Ansible for MQ on Linux

April 2021
Fouad Afkir

Ansible For MQ On Linux

- Ansible Overview
- MQ Deployment
- MQ Configuration
- MQ Release

What is Ansible ?

- Automation Engine
- Human readable automation
- Tasks executed in sequence
- Get productive quickly

Used to ?

- Server Configuration
- Application Deployment
- Provisioning
- Continuous Delivery
- Test automation
- Orchestration …
- …

Agentless

- No agent to exploit or update
- Uses openSSH
- Only Python must be installed on the target hosts

Modules

- Modules controls system resources, packages, files or anything else
- Over 1000 ships with Ansible
- Can be written in any language that can return JSON

Modes

- Ad-hoc
- Playbook
- Ansible Tower

Ad-hoc

- Command line
- ex 1: ansible ‘host’ -m shell ‘dspmqver’
- ex 2: ansible ‘host’ -m ping

Playbooks (1)

Azanzis Tech
DIGITAL PURE PLAYER

YAML Code

```
#Determine the MQ Version
-name
 hosts: all
 tasks:
 -name: dspmqver command
  shell: dspmqver
  register: version
 -debug: var=version.stdout.lines
```

ansible-playbook -l ’host’ dspmqver.yml

Options

-l => further limits the selected host/group patterns

-C => To check (dry run)

-v or -vv or -vvv => verbose mode (to get more details)

Playbooks(2)

- Plain text YAML files that describe ***the desired state*** of something (a kind of cooking recipe)
- Human and machine readable
- Can be used to build entire application environments.

=> Examples on: https://galaxy.ansible.com

Variables

There are many ways to source variables

- Command line (--extra-vars)
- Playbooks
- Files
- Inventories(group vars, host vars)

