

Dokumentation Projekt Ansible-Chocolatey-PowerShell

1 Teilnehmer/innen des Teams:

Klasse: WUP22A	Team: Luca Binder & Carl Otto Strömstedt
-------------------	---

2 Anforderungsdefinition (Meilenstein A)

„Projektname“	
Fachlicher Inhalt: (Allgemeine Beschreibung)	<p>Mit Ansible von Linux aus Automatisierung auf einer Windows 11 VM ausführen.</p> <p>In der Informatik ist das Aufsetzen von neuer Hardware eine Zeitaufwändige Prozess. Wir wollen mit diesem Projekt aufzeigen wie mit Ansible & Chocolatey sehr einfach einem Skript eine ganze Inventory an Maschinen mit Software aufgesetzt werden kann.</p> <p>Luca wird anstatt mit Chocolatey Software aufsetzen, aufzeigen wie man mit Ansible direkt ein Powershell-Skript ausführen kann.</p>
MUSS Kriterien: (Konkrete Features, die umzusetzen sind)	<p>Ablauf des Playbooks (Carl):</p> <ul style="list-style-type: none"> • Host inventory auslesen • Auf den einzelnen hosts choco install ausführen <p>Ablauf des Playbooks (Luca):</p> <ul style="list-style-type: none"> • Host Inventory auslesen • Auf den einzelnen hosts ein Powershell-Skript ausführen

Dokumentation Projekt Ansible-Chocolatey-PowerShell

KANN Kriterien: (Konkrete Features, die optional sind)	Zusatz: <ul style="list-style-type: none"> PowerShell-Skript im Playbook, welches das zweite PowerShell-Skript schedulet.
--	---

2.1 Planung (LB1 / LB2)

MS	Tätigkeit / Abgabe	Soll-Datum	Ist-Datum
A	Projektstart ➤ Team Bildung ➤ Wahl / Ausarbeitung der Anforderungsdefinition (Kap. 2) Abnahme Anforderungsdefinition durch Lehrperson	13.04.2023	13.04.2023
B2	Teamaufgabe 2: ➤ Abgabe: Testvorschrift und Testfälle	18.04.2023	18.04.2023
C2	Einzelaufgabe 4: ➤ Abgabe: Ausgefüllter Systemtest	18.04.2023	18.04.2023

Dokumentation Projekt Ansible-Chocolatey-PowerShell

3 Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

3.1 Ziel des Tests

- Die Maschine im Inventar ist erreichbar
- Carl:
- Die Software wurde auf der Maschine installiert
- Luca:
- PowerShell-Skript ausgeführt

3.2 Art des Tests

Blackbox-Test

3.3 Verwendete Hilfsmittel

Testumgebung mit einer Linux-Maschine und einer Windows 11 VM

3.4 Anforderung an das Testobjekt

- Die Linux-Maschine muss ansible installiert haben
- Die Windows 11 VM muss OpenSSH aktiviert haben und Ansible, Python und Chocolatey installiert haben

3.5 Testvorgaben

Die Erreichbarkeit der Maschine wird mit win_ping getestet:

```
ansible win -m win_ping -i hosts
```

Carl:

Die Software, die installiert ist, wird mit choco list getestet:

```
choco list --localonly
```

Luca:

PowerShell-Skript wurde erfolgreich ausgeführt

Dokumentation Projekt Ansible-Chocolatey-PowerShell

3.6 Abbruchkriterien


- Host ist nicht erreichbar (ping fail)
- Software wurde nicht korrekt installiert (vorgegebene Software im Skript stimmt nicht mit der installierten Software überein)
- PowerShell-Skript failt

4 Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Testprotokoll - Testvalidierung

Projektname	<i>Ansible-Chocolatey-PowerShell</i>
Version (getestetes Programm)	1.0
Projekt-Code (Dateien)	Projektdateien
Fachlicher Ansprechpartner (Namen der Lehrperson)	Thomas Kälin
Autor des Testprotokolls	Carl Otto Strömstedt
Testdatum	17.03.2023
Name Tester	Carl Otto Strömstedt

Testgruppe <<A>>

Beschreibung Testgruppe: Ansible - Chocolatey				
Setup: VM starten				
#	Nötige Eingabe/Aktion	Erwartete Ausgabe/Aktion	Tatsächliche Ausgabe/Aktion	
A1	Ansible win_ping	Ping – Pong	Ping - Pong	X
A2	Choco list	Gleiche Software	Gleiche Software	

Dokumentation Projekt Ansible-Chocolatey-PowerShell

Testgruppe <>

Projektname	<i>Ansible-Chocolatey-PowerShell</i>
Version (<i>getestetes Programm</i>)	<i>1.0</i>
Projekt-Code (<i>Dateien</i>)	<i>Projektdateien</i>
Fachlicher Ansprechpartner (<i>Namen der Lehrperson</i>)	<i>Thomas Kälin</i>
Autor des Testprotokolls	<i>Luca Binder</i>
Testdatum	17.03.2023
Name Tester	Luca Binder

4.0 Testcases Ansible – PowerShell

Preconditions for all testcases:

- VM is running.

