Automate PMM Deployment with Ansible

Max Bubenick
Platform Lead @ Percona
Who am I?

• Working @ Percona since 2013
  • Sr Remote DBA
  • Manager
  • Platform Lead
• Past Work Experience
  • Developer
  • SysAdmin
  • Data Architect / DBA
Agenda

• Pre-requisites
• Inventory
• Playbook
• Roles
  • Docker
  • Pmm-server
  • Pmm-client
  • Pmm-managed
Pre-requisites
Pre-requisites

- Ansible 2.7
- pmm-server RedHat Based (needs to work on Debian support)
- PMM communication ports open
Ansible Inventory

Predefined groups that should not need changes

```yaml
[pmm:children]
pmm-server
pmm-client
pmm-managed

[pmm-server]
server

[pmm-client:children]
mysql
mongodb

[pmm-managed:children]
rds
```
Ansible Inventory

Predefined global vars

[pmm:vars]
global_pmm_version=1.17.1
pmm_ssl_enabled=True
pmm_server_username=pmm
pmm_server_password=VAULT_ME

ALWAYS ENCRYPT PASSWORDS!!!!
Ansible Inventory

We need to define hosts and place them in the appropriate groups based on purpose

[mysql]

[proxysql]

[mongodb]

[rds]
Ansible Playbook
Ansible Playbook

pmm-server group to run roles: docker & pmm

- hosts: pmm-server
  become: true
  gather_facts: yes
  roles:
  - { role: docker, tags: docker }
  - { role: pmm, tags: pmm }
Ansible Playbook

pmm-managed group to run pmm_managed role for RDS Instances

- hosts: pmm-managed
  become: true
  gather_facts: no
  serial: 1
  roles:
  - { role: pmm_managed, tags: pmm-managed }
Ansible Playbook

pmm-client play to run pmm_client role against mysql, mongodb and proxysql servers.

- hosts: pmm-client
  become: true
  gather_facts: yes
  roles:
  - { role: pmm_client, tags: pmm-client }
Ansible Roles: Docker

Deploys Docker. It’s a pmm-server dependency.

Role Defaults:

# Options applied via systemd
docker_options: ""

# Storage driver for Docker
docker_storage_options: --storage-driver=devicemapper
Ansible Roles: pmm

Deploys pmm-server. Depends on Docker.

Role Defaults:

## GENERAL SETTINGS

pmm_version: "{{ global_pmm_version | default('1.17.1') }}"
pmm_install_version: "{{ pmm_version }}"

# Port for unencrypted requests
pmm_http_port: 8080
# Port for encrypted requests
pmm_https_port: 8443

# Force a reset of the data container
pmm_reset_container: no
Ansible Roles: pmm

Role Defaults:

## SECURITY SETTINGS !!! SET SECRETS IN VAULT FILES !!!

# Must be stored stored in vault
pmm_server_username: pmm
pmm_server_password: VAULT_ME

# As documented, this can be an IAM user with CloudWatchReadOnlyAccess access
pmm_aws_access_key_id: your_aws_access_key_id
pmm_aws_secret_access_key: your_aws_secret_access_key

pmm_rootless: no

pmm_ssl_enabled: True
pmm_ssl_dir: pmm-ssl
pmm_ssl_expire_days: 3650
pmm_ssl_cert_address: localhost

# Run the PMM container in rootless mode
# Enable SSL for PMM
# Relative path in /etc/ for generated certs
# Expiry time for the cert (in days)
Ansible Roles: pmm

Role Defaults:

## PROMETHEUS SETTINGS

# Data retention for Prometheus; N.B. set before the container is created/recreated
pmm_metrics_retention: 720h

# Resolution time in Prometheus; default of 5s to reduce load
pmm_metrics_resolution: 5s

# Specify memory available to Prometheus; ansible_memtotal_mb * 128 since pmm_metrics_memory is in KB
pmm_metrics_memory: "{{ ansible_memtotal_mb * 128 }}"

# Applies QUERIES_RETENTION to the Docker container to configure QAN purging
pmm_queries_retention: 8
## Ansible Roles: pmm

