Test Plan for VPNaaS 1.1.0
Fuel Plugin

Revision history

Version	Revision date	Editor	Comment
1.0	05.03.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com)	Created test plan
1.1	07.04.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com)	Modified Developer's specification and Appendix
1.2	08.04.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com)	Deleted Priority and check links
1.3	17.04.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com)	Modified with changes in the mode: test_plan_template
1.4	21.05.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com	Add destructive tests
1.5	29.06.2015	Kristina Kuznetsova (kkuznetsova@mirantis.com	Add system tests

VPNaaS Plugin

VPNaaS (VPN-as-a-Service) is a Neutron extension that introduces VPN feature set in Neutron which based on Openswan (opensource IPSec implementation).
A VPN connection between 2 private subnet, which are placed in 2 different tenants in OpenStack clouds.
The main goal of this document is to describe the test cases for VPNaaS plugin on Mirantis OpenStack project. These tests cases should be used during the acceptance testing for each new release.

Developer's specification

Blueprint: link
Demo: link
Spec review: link
VPNaaS Plugin Guide: link

Limitations

VPNaaS plugin can be enabled only in environments with Neutron as the networking option

Test strategy

Type of implemented tests is functional testing. In future all tests for the CLI will be automated. There is plan to add some integration test cases with others plugins (FWaaS) and some more new functional test cases. We also plan to add tests in Rally. We are going to study what is in Tempest and then we will decided if there is necessary a few changes.

Acceptance criteria

All tests should be passed without any errors and exceptions.

Test environment, infrastructure and tools

Deploy environment (3 controllers, 1 compute, Neutron networking) with default parameters with installing VPNaaS plugin. Other recommendation you can see in the test cases

Product compatibility matrix

Requirement	Version/Comment
Fuel	6.1 release
OpenStack compatibility	2014.2 Juno
Operating systems	Ubuntu 14.04 LTS, CentOS 6.5
Plugin version	VPNaaS Fuel Plugin 1.1.0

Functional testing

Title	**TC 001: Ability to create IKE Policy, rename, edit and delete it**
Test Case ID	create_update_delete_ikepolicy
Steps	CLI: 1) Go with ssh to the fuel ip: ssh root@<fuel ip> 2) Look what nodes we have: fuel node 3) Go to the controller node: ssh node-<node_id> 4) . openrc 5) Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 6) Check that test_ike_policy appeared in the list: neutron vpn-ikepolicy-list 7) Rename created IKE Policy: neutron vpn-ikepolicy-update <ike_policy_id> --name rename_test_ike_policy 8) Check that the name has been changed: neutron vpn-ikepolicy-list 9) Edit rename_test_ike_policy: neutron vpn-ikepolicy-update rename_test_ike_policy --description "edit ike policy" --encryption-algorithm aes-256 10) Check that encryption algorithm has been changed: neutron vpn-ikepolicy-show rename_test_ike_policy 11) Delete IKE Policy: neutron vpn-ikepolicy-delete rename_test_ike_policy 12) Check that rename_test_ike_policy has been deleted: neutron vpn-ikepolicy-list
	Dashboard: 1) Login to OpenStack Horizon dashboard 2) Navigate to Project->Network->VPN 3) Navigate to tab IKE Policies 4) Click on Add IKE Policy 5) Type name test_ike_policy 6) Click on Add 7) Check that in the list of IKE Policies appeared test_ike_policy 8) Click on Edit IKE Policy in the line with test_ike_policy 9) Type name rename_test_ike_policy 10) Click on Save changes 11) Check that the IKE Policy has been changed 12) Click on Edit IKE Policy in the line with rename_test_ike_policy 13) Choose Encryption algorithm aes-256

	14) Click on Save changes 15) Check that in the list encryption algorithm has been changed 16) Click on raw down button in the line with rename_test_ike_policy 17) Click on Delete IKE Policy 18) Check that rename_test_ike_policy has been disappeared
Expected Result	All steps should be passed, we should have the ability to create, rename, edit and delete IKE Policy

Title	**TC 002: Ability to create IPSec Policy, rename, edit and delete it**
Test Case ID	create_update_delete
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 6. Check that IPSec has been created: neutron vpn-ipsecpolicy-list 7. Rename created IPSec Policy: neutron vpn-ipsecpolicy-update test_ipsec_policy --name rename_policy 8. Check that the name has been changed: neutron vpn-ipsecpolicy-list 9. Edit rename_policy: neutron vpn-ipsecpolicy-update rename_policy --encryption-algorithm aes-192 10. Check changes: neutron vpn-ipsecpolicy-show rename_policy 11. Delete IPSec Policy: neutron vpn-ipsecpolicy-delete rename_policy 12. Check that rename_policy has been deleted: neutron vpn-ipsecpolicy-list
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project->Network->VPN 3. Navigate to tab IPSec Policies 4. Click on Add IPSec Policy 5. Type name test_ipsec_policy 6. Click on Add 7. Check that in the list of IPSec Policies appeared test_ipsec_policy 8. Click on Edit IPSec Policy in the line with test_ipsec_policy 9. Type name rename_test_ipsec_policy 10. Click on Save changes

	11. Check that the IPSec Policy has been changed 12. Click on Edit IPSec Policy in the line with rename_test_ike_policy 13. Choose Encryption algorithm aes-256 14. Click on Save changes 15. Check that in the list encryption algorithm has been changed 16. Click on raw down button in the line with rename_test_ipsec_policy 17. Click on Delete IPSec Policy 18. Check that rename_test_ipsec_policy has been disappeared
Expected Result	All steps should be passed, we should have the ability to create, rename, edit and delete IPSec Policy

Title	**TC 003: Ability to create VPN Service and delete it**
Test Case ID	create_update_delete_vpnservice
Steps	1. CLI: 2. Go with ssh to the fuel ip: ssh root@<fuel ip> 3. Look what nodes we have: fuel node 4. Go to the controller node: ssh node-<node_id> 5. . openrc 6. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 7. Check that VPN Service has been created: neutron vpn-service-list 8. Delete test_service: neutron vpn-service-delete test_service 9. Check deleting: neutron vpn-service-list
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project->Network->VPN 3. Navigate to tab VPNService 4. Click on Add VPN Service 5. Type name test_service 6. Select router 7. Select private subnet for this router 8. Click on Add 9. Check that in the list of VPN Services appeared test_service 10. Click on raw down button in the line with test_service 11. Click on Delete VPN Service 12. Check that test_service has been disappeared
Expected Result	All steps should be passed, we should have the ability to create, rename, edit and delete VPN Service