4.1 Testcase 1: Ansible win_ping

Description: Test the Ansible win_ping module for successful communication between the control node and the Windows host.

Preconditions

1. User has a working Ansible setup with a Windows host.
2. The control node has access to the Windows host.

Test Steps

1. Run the Ansible win_ping module on the Windows host.
2. Check the output.

Expected Results

1. The Ansible win_ping module runs without any errors.
2. The output shows a successful 'Ping - Pong' communication between the control node and the Windows host.

Actual Results

1. The Ansible win_ping module runs without any errors.

Dokumentation Projekt Ansible-Chocolatey-PowerShell

2. The output shows a successful 'Ping - Pong' communication between the control node and the Windows host.

4.2 Testcase 2: Set-WindowAnimations

Description: Test if the Set-WindowAnimations function enables or disables the window animations correctly.

Test Steps

1. Run the PowerShell script.
2. Run the Set-WindowAnimations function with the \$Enable parameter set to \$true.
3. Check the 'MinAnimate' registry key value.
4. Run the Set-WindowAnimations function with the \$Enable parameter set to \$false.
5. Check the 'MinAnimate' registry key value.

Expected Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The 'MinAnimate' registry key value is set to '1', indicating that window animations are enabled.
4. The function call completes successfully.
5. The 'MinAnimate' registry key value is set to '0', indicating that window animations are disabled.

Actual Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The 'MinAnimate' registry key value is set to '1', indicating that window animations are enabled.
4. The function call completes successfully.
5. The 'MinAnimate' registry key value is set to '0', indicating that window animations are disabled.

4.3 Testcase 3: Create-MyDocumentsFolders

Description: Test if the Create-MyDocumentsFolders function creates the specified folders and subfolders, as well as the test documents inside each subfolder.

Dokumentation Projekt Ansible-Chocolatey-PowerShell**Test Steps**

1. Run the PowerShell script.
2. Run the Create-MyDocumentsFolders function.
3. Check if the specified folders, subfolders, and test documents are created in the user's 'My Documents' folder.

Expected Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The specified folders ('TBZ', 'ÜK', and 'Sunrise'), subfolders ('m242', 'm122', 'm226b', 'm153', 'Lohnausweis', 'Sunrise_Projekte', 'UEK106', 'UEK304'), and test documents (e.g., 'Micropython_Zusammenfassung.txt', 'PowerShell_Ideen.txt', 'Java_Projekt.txt', 'Datenbank.txt', 'Lohnabrechnung_2022.txt', 'Projektplanung_Vacation_Planner.txt', 'ÜK_Präsentation.txt', 'ÜK_Lehrmaterialien.txt') are created correctly in the user's 'My Documents' folder.

Actual Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The specified folders ('TBZ', 'ÜK', and 'Sunrise'), subfolders ('m242', 'm122', 'm226b', 'm153', 'Lohnausweis', 'Sunrise_Projekte', 'UEK106', 'UEK304'), and test documents (e.g., 'Micropython_Zusammenfassung.txt', 'PowerShell_Ideen.txt', 'Java_Projekt.txt', 'Datenbank.txt', 'Lohnabrechnung_2022.txt', 'Projektplanung_Vacation_Planner.txt', 'ÜK_Präsentation.txt', 'ÜK_Lehrmaterialien.txt') are created correctly in the user's 'My Documents' folder.

Dokumentation Projekt Ansible-Chocolatey-PowerShell**4.4 Testcase 4: Create-DesktopShortcuts**

Description: Test if the Create-DesktopShortcuts function creates shortcuts to the specified folders on the user's desktop.

Test Steps

1. Run the PowerShell script.
2. Run the Create-DesktopShortcuts function.
3. Check if the specified shortcuts are created on the user's desktop.

Expected Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The specified shortcuts ('TBZ.Ink', 'ÜK.Ink', and 'Sunrise.Ink') are created correctly on the user's desktop, each pointing to the corresponding folder in the user's 'My Documents' folder.

Actual Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The specified shortcuts ('TBZ.Ink', 'ÜK.Ink', and 'Sunrise.Ink') are created correctly on the user's desktop, each pointing to the corresponding folder in the user's 'My Documents' folder.

4.5 Testcase 5: Folder Structure and Test Documents

Description: Test if the folder structure and test documents are created correctly and placed in the correct subfolders.

Test Steps

1. Run the PowerShell script.
2. Run the Create-MyDocumentsFolders function.
3. Verify the folder structure and test documents in the user's 'My Documents' folder.

