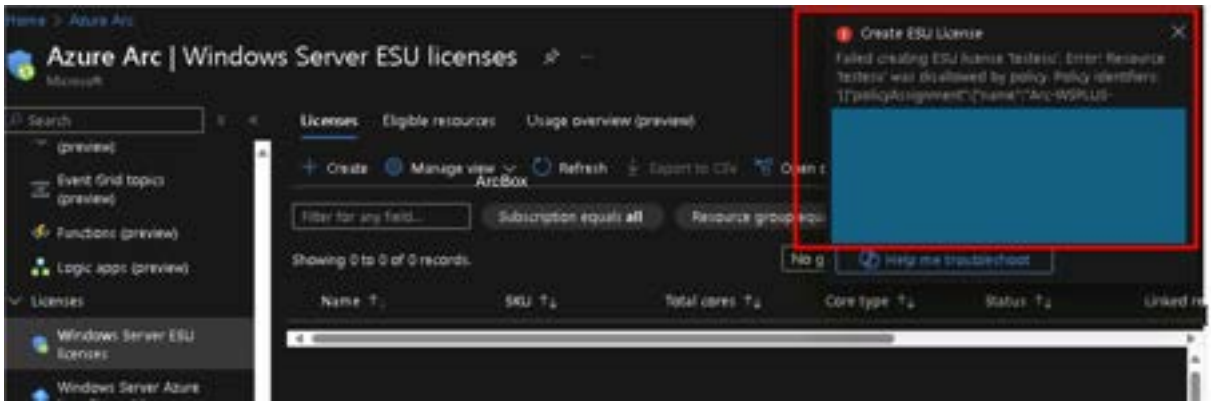
- [ ] When opting for ESU via Azure Arc for Windows Server, you have two licensing choices:

    1. **vCore Licensing**: Pay based on the number of virtual cores (vCores) utilized by the operating system. This option uses the Standard edition rate. If you're operating multiple VMs, the cost will be calculated based on the total number of vCores across all VMs. **There is an 8-core minimum per VM for vCore licensing.**

2. **pCore Licensing**: Pay based on the number of physical cores (pCores) utilized by the host operating system. This option can use either edition. Note that with pCore licensing, up to 2 guest VMs running on a WS Standard host are covered (additional VMs require additional ESU licenses). With the WS Datacenter host, all VMs are covered without the need for additional licenses. **There is a 16-core minimum per server for pCore licensing.**

Please see here for details.

- [ ] 4. In this lab environment you will not be able to create an actual license due to the cost involved with back-billing. So when you attempt to create the license you should see the following message:



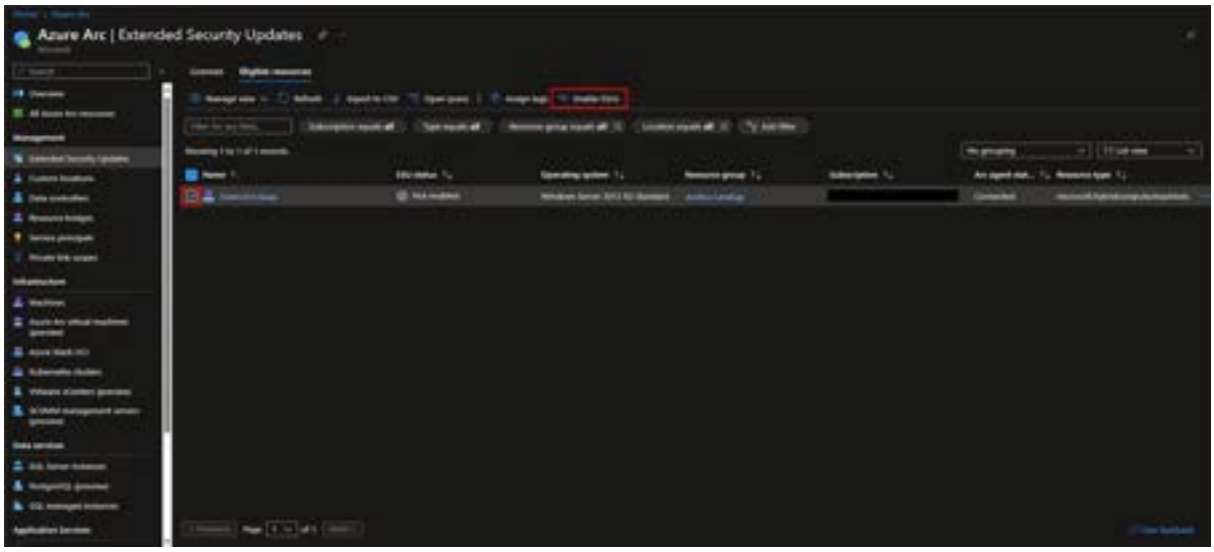
Task 1 has been completed

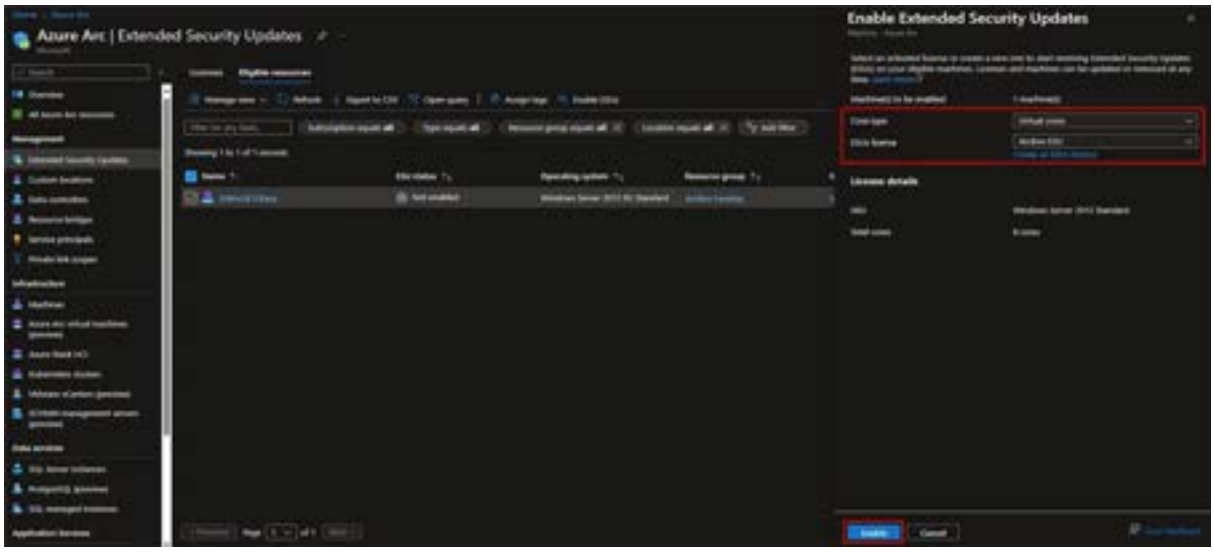
Click **Next** for the next task or **Go back to the main table of content**

Task 2: Link ESU license to Arc-enabled servers

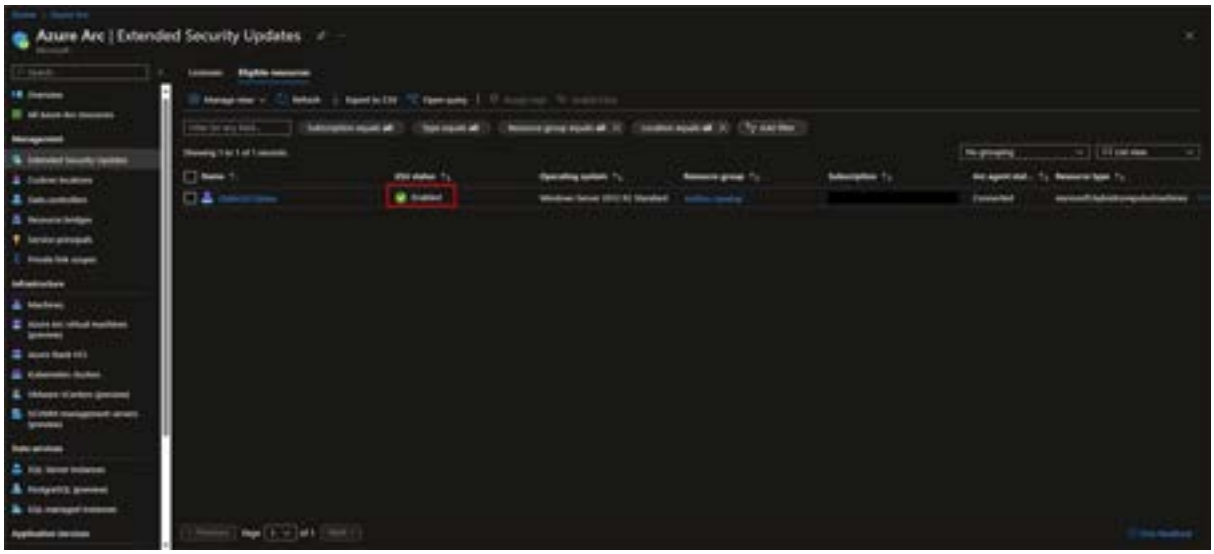
Since this lab environment does not allow the creation of an actual ESU license, you will not be able to link it to any Arc-enabled machine. So the guidance in this task is just informational and you will not have a Windows server 2012 machine provisioned within this lab.

1. Select one or more Arc-enabled servers to link to an Extended Security Update license.





2. The status of the selected machines changes to **Enabled**.



- [ ] Once you've linked a server to an activated ESU license, the server is eligible to receive Windows Server 2012 and 2012 R2 ESUs.

Task 2 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB09: Learn about Windows Admin Center single pane glass management

Table of Contents

===

Exercise 1 - Manage your Arc-enabled Windows machines using the Windows Admin Center

In this exercise you will learn how to use the Windows Admin Center in the Azure portal to manage the Windows operating system of your Arc-enabled servers, known as hybrid machines. You can securely manage hybrid machines from anywhere without needing a VPN, public IP address, or other inbound connectivity to your machine.

Estimated Time to Complete This Lab

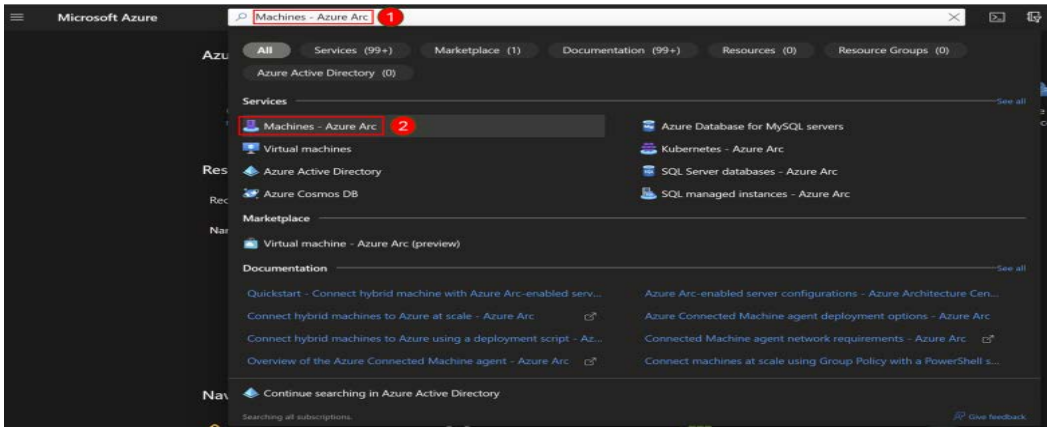
40 minutes

===

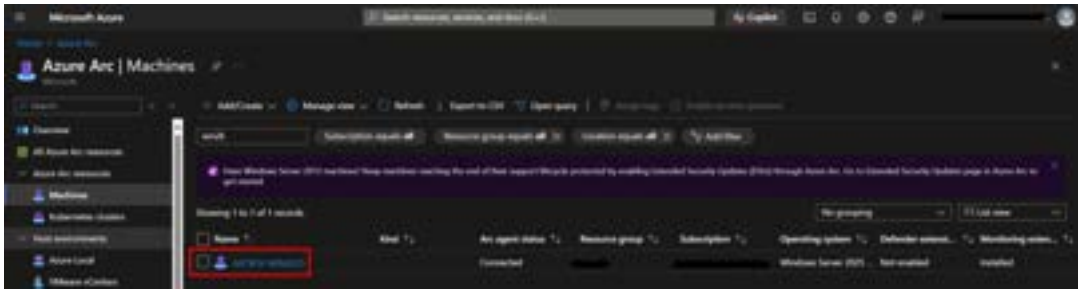
Task 1: Pre-requisites

Pre-requisite 1: Azure permissions

1. [] To install the Windows Admin Center extension for an Arc-enabled server resource, your account must be granted the Owner, Contributor, or Windows Admin Center Administrator Login role in Azure. **Check with your subscription administrator to verify that you have the required roles**.
2. [] Connecting to Windows Admin Center requires you to have Reader and Windows Admin Center Administrator Login permissions at the Arc-enabled server resource. The following steps helps you to set these up.
3. [] Enter "Machines - Azure Arc" in the top search bar in the Azure portal and select it from the displayed services.



4. [] Click on *ArcBox-Win2K25* Arc-enabled Windows server.



5. [] From the selected Windows machine click "Access control (IAM)" then add the role "Admin Center Administrator Login" to your access (use the lab user identity as in the *Resources* tab in the labs environment).

6. [] Follow similar steps to assign yourself Reader permissions at the Arc-enabled server resource.

Pre-requisite 2: Set Windows Server Licensing

1. [] To be able to install the Windows Admin Center extension for an Arc-enabled server resource, you must set the "Windows Server" licensing to either PAYGO (Pay as you go) or attest that you have SA (Software Assurance). In this case we will use the Software Assurance attestation.

[!Important] **Important note about Licensing: The Azure Arc-enabled server must be officially licensed through a valid licensing channel. Unlicensed servers aren't eligible for benefits such as Windows Admin Center**. Refer to this document for further information

Task 1 has been completed

Click **Next** for the next task or **Go back to the main table of content**

===

Task 2: Deploy the Windows Admin Center VM extension

1. [] Open the Azure portal and navigate to the *ArcBox-Win2K25* machine.
2. [] Under the Settings group, select Windows Admin Center, then click "Set up".
3. [] Specify the port on which you wish to install Windows Admin Center, and then select Install.

4. [] If you get the following message after the installation is complete then you need to go back to the previous step and set up the permissions as explained in Pre-requisite.

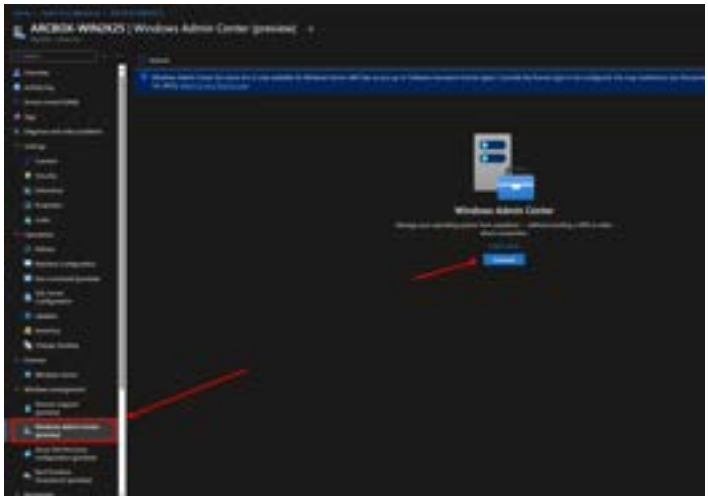
Task 2 has been completed

Click **Next** for the next task or **Go back to the main table of content**

===

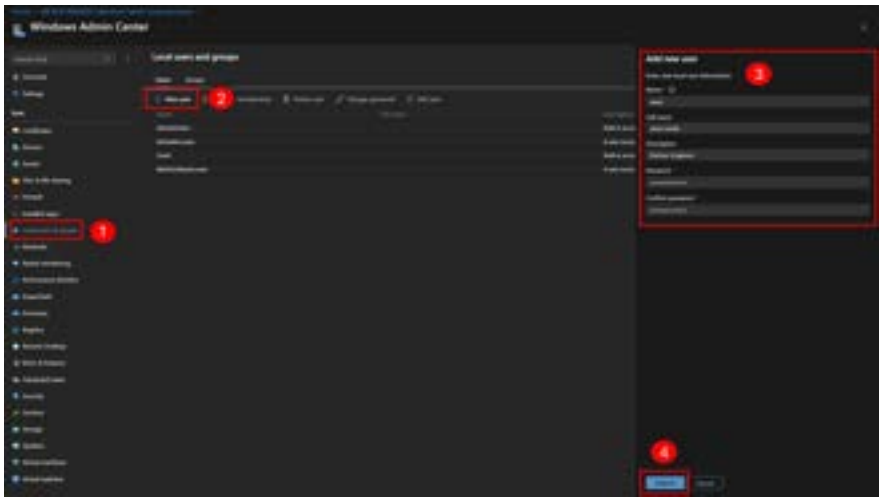
Task 3: Connect and explore Windows Admin Center (preview)

1. [] Once the installation is complete then you can connect to the Windows Admin Center.

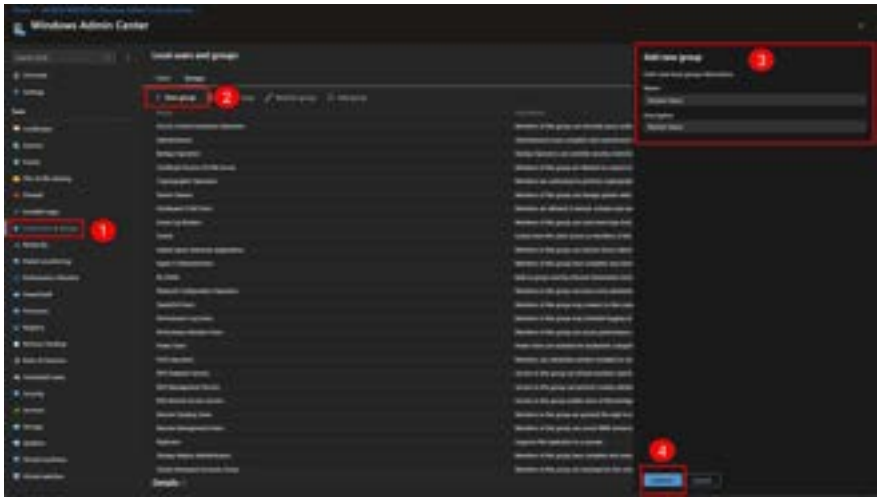


2. [] Start exploring the capabilities offered by the Windows Admin Center to manage your Arc-enabled Windows machine.

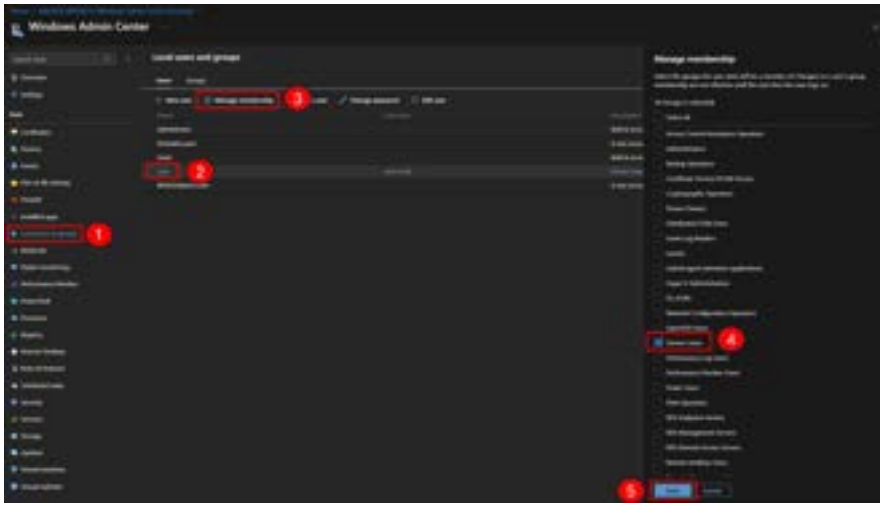
3. [] Let us use the Windows Admin Center to add a local user, a new group and assign the new user to the new group. From the left menu select "Local users & groups". Then from the "Users" tab click "New user". Enter the user details and click on "Submit". Verify that the user has been added.



4. [] Now select the "Groups" tab and click on "New Group". Enter the group details and click on "Submit". Verify that the group has been added.



5. [] Back to the "Users" tab, select the new user you have added, then click "Manage membership". Add the selected user to the new group and save.



Task 3 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

LAB10: Query and inventory your Azure Arc-enabled servers using Azure Resource Graph

---

Student Lab Manual

Table of Contents

Exercise 1 - Use Resource Graph Queries in the Azure Portal and in Powershell to examine your Arc-enabled servers

---

Objective

In this exercise, you will learn how to use the Azure Resource queries both in the Azure Graph Explorer and Powershell to demonstrate inventory management of your Azure Arc connected servers. Note that the results you get by running the graph queries in this module might be different from the sample screenshots as your environment might be different e.g. as a result of working with the other labs.

Estimated Time to Complete This Lab

30 minutes

Explanation

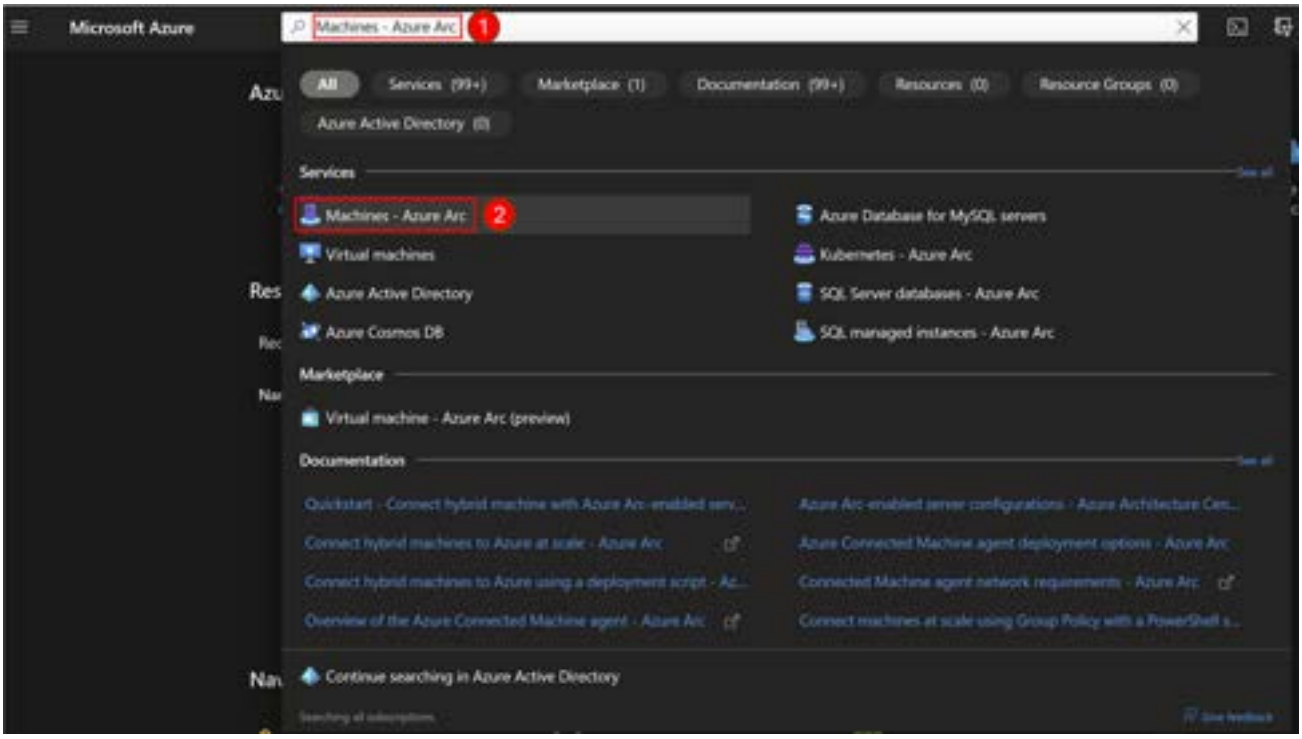
Azure Resource Graph is an extremely powerful extension to Azure Resource Management that provides efficient resource exploration at scale. It also provides the ability to do complex filtering and grouping. It can do this because it uses a subset of the Kusto Query Language (KQL).

There are a number of tables you can query in Azure Resource Graph. The most common table is the "resources" table. This is where all resources in your Azure subscriptions will live. With few exceptions everything in Azure is a resource. Queries can be run against the Azure Resource Graph API, with PowerShell, or in the Azure portal.

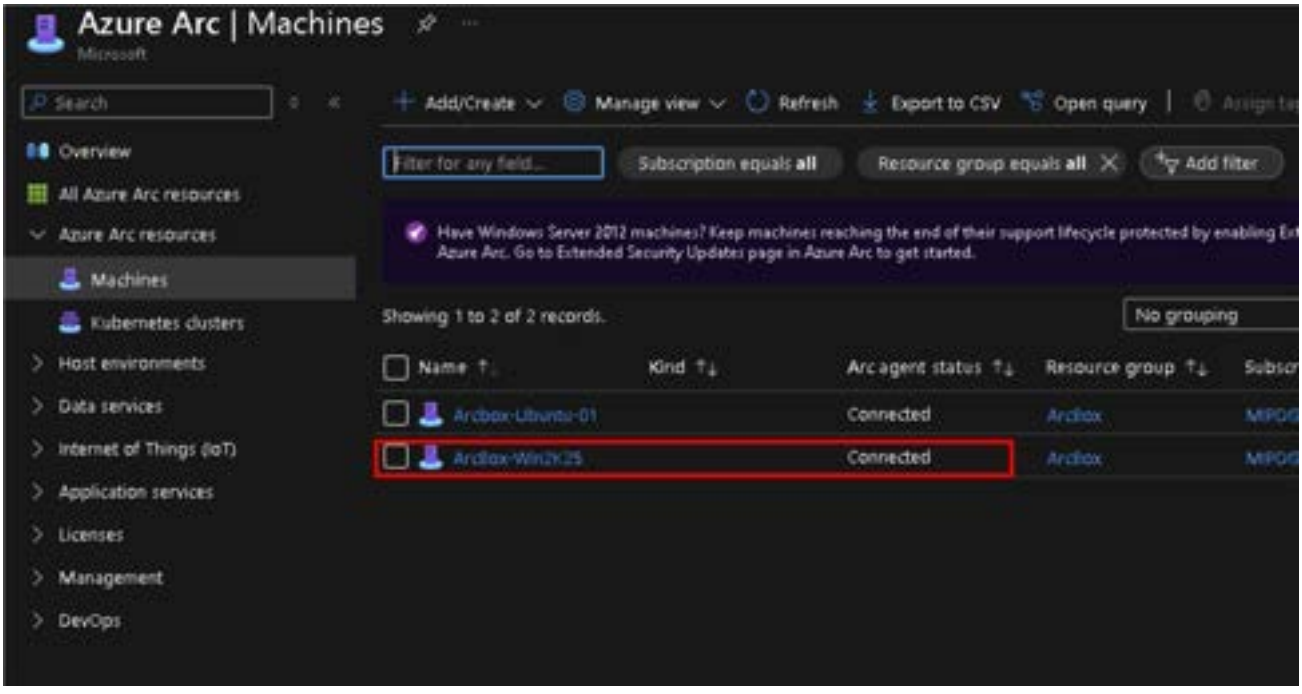
Task 1: Apply resource tags to Azure Arc-enabled servers

In this first step, you will assign Azure resource tags to some of your Azure Arc-enabled servers. This gives you the ability to easily organize and manage server inventory.

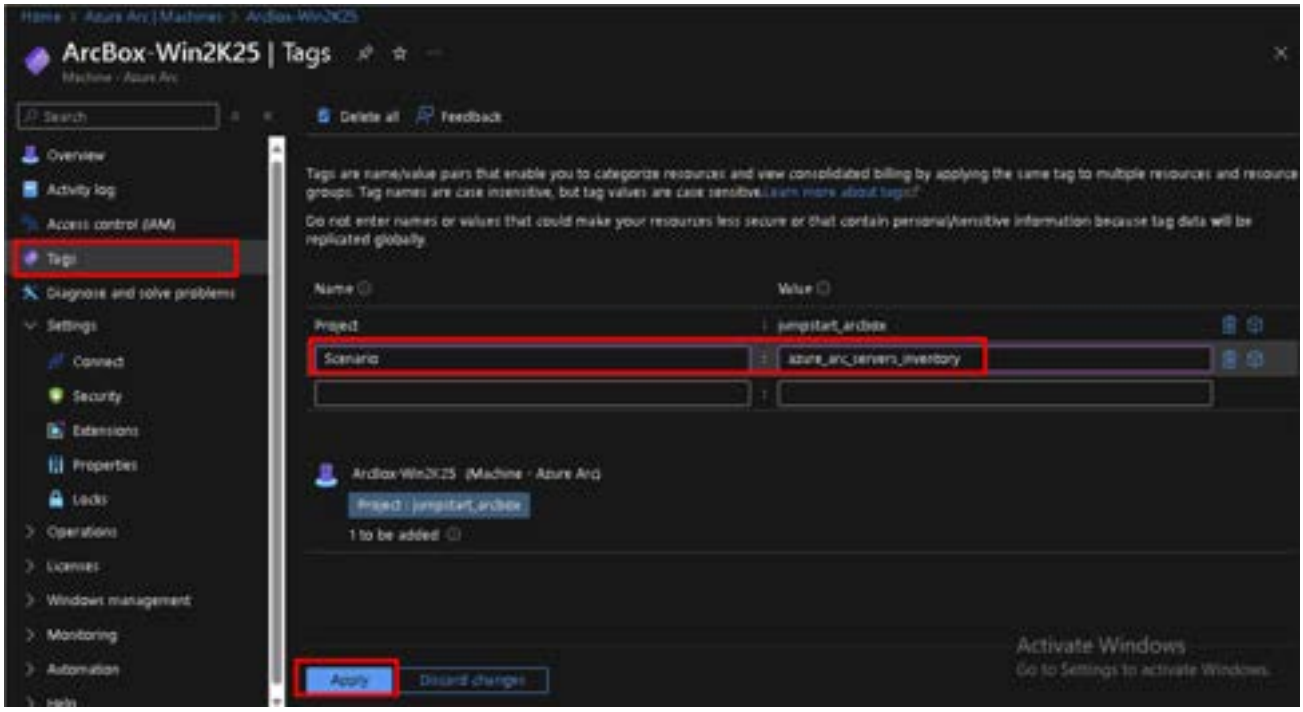
☐ 1. Enter "Machines - Azure Arc" in the top search bar in the Azure portal and select it from the displayed services.



☐ 2. Click on any of your Azure Arc-enabled servers.



3. Click on "Tags". Add a new tag with Name="Scenario" and Value="azure_arc_servers_inventory". Click Apply when ready.



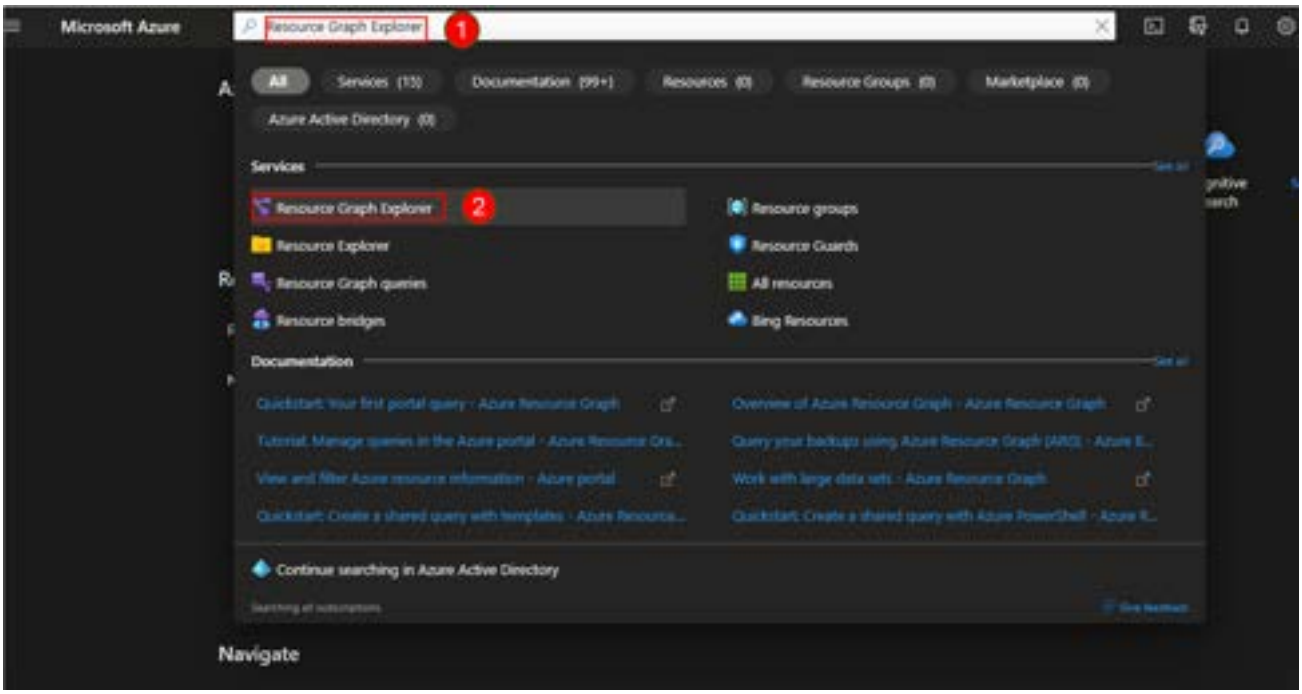
4. Repeat the same process in other Azure Arc-enabled servers if you wish. This new tag will be used later when working with Resource Graph Explorer queries.

Task 1 has been completed

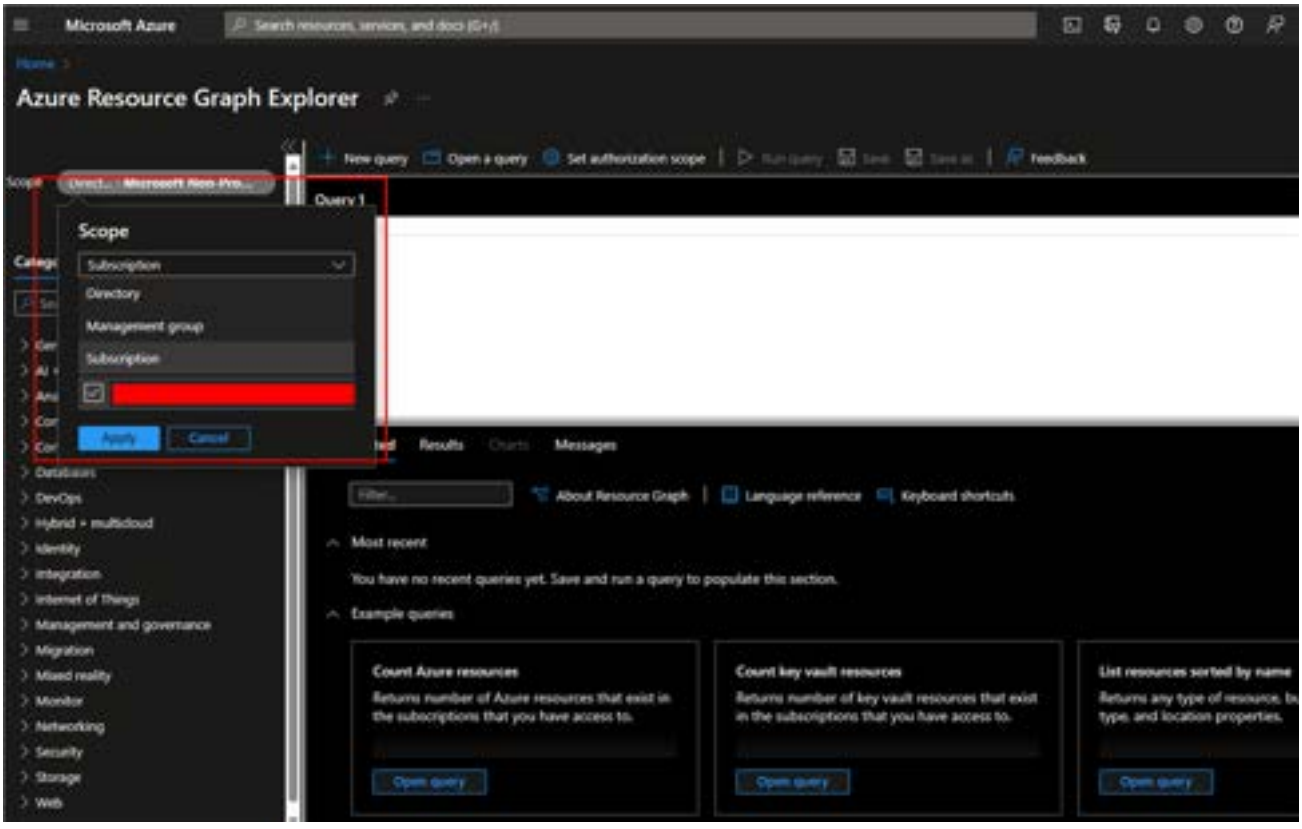
Click **Next** for the next task or **Go back to the main table of content**

Task 2: The Azure Resource Graph Explorer

1. Now we will explore our hybrid server inventory using a number of Azure Graph Queries. Enter "Resource Graph Explorer" in the top search bar in the Azure portal and select it.