Title	**TC 004: Ability to create IPSec Site Connections and delete it**
Test Case ID	create_update_delete_ipsec_site_connection
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 6. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 7. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 8. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_ip> --peer-id <router_ip> --peer-cidrs 10.10.10.1/24 --psk key 9. Check that test_connection has been created: neutron ipsec-site-connection-list 10. Delete connection: neutron ipsec-site-connection-delete test_connection 11. Check deleting: neutron ipsec-site-connection-list
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project->Network->VPN 3. Navigate to tab IKE Policies 4. Click on Add IKE Policy 5. Type name test_ike_policy 6. Click on Add 7. Navigate to tab IPSec Policies 8. Click on Add IPSec Policy 9. Type name test_ipsec_policy 10. Click on Add 11. Navigate to tab VPN Service 12. Click on Add VPN Service 13. Type name test_service 14. Select router 15. Select private subnet for this route 16. Click on Add 17. Navigate to IPSec Site Connections

	18. Click on Add IPSec Site Connections 19. Type name test_connection 20. Select VPN Service test_service 21. Select IKE Policy test_ike_policy 22. Select IPSec Policy test_ipsec_policy 23. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as some address 24. Type Remote peer subnet(s) as some address 25. Type Pre-Shared Key (PSK) string as key 26. Click on Add 27. Check that the new IPSec Connection has been created 28. Click on Delete IPSec Site Connection 29. Check that connection has been disappeared
Expected Result	All steps should be passed, we should have the ability to create, rename, edit and delete IPSec Site Connection

Title	**TC 005: Configure VPNaaS with default parameters between two tenants as admin user, rename and edit ipsec site connection**
Test Case ID	configure_default_vpn_as_admin
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating ip 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet

	16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant_id <test_tenant_id> 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant_id <test_tenant_id> 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created and status is active: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed 29. Return to the tab with node 30. Update test_connection (rename and edit, status should be active for this operations): neutron ipsec-site-connection-update <ipsec_site_connection_id> --psk test 31. Check changes: neutron ipsec-site-connection-show <ipsec_site_connection_id>
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console

	9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy and Click on Add 16. Navigate to tab IPSec Policies 17. Click on Add IPSec Policy 18. Type name test_ipsec_policy 19. Click on Add 20. Navigate to tab VPN Service 21. Click on Add VPN Service 22. Type name test_service 23. Select router 24. Select private subnet for this route 25. Click on Add 26. Navigate to IPSec Site Connections 27. Click on Add IPSec Site Connections 28. Type name test_connection 29. Select VPN Service test_service 30. Select IKE Policy test_ike_policy 31. Select IPSec Policy test_ipsec_policy 32. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 33. Type Remote peer subnet(s) as private network in the second tenant 34. Type Pre-Shared Key (PSK) string as key 35. Click on Add 36. Go to test_tenant 37. Navigate to Project -> Network -> VPN 38. Navigate to tab IKE Policies 39. Click on Add IPSec Policy 40. Type name test_ike_policy and Click on Add 41. Navigate to tab IPSec Policies 42. Click on Add IPSec Policy 43. Type name test_ipsec_policy 44. Click on Add 45. Navigate to tab VPN Service 46. Click on Add VPN Service 47. Type name test_service 48. Select router 49. Select private subnet for this route 50. Click on Add 51. Navigate to IPSec Site Connections

	52. Click on Add IPSec Site Connections 53. Type name test_connection 54. Select VPN Service test_service 55. Select IKE Policy test_ike_policy 56. Select IPSec Policy test_ipsec_policy 57. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 58. Type Remote peer subnet(s) as private network in admin tenant 59. Type Pre-Shared Key (PSK) string as key 60. Click on Add 61. Reopen the page and check that the status of IPSec Site Connection is Active 62. Return to VM_1 Console 63. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 64. Check that traffic is available 65. Return to VM_2 console 66. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 67. Check that traffic is available 68. Go to Horizon 69. Click on Edit Connection in the line with connection 70. Type rename_connection in the field name 71. Type another Remote peer subnet(s) 72. Click on Save Changes 73. Click on the test_connection 74. Check that Remote peer subnet(s) has been changed 75. Return to the previous page 76. Click on raw down button in the line with rename_connection
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 006: Configure VPNaaS with default parameters between two tenants as non-admin user**
Test Case ID	configure_default_vpn_as_nonadmin
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node

	3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating ip 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. vim openrc 19. Rewrite data about non-admin user 20. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant_id <test_tenant_id> 2 21. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant_id <test_tenant_id> 2 22. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 23. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 24. Check that test_connection has been created: neutron ipsec-site-connection-list 25. Return to tab with VM_1 26. Try to ping VM_2: ping <VM_2_private_IP> 27. Check that icmp is allowed 28. Return to tab with VM_2

	29. Try to ping VM_1: ping <VM_1_private_IP> 30. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy and Click on Add 16. Navigate to tab IPSec Policies 17. Click on Add IPSec Policy 18. Type name test_ipsec_policy 19. Click on Add 20. Navigate to tab VPN Service 21. Click on Add VPN Service 22. Type name test_service 23. Select router 24. Select private subnet for this route 25. Click on Add 26. Navigate to IPSec Site Connections 27. Click on Add IPSec Site Connections 28. Type name test_connection 29. Select VPN Service test_service 30. Select IKE Policy test_ike_policy 31. Select IPSec Policy test_ipsec_policy 32. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 33. Type Remote peer subnet(s) as private network in the second tenant 34. Type Pre-Shared Key (PSK) string as key 35. Click on Add 36. Logout and login as non-admin user to test_tenant 37. Navigate to Project -> Network -> VPN 38. Navigate to tab IKE Policies 39. Click on Add IPSec Policy 40. Type name test_ike_policy and Click on Add

	41. Navigate to tab IPSec Policies 42. Click on Add IPSec Policy 43. Type name test_ipsec_policy 44. Click on Add 45. Navigate to tab VPN Service 46. Click on Add VPN Service 47. Type name test_service 48. Select router 49. Select private subnet for this route 50. Click on Add 51. Navigate to IPSec Site Connections 52. Click on Add IPSec Site Connections 53. Type name test_connection 54. Select VPN Service test_service 55. Select IKE Policy test_ike_policy 56. Select IPSec Policy test_ipsec_policy 57. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 58. Type Remote peer subnet(s) as private network in admin tenant 59. Type Pre-Shared Key (PSK) string as key 60. Click on Add 61. Reopen the page and check that the status of IPSec Site Connection is Active 62. Return to VM_1 Console 63. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 64. Check that traffic is available 65. Return to VM_2 console 66. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 67. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants as non-admin user

Title	**TC 007: Configure VPNaaS with non-default parameters between two clouds: IKE Policy with encryption algorithm 'aes128' and IPSec Policy with encryption algorithm 'aes192'**
Test Case ID	configure_vpn_ike-aes128_ipsec-aes192

Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using it's floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-128 test_ile_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-192 test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-128 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key