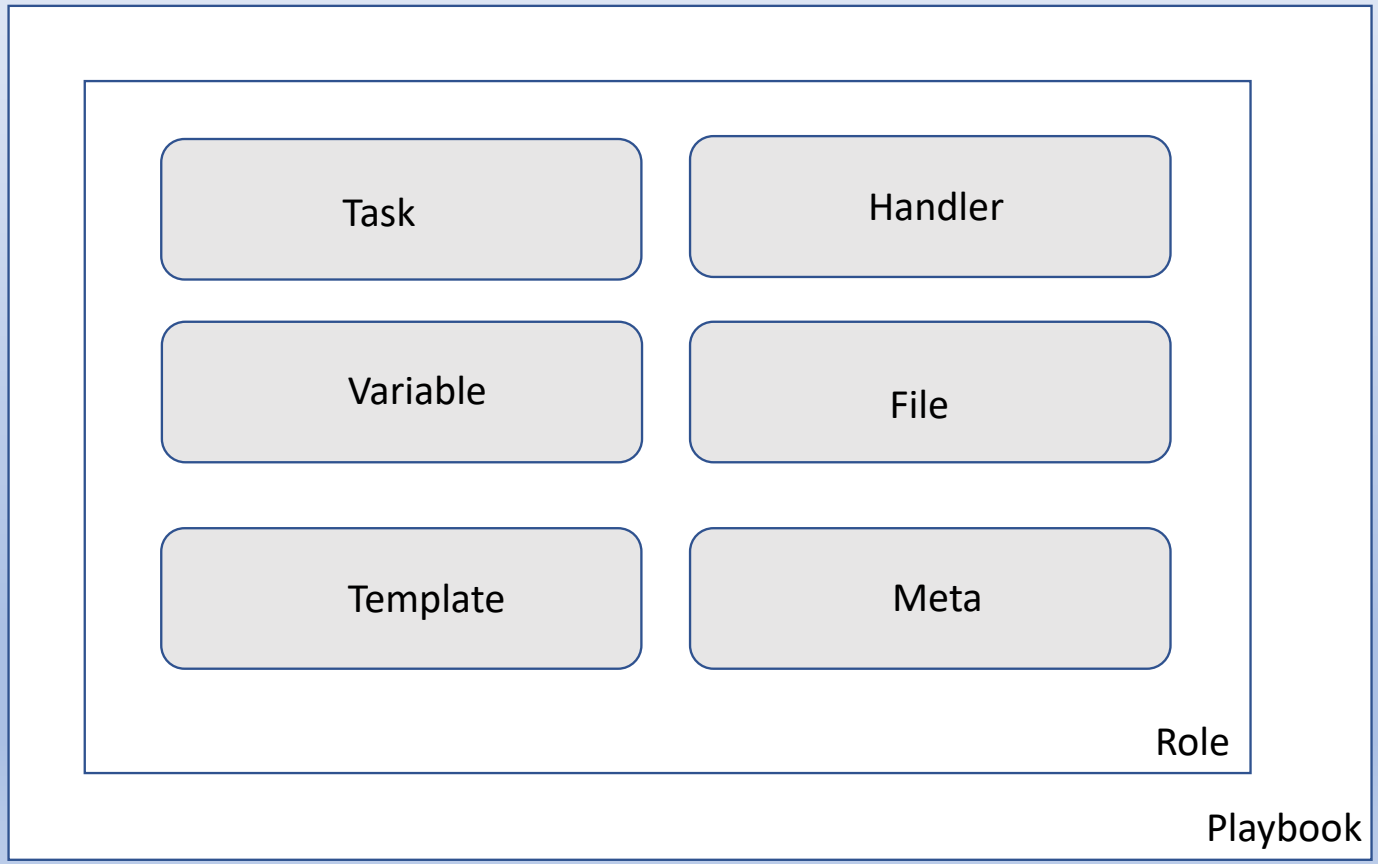
Roles

- Ansible Roles are specific kind of playbooks that are fully self contained with tasks, variables, configuration templates and other files

- ansible-galaxy init <<Role Name>>

Playbook-Role View

Azanzis Tech
DIGITAL PURE PLAYER



MQ Deployment

- New project
- New installation (or Re-installation)
- Only Binaries
- Client or Server (differentiate by variables)

MQ Deployment - Variables


```
---
# By Azanzis Tech
# Defaults file variables for Binaries IBM MQ 9.0 with last fix
mq_user: "mqm"
mq_group: "mqm"
mq_file_systems:
  - "/var/mqm"
  - "/var/mqm/trace"
  - "/var/mqm/errors"
mq_working_directory: "/tmp/ansible/mq"
mq_installation_directory: "/opt/mqm"
mq_binary_file_repository: "ansible-repository/mq"
mq_binary_file: "IBM_MQ_9.0.0.0_LINUX_X86-64.tar.gz"
mq_client_only: false
mq_version: "9.0.0-0"
mq_fix_pack_file_location: "{{ mq_binary_file_repository }}"
mq_fix_pack_file_name: "9.0.0-IBM-MQ-LinuxX64-FP0011.tar.gz"
mq_fix_pack_id: "U90010-9.0.0-11"
```

MQ Deployment - Tasks

- Prequistes/Checks/Controls
- Base Line 9.0.0.0 Install
- Fix Pack 9.0.0.x Install
- Interim Fix Install

Prequisites/Checks/Controls

- mqm user/group
- file systems(/var/mqm, /var/mqm/errors, /var/mqm/trace)
- standalone/active/inactive
- check if MQ already installed + version

Baseline 9.0.0.0 Install (1)

Azanzis Tech
DIGITAL PURE PLAYER

```
- name: unarchive the MQ file
  unarchive:
    src: "{{ mq_working_directory }}/{{ mq_binary_file }}"
    dest: "{{ mq_working_directory }}"
    remote_src: yes
  become_user: root
  become: yes

- name: accept the MQ license
  shell: >
    {{ mq_working_directory }}/MQServer/mqlicense.sh -accept -text_only
  become_user: root
  become: yes
```

Baseline 9.0.0.0 Install (2)


```
- name: execute the installation for packages in list mq_packages_list
  yum:
    name: "{{ mq_working_directory }}/MQServer/MQSeriesRuntime-{{ mq_version }}.x86_64.rpm, {{ mq_working_directory }}/MQServer/MQSeries
    state: installed
  become_user: root
  become: yes
  when: mq_client_only|bool == true

- name: execute the installation for packages in list mq_packages_list
  yum:
    name: "{{ mq_working_directory }}/MQServer/MQSeriesRuntime-{{ mq_version }}.x86_64.rpm,{{ mq_working_directory }}/MQServer/MQSeriesS
    state: installed
  become_user: root
  become: yes
  when: mq_client_only|bool == false
```

