



2018

Azure Automation in Production

Lessons learned in the field



Jakob Gottlieb Svendsen
Principal Consultant / Lead Developer @ CTGlobal
Microsoft Cloud & Data Center MVP
@JakobGSvendsen
jgs@ctglobalservices.com

Agenda

- Authoring
- Structure
- Administration / Source Control
- Logging
- Alerting
- Reporting



Authoring

- PowerShell ISE
 - Install-Module AzureAutomationAuthoringToolkit
 - On Hybrid Worker Remember –**Scope CurrentUser**
- Visual Studio Code
 - Shared resources can be used
 - ISE for administration
 - VSCode for coding.
 - Community extension: **AzureAutomation**



Authoring

Structure - Runbook

- Comment Based Help
- Parameters
 - Mandatory / Optional
 - Input Type
- Preferences
 - ErrorActionPreference = Stop
 - VerbosePreference?
- Catch All Errors
 - Might need more than one try/catch (inside foreach)
- Control Return / Output
 - Use \$null = (Thanks Tobias!)

Use a template!!

Structure - Modular Runbooks

- **Initiation**

- Used by: Jobs/Triggering
- Output: Human readable or JSON

- **Control**

- Used by: Other runbooks
- Output: Standardized object (CTReturnObject)
- Complete a complete or part of a flow.
 - i.e. New-CTEmployee
 - Or New-CTEmployeeMailbox

- **Component**

- Generalized Runbook
- For one specific job, i.e. New-CTADGroup
- Returns standard objects such as ADGroup
- Should not rely on outside resources such as AA Shared Resources

Structure

Structure – Lesson Learned

- Max Output Size per Object = ~1MB



Administration – Source Control

- Use a script repository!
- Git is great! (p.s. Git is not only GitHub and it does not have to be hard or advanced).
- We use Visual Studio Team Services but any source control will do
- Setup events to deliver new runbooks to account(s)
 - Deliver to multiple customers etc.!
- Or setup a release pipeline



Continuous Delivery

Administration – Sync VSTS

- VSTS – Git – Service Hook

- Use script provided by MSFT

<https://github.com/azureautomation/runbooks/blob/master/Utility/ARM/Sync-VSTS.ps1>

- Or Pull Request/Enhanced CT Global Version at

<http://blog.ctglobalservices.com/powershell/jgs/enhanced-sync-vstsgit-runbook-that-support-pull-requests/>

Official Setup Guide

<https://docs.microsoft.com/en-us/azure/automation/automation-scenario-source-control-integration-with-vsts>



Logging - Code

- Log before actions! (Add-Tracelog / Write-Verbose)
- Optional: Log after actions!
- Use TraceLog
 - Collect Log entries in \$TraceLog
 - Write-Verbose \$Tracelog in the end
 - Send \$TraceLog with runbook output

Logging – Log Analytics

- Forward logs to Azure Log Analytics
- Retention up to 2 years at low cost
- Advanced query language
- Categories from Automation
 - **JobLogs** = Job Result/Status
 - **JobStreams** = Job outputs
 - **DscNodeStatus** = DSC Node Status
- Configure log forwarding
 - Categories (de/select data types)



Logging – Log Analytics Fields

- RunbookName_s
- Category: Job
 - ResultType = Status
- Category: JobStream
 - ResultDescription = Output
 - ResultDescription_* = Parsed JSON outputs

4/13/2018 11:59:53.967 AM | AzureDiagnostics

```
... TimeGenerated      : 4/13/2018 11:59:53.967 AM
... StreamType_s      : Output
... OperationName     : Job
... ResultType        : In Progress
... RunbookName_s     : Update-GraphSubscriptions
... Caller_s          : System
... Category          : JobStreams
... Resource           : AUTOMATION
... Tenant_g          : 32feae9a-1648-4ce2-8410-c407540d5983
... JobId_g           : 6705aeae-2a3c-4694-b25d-5b3dcc64a2c6
... SourceSystem       : Azure
... ResourceId        : /SUBSCRIPTIONS/7FA4F2EA-77CD-4629-8FB0-42988A97B934
... ResultDescription  : Disabled items: 1
... SubscriptionId    : 7fa4f2ea-77cd-4629-8fb0-42988a97b934
... ResourceGroup     : OMS-RG
... ResourceProvider   : MICROSOFT.AUTOMATION
... ResourceType      : AUTOMATIONACCOUNTS
... CorrelationId     : b5ff8149-aa6c-4e9f-a21f-0b1e0bff38f1
```

Log Analytics



Alerting

- Azure Monitor
- New Alerts support Automation & Log Analytics
 - No more OMS alerts
 - No requirement for linked account to OMS!