	22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-128 17. Click on Add 18. Navigate to tab IPSec Policies 19. Click on Add IPSec Policy 20. Type name test_ipsec_policy 21. Choose encryption algorithm aes-192 22. Click on Add 23. Navigate to tab VPN Service 24. Click on Add VPN Service 25. Type name test_service 26. Select router 27. Select private subnet for this route 28. Click on Add 29. Navigate to IPSec Site Connections 30. Click on Add IPSec Site Connections 31. Type name test_connection 32. Select VPN Service test_service 33. Select IKE Policy test_ike_policy 34. Select IPSec Policy test_ipsec_policy

	35. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 36. Type Remote peer subnet(s) as private network in the second tenant 37. Type Pre-Shared Key (PSK) string as key 38. Click on Add 39. Go to test_tenant 40. Navigate to Project -> Network -> VPN 41. Navigate to tab IKE Policies 42. Click on Add IPSec Policy 43. Type name test_ike_policy 44. Choose encryption algorithm aes-128 45. Click on Add 46. Navigate to tab IPSec Policies 47. Click on Add IPSec Policy 48. Type name test_ipsec_policy 49. Choose encryption algorithm aes-192 50. Click on Add 51. Navigate to tab VPN Service 52. Click on Add VPN Service 53. Type name test_service 54. Select router 55. Select private subnet for this route 56. Click on Add 57. Navigate to IPSec Site Connections 58. Click on Add IPSec Site Connections 59. Type name test_connection 60. Select VPN Service test_service 61. Select IKE Policy test_ike_policy 62. Select IPSec Policy test_ipsec_policy 63. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 64. Type Remote peer subnet(s) as private network in admin tenant 65. Type Pre-Shared Key (PSK) string as key 66. Click on Add 67. Reopen the page and check that the status of IPSec Site Connection is Active 68. Return to VM_1 Console 69. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 70. Check that traffic is available 71. Return to VM_2 console 72. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 73. Check that traffic is available

Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 008: Configure VPNaaS with non-default parameters: IKE Policy with Perfect Forward Secrecy “group14” and with encryption algorithm ‘3des’ and IPSec Policy with encryption algorithm ‘aes256’**
Test Case ID	configure_vpn_ike-droup14-3des_ipsec-aes256
Prerequisites	Creation non-admin tenant and launching instances
Steps	1. CLI: 2. Go with ssh to the fuel ip: ssh root@<fuel ip> 3. Look what nodes we have: fuel node 4. Go to the controller node: ssh node-<node_id> 5. . openrc 6. Create new tab 7. Go with ssh to the VM_1 using floating IP 8. Try to ping VM_2: ping <VM_2_private_IP> 9. Check that icmp isn’t allowed 10. Create new tab 11. Go with ssh to the VM_2 using floating IP 12. Try to ping VM_1: ping <VM_1_private_IP> 13. Check that icmp isn’t allowed 14. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm 3des --pfs Group14 test_ike_policy 15. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-256 test_ipsec_policy 16. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 17. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 18. Check that test_connection has been created: neutron ipsec-site-connection-list 19. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm 3des --pfs Group14 2

	20. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes256 2 21. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 22. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 23. Check that test_connection has been created: neutron ipsec-site-connection-list 24. Return to tab with VM_1 25. Try to ping VM_2: ping <VM_2_private_IP> 26. Check that icmp is allowed 27. Return to tab with VM_2 28. Try to ping VM_1: ping <VM_1_private_IP> 29. Check that icmp is allowed
	1. Dashboard: 2. Login to OpenStack Horizon dashboard 3. Navigate to Project -> Network -> Network Topology 4. Click on VM_1 and then on Open Console 5. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 6. Check that traffic isn't allowed 7. Go to test_tenant 8. Navigate to Project -> Network -> Network Topology 9. Click on VM_2 and then on Open Console 10. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 11. Check that traffic isn't allowed 12. Return to the admin tenant 13. Navigate to Project -> Network -> VPN 14. Navigate to tab IKE Policies 15. Click on Add IPSec Policy 16. Type name test_ike_policy 17. Choose encryption algorithm 3des and Perfect forward group group 14 18. Click on Add 19. Navigate to tab IPSec Policies 20. Click on Add IPSec Policy 21. Type name test_ipsec_policy 22. Choose encryption algorithm aes-256 23. Click on Add

	24. Navigate to tab VPN Service 25. Click on Add VPN Service 26. Type name test_service 27. Select router 28. Select private subnet for this route 29. Click on Add 30. Navigate to IPSec Site Connections 31. Click on Add IPSec Site Connections 32. Type name test_connection 33. Select VPN Service test_service 34. Select IKE Policy test_ike_policy 35. Select IPSec Policy test_ipsec_policy 36. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 37. Type Remote peer subnet(s) as private network in the second tenant 38. Type Pre-Shared Key (PSK) string as key 39. Click on Add 40. Go to test_tenant 41. Navigate to Project -> Network -> VPN 42. Navigate to tab IKE Policies 43. Click on Add IPSec Policy 44. Type name test_ike_policy 45. Choose encryption algorithm 3des and Perfect forward group as group 14 46. Click on Add 47. Navigate to tab IPSec Policies 48. Click on Add IPSec Policy 49. Type name test_ipsec_policy 50. Choose encryption algorithm aes-256 51. Click on Add 52. Navigate to tab VPN Service 53. Click on Add VPN Service 54. Type name test_service 55. Select router 56. Select private subnet for this route 57. Click on Add 58. Navigate to IPSec Site Connections 59. Click on Add IPSec Site Connections 60. Type name test_connection 61. Select VPN Service test_service 62. Select IKE Policy test_ike_policy 63. Select IPSec Policy test_ipsec_policy

	64. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 65. Type Remote peer subnet(s) as private network in admin tenant 66. Type Pre-Shared Key (PSK) string as key 67. Click on Add 68. Reopen the page and check that the status of IPSec Site Connection is Active 69. Return to VM_1 Console 70. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 71. Check that traffic is available 72. Return to VM_2 console 73. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 74. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 009: Configure VPNaaS with non-default parameters: IKE/IPSec with 'aes256' encryption algorithm**
Test Case ID	configure_vpn_ike-aes256_ipsec-aes256
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using it's floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-256 test_ile_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-256 test_ipsec_policy

15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet
16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key
17. Check that test_connection has been created: neutron ipsec-site-connection-list
18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-256 2
19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-256 2
20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc
21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key
22. Check that test_connection has been created: neutron ipsec-site-connection-list
23. Return to tab with VM_1
24. Try to ping VM_2: ping <VM_2_private_IP>
25. Check that icmp is allowed
26. Return to tab with VM_2
27. Try to ping VM_1: ping <VM_1_private_IP>
28. Check that icmp is allowed

Dashboard:

1. Login to OpenStack Horizon dashboard
2. Navigate to Project -> Network -> Network Topology
3. Click on VM_1 and then on Open Console
4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>)
5. Check that traffic isn't allowed
6. Go to test_tenant
7. Navigate to Project -> Network -> Network Topology
8. Click on VM_2 and then on Open Console
9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>)