Expected Results

1. The script runs without any errors.
2. The function call completes successfully.
3. The folder structure is as follows:

Dokumentation Projekt Ansible-Chocolatey-PowerShell

- 'My Documents'
 - TBZ
 - m242
 - Micropython_Zusammenfassung.txt
 - m122
 - Powershell_Ideen.txt
 - m226b
 - Java_Projekt.txt
 - m153
 - Datenbank.txt
 - ÜK
 - UEK106
 - ÜK_Präsentation.txt
 - UEK304
 - ÜK_Lehrmaterialien.txt
 - Sunrise
 - Lohnausweis
 - Lohnabrechnung_2022.txt
 - Sunrise_Projekte
 - Projektplanung_Vacation_Planner.txt

Dokumentation Projekt Ansible-Chocolatey-PowerShell**Actual Results**

1. The script runs without any errors.
2. The function call completes successfully.
3. The folder structure is as follows:
 - 'My Documents'
 - TBZ
 - m242
 - Micropython_Zusammenfassung.txt
 - m122
 - Powershell_Ideen.txt
 - m226b
 - Java_Projekt.txt
 - m153
 - Datenbank.txt
 - ÜK
 - UEK106
 - ÜK_Präsentation.txt
 - UEK304
 - ÜK_Lehrmaterialien.txt
 - Sunrise
 - Lohnausweis
 - Lohnabrechnung_2022.txt
 - Sunrise_Projekte
 - Projektplanung_Vacation_Planner.txt

4.6 Testcase 6: End-to-End Execution

Description: Test the end-to-end execution of the PowerShell script.

Test Steps

1. Run the PowerShell script.
2. Check if the specified folders, subfolders, and test documents are created in the user's 'My Documents' folder.
3. Check if the specified shortcuts are created on the user's desktop.
4. Check the 'MinAnimate' registry key value.

Dokumentation Projekt Ansible-Chocolatey-PowerShell**Expected Results**

1. The script runs without any errors.
2. The specified folders ('TBZ', 'ÜK', and 'Sunrise'), subfolders ('m242', 'm122', 'm226b', 'm153', 'Lohnausweis', 'Sunrise_Projekte', 'UEK106', 'UEK304'), and test documents (e.g., 'Micropython_Zusammenfassung.txt', 'PowerShell_Ideen.txt', 'Java_Projekt.txt', 'Datenbank.txt', 'Lohnabrechnung_2022.txt', 'Projektplanung_Vacation_Planner.txt', 'ÜK_Präsentation.txt', 'ÜK_Lehrmaterialien.txt') are created correctly in the user's 'My Documents' folder.
3. The specified shortcuts ('TBZ.lnk', 'ÜK.lnk', and 'Sunrise.lnk') are created correctly on the user's desktop, each pointing to the corresponding folder in the user's 'My Documents' folder.
4. The 'MinAnimate' registry key value is set to '1', indicating that window animations are enabled.

Actual Results

1. The script runs without any errors.
2. The specified folders ('TBZ', 'ÜK', and 'Sunrise'), subfolders ('m242', 'm122', 'm226b', 'm153', 'Lohnausweis', 'Sunrise_Projekte', 'UEK106', 'UEK304'), and test documents (e.g., 'Micropython_Zusammenfassung.txt', 'PowerShell_Ideen.txt', 'Java_Projekt.txt', 'Datenbank.txt', 'Lohnabrechnung_2022.txt', 'Projektplanung_Vacation_Planner.txt', 'ÜK_Präsentation.txt', 'ÜK_Lehrmaterialien.txt') are created correctly in the user's 'My Documents' folder.
3. The specified shortcuts ('TBZ.lnk', 'ÜK.lnk', and 'Sunrise.lnk') are created correctly on the user's desktop, each pointing to the corresponding folder in the user's 'My Documents' folder.
4. The 'MinAnimate' registry key value is set to '1', indicating that window animations are enabled.

Dokumentation Projekt Ansible-Chocolatey-PowerShell**Tabellenübersicht - Testcases**

Testcase ID	Testcase Name	Test Result
1	Ansible win_ping	Success
2	Set-WindowAnimations	Success
3	Create-MyDocumentsFolders	Success
4	Create-DesktopShortcuts	Success
5	Folder Structure and Test Docs	Success
6	End-to-End Execution	Success

Dokumentation Projekt Ansible-Chocolatey-PowerShell**Review des Testbeschriebs durch den Tester:***Guter Testbeschrieb und die Tests waren erfolgreich.***5 Sign-Off****Mängelliste:**

- Keine Mängel gefunden

Der Test

(x) wird **erfolgreich** abgenommen.

() wird eingeschränkt abgenommen (Mängel siehe oben).
Der Test wird **trotzdem als erfolgreich** abgenommen erklärt.

() wird **nicht** abgenommen (aufgetretene Mängel siehe oben)

Test ist beendet und wurde korrekt durchgeführt

Test A: **Ja (x)** **Nein ()** Unterschrift (Datum, Carl Strömstedt)

Test B: **Ja (x)** **Nein ()** Unterschrift (Datum, Luca Binder)

Validierung

Ja () **Nein ()**

Unterschrift (Datum, Name *Experte*)