2. The scope of the Resource Graph Explorer can be set as seen below



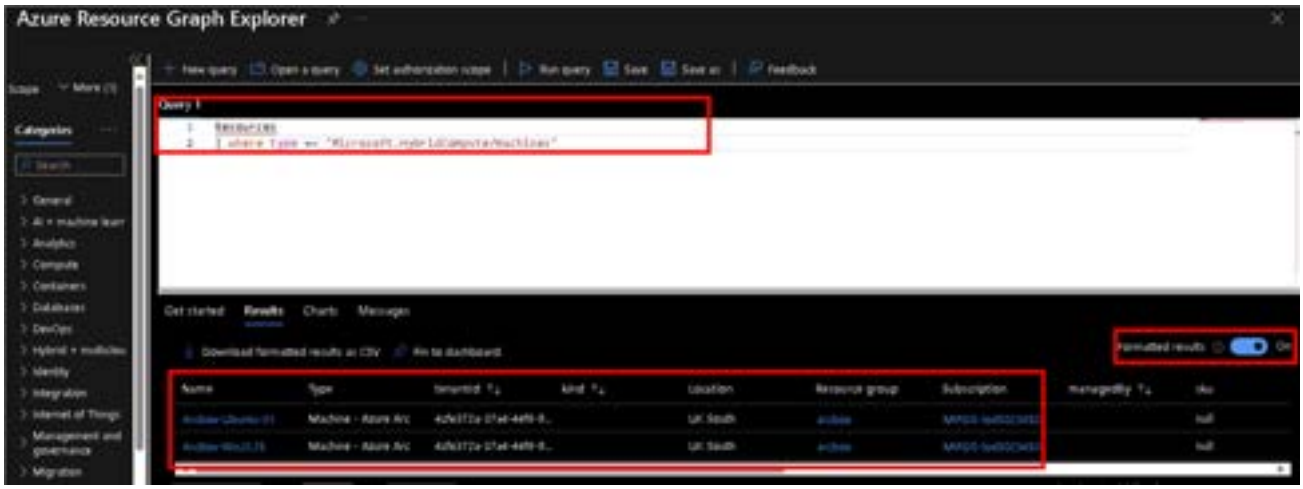
Task 2 has been completed

Click **Next** for the next task or **Go back to the main table of content**

Task 3: Run a query to show all Azure Arc-enabled servers in your subscription

- [ ] 1. In the query window, enter and run the following query and examine the results which should show your Arc-enabled servers. Note the use of the KQL equals operator (=~) which is case insensitive KQL =~ (equals) operator.

```shell
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
```



- [ ] 2. Scroll to the right on the results pane and click "See Details" to see all the Azure Arc-enabled server metadata. Note for example the list of detected properties, we will be using these in the next task.

- [ ] 3. You can also run the same query using PowerShell (e.g. using Azure Cloud Shell) providing that you have added the required module "Az.ResourceGraph" as explained in Run your first Resource Graph query using Azure PowerShell.

To install the PowerShell module, run the following command

```powershell
Install-Module -Name Az.ResourceGraph
```

Then run the query in PowerShell

```powershell
Search-AzGraph -Query "Resources | where type =~ 'Microsoft.HybridCompute/machi
```

Task 3 has been completed

Click **Next** for the next task or **Go back to the main table of content**

Task 4: Query your server inventory using the available metadata

- [ ] 1. Use PowerShell and/or the Resource Graph Explorer to summarize the server count by "logical cores" which is one of the detected properties referred to in the previous task.

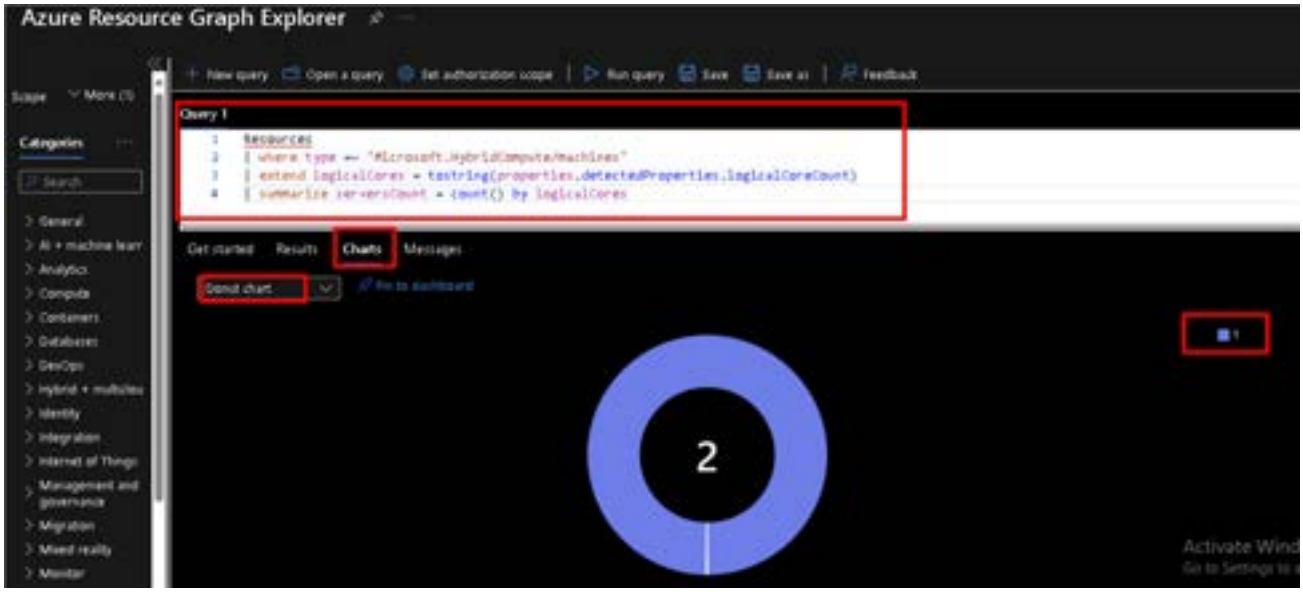
Using the Portal:

```portal
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend logicalCores = tostring(properties.detectedProperties.logicalCoreCount)
| summarize serversCount = count() by logicalCores
```

Using PowerShell:

```powershell
Search-AzGraph -Query  "Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend logicalCores = tostring(properties.detectedProperties.logicalCoreCount)
| summarize serversCount = count() by logicalCores"
```

- [ ] 2. The Graph Explorer allows you to get a graphical view of your results by selecting the "charts" option.



Task 4 has been completed

Click **Next** for the next task or **Go back to the main table of content**

Task 5: Use the resource tags in your Graph Query

1. Let's now build a query that uses the tag we assigned earlier to some of our Azure Arc-enabled servers. Use the following query that includes a check for resources that have a value for the "Scenario" tag. Feel free to use the portal or PowerShell. Check that the results match the servers that you set tags for earlier.

Using the Portal:

```portal
Resources
| where type =~ 'Microsoft.HybridCompute/machines' and isnotempty(tags['Scenario
| extend Scenario = tags['Scenario']
| project name, tags
```

Using PowerShell:

```powershell
Search-AzGraph -Query  "Resources
| where type =~ 'Microsoft.HybridCompute/machines' and isnotempty(tags['Scenario
| extend Scenario = tags['Scenario']
| project name, tags"
```

Task 5 has been completed

Click **Next** for the next task or **Go back to the main table of content**

Task 6: List the extensions installed on the Azure Arc-enabled servers

1. Run the following advanced query which allows you to see what extensions are installed on the Arc-enabled servers. Notice that running the query in PowerShell requires us to escape the $ character as explained in Escape Characters.

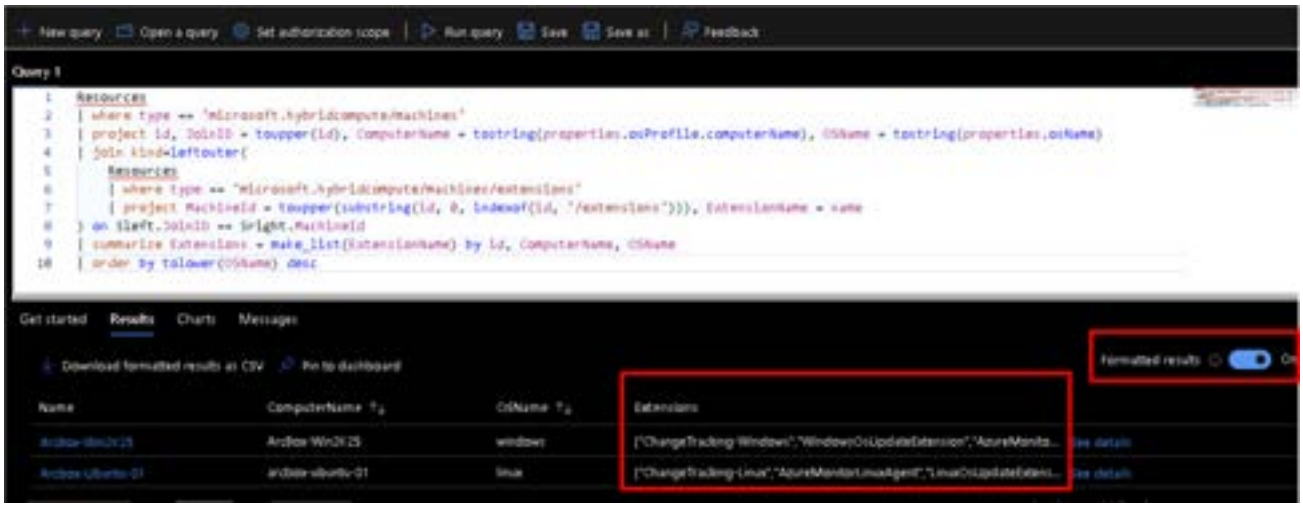
Using the Portal:

```portal
Resources
| where type == 'microsoft.hybridcompute/machines'
| project id, JoinID = toupper(id), ComputerName = tostring(properties.osProfile
| join kind=leftouter(
    Resources
    | where type == 'microsoft.hybridcompute/machines/extensions'
    | project MachineId = toupper(substring(id, 0, indexof(id, '/extensions'))),
) on $left.JoinID == $right.MachineId
| summarize Extensions = make_list(ExtensionName) by id, ComputerName, OSName
| order by tolower(OSName) desc
```

Using PowerShell:

```powershell
Search-AzGraph -Query "Resources
| where type == 'microsoft.hybridcompute/machines'
| project id, JoinID = toupper(id), ComputerName = tostring(properties.osProfile
| join kind=leftouter(
    Resources
    | where type == 'microsoft.hybridcompute/machines/extensions'
    | project MachineId = toupper(substring(id, 0, indexof(id, '/extensions'))),
) on `$left.JoinID == `$right.MachineId
| summarize Extensions = make_list(ExtensionName) by id, ComputerName, OSName
| order by tolower(OSName) desc"
```

2. If you have used the portal to run the query then you should see results similar to those shown in the screenshot below.



Task 6 has been completed

Click **Next** for the next task or **Go back to the main table of content**

Task 7: Query other properties

- [ ] 1. Azure Arc provides additional properties on the Azure Arc-enabled server resource that we can query with Resource Graph Explorer. In the following example, we list some of these key properties, like the Azure Arc Agent version installed on your Azure Arc-enabled servers

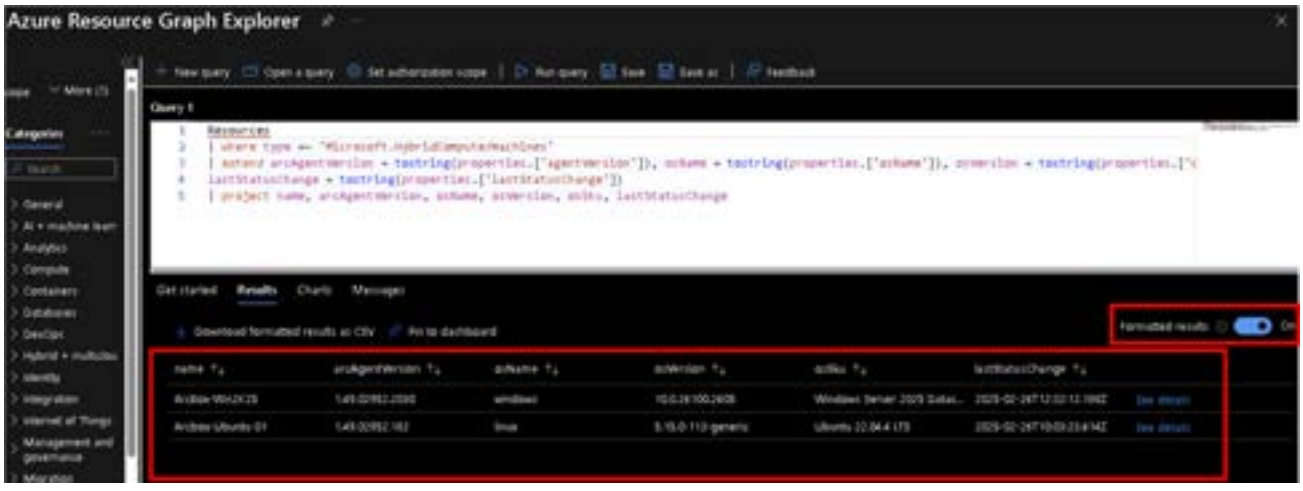
Using the Portal

```portal
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend arcAgentVersion = tostring(properties.['agentVersion']), osName = tostr
lastStatusChange = tostring(properties.['lastStatusChange'])
| project name, arcAgentVersion, osName, osVersion, osSku, lastStatusChange
```

Using PowerShell

```powershell
Search-AzGraph -Query  "Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend arcAgentVersion = tostring(properties.['agentVersion']), osName = tostr
lastStatusChange = tostring(properties.['lastStatusChange'])
| project name, arcAgentVersion, osName, osVersion, osSku, lastStatusChange"
```

- [ ] 2. Running the query in the portal should result in something like the following



Task 7 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or **Go back to the main table of content**

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

---

Objective

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

Estimated Time to Complete This Lab

20 minutes

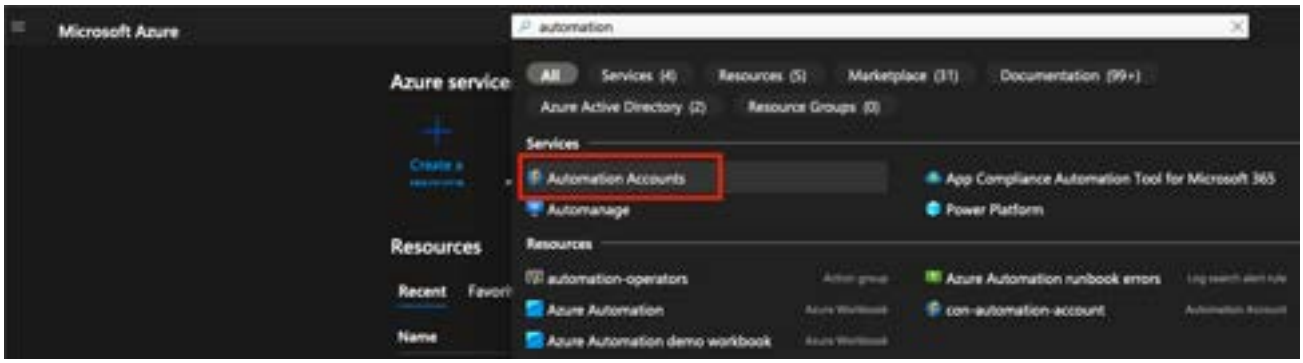
Explanation

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.

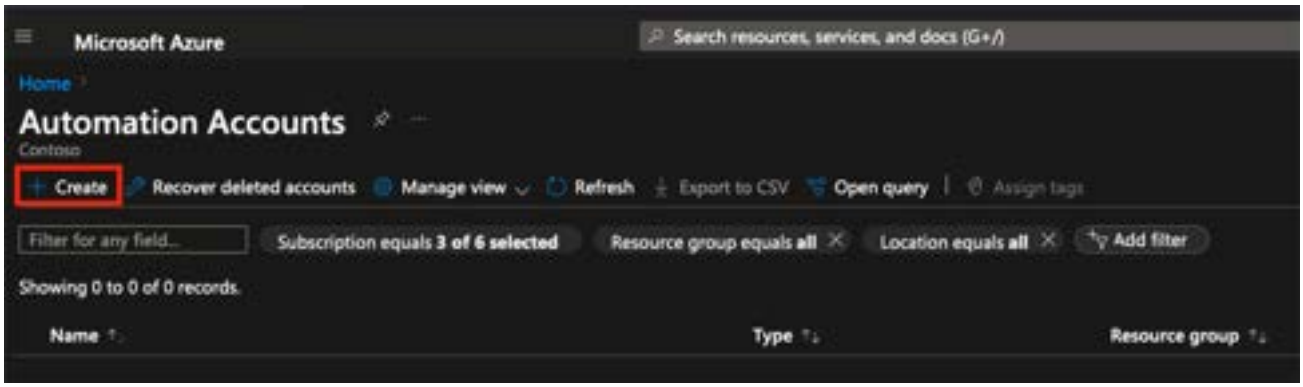
Task 1: Create Automation account using the Azure Portal

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*



2. Click on "Create"

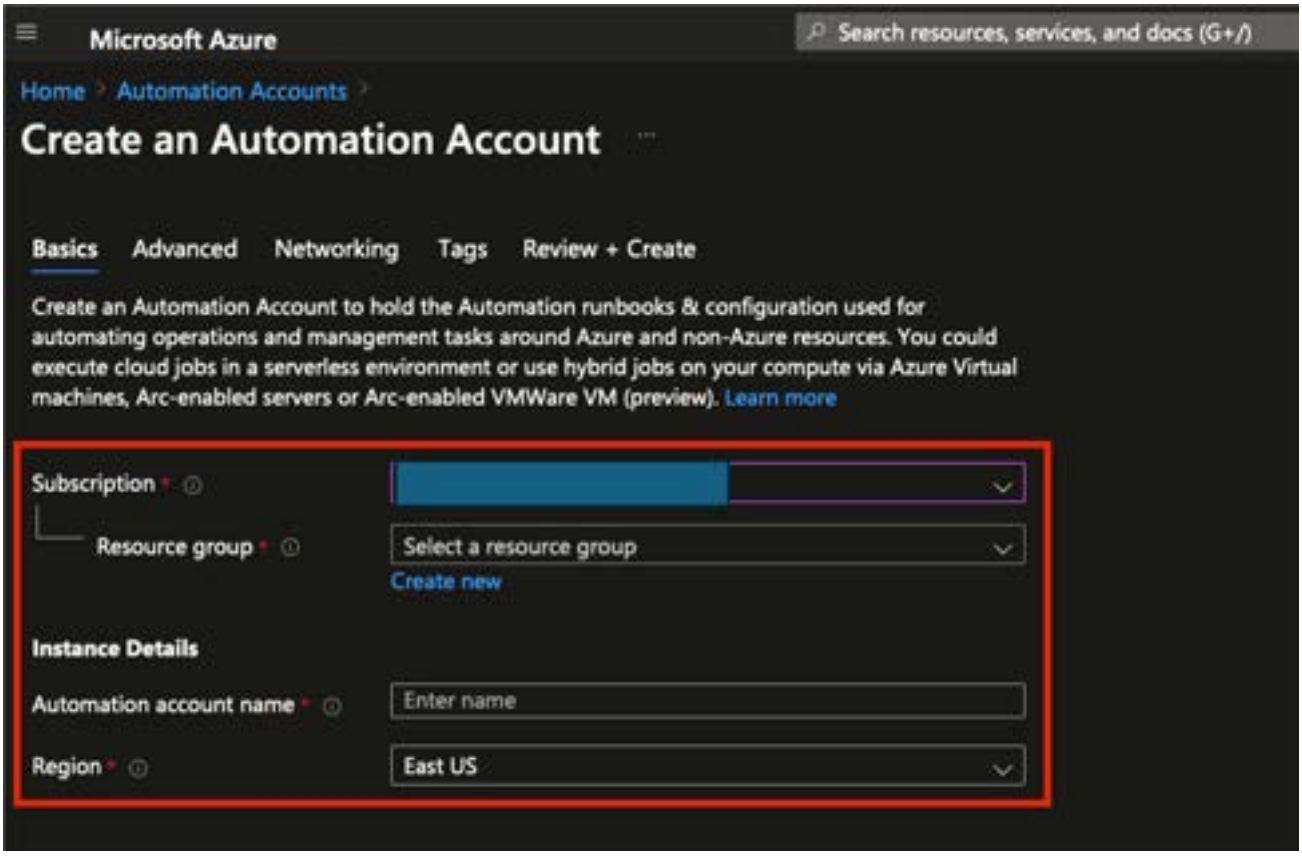


3. Select the subscription and resource group where you have deployed ArcBox.

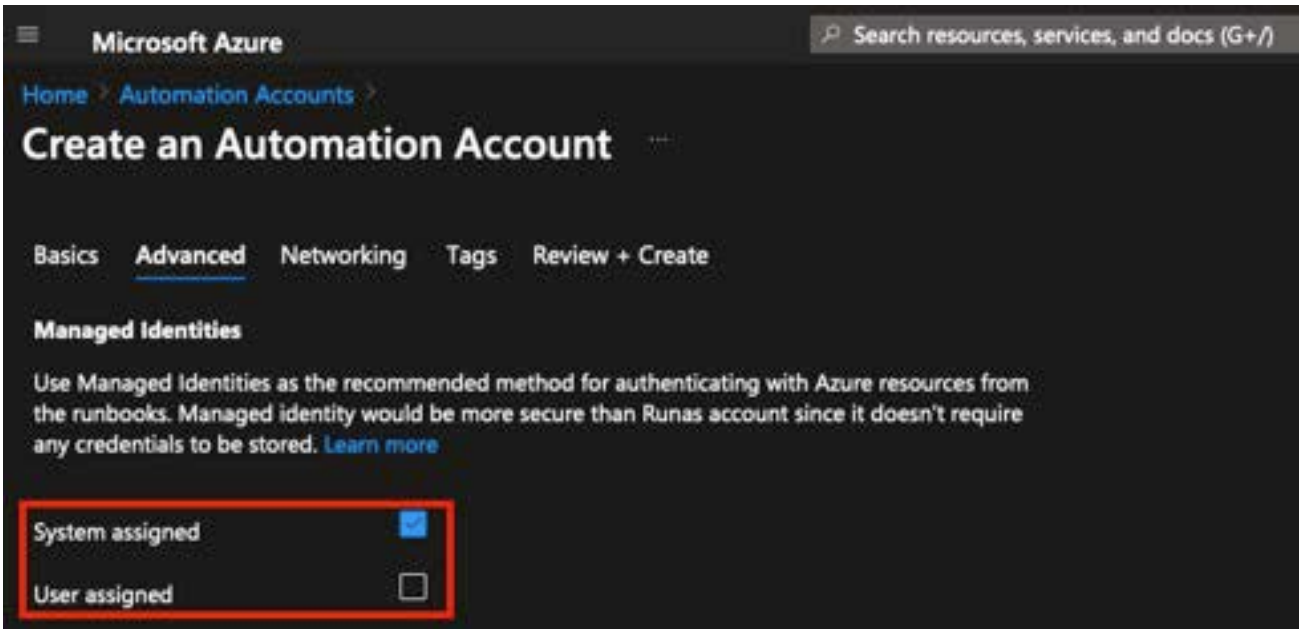
4. Enter *ArcBox-Automation* as the name for the Automation Account.

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

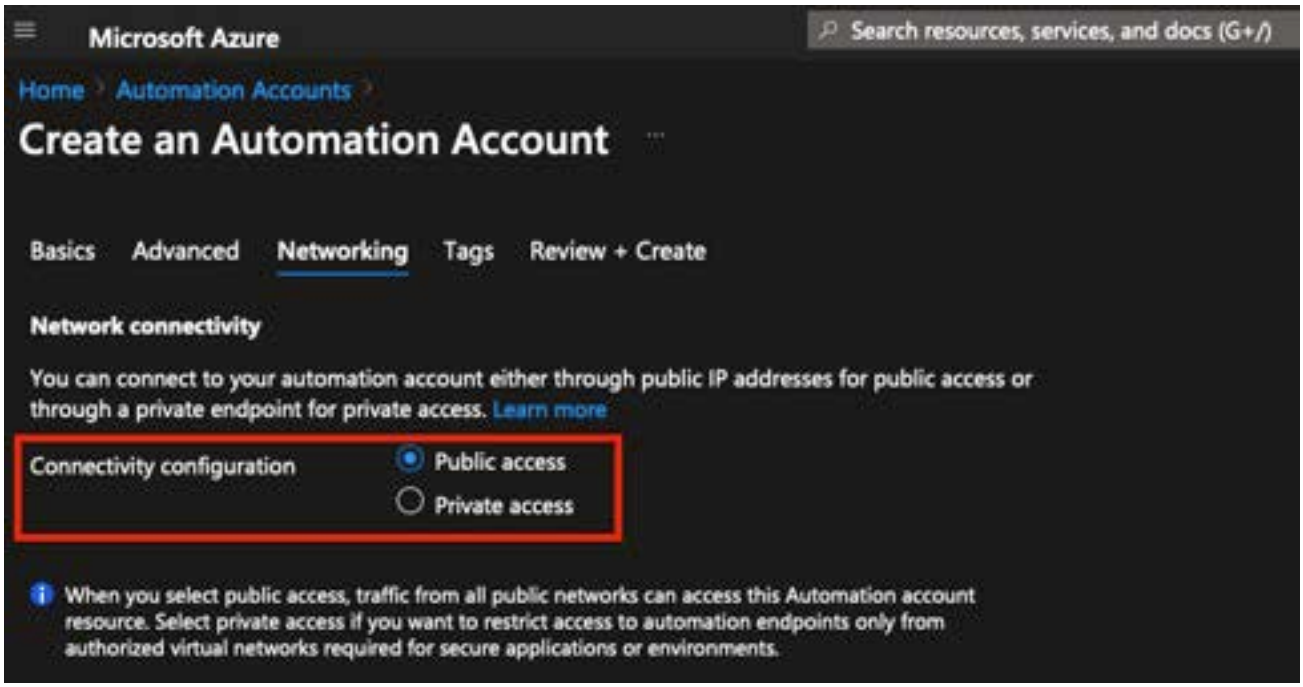
6. Click Next



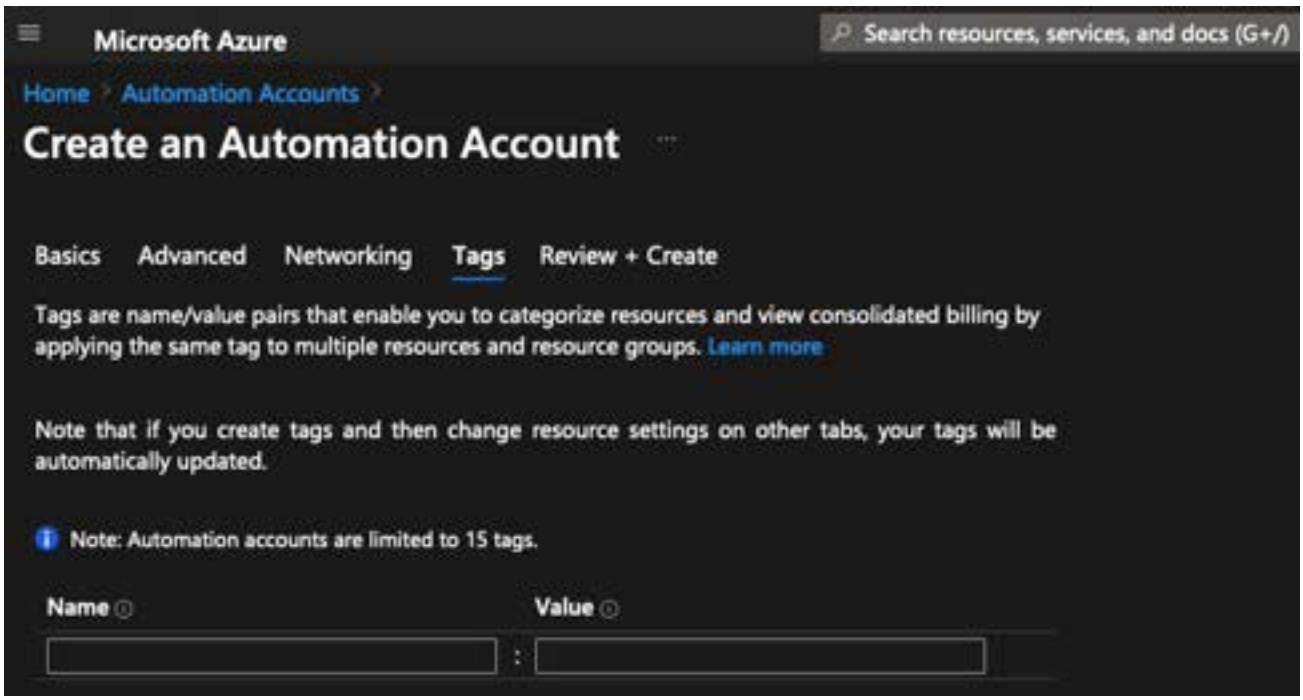
☐ 7. Leave the default settings for *Managed Identities* in place and click Next



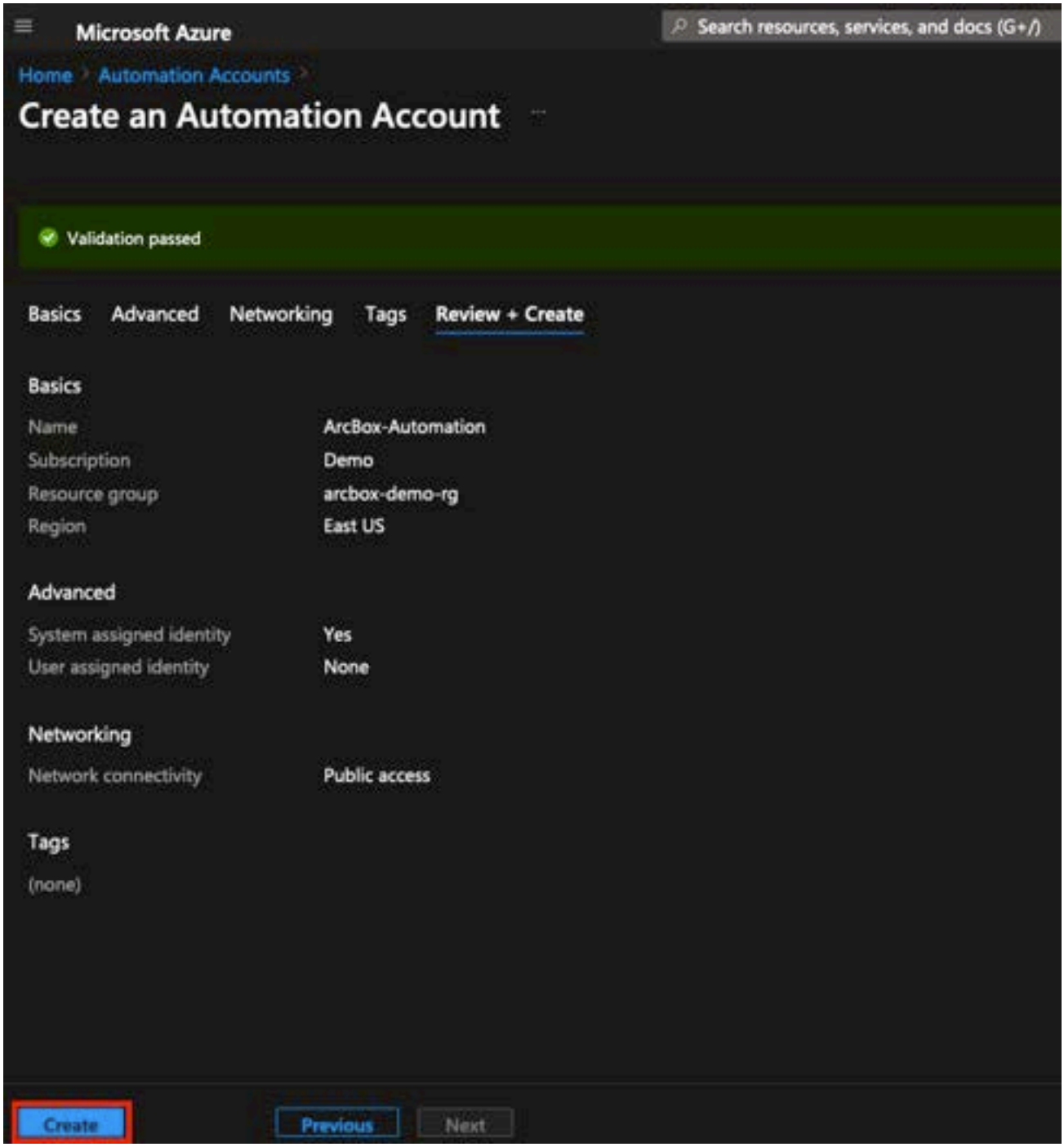
☐ 8. Leave the default settings for *Connectivity configuration* in place and click Next



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



☐ 10. Click Create



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

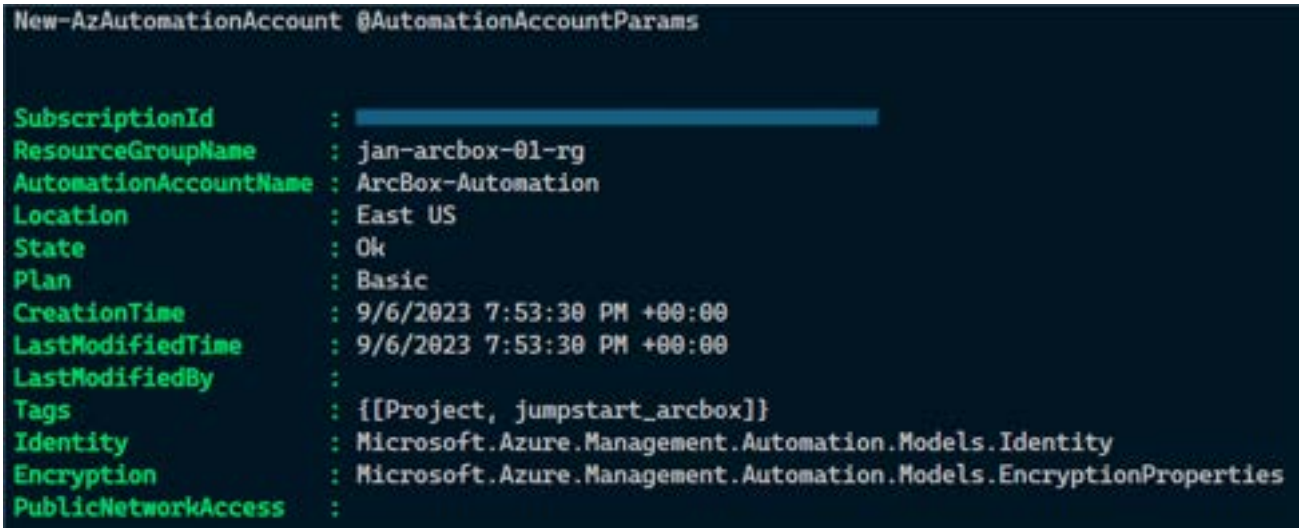
> ☐ This task is an alternative to, and only needed if you have not executed **task 1**. You can jump straight to **task 3** if you have already provisioned an Automation account using the Azure portal.

☐ 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.

```powershell
New-AzAutomationAccount -Location "<Your chosen Location>" -Name "ArcBox-Au
```

The output should look similar to this:



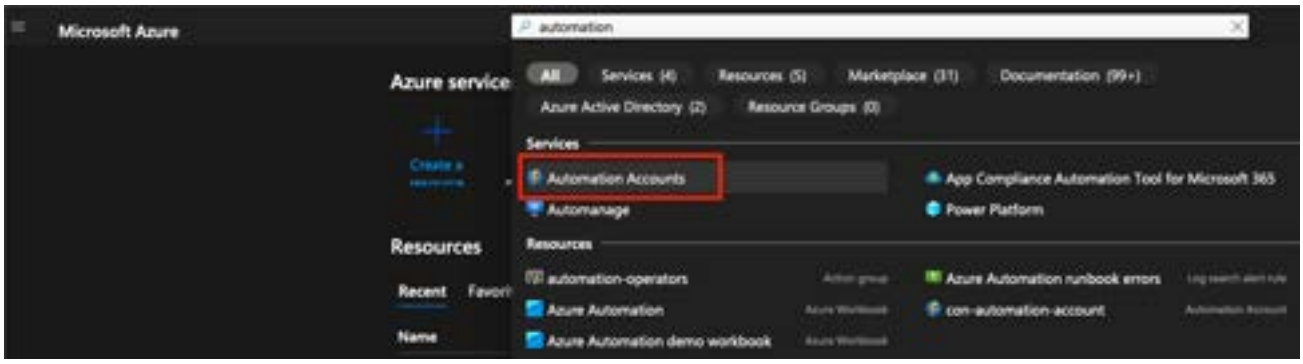
Task 2 has been completed

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

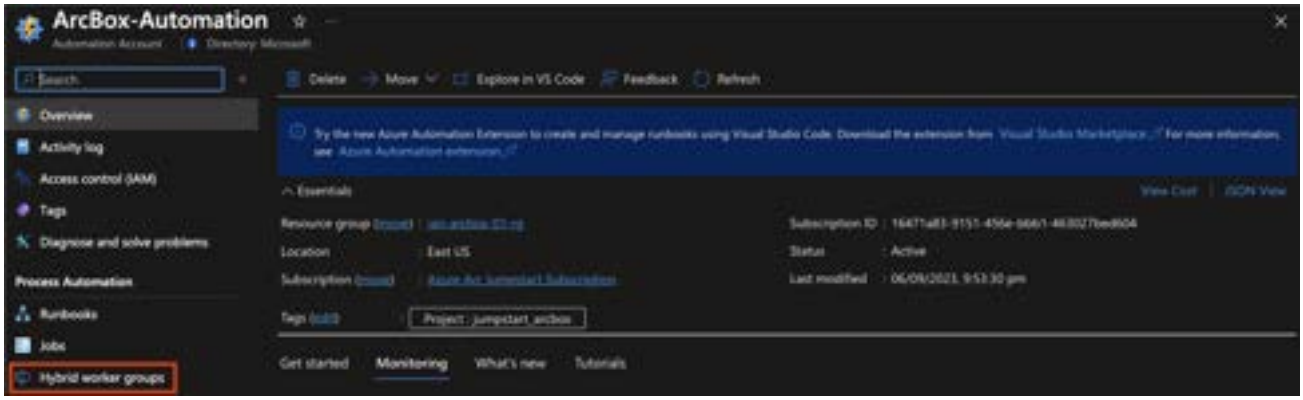
> You have two options to add hybrid runbook workers. You can either follow the steps in this task to use the Azure Portal, or if you prefer to use Powershell then jump to Task **task 4** and continue from there.