Alerting – Azure Monitor

- 2 ways
- Metrics (preview)
 - Triggers in about 5 minutes (my test result)
 - Trigger on job result
 - One Rule, one Runbook.
 - Currently only works on runbooks executed in Azure
- Log Analytics Query
 - Triggers in about 15 minutes (my test result)
 - FAF (Flexible as F*** / Flexibel wie F***)
 - One rule for multiple runbooks / any result.



Azure Monitor Alerts

Alerting – Azure Monitor Metrics

- Metrics monitor Rule
 - Target: Automation Account
 - Type: Platform
 - Metric: Total Jobs
 - Select Runbook + Status condition
- To setup alert, there has to be at least one result for the runbook within 6 hour time period!
- Make a “New-Error” runbook to trigger a failed job to select “Failed” status.
- Example: In session content on GitHub & <https://docs.microsoft.com/en-us/azure/automation/automation-create-alert-triggered-runbook>



Alerting – Azure Monitor Log Analytics

- Log Analytics Rule
 - Query Example:

**AzureDiagnostics | where ResourceProvider ==
"MICROSOFT.AUTOMATION" and ResultType ==
"Failed"**

- Runbook Example:

<https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-alerts-actions#runbook-actions>



Reporting

- PowerShell
 - Just because Automation rocks!
 - I mean.. PowerShell FTW !!!
 - But... Sometimes... GUI... can... be... nice (hides in the corner)
- PowerBI
 - Great Visual Tool
 - Advanced manipulation
 - Use JSON to output and show objects! (PowerBI can convert json back to objects!)



Reporting - PowerBI

- Author and Test Query in Log Analytics
- Get PowerBI query code
 - Export -> PowerBI
- JSON Outputs

Reporting – Log Analytics Query

- Query Example:

```
AzureDiagnostics |  
where ResourceProvider == "MICROSOFT.AUTOMATION" |  
where TimeGenerated > ago(730d)
```

PowerBI

Reporting – Lessons Learned

- JSON output is automatically expanded
- Set time range in Query! (ago())



Administration – Export/Import

- Preview / Prototype!
- Export Runbooks
- Auto Export referenced runbook
- Auto export used assets
- Join the project:

<https://github.com/JakobGSvendsen/AzureAutomationImportExport>

Summary

- Authoring – VS Code Rules!
- Structure – Use Templates!
- Administration – Use Source Control / Webhooks
- Logging – Log Analytics rules!
- Alerting – Azure Monitor – Metrics or Log Analytics
- Reporting – PowerBI!



Next Steps

- Now: 15 min break
- Grab a coffee
- Stay here to enjoy next presentation
- Change track and switch to another room
- Ask me questions or meet me in a breakout session room afterwards



Questions?



about_Author

- Jakob Gottlieb Svendsen
- Principal Consultant / Lead Developer @ CTGlobal
- Microsoft Cloud & Data Center MVP
- Twitter: @JakobGSvendsen
- Email: jgs@ctglobalservices.com
- GitHub: <https://github.com/JakobGSvendsen>
- LinkedIn: <https://www.linkedin.com/in/jakobgsvendsen/>



Simple Console Demos

Embed one-liners right into your slides:

```
PS> Get-ExecutionPolicy  
Bypass
```



Demo

Try and use demos instead of dead slides.

Please do not dive into zillions of pages of code.
Instead, if you can, turn your logical code blocks into functions, and store them in a module.

Share this module with your audience.


```
# use this template for code samples
# and follow these instructions to show perfectly
# color-coded PowerShell code:
```

```
# paste code to PowerShell ISE, select it, and copy it
# to the clipboard
```

```
Paste-Code -ToISE | Select-Code | Copy-ToClipboard
```

```
# to insert it with full color-coding into a slide,
# paste the code to your PPT slide. It will be black,
# and there is a toolbutton labelled (Ctrl) in PPT.
Click-ToolButton -Choose SymbolWithClipboardAndBrush
# Click the toolbutton, and choose the button that shows
# a clipboard with a brush. This will add color-coding
# back to the pasted code
```

PLEASE NOTE: You do not need to use PowerShell ISE to code, or to demo. Use whatever editor you like best.

These steps use PowerShell ISE to color-code your code correctly and consistently, and insert the color-coded code into the slide. Please help make all code look consistent. Many thanks!

