SAS® Configuration Management with Ansible
What is configuration management?

“Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.”

ANSI-EIA-649-A Standard: NATIONAL CONSENSUS STANDARD FOR CONFIGURATION MANAGEMENT
Why is configuration management important?

The practice of handling changes systematically so that a system maintains its integrity over time

• Reduce risks due to ad-hoc changes
  • Operating System Configuration
  • SAS Home Configuration
  • SAS Level Configuration

Configuration Management for SAS

• Can be applied to many aspects of a “system”

  • Data Integration/ETL
  • Reports
  • Custom built SAS Applications/Programs
  • Environment
SAS Environment configuration

Overview

• We will look at three configuration aspects of a SAS Environment

• Pre-requisites
• Binaries (SAS Home)
• Configuration (e.g. Lev1)
SAS Environment

Pre-requisites

• Users
  • SAS Installation User
  • SAS Spawned Servers User

• Groups
  • SAS Server Users

• Directory Permissions
  • For SAS Home
  • For SAS Config
  • For SAS Work

• Disks
  • Space

• Authentication
  • Active Directory
  • LDAP

• Operating System Rights
  • Log on as a Batch Job

• Kernel Parameters
  • Open Files
SAS Environment

SAS Home Directory

- File permissions
  - setuid bit
- sassw.config
  - SASENVIRONMENTSURL
- sasv9_local.cfg
  - -WORK
  - -MEMSIZE

- SAS Deployment Agent
  - daemon/service
- SAS Data Management Server
  - daemon/service
- Deployment Registry
  - Check for Hot Fixes
SAS Environment
Configuration (e.g. Lev1) Directory

- Directory Permissions
- Object Spawner config
- SASApp
  - *_usermod files
- SASMeta
  - Security
  - User Sync
Ansible

What is Ansible?

- Multiple hosts
- On-premise / Multi-Cloud / Hybrid
- No agents required
- Supports *nix / Windows
- Extendable
Ansible
The Controller

ansible Host

Linux OS or Windows Subsystem for Linux
Communication with all hosts to be managed.
Python 2.7 / 3x
Ansible

Your Hosts

- Can communicate with your controller
- Multi - OS Supported.
- Windows requires Powershell remoting.
- SSH Preferred
# Ansible Project

## Variables
- Host
- Group
- Task

## Inventory

## Facts

## Plays

## Templates

## Roles
Ansible Inventories

- Dynamic or Static.
- Define your servers and group them by management function.
- E.G: AWS, tags are your friends.
- Connection params support using variables.
Ansible

Facts

- Describe your Hosts and Applications.
- Ansible will collect Host Facts.
- Custom processes can create Application facts.
- For SAS,
  - sasmw
  - sas_config
  - sas_install
Ansible Plays

- Execute commands on your target hosts.
- Executed within a Playbook
- Accepts input/output variables, filters, looping
- Can be async.

```yaml
- name: Gather IR Studio Config Files
  become: yes
  become_user: sas
  become_method: sudo
  find:
    paths: "{{sas_config}}/{{sas_lev}}/{{ir_studio_home}}/work/"
    patterns: "*.db"
  register: ir_studio_conf

- name: Update IR Studio
  become: yes
  become_user: sas
  become_method: sudo
  replace:
    path: "{{ item.path }}"
    regexp: "Not\ Running"
    replace: "Running"
    backup: no
    with_items: "{{ ir_studio_conf.files }}"

- name: Update Web Config Files with Public DNS
  replace:
    path: "{{ item.file }}"
    regexp: "{{ item.rgx }}"
    replace: "{{ ec2_ip_address }}"
    backup: yes
    with_items: "{{ midtier_sed }}"
```
Ansible Variables

- Supports INI, Json, Dictionary, YAML structure
- Can be set nearly anywhere in your project.
- Beware of precedence.
Ansible
Variable Precedence

• Be aware of variable precedence

• Example:
  • Ansible Role Default
    • allow_x_cmd: yes
  • Ansible Playbook Variable
    • allow_x_cmd: no

• Which one is set?
/*
 * Note: This file is managed by Ansible. Do not edit directly.
 * Doing so will mean your going to have a bad time.
 *
* /

