Ancillary Ansible Playbooks

Andrew Mallett
Linux Author and Trainer

@theurbanpenguin www.theurbanpenguin.com

Overview

Ancillary Tasks

- Archive Files
- Importing tasks
  - dynamic (include)
  - static (import)
- Scheduling cron jobs
- Managing VDO Storage
- Including Playbooks

```
- name 'Immediate Compressed archive of /etc'
  archive:
    path: '/etc/'
    dest: "/tmp/etc-{{ ansible_hostname }}.tgz"
```

Archiving Files

We can create compressed tar archives using Ansible and the archive module. The backup is local unless the destination path specified is a network mount. Note that this is just a task we can store this as its own YAML file but it cannot be executed independently

```
- name: 'Manage Server Backup'
  hosts: all
  become: true
  gather_facts: true
  tasks:
  - include_tasks: backup.yaml
```

Code Re-Use

Creating independent tasks allow for us to import them into Plays that need them. Although created below tasks: they do not form part of the tasks: dictionary as the list is aligned with the play. Using include_tasks we can process variables generated from the Playbook, using import_tasks the assignment is static and processed before the rest of the Play

Demo

Creating Backups

- backup.yaml
- include task in Play

```
- name: 'schedule weekly backup'
  ansible.builtin.cron:
    name: 'backup /etc'
    weekday: '5'
    minute: '0'
    hour: '2'
    user: root
    job: "tar -czf /tmp/etc-{{ ansible_hostname }}.tgz /etc"
    cron_file: etc_backup
```

Scheduling Using Cron

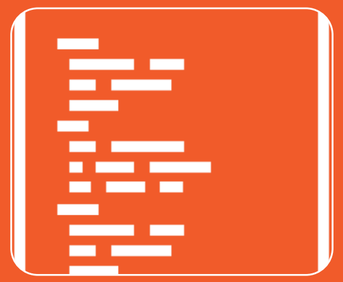
To schedule a similar backup we can use the cron system in Linux. Again, we can create and independent task to include where it is needed.

Demo

Scheduling Backups

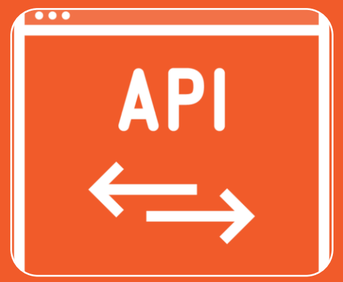
- schedule.yaml
- include task in Play

VDO is not a Filesystem



Filesystem

- EXT4
- XFS



VDO

- Logical abstraction layer

Physical Storage

- Disk
- Partition

We will need to add a new virtual disk of at least 5GB to the RHEL and CentOS Systems

```
- name: install vdo
  package:
    name:
      - vdo
      - kmod-kvdo
    state: latest
```

Installing VDO

We can create a YAML file to install the required packages

```
- name: start vdo
  service:
    name: vdo
    state: started
    enabled: True
```

Starting the Service

VDO has a service that is required to be running and enabled

```
- name: create vdo1
  vdo:
    name: vdo1
    state: present
    device: /dev/sdb
    logicalsize: 20G
```

Creating the VDO Device

We can create the VDO device using the vdo module

```
- name: format
  filesystem:
    type: xfs
    dev: /dev/mapper/vdo1
```

Creating the Filesystem

The next step is to create the filesystem

```
- name: mountpoint
  file:
    path: /vdo1
    state: directory
```

Creating the Mount Point

To access the filesystem, we need a mount point directory

```
- name: mount
  mount:
    path: /vdo1
    fstype: xfs
    state: mounted
    src: /dev/mapper/vdo1
    opts: defaults,x-systemd.requires=vdo.service
```

Mounting the Filesystem

The final step is to mount the filesystem to the mont point. this also creates and entry in the /etc/fstab file

Demo

Managing VDO

- Create tasks
- Create Playbook

Demo

Importing Playbooks

Summary

Ancillary Tasks

- archive
- cron
- vdo
  - ensure kernel is updated and system booted
  - add 8 GB drive to RHEL and CentOS
- Modular Code
  - importing and including tasks
  - importing Playbooks

Congratulations