Minio - Distributed Mode

Ansible role to deploy and configure Minio in distributed mode

Peter Murphy

My Company

Table of contents

1. Home

1.1 Role Name

minio_install

1.2 Description

Role to install and configure Minio in distributed mode

1.3 Dependencies

None

1.4 Information

Author	Company	License	Minimum Ansible Version
Peter Murphy	My Company	None	2.9

2. Defaults

2.1 main.yml

2.1.1 minio_server_install_volume_groups

Volume Group settings ...

```
vg_min:
  pv: /dev/sdd
```

Where referenced
tasks/volumes.yml

2.1.2 minio_server_install_volumes

```
lv_min:
  drive: vg_min
  size: +100%FREE
```

2.1.3 minio_server_install_dir_mounts

Directory for logical volume mount ...

```
/var/lib/minio:
  src: /dev/vg_min/lv_min
```

Where referenced
tasks/volumes.yml templates/minio.env.j2

2.1.4 minio_server_datadirs

Minio server data directory ...

```
/var/lib/minio
...
```

Type	Vault required	Where referenced
string	True	templates/minio.env.j2

2.1.5 minio_user

Minio user ...

```
minio
```

```
...
```

Type	Vault required	Where referenced
string	True	tasks/directories.yml tasks/minio/users.yml templates/minio.init.j2 templates/minio.service.j2

2.1.6 minio_group

Minio user group ...

```
minio
...
```

Where referenced
tasks/directories.yml tasks/minio/minio.yml tasks/minio/users.yml templates/minio.service.j2

2.1.7 minio_server_download_base_url

Base URL to download minio from ...

```
https://dl.minio.io/server/minio/release/linux-amd64
...
```

Where referenced
vars/main.yml

2.1.8 minio_server_bin

Minio server bin directory ...

```
/usr/local/bin/minio
...
```

Where referenced
tasks/minio/minio.yml templates/minio.init.j2 templates/minio.service.j2

2.1.9 minio_server_envfile

Path to the file containing the ENV variables for the Minio server ...

```
/etc/default/minio
```

```
...
```

Where referenced
tasks/minio/minio.yml
templates/minio.init.j2
templates/minio.service.j2

2.1.10 minio_port

Minio server port ...

```
'9091'
```

Where referenced
tasks/minio/firewall.yml
vars/main.yml

2.1.11 minio_server_opts

Additional Minio server CLI options ...

```
''
```

Where referenced
templates/minio.env.j2

2.1.12 minio_access_key

Minio access key ...

```
''
```

Where referenced
templates/minio.env.j2

2.1.13 minio_secret_key

Minio secret key ...

```
''
```

Where referenced
templates/minio.env.j2

2.1.14 minio_server_env_extra

Additional environment variables to be set in minio server environment ...

```
''
```

Where referenced
templates/minio.env.j2

3. Vars

3.1 main.yml

3.1.1 minio_server_download_url

URL to download minio from ...

```
'{{ minio_server_download_base_url }}/minio'
```

Where referenced
tasks/minio/minio.yml

3.1.2 minio_server_addr

Minio server listen address ...

```
:{{ minio_port }}
...
```

Where referenced
templates/minio.env.j2
templates/minio.service.j2

4. Tasks

4.1 Flow

```
graph LR
main.yml(main.yml) --> minio/users.yml(minio/users.yml)
main.yml(main.yml) --> volumes.yml(volumes.yml)
main.yml(main.yml) --> minio/minio.yml(minio/minio.yml)
main.yml(main.yml) --> minio/firewall.yml(minio/firewall.yml)
volumes.yml(volumes.yml) --> directories.yml(directories.yml)
```

4.2 Tasks: main

4.2.1 main.yml

- Set proxy server if defined
  Tags: minio
- Configure Minio user
  Tags: minio,users
- Configure logical volumes
  Tags: minio,volumes
- Download, Install and Configure Minio
  Tags: minio,install
- Configure Firewall
  Tags: minio,firewall

