OpenStack-Ansible Documentation: os_sahara role

Release 18.1.0.dev204

OpenStack-Ansible Contributors

Aug 31, 2023

CONTENTS

CHAPTER

ONE

CONFIGURING THE DATA PROCESSING (SAHARA) SERVICE (OPTIONAL)

Note: This feature is experimental at this time and it has not been fully production tested yet.

Sahara provide users with a simple means to provision data processing frameworks (such as Hadoop, Spark and Storm) on OpenStack.

Sahara is configured using the /etc/openstack_deploy/conf.d/sahara.yml file and the /etc/openstack_deploy/user_variables.yml file.

1.1 Configuring target hosts

Modify /etc/openstack_deploy/conf.d/sahara.yml by adding a list containing the infrastructure target hosts in the sahara-infra_hosts section:

In sahara.yml:

```
sahara-infra_hosts:
  infra01:
    ip: INFRA01_IP_ADDRESS
  infra02:
    ip: INFRA02_IP_ADDRESS
  infra03:
    ip: INFRA03_IP_ADDRESS
```

Replace *_IP_ADDRESS with the IP address of the br-mgmt container management bridge on each target host.

This hosts will be used to deploy the containers where sahara will be installed.

1.2 Configuring the cluster network

Sahara is configured to use the neutron implementation of OpenStack Networking.

1.2.1 Floating IP management

By default sahara is configured to use fixed IP addresses for access. This is controlled by the sahara_use_floating_ips variable. By changing sahara_use_floating_ips to True the user may specify a floating IP address pool for each node group directly.

In user_variables.yml:

```
sahara_use_floating_ips: False
```

Warning: When using floating IP addresses for management **every** instance in the cluster must have a floating IP address, otherwise sahara will not be able to utilize that cluster.

When using fixed IP addresses (sahara_use_floating_ips=False) the user will be able to choose the fixed IP network for all instances in a cluster. It is important to ensure that all instances running sahara have access to the fixed IP networks.

1.3 Object Storage access using proxy users

By default sahara is configured to use proxy users and delegated trusts for Object Storage access. In that way, users are not required to enter credentials for their data sources and job binaries referenced in Object Storage. To disable this functionality change the following variable to False.

In user_variables.yml:

```
sahara_use_domain_for_proxy_users: True
```

Also, is it possible to change which roles the trust users will receive in the proxy domain, by default it receives the member role.

In user_variables.yml:

```
sahara_proxy_user_role_names: member
```

Warning: In the context of the proxy user, any roles that are required for Object Storage access by the project owning the object store must be delegated to the proxy user for authentication to be successful.

1.4 Configuring cluster instances NTP

By default sahara will enable the NTP service on all cluster instances if the NTP package is included in the image. The default NTP server will be pool.ntp.org this can be overridden using the sahara_default_ntp_server variable.

In user_variables.yml:

```
sahara_default_ntp_server: "pool.ntp.org"
```

1.5 Configuring plugins

The following plugins are installed and loaded by default:

```
sahara_plugin_base:
  - vanilla
  - spark
  - cdh
  - ambari
```

To add/remove plugins, just change the sahara_plugin_base variable accordingly, in the user_variables.yml file.

1.6 Configuring notifications

Sahara can be configured to send notifications to the OpenStack Telemetry module. By default, the variable is set to true if there are any Ceilometer hosts in the environment. To change this, the following variable must be set:

In user_variables.yml:

```
sahara_ceilometer_enabled: True
```

1.7 Dashboard

To enable the Data Processing panel on horizon, the following variable should be set:

In user_variables.yml:

```
horizon_enable_sahara_ui: True
```

1.8 Setting up Sahara

Run the setup-hosts playbook, to create the sahara containers, and the repo-build playbook to update the repository with the sahara packages.

```
# cd /opt/openstack-ansible/playbooks
# openstack-ansible setup-hosts.yml
# openstack-ansible repo-build.yml
```

Run the sahara and horizon playbooks to install sahara and enable the Data Processing panel in horizon:

```
# cd /opt/openstack-ansible/playbooks
# openstack-ansible os-sahara-install.yml
# openstack-ansible os-horizon-install.yml
```

This role installs the following Systemd services:

- sahara-api
- sahara-engine

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

CHAPTER

TWO

DEFAULT VARIABLES

```
## Verbosity Options
debug: False

#python venv executable
sahara_venv_python_executable: "{{ openstack_venv_python_executable | default(
↪'python3') }}"

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
sahara_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
sahara_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((sahara_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'
sahara_package_state: "{{ package_state | default('latest') }}"

sahara_git_repo: https://opendev.org/openstack/sahara
sahara_git_install_branch: master
sahara_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
sahara_git_constraints:
  - "--constraint {{ sahara_upper_constraints_url }}"

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

# Name of the virtual env to deploy into
sahara_venv_tag: "{{ venv_tag | default('untagged') }}"
sahara_bin: "/openstack/venvs/sahara-{{ sahara_venv_tag }}/bin"

sahara_etc_dir: "{{ sahara_bin | dirname }}/etc/sahara"

sahara_db_config: /etc/sahara/sahara.conf
```

(continues on next page)

(continued from previous page)

```
# Enable/Disable Ceilometer
sahara_ceilometer_enabled: "{{ (groups['ceilometer_all'] is defined) and␣
↪(groups['ceilometer_all'] | length > 0) }}"

sahara_profiler_enabled: False
sahara_fatal_deprecations: False

## System info
sahara_system_user_name: sahara
sahara_system_group_name: sahara
sahara_system_shell: /bin/false
sahara_system_comment: sahara system user
sahara_system_user_home: "/var/lib/{{ sahara_system_user_name }}"

sahara_engine_host: "{{ internal_lb_vip_address }}"

## Oslo Messaging Info
# RPC
sahara_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default('rabbitmq_
↪all') }}"
sahara_oslomsg_rpc_setup_host: "{{ (sahara_oslomsg_rpc_host_group in groups)␣
↪| ternary(groups[sahara_oslomsg_rpc_host_group][0], 'localhost') }}"
sahara_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit') }}
↪"
sahara_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }}"
sahara_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
sahara_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
sahara_oslomsg_rpc_userid: sahara
sahara_oslomsg_rpc_vhost: /sahara
sahara_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2
↪') }}"
sahara_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Notify
sahara_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
sahara_oslomsg_notify_setup_host: "{{ (sahara_oslomsg_notify_host_group in␣
↪groups) | ternary(groups[sahara_oslomsg_notify_host_group][0], 'localhost')␣
↪}}"
sahara_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
sahara_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.0.1
↪') }}"
sahara_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
sahara_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
sahara_oslomsg_notify_userid: "{{ sahara_oslomsg_rpc_userid }}"
sahara_oslomsg_notify_password: "{{ sahara_oslomsg_rpc_password }}"
sahara_oslomsg_notify_vhost: "{{ sahara_oslomsg_rpc_vhost }}"
```

(continues on next page)

(continued from previous page)

```
sahara_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version | default(
↪'TLSv1_2') }}"
sahara_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file | default('
↪') }}"

## (Qdrouterd) info
# TODO(ansmith): Change structure when more backends will be supported
sahara_oslomsg_amqp1_enabled: "{{ sahara_oslomsg_rpc_transport == 'amqp' }}"

## Database info
sahara_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
sahara_db_setup_python_interpreter: "{{ openstack_db_setup_python_interpreter␣
↪| default((sahara_db_setup_host == 'localhost') | ternary(ansible_playbook_
↪python, ansible_facts['python']['executable'])) }}"
sahara_galera_address: "{{ galera_address | default('127.0.0.1') }}"
sahara_galera_database: sahara
sahara_galera_user: sahara
sahara_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
sahara_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
sahara_galera_port: "{{ galera_port | default('3306') }}"
sahara_db_max_overflow: "{{ openstack_db_max_overflow | default('50') }}"
sahara_db_max_pool_size: "{{ openstack_db_max_pool_size | default('5') }}"
sahara_db_pool_timeout: "{{ openstack_db_pool_timeout | default('30') }}"
sahara_db_connection_recycle_time: "{{ openstack_db_connection_recycle_time |␣
↪default('600') }}"

sahara_api_bind_address: "{{ openstack_service_bind_address | default('0.0.0.0
↪') }}"
sahara_api_service_port: 8386

## Service Type and Data
sahara_service_region: "{{ service_region | default('RegionOne') }}"
sahara_service_name: sahara
sahara_service_port: 8386
sahara_service_proto: http
sahara_service_engine_proto: "{{ sahara_service_proto }}"
sahara_service_publicuri_proto: "{{ openstack_service_publicuri_proto |␣
↪default(sahara_service_proto) }}"
sahara_service_adminuri_proto: "{{ openstack_service_adminuri_proto |␣
↪default(sahara_service_proto) }}"
sahara_service_internaluri_proto: "{{ openstack_service_internaluri_proto |␣
↪default(sahara_service_proto) }}"
sahara_service_type: data-processing
sahara_service_description: "Sahara Data Processing Service"
sahara_service_user_name: sahara
sahara_service_project_name: service
sahara_service_project_domain_id: default
sahara_service_user_domain_id: default
sahara_service_project_domain_name: Default
```