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

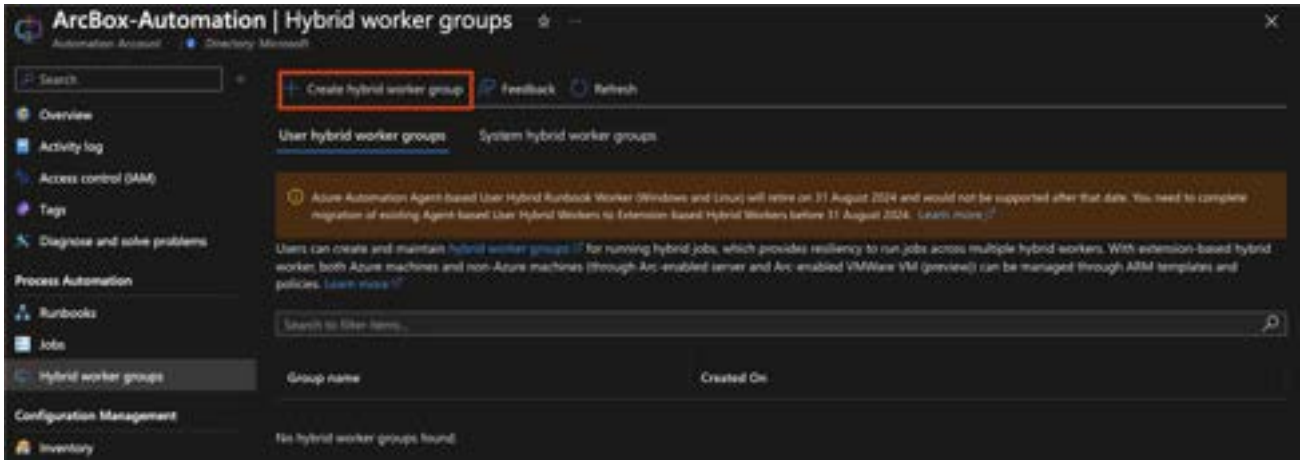


2. Navigate to the *ArcBox-Automation* account you created previously

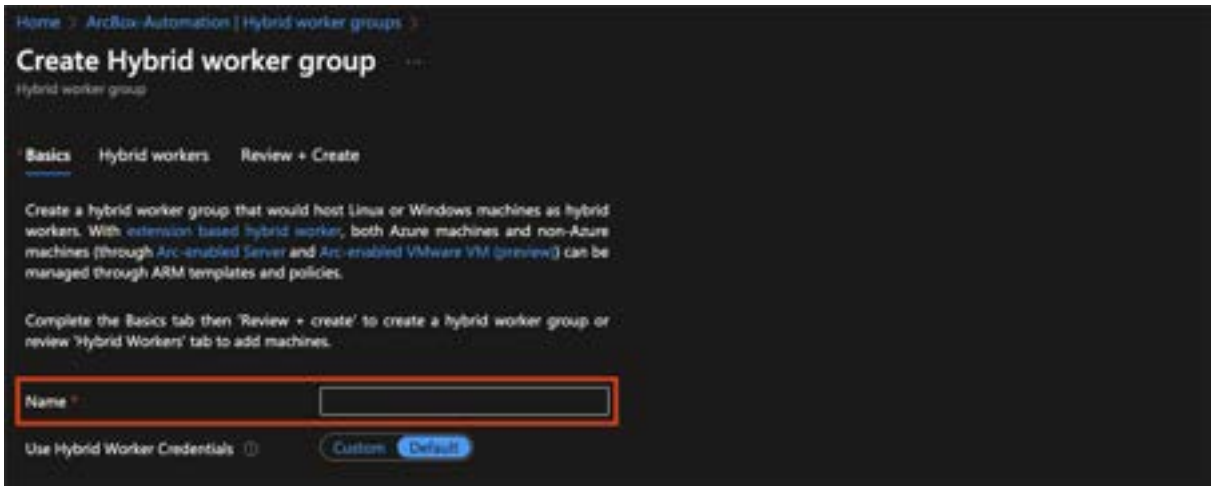
3. Select *Hybrid worker groups*:



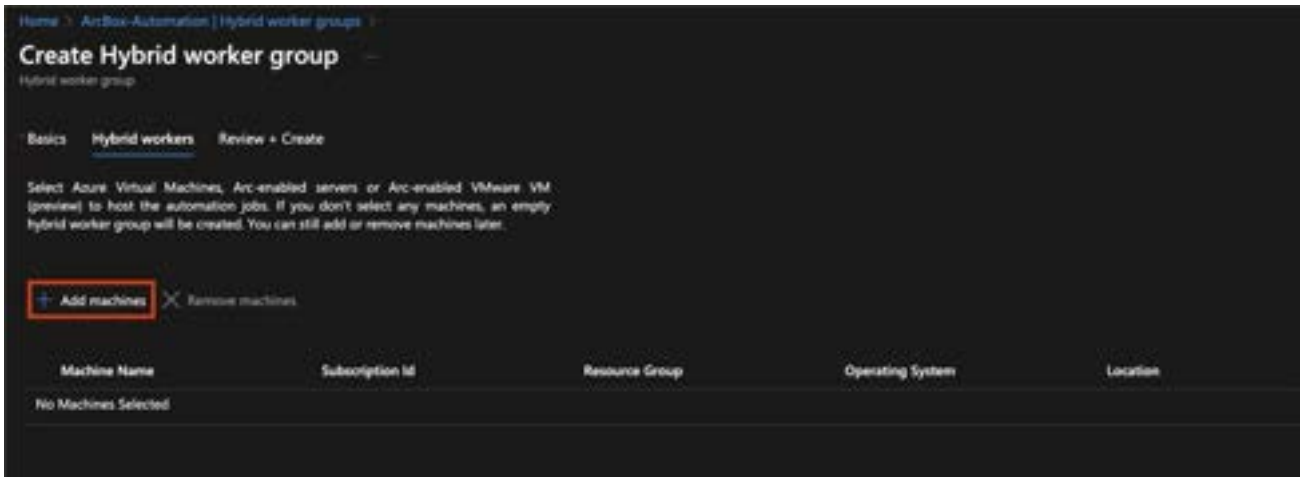
4. Click *Create hybrid worker group*:



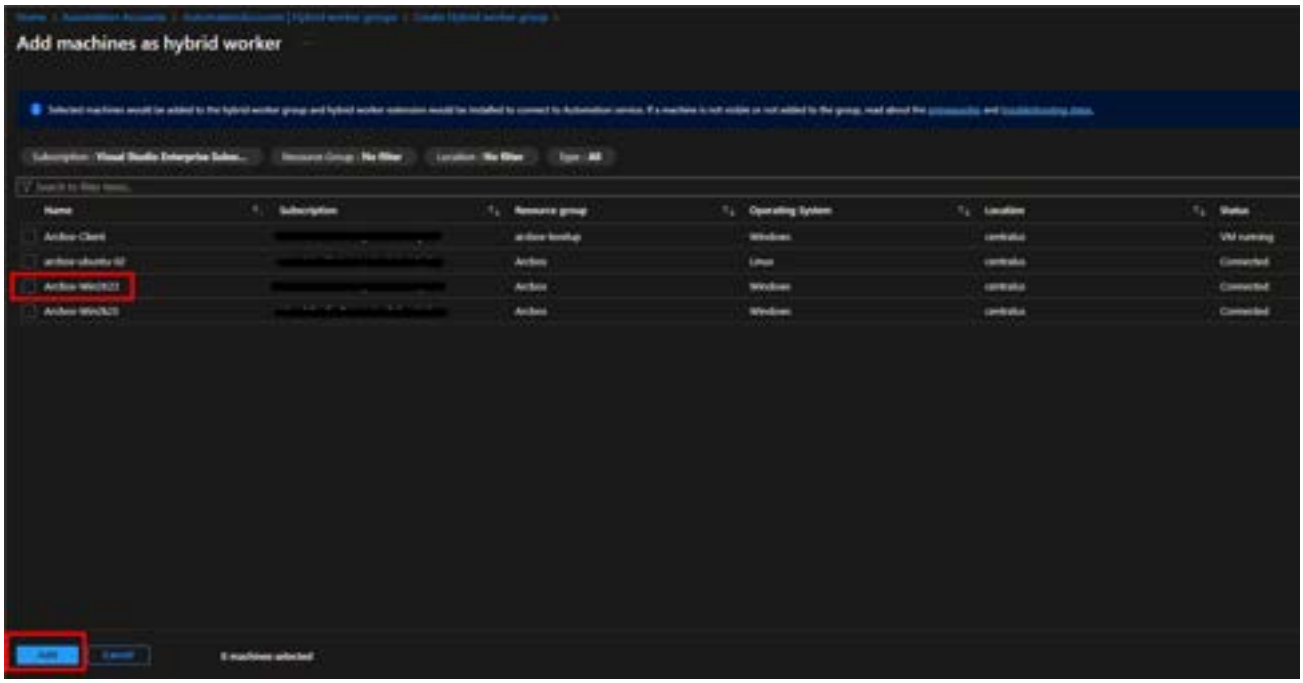
5. Type *windows-workers* as the name of the new Hybrid worker group, leave the default value for *Use Hybrid Worker Credentials* and click Next



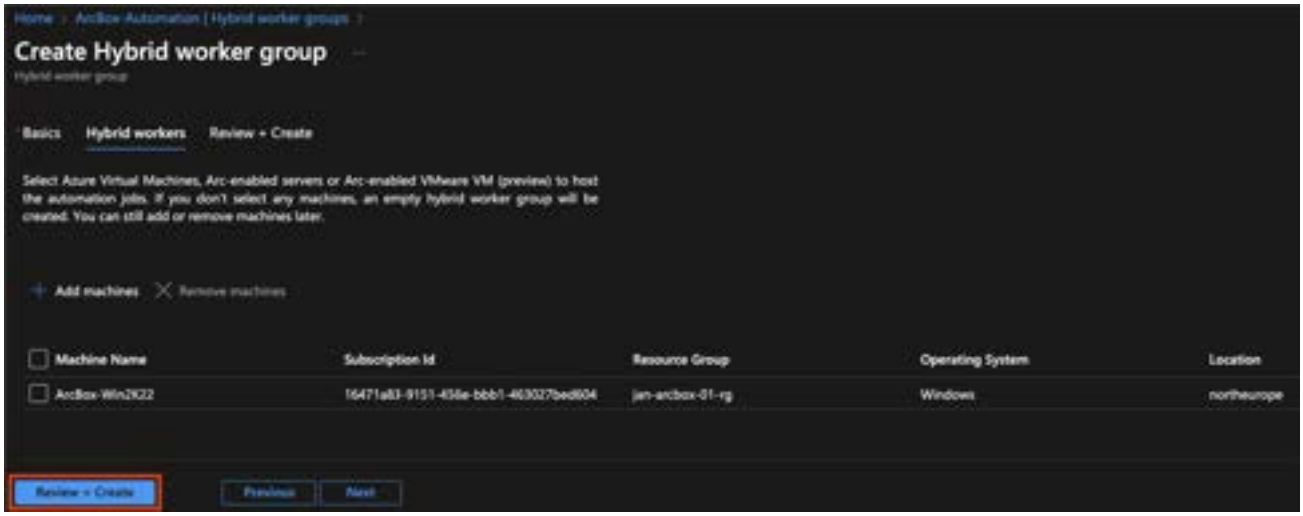
6. Click *Add machines*:



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

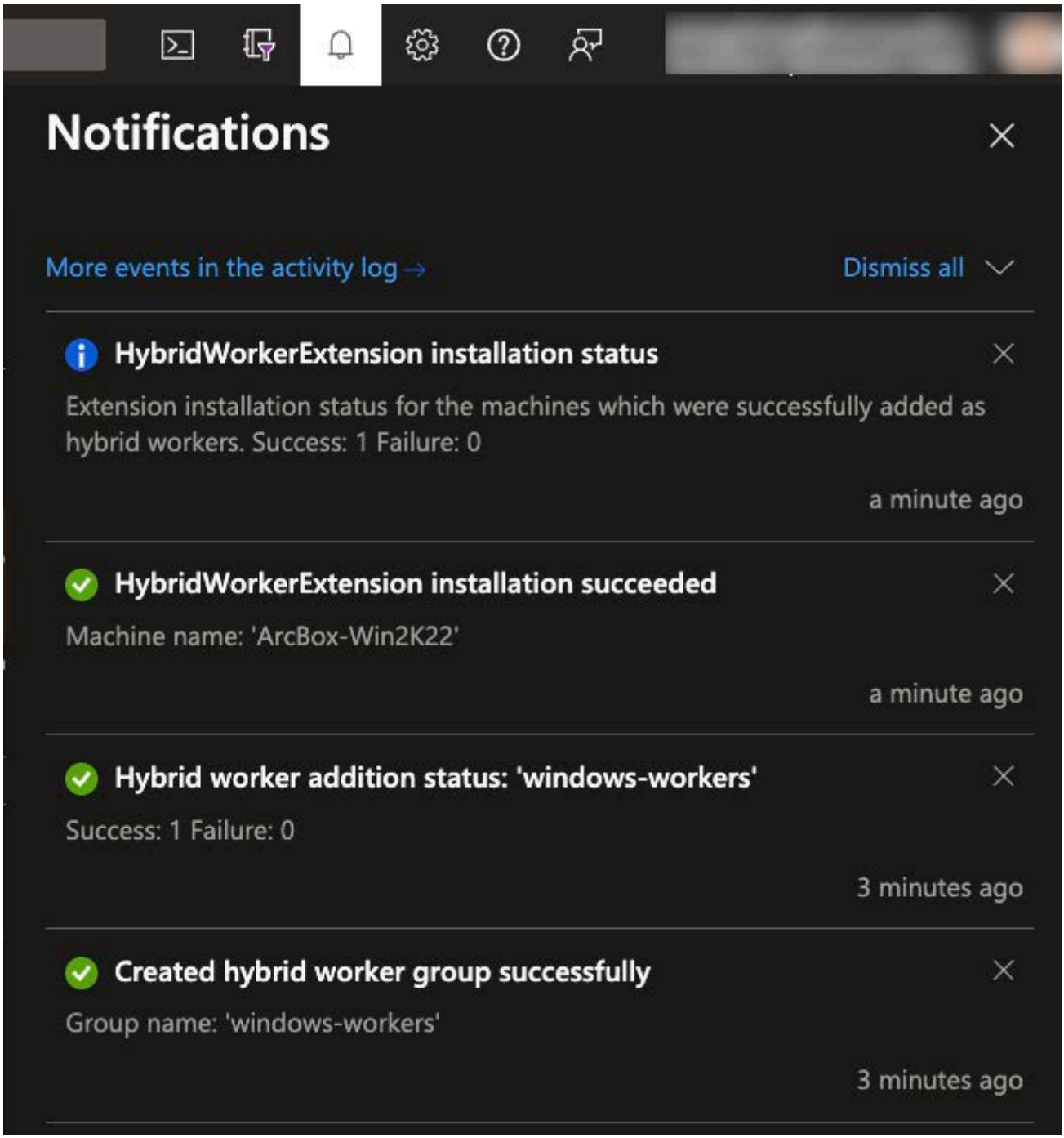


☐ 8. Click *Review + Create*:



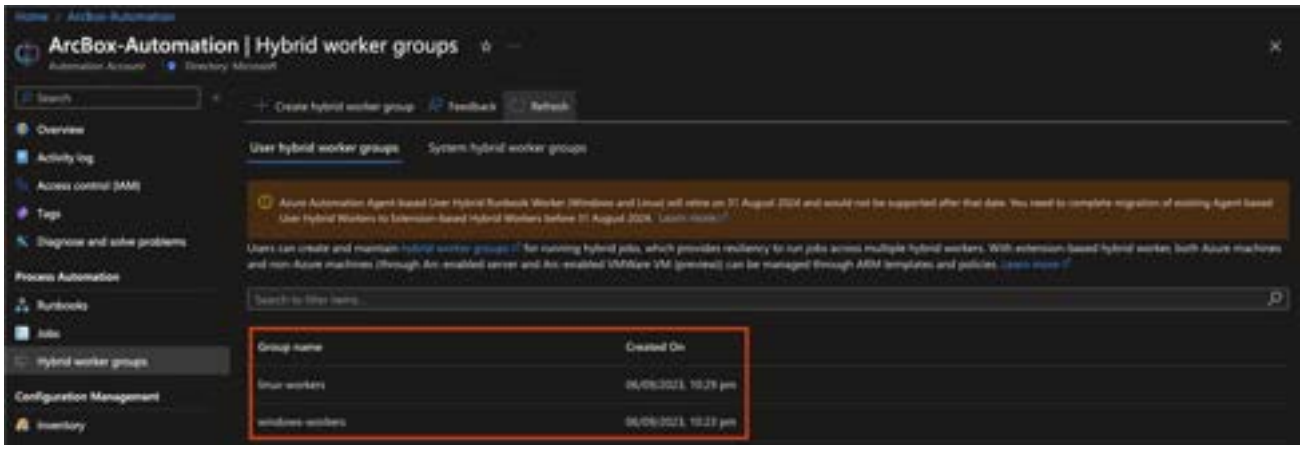
☐ 9. Click *Create*:

☐ 10. Wait for the following activities to be finished:



☐ 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:



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

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.

```powershell
# Retrieve service URL for Automation account (used when registering Arc-enabled
$AutomationAccount = Get-AzResource -ResourceGroupName "ArcBox" -Name "ArcBox-Au
$AutomationAccountInfo = Invoke-AzRestMethod -SubscriptionId $AutomationAccount.
$AutomationHybridServiceUrl = ($AutomationAccountInfo.Content | ConvertFrom-Json

# Create the Linux Hybrid Worker Group
New-AzAutomationHybridRunbookWorkerGroup -Name linux-workers -ResourceGroupName /

#Get the Arc-enabled Linux VM and add to the Hybrid Worker to the group
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-Ubu
New-AzAutomationHybridRunbookWorker -ResourceGroupName ArcBox -AutomationAccount
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-Ubu
$settings = @{
      "AutomationAccountURL"  = $AutomationHybridServiceUrl
  }
New-AzConnectedMachineExtension -ResourceGroupName $ArcResource.ResourceGroupNam

# Create the Windows Hybrid Worker Group
New-AzAutomationHybridRunbookWorkerGroup -Name windows-workers -ResourceGroupNam

#Get the Arc-enabled Windows VM and add to the Hybrid Worker to the group
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name ArcBox-Win
New-AzAutomationHybridRunbookWorker -ResourceGroupName ArcBox -AutomationAccount
$ArcResource = Get-AzConnectedMachine -ResourceGroupName ArcBox -Name Arcbox-Win
$settings = @{
      "AutomationAccountURL"  = $AutomationHybridServiceUrl
  }
New-AzConnectedMachineExtension -ResourceGroupName $ArcResource.ResourceGroupNam
```

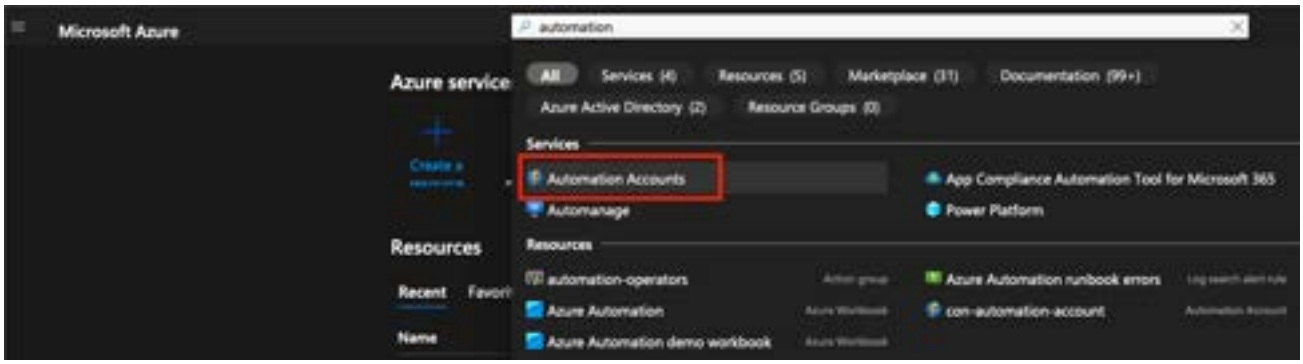
3. It will take few minutes to execute the script.

Task 4 has been completed

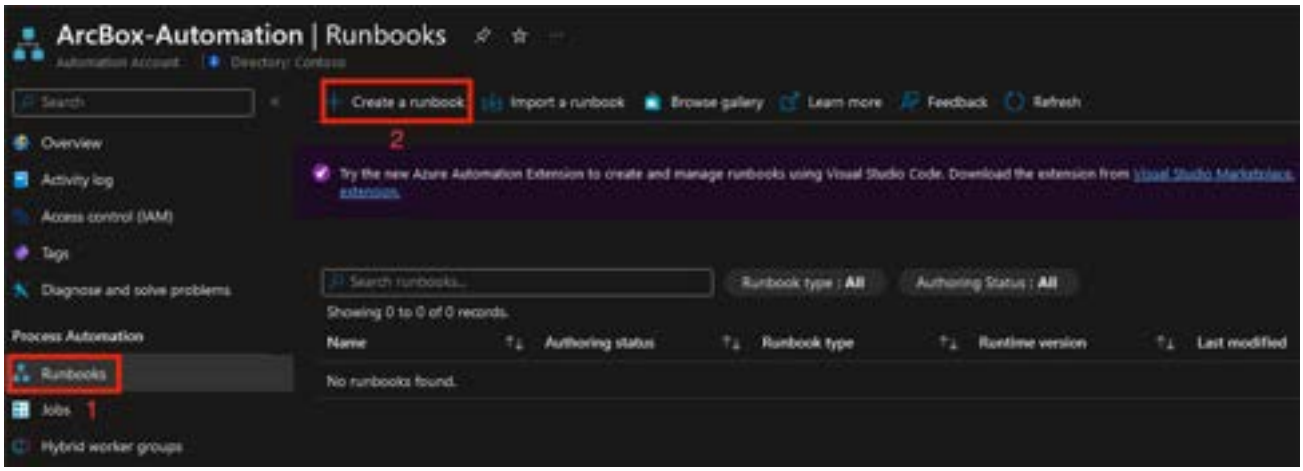
Click **Next** for the next task or **go back to the main table of content**

Task 5 - Create and start a runbook

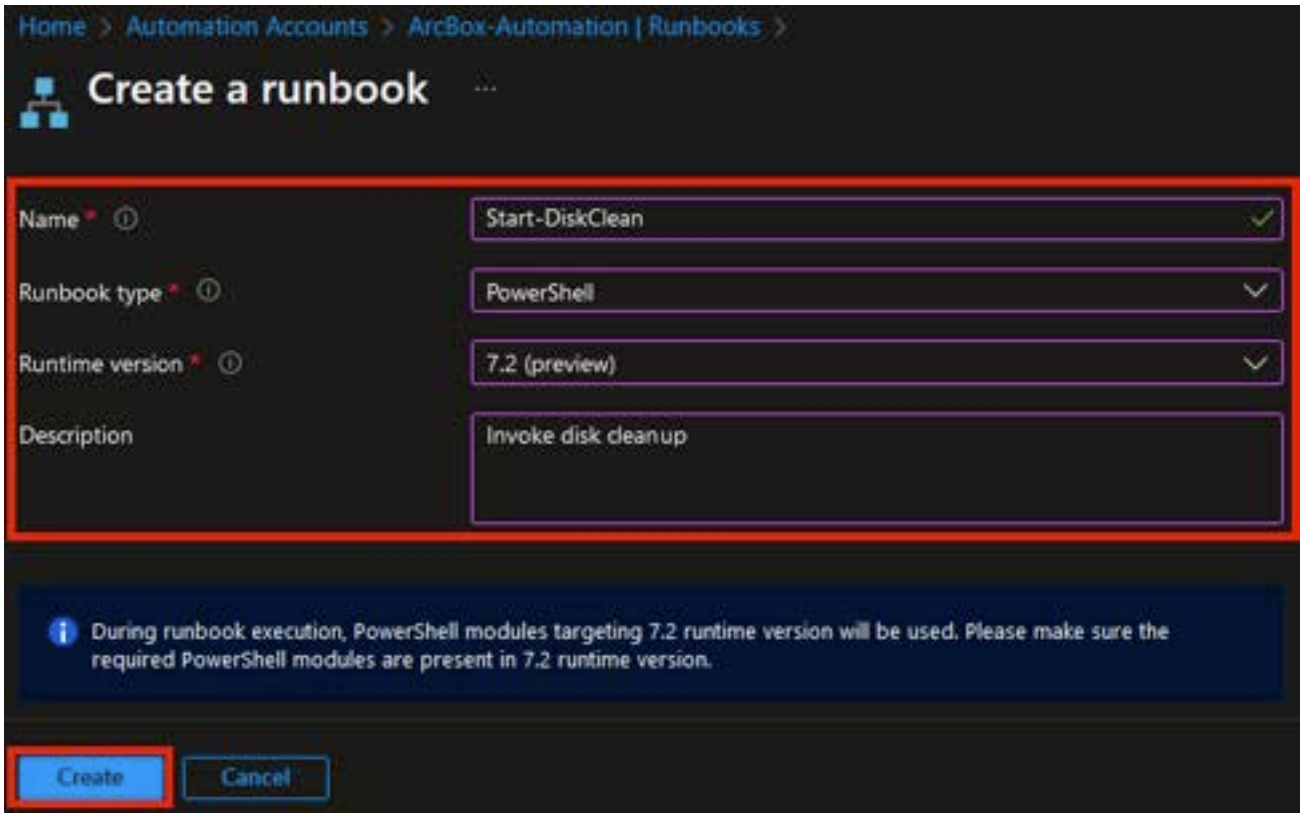
- [ ] 1. In the Azure Portal, search for *automation* and navigate to *Automation accounts*



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



- [ ] 3. Enter the following values then click *Create*
    - Name: Start-DiskClean
    - Runbook type: PowerShell
    - Runtime version: 7.2
    - Description: Invoke disk cleanup



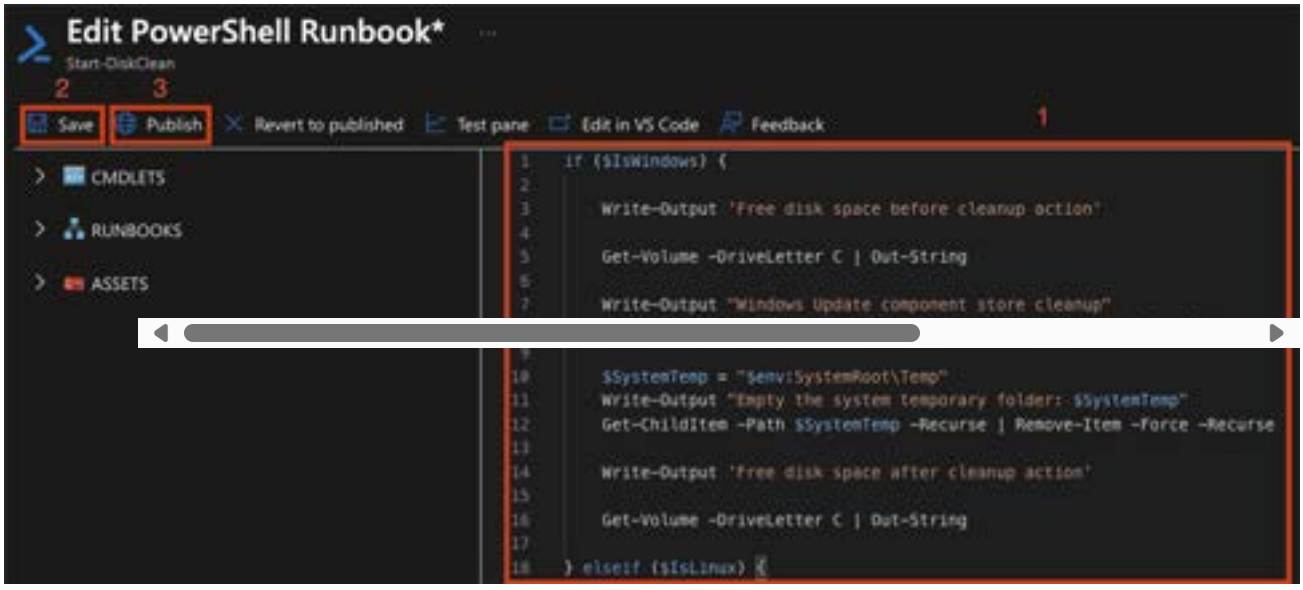
☐ 4. After provisioning, the runbook editor will open the newly created runbook:

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

```powershell
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
```

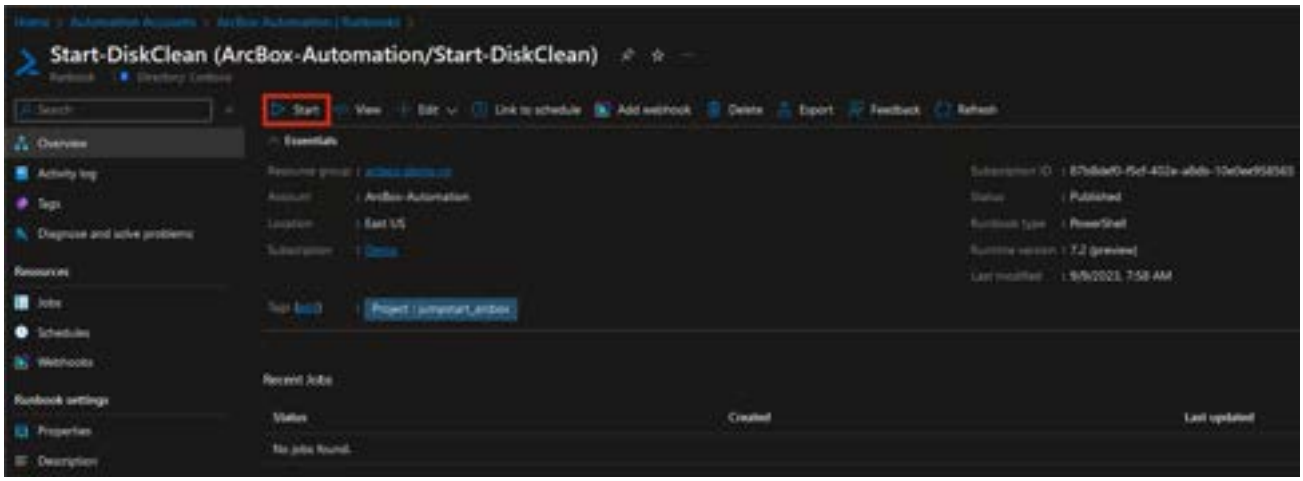
```
  Write-Output 'Free disk space after cleanup action'

  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 thres
  $filesToDelete = Get-ChildItem -Path $logDir -File | Where-Object { $_.La

  # 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

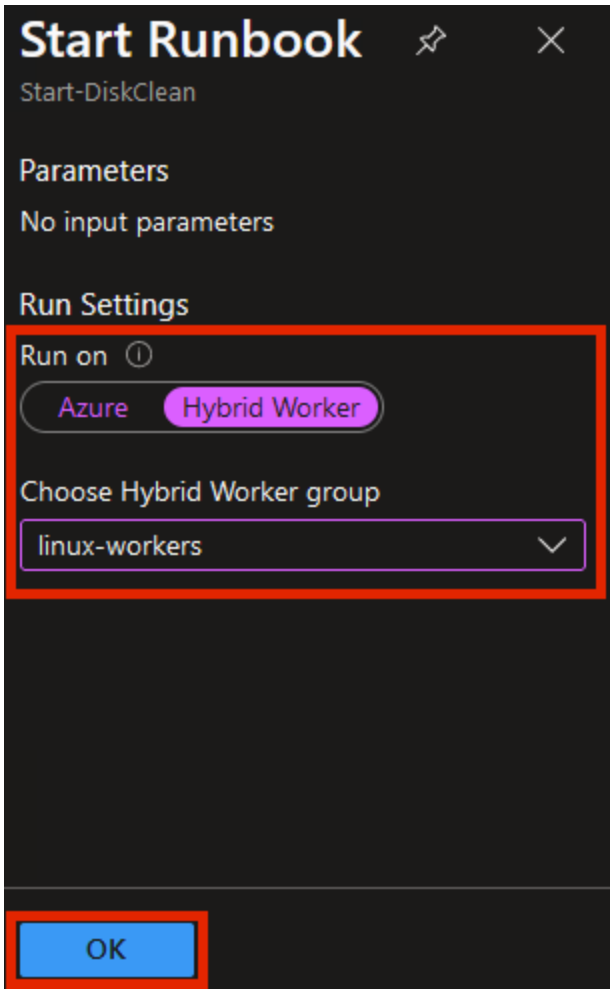


☐ 7. Click *Start*

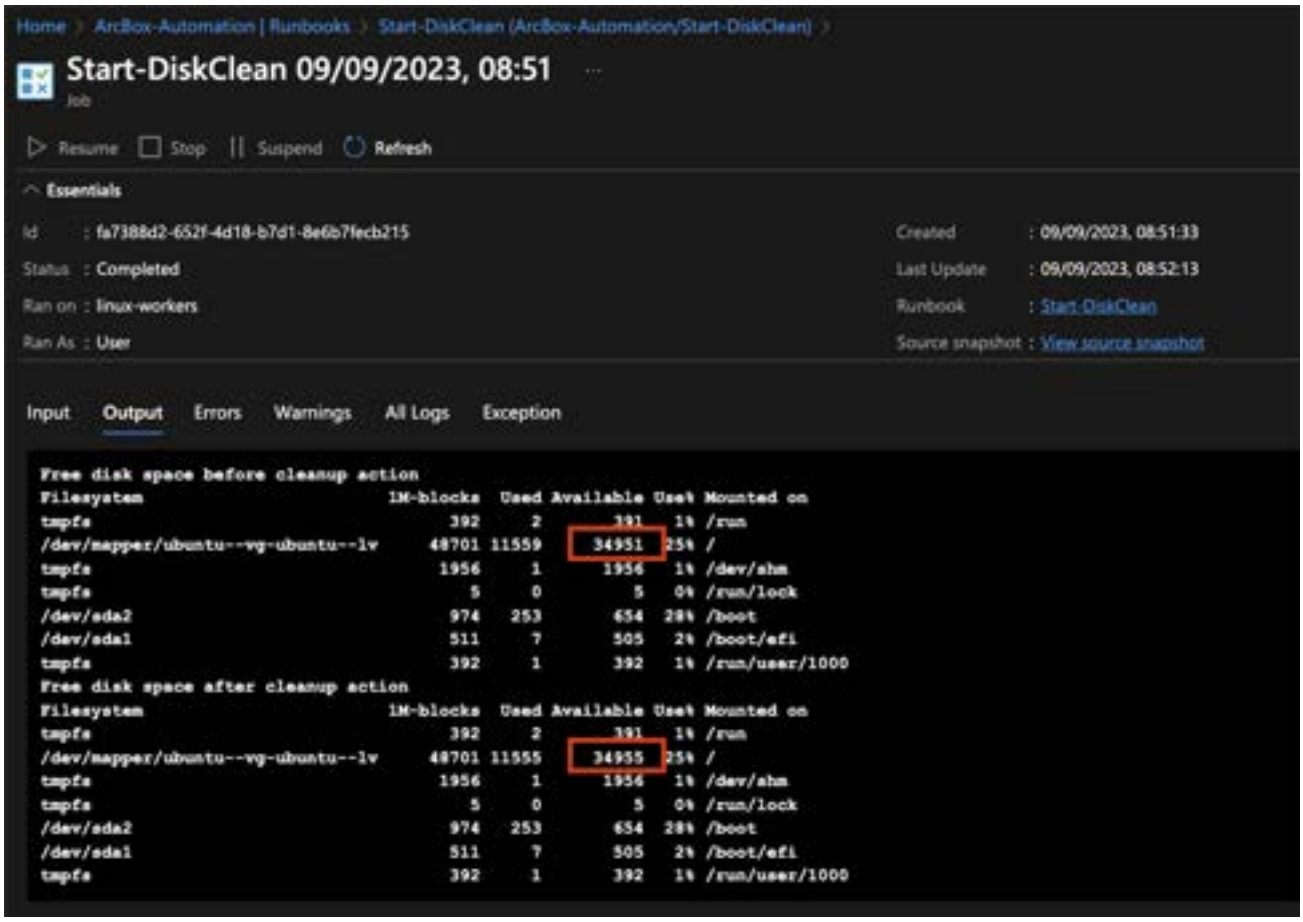
Note: You may need to click *Refresh* for the *Start* button to become active



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

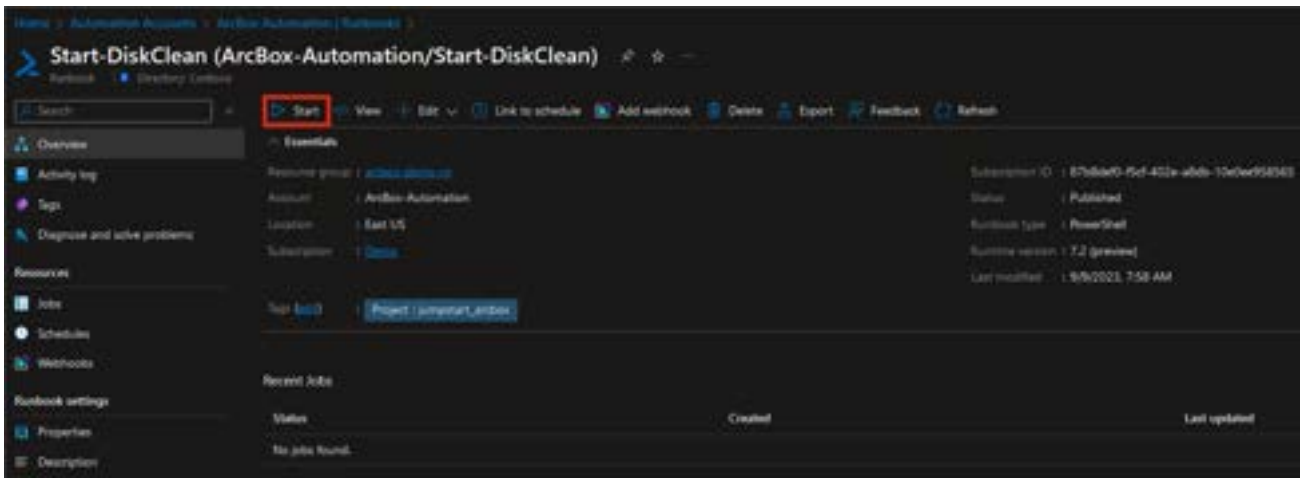


- [ ] 9. Click on the *Output* tab and wait for the job to finish. You should notice that the amount of free space has increased after the cleanup action has completed.

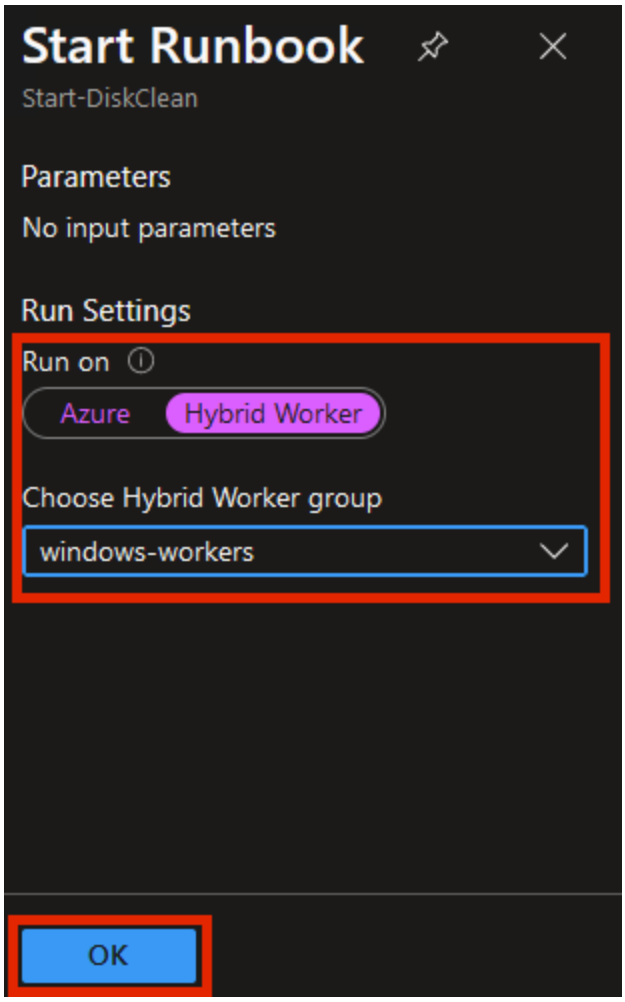


- [ ] The provided runbook is a starting point for cleaning a single directory. Additional logic and directories may be added as required for specific scenarios. For example, it is also possible to add logic to connect to other machines in order to perform cleanup actions on those.

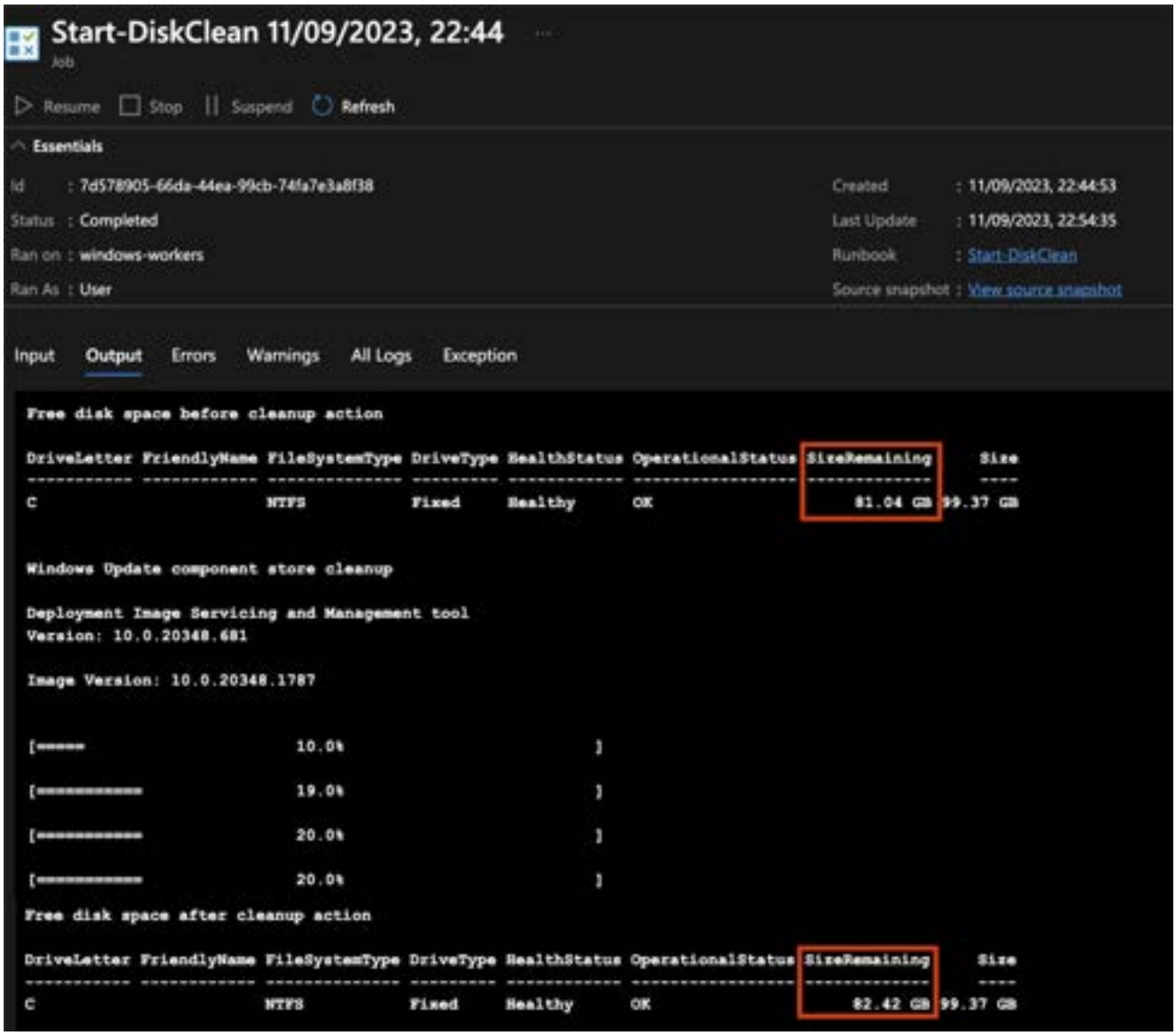
- [ ] 10. Next, you will be running the same runbook on a Windows machine. Navigate back to the runbook overview page for *Start-DiskClean* and click *Start*.



- [ ] 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.



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

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

or

Azure PowerShell

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

- [ ] 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

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:

```PowerShell
# 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
if (!(Get-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -ErrorAction Silen
  Write-Output "Firewall Rule "OpenSSH-Server-In-TCP" does not exist, creat
  New-NetFirewallRule -Name "OpenSSH-Server-In-TCP" -DisplayName "OpenSSH S
} else {
  Write-Output "Firewall rule 'OpenSSH-Server-In-TCP' has been created and
}
```

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

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

Azure CLI

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

or

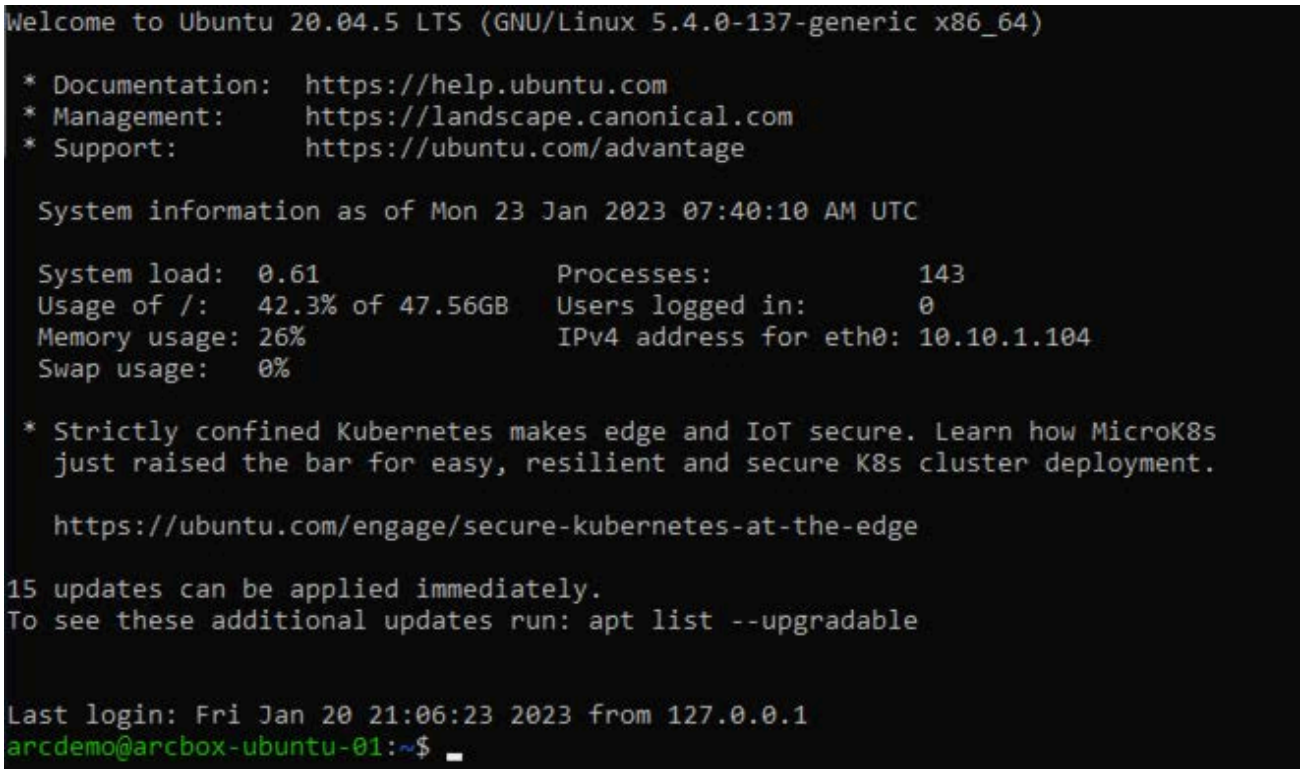
Azure PowerShell

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

2. 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)

3. 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.



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

Azure CLI

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

or

Azure PowerShell

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

- [ ] 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.

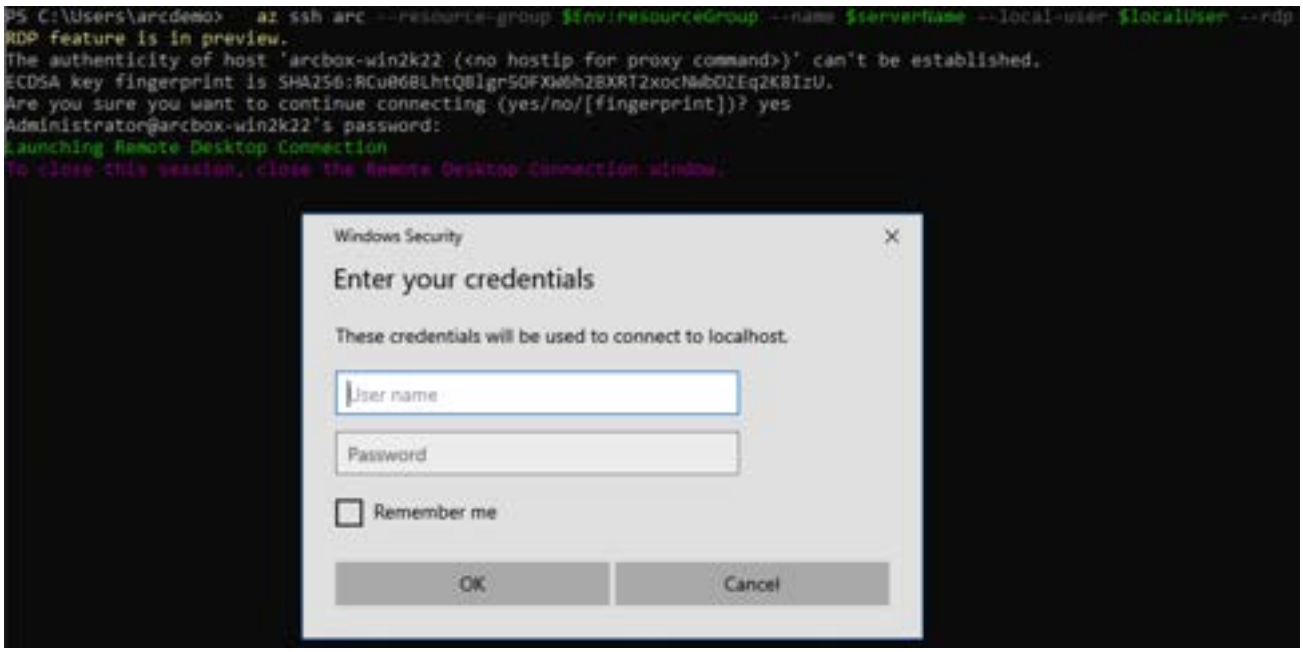
Azure CLI

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

```

or

Azure PowerShell

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

```



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:

```shell
$serverName = "ArcBox-Ubuntu-01"
$resourceGroup = "ArcBox"
az connectedmachine extension create --machine-name $serverName --resource
```

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

`shell # 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

```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:

```shell
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.

- ☐ **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 client machine using the following command:

```PowerShell
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:

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

```PowerShell
#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:

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

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

```PowerShell
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, and the location of your Arc-connected machine:

```
PowerShell
az connectedmachine run-command create --resource-group <Resource Group Nar
```

For example:

```
PowerShell
az connectedmachine run-command create --resource-group ArcBox --machine-na
```

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:

```
PowerShell "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:

```
PowerShell
New-AzConnectedMachineRunCommand -ResourceGroupName "<Resource Group Name>
```

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

```
InstanceViewExecutionMessage      : RunCommand script execution completed
InstanceViewExecutionState        : 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:

```PowerShell
az connectedmachine run-command create --resource-group <Resource Group Nar
```

Or in PowerShell:

```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:

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:

```
PowerShell

az storage account create --name <Storage account name> --resource-group <
```

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

```
PowerShell

az storage container create --name <container name> --account-name <Storag
```

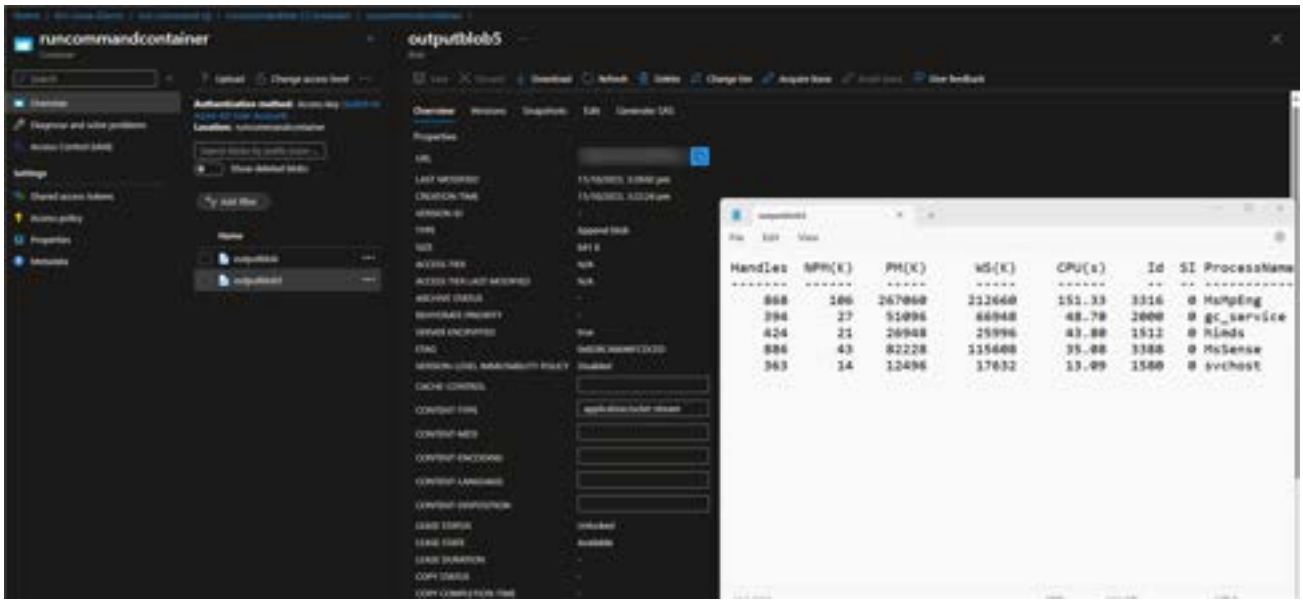
3. Create a blob SAS URI with the following permissions: Read, Write, Create, delete, and append. You will need an end date for the validity of the SAS token, for example one day from the current date. Also, to be able to use the SAS URI in our run command you will need to remove any double quotes from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

```
PowerShell

$end=(Get-date).AddDays(1) | Get-Date -UFormat '+%Y-%m-%dT%H:%MZ'
$sasuri = $(az storage blob generate-sas --account-name <storage account na
```

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:

```
PowerShell

az connectedmachine run-command create --resource-group "<Resource Group Na

New-AzConnectedMachineRunCommand -ResourceGroupName "<Resource Group Name>
```

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.



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**

LAB12: Connect a SQL Server to Azure Arc

Azure Arc automatically installs the Azure extension for SQL Server when a server connected to Azure Arc has SQL Server installed. All the SQL Server instance resources are automatically created in Azure, providing a centralized management platform for all your SQL Server instances.

Student Lab Manual

Table of Contents

Exercise 1 - Pre-Requisites

Objective

Review the necessary pre-requisites to Arc-enable SQL Servers.

Estimated Time to Complete This Exercise

10 minutes

Task 1: Review Pre-requisites

Before you add an Azure Arc-enabled SQL Server you will need the following pre-requisites. **Note that in this lab environment these pre-requisites have already been provided and are listed here for information only.**

- [ ] 1. A virtual or physical machine running SQL Server: The machine hosting SQL Server must be connected to the internet directly or via a proxy server.
- [ ] 2. A user account with permissions: An account with local admin rights. Learn more about the required Azure permissions.
- [ ] 3. PowerShell: Powershell must be installed on the computer executing the onboarding script. Learn how to install Powershell
- [ ] 4. Registered resource providers: The Microsoft.AzureArcData and Microsoft.HybridCompute resource providers must be registered. Learn more about registering resource providers

Task 1 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 2 - Review Existing Arc Resources

Objective

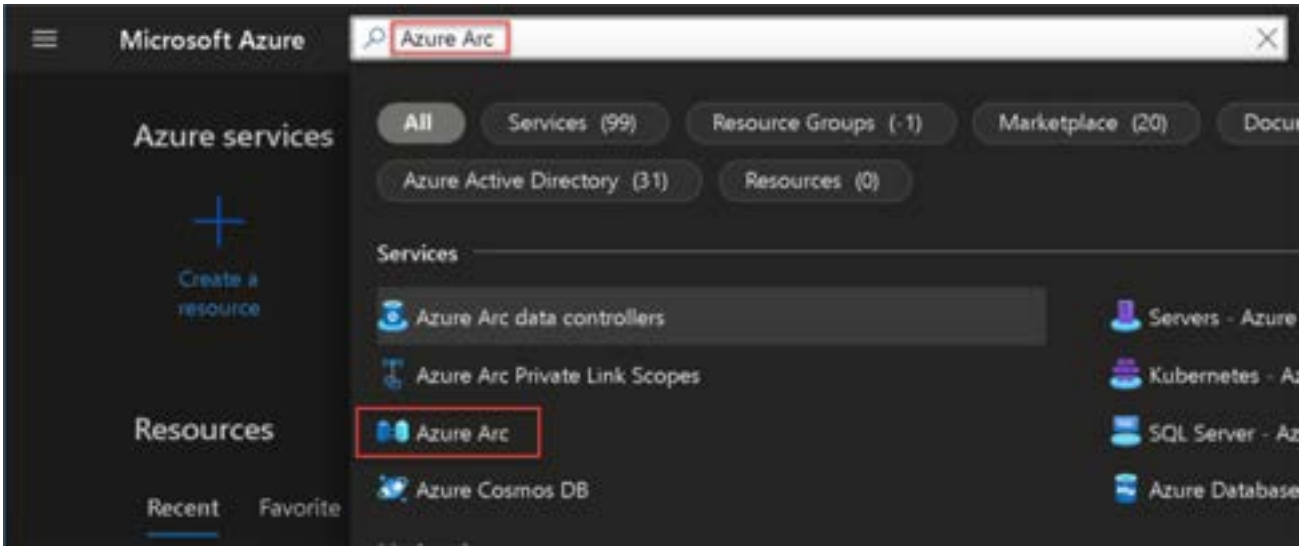
The deployment process that you have walked through in Lab01 should have set up a number of Server VMs running on Hyper-V in the ArcBox-Client machine. One of these Servers *ArcBox-SQL* has a SQL server deployed. In this exercise you will Arc-enable this SQL server.

Estimated Time to Complete This Exercise

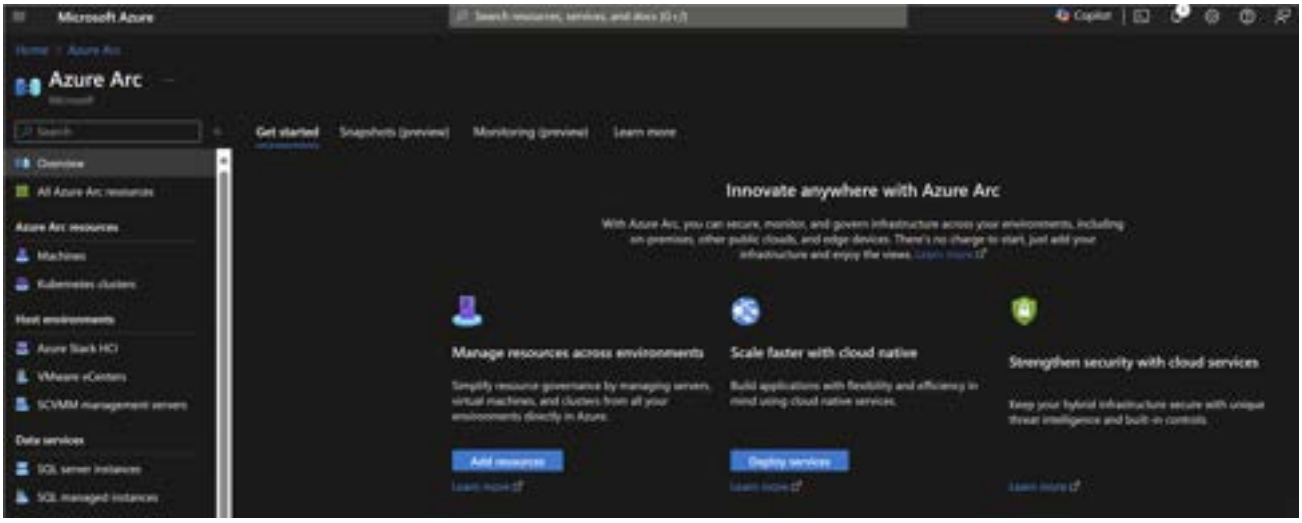
15 minutes

Task 1: Use the Azure portal to examine you Arc-enabled machines inventory

1. In the Azure portal search for *Azure Arc* in the search bar



2. In the Azure Arc page you should see the *Overview* option

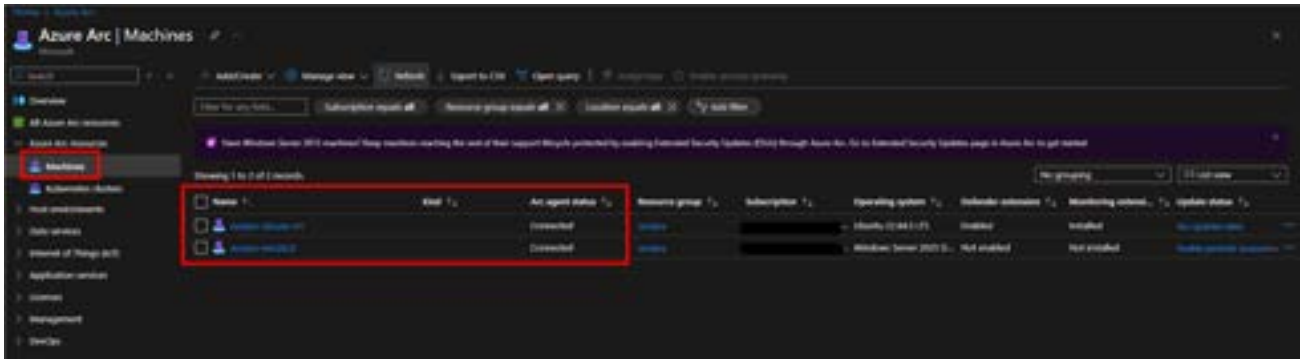


3. Click the *All Azure Arc Resources* option, and you should see the Arc-enabled Windows and Linux servers, but you notice that the *ArcBox-SQL* Windows machine is not Arc-enabled, neither is the SQl Server deployed on it

Note that the Azure Arc resources showing in your portal might be different from the picture below but you should not see the *ArcBox-SQL* machine unless you have onboarded the the Windows server *ArcBox-SQL* optionally in Lab02.

If during Lab02 of this workshop you have also onboarded the Windows server *ArcBox-SQL* then you will see two Arc-enabled resources one for the Windows server and one for the SQL server

hosted inside the Windows server. In the latter case you can jump to Exercise 3 - Task 2.



Task 1 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 3 - Add SQL Server to Azure Arc

Objective

In this exercise you will onboard a SQL Server to Azure Arc by first onboarding the Windows Server hosting the SQL Server. Once the Windows Server is Arc-enabled, it will automatically Arc-enable the SQL Server.

Estimated Time to Complete This Lab

20 minutes

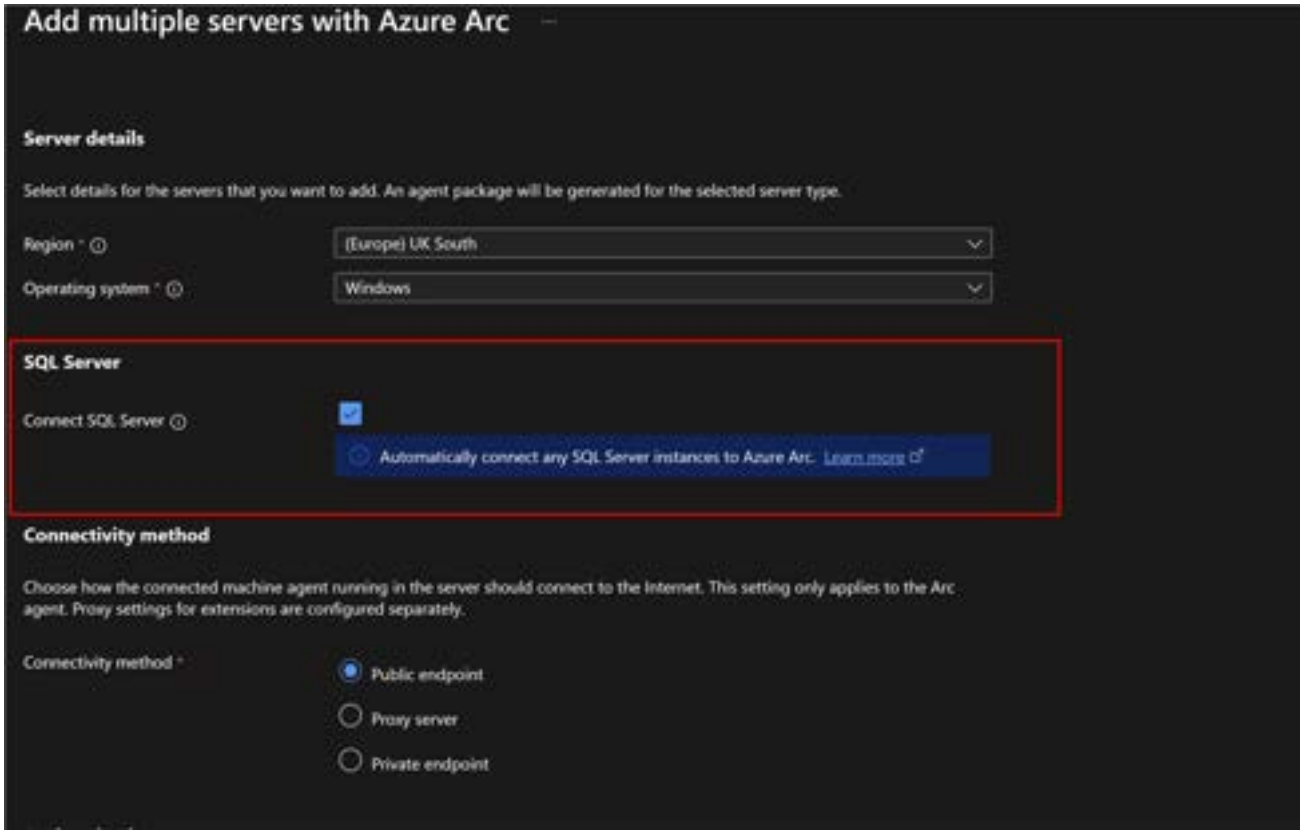
Explanation

Connecting machines in your hybrid environment directly with Azure can be accomplished using different methods, depending on your requirements and the tools you prefer to use. In this instance you will use the same method of *Lab02* to onboard the Windows machine that hosts the SQL Server.

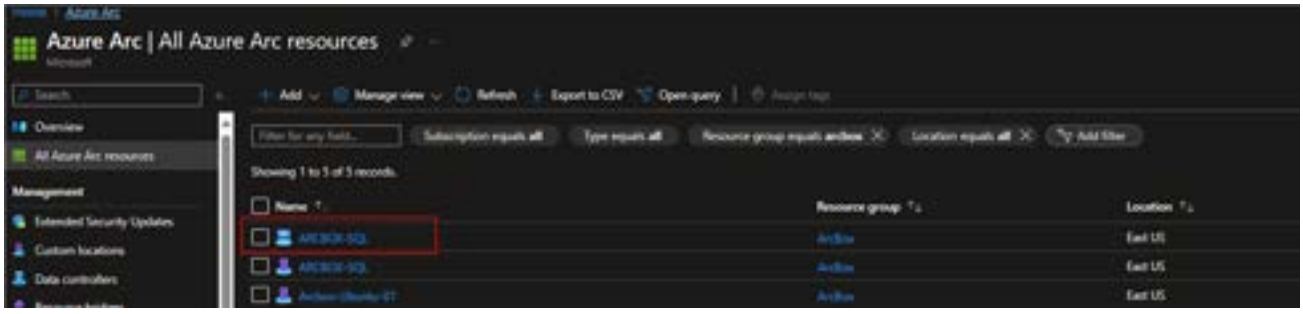
Task 1: Onboard a Windows Server to Azure Arc to automatically Arc-enable the SQL Server

1. Refer to *Lab02* , *Exercise 2* and follow the steps to onboard the nested *ArcBox-SQL* VM. However, this time make sure that you tick the *Connect SQL Server* option as shown below. This will automatically onboard the SQL Server to Arc in addition to the Windows server.

Task 2: Verify that the SQL Server has been onboarded



1. After 5-10 minutes you should notice that the SQL Server *ArcBox-SQL* has also been onboarded. Confirm that the *ArcBox-SQL* VM has been onboarded by viewing all the Azure Arc resources as explained above in *Exercise 1* of this lab.



- [ ] **If after more than 10 minutes** you still do not see the SQL Server *ArcBox-SQL* in the portal, then go to the ***ArcBox-SQL* Windows server** from the Arc page on the Azure portal and check the installed extensions. If you do not see the *WindowsAgent.SqlServer* extension then install it manually by adding it from the portal. After the installation completes you should see the SQL server onboarded.

LAB13: Connect SQL Servers to Azure Arc using offline MSI installer

---

> **This lab shows an alternative method to that of the previous lab for onboarding SQL Servers. You can treat it as an optional lab if you want to move to other labs that explore the capabilities that Azure Arc facilitate for managing SQL Servers**.

In the previous Lab you looked at onboarding a SQL Server to Azure Arc automatically by onboarding the machine that hosts the SQL Server to Azure Arc. In this lab you will look at an alternative method of onboarding the SQL Server using an MSI installer. This method is useful when you want to onboard a number of SQL Servers at scale or if the automatic method of onboarding was not successful. The exercises will be carried out using one server only (due to resource limitations on the lab infrastructure) but the method is applicable to multiple server deployment. Also note that you will be starting from the point where the Windows Server VM itself is connected to Arc and you will only be onboarding the SQL Server that is running on the Windows VM.

The onboarding involves installing an extension on the Arc-enabled Windows machine, namely the *WindowsAgent.SqlServer* extension. This extension, once installed, will detect any running SQL Servers and onboard them automatically to Arc.

Student Lab Manual

Table of Contents

Exercise 1 - Disconnect the Arc-enabled SQL Server instances from Azure Arc using Azure portal

Objective

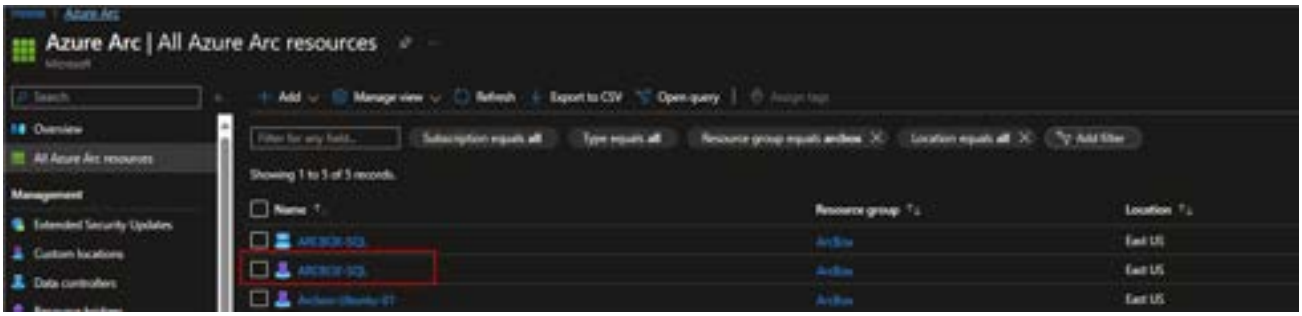
In this exercise you will disconnect the SQL Servers from Arc so that you can practice with other methods of onboarding them at scale. You will keep the Windows Server machine that has the SQL Server installed on it connected to Arc, and you will only remove the SQL Server Arc registration.

Estimated Time to Complete This Exercise

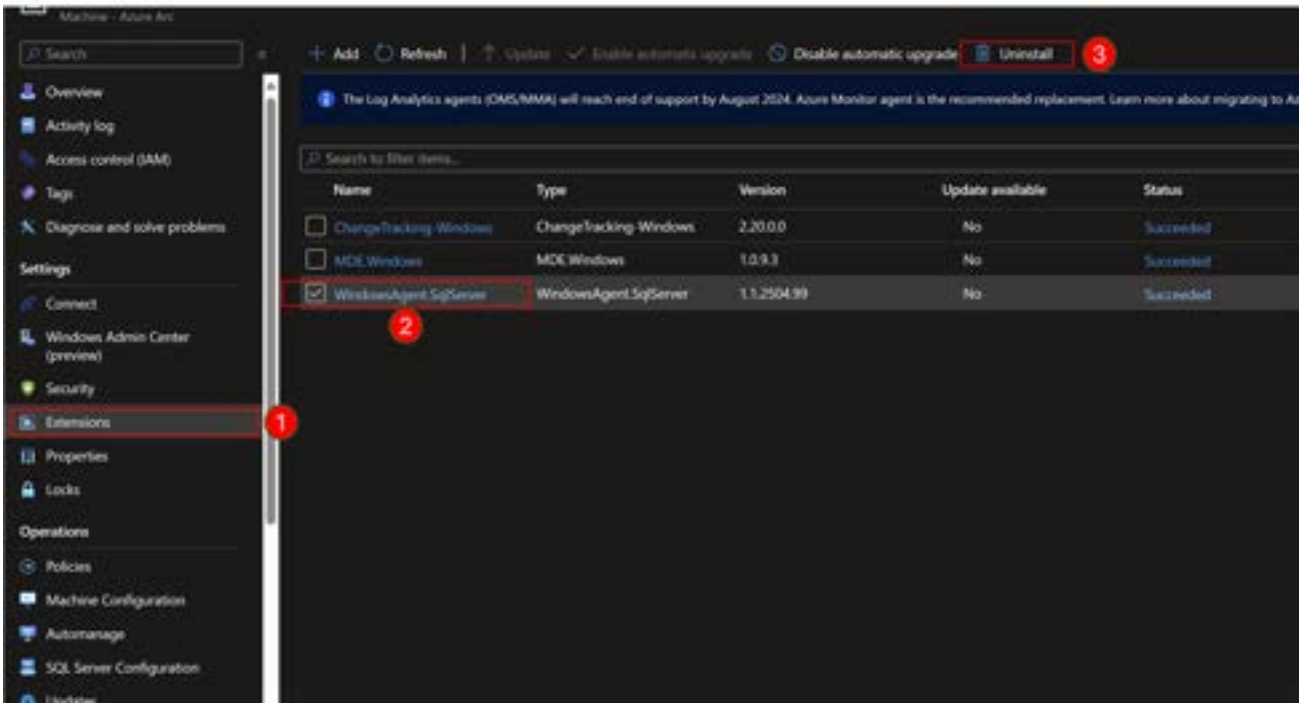
15 minutes

Task 1: Uninstall Azure extension for SQL Server

1. From the Azure portal go to *Azure Arc* page and select the *All Azure Arc Resources* option. Select the *ArcBox-SQL* machine hosting a SQL server



2. On the Server page , under *Extensions* select the *WindowsAgent.SqlServer* extension and then click *Uninstall*. Confirm the request.



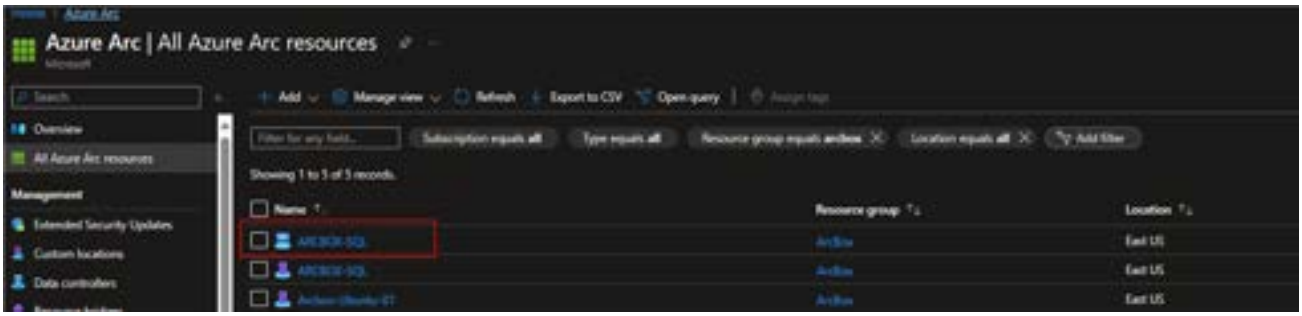
3. Wait for the Agent to be uninstalled and disappear from the list of extensions.

Task 1 has been completed

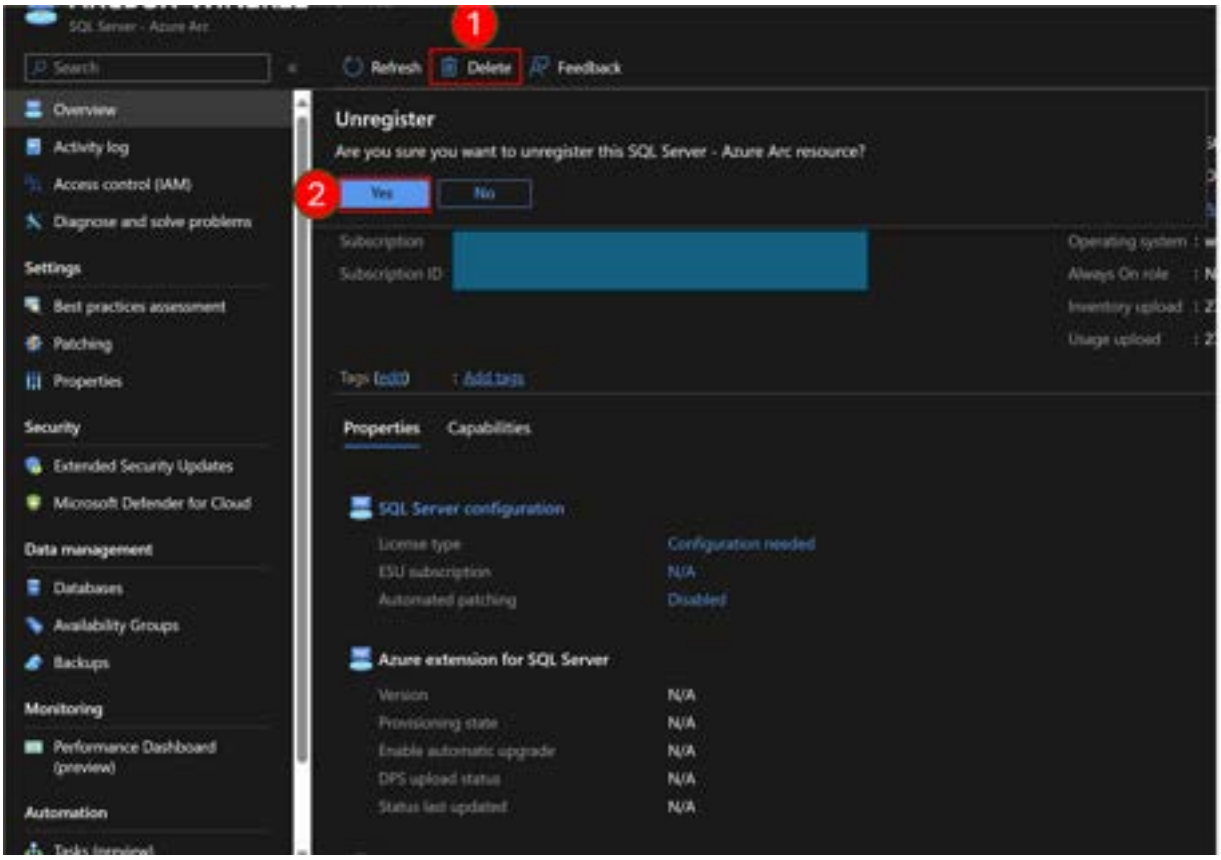
Click **Next** for the next exercise or **Go back to the main table of content**

Task 2: Remove the Arc-enabled SQL Server

☐ 1. From the Azure portal go the *Azure Arc* page and select the *All Azure Arc Resources* option. Select the SQL Server *ArcBox-SQL*



☐ 2. Click on the Delete tab and Confirm that you want to unregister the resource when prompted



Task 2 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 2 - Install SQL Server extension using offline MSI installer and using PowerShell remote execution

Objective

There are many ways to install SQL server extension depending on the your preference. In this module you will learn how to install SQL Server extension using offline MSI installer and using PowerShell remoting.

Estimated Time to Complete This Exercise

30 minutes

Task 1: Connect A SQL Server to Arc using remote Powershell

- [ ] 1. Logon to ArcBox-Client VM, open Windows explorer and navigate to C:\ArcBox folder. Verify that the *InstallArcSQLExtensionAtScale.ps1* PowerShell script file exists.

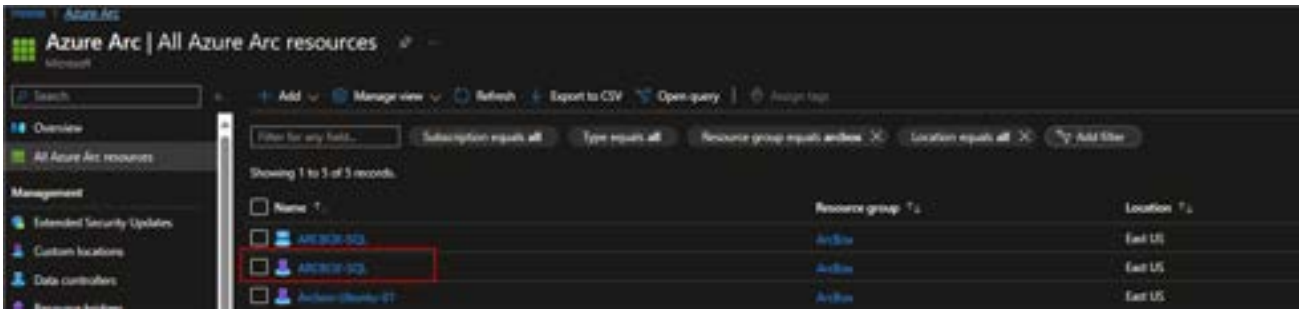
- [ ] 2. Open *InstallArcSQLExtensionAtScale.ps1* in PowerShell ISE editor to review script. Note that the script attempts to download the .msi installer and then tries to run the installer on the list of servers (in this case only one server is specified, *ArcBox-SQL*). The action installs the *WindowsAgent.SqlServer* extension on the specified servers remotely.

- [ ] 3. Open PowerShell command line window and change directory to "C:\ArcBox" folder. Then run the following commands to install the SQL Server extension at scale , **after inserting the password *JS123!!* in the script**.