	10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-256 and Click on Add 17. Navigate to tab IPSec Policies 18. Click on Add IPSec Policy 19. Type name test_ipsec_policy 20. Choose encryption algorithm aes-256 21. Click on Add 22. Navigate to tab VPN Service 23. Click on Add VPN Service 24. Type name test_service 25. Select router 26. Select private subnet for this route 27. Click on Add 28. Navigate to IPSec Site Connections 29. Click on Add IPSec Site Connections 30. Type name test_connection 31. Select VPN Service test_service 32. Select IKE Policy test_ike_policy 33. Select IPSec Policy test_ipsec_policy 34. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 35. Type Remote peer subnet(s) as private network in the second tenant 36. Type Pre-Shared Key (PSK) string as key 37. Click on Add 38. Go to test_tenant 39. Navigate to Project -> Network -> VPN 40. Navigate to tab IKE Policies 41. Click on Add IPSec Policy 42. Type name test_ike_policy 43. Choose encryption algorithm aes-256 and Click on Add 44. Navigate to tab IPSec Policies 45. Click on Add IPSec Policy 46. Type name test_ipsec_policy 47. Choose encryption algorithm aes-256 48. Click on Add 49. Navigate to tab VPN Service 50. Click on Add VPN Service 51. Type name test_service 52. Select router

	53. Select private subnet for this route 54. Click on Add 55. Navigate to IPSec Site Connections 56. Click on Add IPSec Site Connections 57. Type name test_connection 58. Select VPN Service test_service 59. Select IKE Policy test_ike_policy 60. Select IPSec Policy test_ipsec_policy 61. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 62. Type Remote peer subnet(s) as private network in admin tenant 63. Type Pre-Shared Key (PSK) string as key 64. Click on Add 65. Reopen the page and check that the status of IPSec Site Connection is Active 66. Return to VM_1 Console 67. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 68. Check that traffic is available 69. Return to VM_2 console 70. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 71. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 010: Configure VPNaaS with non-default parameters: IKE/IPSec with '3des' encryption algorithm**
Test Case ID	configure_vpn_ike-3des_ipsec-3des
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed

	9. Create new tab 10. Go with ssh to the VM_2 using floating ip 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm 3des test_ile_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm 3des test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm 3des 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm 3des 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard

	2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm 3des and Click on Add 17. Navigate to tab IPSec Policies 18. Click on Add IPSec Policy 19. Type name test_ipsec_policy 20. Choose encryption algorithm 3des 21. Click on Add 22. Navigate to tab VPN Service 23. Click on Add VPN Service 24. Type name test_service 25. Select router 26. Select private subnet for this route 27. Click on Add 28. Navigate to IPSec Site Connections 29. Click on Add IPSec Site Connections 30. Type name test_connection 31. Select VPN Service test_service 32. Select IKE Policy test_ike_policy 33. Select IPSec Policy test_ipsec_policy 34. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 35. Type Remote peer subnet(s) as private network in the second tenant 36. Type Pre-Shared Key (PSK) string as key 37. Click on Add 38. Go to test_tenant 39. Navigate to Project -> Network -> VPN 40. Navigate to tab IKE Policies 41. Click on Add IPSec Policy 42. Type name test_ike_policy 43. Choose encryption algorithm 3des and Click on Add 44. Navigate to tab IPSec Policies

	45. Click on Add IPSec Policy 46. Type name test_ipsec_policy 47. Choose encryption algorithm 3des 48. Click on Add 49. Navigate to tab VPN Service 50. Click on Add VPN Service 51. Type name test_service 52. Select router 53. Select private subnet for this route 54. Click on Add 55. Navigate to IPSec Site Connections 56. Click on Add IPSec Site Connections 57. Type name test_connection 58. Select VPN Service test_service 59. Select IKE Policy test_ike_policy 60. Select IPSec Policy test_ipsec_policy 61. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 62. Type Remote peer subnet(s) as private network in admin tenant 63. Type Pre-Shared Key (PSK) string as key 64. Click on Add 65. Reopen the page and check that the status of IPSec Site Connection is Active 66. Return to VM_1 Console 67. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 68. Check that traffic is available 69. Return to VM_2 console 70. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 71. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 011: Configure VPNaaS with non-default parameters between two clouds: IKE Policy with Perfect Forward Secrecy "group2" and with encryption algorithm 'aes192' and IPSec Policy with encryption algorithm 'aes128'**
Test Case ID	configure_vpn_ike-group2-aes192_ipsec-aes128
Prerequisites	Creation non-admin tenant and launching instances

Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using it's floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-192 --pfs group2 test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-128 test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 --pfs group2 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-128 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list

	23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 29. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-192 and Perfect forward group group 2 17. Click on Add 18. Navigate to tab IPSec Policies 19. Click on Add IPSec Policy 20. Type name test_ipsec_policy 21. Choose encryption algorithm aes-128 22. Click on Add 23. Navigate to tab VPN Service 24. Click on Add VPN Service 25. Type name test_service 26. Select router 27. Select private subnet for this route 28. Click on Add 29. Navigate to IPSec Site Connections 30. Click on Add IPSec Site Connections 31. Type name test_connection 32. Select VPN Service test_service 33. Select IKE Policy test_ike_policy 34. Select IPSec Policy test_ipsec_policy 35. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network

	36. Type Remote peer subnet(s) as private network in the second tenant 37. Type Pre-Shared Key (PSK) string as key 38. Click on Add 39. Go to test_tenant 40. Navigate to Project -> Network -> VPN 41. Navigate to tab IKE Policies 42. Click on Add IPSec Policy 43. Type name test_ike_policy 44. Choose encryption algorithm aes-192 and Perfect forward group as group 2 45. Click on Add 46. Navigate to tab IPSec Policies 47. Click on Add IPSec Policy 48. Type name test_ipsec_policy 49. Choose encryption algorithm aes-128 50. Click on Add 51. Navigate to tab VPN Service 52. Click on Add VPN Service 53. Type name test_service 54. Select router 55. Select private subnet for this route 56. Click on Add 57. Navigate to IPSec Site Connections 58. Click on Add IPSec Site Connections 59. Type name test_connection 60. Select VPN Service test_service 61. Select IKE Policy test_ike_policy 62. Select IPSec Policy test_ipsec_policy 63. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 64. Type Remote peer subnet(s) as private network in admin tenant 65. Type Pre-Shared Key (PSK) string as key 66. Click on Add 67. Reopen the page and check that the status of IPSec Site Connection is Active 68. Return to VM_1 Console 69. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 70. Check that traffic is available 71. Return to VM_2 console 72. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 73. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 012: Configure VPNaaS with non-default parameters: IKE Policy with encryption algorithm 'aes128' and IKE version 'v2', IPSec Policy with encryption algorithm 'aes192'**
Test Case ID	configure_vpn_ike-aes128_ipsec-aes192-v2
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-128 --ike-version v2 test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-192 test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-128 --ike-version v2 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 2

