Chemnitzer Linux Tage 2025

Ansible Automatisierung in hybriden Umgebungen

Red Hat
Ansible Automation
Platform

ASPICON
Your business.
Always on!

Wer wir sind und was wir machen

- ASPICON ist Servicepartner für verschiedene Unternehmen (SMB, DAX) und (KRITIS) Organisationen
- Entwicklung und Umsetzung von Konzepten für sichere, hochverfügbare und performante Datenbank-Infrastrukturen
- Über 600 Kunden
- 130 Managed Service Kunden mit über 400 Servicevereinbarungen
- Überwachung von ca. 258 produktiven Datenbanken
- 3.500 Consulting Projekte in den letzten 7 Jahren

ASPICON
Your business. Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Das erwartet euch heute

Generated by Adobe Firefly

Agenda

- Vorbereitung einer Workstation
  - Python virtual Environment
  - Ansible Konfiguration
- Ansible User auf den Hosts
- Demo Environment & Inventory
- Definition der Connection-Variablen
  - `demo-01` | Windows Server 2012 -> WinRM
  - `demo-02` | Windows Server 2019 -> WinRM & PSRP & Certificate Validation & Certificate Authentication
  - `demo-03` | Windows Server 2022 -> SSH & PubKey
- Demo Playbook (`pb_whoami.yml`)
- Benchmark (WinRM vs. PSRP vs. SSH vs. SSH)
- Fragen

ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Hybride Umgebungen

Generated by Adobe Firefly

Hände hoch, …

- … wer betreibt hybride Umgebungen (gezwungen oder freiwillig)?
  - Linux
    - Red Hat Enterprise Linux
    - Oracle Linux
    - SUSE Linux Enterprise Server
    - Ubuntu Server
    - Debian
    - …
  - Windows
    - Windows Server 2012 R2
    - …
    - Windows Server 2025
  - FreeBSD
  - AIX
  - …

Generated by Adobe Firefly

ASPICON
Your business. Always on!

Hände hoch, …

- … wer hat Erfahrung mit Ansible, AWX, AAP …?

Generated by Adobe Firefly

ASPICON
Your business.
Always on!

Hybride Umgebungen und wie man sie los wird

Schlecht

bis

gar nicht!

Generated by Adobe Firefly

Aber …

müssen wir auch überhaupt nicht ;-)

ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Vorbereitung der Workstation

Generated by Adobe Firefly

Vorbereitung der Workstation | Python Virtual Environment

- Python Virtual Environment
  - Environment aktivieren
  - Update pip & setuptools
  - Ansible installieren
  - Zusätzliche Module installieren

```
$> cd ~/
$> python3 -m venv ENVIRONMENT NAME
$> . ~/ENVIRONMENT_NAME/bin/activate

$> pip install --upgrade pip setuptools
Requirement already satisfied: pip in
/home/people/mde/NOGIT/VENV/3.11/lib/python3.11/site-packages (25.0.1)
Requirement already satisfied: setuptools in
/home/people/mde/NOGIT/VENV/3.11/lib/python3.11/site-packages (75.8.2)

$> pip index versions ansible-core
WARNING: pip index is currently an experimental command. It may be
removed/changed in a future release without prior warning.
ansible-core (2.18.3)
Available versions: 2.18.3, 2.18.2, 2.18.1, 2.18.0, 2.17.9, 2.17.8,
2.17.7, 2.17.6, 2.17.5, 2.17.4, 2.17.3

$> pip install ansible==2.17.9

$> pip install pypsrp pywinrm
```


ASPICON
Your business.
Always on!

Vorbereitung der Workstation | Ansible Config

- Ansible Vorbereiten
  - Ansible Config
    - ./ansible.cfg
    - ANSIBLE_CONFIG
    - ~/.ansible.cfg
    - /etc/ansible/ansible.cfg
  - Inventory Anlegen
  - SSH-Config
    - Optional, jedoch empfohlen
  - Host …
  - Fertig -> Viel Spaß 😃

```
[defaults]
inventory = /path/to/inv_clt_2025/
host_key_checking = false
timeout = 30
callbacks_enabled = timer, profile_tasks, profile_roles

[privilege_escalation]
become = true
become_method = sudo
become_user = root
become_ask_pass = false

[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s -o PreferredAuthentications=publickey
pipelining = true

[diff]
always = true

[galaxy]
server_list = galaxy

[galaxy_server.galaxy]
url=https://galaxy.ansible.com/
```


ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Konfiguration der Hosts

Generated by Adobe Firefly

Ansible & Linux | SSH

- Vorbereitung des Hosts
  - Ansible remote user
  - Sudo
  - SSH PubKey

- Ergebnis
  - Sicheres Login
  - Unprivilegierter User
  - Eskalation möglich, nach Angabe weiterer Credentials
  - Fertig -> Viel Spaß 😃

By Xah Lee. Date: 2020-01-13

ASPICON
Your business. Always on!

Ansible & Windows | WinRM, PSRP & SSH

- Mögliche Connections
  - WinRM - Windows Remote Management
    - https://github.com/AlbanAndrieu | ansible-windows
      - Konfiguration von WinRM Listener
      - Firewall Port 5986 (https)
  - PSRP - PowerShell Remoting Protocol
    - Nutzt WinRM als Transportprotokoll
    - Schneller, Weniger Timeout-Probleme, Besserer Proxy-Support
  - SSH
    - Optional features -> OpenSSH Server
    - services.msc -> Start & Autostart
    - SSH PubKey
  - Ansible User
    - secpol.mmc
      - Security Settings -> Local Policies -> User Rights Assignment -> Impersonate a client after authentication -> [Add User or Group…]
- Ergebnis
  - Sicheres Login, mit unterschiedlich viel Aufwand, aber immer Aufwand
  - Fertig -> Je nach Fetisch, ebenfalls viel Spaß 😅

ASPICON
Your business. Always on!

Ansible & Windows | Händische Einrichtung von WinRM

Wollt ihr das wirklich?

ASPICON
Your business. Always on!

WinRM | Prüfung Listener 1/2

```
PS C:\Windows\system32> get-WSManInstance  -ResourceURI "winrm/config/Listener" -Enumerate


cfg                   : http://schemas.microsoft.com/wbem/wsman/1/config/listener
xsi                   : http://www.w3.org/2001/XMLSchema-instance
lang                  : en-US
Address               : *
Transport             : HTTP
Port                  : 5985
Hostname              :
Enabled               : true
URLPrefix             : wsman
CertificateThumbprint :
ListeningOn           : {127.0.0.1, 192.168.137.88, ::1, fe80::9193:cbf9:bd39:6652%8}
...
```


ASPICON
Your business.
Always on!

