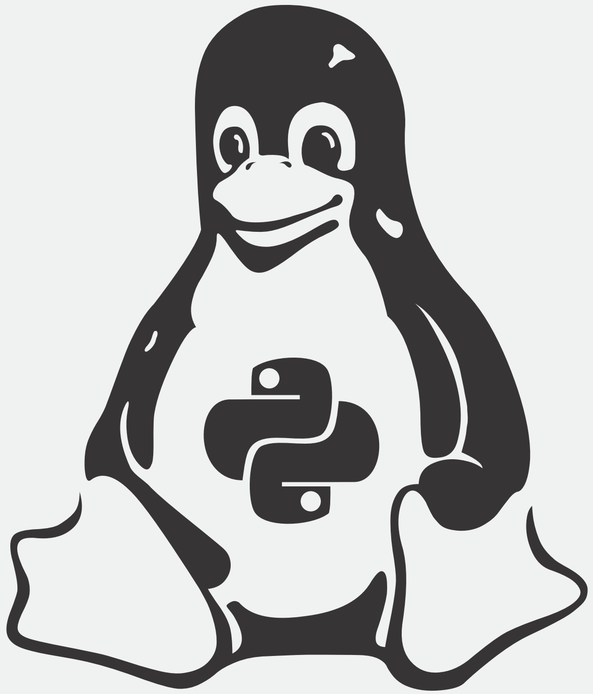
Managing sensitive data with Ansible vault

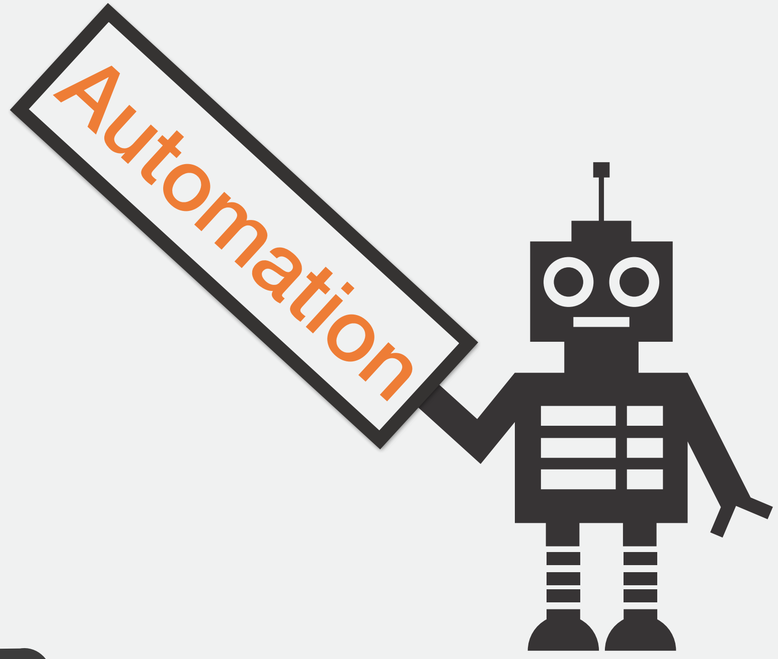
confirm IT SOLUTIONS

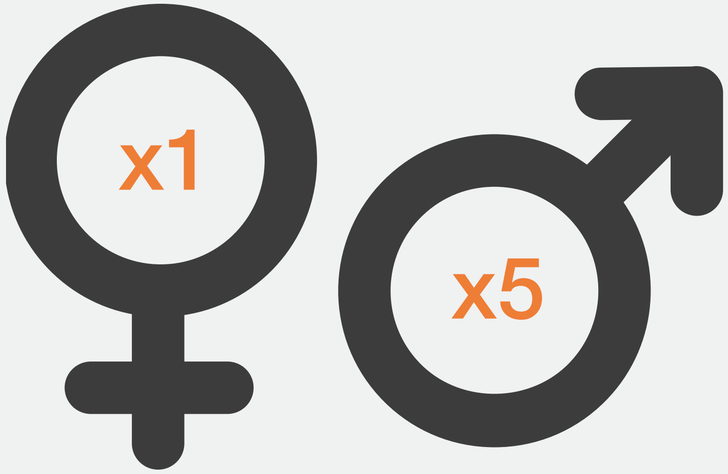
Introduction

About me and our company



About Me





About Us

- Linux & AIX **consulting** with **DevOps** experience
- Infrastructure **Automation**
- official Ansible **Partner**
- **consulting** & **training** for Ansible, Python & git

What we offer

Ansible Vault

getting started

Vault is a feature of ansible that allows **keeping sensitive data** such as passwords or keys in **encrypted files**, rather than as plaintext in your playbooks or roles.

What is Ansible vault

- Binary included in the **Ansible core** package
- **AES-256** algorithm encrypted
- Decrypted **on runtime**
- **Limitation**: one Vault password per Ansible playbook

Tech Specs

ansible-vault <option>

create	Create new encrypted file
encrypt	Encrypt existing file
edit	Edit encrypted file
rekey	Change encryption password
view	View encrypted file

How to use it

1. Create **ansible-vault** variable file
2. Use it in your Ansible Project
3. Run ansible / ansible-playbook with
    - --**ask-vault-pass** Option
    - or define a **password file**
        - define in ansible.cfg **vault_password_file** path
        - **--vault-password-file** Option
        - as variable **ANSIBLE_VAULT_PASSWORD_FILE**

- encrypt **YAML files**
  - e.g. group_vars
  - e.g. host_vars
- since Ansible v2.3 **single encrypted variables**
  - use the **!vault tag**

What can be encrypted

- sensitive data for automated deployments
  - SSL private keys
  - SSH private keys
  - secrets / credentials

What should be encrypted

Tips & Tricks

make your life easier

- **layer of indirection**
- **prefix** your variables **vault_**<variablename>
- **save variables** into vault files or directories

implicit		explicit
group_vars/dev/vault.yml	vs	vault_vars/dev.yml
group_vars/prod/vault.yml		vault_vars/prod.yml

Define your variables

- set ”**no_log: true**” per task
- use **different passwords per environments**
- only **encrypt sensitive data**
- use a **strong encryption password**
- always use a **private git repo / restrict access**

Best Practices

- technically could be **still compromised**
- what **pushed to git, stays on git**
- secure your password file
  - **owner** & file **permissions**
  - **outside** the git repo / .gitignore

Are we safe now?

Demo

managing SSL private keys

use case

managing database credentials

use case

single encrypted variable

short demo

The End

Thank you for listening


pstauffer8


pstauffer


https://www.xing.com/profile/Pascal_Stauffer


https://www.linkedin.com/in/pascal-stauffer-5030775b


confirm.ch


blog.confirm.ch

- http://docs.ansible.com/ansible/playbooks_best_practices.html#variables-and-vaults
- https://blog.confirm.ch/deploying-ssl-private-keys-with-ansible
- http://docs.ansible.com/ansible/playbooks_vault.html
- http://docs.ansible.com/ansible/playbooks_vault.html#single-encrypted-variable

Source