	20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-128 17. Choose IKE version v2 18. Click on Add 19. Navigate to tab IPSec Policies 20. Click on Add IPSec Policy 21. Type name test_ipsec_policy 22. Choose encryption algorithm aes-192 23. Click on Add 24. Navigate to tab VPN Service 25. Click on Add VPN Service 26. Type name test_service

	27. Select router 28. Select private subnet for this route 29. Click on Add 30. Navigate to IPSec Site Connections 31. Click on Add IPSec Site Connections 32. Type name test_connection 33. Select VPN Service test_service 34. Select IKE Policy test_ike_policy 35. Select IPSec Policy test_ipsec_policy 36. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 37. Type Remote peer subnet(s) as private network in the second tenant 38. Type Pre-Shared Key (PSK) string as key 39. Click on Add 40. Go to test_tenant 41. Navigate to Project -> Network -> VPN 42. Navigate to tab IKE Policies 43. Click on Add IPSec Policy 44. Type name test_ike_policy 45. Choose encryption algorithm aes-128 46. Choose IKE version v2 47. Click on Add 48. Navigate to tab IPSec Policies 49. Click on Add IPSec Policy 50. Type name test_ipsec_policy 51. Choose encryption algorithm aes-192 52. Click on Add 53. Navigate to tab VPN Service 54. Click on Add VPN Service 55. Type name test_service 56. Select router 57. Select private subnet for this route 58. Click on Add 59. Navigate to IPSec Site Connections 60. Click on Add IPSec Site Connections 61. Type name test_connection 62. Select VPN Service test_service 63. Select IKE Policy test_ike_policy 64. Select IPSec Policy test_ipsec_policy 65. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 66. Type Remote peer subnet(s) as private network in admin tenant 67. Type Pre-Shared Key (PSK) string as key

	68. Click on Add 69. Reopen the page and check that the status of IPSec Site Connection is Active 70. Return to VM_1 Console 71. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 72. Check that traffic is available 73. Return to VM_2 console 74. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 75. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 013: Configure VPNaaS with non-default parameters: IKE Policy with encryption algorithm '3des', IPSec Policy with encryption algorithm 'aes192'**
Test Case ID	configure_vpn_ike-3des_ipsec-aes192
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm 3des test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-192 test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id

	<vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_id> --peer-id <router_from_test_tenant_id> --peer-cidr 10.1.1.0/24 --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm 3des 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy

	15. Type name test_ike_policy 16. Choose encryption algorithm 3des 17. Click on Add 18. Navigate to tab IPSec Policies 19. Click on Add IPSec Policy 20. Type name test_ipsec_policy 21. Choose encryption algorithm aes-192 22. Click on Add 23. Navigate to tab VPN Service 24. Click on Add VPN Service 25. Type name test_service 26. Select router 27. Select private subnet for this route 28. Click on Add 29. Navigate to IPSec Site Connections 30. Click on Add IPSec Site Connections 31. Type name test_connection 32. Select VPN Service test_service 33. Select IKE Policy test_ike_policy 34. Select IPSec Policy test_ipsec_policy 35. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 36. Type Remote peer subnet(s) as private network in the second tenant 37. Type Pre-Shared Key (PSK) string as key 38. Click on Add 39. Go to test_tenant 40. Navigate to Project -> Network -> VPN 41. Navigate to tab IKE Policies 42. Click on Add IPSec Policy 43. Type name test_ike_policy 44. Choose encryption algorithm 3des 45. Click on Add 46. Navigate to tab IPSec Policies 47. Click on Add IPSec Policy 48. Type name test_ipsec_policy 49. Choose encryption algorithm aes-192 50. Click on Add 51. Navigate to tab VPN Service 52. Click on Add VPN Service 53. Type name test_service 54. Select router 55. Select private subnet for this route 56. Click on Add 57. Navigate to IPSec Site Connections

	58. Click on Add IPSec Site Connections 59. Type name test_connection 60. Select VPN Service test_service 61. Select IKE Policy test_ike_policy 62. Select IPSec Policy test_ipsec_policy 63. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 64. Type Remote peer subnet(s) as private network in admin tenant 65. Type Pre-Shared Key (PSK) string as key 66. Click on Add 67. Reopen the page and check that the status of IPSec Site Connection is Active 68. Return to VM_1 Console 69. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 70. Check that traffic is available 71. Return to VM_2 console 72. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 73. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 014: Configure VPNaaS with non-default parameters: IKE Policy with encryption algorithm '3des', IPSec Policy with encryption algorithm 'aes192' and Transform Protocol ah**
Test Case ID	configure_vpn_ike-3des_ipsec-aes192-ah
Prerequisites	Creation non-admin tenant and launching instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using floating IP

	11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm 3des test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-192 --transform-protocol ah test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm 3des 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 --transform-protocol ah 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1 24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology

	3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-3des 17. Click on Add 18. Navigate to tab IPSec Policies 19. Click on Add IPSec Policy 20. Type name test_ipsec_policy 21. Choose encryption algorithm aes-192 22. Choose transform protocol ah 23. Click on Add 24. Navigate to tab VPN Service 25. Click on Add VPN Service 26. Type name test_service 27. Select router 28. Select private subnet for this route 29. Click on Add 30. Navigate to IPSec Site Connections 31. Click on Add IPSec Site Connections 32. Type name test_connection 33. Select VPN Service test_service 34. Select IKE Policy test_ike_policy 35. Select IPSec Policy test_ipsec_policy 36. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 37. Type Remote peer subnet(s) as private network in the second tenant 38. Type Pre-Shared Key (PSK) string as key 39. Click on Add 40. Go to test_tenant 41. Navigate to Project -> Network -> VPN 42. Navigate to tab IKE Policies 43. Click on Add IPSec Policy 44. Type name test_ike_policy 45. Choose encryption algorithm 3des

	46. Click on Add 47. Navigate to tab IPSec Policies 48. Click on Add IPSec Policy 49. Type name test_ipsec_policy 50. Choose encryption algorithm aes-192 51. Choose transform protocol ah 52. Click on Add 53. Navigate to tab VPN Service 54. Click on Add VPN Service 55. Type name test_service 56. Select router 57. Select private subnet for this route 58. Click on Add 59. Navigate to IPSec Site Connections 60. Click on Add IPSec Site Connections 61. Type name test_connection 62. Select VPN Service test_service 63. Select IKE Policy test_ike_policy 64. Select IPSec Policy test_ipsec_policy 65. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 66. Type Remote peer subnet(s) as private network in admin tenant 67. Type Pre-Shared Key (PSK) string as key 68. Click on Add 69. Reopen the page and check that the status of IPSec Site Connection is Active 70. Return to VM_1 Console 71. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 72. Check that traffic is available 73. Return to VM_2 console 74. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 75. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 015: Re-configure VPN connection between two tenants**
Test Case ID	reconfigure_vpn
Prerequisites	Creation non-admin tenant and launching instances

Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the fuel ip: ssh root@<fuel ip> 7. Look what nodes we have: fuel node 8. Go to the controller node: ssh node-<node_id> 9. ssh <user_name>@<VM_1_IP> 10. Try to ping VM_2: ping <VM_2_private_IP> 11. Check that icmp isn't allowed 12. Create new tab 13. Go with ssh to the fuel ip: ssh root@<fuel ip> 14. Look what nodes we have: fuel node 15. Go to the controller node: ssh node-<node_id> 16. ssh <user_name>@<VM_2_IP> 17. Try to ping VM_1: ping <VM_1_private_IP> 18. Check that icmp isn't allowed 19. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 20. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 21. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 22. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 23. Check that test_connection has been created: neutron ipsec-site-connection-list 24. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant_id <test_tenant_id> 2 25. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant_id <test_tenant_id> 2 26. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 27. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id

	<router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 28. Check that test_connection has been created: neutron ipsec-site-connection-list 29. Return to tab with VM_1 30. Try to ping VM_2: ping <VM_2_private_IP> 31. Check that icmp is allowed 32. Return to tab with VM_2 33. Try to ping VM_1: ping <VM_1_private_IP> 34. Check that icmp is allowed 35. Return to node's tab 36. Update Pre-Shared Key (PSK) string for the admin tenant: neutron ipsec-site-connection-update <ipsec_service_connection_id> --psk test 37. Update Pre-Shared Key (PSK) string for the test tenant: neutron ipsec-site-connection-update <ipsec_service_connection_id> --psk test 38. Return to tab with VM_1 39. Try to ping VM_2: ping <VM_2_private_IP> 40. Check that icmp is allowed 41. Return to tab with VM_2 42. Try to ping VM_1: ping <VM_1_private_IP> 43. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy and Click on Add 16. Navigate to tab IPSec Policies 17. Click on Add IPSec Policy 18. Type name test_ipsec_policy 19. Click on Add 20. Navigate to tab VPN Service 21. Click on Add VPN Service

	22. Type name test_service 23. Select router 24. Select private subnet for this route 25. Click on Add 26. Navigate to IPSec Site Connections 27. Click on Add IPSec Site Connections 28. Type name test_connection 29. Select VPN Service test_service 30. Select IKE Policy test_ike_policy 31. Select IPSec Policy test_ipsec_policy 32. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 33. Type Remote peer subnet(s) as private network in the second tenant 34. Type Pre-Shared Key (PSK) string as key 35. Click on Add 36. Reopen the page and check that the status of IPSec Site Connection is Active 37. Go to test_tenant 38. Navigate to Project -> Network -> VPN 39. Navigate to tab IKE Policies 40. Click on Add IPSec Policy 41. Type name test_ike_policy and Click on Add 42. Navigate to tab IPSec Policies 43. Click on Add IPSec Policy 44. Type name test_ipsec_policy 45. Click on Add 46. Navigate to tab VPN Service 47. Click on Add VPN Service 48. Type name test_service 49. Select router 50. Select private subnet for this route 51. Click on Add 52. Navigate to IPSec Site Connections 53. Click on Add IPSec Site Connections 54. Type name test_connection 55. Select VPN Service test_service 56. Select IKE Policy test_ike_policy 57. Select IPSec Policy test_ipsec_policy 58. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 59. Type Remote peer subnet(s) as private network in admin tenant 60. Type Pre-Shared Key (PSK) string as key 61. Click on Add

	62. Reopen the page and check that the status of IPSec Site Connection is Active 63. Return to VM_1 Console 64. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 65. Check that traffic is available 66. Return to VM_2 console 67. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 68. Check that traffic is available 69. Return to admin tenant 70. Navigate to Project -> Network -> VPN 71. Navigate to IPSec Site Connections 72. Click on Edit Connection 73. Type Pre-Shared Key (PSK) string as test instead of key 74. Click on Save Settings 75. Go to the test tenant 76. Navigate to Project -> Network -> VPN 77. Navigate to IPSec Site Connections 78. Click on Edit Connection 79. Type Pre-Shared Key (PSK) string as test instead of key 80. Click on Save Settings 81. Return to VM_1 Console 82. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 83. Check that traffic is available 84. Return to VM_2 console 85. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 86. Check that traffic is available 87. Check that IPSec Site Connection statuses are Active
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 016: Configure VPNaaS with non-default parameters: IKE Policy with encryption algorithm ‘aes256’ and IPSec Policy with encryption algorithm ‘aes128’ and with Perfect Forward Secrecy “group14”**
Test Case ID	configure_vpn_ike-aes256_ipsec-aes128-group14
Prerequisites	Creation non-admin tenant and launching instances
	CLI:

Steps	1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using it's floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-256 test_ile_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-128 --pfs group14 test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-256 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-128 -pfs group14 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Return to tab with VM_1

	24. Try to ping VM_2: ping <VM_2_private_IP> 25. Check that icmp is allowed 26. Return to tab with VM_2 27. Try to ping VM_1: ping <VM_1_private_IP> 28. Check that icmp is allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy 16. Choose encryption algorithm aes-256 17. Click on Add 18. Navigate to tab IPSec Policies 19. Click on Add IPSec Policy 20. Type name test_ipsec_policy 21. Choose encryption algorithm aes-128 22. Choose Perfect forward group group 14 23. Click on Add 24. Navigate to tab VPN Service 25. Click on Add VPN Service 26. Type name test_service 27. Select router 28. Select private subnet for this route 29. Click on Add 30. Navigate to IPSec Site Connections 31. Click on Add IPSec Site Connections 32. Type name test_connection 33. Select VPN Service test_service 34. Select IKE Policy test_ike_policy 35. Select IPSec Policy test_ipsec_policy 36. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network

	37. Type Remote peer subnet(s) as private network in the second tenant 38. Type Pre-Shared Key (PSK) string as key 39. Click on Add 40. Go to test_tenant 41. Navigate to Project -> Network -> VPN 42. Navigate to tab IKE Policies 43. Click on Add IPSec Policy 44. Type name test_ike_policy 45. Choose encryption algorithm aes-256 and Perfect forward group as group 2 46. Click on Add 47. Navigate to tab IPSec Policies 48. Click on Add IPSec Policy 49. Type name test_ipsec_policy 50. Choose encryption algorithm aes-128 51. Choose Perfect forward group group 14 52. Click on Add 53. Navigate to tab VPN Service 54. Click on Add VPN Service 55. Type name test_service 56. Select router 57. Select private subnet for this route 58. Click on Add 59. Navigate to IPSec Site Connections 60. Click on Add IPSec Site Connections 61. Type name test_connection 62. Select VPN Service test_service 63. Select IKE Policy test_ike_policy 64. Select IPSec Policy test_ipsec_policy 65. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 66. Type Remote peer subnet(s) as private network in admin tenant 67. Type Pre-Shared Key (PSK) string as key 68. Click on Add 69. Reopen the page and check that the status of IPSec Site Connection is Active 70. Return to VM_1 Console 71. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 72. Check that traffic is available 73. Return to VM_2 console 74. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 75. Check that traffic is available

Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 017: Configure VPNaaS with non-default parameters: IKE Policy with encryption algorithm 'aes192' and IPSec Policy with encryption algorithm 'aes256' and with Perfect Forward Secrecy "group2"**
Test Case ID	configure_vpn_ike-aes192_ipsec-aes256-group2
Prerequisites	Creation non-admin tenant and launching instances
Steps	1. CLI: 2. Go with ssh to the fuel ip: ssh root@<fuel ip> 3. Look what nodes we have: fuel node 4. Go to the controller node: ssh node-<node_id> 5. . openrc 6. Create new tab 7. Go with ssh to the VM_1 using it's floating IP 8. Try to ping VM_2: ping <VM_2_private_IP> 9. Check that icmp isn't allowed 10. Create new tab 11. Go with ssh to the VM_2 using it's floating IP 12. Try to ping VM_1: ping <VM_1_private_IP> 13. Check that icmp isn't allowed 14. Create new IKE Policy: neutron vpn-ikepolicy-create --encryption-algorithm aes-192 test_ile_policy 15. Create IPSec Policy: neutron vpn-ipsecpolicy-create --encryption-algorithm aes-256 --pfs group2 test_ipsec_policy 16. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 17. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 18. Check that test_connection has been created: neutron ipsec-site-connection-list 19. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-192 2

	20. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant-id <test_tenant_id> --encryption-algorithm aes-256 -pfs group2 2 21. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 22. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 23. Check that test_connection has been created: neutron ipsec-site-connection-list 24. Return to tab with VM_1 25. Try to ping VM_2: ping <VM_2_private_IP> 26. Check that icmp is allowed 27. Return to tab with VM_2 28. Try to ping VM_1: ping <VM_1_private_IP> 29. Check that icmp is allowed
	1. Dashboard: 2. Login to OpenStack Horizon dashboard 3. Navigate to Project -> Network -> Network Topology 4. Click on VM_1 and then on Open Console 5. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 6. Check that traffic isn't allowed 7. Go to test_tenant 8. Navigate to Project -> Network -> Network Topology 9. Click on VM_2 and then on Open Console 10. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 11. Check that traffic isn't allowed 12. Return to the admin tenant 13. Navigate to Project -> Network -> VPN 14. Navigate to tab IKE Policies 15. Click on Add IPSec Policy 16. Type name test_ike_policy 17. Choose encryption algorithm aes-192 18. Click on Add 19. Navigate to tab IPSec Policies 20. Click on Add IPSec Policy 21. Type name test_ipsec_policy 22. Choose encryption algorithm aes-256 23. Choose Perfect forward group group 2 24. Click on Add

	25. Navigate to tab VPN Service 26. Click on Add VPN Service 27. Type name test_service 28. Select router 29. Select private subnet for this route 30. Click on Add 31. Navigate to IPSec Site Connections 32. Click on Add IPSec Site Connections 33. Type name test_connection 34. Select VPN Service test_service 35. Select IKE Policy test_ike_policy 36. Select IPSec Policy test_ipsec_policy 37. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 38. Type Remote peer subnet(s) as private network in the second tenant 39. Type Pre-Shared Key (PSK) string as key 40. Click on Add 41. Go to test_tenant 42. Navigate to Project -> Network -> VPN 43. Navigate to tab IKE Policies 44. Click on Add IPSec Policy 45. Type name test_ike_policy 46. Choose encryption algorithm aes-192 47. Click on Add 48. Navigate to tab IPSec Policies 49. Click on Add IPSec Policy 50. Type name test_ipsec_policy 51. Choose encryption algorithm aes-256 52. Choose Perfect forward group group 2 53. Click on Add 54. Navigate to tab VPN Service 55. Click on Add VPN Service 56. Type name test_service 57. Select router 58. Select private subnet for this route 59. Click on Add 60. Navigate to IPSec Site Connections 61. Click on Add IPSec Site Connections 62. Type name test_connection 63. Select VPN Service test_service 64. Select IKE Policy test_ike_policy 65. Select IPSec Policy test_ipsec_policy

	66. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 67. Type Remote peer subnet(s) as private network in admin tenant 68. Type Pre-Shared Key (PSK) string as key 69. Click on Add 70. Reopen the page and check that the status of IPSec Site Connection is Active 71. Return to VM_1 Console 72. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 73. Check that traffic is available 74. Return to VM_2 console 75. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 76. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 018: Negative: Configure VPNaaS with different Pre-Shared Key (PSK) string**
Test Case ID	configure_vpn_with_different_key
Prerequisites	Creation non-admin tenant and launching ubuntu instances
Steps	CLI: 1. Go with ssh to the fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using it's floating IP 7. Try to ping VM_2: ping <VM_2_private_IP> 8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy

	15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Create new IKE Policy for the second tenant: neutron vpn-ikepolicy-create --tenant_id <test_tenant_id> 2 19. Create IPSec Policy for the second tenant: neutron vpn-ipsecpolicy-create --tenant_id <test_tenant_id> 2 20. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 21. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk test --tenant_id <test_tenant_id> 22. Check that test_connection has been created: neutron ipsec-site-connection-list 23. Check that status is down: neutron ipsec-site-connection-show <ipsec_site_connection_id> 24. Return to tab with VM_1 25. Try to ping VM_2: ping <VM_2_private_IP> 26. Check that icmp isn't allowed 27. Return to tab with VM_2 28. Try to ping VM_1: ping <VM_1_private_IP> 29. Check that icmp isn't allowed
	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console

	9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy and Click on Add 16. Navigate to tab IPSec Policies 17. Click on Add IPSec Policy 18. Type name test_ipsec_policy 19. Click on Add 20. Navigate to tab VPN Service 21. Click on Add VPN Service 22. Type name test_service 23. Select router 24. Select private subnet for this route 25. Click on Add 26. Navigate to IPSec Site Connections 27. Click on Add IPSec Site Connections 28. Type name test_connection 29. Select VPN Service test_service 30. Select IKE Policy test_ike_policy 31. Select IPSec Policy test_ipsec_policy 32. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network 33. Type Remote peer subnet(s) as private network in the second tenant 34. Type Pre-Shared Key (PSK) string as key 35. Click on Add 36. Reopen the page and check that the status of IPSec Site Connection is Active 37. Go to test_tenant 38. Navigate to Project -> Network -> VPN 39. Navigate to tab IKE Policies 40. Click on Add IPSec Policy 41. Type name test_ike_policy and Click on Add 42. Navigate to tab IPSec Policies 43. Click on Add IPSec Policy 44. Type name test_ipsec_policy 45. Click on Add 46. Navigate to tab VPN Service 47. Click on Add VPN Service 48. Type name test_service 49. Select router 50. Select private subnet for this route