Baseline 9.0.0.0 Install (3)


```
- name: set the installation
  shell: "{{ mq_installation_directory }}/bin/setmqinst -i -p {{ mq_installation_directory }}"
  become_user: root
  become: yes

- name: inject standard mqs.ini for server installation
  copy:
    src: "mqs.ini"
    dest: "/var/mqm/mqs.ini"
    owner: "{{ mq_user }}"
    group: "{{mq_group }}"
    mode: "0664"
  become_user: "{{ mq_user }}"
  become: yes
  when:  mq_client_only|bool == false
```

MQ Deployment - Execution

$ ansible-playbook -l ‘host-mq’ mq-9.0-install-linux.yml

```
---
- name: Installation of IBM MQ 9.0.0 - latest fixpack will then be applied
  hosts: all

#
  roles:
    - mq-9.0-install-linux
```

MQ Configuration

- QM.INI setup
- Objects creation/import
- Security setup
- Customization (CCDT,...)

MQ Configuration – Variables (1)

Azanzis Tech
DIGITAL PURE PLAYER

```
---
# By Azanzis Tech
# Defaults variable file for configuration of queueu Manager 9.0.0.x
vars:
mq_working_directory: "/SRV/CIS/mq"
mq_installation_directory: "/opt/mqm"
mq_version: "9.0.0-0"
mq_startup_mqserver_script: "mqserver.sh"
mq_startup_working_directory: "/etc/init.d"
mq_queue_manager: " "
mq_port_number: " "
mq_logprimary_files: "LogPrimaryFiles=10"
mq_logsecondary_files: "LogSecondaryFiles=2"
mq_logfilepages: "LogFilePages=16384"
mq_logbufferpages: "LogBufferPages=1024"
mq_user: "mqm"
mq_group: "mqm"
mq_qm_name: "{{ mq_queue_manager }}"
```

MQ Configuration – Variables (2)

Azanzis Tech
DIGITAL PURE PLAYER

```
mq_keystore_name: "key"
mq_cert_ca_root: ""
mq_cert_ca_issuing_: ""
mq_cert_refresh_command : "MQ_CERT_REFRESH_COMMAND.mqsc"
mq_cert_command: "MQ_CERT_COMMAND.mqsc"
mq_cert_keysize: "4096"
mq_cert_sig_alg: "SHA256WithRSA"
mq_cert_o: ""
mq_cert_c: ""
mq_cert_dn: ""
mq_ccdt_directory: "/var/mqm/qmgrs/{{ mq_queue_manager }}/@ipcc"
mq_ccdt_file: "AMQCLCHL.TAB"
mq_cipher: "TLS_RSA_WITH_AES_256_CBC_SHA256"
mq_ccdt_path_prod: "/srv/nfs/tabfilerepo/CCDT/"
mq_ccdt_path_nprod: "/srv/nfs/tabfilerepo/CCDT/"
mq_cert_dn: " "
```

MQ Configuration – Tasks

- Prequistes/Checks/Controls
- Qm.ini: add & replacement
- Objects creation
- OAM Setup
- TLS Setup
- CCDT
- Automatic Startup (SystemD)
- …

qm.ini + objects setup (1)


```
- name: make sure the right ownership is set for log file systems
  file:
    path: "/var/mqm/log/{{ mq_queue_manager }}"
    owner: "{{ mq_user }}"
    group: "{{ mq_group }}"
  become_user: root
  become: yes

- name: make sure the right ownership is set for qmgrs file systems
  file:
    path: "/var/mqm/qmgrs/{{ mq_queue_manager }}"
    owner: "{{ mq_user }}"
    group: "{{ mq_group }}"
  become_user: root
  become: yes

- name: create the queue managers
  shell: "crtmqm -d SYSTEM.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE {{ mq_queue_manager }} -ld /var/mqm/log
  become_user: mqm
  become: yes
- name: start the queue managers
  shell: "strmqm {{ mq_queue_manager }}"
  become_user: "{{ mq_user }}"
  become: yes
```