```powershell
$secWindowsPassword = ConvertTo-SecureString "<Insert password before running>"
.\InstallArcSQLExtensionAtScale.ps1 -remoteWindowsAdminPassword $secWindowsPassw
```

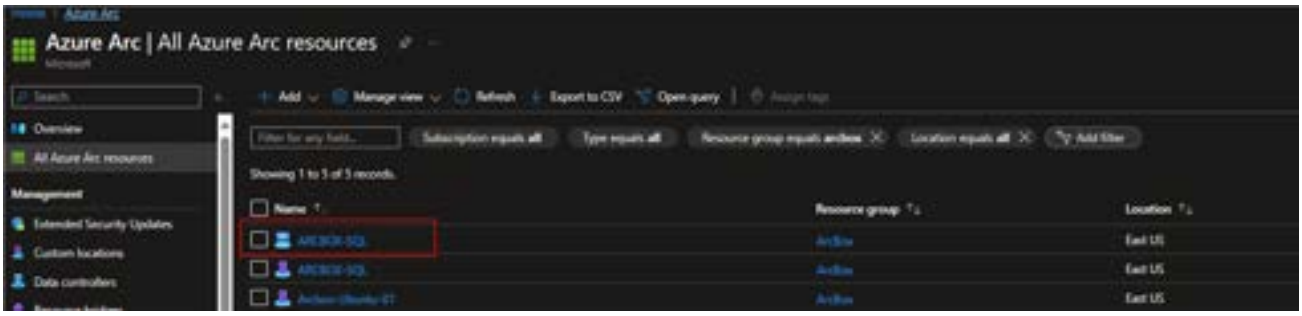
- [ ] 4. In case some of the servers are already onboarded to Arc, you might see a message indicating that the extension already exists. Otherwise the script will install the extension on the specified list of machines. Wait for the script execution to complete.

- [ ] 5. From the Azure portal go to *Azure Arc* page and select the *All Azure Arc Resources* option. Select the *ArcBox-SQL* machine hosting a SQL server.



- [ ] 6. On the Server page , under *Extensions* verify that the *WindowsAgent.SqlServer* extension has been re-installed.

7. From the Azure portal go to *Azure Arc* page and select the *All Azure Arc Resources* option. Verify that the *ArcBox-SQL* SQL Server is now Arc-enabled again.



Task 1 has been completed

Click **Next** for the next lab or **Go back to the main table of content**

LAB14: Get insights on your Arc-enabled SQL Servers and Databases using the Azure Resource Graph

---

Student Lab Manual

Table of Contents

Exercise 1 - Collect insights on your SQL Server instances

Objective

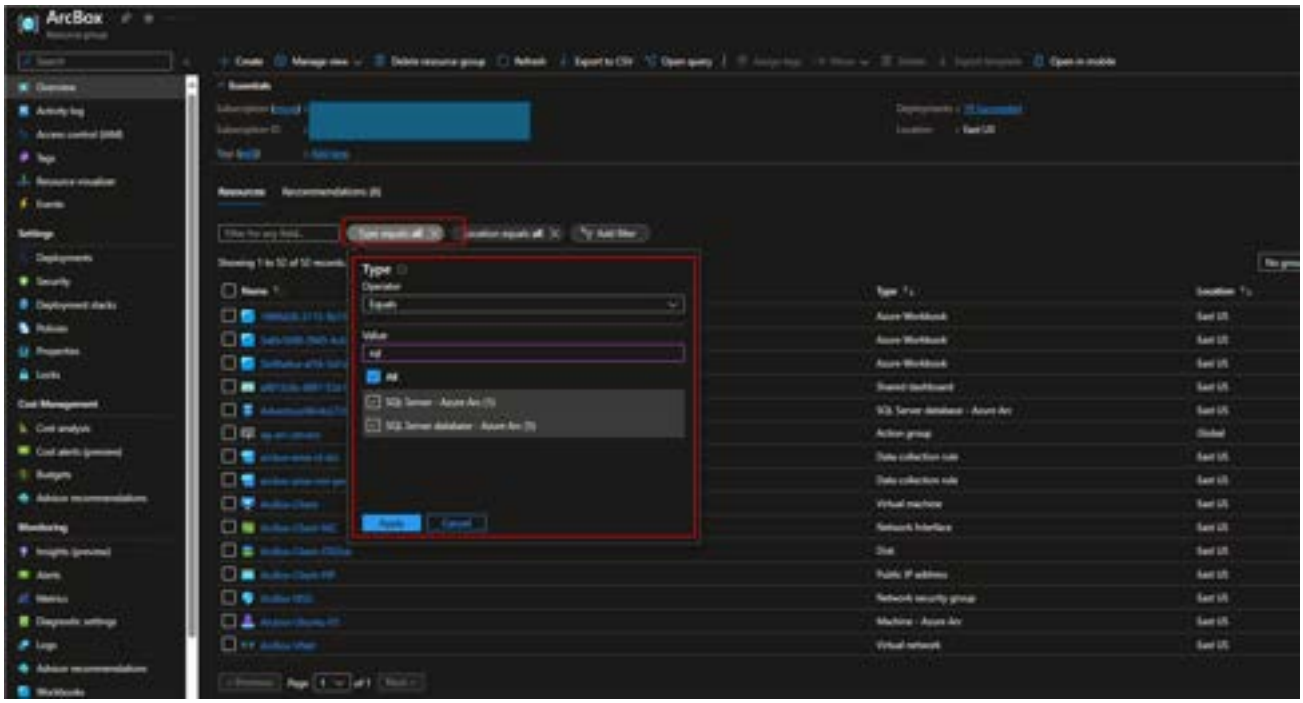
In this exercise you will examine the Arc-enabled SQL Servers and the databases deployed on them.

Estimated Time to Complete This Exercise

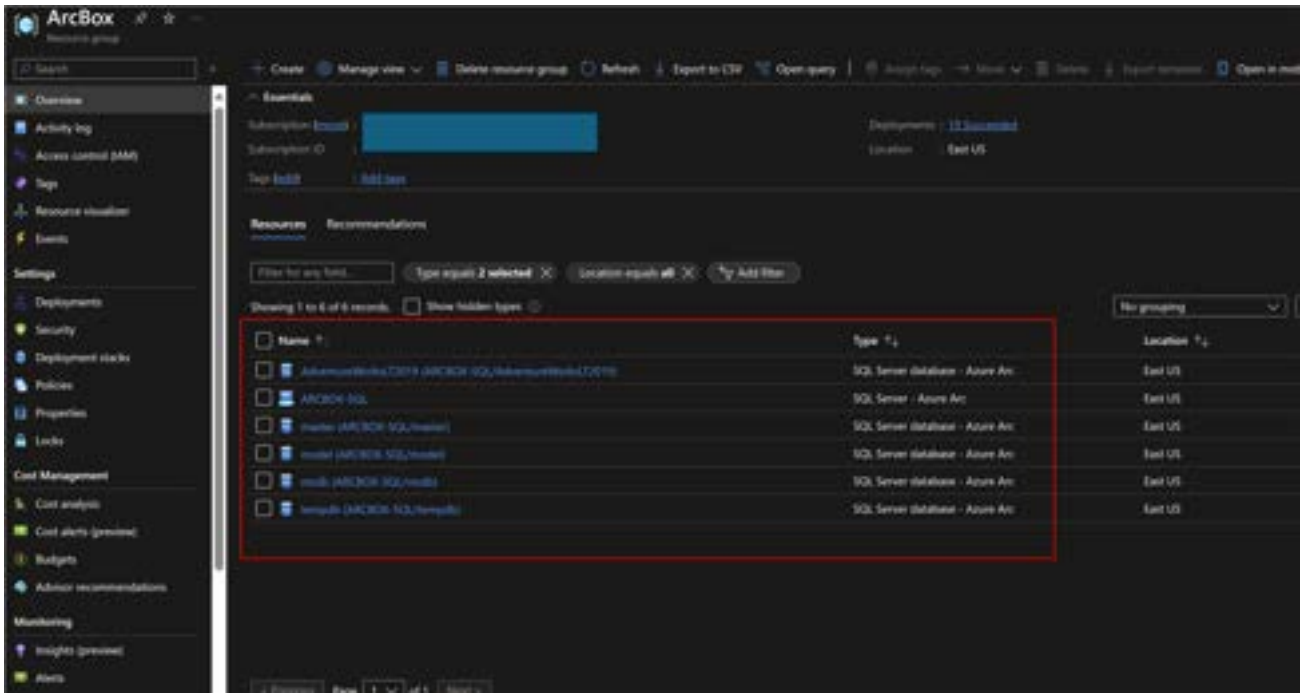
30 minutes

Task 1: Use the Azure Portal to examine your SQL servers and databases

1. From the Azure portal home page, look for Resource Groups and select the one where you have been using in this workshop. Review the of *SQL Server – Azure Arc* and *SQL Server database – Azure Arc* resources that have now been onboarded. To make it easier you can filter by type or order by Type



2. Your SQL Servers are displayed in the format *Server Instance* and your databases are in the format *Database (Server Instance/Database)* where *Instance* only applies for named instances



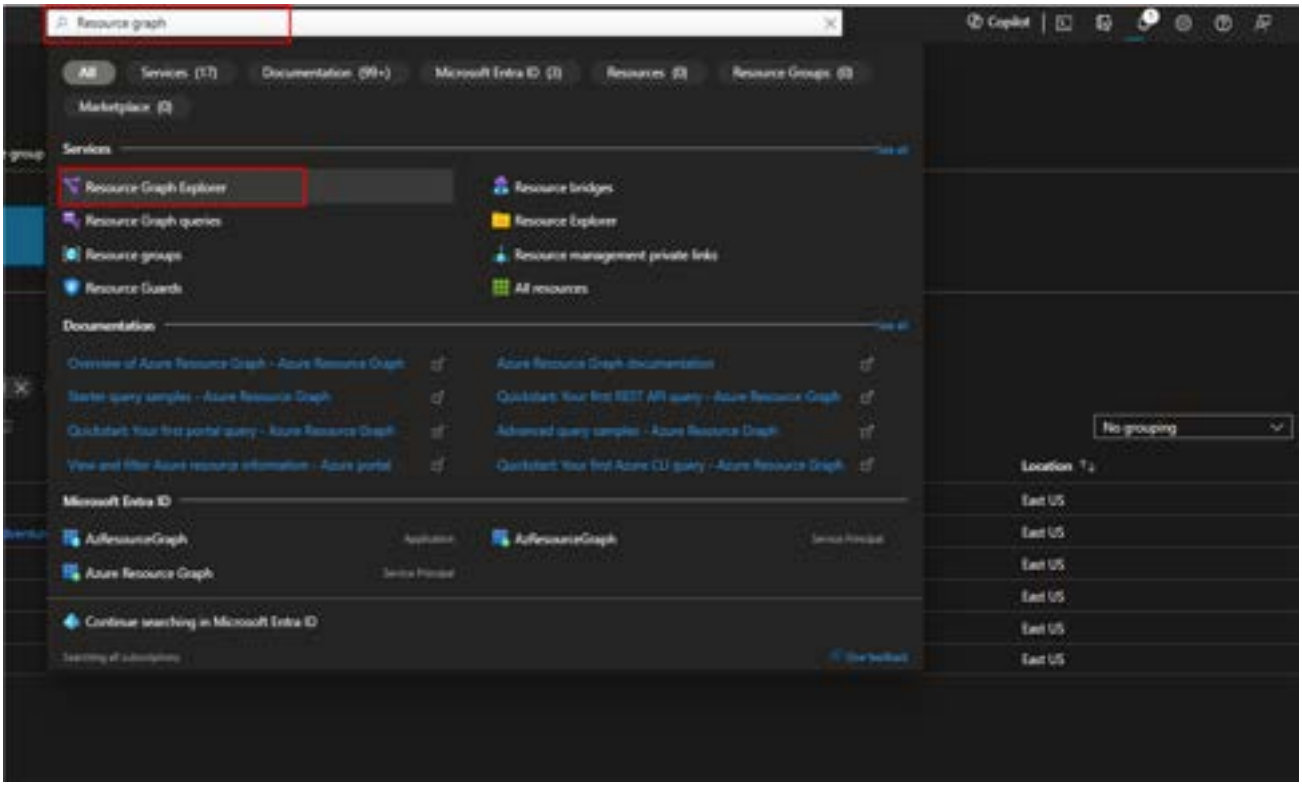
- [ ] 3. Click on any SQL server or database to examine further details and properties about them

Task 1 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

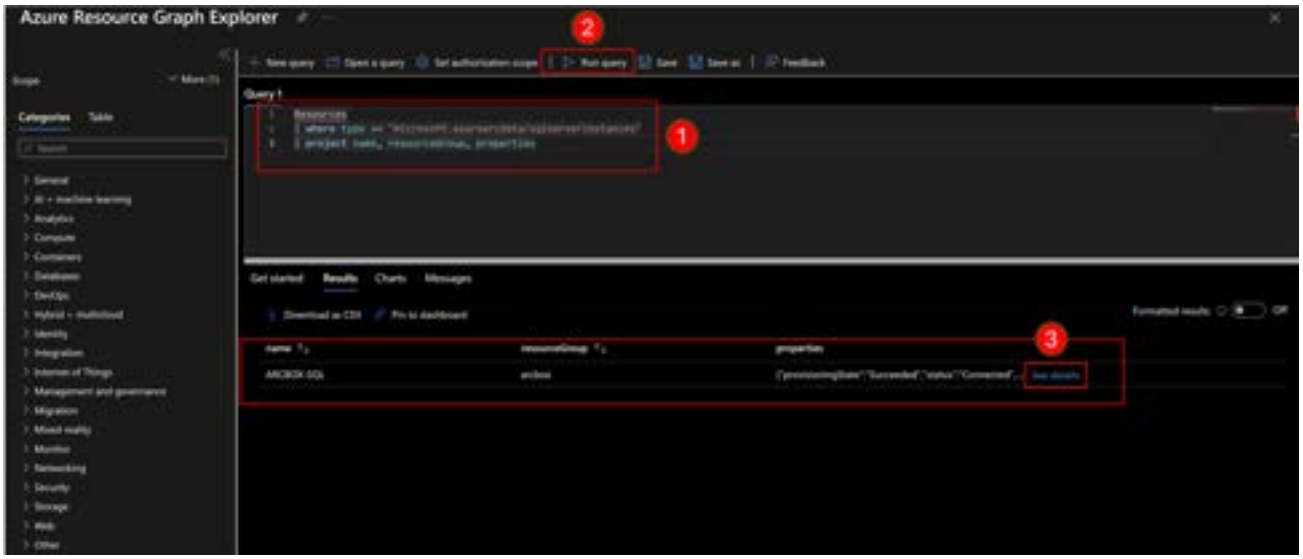
Task 2: Query SQL Servers and Databases using the Resource Graph Explorer

1. From the Home page of the Azure portal, click on the search bar and type Resource graph and then select *Resource Graph Explorer*

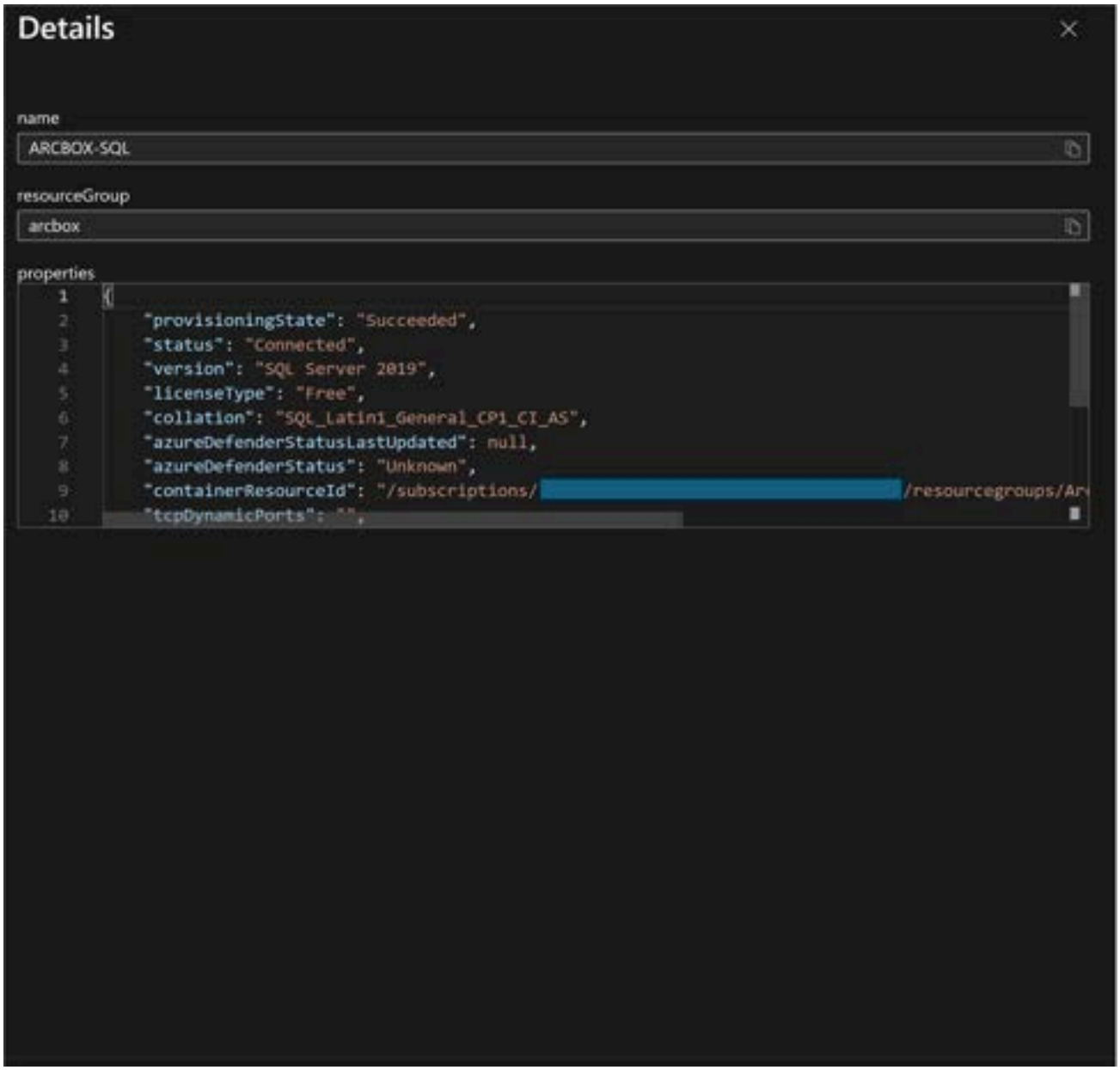


2. Copy and paste the following into the Query window and then click *Run query*. It will list all of your Arc-Enabled SQL Servers by Resource group.

```shell
Resources
| where type == "microsoft.azurearcdata/sqlserverinstances"
| project name, resourceGroup, properties
```

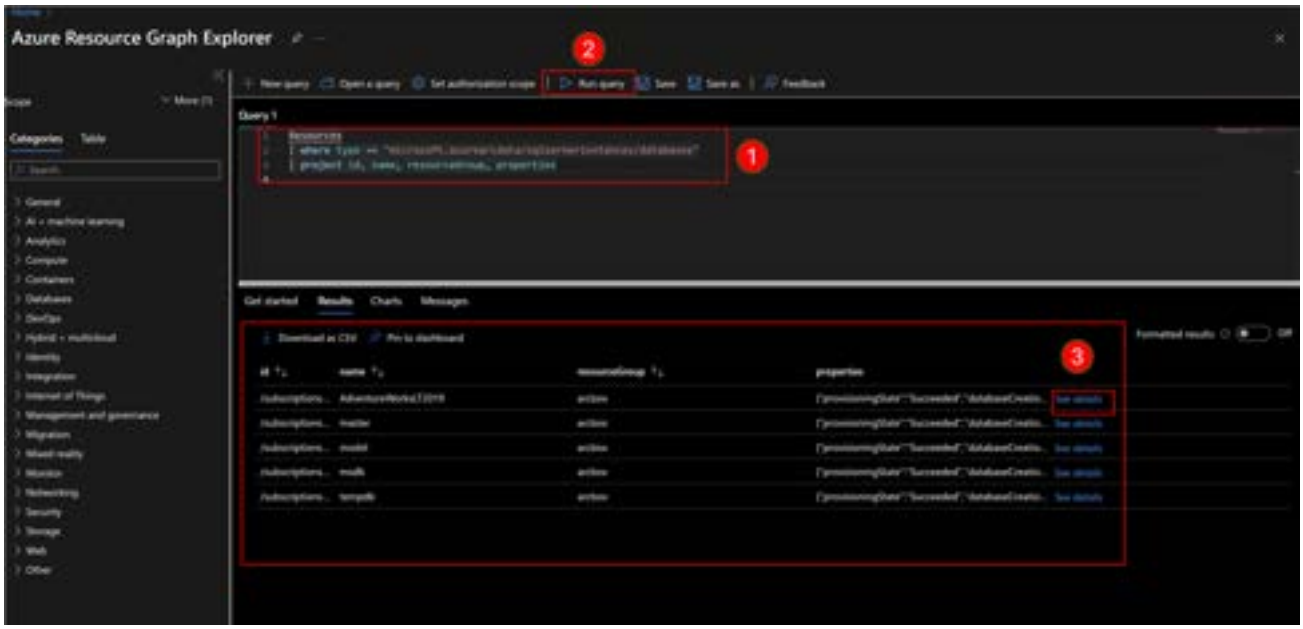


☐ 3. Examine the information displayed by the *See details* link



4. Run the following query to return details about your databases. Examine the results from clicking the *See details* link for each database

```shell
Resources
| where type == "microsoft.azurearcdata/sqlserverinstances/databases"
| project id, name, resourceGroup, properties
```



5. Click on the See details links to provide a better view of the properties for each database in the result set

6. Run the following query to show the compatibility levels of the databases on all your Arc-enabled servers