4.3 Tasks: users

4.3.1 minio/users.yml

- Create Minio group
  Tags: minio,users
- Create Minio user
  Tags: minio,users

4.4 Tasks: file-system

4.4.1 volumes.yml

- Configure LVM volume groups
  Tags: minio,volumes
- Configure LVM logical volumes
  Tags: minio,volumes
- Create the xfs filesystems
  Tags: minio,volumes
- Create directories
  Tags: minio,volumes
- Create list of mounted devices
  Tags: minio,volumes
- Mount and bind a volume
  Tags: minio,volumes

4.4.2 directories.yml

- Get status of the FS object
  Tags: minio,volumes,dirs
- Create directory if not already created
  Tags: minio,volumes,dirs

4.5 Tasks: minio

4.5.1 minio/minio.yml

- Get the Minio server checksum
  Tags: minio,install
- Download the Minio server
  Tags: minio,install
- Generate the Minio server envfile
  Tags: minio,install
- Create the Minio server systemd config
  Tags: minio,install
- Create the Minio server init.d config
  Tags: minio,install
- Enable and start the Minio service
  Tags: minio,install

4.5.2 minio/firewall.yml

- Enable firewalld
  Tags: minio,firewall
- Open firewall ssh port
  Tags: minio,firewall
- Open firewall minio ports
  Tags: minio,firewall

5. Templates

5.1 minio.env.j2

```
{{ ansible_managed | comment }}

# Minio local/remote volumes.
{% if groups['minio_cluster'] | length > 0 %}
# MINIO_VOLUMES="{{ groups['minio_cluster'] | join(' ') }}"
MINIO_VOLUMES="{% for host in groups['minio_cluster'] %}http://{{ host }}{{ minio_server_datadirs }}{% if not loop.last %} {% endif %}{% endfor %}"
{% else %}
MINIO_VOLUMES="{{ minio_server_install_dir_mounts | join(' ') }}"
{% endif %}
# Minio cli options.
MINIO_OPTS="--address {{ minio_server_addr }} {{ minio_server_opts }}"

{% if minio_access_key %}
# Access Key of the server.
MINIO_ACCESS_KEY="{{ minio_access_key }}"
{% endif %}
{% if minio_secret_key %}
# Secret key of the server.
MINIO_SECRET_KEY="{{ minio_secret_key }}"
{% endif %}

{{ minio_server_env_extra }}
```

