Instructions for Installing Intunebackup

If you are unfamiliar with Azure Runbooks, here is a guide to get you started:

https://andrewstaylor.com/2024/03/04/getting-started-with-graph-and-azure-automation/

If you are updating from a previous version, copy the files to your server, edit as required and run https://yourdomain/update.php

Step 1

- Download the latest version of the Intune Backup and Restore PowerShell script from here:

https://github.com/andrew-s-taylor/public/blob/main/Powershell%20Scripts/Intune/intune-backup-restore-withgui.ps1

- Create a secure password, this will be used when the web app communicates with the script as an extra layer of security (it won't run without it)
- Find this code and add your secret

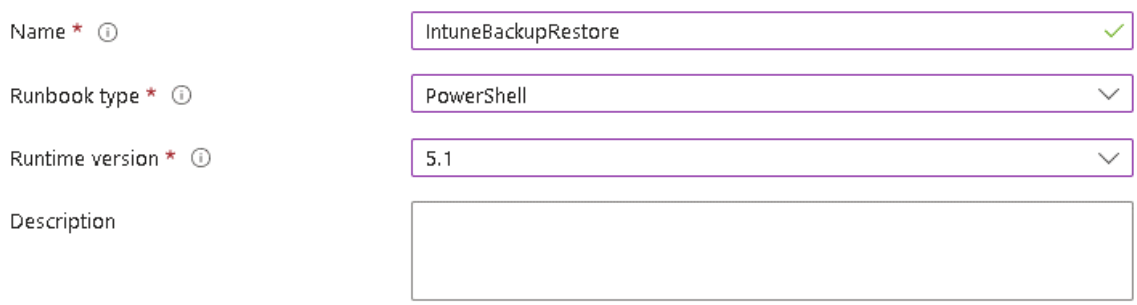
```
##Set my password

$webhooksecret = ""
```

Step 2

- Navigate to Azure Automation Accounts and either create a new account, or use a current one for this runbook. Make sure it has Public Access networking so the web app can communicate (unless it is also running in Azure)

Step 3

- Create a new Runbook in your Automation Account using PowerShell 5.1 for backup and restore



Step 4

Paste the PowerShell script into the window, Save and Publish (you can't test because the variables are sent via a webhook)

Step 5

- Click on webhooks and create a new webhook.
- Set an expiry and make a note of the URL as you will need that later and it can't be accessed after creation.
- The run settings and parameters can all be left as default values.

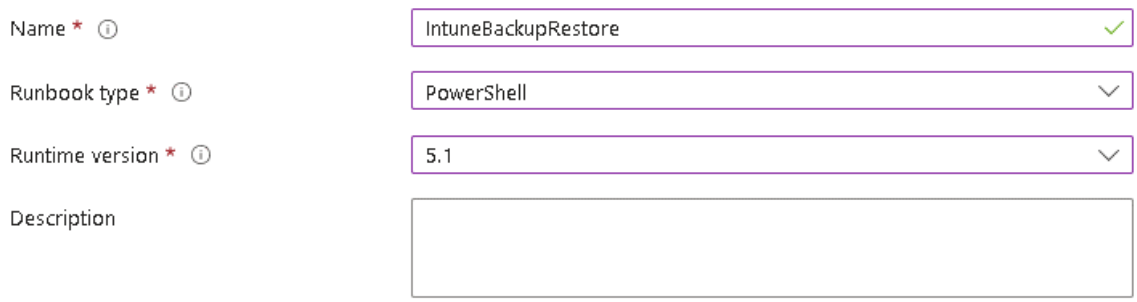
Step 6

- Download the latest version of the Win32 app script from here:

https://github.com/andrew-s-taylor/public/blob/main/Powershell%20Scripts/Intune/deploy-winget-win32-multiple.ps1

Step 7

- Create a new Runbook in your Automation Account using PowerShell 5.1 for app deployment



Step 8

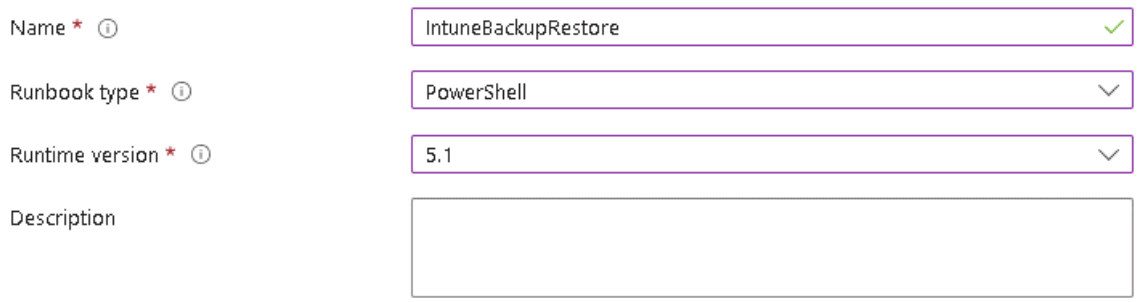
Paste the PowerShell script into the window, Save and Publish (you can't test because the variables are sent via a webhook)

Step 9

- Click on webhooks and create a new webhook.
- Set an expiry and make a note of the URL as you will need that later and it can't be accessed after creation.
- The run settings and parameters can all be left as default values.

Step 10

- Create a new Runbook in your Automation Account using PowerShell 5.1 drift monitoring



Step 11

Paste the PowerShell script into the window, Save and Publish (you can't test because the variables are sent via a webhook)

Step 12

- Click on webhooks and create a new webhook.
- Set an expiry and make a note of the URL as you will need that later and it can't be accessed after creation.
- The run settings and parameters can all be left as default values.

Step 13

- Upload the monitor-drift.ps1 file into the runbook. No amendments are required

Step 14

- If using templated deployment, edit processdemo.php and add your Git details to the JSON containing the policies.

An example baseline json is included in the archive file.

Step 15

- Extract the web app files and upload them to your web server.

Step 16

- Create a MySQL database to store the web app details as well as an account with full access to it.

Step 17

- Navigate to:
  http(s)://yourwebaddress/install
- Run through the installation steps, entering your details as required. If you make any mistakes, edit the config.php afterwards to set them correctly (it will warn you of anything obvious)

Step 18

- Remove the Install directory, you are all setup.
- Remember to populate your profile with tenant details, git repos etc.

Step 19

- Add a daily cron job on your webserver pointing to https://yourdomain/cron.php
  No additional inputs are required, this loops through all configured tenants and runs a drift check