MAAS-ansible-playbook

An Ansible playbook for installing and configuring MAAS, further documentation is found here.

Versions

This playbook has been tested with Ansible version 5.10.0 and above. We recommend using the latest available stable version of Ansible (currently 7.x). The `netaddr` Python library needs to be installed on the machine on which Ansible is used; note that this is not required on remote hosts.

Install

```
git clone git@github.com:maas/maas-ansible-playbook
```

Setup

This playbook has several main roles, each has a corresponding group to assign hosts to. They are as follows:

- **maas_pacemaker**: This role and group are for a set of hosts to be configured as a Pacemaker cluster to manage HA Postgres. It is recommended that `maas_corosync` is a child group of this group. It is also recommended that these are the same hosts as the `maas_postgres` group. This role is optional, but required for HA Postgres.

- **maas_corosync**: This role and group are for a set of hosts to be configured to run a Corosync cluster, used for managing quorum of a HA Postgres cluster. This role is optional, but required for HA Postgres.

- **maas_postgres**: This role and group are for a host to be configured as a Postgres instance for the MAAS stack. If more than one host is assigned this role, Postgres will be configured in HA, which then makes the `maas_corosync` and `maas_pacemaker` roles required. This role is required.

- **maas_region_controller**: This role and group are for hosts to be configured as MAAS Region Controllers. This role is required at least once.

- **maas_rack_controller**: This role and group are for hosts to be configured as MAAS Rack Controllers. This role is required at least once.

- **maas_postgres_proxy**: This role and group are for hosts to be configured as an HAProxy instance in front of the HA Postgres cluster, in order to ensure queries are directed to the primary instance. We recommend this group is assigned to the same hosts as `maas_region_controller`, this will have HAProxy's listener bind to localhost and route queries for that specific region controller. This role is optional, but recommended for HA Postgres.

- **maas_proxy**: This role and group are for hosts to be configured as an HAProxy instance in front of region controllers for HA stacks. This role is optional.

Host file

Two formats can be used to define the inventory (`ini` and `yaml`). Below, the same inventory can be found twice: defined in both formats. Only one is to be chosen when running the playbook.

More information can be found on the Ansible inventory documentation page.

ini

```
[maas_corosync]
db01.example.com
db02.example.com
db03.example.com

[maas_pacemaker:children]
maas_corosync

[maas_postgres]
db01.example.com
db02.example.com
db03.example.com

[maas_postgres_proxy]
region01.example.com
region02.example.com
region03.example.com

[maas_region_controller]
region01.example.com
region02.example.com
region03.example.com

[maas_rack_controller]
region01.example.com
rack01.example.com
rack02.example.com

[maas_proxy]
proxy01.example.com
```

yaml

```
---
all:
  children:
    maas_pacemaker:
      children:
        maas_corosync:
          hosts:
            db01.example.com
            db02.example.com
            db03.example.com
    maas_postgres:
      hosts:
        db01.example.com:
        db02.example.com:
        db03.example.com:
    maas_proxy:
      hosts:
        proxy01.example.com:
    maas_postgres_proxy:
      hosts:
        region01.example.com:
    maas_region_controller:
      hosts:
        region01.example.com:
        region02.example.com:
        region03.example.com:
    maas_rack_controller:
      hosts:
        region01.example.com:
        rack01.example.com:
        rack02.example.com:
```

Note: the Pacemaker role requires host-specifc variables that should be defined in the hosts file, they are as follows:

- `maas_pacemaker_fencing_driver`: The Pacemaker STONITH fencing driver, defaults to `ipmilan`. This driver is used to forcefully remove a member from the Pacemaker cluster when it exhibits erroneous behavior. Pacemaker will list the available drivers with the following command: `stonith_admin --list-installed`

- `maas_pacemaker_stonith_params`: The parameters specific to the fencing driver selected. These must be defined when using Pacemaker, to see the parameters of a given driver, run `stonith_admin --metadata --agent <driver>`

Run

This playbook requires a user to set the following set of variables:

- **maas_version**: The intended version of MAAS to install.
  Example: `'3.2'`

- **maas_postgres_password**: The password for the MAAS postgres user.
  Example: `'my_password'`

- **maas_installation_type**: The intended type of MAAS installation.
  Possible values are: `'deb'` or `'snap'`

- **maas_url**: The MAAS URL where MAAS will be accessed and which rack controllers should use for registering with region controllers.
  Example: `'http://proxy01.example.com:5240/MAAS'`

There are additional optional variables that can be passed to the playbooks:

