PUZZLE ITC
changing IT for the better

LDAP and Ansible?

LDAP as Ansible's Source of Truth

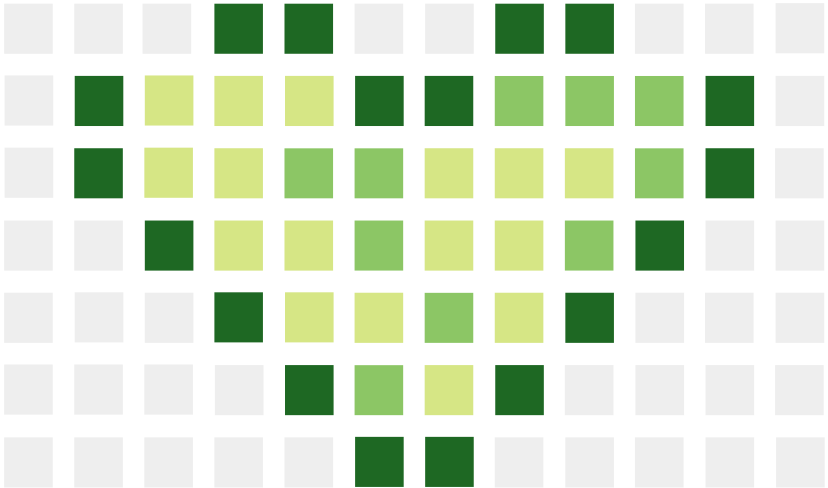
Pröhl, Mark
proehl@puzzle-itc.de

Agenda

- Introduction
- Ansible Overview
- LDAP in the Ansible Ecosystem
- LDAP as the “Source of Truth” in Ansible?
- Design
- Demo and Discussion

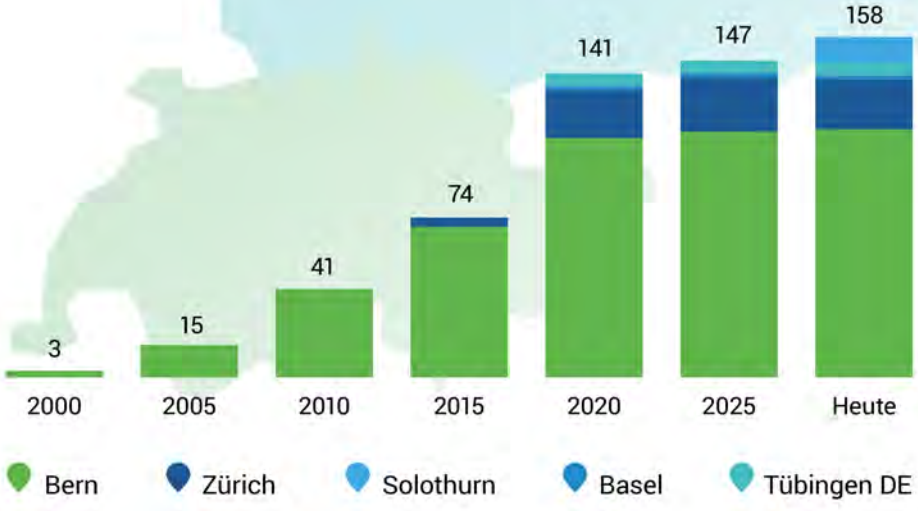
Introduction

CHANGING IT FOR THE BETTER
PUZZLE ITC
WERTE
INFRASTRUCTURE & CLOUD
PLATFORM ENGINEERING
SOFTWARE DEVELOPMENT & ARCHITECTURE
DIGITAL TRANSFORMATION
PROJEKTE MANAGEMENT CONSULTING
PUZZLE MOBILITY



We contribute

Locations & Members

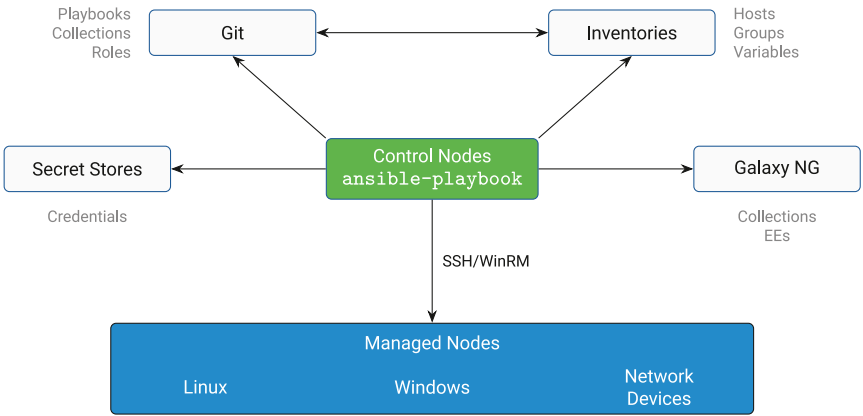


Nice to meet you

```
givenName: Mark
sn:: UHLDtmhsCg==
mail: proehl@puzzle-itc.de
url: https://www.puzzle-itc.de/author/mproehl
labeledURI: Mastodon:https://mastodon.social/@mproehl
```