WinRM | Prüfung Listener 2/2

```
...
cfg                   : http://schemas.microsoft.com/wbem/wsman/1/config/listener
xsi                   : http://www.w3.org/2001/XMLSchema-instance
lang                  : en-US
Address               : *
Transport             : HTTPS
Port                  : 5986
Hostname              :
Enabled               : true
URLPrefix             : wsman
CertificateThumbprint : 2a0de7828d810bf6c847c6a94948ae2ecead8e4a
ListeningOn           : {127.0.0.1, 192.168.XXX.XXX, ::1, fe80::5efe:192.168.XXX.XXX%14...}


PS C:\Windows\system32> winrm delete winrm/config/Listener?Address=*+Transport=HTTP
```


ASPICON
Your business.
Always on!

Ansible & Windows | Authentifikation

- Mögliche Wege (WinRM & PSRP)
  - Basic
    - Username und Passwort base64 encoded -.-
  - Kerberos
    - Empfohlen für Windows-Domänen
    - Credential Delegation
    - Verschlüsselte Nachrichten per HTTP
  - CredSSP
    - Credential Delegation
    - Verschlüsselte Nachrichten per HTTP
    - Anfällig für Pass-the-Hash & Man-in-the-Middle
  - Certificate
    - Ähnlich SSH PubKey Authentifikation
  - NTLM
    - Windows New Technology LAN Manager
    - Mittlerweile deprecated

ASPICON
Your business.
Always on!

WinRM | Zertifikats-Validierung | 1/3

1. Zertifikat beschaffen
   - Möglichst kein Self-signed
2. Import von …
   - … Host-Zertifikat und -Key (als PFX, P12, …) -> LocalMachine\My
   - … root-CA -> Trusted Root Certificate Authorities (wenn nur Self-signed verfügbar)
3. Ersetzen des automatisch generierten Zertifikat für WinRM
   - Listener löschen
   - Thumbprint holen
   - Neuen Listener anlegen
4. Zertifikats-Chain auf Ansible-Host bereitstellen
5. Inventory entsprechend bestücken
   - `ansible_psrp_cert_trust_path: /path/to/chain`
   - `ansible_psrp_cert_validation: validate (default)`

ASPICON
Your business. Always on!

WinRM | Zertifikats-Validierung | 2/3

```
PS C:\Windows\system32> winrm delete winrm/config/Listener?Address=*+Transport=HTTPS
PS C:\Windows\system32> Get-ChildItem -Path Cert:\LocalMachine\My
...
PS C:\Windows\system32> $thumbprint = "<THUMBPRINT>"
PS C:\Windows\system32> $hostname = "demo-02"
PS C:\Windows\system32> $listenerParams = @{
    ResourceURI = 'winrm/config/listener'
    SelectorSet = @{
        Transport = "HTTPS"
        Address   = "*"
    }
    ValueSet    = @{
        Hostname = $hostname
        CertificateThumbprint = $thumbprint
        Enabled               = $true
        Port                  = 5986
    }
}
PS C:\Windows\system32> New-WSManInstance @listenerParams
```


ASPICON
Your business.
Always on!

WinRM | Zertifikats-Validierung | 3/3

… Self-Signed Zertifikat

```
...
fatal: [demo-01]: UNREACHABLE! => {"changed": false, "msg": "ntlm: HTTPSConnectionPool(host='demo-01', port=5986): Max
retries exceeded with url: /wsman (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: self-signed certificate (_ssl.c:992)')))", "unreachable": true}
```

… anderweitig ungültiges Zertifikat

```
...
fatal: [demo-02]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via PSRP:
HTTPSConnectionPool(host='demo-02', port=5986): Max retries exceeded with url: /wsman (Caused by
SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local
issuer certificate (_ssl.c:992)')))", "unreachable": true}
```


ASPICON
Your business.
Always on!

WinRM | Zertifikats-Authentifizierung | 1/3

```
# Zertifikats-Authentifizierung einschalten
PS C:\Windows\system32> Set-Item -Path WSMan:\localhost\Service\Auth\Certificate -Value $true

# Zertifikat bauen
$> USERNAME="ansible"
$> cat > openssl.conf << EOL
distinguished_name = req_distinguished_name

[req_distinguished_name]
[v3_req_client]
extendedKeyUsage = clientAuth
subjectAltName = otherName:1.3.6.1.4.1.311.20.2.3;UTF8:${USERNAME}@localhost
EOL
$> openssl req -new -sha256 \
    -subj "/CN=${USERNAME}" \
    -newkey rsa:2048 \
    -nodes \
    -keyout ansible.key \
    -out ansible.csr \
    -config openssl.conf \
    -reqexts v3_req_client
```


ASPICON
Your business.
Always on!

WinRM | Zertifikats-Authentifizierung | 2/3

```
$> openssl x509 \
    -req \
    -in ansible.csr \
    -sha256 \
    -out ansible.pem \
    -days 365 \
    -extfile openssl.conf \
    -extensions v3_req_client \
    -key ansible.key

$> rm openssl.conf ansible.csr

$> openssl pkcs12 -export -out ansible.p12 -inkey ansible.key -in ansible.pem
```


ASPICON
Your business.
Always on!

WinRM | Zertifikats-Authentifizierung | 3/3

```
# Import
# Certificates (Local Computer) -> Trusted Root Certification Authorities

# Zertifikat auf den lokalen User mappen
PS C:\Windows\system32> Get-ChildItem Cert:\LocalMachine\Root
...

PS C:\Windows\system32> $thumbprint = "<THUMBPRINT>"
PS C:\Windows\system32> New-Item -Path WSMan:\localhost\ClientCertificate `
         -Subject 'ansible@localhost' `
         -URI * `
         -Issuer $thumbprint `
         -Credential (Get-Credential) `
         -Force

# Es öffnet sich ein Dialog
# User Credentials eingeben und bestätigen


# Fertig
```


ASPICON
Your business.
Always on!

SSH | Installation OpenSSH Server

```
PS C:\> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'


Name  : OpenSSH.Client~~~~0.0.1.0
State : Installed

Name  : OpenSSH.Server~~~~0.0.1.0
State : Installed


PS C:\> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0


PS C:\> Start-Service sshd
PS C:\> Set-Service -Name sshd -StartupType Automatic



$> nc -vz demo-03 22
Connection to demo-03 (192.168.XXX.XXX) 22 port [tcp/ssh] succeeded!
```


ASPICON
Your business.
Always on!

