OpenStack-Ansible Documentation: os_aodh role

Release 18.1.0.dev195

OpenStack-Ansible Contributors

Mar 28, 2024

CONTENTS

1 Configuring the Aodh service (optional) **1**

1.1 Configuring the hosts 1

2 Default variables **3**

3 Example playbook **9**

4 Dependencies **11**

5 Tags **13**

CHAPTER

ONE

CONFIGURING THE AODH SERVICE (OPTIONAL)

The Telemetry alarming services perform the following functions:

- Creates an API endpoint for controlling alarms.
- Allows you to set alarms based on threshold evaluation for a collection of samples.

1.1 Configuring the hosts

Configure Aodh by specifying the `metering-alarm_hosts` directive in the `/etc/openstack_deploy/conf.d/aodh.yml` file. The following shows the example included in the `etc/openstack_deploy/conf.d/aodh.yml.example` file:

```
# The infra nodes that the Aodh services run on.
metering-alarm_hosts:
  infra1:
    ip: 172.20.236.111
  infra2:
    ip: 172.20.236.112
  infra3:
    ip: 172.20.236.113
```

The `metering-alarm_hosts` provides several services:

- An API server (`aodh-api`): Runs on one or more central management servers to provide access to the alarm information in the data store.
- An alarm evaluator (`aodh-evaluator`): Runs on one or more central management servers to determine alarm fire due to the associated statistic trend crossing a threshold over a sliding time window.
- A notification listener (`aodh-listener`): Runs on a central management server and fire alarms based on defined rules against event captured by ceilometers modules notification agents.
- An alarm notifier (`aodh-notifier`). Runs on one or more central management servers to allow the setting of alarms to base on the threshold evaluation for a collection of samples.

These services communicate by using the OpenStack messaging bus. Only the API server has access to the data store.

To install aodh on an existing OpenStack-Ansible environment, run the `os-aodh-install.yml` playbook.

To clone or view the source code for this repository, visit the role repository for os_aodh.

CHAPTER

TWO

DEFAULT VARIABLES

```
## Verbosity Options
debug: False

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
aodh_service_setup_host: "{{ openstack_service_setup_host | default('localhost
↪') }}"
aodh_service_setup_host_python_interpreter: "{{ openstack_service_setup_host_
↪python_interpreter | default((aodh_service_setup_host == 'localhost') |␣
↪ternary(ansible_playbook_python, ansible_facts['python']['executable'])) }}"

# Set the package install state for distribution packages
# Options are 'present' and 'latest'
aodh_package_state: "{{ package_state | default('latest') }}"

# Set installation method.
aodh_install_method: "{{ service_install_method | default('source') }}"
aodh_venv_python_executable: "{{ openstack_venv_python_executable | default(
↪'python3') }}"

## The git source/branch
aodh_git_repo: https://opendev.org/openstack/aodh
aodh_git_install_branch: master
aodh_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
aodh_git_constraints:
  - "--constraint {{ aodh_upper_constraints_url }}"

aodh_pip_install_args: "{{ pip_install_options | default('') }}"

# Name of the virtual env to deploy into
aodh_venv_tag: "{{ venv_tag | default('untagged') }}"
aodh_bin: "{{ _aodh_bin }}"

## System info
aodh_system_user_name: aodh
```

(continues on next page)

(continued from previous page)

```
aodh_system_group_name: aodh
aodh_system_shell: /bin/false
aodh_system_comment: aodh system user
aodh_system_user_home: "/var/lib/{{ aodh_system_user_name }}"

## Database info
aodh_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
aodh_db_setup_python_interpreter: "{{ openstack_db_setup_python_interpreter |␣
↪default((aodh_db_setup_host == 'localhost') | ternary(ansible_playbook_
↪python, ansible_facts['python']['executable'])) }}"
aodh_db_address: "{{ galera_address | default('127.0.0.1') }}"
aodh_database_name: aodh
aodh_database_user: aodh
aodh_db_type: "mysql+pymysql"
aodh_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
aodh_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
aodh_connection_string: "{{ aodh_db_type }}://{{ aodh_database_user }}:{{␣
↪aodh_container_db_password }}@{{ aodh_db_address }}/{{ aodh_database_name }}
↪?charset=utf8{% if aodh_galera_use_ssl | bool %}&ssl_verify_cert=true{% if␣
↪aodh_galera_ssl_ca_cert | length > 0 %}&ssl_ca={{ aodh_galera_ssl_ca_cert }}
↪{% endif %}{% endif %}"
aodh_galera_port: "{{ galera_port | default('3306') }}"
aodh_db_max_overflow: "{{ openstack_db_max_overflow | default('50') }}"
aodh_db_max_pool_size: "{{ openstack_db_max_pool_size | default('5') }}"
aodh_db_pool_timeout: "{{ openstack_db_pool_timeout | default('30') }}"
aodh_db_connection_recycle_time: "{{ openstack_db_connection_recycle_time |␣
↪default('600') }}"

# Oslo Messaging
# RPC
aodh_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default('rabbitmq_
↪all') }}"
aodh_oslomsg_rpc_setup_host: "{{ (aodh_oslomsg_rpc_host_group in groups) |␣
↪ternary(groups[aodh_oslomsg_rpc_host_group][0], 'localhost') }}"
aodh_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit') }}"
aodh_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }}"
aodh_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
aodh_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
aodh_oslomsg_rpc_userid: aodh
aodh_oslomsg_rpc_vhost: /aodh
aodh_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2
↪') }}"
aodh_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Notify
aodh_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
aodh_oslomsg_notify_setup_host: "{{ (aodh_oslomsg_notify_host_group in␣
↪groups) | ternary(groups[aodh_oslomsg_notify_host_group][0], 'localhost') }}
↪"
```

