

Deploying OpenAFS Cells using Ansible

Cheyenne Wills
OpenAFS 2019 Workshop



Overview

- **What is Ansible**
- **Using Ansible to deploy OpenAFS**

What is Ansible

- **Open source tool**
 - **Software provisioning**
 - **Configuration Management**
 - **Application deployment**

What is Ansible - cont

- **Light weight**
 - Does not require “agents” on target hosts (systems)
 - Only ssh and python are required on target hosts
 - Controlled by a collection of YAML configuration files

What is Ansible - cont

Example Inventory:

- 6 hosts: afs01-afs06
- 4 groups:
 - afs_kdcs
 - afs_databases
 - afs_fileservers
 - afs_clients

```
>>> cat hosts
[afs_kdcs]
afs01.example.com

[afs_databases]
afs01.example.com
afs02.example.com
afs03.example.com

[afs_fileservers]
afs04.example.com
afs05.example.com

[afs_clients]
afs[01:06].example.com
```



What is Ansible - cont

- **Tasks**
 - An operation that is performed on the target hosts
 - Each task uses an Ansible modules
 - Ansible modules are idempotent
 - i.e. the tasks can be “restarted”

```
- name: Create cell-wide configuration files
  run_once: true
  local_action:
    module: template
    src: "{{ item }}.j2"
    dest: "{{ inventory_dir }}/files/{{ item }}"
  with_items:
    - CellServDB
    - ThisCell
    - UserList

- name: Create the Kerberos service key
  command: >
    {{ afs_kadmin }}
    -w {{ afs_admin_password }}
    -p {{ afs_admin_principal }}@{{ afs_realm }}
    -r "{{ afs_realm }}"
    -q "add_principal -randkey afs/{{ afs_cell }}@{{ afs_realm }}"
  register: kadmin_results
  changed_when: >
    kadmin_results.rc == 0
    and not "already exists while creating" in kadmin_results.stderr
  when: inventory_hostname == ansible_play_hosts[0]

- include_tasks: "system-{{ afs_firewall }}.yaml"
  when: afs_firewall is defined

- name: Setup SELinux mode
  selinux:
    state: "{{ afs_selinux_mode }}"
    policy: targeted
  when: afs_selinux and afs_selinux_mode in ['disabled','enforcing','permissive']

- name: Setup SELinux policies
  include_tasks: selinux-policies.yaml
  when: afs_selinux
```

What is Ansible - cont

- **Roles**

- **A standardized directory layout with a set of tasks and other configuration files**
- **Represents a discrete “unit” e.g. a single service**
- **Promotes reuse and modularity**



What is Ansible - cont

Role for Kerberos server

```
>>> tree openafs_krbserver
openafs_krbserver
├── defaults
│   └── main.yaml
├── files
│   └── debian
│       └── policy-rc.d
├── handlers
│   └── main.yaml
├── tasks
│   ├── install-apt.yaml
│   ├── install-yum.yaml
│   ├── main.yaml
│   └── system-firewalld.yaml
├── templates
│   ├── kadm5.acl.j2
│   ├── kdc.conf.j2
│   └── krb5.conf.j2
└── vars
    ├── debian.yaml
    └── redhat.yaml

7 directories, 12 files
>>> □
```




What is Ansible - cont

- **Playbooks**
 - Declares the configurations and lists the hosts, roles and other tasks that need to be performed

```
>>> cat servers.yaml
---
- name: Install OpenAFS servers
  hosts:
    - afs_databases
    - afs_fileservers
  become: yes
  any_errors_fatal: true
  roles:
    - openafs_server
>>> □
```



Using Ansible to deploy OpenAFS

- **openafs-contrib/ansible-openafs**
 - Github project
 - Collection of ansible roles to setup and deploy an OpenAFS environment

Using Ansible to deploy OpenAFS

Roles:

- `openafs_krbserver`
- `openafs_krbclient`
- `openafs_server`
- `openafs_client`
- `openafs_cell`
- `openafs_devel`
- `openafs_robotest`



Using Ansible to deploy OpenAFS

- **Sample Playbooks:**
 - Realm - Kerberos setup
 - Servers - Establishes OpenAFS servers
 - Clients - Setup clients
 - Cell - Configures a cell



Using Ansible to deploy OpenAFS

Example inventory directory structure

```
>>> tree example.com/  
example.com/  
├── files  
├── group_vars  
│   ├── afs_fileservers.yaml  
│   └── all  
│       ├── cell.yaml  
│       └── vault  
└── hosts  
  
4 directories, 3 files  
>>> █
```



Using Ansible to deploy OpenAFS

Example inventory variables to define the cell

```
>>> cat cell.yaml
---
afs_cell: example.com
afs_realm: EXAMPLE.COM

afs_volumes:
  - name: test

afs_users:
  - name: user1
  - name: user2
  - name: user3
  - name: user4

afs_groups:
  - name: group1
    members:
      - user1
      - user2
>>>
```



Using Ansible to deploy OpenAFS

Example inventory variables to define the partitions

```
>>> cat afs_fileservers.yaml
---
afs_partitions:
  - a
  - b
  - c
>>> □
```



Using Ansible to deploy OpenAFS

<< Time compress 12 minutes into 30 secs >>



Using Ansible to deploy OpenAFS

It's Alive!!

OpenAFS
installed,
configured and
running

```
[cwills@afs06 ~]$ fs checkserver
All servers are running.
[cwills@afs06 ~]$ vos listvol -server afs04
Total number of volumes on server afs04 partition /vicepa: 5
root.afs                536870912 RW          4 K On-line
root.afs.readonly       536870913 RO          4 K On-line
root.cell                536870915 RW          3 K On-line
root.cell.readonly      536870916 RO          3 K On-line
test                    536870918 RW          2 K On-line

Total volumes onLine 5 ; Total volumes offLine 0 ; Total busy 0

Total number of volumes on server afs04 partition /vicepb: 0

Total volumes onLine 0 ; Total volumes offLine 0 ; Total busy 0

Total number of volumes on server afs04 partition /vicepc: 0

Total volumes onLine 0 ; Total volumes offLine 0 ; Total busy 0

[cwills@afs06 ~]$ █
```