SSH | Konfiguration OpenSSH Server

```
# C:\ProgramData\ssh\sshd_config

...
# Logging
SyslogFacility LOCAL0
LogLevel DEBUG1
...

...
# Common PubKey
#Match Group administrators
#       AuthorizedKeysFile __PROGRAMDATA__/ssh/administrators_authorized_keys



# Kein Eintrag in der ~/.ssh/config der Workstation
```


ASPICON
Your business.
Always on!

SSH | Einrichtung Public-Key-Authentifizierung

```
PS C:\> ssh -Q key
ssh-ed25519
ssh-ed25519-cert-v01@openssh.com
...

$> ssh ansible@demo-03
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

ansible@DEMO-03 C:\Users\ansible>mkdir .ssh
ansible@DEMO-03 C:\Users\ansible>exit

$> cat ~/.ssh/id_ed25519.pub | ssh ansible@demo-03 "echo $(</dev/stdin) >> .ssh\authorized_keys"
$> ssh -i ~/.ssh/id_ed25519.pub ansible@demo-03

Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

ansible@DEMO-03 C:\Users\ansible>
```


ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

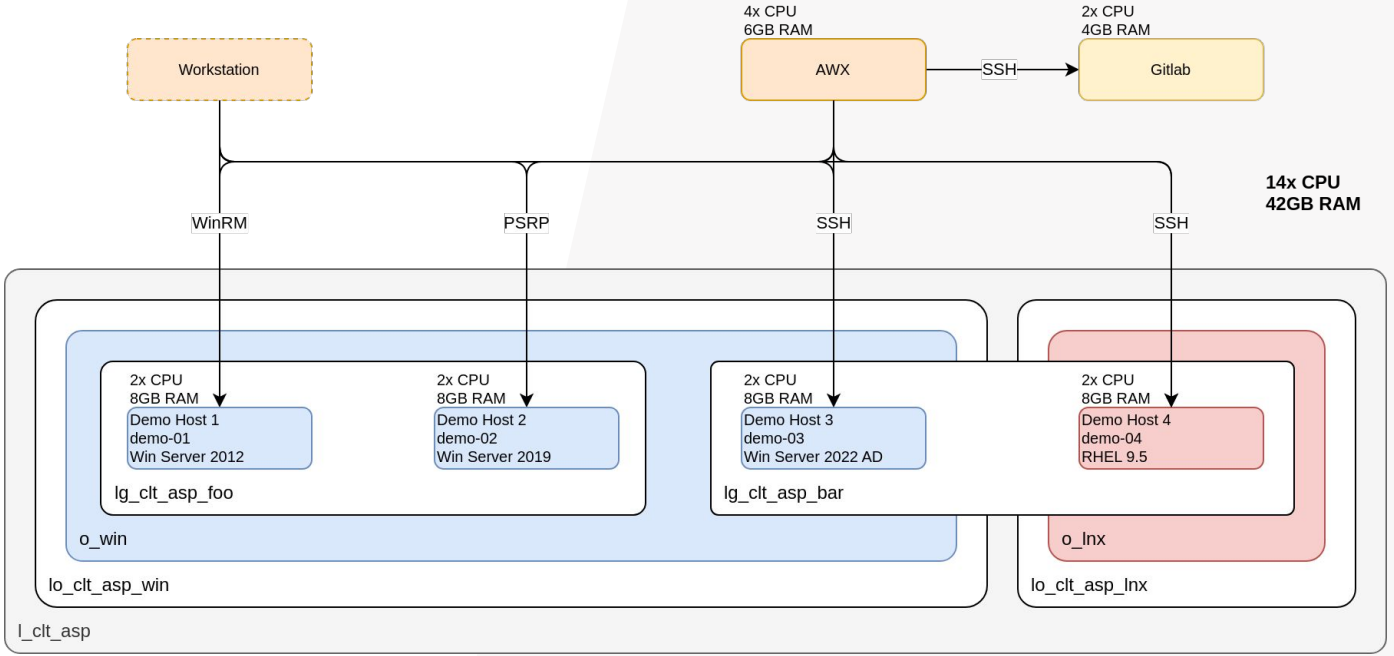
Aufbau Demo

Environment & Inventory

Generated by Adobe Firefly

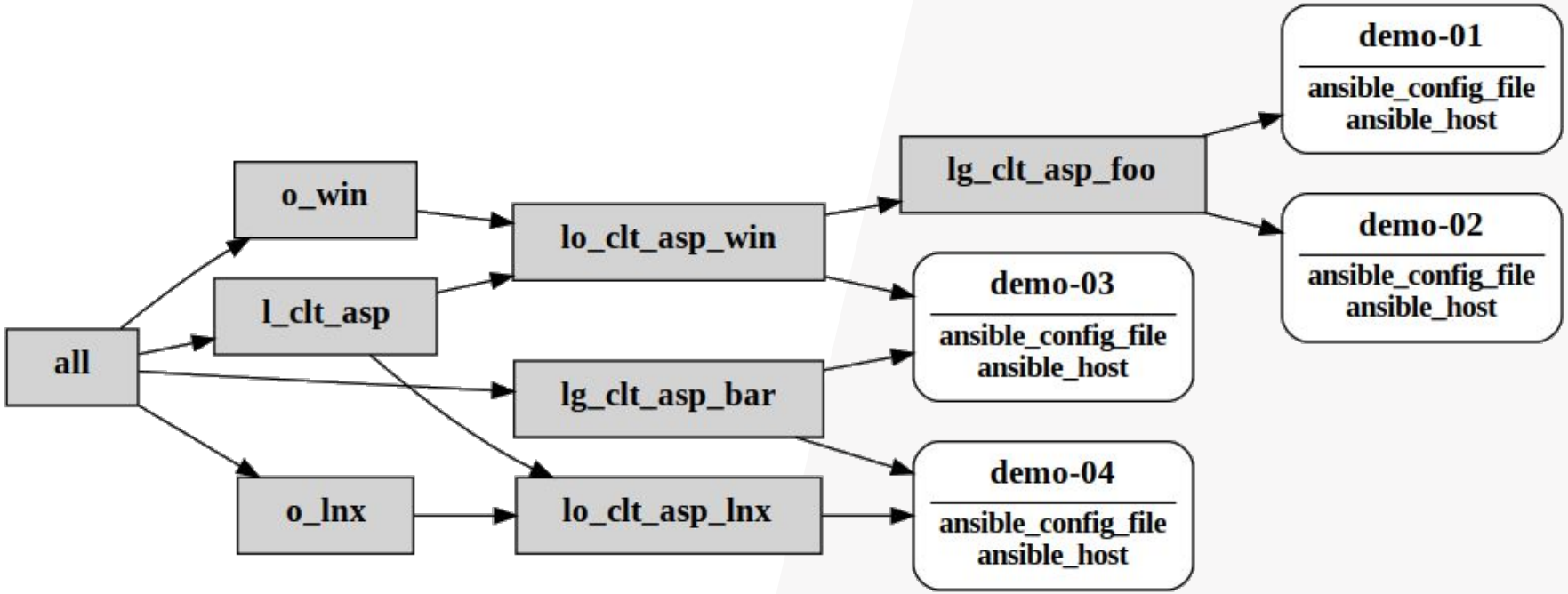
Demo Environment

- Windows
  - Server 2012
  - Server 2019
  - Server 2022
- Linux
  - Red Hat 9.5
- AWX
- GitLab Omnibus



ASPICON
Your business.
Always on!

Demo Inventory | Hosts und Gruppen



ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Definition der Connection-Variablen

Generated by Adobe Firefly

Demo Inventory | Erste Schritte

```yaml
---
# demo-01 - Win Server 2012 R2
# WinRM

ansible_connection: winrm
# ansible_winrm_transport: ntlm
# ansible_winrm_scheme: https
# ansible_winrm_message_encryption: auto
ansible_winrm_server_cert_validation: ignore


ansible_user: ansible
ansible_password: !vault |...

# ansible_become: false
ansible_become_method: runas
ansible_become_user: Administrator
# ansible_become_pass: !vault |...
```

```yaml
---
# demo-02 - Win Server 2019
# PSRP + Certificates

ansible_connection: psrp
# ansible_psrp_auth: ntlm
# ansible_psrp_protocol: https
# ansible_psrp_message_encryption: auto
ansible_psrp_cert_validation: ignore


ansible_user: ansible
ansible_password: !vault |...

# ansible_become: false
ansible_become_method: runas
ansible_become_user: Administrator