(continues on next page)

(continued from previous page)

```
sahara_service_user_domain_name: Default
sahara_service_role_names:
  - admin
  - service
sahara_service_token_roles:
  - service
sahara_service_token_roles_required: "{{ openstack_service_token_roles_
↪required | default(True) }}"
sahara_service_publicuri: "{{ sahara_service_publicuri_proto }}://{{ external_
↪lb_vip_address }}:{{ sahara_service_port }}"
sahara_service_internaluri: "{{ sahara_service_internaluri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ sahara_service_port }}"
sahara_service_adminuri: "{{ sahara_service_adminuri_proto }}://{{ internal_
↪lb_vip_address }}:{{ sahara_service_port }}"

## Keystone authentication middleware
sahara_keystone_auth_plugin: password

## Sahara config
sahara_use_floating_ips: False
sahara_use_namespaces: False
sahara_global_remote_threshold: 100
sahara_cluster_remote_threshold: 70
sahara_os_region_name: "{{ sahara_service_region }}"
sahara_default_ntp_server: "pool.ntp.org"

sahara_use_domain_for_proxy_users: True
sahara_proxy_user_domain_name: sahara_proxy
sahara_proxy_user_role_names: member

# Other plugins can be added to the system by simply extending the list␣
↪`sahara_plugin_base`.
sahara_plugin_base:
  - ambari
  - cdh
  - mapr
  - spark
  - storm
  - vanilla

sahara_plugin_loaded_base: "{% for plugin in sahara_plugin_base %}{{ plugin }}
↪{% if not loop.last %},{% endif %}{% endfor %}"
sahara_plugin_pip_packages:
  - sahara-plugin-ambari
  - sahara-plugin-cdh
  - sahara-plugin-mapr
  - sahara-plugin-spark
  - sahara-plugin-storm
  - sahara-plugin-vanilla
```

(continues on next page)

(continued from previous page)

```
## Cap the maximum number of threads / workers when a user value is unspecified.
sahara_api_workers_max: 16
sahara_api_workers: "{{ [[(ansible_facts['processor_vcpus']//ansible_facts['processor_threads_per_core'])|default(1), 1] | max * 2, sahara_api_workers_max] | min }}"

sahara_task_executor: taskflow
sahara_digest_algorithm: sha256
sahara_http_keepalive: True

## Sahara policy
sahara_policy_file: policy.json
sahara_policy_default_rule: default
sahara_policy_dirs: policy.d

## Policy vars
# Provide a list of access controls to update the default policy.json with. These changes will be merged
# with the access controls in the default policy.json. E.g.
#sahara_policy_overrides:
#  "clusters:get_all": ""
#  "clusters:create": ""

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

# Common pip packages
sahara_pip_packages:
  - cryptography
  - keystonemiddleware
  - osprofiler
  - PyMySQL
  - pymemcache
  - python-memcached
  - python-saharaclient
  - python-keystoneclient
  - "git+{{ sahara_git_repo }}@{{ sahara_git_install_branch }}#egg=sahara"
  - systemd-python
  - warlock

# Memcached override
sahara_memcached_servers: "{{ memcached_servers }}"

sahara_optional_oslomsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

# Specific pip packages provided by the user
sahara_user_pip_packages: []
```

(continues on next page)