```shell
resources
| where type == "microsoft.azurearcdata/sqlserverinstances"
| project InstanceId = id, InstanceName = name, Version = tostring(properties.ve
| join kind = inner (
 resources
  | where type == "microsoft.azurearcdata/sqlserverinstances/databases"
  | where name !in ("master", "model", "msdb", "tempdb")
  | project InstanceId = substring(id, 0, indexof(id, "Databases", 0) - 1), Datab
) on InstanceId
| project InstanceName, DatabaseName, CompatibilityLevel = strcat(Version, " - L
```

Task 2 has been completed

Click **Next** for the next lab or **Go back to the main table of content**

LAB15: Enable and run SQL Server best practices assessment

Best practices assessment provides a mechanism to evaluate the configuration of your SQL Server. After you enable best practices assessment, an assessment scans your SQL Server instance and databases to provide recommendations for things like:

- SQL Server and database configurations
- Index management
- Deprecated features
- Enabled or missing trace flags
- Statistics
- & more

Student Lab Manual

Table of Contents

Exercise 1 - Carry out SQL Server Best Practices Assessment

Objective

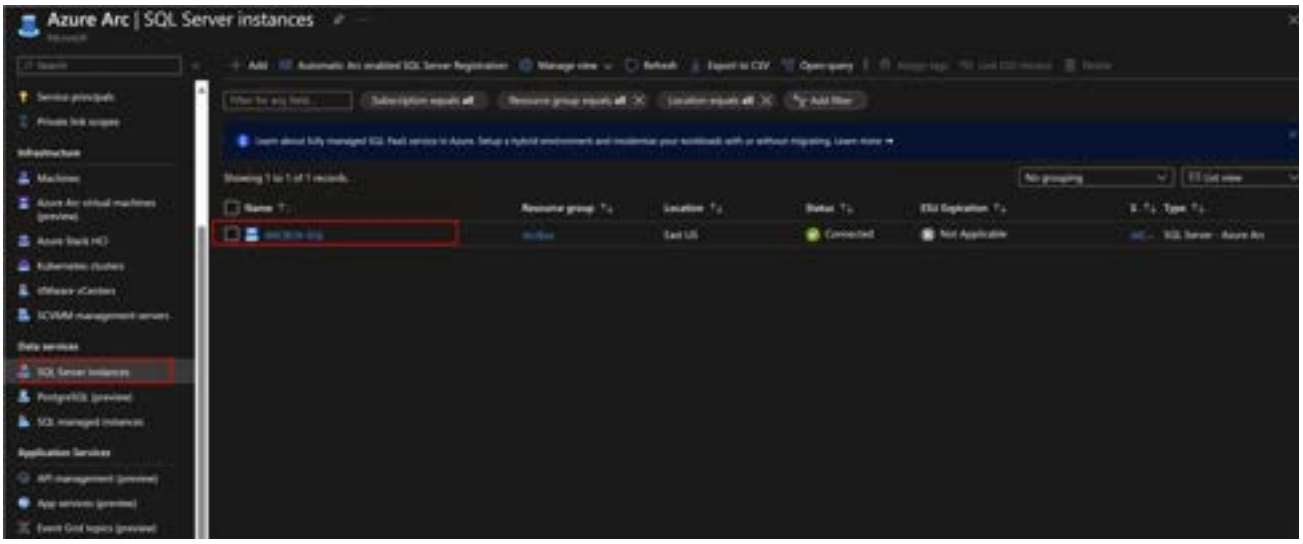
In this exercise you will set up and execute SQL Server Best Practices Assessment for your Arc-enabled SQl Server.

Estimated Time to Complete This Exercise

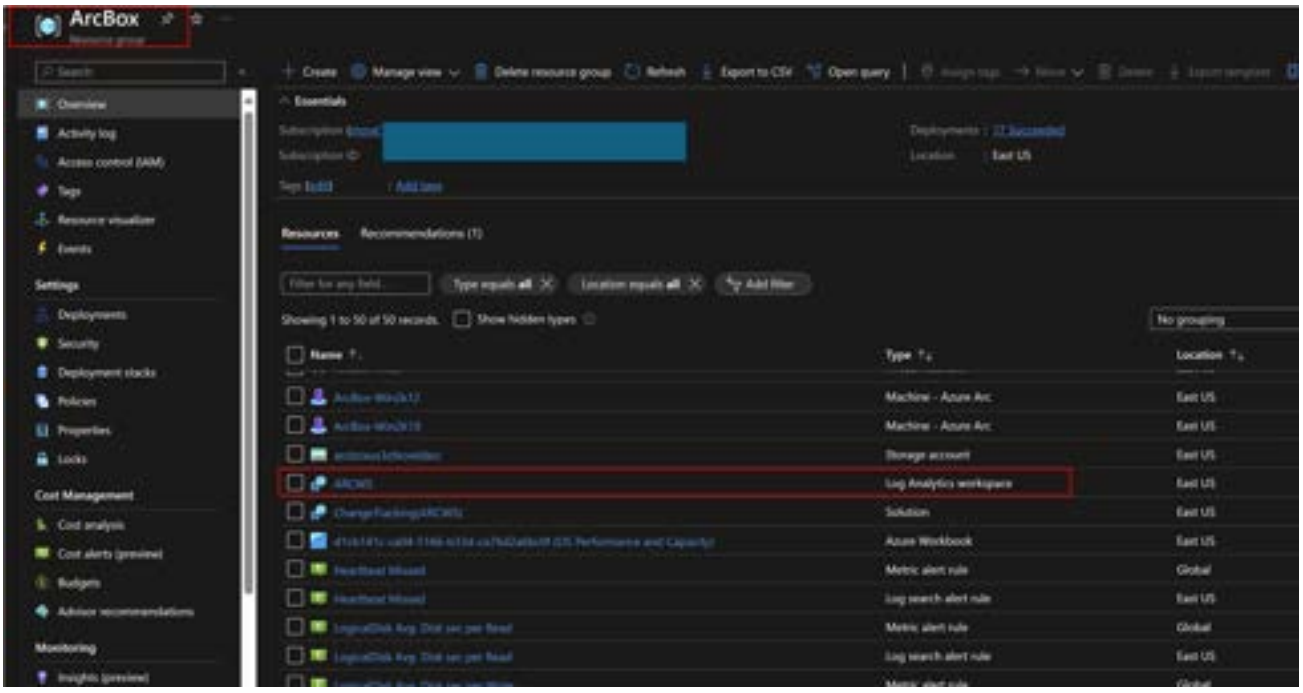
30 minutes

Task 1: SQL Server Best Practice Assessment Prerequisites

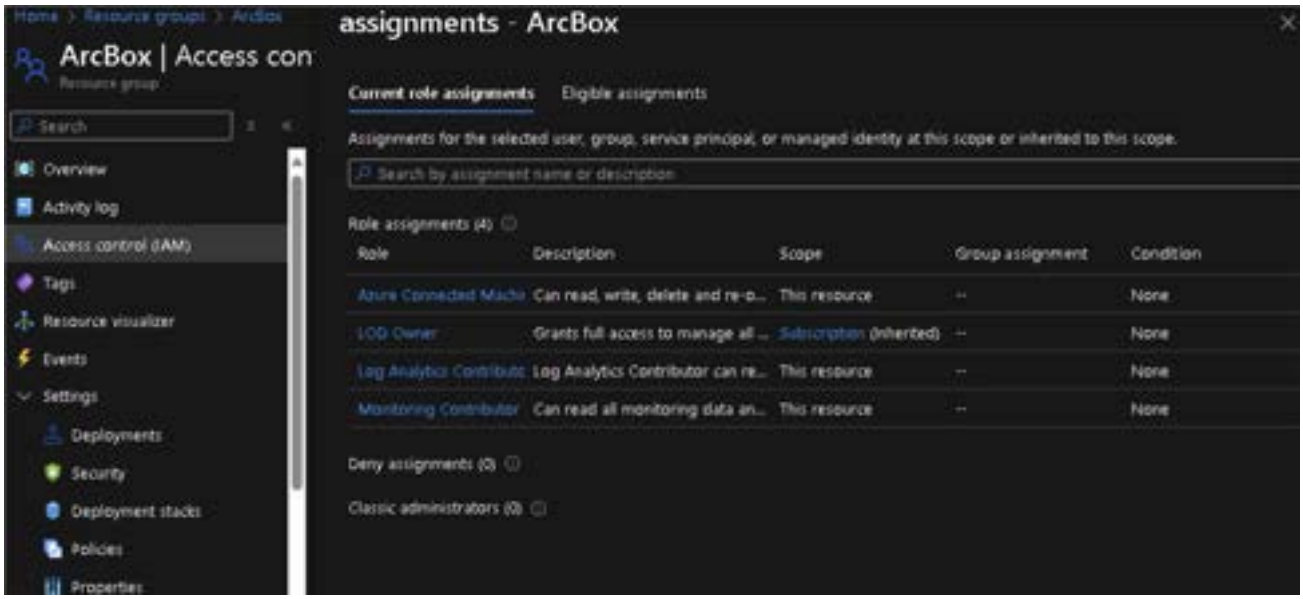
1. Make Sure that your SQL Server instance is connected to Azure Arc. From Azure portal go the *Azure Arc* page and select *SQL Server instances*. You should see the *ArcBox-SQL* SQL server connected.



2. Make sure that you have a Log Analytics workspace deployed in your resource group so that you can upload assessment results to it. The set-up scripts for this workshop should have set up one for you already in your lab resource group.



3. Make sure that you have the following roles enabled and active **on the Resource Group Level** : *Monitoring Contributor*, *Log Analytics Contributor* and *Azure Connected Machine Resource Administrator*. If these roles are not enabled then go to the _Access Control (IAM) tab on the Resource Group page and enable them **making sure to set the Assignment Type to Active**.

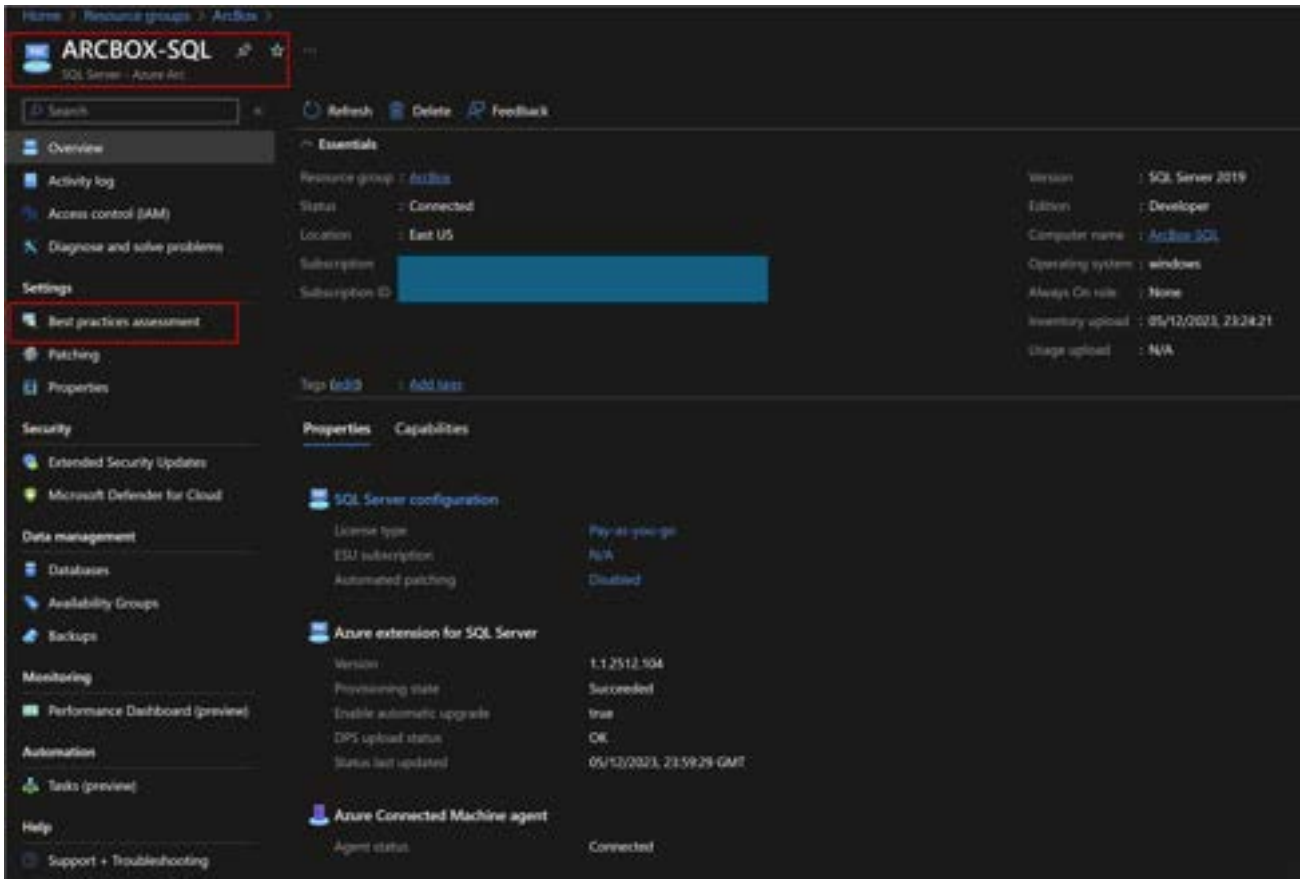


Task 1 has been completed

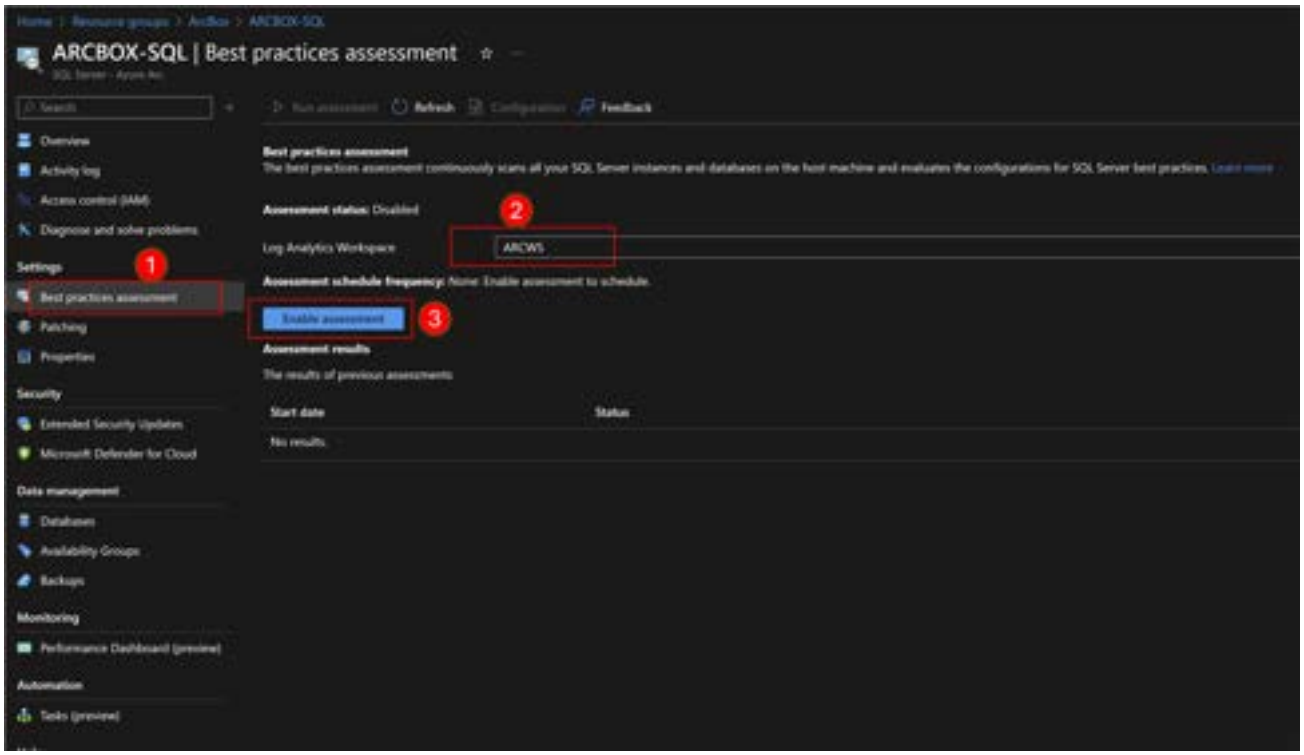
Click **Next** for the next exercise or **Go back to the main table of content**

Task 2: Enable and run SQL Server best practices assessment

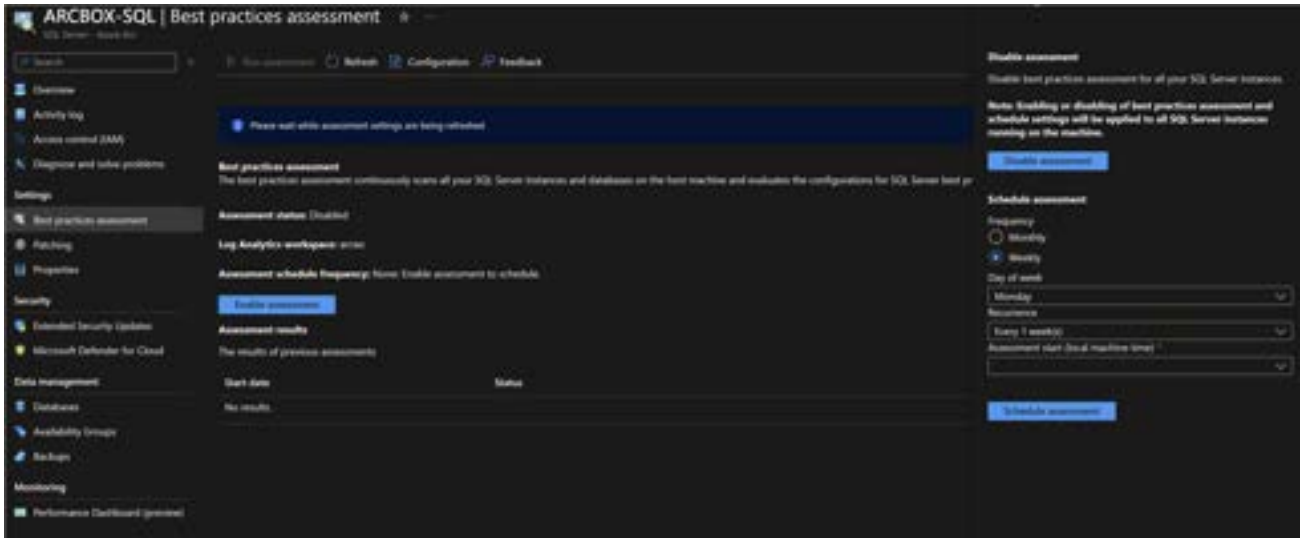
1. Go to your Arc-enabled SQL Server resource and click on the *Best Practices Assessment* under the *Setting* menu. If you are asked for a license type then follow the portal instructions and assign a PAYG license.



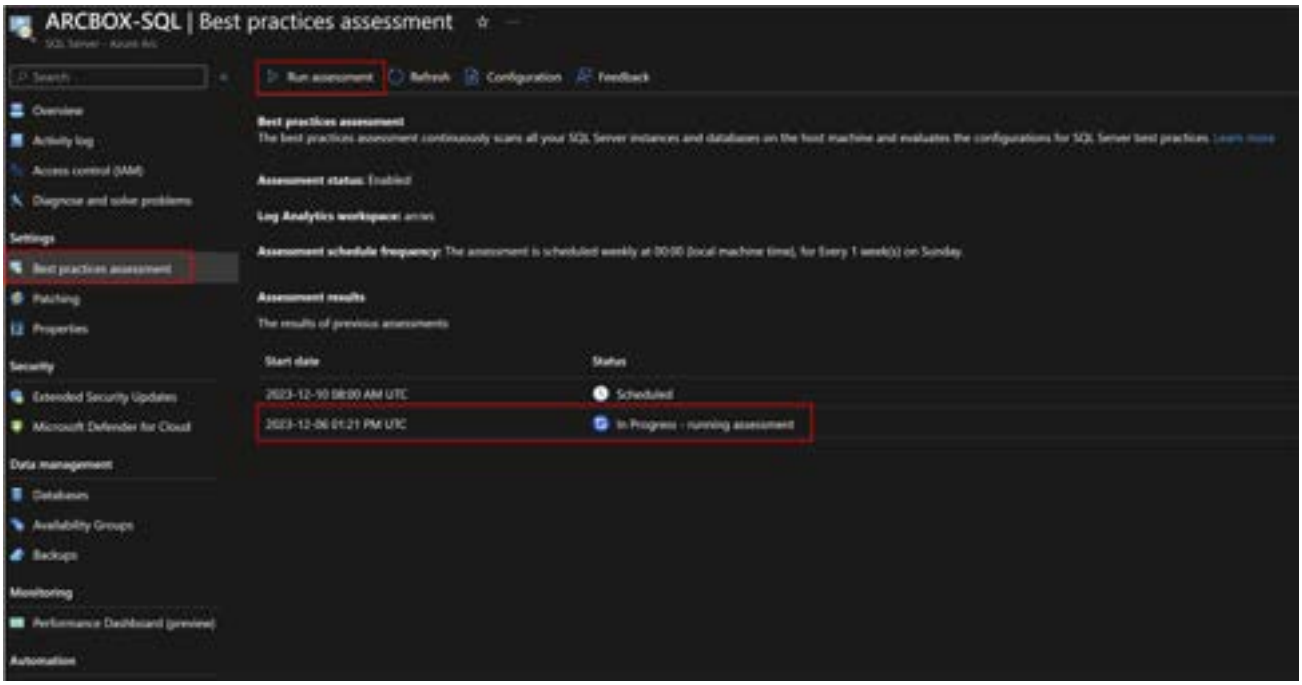
2. Select the log analytics workspace that you have in the same subscription as your Arc-enabled SQL Server resource and click on *Enable Assessment*.



- [ ] 3. Wait until Best Practice Assessment is enabled which might take few minutes. Once the deployment of the best practice assessment is completed you can either start the assessment manually or you can use the *Configuration* option to either schedule or disable the assessment. Examine the *Configuration* settings **but do not click *Schedule assessment*** as you will use the manual option in this lab. Close the *Configuration* settings.



- [ ] 4. Click *Run assessment* to manually start the assessment. You should see the progress indicator after a couple of minutes. Note that the assessment will take a bit of time to complete and you might need to refresh the page to see the results.



Task 2 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

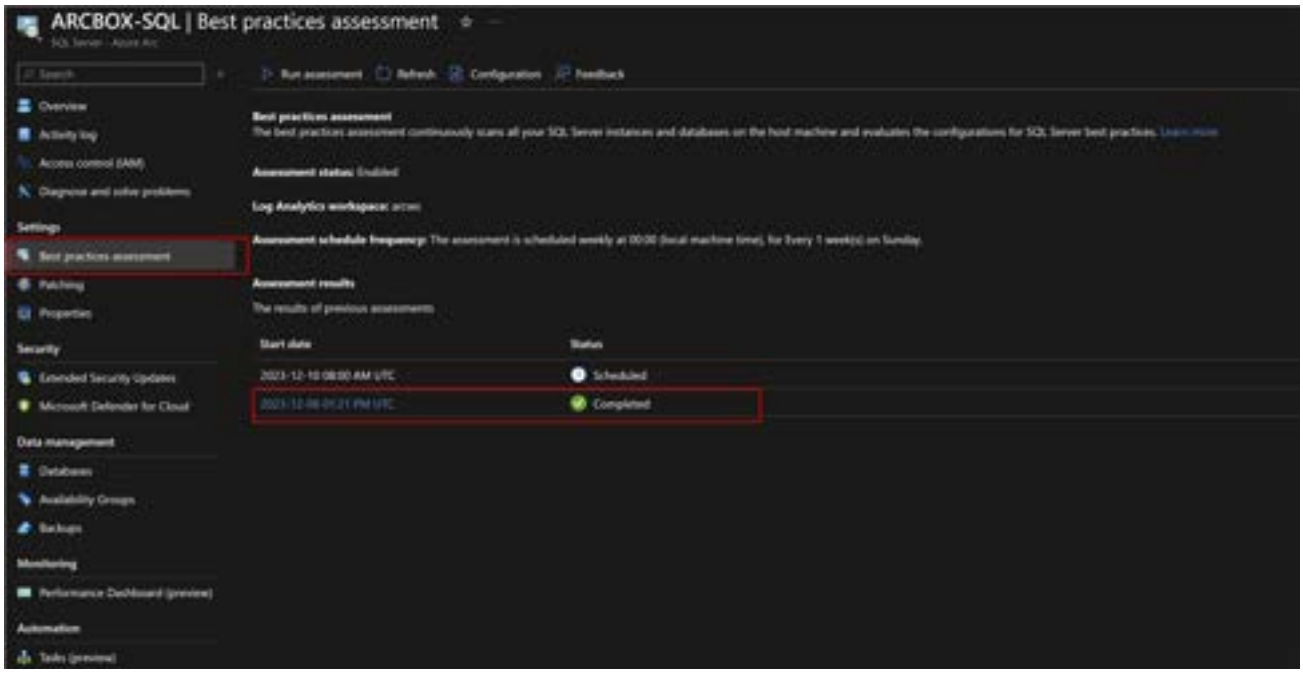
Exercise 2 - Work with Best Practices Assessment results

Estimated Time to Complete This Exercise

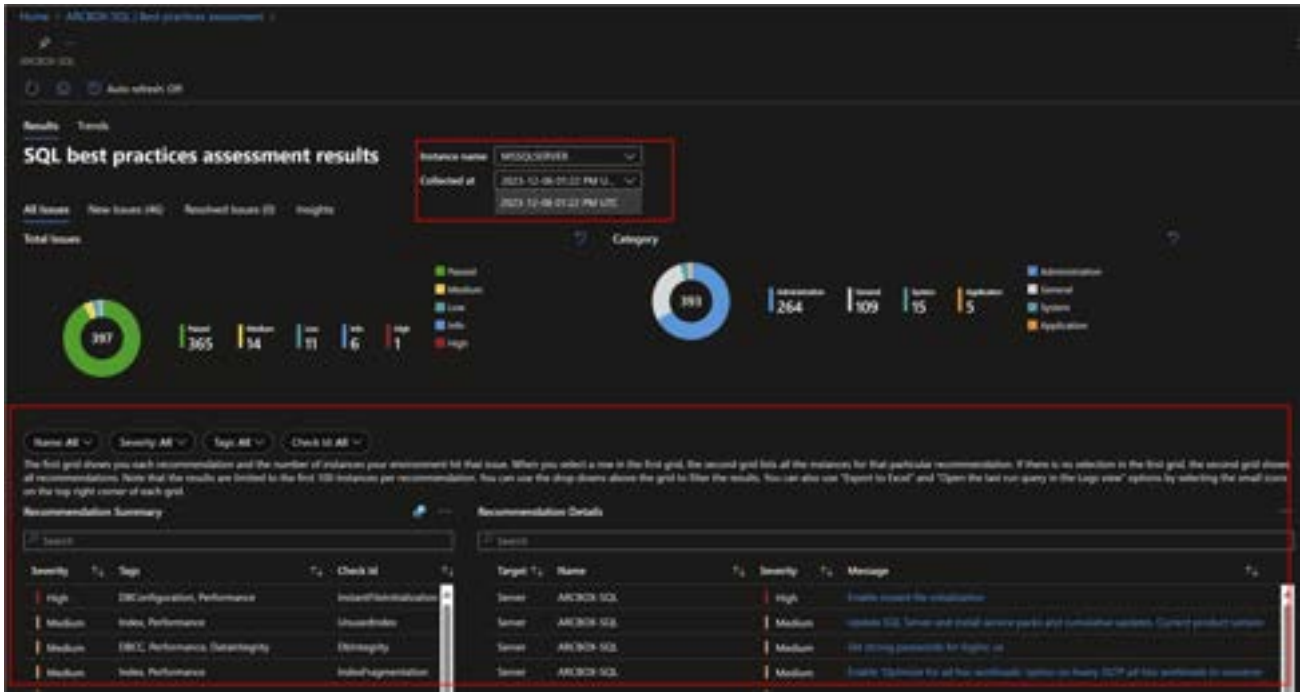
45 minutes

Task 1: View and resolve assessment issues

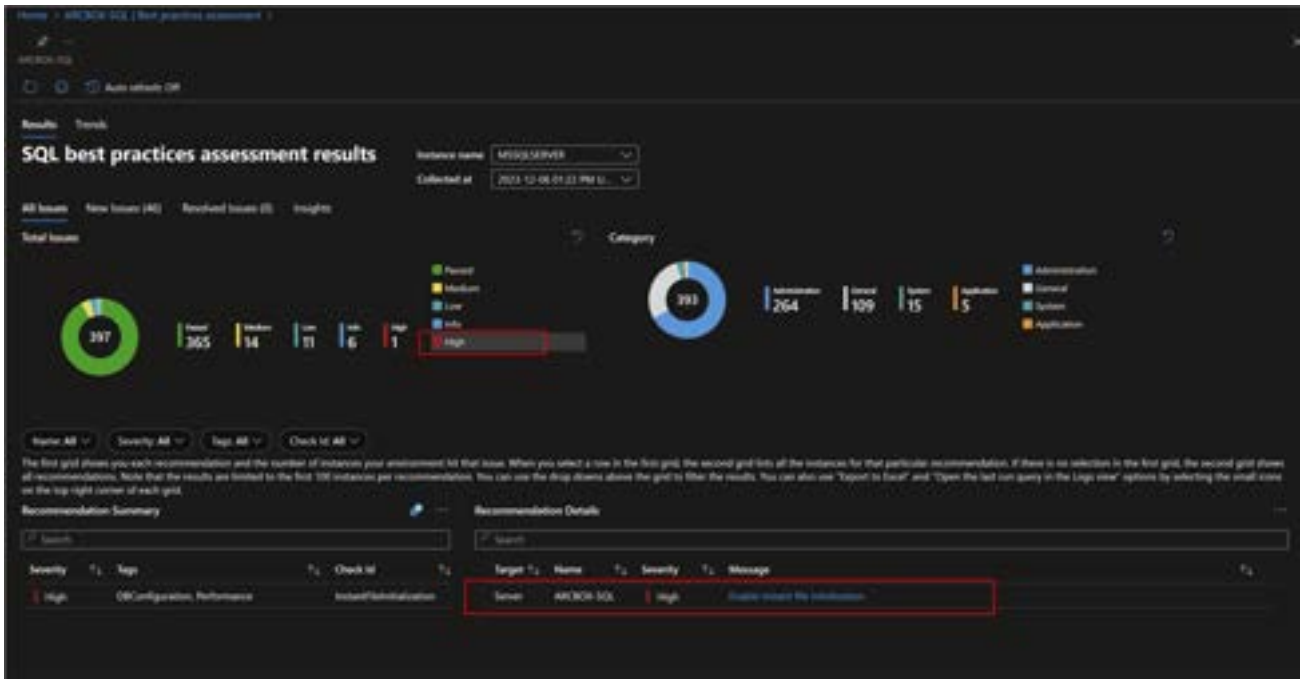
1. Once the assessment is completed, click on the assessment to move to the results screen.



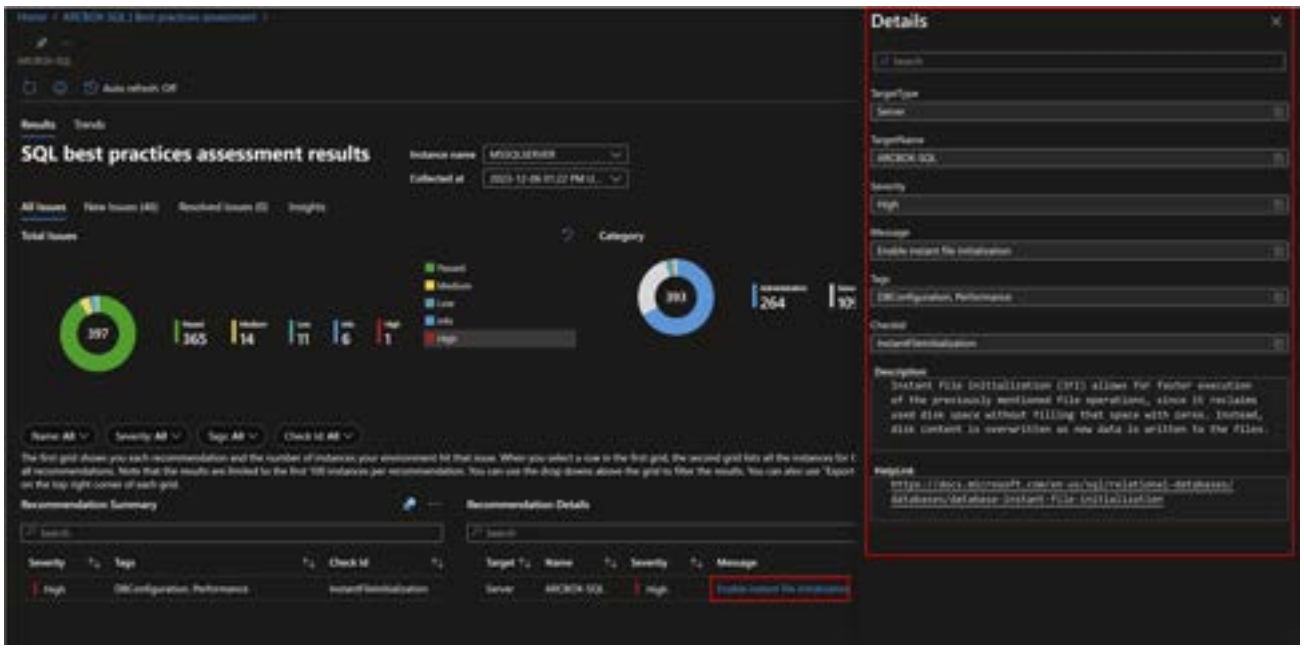
2. On the Results page, if there several assessments that are completed, you can choose from *Collected at* dropdown menu. The Results page reports all the issues categorized based on their severity. The recommendations are organized into All, New and Resolved tabs. The tabs can be used to view all the recommendations from the currently selected run, the newer recommendations compared to the previous run, and the resolved recommendations from the previous runs respectively. The tabs help to keep track of the progress between the runs.



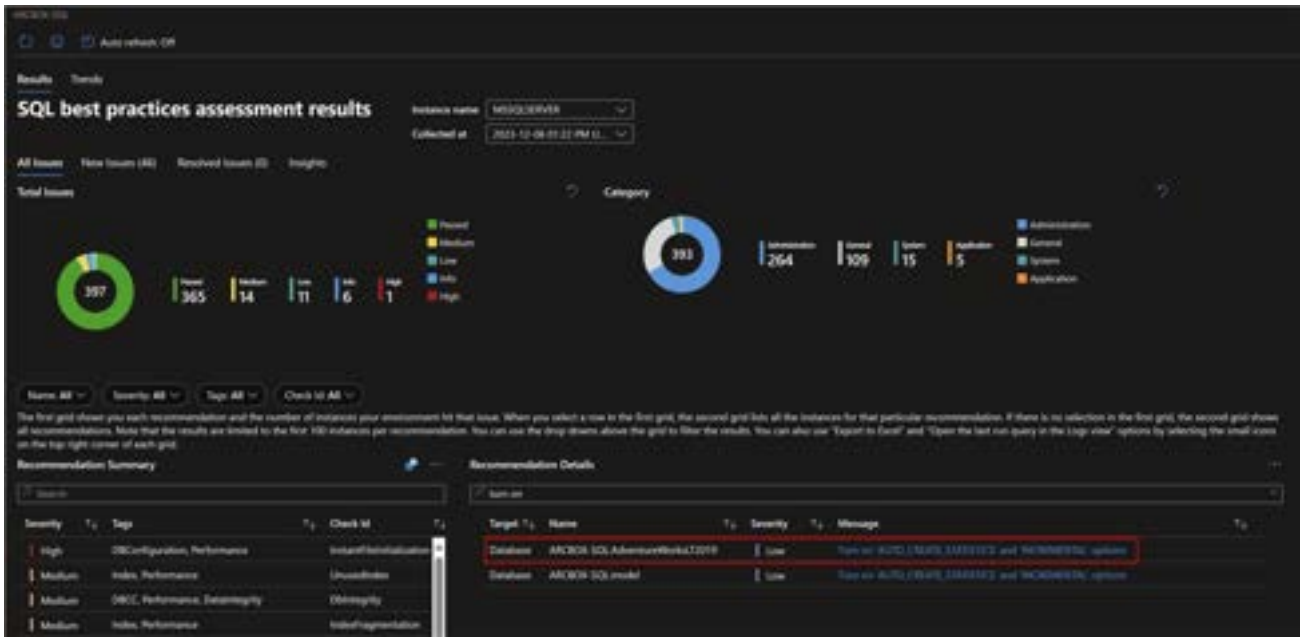
3. To filter the issues by high severity, click on “High” on the Total Issues chart. To clear the filter click on high once again. Filtering can be applied by categories as well.



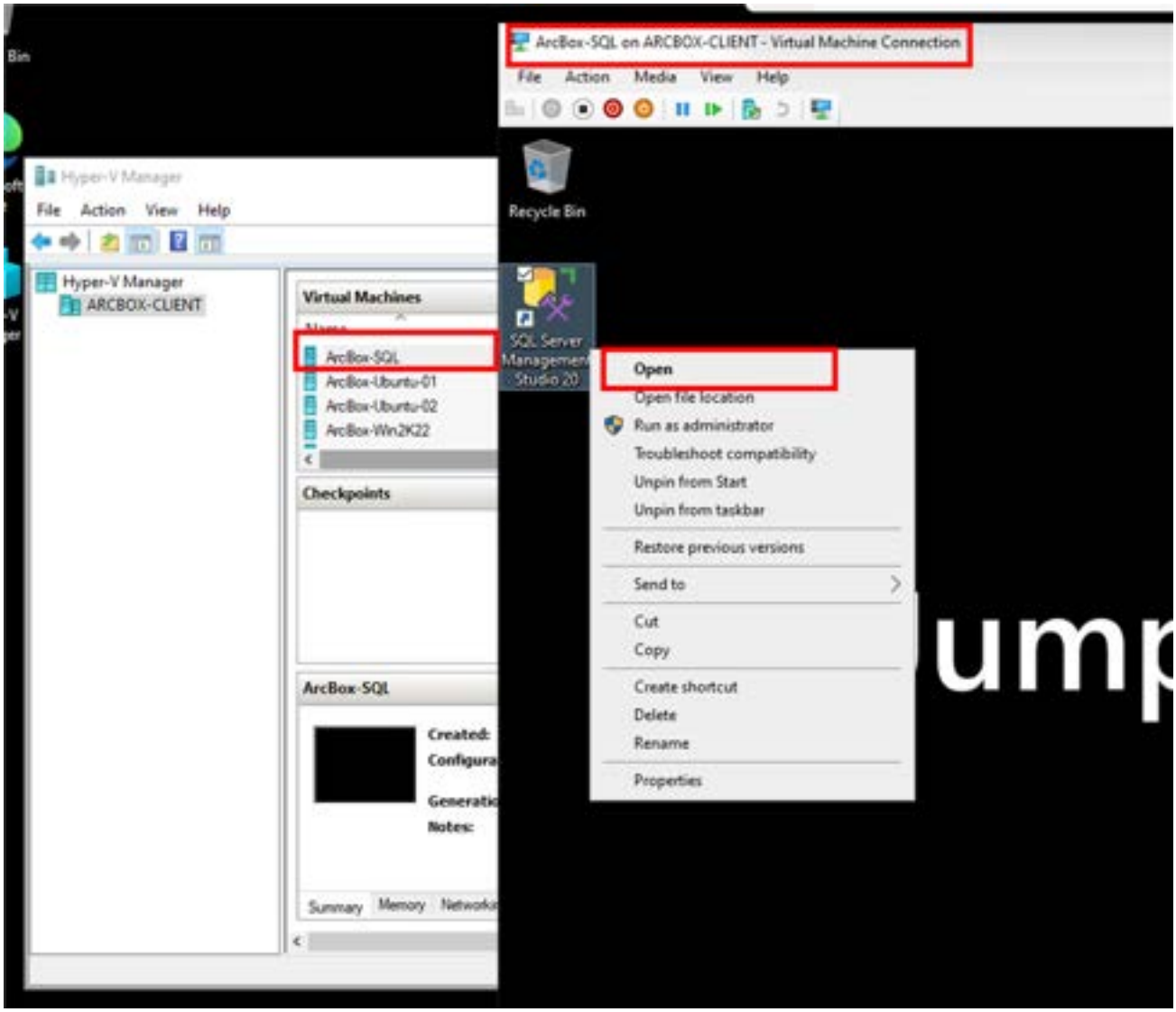
4. To view details for a recommendation, in the recommendation summary menu click on the high severity issue, then in the *Recommendation Details* table click on the message item.



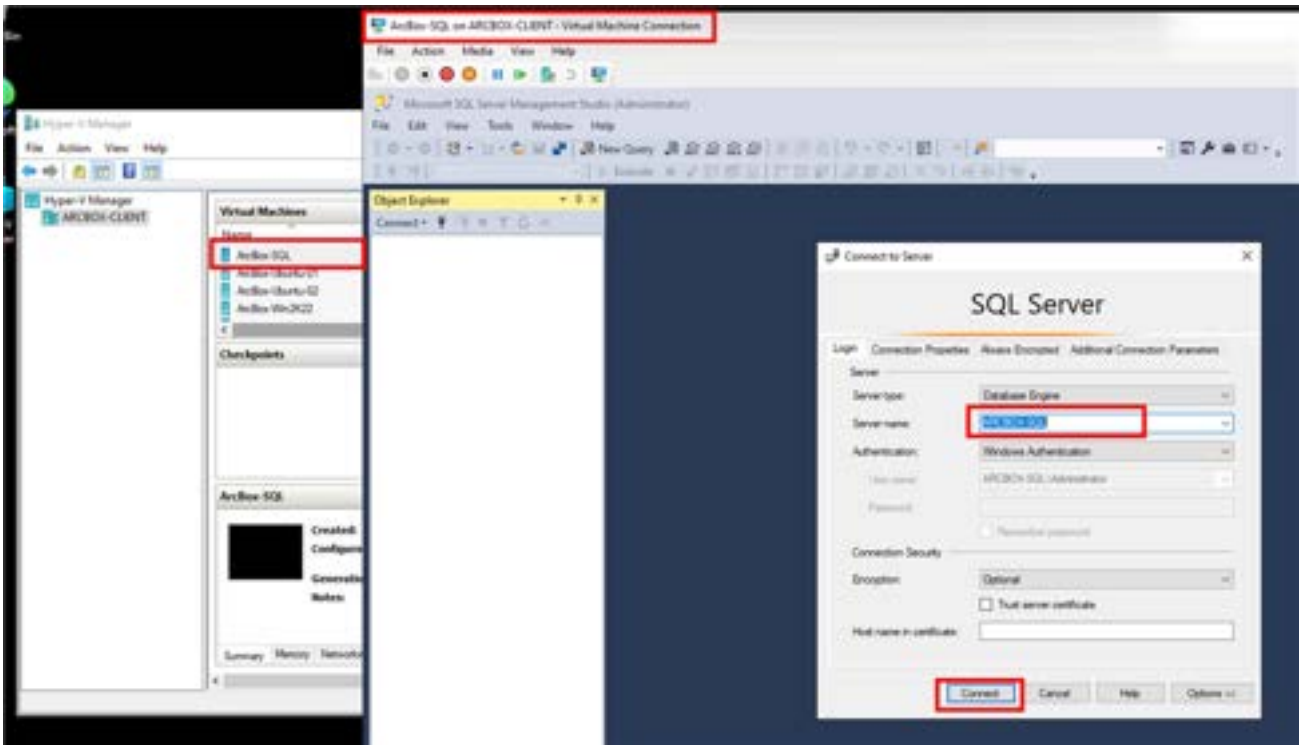
5. You are going to address a specific recommendation now. Remove any filters and scroll through the *Recommendation Details* table (or use the associated search field) to find the issue "Turn on 'Auto_Create_Statistics' and 'Incremental' options" for the *AdventureWorks* database.



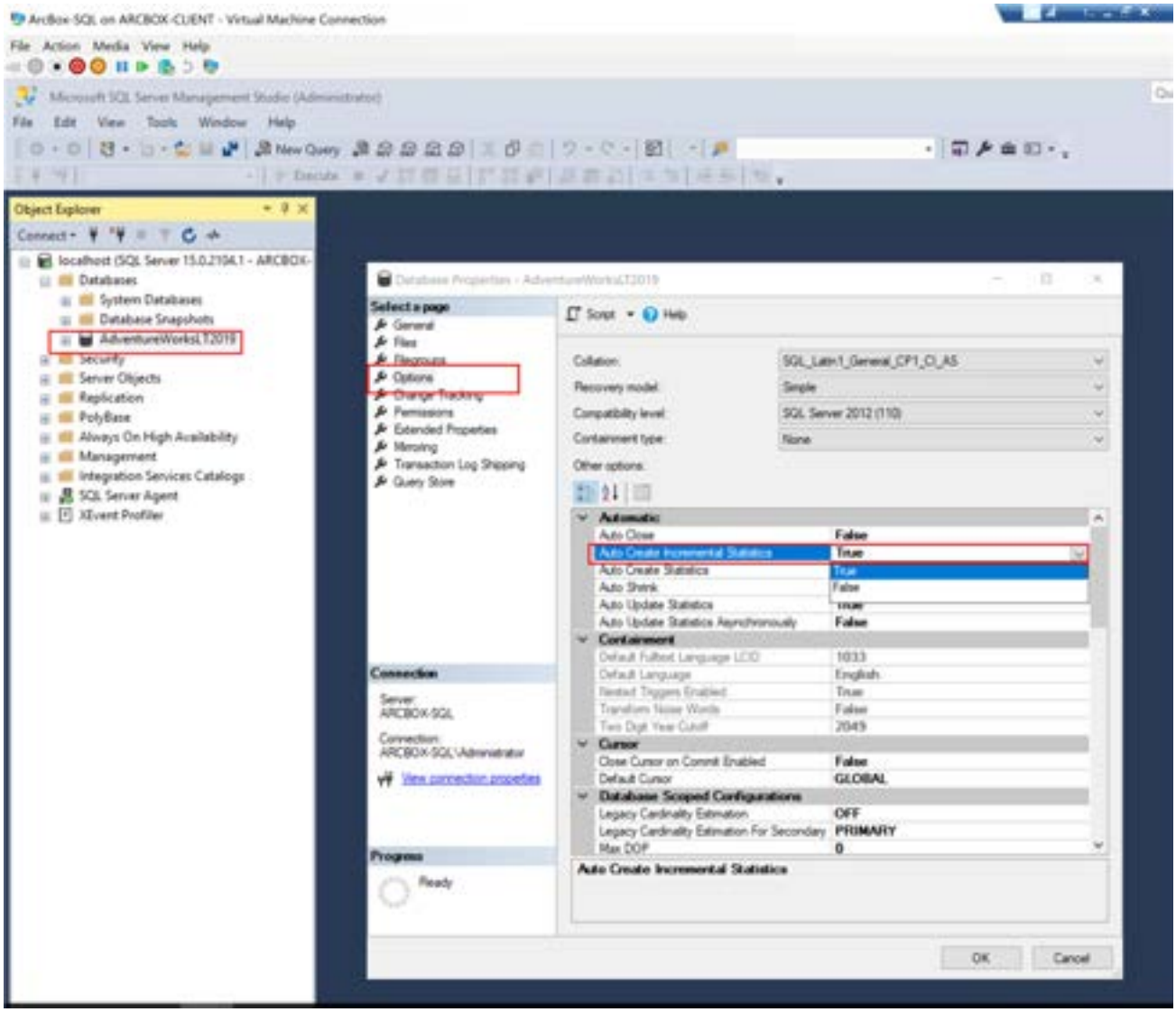
- [ ] 6. To Fix the issue "Turn on 'Auto_Create_Statistics' and 'Incremental' options", logon to the ArcBox-Client_ machine and connect to *ArcBox-SQL* from the Hyper-V console (password *JS123!!*). Start the *SQL Server Management Studio* application and connect to the SQL Server instance.



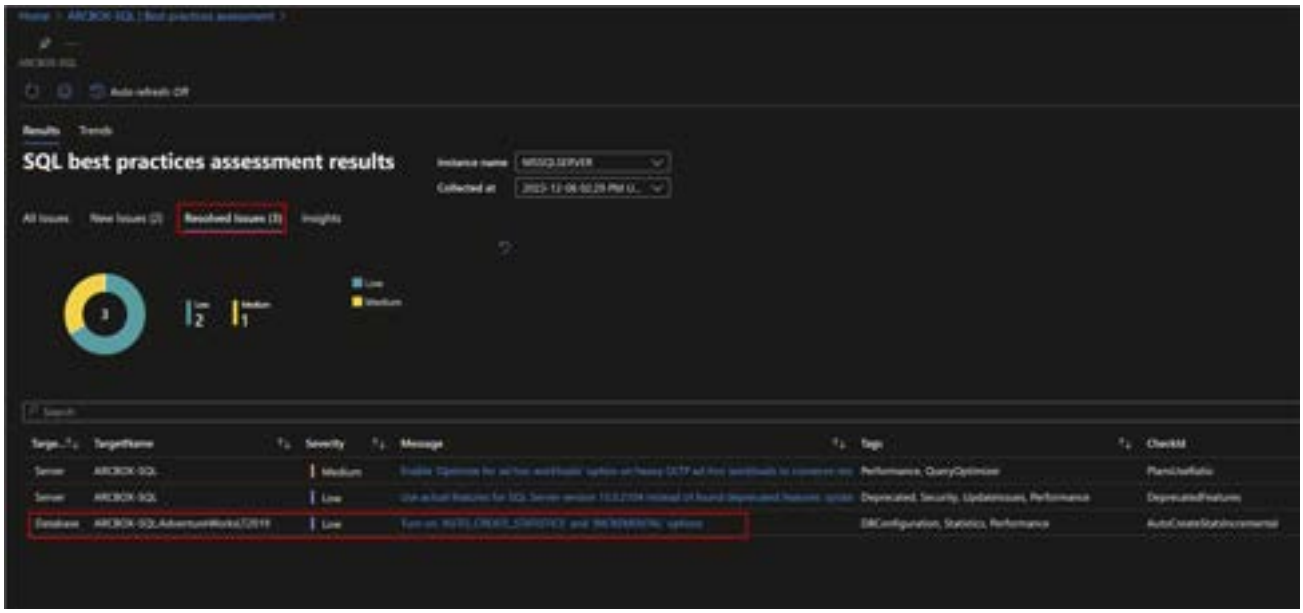
- [ ] 7. Connect to the SQL Server instance



8. Right-Click on the *AdventureWorksLT2019* database and select the *Properties* then *Options*. Find the *Auto Create Incremental Statistics* option and change it from False to True. Confirm your action by clicking *OK*.



- [ ] 9. Rerun assessment and check the resolved issues. (To re-run the assessment, go to the *Best practices assessment* page on the Arc-enabled SQL Server in the Azure portal as you did in the previous exercise). Once the new assessment is complete, go the *Resolved Issues* tab and check that your issues has been resolved.

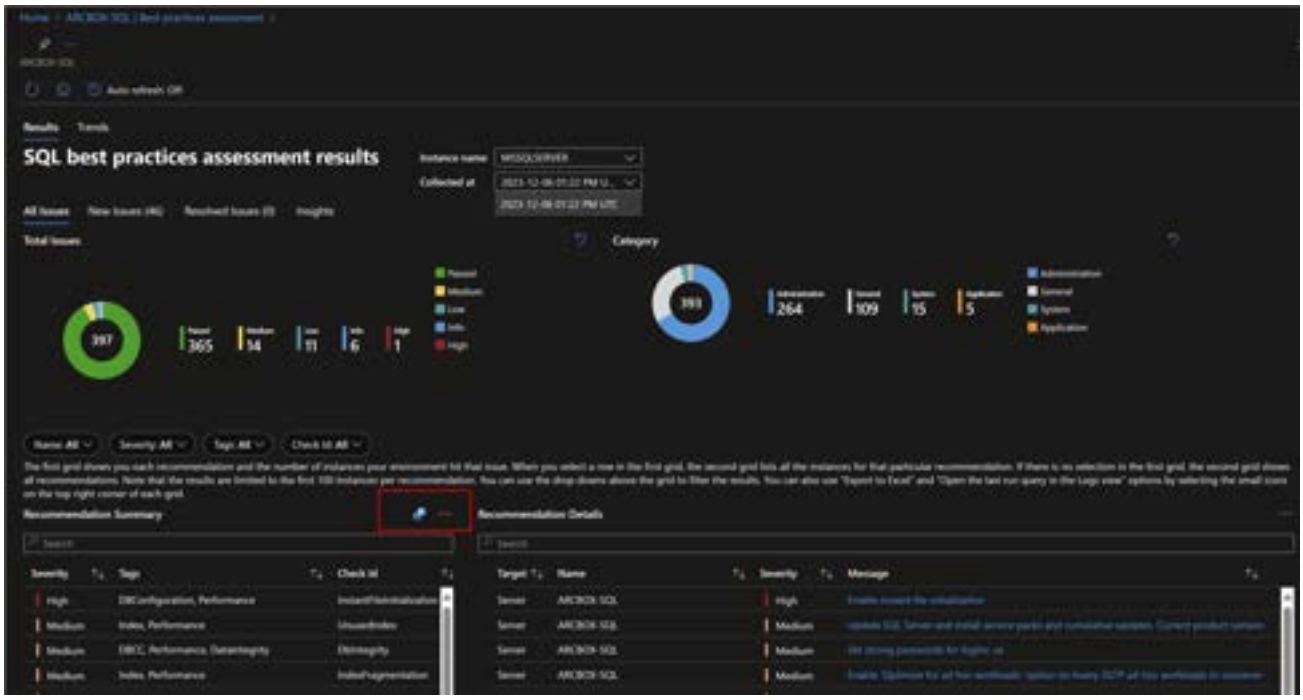


Task 1 has been completed

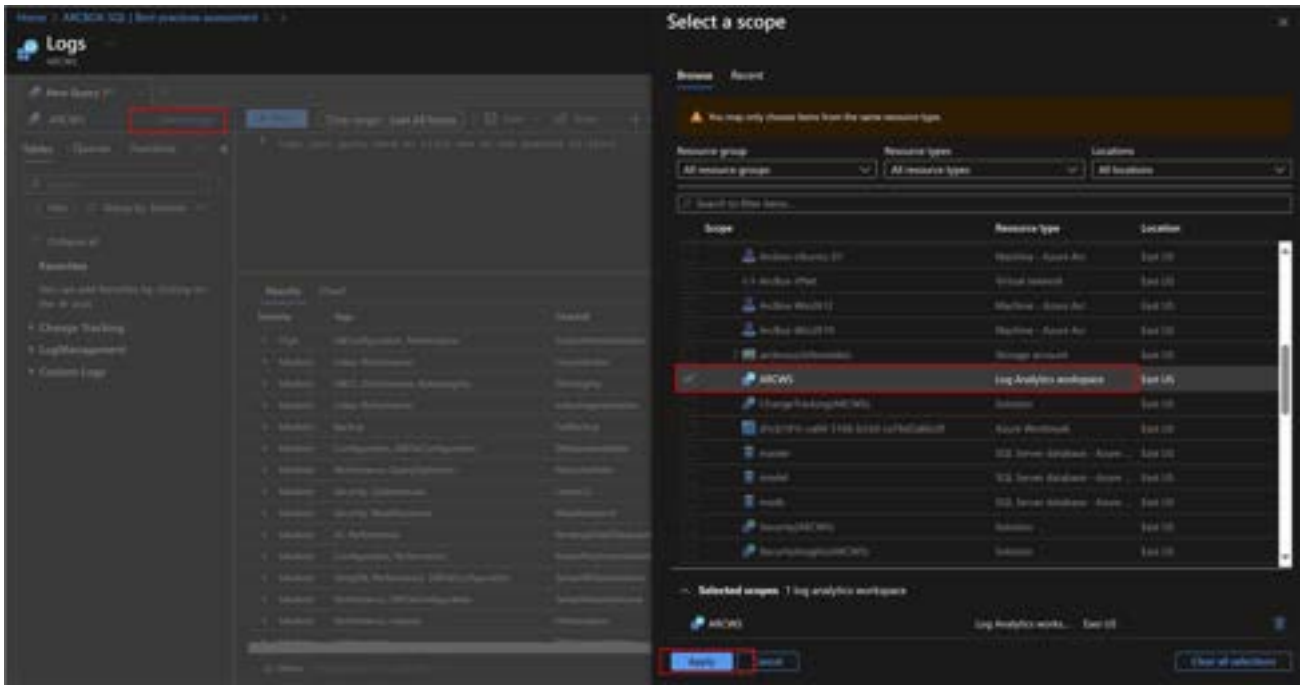
Click **Next** for the next exercise or **Go back to the main table of content**

Task 2: Use Log Queries with Assessment results

1. You can get more detailed insights using Log Analytics Queries. Click on *Query logs* on the top of the grid in assessment results, this will take you to query editor.



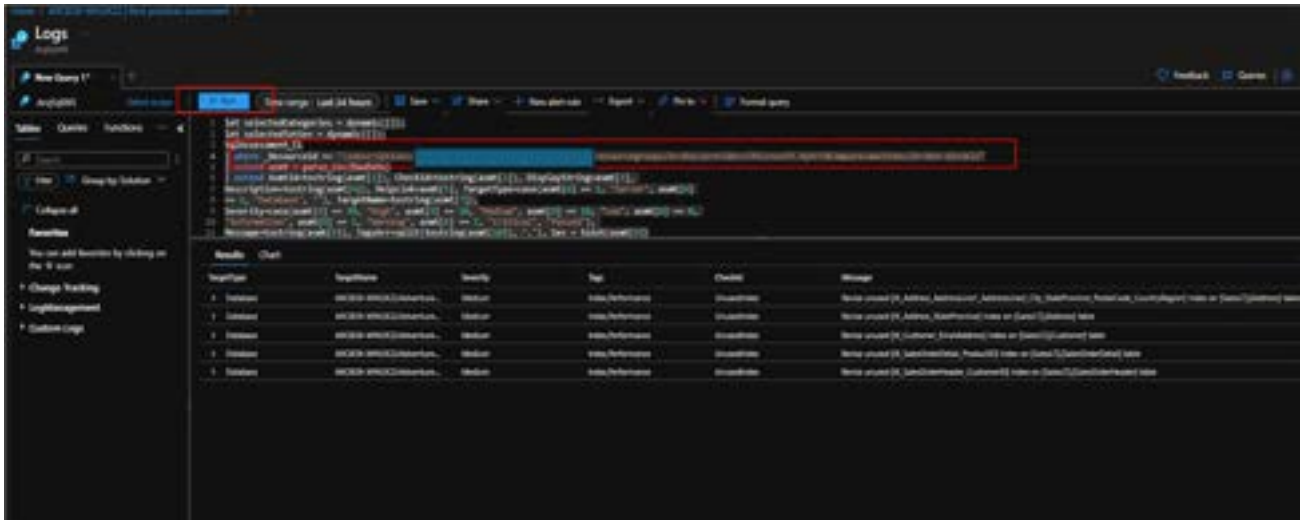
2. Click *Select scope* and then Select the specific "Log Analytics workspace" used for uploading assessment data from Arc-enabled SQL Server and click *Apply*.



3. Query the logs to gain more insights into your SQL Server environment. For example, you can use the following query to identify all unused indexes in a specific instance. Enter the Subscription Id

and Resource group name in the forth line of the query then paste the query into the Query pane and click the *Run* button. Then view the results of your query.