# ansible_become_pass: !vault |...
```

```yaml
---
# demo-03 - Win Server 2022 21H2
# SSH

# ansible_connection: ssh
ansible_shell_type: cmd




ansible_user: aspicon_ansible
# ansible_password: !vault |...

# ansible_become: false
ansible_become_method: runas
ansible_become_user: Administrator
# ansible_become_pass: !vault |...
```


ASPICON
Your business.
Always on!

SSH | ansible_shell_type nicht korrekt gesetzt

```
...
demo-03 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to create temporary directory. In some cases, you may have been able to authenticate and did not have
permissions on the target directory. Consider changing the remote tmp path in ansible.cfg to a path rooted in \"/tmp\",
for more error information use -vvv. Failed command was: ( umask 77 && mkdir -p \"` echo
C:/Users/ansible/AppData/Local/Temp `\"&& mkdir \"` echo
C:/Users/ansible/AppData/Local/Temp/ansible-tmp-1741340014.959235-1007097-216899786290231 `\" && echo
ansible-tmp-1741340014.959235-1007097-216899786290231=\"` echo
C:/Users/ansible/AppData/Local/Temp/ansible-tmp-1741340014.959235-1007097-216899786290231 `\" ), exited with result 1",
    "unreachable": true
}



...
fatal: [demo-03]: FAILED! => {"ansible facts": {"discovered interpreter python": "/usr/bin/python"}, "changed": false,
"module stderr": "The system cannot find the path specified.\r\n", "module_stdout": "", "msg": "MODULE FAILURE\nSee
stdout/stderr for the exact error", "rc": 1}
```


ASPICON
Your business.
Always on!

Demo Inventory | Zertifikats-Validierung

- Connection: `psrp`
- Authentication: `ntlm`
- Protokoll: `https`
  - `port` != 5985 or `port` not defined
- Message Encryption: `always`
  - `auto` -> `true` when not TLS/HTTPS
- Zertifikatsvalidierung: **`validate`**
  - war ja der Sinn der Übung

```
_

ansible_connection: psrp

ansible_psrp_auth: ntlm
ansible_psrp_protocol: https
ansible_psrp_message_encryption: always



ansible_psrp_ca_cert: "{{ clt_asp_cert_base_path }}/authority/rootca.chain"
ansible_psrp_cert_validation: validate

ansible_user: ansible
ansible_password: !vault |

ansible_become: false
ansible_become_method: runas
```


ASPICON
Your business.
Always on!