qm.ini + objects setup (2)


```
- name: copy the hardening files
    template:
      src: "CustomQmgrV9.mqsc"
      dest: "{{ mq_working_directory }}/{{ mq_queue_manager }}-CustomQmgrV9.mqsc.mqsc"
      owner: "{{ mq_user }}"
      group: "{{ mq_group }}"
      mode: "0755"
    become_user: root
    become: yes

  - name: apply hardening script for all the queue managers
    shell: "runmqsc {{ mq_queue_manager }} < {{ mq_working_directory }}/{{ mq_queue_manager }}-CustomQmgrV9.mqsc > $H
    become_user: "{{ mq_user }}"
    become: yes
```

qm.ini + objects setup (3)


```
- name: Custom log primary  - replace 1#
    replace:
      path: "/var/mqm/qmgrs/{{ mq_queue_manager }}/qm.ini"
      regexp: "LogPrimaryFiles=10"
      replace: "{{ mq_logprimary_files }}"
    become_user: "{{ mq_user }}"
    become: yes

  - name: Custom log secondary - replace 2#
    replace:
      path: "/var/mqm/qmgrs/{{ mq_queue_manager }}/qm.ini"
      regexp: "LogSecondaryFiles=2"
      replace: "{{ mq_logsecondary_files }}"
    become_user: "{{ mq_user }}"
    become: yes

  - name: Custom log FilePages  - replace 3#
    replace:
      path: "/var/mqm/qmgrs/{{ mq_queue_manager }}/qm.ini"
      regexp: "LogFilePages=16384"
      replace: "{{ mq_logfilepages }}"
    become_user: "{{ mq_user }}"
    become: yes

  - name: Custom log BufferPages - replace 4#
    replace:
      path: "/var/mqm/qmgrs/{{ mq_queue_manager }}/qm.ini"
      regexp: "LogBufferPages=512"
      replace: "{{ mq_logbufferpages }}"
    become_user: "{{ mq_user }}"
    become: yes
```

OAM Setup

Azanzis Tech
DIGITAL PURE PLAYER

```
- name: apply setmqaut script for middleware group
  shell: "{{ mq_working_directory }}/setmqaut_for_mid.sh"
  become_user: "{{ mq_user }}"
  become: yes

- name: apply amqoamd for mid
  shell: "/opt/mqm/bin/amqoamd -m {{ mq_queue_manager }} -s | grep mid"
  register: output_of_setmqaut_mid
  become_user: "{{ mq_user }}"
  become: yes

- name: show setmqaut script for midftr
  debug:
    var: output_of_setmqaut_mid.stdout_lines
```

MQ Release

- MQ Updates
- IBM Security Bulletins – CVE Score
- Fix Pack
- Interim Fix
- Client or Server (differentiate by variables)

MQ Release - Variables


```
---
# By Azanzis Tech
# Defaults variables used for Upgrade of IBM MQ 9.0.0.x
vars:
mq_fix_pack_file_location: ""
mq_fix_pack_file_name: "9.0.0-IBM-MQ-LinuxX64-FP0011.tar.gz"
mq_fix_pack_archive: "9.0.0-IBM-MQ-LinuxX64-FP0011.tar"
mq_fix_pack_id: "U90010-9.0.0-11"
mq_install_working_directory: "/tmp/mq"
mq_install_directory: "/opt/mqm"
mq_installation_directory: "/opt/mqm"
mq_space_needed_total_fix: 600
mq_space_needed_unarchive_fix: 50
mq_server: true
mq_client_only: false
mq_errors_directory: "/var/mqm/errors"
mq_trace_directory: "/var/mqm/trace"
mq_space_needed_total: 900
mq_space_needed_unarchive: 450
mq_user: "mqm"
mq_group: "mqm"
mq_file_systems:
  - "/var/mqm"
  - "/var/mqm/trace"
  - "/var/mqm/errors"
mq_working_directory: "/tmp/DA/mq"
mq_backup_directory: "/tmp/DA/mq/backup"
mq_version: "9.0.0-0"
mq_config_repository: " "
mq_install_interim_fix_mode: "install"
```