```shell
let selectedCategories = dynamic([]);
let selectedTotSev = dynamic([]);
SqlAssessment_CL
| where _ResourceId =~ "/subscriptions/XXXXXXXXXXXX/resourcegroups/XXXX/provider
| extend asmt = parse_csv(RawData)
| extend AsmtId=tostring(asmt[1]), CheckId=tostring(asmt[2]), DisplayString=asmt
Description=tostring(asmt[4]), HelpLink=asmt[5], TargetType=case(asmt[6] == 1, "
== 2, "Database", ""), TargetName=tostring(asmt[7]),
Severity=case(asmt[8] == 30, "High", asmt[8] == 20, "Medium", asmt[8] == 10, "Lo
"Information", asmt[8] == 1, "Warning", asmt[8] == 2, "Critical", "Passed"),
Message=tostring(asmt[9]), TagsArr=split(tostring(asmt[10]), ","), Sev = toint(a
| where CheckId == "UnusedIndex"
| project
TargetType,
TargetName,
Severity,
Tags=strcat_array(array_slice(TagsArr, 1, -1), ','),
CheckId,
Message
| distinct *
```



Task 2 has been completed

Click **Next** for the next lab or **Go back to the main table of content**

LAB16: Protect your Arc-enabled SQL Server with Microsoft Defender for Cloud

---

Student Lab Manual

Table of Contents

Exercise 1 - Enable Defender for Cloud for SQL Server on machines

Objective

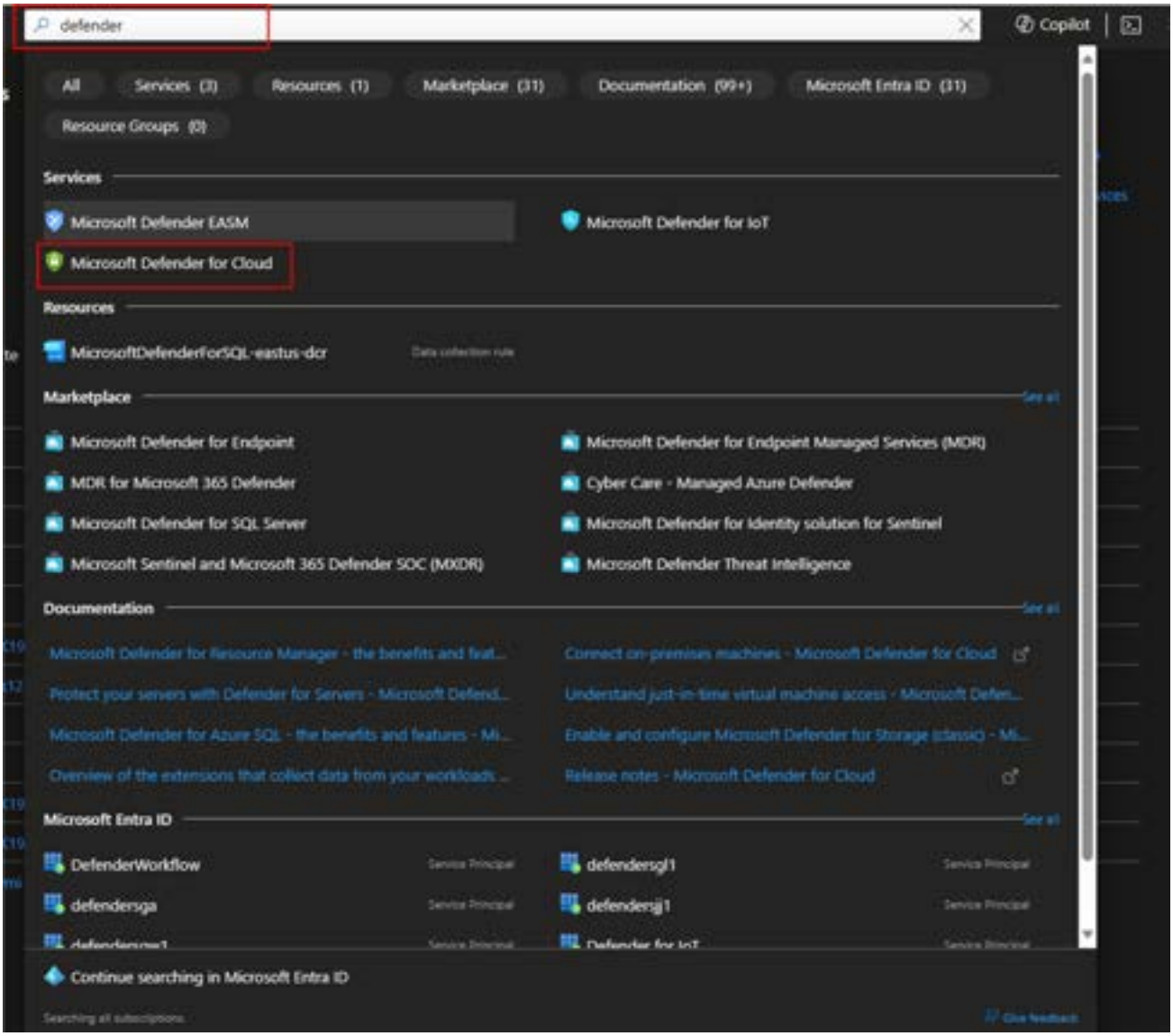
In this exercise you will learn how to enable the Azure Defender for Cloud service to protect your Arc-enabled SQL server.

Estimated Time to Complete This Exercise

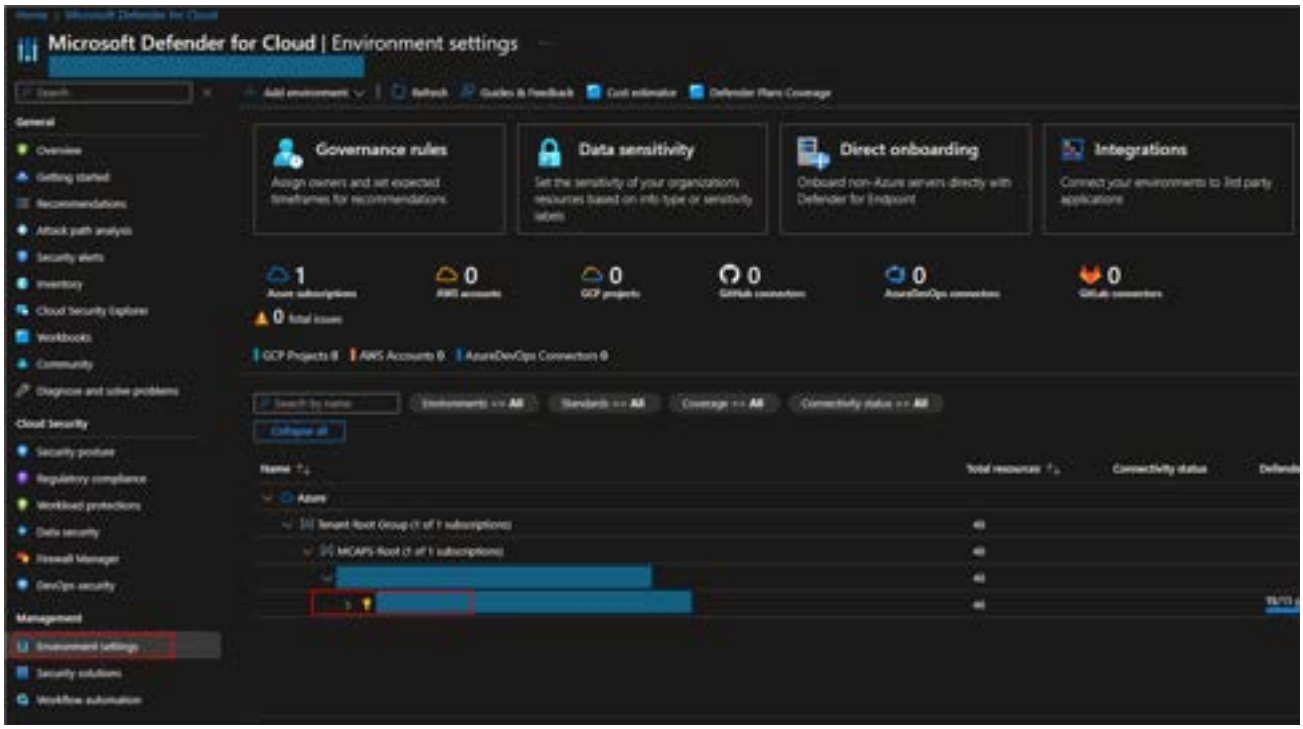
30 minutes

Task 1: Enable Defender for Cloud to protect your Arc-enabled SQL servers at the subscription level

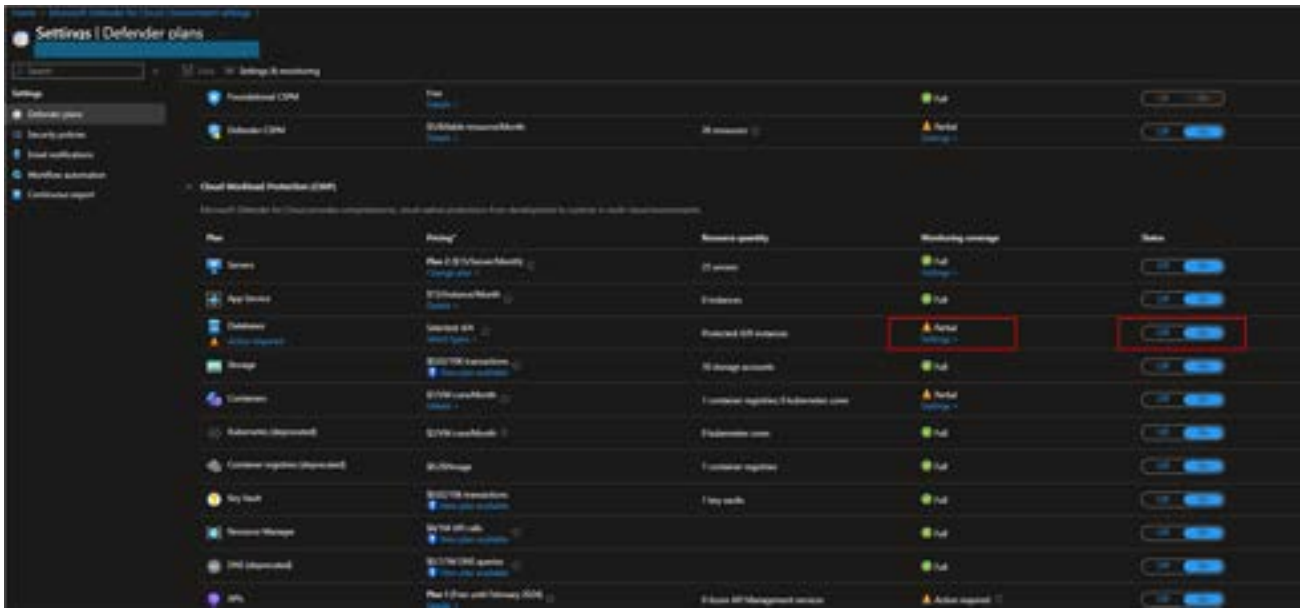
- [ ] 1. In the Azure Portal search for *Microsoft Defender for CLoud* and select it from the search results



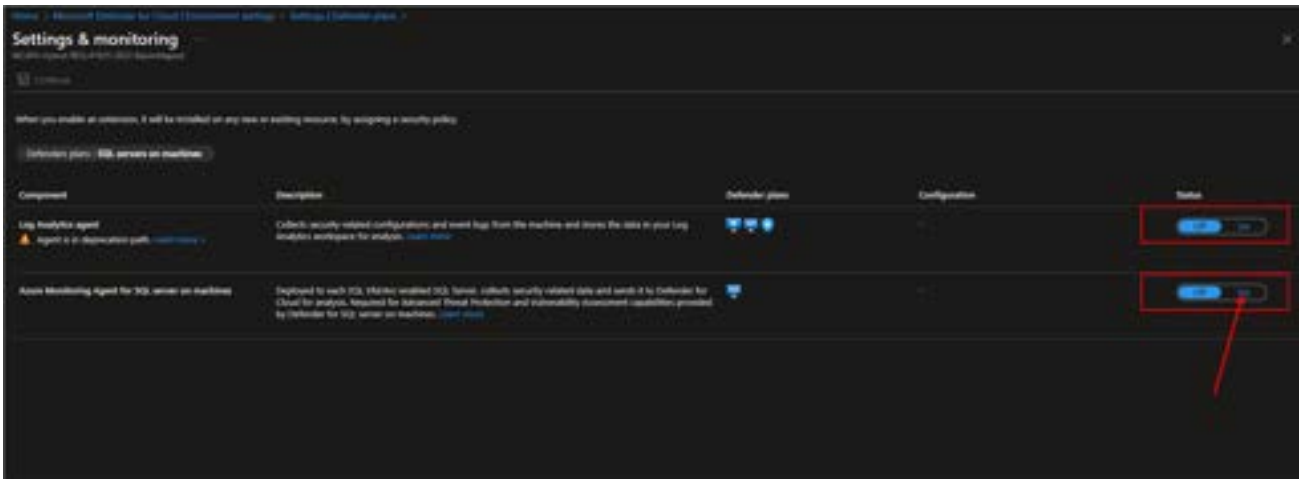
- [ ] 2. From the *Microsoft Defender for Cloud* page select *Environment settings* and under the hierarchy select your subscription



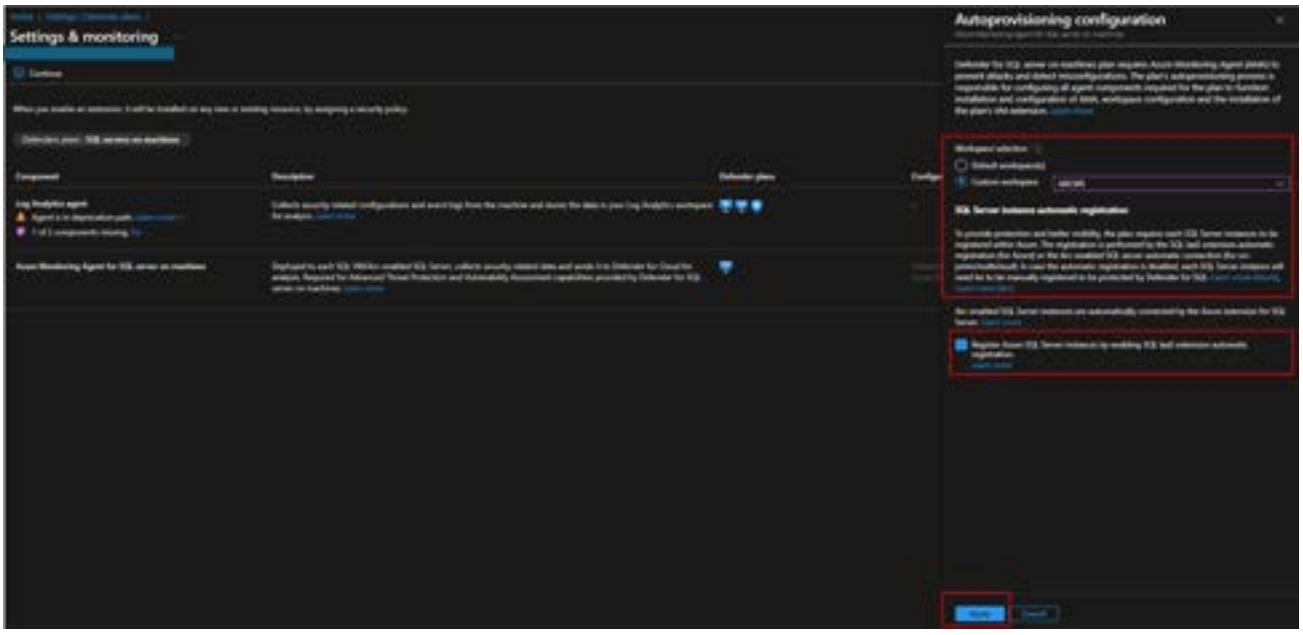
☐ 3. On the subscription Defender plans settings set the *Status* to *On* if it is not set already. Then click on *Settings* under the *Monitoring coverage* column



☐ 4. On the *Settings & monitoring* page set the *Status of the _Azure Monitoring Agent for Sql server on machines* to *On*

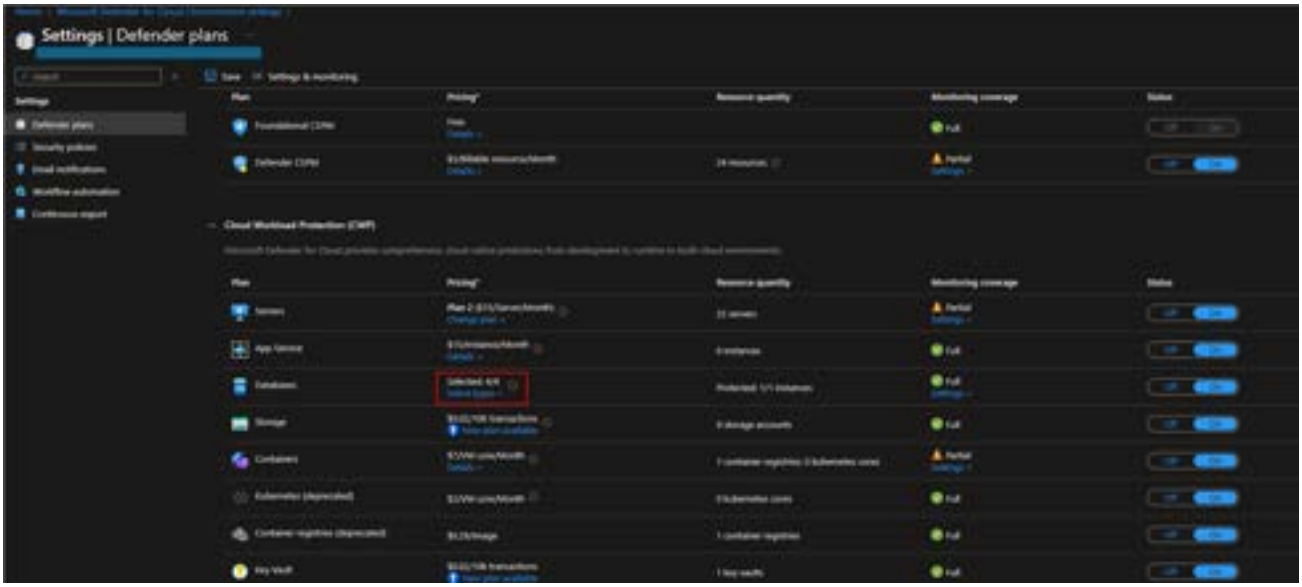


☐ 5. On the *Autoprovisioning configuration* screen select the Log Analytics workspace, tick the option to *Register Azure SQL Server instances by enabling SQL IaaS extension automatic registration* then click *Apply*

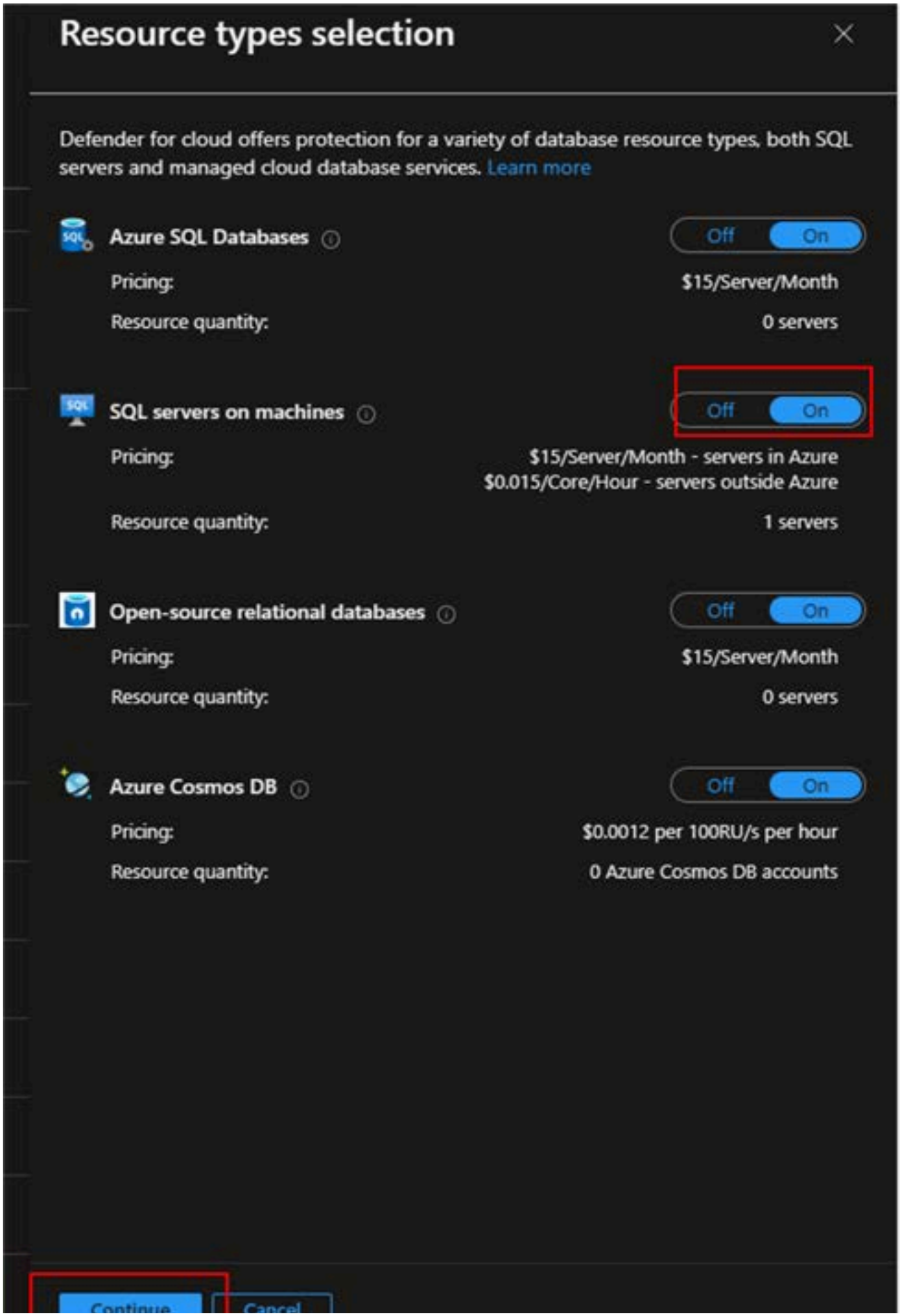


☐ 6. Select *Continue* to save the settings

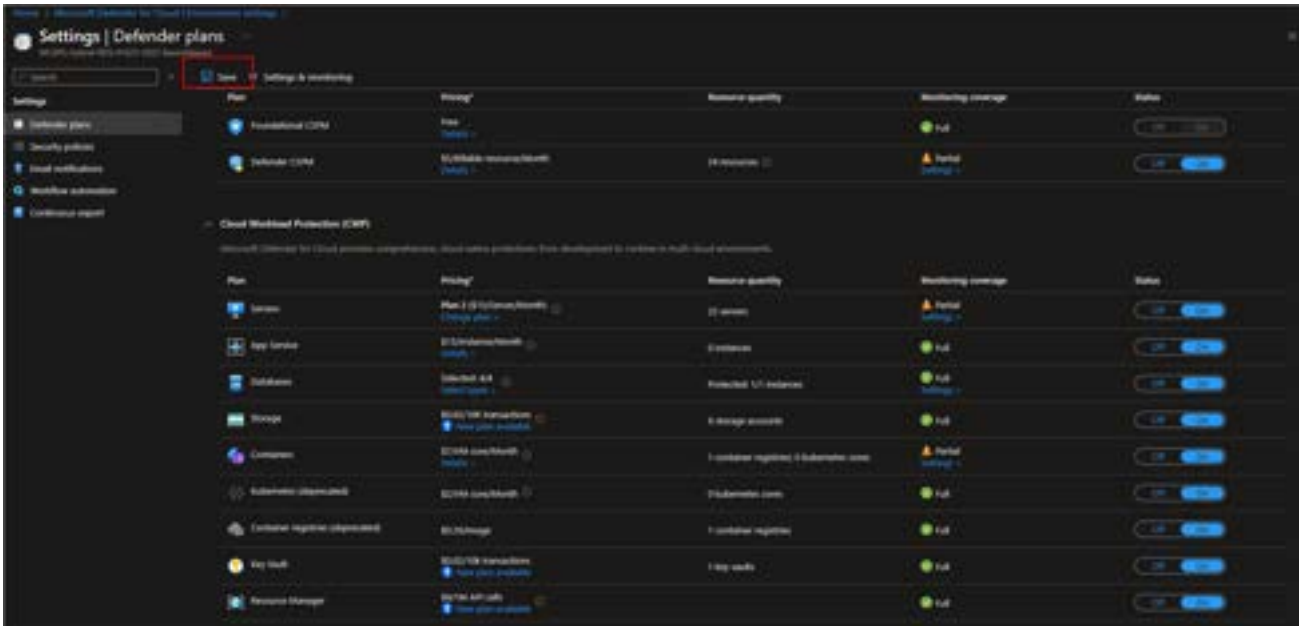
7. Once you are back on the Defender plan settings page, click on *Select types* under the *Pricing* column



8. Make sure that the *SQL servers on machines* option is set to *On*, then click *Continue*


Resource types selection
Defender for cloud offers protection for a variety of database resource types, both SQL servers and managed cloud database services. Learn more
Azure SQL Databases
Off
On
Pricing:
$15/Server/Month
Resource quantity:
0 servers
SQL servers on machines
Off
On
Pricing:
$15/Server/Month - servers in Azure
$0.015/Core/Hour - servers outside Azure
Resource quantity:
1 servers
Open-source relational databases
Off
On
Pricing:
$15/Server/Month
Resource quantity:
0 servers
Azure Cosmos DB
Off
On
Pricing:
$0.0012 per 100RU/s per hour
Resource quantity:
0 Azure Cosmos DB accounts
Continue
Cancel

9. Save the Settings for Defender



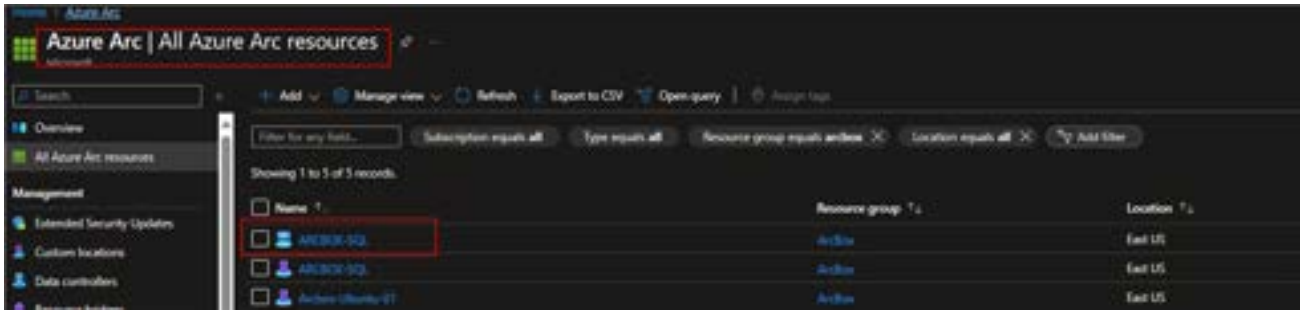
Task 1 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

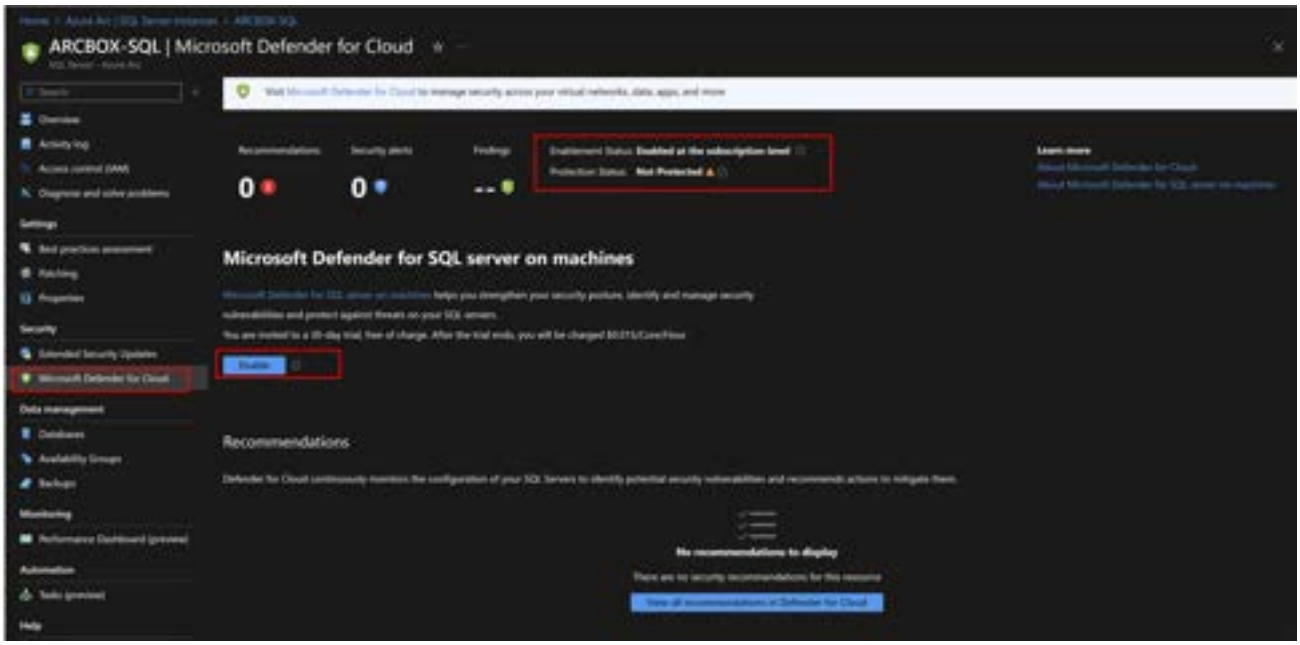
Task 2: Enable Defender for Cloud to protect a specific Arc-enabled SQL server

In the previous task you set up the Defender for Cloud to protect your SQL servers at the subscription level, by setting up a policy to ensure the required extensions are installed on all SQL servers. In this step you will look at enabling Defender for Cloud to protect a single SQL Server.

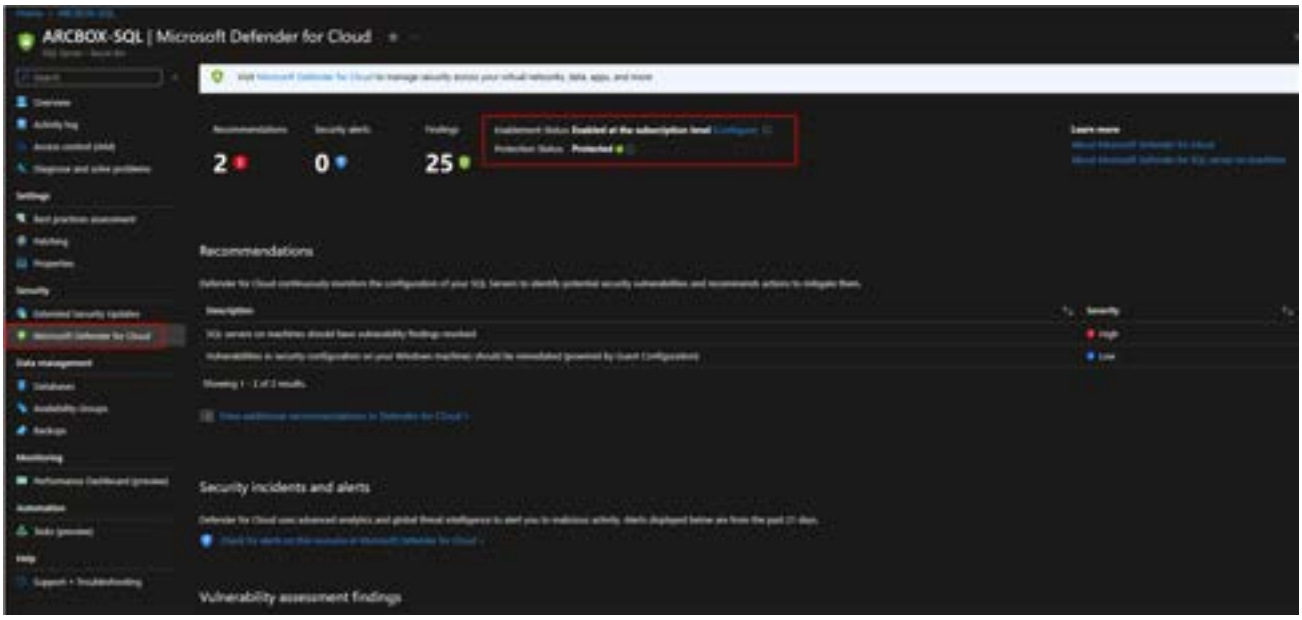
1. From the Azure portal go to *Azure Arc* | *All Azure Arc resources* page and select your SQL Server.



2. On the Server page select *Microsoft Defender for Cloud* and check if Defender is enabled. If it is not enabled then click the *Enable* button. Note that although you set up Defender to protect your SQL servers at the subscription level in the previous task, the effect of that policy setting might not have propagated yet to your specific SQL server



3. Defender for Cloud is now enabled for your SQL server. There might be some delay in showing the *Protection Status* but you can continue with the rest of the lab