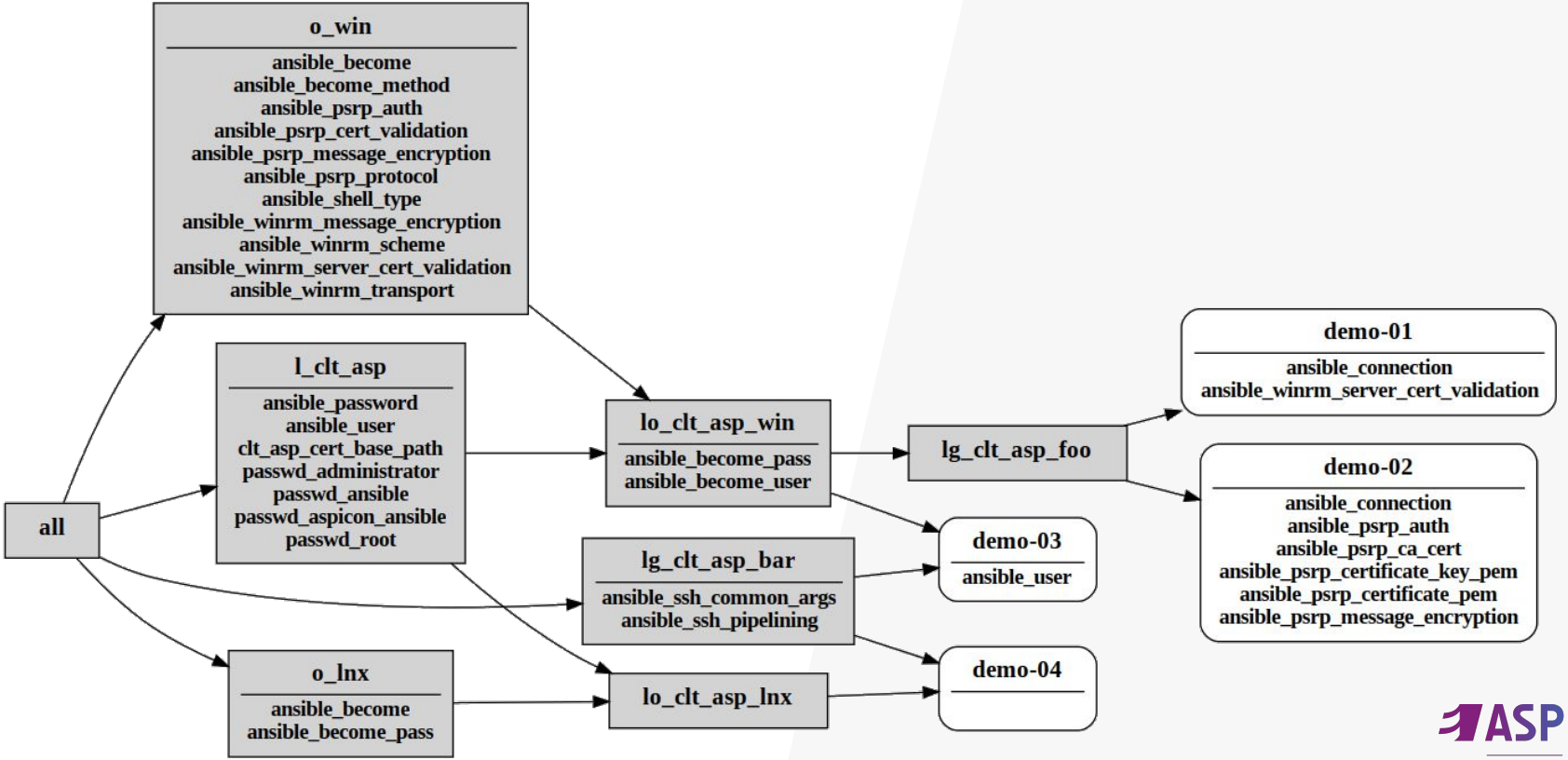
Demo Inventory | Zertifikats-Authentifizierung

- Connection: `psrp`
- Protokoll: `https`
- Message Encryption: `auto (default)`
- Validierung: `validate`
- **`ansible_psrp_auth: certificate`**
- **`ansible_psrp_certificate_pem`**
- **`ansible_psrp_certificate_key_pem`**

```
---
ansible_connection: psrp

ansible_psrp_auth: certificate
ansible_psrp_certificate_pem: "{{ ... }}/clients/ansible.pem"
ansible_psrp_certificate_key_pem: "{{ ... }}/clients/ansible.key"

ansible_psrp_protocol: https
ansible_psrp_ca_cert: "{{ clt_asp_cert_base_path
}}/authority/rootca.chain"
ansible_psrp_cert_validation: validate

ansible_user: ansible
ansible_password: !vault |

ansible_become: false
ansible_become_method: runas
```


ASPICON
Your business.
Always on!