(continues on next page)

(continued from previous page)

```
aodh_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit
↪') }}"
aodh_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.0.1
↪') }}"
aodh_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
aodh_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
aodh_oslomsg_notify_userid: "{{ aodh_oslomsg_rpc_userid }}"
aodh_oslomsg_notify_password: "{{ aodh_oslomsg_rpc_password }}"
aodh_oslomsg_notify_vhost: "{{ aodh_oslomsg_rpc_vhost }}"
aodh_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version | default(
↪'TLSv1_2') }}"
aodh_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file | default('')␣
↪}}"

## (Qdrouterd) info
# TODO(ansmith): Change structure when more backends will be supported
aodh_oslomsg_amqp1_enabled: "{{ aodh_oslomsg_rpc_transport == 'amqp' }}"

## uWSGI setup
aodh_wsgi_threads: 1
aodh_wsgi_processes_max: 16
aodh_wsgi_processes: "{{ [[(ansible_facts['processor_vcpus']//ansible_facts[
↪'processor_threads_per_core'])|default(1), 1] | max * 2, aodh_wsgi_
↪processes_max] | min }}"

#Aodh services info
aodh_role_name: admin

## Service Type and Data
aodh_service_region: "{{ service_region | default('RegionOne') }}"
aodh_service_endpoint_type: internalURL
aodh_service_name: aodh
aodh_service_bind_address: "{{ openstack_service_bind_address | default('0.0.
↪0.0') }}"
aodh_service_port: 8042
aodh_service_proto: http
aodh_service_type: alarming
aodh_service_description: "Telemetry"
aodh_service_project_domain_id: default
aodh_service_user_domain_id: default
aodh_service_user_name: aodh
aodh_keystone_auth_plugin: "{{ aodh_keystone_auth_type }}"
aodh_keystone_auth_type: password
aodh_service_tenant_name: service
aodh_service_project_name: service
aodh_service_publicuri: "{{ aodh_service_proto }}://{{ external_lb_vip_
↪address }}:{{ aodh_service_port }}"
aodh_service_publicurl: "{{ aodh_service_publicuri }}"
aodh_service_internaluri: "{{ aodh_service_proto }}://{{ internal_lb_vip_
↪address }}:{{ aodh_service_port }}"
```

(continues on next page)

(continued from previous page)

```
aodh_service_internalurl: "{{ aodh_service_internaluri }}"
aodh_service_adminuri: "{{ aodh_service_proto }}://{{ internal_lb_vip_address␣
↪}}:{{ aodh_service_port }}"
aodh_service_adminurl: "{{ aodh_service_adminuri }}"

aodh_service_in_ldap: "{{ service_ldap_backend_enabled | default(False) }}"

# Common pip packages
aodh_pip_packages:
  - "git+{{ aodh_git_repo }}@{{ aodh_git_install_branch }}#egg=aodh"
  - aodh
  - ceilometermiddleware
  - cryptography
  - gnocchiclient
  - osprofiler
  - PyMySQL
  - pymemcache
  - python-memcached
  - sqlalchemy
  - sqlalchemy-utils
  - systemd-python
  - warlock
aodh_user_pip_packages: []

aodh_optional_oslomsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

aodh_memcached_servers: "{{ memcached_servers }}"

aodh_alarm_notifier_init_overrides: {}
aodh_alarm_evaluator_init_overrides: {}
aodh_api_init_overrides: {}
aodh_listener_init_overrides: {}

## Service Name-Group Mapping
aodh_services:
  aodh-api:
    group: aodh_api
    service-name: aodh-api
    init_config_overrides: "{{ aodh_api_init_overrides }}"
    wsgi_app: True
    wsgi_name: aodh-api
    uwsgi_overrides: "{{ aodh_uwsgi_conf_overrides }}"
    uwsgi_port: "{{ aodh_service_port }}"
    uwsgi_bind_address: "{{ aodh_service_bind_address }}"
  aodh-notifier:
    group: aodh_alarm_notifier
    service_name: aodh-notifier
    execstarts: "{{ aodh_bin }}/aodh-notifier"
```