### Role Defaults:

```yaml
## DOCKER CONTAINER SETTINGS

# Port mapping
pmm_docker_ports:
- '{% do pmm_http_port %}:{% do 8080 if pmm_rootless else 80 %}/tcp'
- '{% do pmm_https_port %}:{% do 8443 if pmm_rootless else 443 %}/tcp'

# Restrict memory usage for Docker; divided by 2 since this value is required in MB
pmm_docker_memory: "{{ (ansible_memtotal_mb / 2) | int }}MB"

pmm_docker_env:
  SERVER_USER: "{{ pmm_server_username }}"
  SERVER_PASSWORD: "{{ pmm_server_password }}"
  METRICS_RETENTION: "{{ pmm_metrics_retention }}"
  METRICS_RESOLUTION: "{{ pmm_metrics_resolution }}"
  METRICS_MEMORY: "{{ pmm_metrics_memory }}"
  DISABLE_UPDATES: 1
  QUERIES_RETENTION: "{{ pmm_queries_retention }}"
```
Ansible Roles: pmm

Role Defaults:

## EXTENDING PMM SETTINGS

# Prometheus recording rules

# Period for rules evaluation in Prometheus
pmm_prometheus_rules_evaluation: 5m

# A list of files specifying recording rules
pmm_prometheus_rules_files: []

# Enable custom dashboards deployment
pmm_custom_dashboards_enabled: yes

# List of dashboards to be added
pmm_custom_dashboards: []
Ansible Roles: pmm-client

Deploys pmm-client

Role Defaults:

```yaml
## GENERAL SETTINGS
pmm_version: "{{ global_pmm_version | default('1.17.1') }}"
pmm_install_version: "{{ pmm_version }}"

pmm_client_enabled: yes  # enable/disable pmm-client deployment
pmm_client_bind_address: "{{ ansible_ssh_host }}"  # Set the bind address (IPv4)
pmm_client_check_for_vividcortex: false  # Handle Vividcortex agent during tasks

linux_metrics_port: 42000
mysql_queries_port: 42001
mysql_metrics_port: 42002
mongodb_metrics_port: 42003
proxysql_metrics_port: 42004
```
Ansible Roles: pmm-client

Role Defaults:

```yaml
## Prom exporters settings

pmm_client_mysqld_exporter_extra_args: []

pmm_client_node_exporter_extra_args: []

pmm_client_disable_tablestats_limit: 1000
```
Ansible Roles: pmm-client

Role Defaults:

```yaml
## USERS SETTINGS
pmm_defaults_file: /usr/local/percona/.my.cnf # Used with pmm-admin for MySQL credentials

#pmm_src_defaults_file: /path/to/.my.cnf # Source for pmm_defaults_file

# These must be placed in vault
pmm_client_username: pmm
pmm_client_passwd: Vault_m3!

pmm_mysql_username: "{{ pmm_client_username }}"
pmm_mysql_passwd: "{{ pmm_client_passwd }}"

pmm_mongodb_username: "{{ pmm_client_username }}"
pmm_mongodb_passwd: "{{ pmm_client_passwd }}"
```
Ansible Roles: pmm-managed

Uses pmm-managed API to add RDS servers

Role Defaults:

pmm.managed_enabled: yes

# Specify the name used by pmm-managed to add the host
pmm.rds_host_name: '{% ansible_host.split(".")[0] %}'

# Specify the region used by pmm-managed to add the host
pmm.rds_host_region: '{% ansible_host.split(".")[2] %}'

For more details on PMM Managed API:
Source Code

Code is in public GitHub repo

https://github.com/maxbube/ansible-pmm

$ git clone git@github.com:maxbube/ansible-pmm.git
Thank You to Our Sponsors
Rate My Session

Introducing gh-ost: triggerless, painless, trusted online schema migrations

11:20 - 12:10, Matterhorn 2

Rate & Review

Tap a star to rate

Feedback (optional)

Submit
Thank you!

Q & A

Contact me:
@maxbube
max.bubenick@percona.com