MQ Release - Tasks

- Prequisites/Checks/Controls
- Backup (dmpmqcfg)
- Stop or Unmanage (pacemaker) of QMGR(s) + all apps using MQ process
- Check if MQ process are still in use
- Deploy of FixPack or Interim Fix or both
- Start or Manage (pacemaker) of QMGR(s) + all apps using MQ process

Backup - dmpmqcfg


```
- name: Get the list of all the queue managers
  action: shell dspmq | sed 's/QMNAME(//g' | cut -d ")" -f 1
  register: mq_all_qmgrs
  when: is_standalone_lv5 or is_active


- name: do a backup of MQ definitions for all queue managers
  action: shell dmpmqcfg -m {{ item }} -a > /{{ mq_backup_directory }}/{{ item }}.mqsc
  when: is_standalone or is_active
  with_items: '[ {{ mq_all_qmgrs["stdout_lines"]  | default ( omit ) }} ]'
  become_user: mqm
  become: yes
  register: mq_dmpmqcfg_output
```

Step STOP

```
#Stop Queue Manager
# for a server installation stop the active queue managers
- name: Show the list of queue manager(s) with their status
  action: debug var=mq_qmgrs_running
  when: "mq_server == true"
- name: stop the queue managers with endmqm -i
  action: shell endmqm -i {{ item }}
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  register: mq_endmqm_output
  become_user: mqm
  become: yes
  when: mq_instance
- name: Show the output of the stop qmgr command(s)
  action: debug var=mq_endmqm_output


#stop cluster mq
- name: disable cluster resource for MQ
  shell : "pcs resource disable QMGR-Resource{{ item }}"
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  when: mq_is_active

#stop cluster mq
- name: unmanage cluster resource for MQ
  shell : "pcs resource unmanage QMGR-Resource{{ item }}"
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  when: mq_is_active
```

Azanzis Tech
DIGITAL PURE PLAYER

Check if MQ process are still in use


```
#Check if MQ binaries are still in use
  - name: execute lsof on mqm
    shell: "lsof | grep {{ mq_install_directory }} | wc -l"
    become_user: root
    become: yes
    register: result
    retries: 10
    delay: 10
    until: result.stdout | int == 0
  - name: show value
    debug:
      var: "{{ result.stdout | int }}"
  - name: test condition
    fail:
       msg: "Can not start upgrade - MQ libraries still locked by processe(s) {{ result.stdout | int }}"
    when: result.stdout | int > 0
```

Deploy FixPack


```
- name: unarchive the fix pack binary file
    unarchive:
     src: "{{ mq_install_working_directory }}/{{ mq_fix_pack_file_name }}"
     dest: "{{ mq_install_working_directory }}"
     remote_src: yes
    become_user: root
    become: yes
    when: mq_os_type.stdout == "Linux"

  - name: apply the fix pack
    debug:
     msg: "Apply the fix pack {{ mq_fix_pack_file_name }}"

  - name: list packages to be updated
    shell: "rpm -qa | grep ^MQ | cut -d '-' -f 1 | sort -u | sed 's/$/-{{ mq_fix_pack_id }}.x86_64.rpm,/g' | se
    register: rpm_result

  - name: show the packages
    debug:
      var: mq_fix_pack_id

  - name: update the selected packages
    yum:
      name: "{{ rpm_result.stdout }}"
      state: present
    become_user: root
    become: yes
```

Step START


```
#Start Queue Manager
- name: Restart queue manager(s) that were stopped
  action: command strmqm {{ item }}
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  register: mq_strmqm_output
  become_user: mqm
  become: yes
  when: mq_instance and mq_server == true and is_standalone
- name: Show the output of the start qmgr command(s)
  action: debug var=mq_strmqm_output
  when: mq_instance and mq_server == true and is_standalone


# start cluster mq
- name: unmanage cluster resource for MQ
  shell : "pcs resource manage QMGR-Resource{ item }}"
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  when: mq_is_active

- name: enable cluster resource for MQ
  shell : "pcs resource enable QMGR-Resource{{ item }}"
  with_items: '[ {{ mq_qmgrs_running["stdout_lines"] }} ]'
  when: mq_is_active
```

Ansible For MQ On Linux

?