Demo Inventory



ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Demo pb_whoami

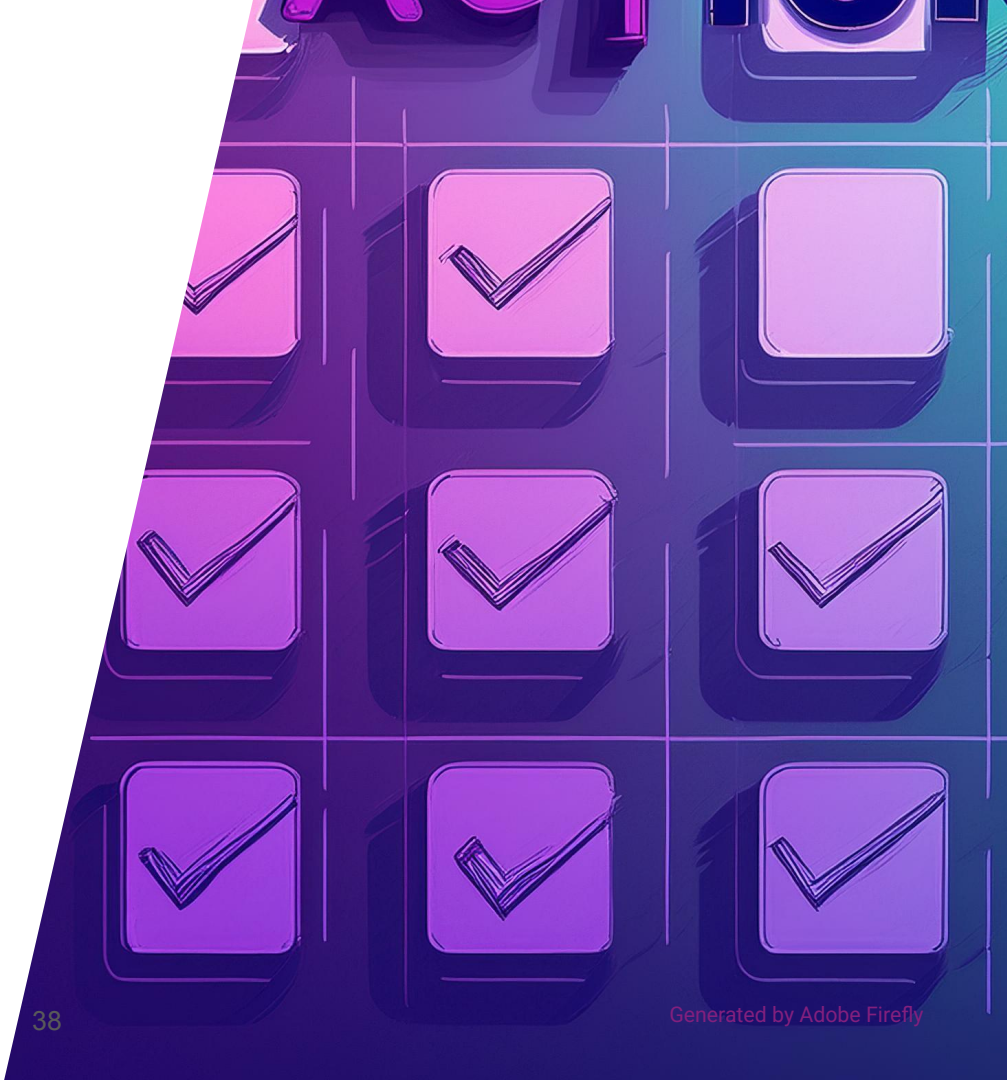
Generated by Adobe Firefly

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Benchmark

WinRM vs. PSRP vs. SSH vs. SSH



Generated by Adobe Firefly

Benchmark | WinRM vs. PSRP vs. SSH vs. SSH

```yaml
- name: Benchmark windows hosts
  block:
  # < / Benchmark windows hosts >
  - name: Ensure the benchmark user exists
    ansible.windows.win_user:
      name: ansible_test
      password: "SecureP@ssw0rd!"
      state: present

  - name: Ensure correct description of the benchmark user
    ansible.windows.win_user:
      name: ansible_test
      description: "Benchmark user for CLT 2025"

  - name: Ensure a temporary file exists
    ansible.windows.win_tempfile:
      state: file
    register: temp_file_win

  - name: Ensure service is running
    ansible.windows.win_service:
      name: wuauserv
      state: started

  - name: Ensure the benchmark user is removed
    ansible.windows.win_user:
      name: ansible_test
      state: absent
  # < / Benchmark windows hosts >
  when: ansible_os_family == "Windows"
```

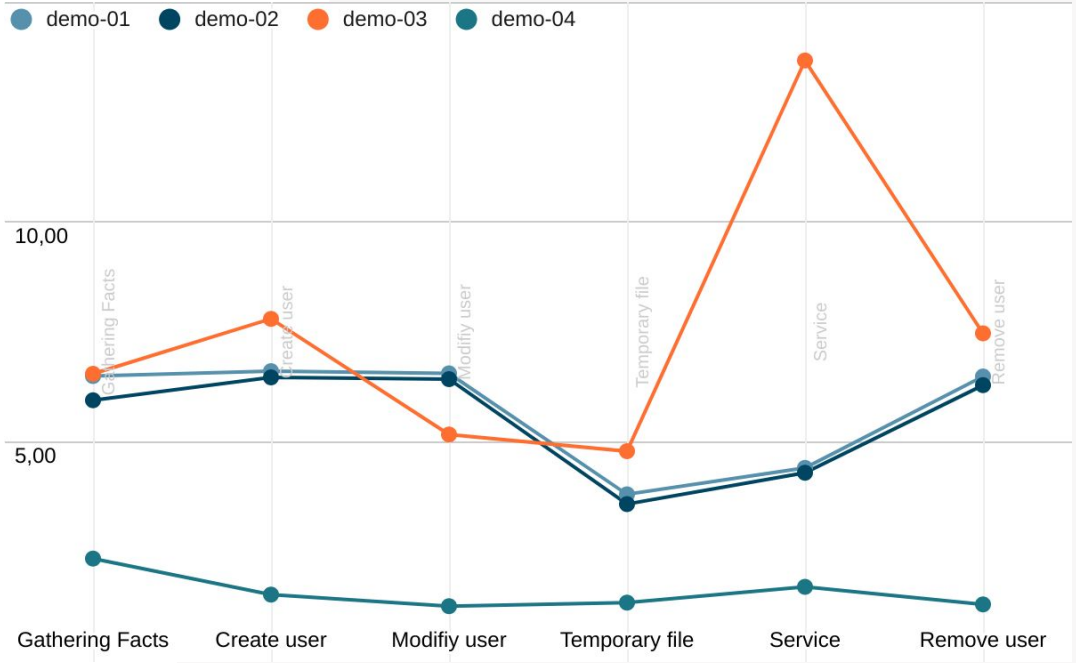
```yaml
- name: Benchmark linux hosts
  block:
  # < Benchmark linux hosts >
  - name: Ensure the benchmark user exists
    ansible.builtin.user:
      name: ansible_test
      password: "{{ 'SecureP@ssw0rd!' | password_hash('sha512') }}"
      state: present

  - name: Ensure correct shell of the benchmark user
    ansible.builtin.user:
      name: ansible_test
      shell: /bin/bash

  - name: Ensure a temporary file exists
    ansible.builtin.tempfile:
      state: file
    register: temp_file_linux

  - name: Ensure service is running
    ansible.builtin.service:
      name: crond
      state: started

  - name: Ensure the benchmark user is removed
    ansible.builtin.user:
      name: ansible_test
      state: absent
  # < / Benchmark linux hosts >
  when: ansible_os_family != "Windows"
```


ASPICON
Your business.
Always on!

Benchmark | WinRM vs. PSRP vs. SSH vs. SSH

Playbook: `pb_bench.yml`

- Ensure the benchmark user exists
- Ensure correct description/shell of the benchmark user
- Ensure a temporary file exists
- Ensure service is running
- Ensure the benchmark user is removed



ASPICON
Your business. Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Häufig gestellte Fragen

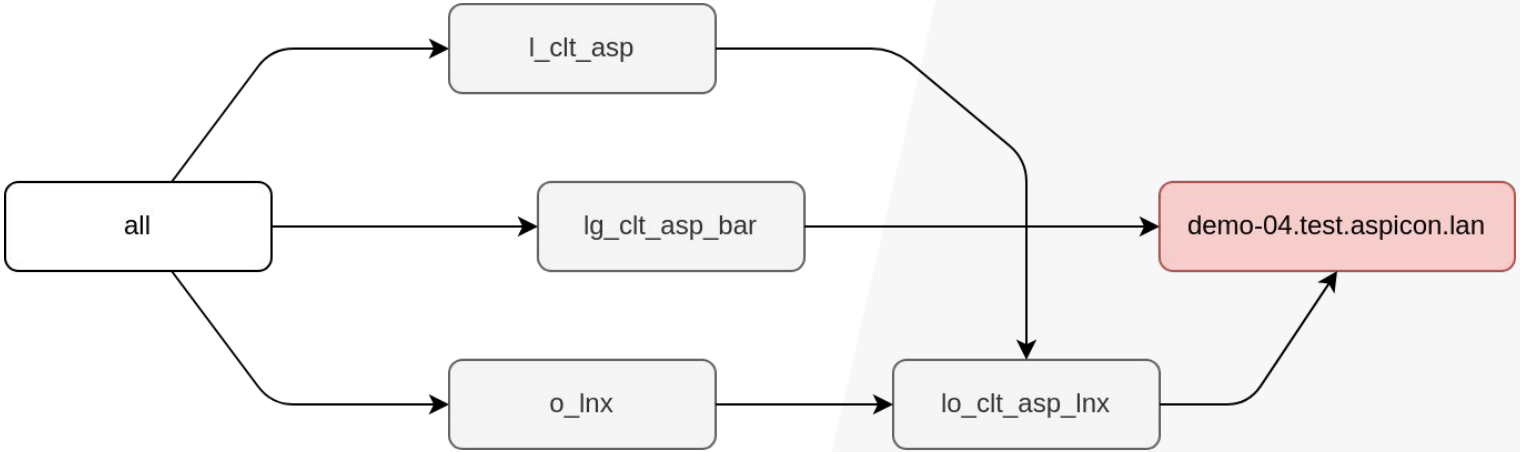
Generated by Adobe Firefly

Aber Micha, du benutzt gar nicht ansible_winrm_user und …_password bzw. ansible_psrp_user …

