목차

RabbitMQ 성능 튜닝 with OVS 3
참조링크 5

RabbitMQ 성능 튜닝 with OVS

Tune the RabbitMQ performance in the OpenStack with OVS deployments Proper configuration of Nova and Neutron services in your Reclass deployment model allows for decreasing the load on the RabbitMQ service making the service more stable under high load in the deployments with 1000+ nodes.

To tune the RabbitMQ performance on a new MCP OpenStack deployment:

Generate a deployment metadata model for your new MCP OpenStack as described in Create a deployment metadata model using the Model Designer UI.

Open the cluster level of your Git project repository.

In openstack/gateway.yml, define the following parameters as required. For example:

```
neutron:
  gateway:
    dhcp_lease_duration: 86400
    message_queue:
      rpc_conn_pool_size: 300
      rpc_thread_pool_size: 2048
      rpc_response_timeout: 3600
```

In openstack/compute/init.yml, define the following parameters as required. For example:

```
neutron:
  compute:
    message_queue:
      rpc_conn_pool_size: 300
      rpc_thread_pool_size: 2048
      rpc_response_timeout: 3600
```

In openstack/control.yml, define the following parameters as required. For example:

```
nova:
  controller:
    timeout_nbd: 60
    heal_instance_info_cache_interval: 600
    block_device_creation_timeout: 60
    vif_plugging_timeout: 600
    message_queue:
      rpc_poll_timeout: 60
      connection_retry_interval_max: 60
      default_reply_timeout: 60
```

```
      default_send_timeout: 60
      default_notify_timeout: 60
```

In openstack/compute/init.yml, define the following parameters as required. For example:

```
nova:
  compute:
    timeout_nbd: 60
    heal_instance_info_cache_interval: 600
    block_device_creation_timeout: 60
    vif_plugging_timeout: 600
    message_queue:
      rpc_poll_timeout: 60
      connection_retry_interval_max: 60
      default_reply_timeout: 60
      default_send_timeout: 60
      default_notify_timeout: 60
```

In openstack/control.yml, define the following parameters as required. For example:

```
neutron:
  server:
    dhcp_lease_duration: 86400
    agent_boot_time: 7200
    message_queue:
      rpc_conn_pool_size: 300
      rpc_thread_pool_size: 2048
      rpc_response_timeout: 3600
```

Optional. Set additional parameters to improve the RabbitMQ performance.

The following parameters should be set in correlation with each other. For example, the value of the report_interval parameter should be a half or less than the value of the agent_down_time parameter. The report_interval parameter should be set on all nodes where the Neutron agents are running.

In openstack/control.yml, define the agent_down_time parameter as required. For example:

```
neutron:
  server:
    agent_down_time: 300
```

In openstack/compute/init.yml and openstack/gateway.yml, define the report_interval parameter as required. For example:

```
neutron:
  compute:
    report_interval: 120
```

Caution

The time of workload being unavailable can be increased in case of the Neutron agents failover. Though, the number of the AMQP messages in the RabbiMQ queues can be lower.

Optional. To speed up message handling by the Neutron agents and Neutron API, define the rpc_workers parameter in openstack/control.yml. The defined number of workers should be equal to the number of CPUs multiplied by two. For example, if the number of CPU is 24, set the rpc_workers parameter to 48:

```
neutron:
  server:
     rpc_workers: 48
```

Optional. Set the additional parameters for the Neutron server role to improve stability of the networking configuration:

Set the allow_automatic_dhcp_failover parameter to false. If set to true, the server reschedules nets from the failed DHCP agents so that the alive agents catch up the net and serve DHCP. Once the agent reconnects to RabbitMQ, the agent detects that its net has been rescheduled and removes the DHCP port, namespace, and flows. This parameter was implemented for the use cases when the whole gateway node goes down. In case of the RabbitMQ instability, agents do not actually go down, and the data plane is not affected. Therefore, we recommend that you set it to false. But you should consider the risks of a gateway node going down as well before setting the allow_automatic_dhcp_failover parameter. Define the dhcp_agents_per_network parameter that sets the number of the DHCP agents per network. To have one DHCP agent on each gateway node, set the parameter to the number of the gateway nodes in your deployment. For example, dhcp_agents_per_network: 3. Configuration example:

```
neutron:
  server:
    dhcp_agents_per_network: 3
    allow_automatic_dhcp_failover: false
```

Proceed to the new MCP OpenStack environment configuration and deployment as required.

참조링크

- https://docs.mirantis.com/mcp/q4-18/mcp-deployment-guide/advanced-config/tune-rabbitmq-perf.html

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/rabbitmq_%EC%84%B1%EB%8A%A5_%ED%8A%9C%EB%8B%9D_with_ovs

Last update: **2023/04/11 03:30**