5.2 minio.init.j2

```
#! /bin/sh
### BEGIN INIT INFO
# Provides:          minio
# Required-Start:    $syslog $network
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Distributed object storage server built for cloud applications and devops.
# Description:       Distributed object storage server built for cloud applications and devops.
### END INIT INFO

{{ ansible_managed | comment }}

# Do NOT "set -e"

# PATH should only include /usr/* if it runs after the mountnfs.sh script
PATH=/sbin:/usr/sbin:/bin:/usr/bin
NAME=minio
SERVICEVERBOSE=yes
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME
WORKINGDIR=/usr/local/
DAEMON="{{ minio_server_bin }}"
USER="{{ minio_user }}"

# Read configuration variable file if it is present
[ -r "{{ minio_server_envfile }}" ] && . {{ minio_server_envfile }}

# Make sure the MINIO_VOLUMES variable is defined
[ -n "${MINIO_VOLUMES}" ] || log_daemon_msg "Variable MINIO_VOLUMES not set in {{ minio_server_envfile }}"

# Set the DAEMON_ARGS variable
DAEMON_ARGS="server $MINIO_OPTS $MINIO_VOLUMES"

# Specifies the maximum file descriptor number that can be opened by this process
ulimit -n 65536

# Exit if the package is not installed
[ -x "$DAEMON" ] || exit 0

# Load the VERBOSE setting and other rcS variables
. /lib/init/vars.sh

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.2-14) to ensure that this file is present
# and status_of_proc is working.
. /lib/lsb/init-functions

#
# Function that starts the daemon/service
#
do_start()
{
    # Return
    #   0 if daemon has been started
    #   1 if daemon was already running
    #   2 if daemon could not be started
    sh -c "USER=$USER start-stop-daemon --start --quiet --pidfile $PIDFILE --make-pidfile \\
            --test --chdir $WORKINGDIR --chuid $USER \\
            --exec $DAEMON -- $DAEMON_ARGS > /dev/null \\
            || return 1"
    sh -c "USER=$USER start-stop-daemon --start --quiet --pidfile $PIDFILE --make-pidfile \\
            --background --chdir $WORKINGDIR --chuid $USER \\
            --exec $DAEMON -- $DAEMON_ARGS \\
            || return 2"
}

#
# Function that stops the daemon/service
#
do_stop()
{
    # Return
    #   0 if daemon has been stopped
    #   1 if daemon was already stopped
    #   2 if daemon could not be stopped
    #   other if a failure occurred
    start-stop-daemon --stop --quiet --retry=TERM/1/KILL/5 --pidfile $PIDFILE --name $NAME
    RETVAL="$?"
    [ "$RETVAL" = 2 ] && return 2
    start-stop-daemon --stop --quiet --oknodo --retry=0/1/KILL/5 --exec $DAEMON
    [ "$?" = 2 ] && return 2
    # Many daemons don't delete their pidfiles when they exit.
    rm -f $PIDFILE
    return "$RETVAL"
}


case "$1" in
  start)
    [ "$SERVICEVERBOSE" != no ] && log_daemon_msg "Starting" "$NAME"
    do_start
    case "$?" in
        0|1) [ "$SERVICEVERBOSE" != no ] && log_end_msg 0 ;;
        2) [ "$SERVICEVERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  stop)
    [ "$SERVICEVERBOSE" != no ] && log_daemon_msg "Stopping" "$NAME"
    do_stop
    case "$?" in
        0|1) [ "$SERVICEVERBOSE" != no ] && log_end_msg 0 ;;
        2) [ "$SERVICEVERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  status)
    status_of_proc "$DAEMON" "$NAME" && exit 0 || exit $?
    ;;
  restart|force-reload)
    log_daemon_msg "Restarting" "$NAME"
    do_stop
    case "$?" in
      0|1)
        do_start
        case "$?" in
            0) log_end_msg 0 ;;
            1) log_end_msg 1 ;; # Old process is still running
            *) log_end_msg 1 ;; # Failed to start
        esac
        ;;
      *)
        # Failed to stop
        log_end_msg 1
        ;;
        esac
    ;;
  *)
        echo "Usage: $SCRIPTNAME {start|stop|status|restart|force-reload}" >&2
        exit 3
        ;;
esac

:
```

5.3 minio.service.j2

```
{{ ansible_managed | comment }}

[Unit]
Description=Minio
Documentation=https://docs.minio.io
Wants=network-online.target
After=network-online.target
AssertFileIsExecutable={{ minio_server_bin }}

# Avoid noisy crashloops
StartLimitIntervalSec=60
StartLimitBurst=5

[Service]
WorkingDirectory=/usr/local/

User={{ minio_user }}
Group={{ minio_group }}

PermissionsStartOnly=true

EnvironmentFile={{ minio_server_envfile }}
ExecStartPre=/bin/bash -c "[ -n \"${MINIO_VOLUMES}\" ] || echo \"Variable MINIO_VOLUMES not set in {{ minio_server_envfile }}\""

ExecStart={{ minio_server_bin }} server $MINIO_OPTS $MINIO_VOLUMES

# Let systemd always restart this service, in limits defined by StartLimitIntervalSec and StartLimitBurst.
Restart=always

StandardOutput=journal
StandardError=inherit

# Specifies the maximum file descriptor number that can be opened by this process
LimitNOFILE=65536

# Disable timeout logic and wait until process is stopped
TimeoutStopSec=0

# SIGTERM signal is used to stop Minio
KillSignal=SIGTERM

SendSIGKILL=no

SuccessExitStatus=0

{% if (minio_server_addr.split(':')[-1] | int) < 1024 %}
AmbientCapabilities=CAP_NET_BIND_SERVICE
{% endif %}

[Install]
WantedBy=multi-user.target
```

6. Handlers

6.1 main.yml

- Reload minio systemd
- Restart minio
- Reload firewalld service

7. Appendix

7.1 References

Description	Link
Distributed Minio QuickStart Guide	https://docs.min.io/docs/distributed-minio-quickstart-guide.html
Peter Murphy	https://www.linkedin.com/in/peter-murphy-61493974/
ansible-mdgen	https://pypi.org/project/ansible-mdgen/