- Ansible remote user
  - remote_user
  - ansible_user
  - ansible_winrm_user
  - ansible_psrp_user
- Ansible remote user password
  - ansible_password
  - ansible_ssh_pass
  - ansible_ssh_password
  - ansible_winrm_pass
  - ansible_winrm_password
- Zertifikatsvalidierung
  - cert_validation
  - ansible_psrp_cert_validation
- PEM certificate chain
  - ca_cert
  - ansible_psrp_ca_cert
  - ansible_psrp_cert_trust_path
- ...

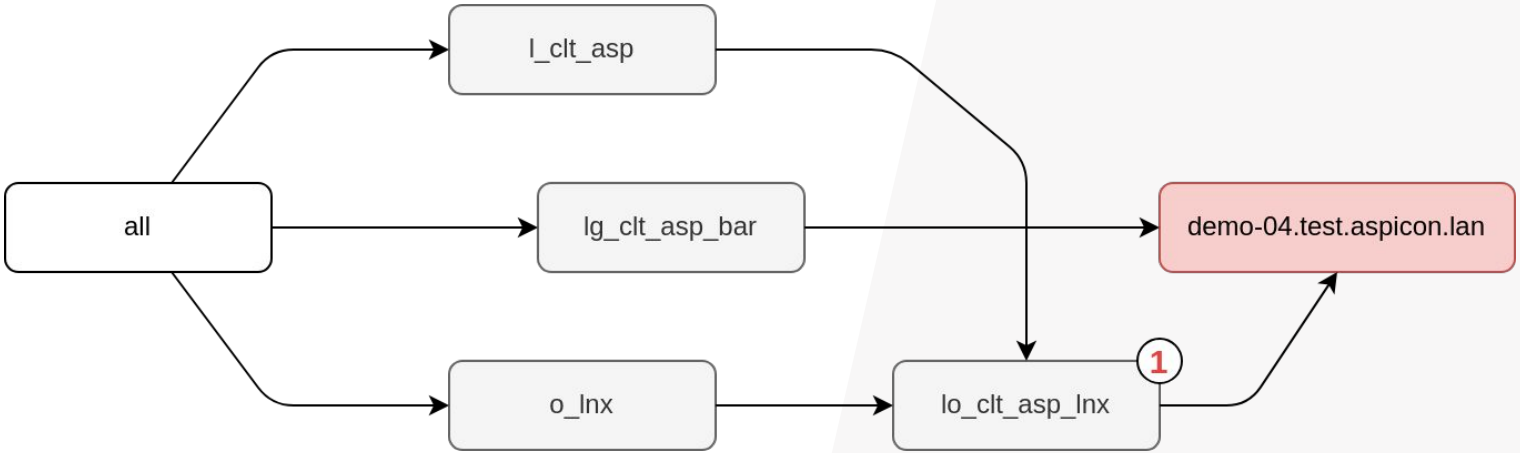
ASPICON
Your business. Always on!

Die `host_vars` überschreiben `group_vars`, das ist klar. Aber Micha, wie zur Hölle funktioniert das innerhalb der `group_vars`, das ist doch alles eine Ebene …



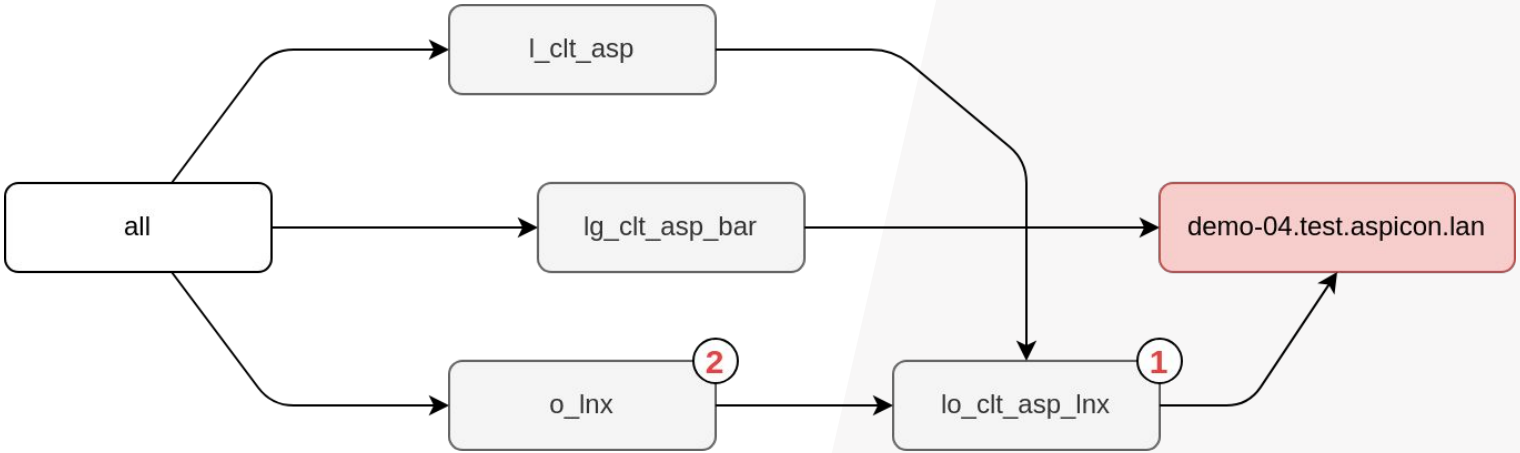
ASPICON
Your business.
Always on!