Task 2 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

Exercise 2 - Generate Defender for Cloud Incidents and Alerts

Objective

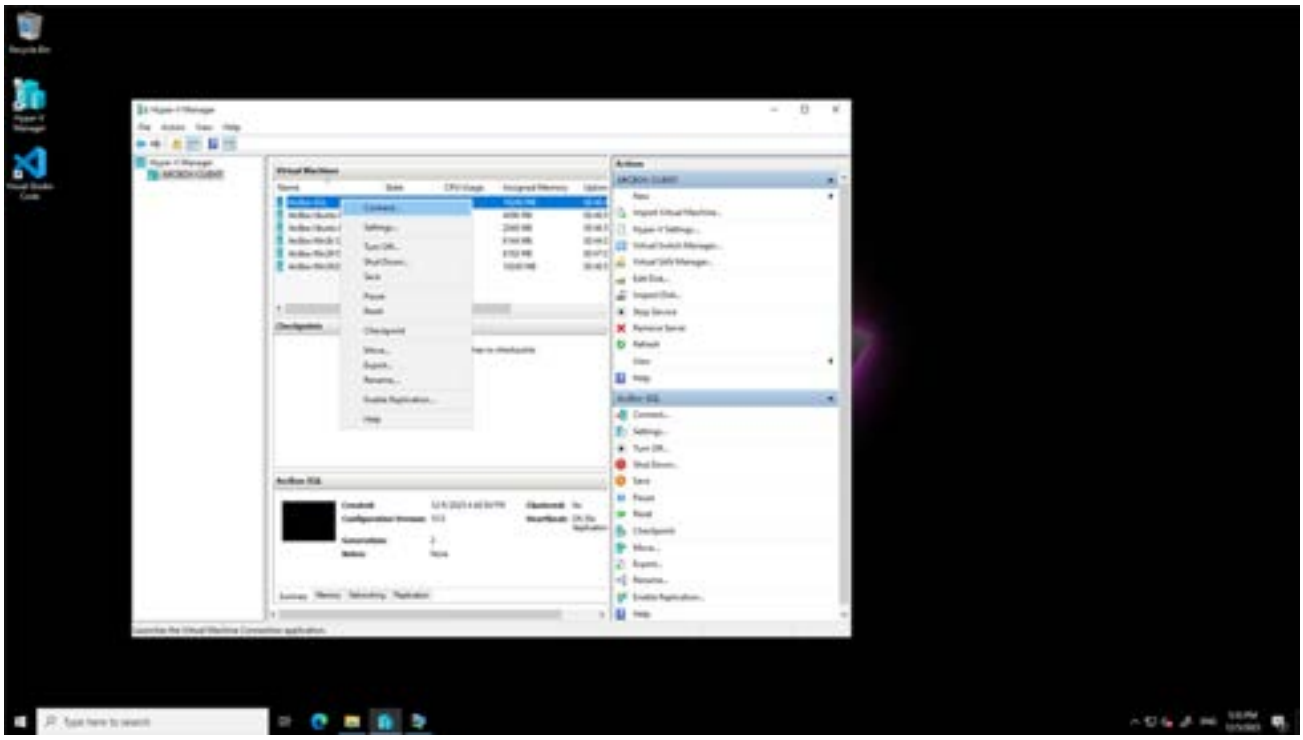
In this exercise you will simulated a number of security incidents on your Arc-enabled SQL server and examine the alerts generated by Defender for Cloud.

Estimated Time to Complete This Exercise

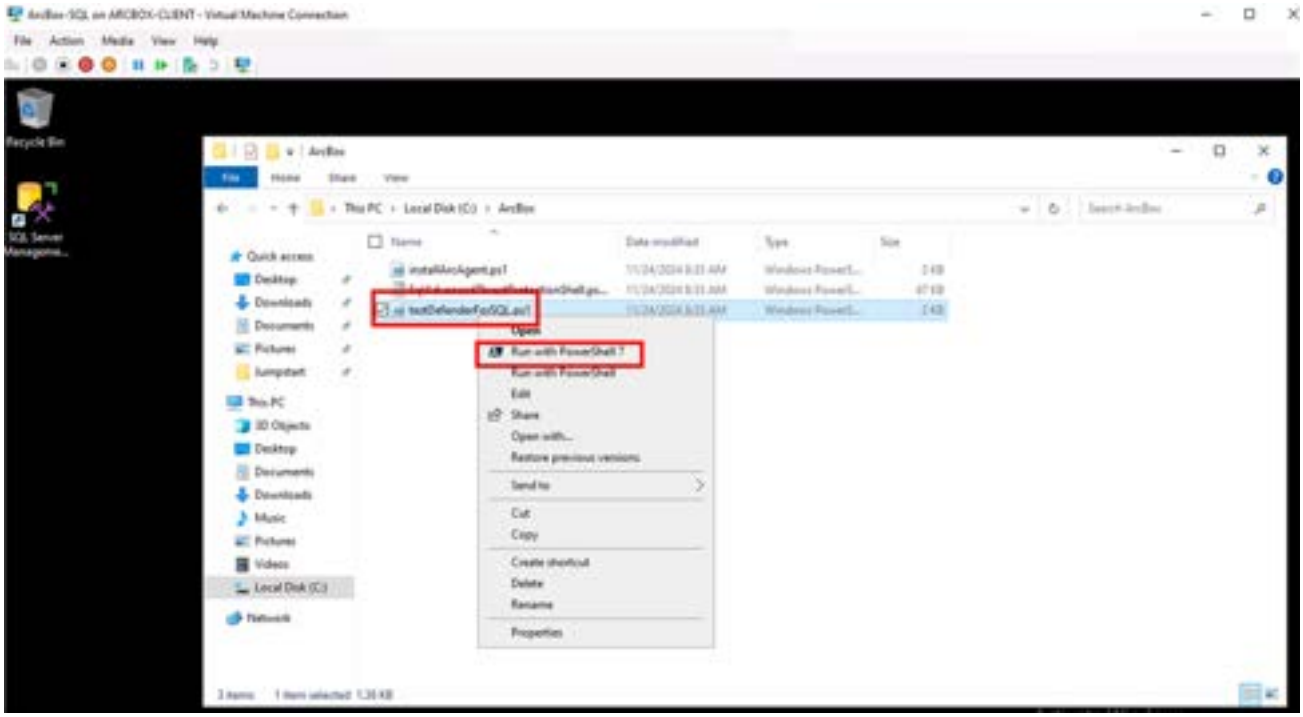
30 minutes

Task 1: Simulate security incidents

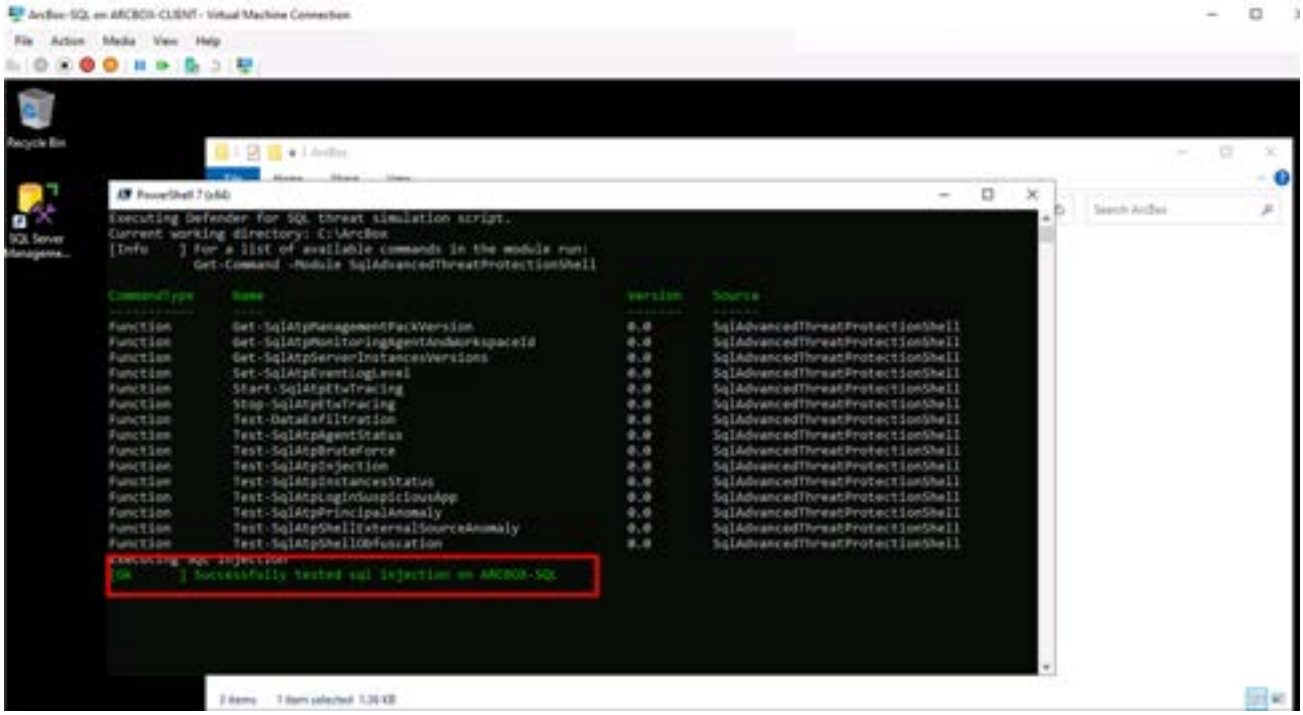
1. Connect and login to the ArcBox-Client machine. Open Hyper-V Manager and connect to the *ArcBox-SQL* virtual machine (Administrator password *JS123!!*)

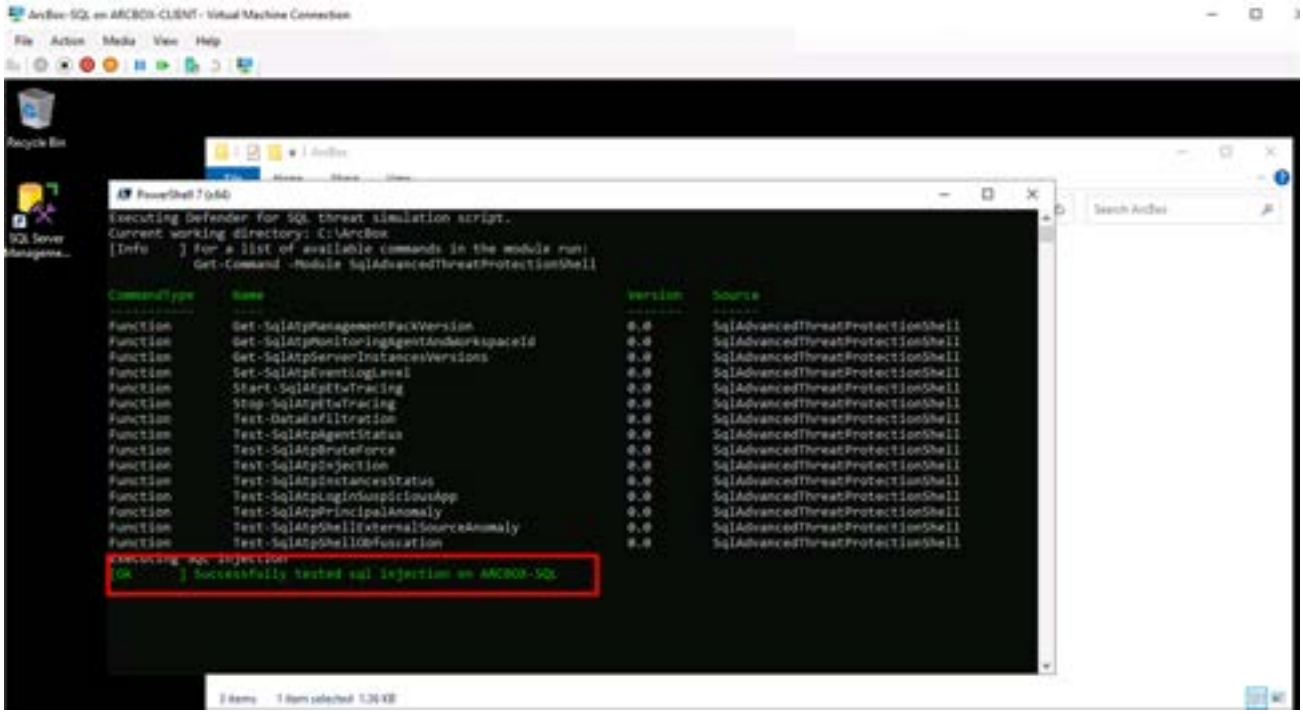


2. **On the nested *ArcBox-SQL*** Open the file explorer and navigate to the *ArcBox* folder on the *C:* drive. Right click on the *testDefenderForSQL.ps1* file and run with PowerShell.

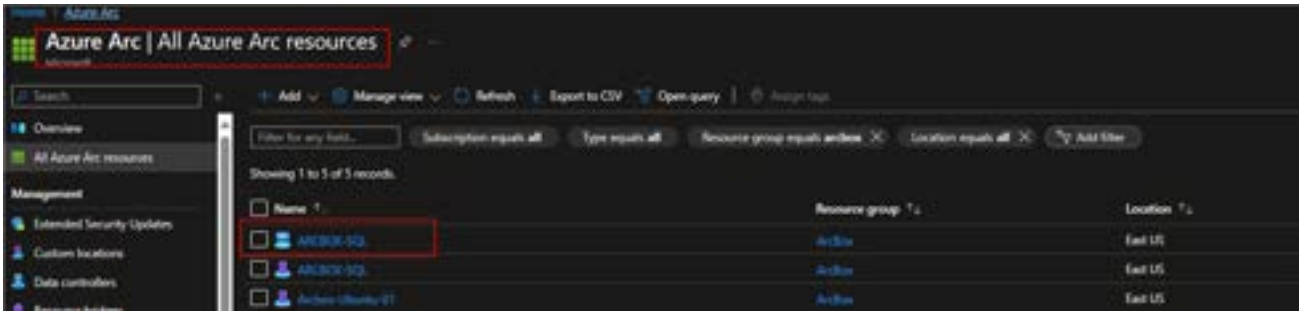


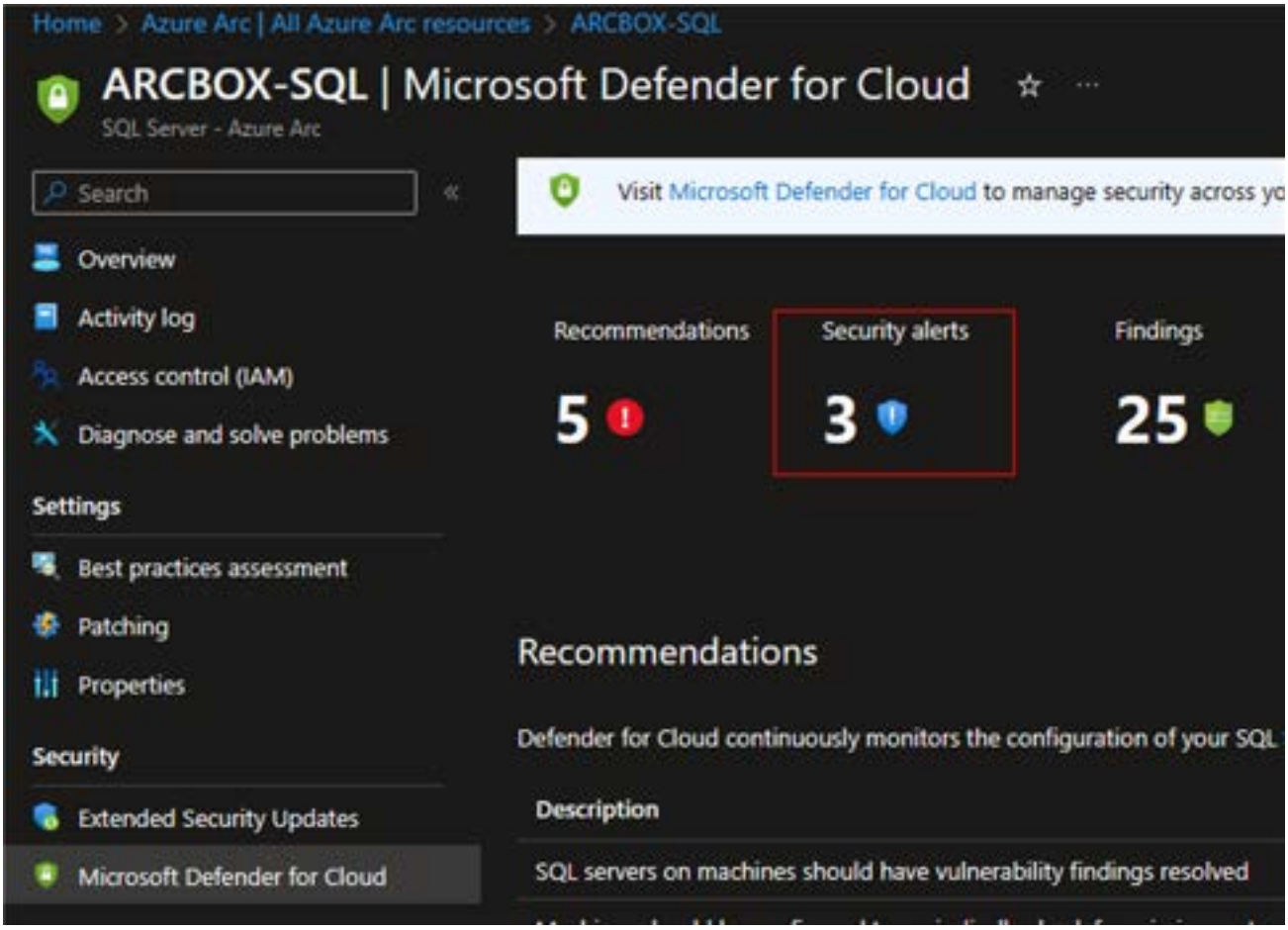
3. The script will simulate a number of threats on the Arc-enabled SQL server.



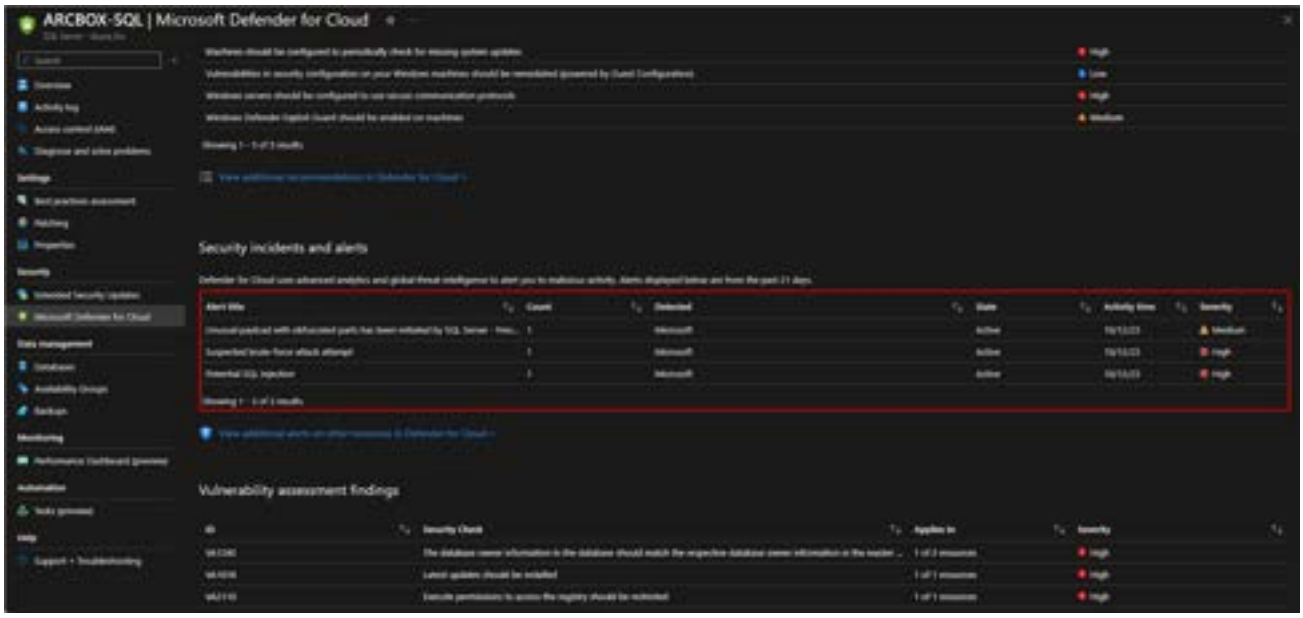


4. Go to your Arc SQL Server *ArcBox-SQL* and select *Microsoft Defender for Cloud*. You should see that some alerts have been generated

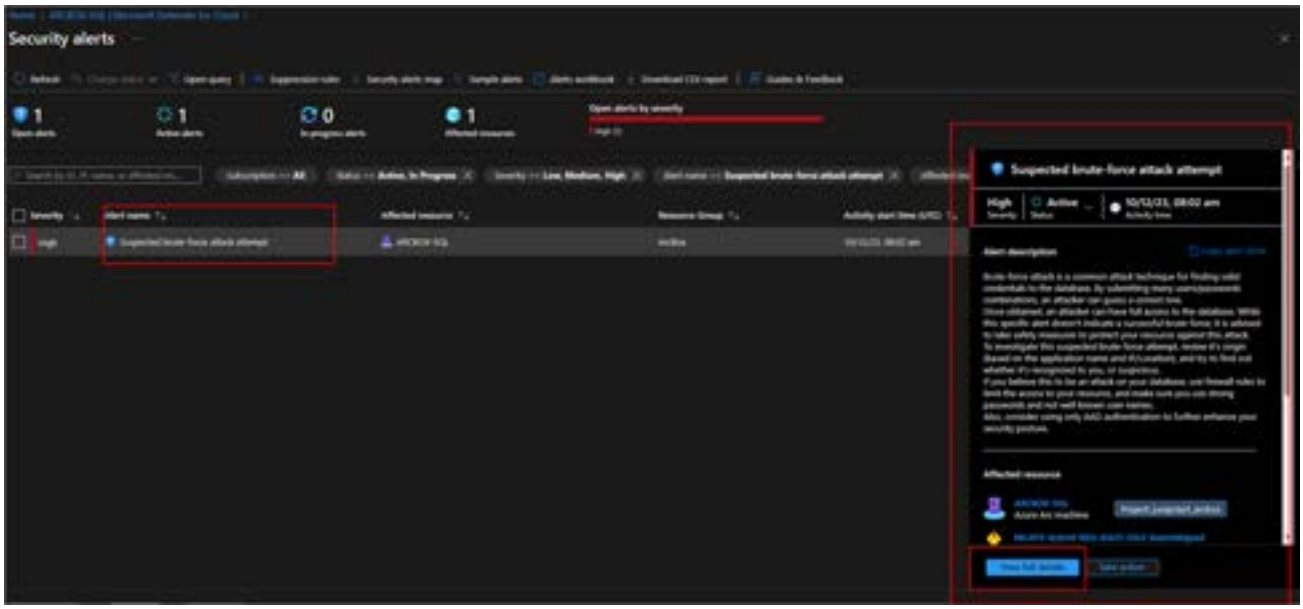




☐ 5. You can scroll down further to find your alerts

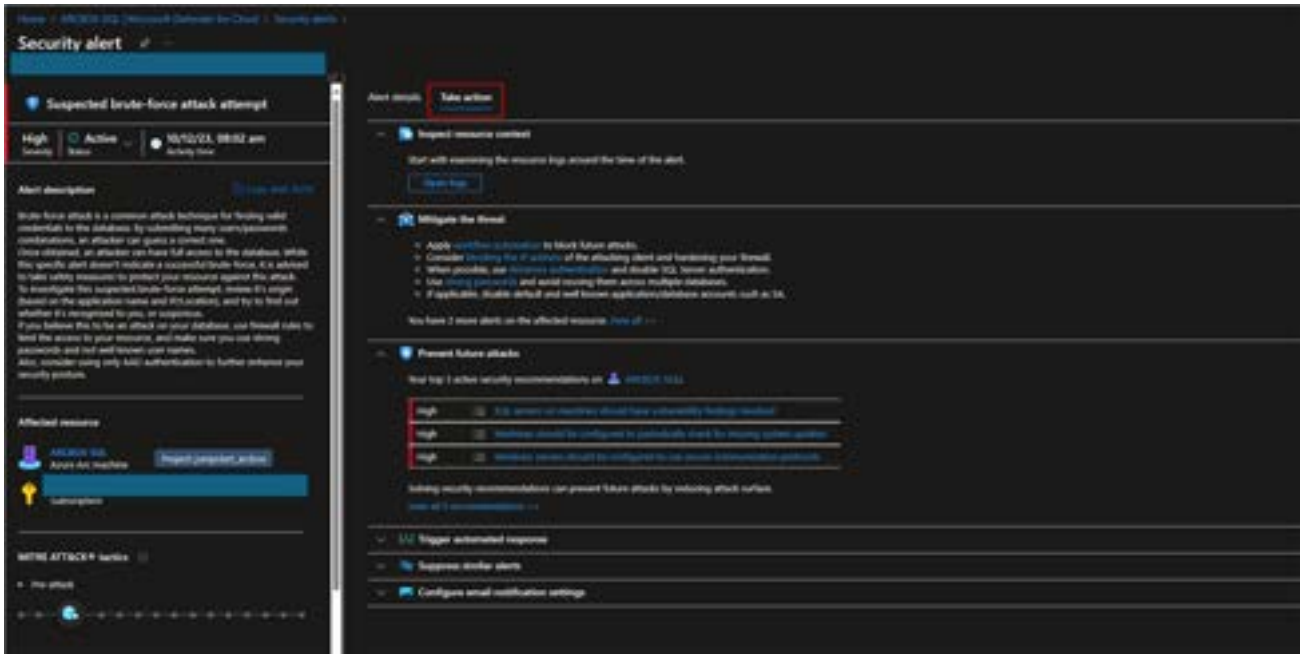


- [ ] 6. Click on the *Suspected brute-force attack attempt* and on the next screen click on one of the instances of *Suspected brute-force attack attempt* (if there is more than one). This will open a screen with the alert description



- [ ] 7. Click *View full details* to see more information on the user, client ip address and cause

- [ ] 8. Click on *Take action* to get the option to review logs as well as information to help mitigate the attack and prevent future attacks. You can also trigger a response, suppress alerts or configure email notifications



- [ ] 9. Optionally repeat the alert investigation for the other alerts (e.g. *Potential SQL Injection* alert)

Task 2 has been completed

Click **Next** for the next exercise or **Go back to the main table of content**

LAB17: Select the optimal Azure SQL target using Migration Assessment

In this lab you will learn how to use the Arc-enabled SQL Server assessment for migration to Azure. This assessment plays a vital role in the success of your cloud migration and modernization journey.

The assessment aggregates all the configuration and performance data and tries to find the best match across various Azure SQL service tiers and configurations and picks a configuration that can match or exceed the SQL instance performance requirements, optimizing the cost.

Student Lab Manual

Table of Contents

Exercise 1: Review and analyze the migration assessment results

Objective

In this exercise you will review the readiness for target deployment types and the Azure SQL size recommendation. The assessment takes into consideration three migration targets; Azure SQL Managed Instances, SQL Server on Azure VM and Azure SQL database.

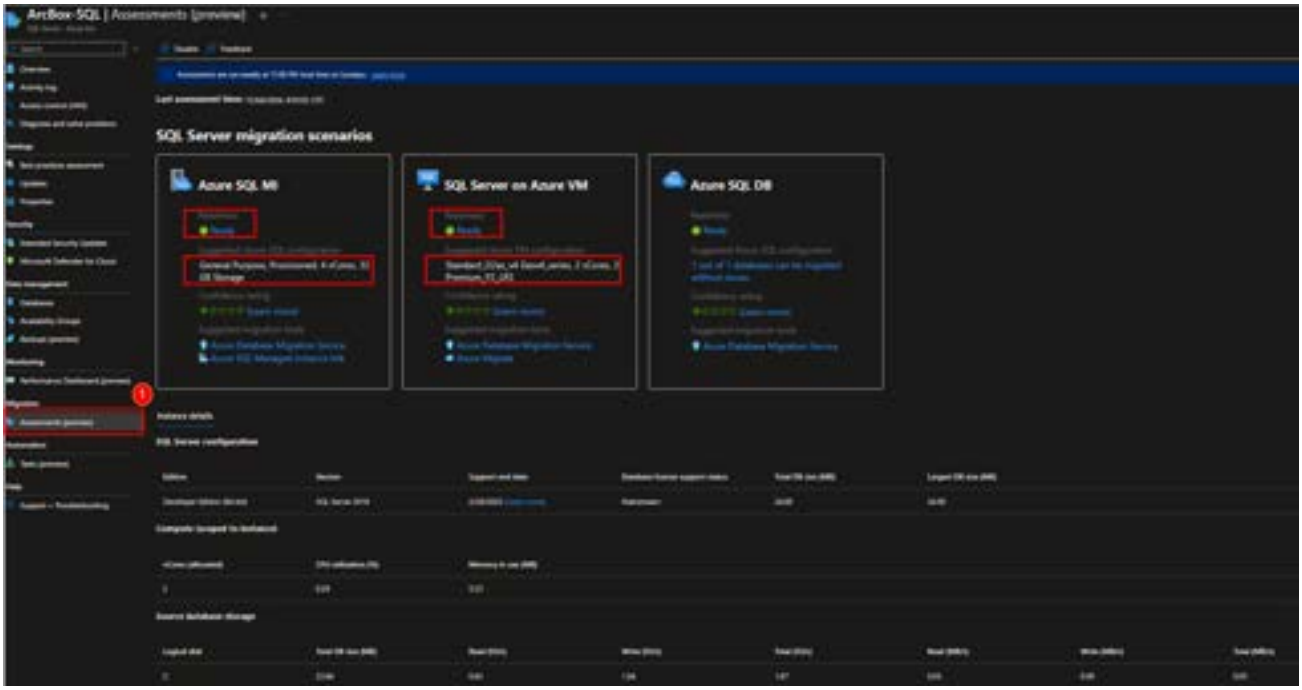
Estimated Time to Complete This Exercise

30 minutes

Task 1: Review the migration readiness to Azure SQL Managed Instance (MI) and SQL Server on Azure VM

- [ ] 1. From the Azure portal go to *Azure Arc* | *All Azure Arc resources* page and select your SQL Server.

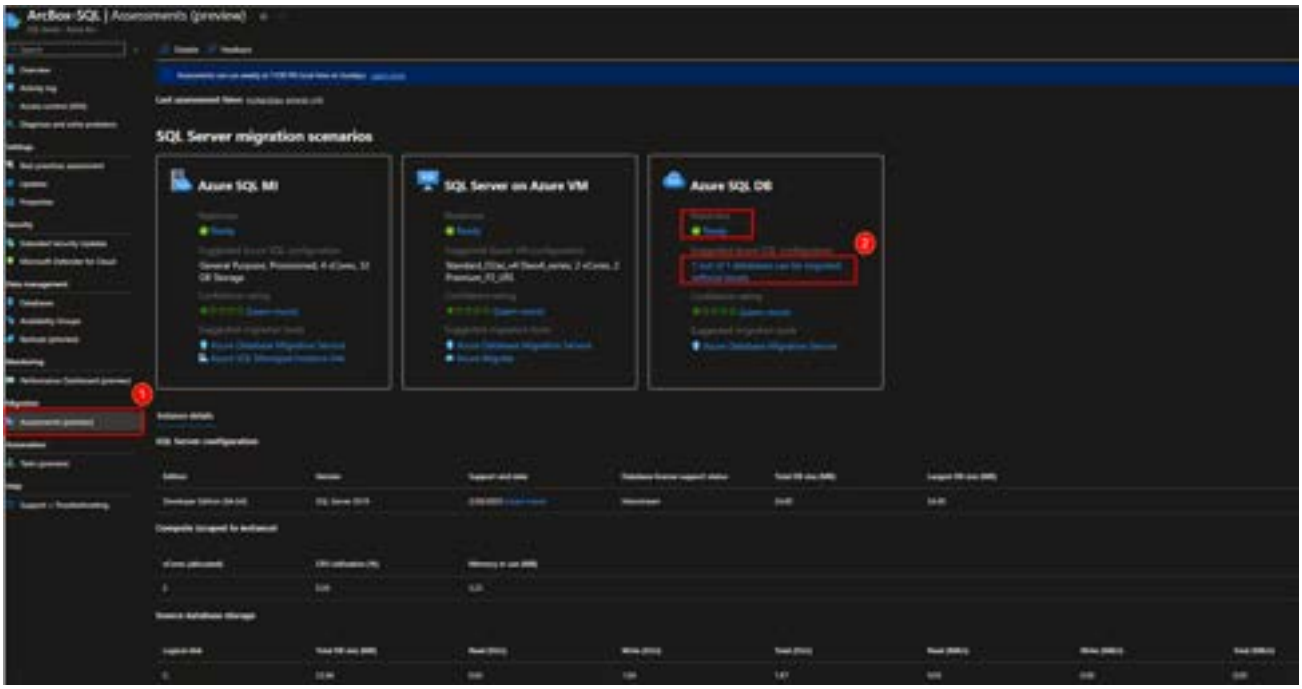
- [ ] 2. From the SQL server resource page select *Assessments* (preview) under the *Migration* folder in the left pane. Review the *Readiness* status and the recommended configurations to host your Arc-enabled SQL Server on an Azure SQL MI or on Azure VM. The readiness is based on the performance evaluation for the Arc-enabled SQL Server instances.



- [ ] 3. Review the following link to understand the assessment confidence rating.

Task 2: Review the migration readiness to Azure SQL DB

- [ ] 1. On the Assessment page, review the *Readiness* status and the recommended configurations to host specific databases from you Arc-enabled SQL Server to Azure SQL database. Click on the link showing the number of databases that can be migrated without issues.



☐ 2. Note the database name and recommended Azure SQL DB compute and storage.

☐ 3. Click the *Ready* link under *Migration readiness* to see migration detail such as issues and warnings.

Exercise 1 has been completed

Click **Next** for the next lab or **Go back to the main table of content**