(continues on next page)

(continued from previous page)

```
    init_config_overrides: "{{ aodh_alarm_notifier_init_overrides }}"
  aodh-evaluator:
    group: aodh_alarm_evaluator
    service_name: aodh-evaluator
    execstarts: "{{ aodh_bin }}/aodh-evaluator"
    init_config_overrides: "{{ aodh_alarm_evaluator_init_overrides }}"
  aodh-listener:
    group: aodh_listener
    service_name: aodh-listener
    execstarts: "{{ aodh_bin }}/aodh-listener"
    init_config_overrides: "{{ aodh_listener_init_overrides }}"

aodh_required_secrets:
  - memcached_encryption_key
  - aodh_container_db_password
  - aodh_oslomsg_rpc_password
  - aodh_oslomsg_notify_password
  - aodh_service_password

install_test_packages: False

## Tunable overrides
aodh_policy_overrides: {}
aodh_aodh_conf_overrides: {}
aodh_api_paste_ini_overrides: {}
aodh_uwsgi_conf_overrides: {}
```

CHAPTER

THREE

EXAMPLE PLAYBOOK

```yaml
---
- name: Install aodh services
  hosts: aodh_all
  user: root
  roles:
    - role: "os_aodh"
  vars:
    # Needed for aodh to talk to MongoDB
    aodh_container_db_password: "secrete"
    # Password used for Keystone aodh service user
    aodh_service_password: "secrete"
    # Needed for aodh to talk to memcached
    memcached_servers: 127.0.0.1
    memcached_encryption_key: "some_key"
    # Needed for aodh to locate and connect to Oslo.Messaging
    aodh_oslomsg_rpc_password: "secrete"
    # Needed to setup the aodh service in Keystone
    keystone_admin_user_name: admin
    keystone_admin_tenant_name: admin
    keystone_auth_admin_password: "SuperSecretePassword"
    keystone_service_adminuri_insecure: false
    keystone_service_internaluri_insecure: false
    keystone_service_internaluri: "http://1.2.3.4:5000"
    keystone_service_internalurl: "{{ keystone_service_internaluri }}/v3"
    keystone_service_adminuri: "http://5.6.7.8:5000"
    keystone_service_adminurl: "{{ keystone_service_adminuri }}/v3"
    galera_root_user: root
  vars_prompt:
    - name: "galera_root_password"
      prompt: "What is galera_root_password?"
```

CHAPTER FOUR

DEPENDENCIES

This role needs pip >= 7.1 installed on the target host.

To use this role, define the following variables:

```
# Needed for aodh to talk to MongoDB
aodh_container_db_password: "secrete"
# Password used for Keystone aodh service user
aodh_service_password: "secrete"
# Needed for aodh to talk to memcached
memcached_servers: 127.0.0.1
memcached_encryption_key: "some_key"
# Needed for aodh to locate and connect to Oslo.Messaging
aodh_oslomsg_rpc_transport: rabbit
aodh_oslomsg_rpc_password: "secrete"
aodh_oslomsg_rpc_servers: "10.100.100.2"
aodh_oslomsg_rpc_use_ssl: true
aodh_oslomsg_rpc_port: 5671
aodh_oslomsg_notify_transport: rabbit
aodh_oslomsg_notify_password: "secrete"
aodh_oslomsg_notify_servers: "10.100.100.2"
aodh_oslomsg_notify_use_ssl: true
aodh_oslomsg_notify_port: 5671
# Needed to setup the aodh service in Keystone
keystone_admin_user_name: admin
keystone_admin_tenant_name: admin
keystone_auth_admin_password: "SuperSecretePassword"
keystone_service_adminuri_insecure: false
keystone_service_internaluri_insecure: false
keystone_service_internaluri: "http://1.2.3.4:5000"
keystone_service_internalurl: "{{ keystone_service_internaluri }}/v3"
keystone_service_adminuri: "http://5.6.7.8:5000"
keystone_service_adminurl: "{{ keystone_service_adminuri }}/v3"
```

CHAPTER FIVE

TAGS

This role supports two tags: `aodh-install` and `aodh-config`. The `aodh-install` tag can be used to install and upgrade. The `aodh-config` tag can be used to maintain the configuration of the service.