- Admin credentials (if setting up an admin user)

  - **admin_username**: The username of the admin user
    Default: `admin`

  - **admin_password**: The password of the admin user
    Default: `admin`

  - **admin_email**: The email address of the admin user
    Default: `admin@email.com`

  - **admin_id**: the Launchpad or GitHub id of the admin user to import ssh keys. (Optional)

- **maas_proxy_postgres_proxy_enabled**: Use postgres proxy uri
  Default: `false`

- **enable_tls**: Whether MAAS should enable TLS
  Default: `false` (only valid for MAAS >= 3.2)

- **enable_firewall**: Whether MAAS should configure the firewall
  Default: `true`

- MAAS Vault (only valid for MAAS >= 3.3)

  - **vault_integration**: Whether MAAS Should use Vault for secret storage
    Default: `false`
  - **vault_url**: The URL of the MAAS Vault
    Default: undefined
  - **vault_approle_id**: The approle id of the MAAS Vault
    Default: undefined
  - **vault_wrapped_token**: The wrapped token for the MAAS Vault
    Default: undefined
  - **vault_secrets_path**: The secrets path of the MAAS Vault
    Default: undefined
  - **vault_secret_mount**: The secret mount of the MAAS Vault
    Default: undefined

- **http_proxy**: The HTTP Proxy to use
  Default: proxy not used

- **https_proxy**: The HTTPS Proxy to use
  Default: proxy not used

- Observability

  - **o11y_enable**: Whether Observability features should be enabled
    Default: `false`
  - **o11y_prometheus_url**: Prometheus Remote Write Receiver endpoint URL
    Default: undefined
  - **o11y_loki_url**: Loki endpoint URL
    Default: undefined

When the Observability features are enabled, at least one of the endpoints MUST be defined.

The following variables are only required when using HA Postgres:

- **maas_postgres_floating_ip**: The floating IP used internally to the HA Postgres cluster for the purpose of replication. A static reservation for this IP should be made in your network to avoid overlap.

- **maas_postgres_floating_ip_prefix_len**: The prefix length of the subnet within your network that the floating IP is within.

Deploy the MAAS stack

```
ansible-playbook -i ./hosts\
    --extra-vars="maas_version=3.2 maas_postgres_password=example
maas_installation_type=deb maas_url=http://example.com:5240/MAAS"\
    ./site.yaml
```

Deploy the MAAS stack with Observability enabled

```
ansible-playbook -i ./hosts \
    --extra-vars="maas_version=3.3 \
        maas_postgres_password=example \
        maas_installation_type=snap \
        maas_url=http://example.com:5240/MAAS \
        o11y_enable=true \
        o11y_prometheus_url=http://prometheus-server:9090/api/v1/write \
        o11y_loki_url=http://loki-server:3100/loki/api/v1/push" \
    ./site.yaml
```

Teardown the MAAS stack

```
ansible-playbook -i ./hosts ./teardown.yaml
```

Backup the MAAS stack

Backup MAAS requires the `maas_backup_download_path` variable to be set, it will designate a path local to where the playbook is being run to download backup archives. This path must exist prior to running the playbook. `maas_installation_type` is also required.

```
ansible-playbook -i ./hosts --extra-vars="maas_backup_download_path=/tmp/maas_backups/ maas_installation_type=deb" ./backup.yaml
```

Restore from Backup

Since backup is per-host, it is recommended to set the `maas_backup_file` variable as a host variable, or filter execution to a specific host, but `maas_backup_file` should be set to a gzipped tar file from the backup action and `maas_installation_type` should be set as well.

The below example assumes `maas_backup_file` is set in `./hosts`

```
ansible-playbook -i ./hosts --extra-vars="maas_installation_type=deb" ./restore.yaml
```

Create a new admin user

For an already existing MAAS installation, a new admin user can be created. The below example assumes the host for which the account is to be created is set under `[maas_region_controller]` in `./hosts`. `user_ssh` allows for upload of a Launchpad (`lp:id`) or GitHub (`gh:id`) public key, or can be left blank (`user_ssh=`).

```
ansible-playbook -i ./hosts --extra-vars="user_name=newuser user_pwd=newpwd user_email=user@email.com user_ssh=lp:id" ./createadmin.yaml
```

Export Observability alert rules

MAAS has a curated collection of alert rules for Prometheus and Loki. You can export these rules using the following command, where `o11y_alertrules_dest` is the directory where the files should me placed.

```
ansible-playbook --extra-vars="o11y_alertrules_dest=/tmp" ./alertrules.yaml
```