Infrastructure Compononents
Inventories
Variables

Ansible Infrastructure Compononents



Ansible Inventory (beginner level)

```
[all]
node0001 ansible_host=node0001.example.com
node0002 ansible_host=node0002.example.com
node0003 ansible_host=node0002.example.com

[dev]
node0001 foo=bar

[prod]
node0002
node0003

[all:vars]
ansible_user = ansible
ansible_password = P@ssw0rd

[prod:vars]
ansible_password = T0pS3cr3t
```

- INI file (e.g. `hosts.ini`)
- Hosts: `node0001`, `node0002`, `node0003`
- Groups: `all`, `dev`, `test`
- Variables: `ansible_host`, `ansible_user`, `ansible_password` and `foo`

Ansible Inventory with `host_group_vars`

hosts.ini

```
[all]
node0001
node0002
node0003

[dev]
node0001

[prod]
node0002
node0003
```

host_vars/node0001.yml

```
ansible_host: node0001.example.com
foo: bar
```

host_vars/node0002.yml

```
ansible_host: node0002.example.com
```

host_vars/node0003.yml

```
ansible_host: node0003.example.com
```

group_vars/all.yml

```
ansible_user: ansible
ansible_password: P@ssw0rd
```

group_vars/prod.yml

```
ansible_user: ansible
ansible_password: T0pS3cr3t
```

Ansible Variables

- Simple variables (integers, strings, …), e.g.:

```
ldap_port: 389
openldap_version: "2.6.15"
foo: bar
```

- Complex data structures possible:

```
users:
  - name: John Doe
    phone: 1234567
    mail: jdoe@example.com
  - name: Max Muster
    phone: 9876543
    mail: maxm@example.com
```

- Global variable namespace per managed node
- Common practice:
  manage all variables in a git-base inventory

LDAP in the Ansible Ecosystem

```
community.general.ldap_*
microsoft.ad.*
```

Ansible Collection `community.general`

These modules are primarily used to manage LDAP entries:

- `community.general.ldap_attrs`
- `community.general.ldap_entry`
- `community.general.ldap_inc`
- `community.general.ldap_passwd`
- `community.general.ldap_search`

Exeception: `community.general.ldap_search` performs LDAP search operations. Search results can be further processed within Ansible Playbooks.

Very little functionality related to inventory and variables

Ansible Collection `microsoft.ad`

Collection for managing Active Directory via Ansible

Many plugins (Filter, Inventory) and Modules

Some are also useful for native LDAP

The inventory plugin `microsoft.ad.ldap`, in particular, is a good fit here

Inventory plugin `microsoft.ad.ldap`

YAML config file:

```
---
plugin: microsoft.ad.ldap
server: dc1.example.com
port: 389
auth_protocol: simple
username: cn=ansible_bind,dc=example,dc=com
password: P@ssw0rd
search_base: dc=example,dc=com
search_scope: subtree
filter: "(objectCategory=Computer)"
...
```

Some imperfections

- File name has to be “`*.microsoft.ad.ldap.yml`”
- AD attributes `name` and `dnsHostName` are hard-coded

Inventory Variable with `microsoft.ad.ldap`

YAML config file (continued):

```
...
attributes:
  sAMAccountName:
    sam_account_name:
  objectSid:
    computer_sid:
  memberOf:
    computer_membership: >
      this
      | microsoft.ad.parse_dn
      | map(attribute="0.1")

...
```

Define variables:

- `attributes:` defines Ansible variables from LDAP attributes
- `compose:` creates variables using Jinja2 filters

Inventory Groups with `microsoft.ad.ldap`

YAML config file (continued):

```
groups:
  windows: true
  production: '"Production Group" in computer_membership'
keyed_groups:
  - key: location | default(omit)
    prefix: site
    default_value: unknown
```

(Examples are from the modules documentation page)

Other LDAP-related Ansible stuff…

- Management of 389DS and Red Hat Directory Server: `389ds.ansible_ds`
- Management of OpenLDAP Deployments: `mrlesmithjr.openldap`
- LDAP Lookup Plugin by Thomas Quinot
  `https://github.com/quinot/ansible-plugin-lookup_ldap`

(Included for completeness)

LDAP as the “Source of Truth” in Ansible?

