

On-Premise Azure DevOps Self-Hosted Agents

Afvikling af vores scripts uden lokal Task Scheduler og med koden i Azure Repos



You

Explain briefly, how we on our on-premise Azure DevOps can set a Self-Hosted Agent running a CopyFolder.ps1 cron job each day at midnight. Please explain in 10 lines. Also what happens behind the scenes.



ChatGPT

1. **Install Self-Hosted Agent:** On your on-premise server, download and install the Azure DevOps self-hosted agent from your Azure DevOps organization's Agent Pools settings.
2. **Configure Agent:** Follow the installation prompts to configure the agent, ensuring it's connected to your Azure DevOps organization and assigned to a specific agent pool.
3. **Create PowerShell Script:** Place your `CopyFolder.ps1` script in a known directory on the server where the agent is installed.
4. **Commit Script to Azure Repos:** If not already, commit the script to your Azure Repos within your project to maintain version control.
5. **Create YAML Pipeline:** In Azure DevOps, create a new pipeline (`copy-folder-pipeline.yml`) using the YAML syntax to define the cron schedule (`cron: "0 0 * * *"` for midnight execution), specify the self-hosted agent pool, and include a step to execute `CopyFolder.ps1`.
6. **Schedule with Cron:** In the YAML pipeline, use the `schedules` section to set the cron job for midnight execution using `cron: "0 0 * * *"`.
7. **Pipeline Execution:** Commit and push the YAML file to Azure Repos. Set up the pipeline in Azure DevOps to point to this YAML file.
8. **Automated Trigger:** The pipeline is now set to automatically trigger at midnight every day, where Azure DevOps sends the job to the self-hosted agent.
9. **Behind the Scenes:** At midnight, the self-hosted agent fetches the latest version of `CopyFolder.ps1` from Azure Repos, executes it on the local server, and reports the outcome back to Azure DevOps.
10. **Monitoring and Logs:** You can monitor execution results, view logs, and manage pipeline configurations directly from Azure DevOps.

This setup leverages Azure DevOps for CI/CD with a self-hosted agent on your premises, automating script execution while maintaining control and visibility through Azure DevOps.



CI (testing)

```
Präsentation > ! ci.yml
1  trigger:
2  | branches:
3  |   include:
4  |     - main
5  | paths:
6  |   include:
7  |     - PowerShellScripts/scripts/*
8
9  pool:
10 |   name: 'YourAgentPoolName'
11
12 steps:
13 - checkout: self
14
15 - script: |
16   | Install-Module -Name PSScriptAnalyzer -Force -Scope CurrentUser
17   | displayName: 'Install Analysis Tools'
18
19 - script: |
20   | # Run analysis on all PowerShell scripts
21   | Get-ChildItem -Path $(Build.SourcesDirectory)/ `
22   | PowerShellScripts/scripts/*.ps1 -Recurse |
23   | ForEach-Object { Invoke-ScriptAnalyzer -Path $_.FullName }
24   | *displayName: 'Analyze PowerShell Scripts'
25   | ~~~
```

CD (cron jobs)

```
1  schedules:
2  - cron: "0 * * * *"
3    displayName: 'Hourly Copy Folder'
4    branches:
5    | include:
6    |   - main
7    | always: true
8
9  trigger: none
10
11 pool:
12 |   name: 'YourAgentPoolName'
13
14 steps:
15 - checkout: self
16 - script: |
17   | powershell -File $(Build.SourcesDirectory)/ `
18   | PowerShellScripts/scripts/CopyAFolder.ps1
19   | *displayName: 'Copy A Folder'
20
```

Repo Struktur

Præsentation > ! struture.yml

```
1  /PowerShellScripts
2  ci.yml
3    /scripts
4      CopyAFolder.ps1
5      DeleteOldFile.ps1
6      DuplicateFile.ps1
7    /pipelines
8      copy-a-folder-pipeline.yml
9      delete-old-file-pipeline.yml
10     duplicate-file-pipeline.yml
```