Precedence innerhalb von `group_vars`



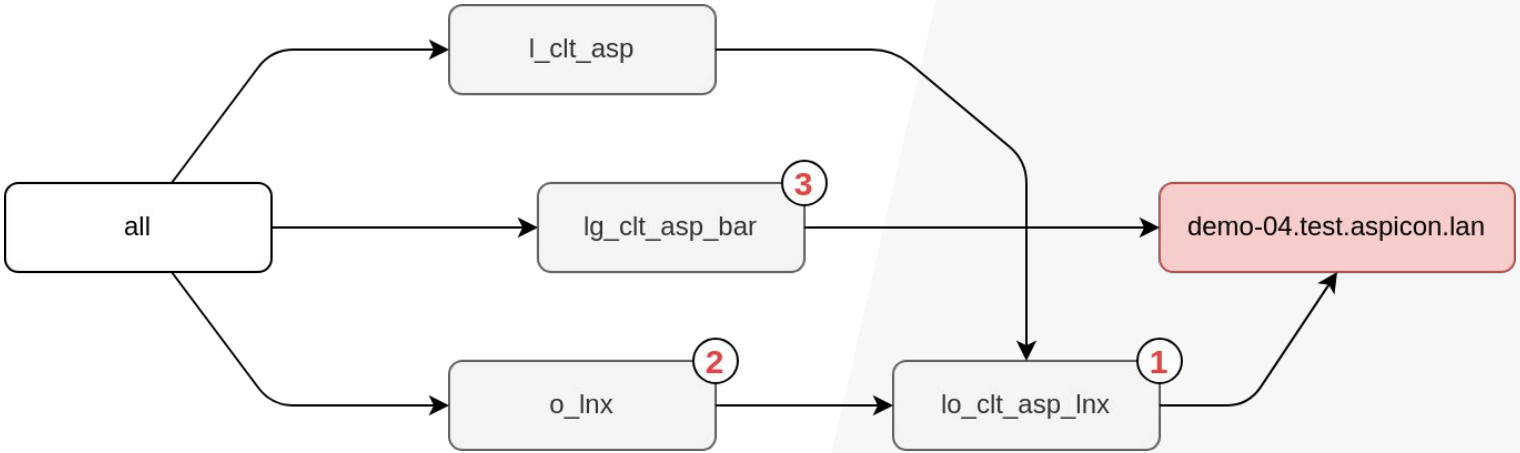
ASPICON
Your business.
Always on!

Precedence innerhalb von `group_vars`



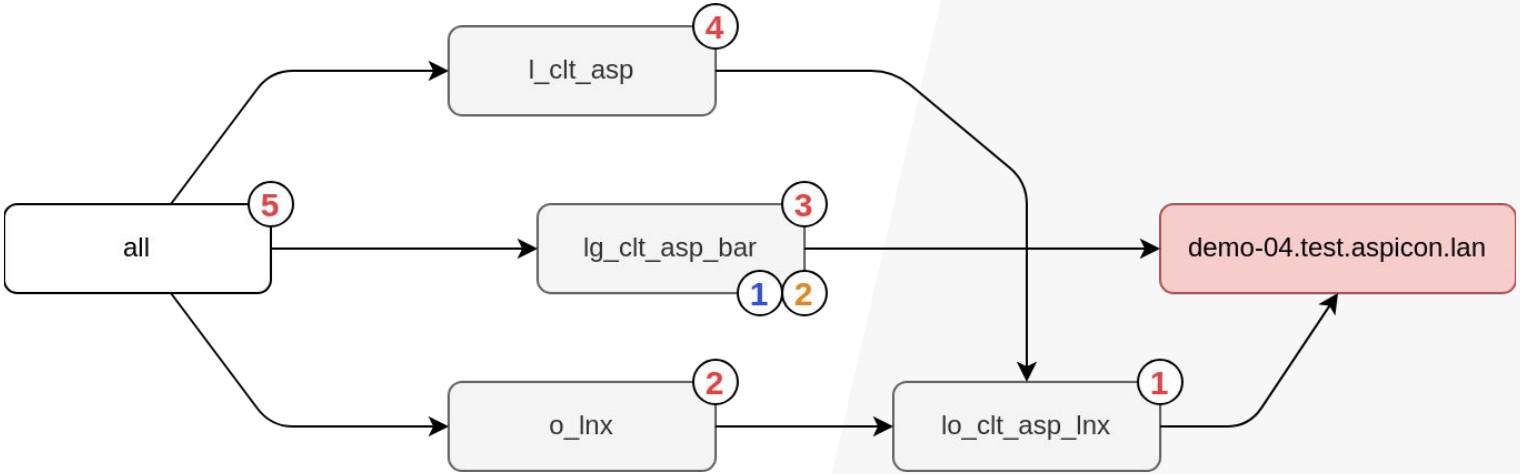
ASPICON
Your business.
Always on!

Precedence innerhalb von `group_vars`



ASPICON
Your business.
Always on!

Precedence innerhalb von `group_vars`



ASPICON
Your business.
Always on!

User bzw. Gruppen einschränken, Become mit anderen Credentials

- Einschänkung Login
  - AllowGroups
  - AllowUsers
  - DenyGroups
  - DenyUsers
- Eskalation mit zweitem Satz Credentials
  - SELF_AUTH -> Ausnahmen
  - TARGET_AUTH
    - Alle die nicht in der anderen Gruppe sind
    - Authentication mit den Credentials des Ziel-Users

```
#> cat /etc/ssh/sshd_config |
grep -i 'allow\|deny'
AllowGroups root ansible aspicon
DenyGroups e39d05b72f2...
DenyUsers e39d05b72f25...
```

```
#> cat /etc/sudoers.d/ansible
# SELF_AUTH: use their own password to sudo
User_Alias SELF_AUTH = foobar

# TARGET_AUTH: use target's password to sudo
User_Alias TARGET_AUTH = ALL,!SELF_AUTH

Defaults:TARGET_AUTH
!requiretty,targetpw,timestamp_timeout=0
Defaults:SELF_AUTH
!requiretty,!targetpw,timestamp_timeout=0
TARGET_AUTH ALL=(ALL) ALL
SELF_AUTH ALL=(ALL) ALL
```


ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Fragen?

Generated by Adobe Firefly

Links

- Ansible Doku
  - builtin-winrm-connection
  - builtin-psrp-connection

- Demo aus dem Vortrag -> clt-demo_2025
- ansible-inventory-grapher
- ConfigureRemotingForAnsible.ps1

- Das und mehr traust du dir auch zu? -> aspicon.de/job-karriere

ASPICON
Your business.
Always on!

ASPICON
Your business.
Always on!

Ansible Automatisierung in hybriden Umgebungen

Danke für eure Aufmerksamkeit