Shortcomings

Our Vision

Standard Approach Shortcomings

- Only flat data structures
- Often simple mapping of strings or integers to variables
- They focus on one implementation (Active Directory bias)
- Lacking flexibility for custom or generic LDAP schemas
- Constrained by existing directory structures and contents
- Do not embrace posibillity of creating a custom LDAP service purposely built as an Ansible configuration backend

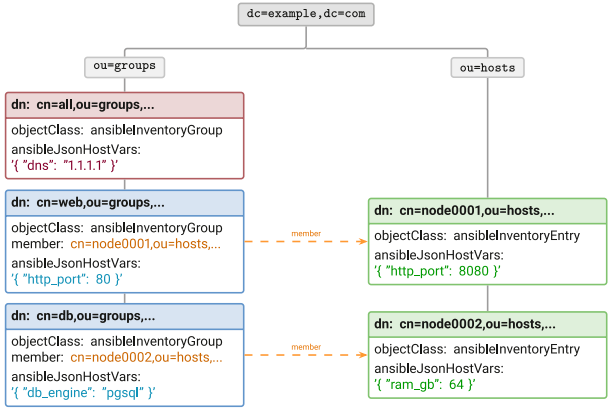
Our Vision

- Centralized configuration payloads: Entire, complex infrastructure configurations reside directly on the LDAP objects (e.g., something like `ansibleInventoryEntry` or `ansibleInventoryGroup` object classes).
- Storing JSON data in specialized attributes (something like `ansibleJsonHostVars`)
- Hierarchical lookups, similar to Jerakia or Hiera, but strictly LDAP-native. Variable inheritance in three stages:
  - *Base:* The `all` group provides the global foundation
  - *Groups:* Host groups (like web, db) add to or selectively overwrite sections.
  - *Host:* The specific host object overrides the most granular values.
- Custom `inventory`, `lookup` and `vars` plugins implement the LDAP search operations and the data merging

Design

LDAP Structure
Complex Data Structures
Hierarchical Variable Lookup

LDAP Structure for Hierarchical Lookups



Complex Data Structures

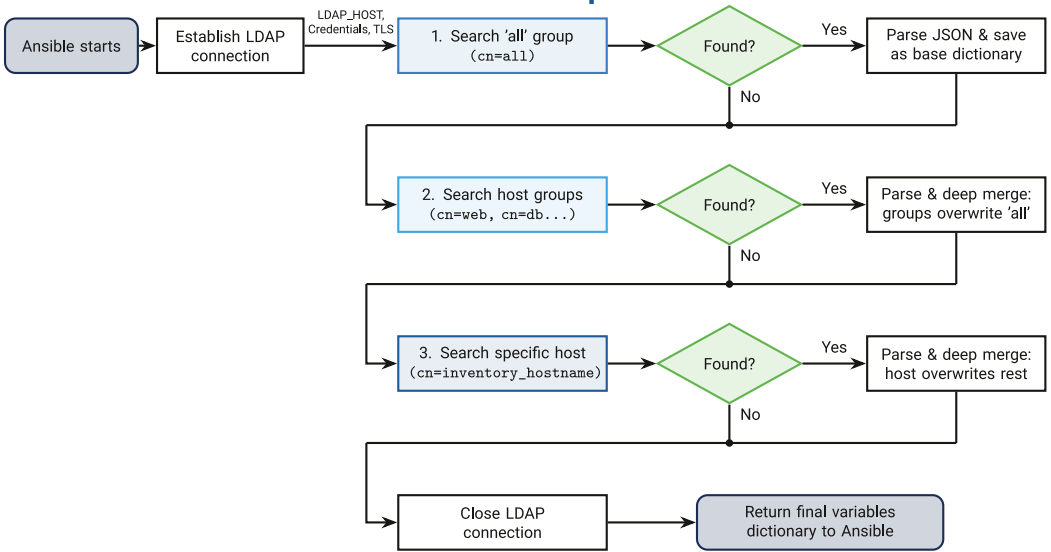
dn: cn=all,ou=groups,dc=example,dc=com

objectClass: puzzleAnsibleInventoryGroup

puzzleAnsibleJsonHostVars: '{ "dns_servers": ["1.1.1.1", "1.0.0.1"] }'
puzzleAnsibleJsonHostVars: '{ "timezone": "Europe/Berlin" }'
puzzleAnsibleJsonHostVars:

```
'{      "system_config": {
        "firewall": {
          "default_policy": "drop",
          "open_tcp_ports": [22],
          "logging": {
            "level": "warning",
            "remote": true
          }
        },
        "monitoring": {
          "agent": "prometheus",
          "enabled": true
        }
    }
}'
```

Hierarchical Variable Lookups



Demo and Discussion

Merci!

Mehr Informationen zu Puzzle:
www.puzzle.ch