(continued from previous page)

```
sahara_engine_init_overrides: {}
sahara_api_init_overrides: {}

## Service Names
sahara_services:
  sahara-api:
    group: sahara_api
    service_name: sahara-api
    init_config_overrides: "{{ sahara_api_init_overrides }}"
    wsgi_app: True
    wsgi_name: sahara-wsgi-api
    uwsgi_overrides: "{{ sahara_api_uwsgi_ini_overrides }}"
    uwsgi_port: "{{ sahara_api_service_port }}"
    uwsgi_bind_address: "{{ sahara_api_bind_address }}"
    uwsgi_tls: "{{ sahara_backend_ssl | ternary(sahara_uwsgi_tls, {}) }}"
  sahara-engine:
    group: sahara_engine
    service_name: sahara-engine
    init_config_overrides: "{{ sahara_engine_init_overrides }}"
    execstarts: "{{ sahara_bin }}/sahara-engine"

## Sahara uWSGI settings
sahara_wsgi_processes_max: 16
sahara_wsgi_processes: "{{ [[ansible_facts['processor_vcpus']|default(1), 1]␣
↪| max * 2, sahara_wsgi_processes_max] | min }}"
sahara_wsgi_threads: 1
sahara_uwsgi_tls:
  crt: "{{ sahara_ssl_cert }}"
  key: "{{ sahara_ssl_key }}"

## Tunable overrides
sahara_api_paste_ini_overrides: {}
sahara_conf_overrides: {}
sahara_rootwrap_conf_overrides: {}
sahara_policy_overrides: {}
sahara_api_uwsgi_ini_overrides: {}

###
### Backend TLS
###

# Define if communication between haproxy and service backends should be
# encrypted with TLS.
sahara_backend_ssl: "{{ openstack_service_backend_ssl | default(False) }}"

# Storage location for SSL certificate authority
sahara_pki_dir: "{{ openstack_pki_dir | default('/etc/openstack_deploy/pki') }
↪}"
```

(continues on next page)

(continued from previous page)

```
# Delegated host for operating the certificate authority
sahara_pki_setup_host: "{{ openstack_pki_setup_host | default('localhost') }}"

# sahara server certificate
sahara_pki_keys_path: "{{ sahara_pki_dir ~ '/certs/private/' }}"
sahara_pki_certs_path: "{{ sahara_pki_dir ~ '/certs/certs/' }}"
sahara_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_
↪cert_name | default('ExampleCorpIntermediate') }}"
sahara_pki_regen_cert: ''
sahara_pki_san: "{{ openstack_pki_san | default('DNS:' ~ ansible_facts[
↪'hostname'] ~ ',IP:' ~ management_address) }}"
sahara_pki_certificates:
  - name: "sahara_{{ ansible_facts['hostname'] }}"
    provider: ownca
    cn: "{{ ansible_facts['hostname'] }}"
    san: "{{ sahara_pki_san }}"
    signed_by: "{{ sahara_pki_intermediate_cert_name }}"

# sahara destination files for SSL certificates
sahara_ssl_cert: /etc/sahara/sahara.pem
sahara_ssl_key: /etc/sahara/sahara.key

# Installation details for SSL certificates
sahara_pki_install_certificates:
  - src: "{{ sahara_user_ssl_cert | default(sahara_pki_certs_path ~ 'sahara_'␣
↪~ ansible_facts['hostname'] ~ '-chain.crt') }}"
    dest: "{{ sahara_ssl_cert }}"
    owner: "{{ sahara_system_user_name }}"
    group: "{{ sahara_system_user_name }}"
    mode: "0644"
  - src: "{{ sahara_user_ssl_key | default(sahara_pki_keys_path ~ 'sahara_' ~␣
↪ansible_facts['hostname'] ~ '.key.pem') }}"
    dest: "{{ sahara_ssl_key }}"
    owner: "{{ sahara_system_user_name }}"
    group: "{{ sahara_system_user_name }}"
    mode: "0600"

# Define user-provided SSL certificates
#sahara_user_ssl_cert: <path to cert on ansible deployment host>
#sahara_user_ssl_key: <path to cert on ansible deployment host>
```

CHAPTER

THREE

DEPENDENCIES

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

To use this role, define the following variables:

```
sahara_galera_address
sahara_container_mysql_password
sahara_service_password
sahara_rabbitmq_password
```

CHAPTER FOUR

EXAMPLE PLAYBOOK

```yaml
- name: Install sahara server
  hosts: sahara_all
  user: root
  roles:
    - { role: "os_sahara", tags: [ "os-sahara" ] }
  vars:
    external_lb_vip_address: 172.16.24.1
    internal_lb_vip_address: 192.168.0.1
    sahara_galera_address: "{{ internal_lb_vip_address }}"
    sahara_container_mysql_password: "SuperSecretePassword1"
    sahara_service_password: "SuperSecretePassword2"
    sahara_rabbitmq_password: "SuperSecretePassword3"
    galera_root_user: root
  vars_prompt:
    - name: "galera_root_password"
      prompt: "What is galera_root_password?"
```

CHAPTER

FIVE

TAGS

This role supports two tags: `sahara-install` and `sahara-config`. The `sahara-install` tag can be used to install and upgrade. The `sahara-config` tag can be used to manage configuration.