{% if '/opt/saswork' in ansible_mounts %}
  - work /opt/saswork
  - utilloc /opt/saswork
{% else %}
  - work /home/!username
  - utilloc /home/!username
{% endif %}

-MEMSIZE {{ ansible_memory_mb.real.total/4|round|int }}G
-SORTSIZE {{ ansible_memory_mb.real.total/8|round|int }}G
-BUFSIZE 64K
-IBUFSIZE 64K
-IBUFNO 32767
-BUFNO 10
-UBUFNO 10
-UBUFNO 10
-ALIGNSASIOFILES

---

- name: Create SAS Usermods Config
  template:
    src: sas_v9_usermodstmpl
    src: "{{ sas_config_home }}/{{ item }}/sasv9_usermods.cfg"
    with_items:
    - Lev1/SASApp
    - Lev1/SASMeta
## Ansible Roles

<table>
<thead>
<tr>
<th>Folder</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks</td>
<td>Playbooks which execute specific tasks</td>
</tr>
<tr>
<td>Handlers</td>
<td>Handlers are special tasks that can execute based on conditions. Examples might include restarting a service after a specific file has changed.</td>
</tr>
<tr>
<td>Files</td>
<td>Static files which are to be transported to your target host.</td>
</tr>
<tr>
<td>Templates</td>
<td>Jinja2 templates which Ansible will parse into files before execution</td>
</tr>
<tr>
<td>Vars</td>
<td>Variables</td>
</tr>
<tr>
<td>Defaults</td>
<td>Default variables</td>
</tr>
<tr>
<td>Meta</td>
<td>Metadata about the role for ansible-galaxy to identify the role</td>
</tr>
</tbody>
</table>
Ansible
Structuring Your Project

- `ansible-playbook -i environments/production site.yml -l compute_servers`
Mapping SAS Configuration Tasks to Ansible

### Plays

<table>
<thead>
<tr>
<th>Installation</th>
<th>Configuration</th>
<th>BAU</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata_servers.yml</td>
<td>os.yml</td>
<td>di_provision.yml</td>
</tr>
<tr>
<td>midtier_servers.yml</td>
<td>config.yml</td>
<td>hotfix.yml</td>
</tr>
<tr>
<td>compute_servers.yml</td>
<td>sasconfig.yml</td>
<td>promotion.yml</td>
</tr>
<tr>
<td>hadoop_servers.yml</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Structure into small includable plays.
- Use tags
- Handle idempotence in shell script Plays
Mapping SAS Configuration Tasks to Ansible Playbooks

• Structure Playbooks as an amalgam of Plays and Roles.

• site.yml should be constructed of only includes.
Mapping SAS Configuration Tasks to Ansible

Roles

```yaml
- hosts: all
  become: yes
  become_method: sudo

vars:
  selerity_customer: dummy

vars_files:
- "group_vars/common.yml"
- "group_vars/customer_{{ selerity_customer }}.yml"

roles:
- role: selerity.os.roles
- role: selerity.os.packages
- role: selerity.os.filesystem
- role: selerity.os.kernel
- role: selerity.os.pam
- role: selerity.os.tz
```
Governance and Maintenance
Presenter
Michael Dixon, Managing Director, Selerity

Michael's love for all things technical – especially in taking things apart, seeing how they work, and putting them back together – makes him a natural problem solver. His client-centric and solutions-based approach has enabled him to work seamlessly with businesses across multiple segments - helping them exceed their goals in the process.

@SelerityMichael
Cameron’s love for Computer Science began as a child programming games on his Vic 20. He started using SAS in the mid 90’s. Over a 20 year career, Cameron has become a leading consultant on analytic and data architecture, strategy, governance and operations and is a trusted advisor to many leading companies in the Australia – Pacific region.
Your feedback counts!

Don't forget to complete the session survey in your conference mobile app.

1. Go to the Agenda icon in the conference app.
2. Find this session title and select it.
3. On the sessions page, scroll down to Surveys and select the name of the survey.
4. Complete the survey and click Finish.
SAS GLOBAL FORUM 2018

April 8 - 11 | Denver, CO
Colorado Convention Center

#SASGF