	51. Click on Add 52. Navigate to IPSec Site Connections 53. Click on Add IPSec Site Connections 54. Type name test_connection 55. Select VPN Service test_service 56. Select IKE Policy test_ike_policy 57. Select IPSec Policy test_ipsec_policy 58. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 59. Type Remote peer subnet(s) as private network in admin tenant 60. Type Pre-Shared Key (PSK) string as test 61. Click on Add 62. Reopen the page and check that the status of IPSec Site Connection is Active 63. Return to VM_1 Console 64. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 65. Check that traffic isn't available 66. Return to VM_2 console 67. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 68. Check that traffic isn't available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 019: Configure VPNaaS with default parameters between ubuntu and centos cloud as admin user**
Test Case ID	configure_default_vpn_between_clouds
Prerequisites	1 pre-deployed ubuntu+neutron gre cloud and 1 pre-deployed cenOS+neutron vlan cloud with one launched instance on internal network in each tenant
Steps	CLI: 1. Go with ssh to the first fuel ip: ssh root@<fuel ip> 2. Look what nodes we have: fuel node 3. Go to the controller node: ssh node-<node_id> 4. . openrc 5. Create new tab 6. Go with ssh to the VM_1 using floating ip 7. Try to ping VM_2: ping <VM_2_private_IP>

	8. Check that icmp isn't allowed 9. Create new tab 10. Go with ssh to the VM_2 using it's floating IP 11. Try to ping VM_1: ping <VM_1_private_IP> 12. Check that icmp isn't allowed 13. Create new IKE Policy: neutron vpn-ikepolicy-create test_ike_policy 14. Create IPSec Policy: neutron vpn-ipsecpolicy-create test_ipsec_policy 15. Create VPN Service: neutron vpn-service-create --name test_service router04 net04__subnet 16. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_test_tenant_external_gateaway_ip> --peer-id <router_from_test_tenant_external_gateaway_ip> --peer-cidr <private_net_in_test_tenant_address> --psk key 17. Check that test_connection has been created: neutron ipsec-site-connection-list 18. Go with ssh to the fuel ip second tenant: ssh root@<fuel ip> 19. Look what nodes we have: fuel node 20. Go to the controller node: ssh node-<node_id> 21. . openrc 22. Create new IKE Policy: neutron vpn-ikepolicy-create --tenant_id <test_tenant_id> 2 23. Create IPSec Policy: neutron vpn-ipsecpolicy-create --tenant_id <test_tenant_id> 2 24. Create VPN Service for the test_tenant: neutron vpn-service-create --name 2 --tenant_id <test_tenant_id> 1 0c8327d5-be16-423d-aa45-77563b67e8fc 25. Create IPSec Service Connection: neutron ipsec-site-connection-create --name test_connection --vpnservice-id <vpn_service_id> --ikepolicy-id <ike_policy_id> --ipsecpolicy-id <ipsec_policy_id> --peer-address <router_from_admin_tenant_external_gateaway_ip> --peer-id <router_from_admin_tenant_external_gateaway_ip> --peer-cidr <private_net_in_admin_tenant_address> --psk key 26. Check that test_connection has been created and status is active: neutron ipsec-site-connection-list 27. Return to tab with VM_1 28. Try to ping VM_2: ping <VM_2_private_IP> 29. Check that icmp is allowed 30. Return to tab with VM_2 31. Try to ping VM_1: ping <VM_1_private_IP> 32. Check that icmp is allowed

	Dashboard: 1. Login to OpenStack Horizon dashboard 2. Navigate to Project -> Network -> Network Topology 3. Click on VM_1 and then on Open Console 4. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 5. Check that traffic isn't allowed 6. Go to test_tenant 7. Navigate to Project -> Network -> Network Topology 8. Click on VM_2 and then on Open Console 9. Try to send icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic isn't allowed 11. Return to the admin tenant 12. Navigate to Project -> Network -> VPN 13. Navigate to tab IKE Policies 14. Click on Add IPSec Policy 15. Type name test_ike_policy and Click on Add 16. Navigate to tab IPSec Policies 17. Click on Add IPSec Policy 18. Type name test_ipsec_policy 19. Click on Add 20. Navigate to tab VPN Service 21. Click on Add VPN Service 22. Type name test_service 23. Select router 24. Select private subnet for this route 25. Click on Add 26. Navigate to IPSec Site Connections 27. Click on Add IPSec Site Connections 28. Type name test_connection 29. Select VPN Service test_service 30. Select IKE Policy test_ike_policy 31. Select IPSec Policy test_ipsec_policy 32. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the second tenant in external network 33. Type Remote peer subnet(s) as private network in the second tenant 34. Type Pre-Shared Key (PSK) string as key 35. Click on Add 36. Go to test_tenant 37. Navigate to Project -> Network -> VPN 38. Navigate to tab IKE Policies 39. Click on Add IPSec Policy 40. Type name test_ike_policy and Click on Add 41. Navigate to tab IPSec Policies 42. Click on Add IPSec Policy

	43. Type name test_ipsec_policy 44. Click on Add 45. Navigate to tab VPN Service 46. Click on Add VPN Service 47. Type name test_service 48. Select router 49. Select private subnet for this route 50. Click on Add 51. Navigate to IPSec Site Connections 52. Click on Add IPSec Site Connections 53. Type name test_connection 54. Select VPN Service test_service 55. Select IKE Policy test_ike_policy 56. Select IPSec Policy test_ipsec_policy 57. Type Peer gateway public IPv4/IPv6 Address or FQDN and Peer router identity for authentication (Peer ID) as router's ip in the admin tenant in external network: 58. Type Remote peer subnet(s) as private network in admin tenant 59. Type Pre-Shared Key (PSK) string as key 60. Click on Add 61. Reopen the page and check that the status of IPSec Site Connection is Active 62. Return to VM_1 Console 63. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 64. Check that traffic is available 65. Return to VM_2 console 66. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 67. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Destructive testing

Title	**TC 020: Configure VPNaaS with default parameters between two tenants as admin user and ban p_neutron-vpn-agent**
Test Case ID	destructive_ban_vpn_agent
Prerequisites	Creation non-admin tenant and launching instances

Steps	1. Configure VPN with default parameters between two tenants as in the case 5 2. Go in the CLI on the controller-node 3. Ban p_neutron-vpn-agent: pcs resource ban p_neutron-vpn-agent 4. Wait 30 sec 5. Return to VM_1 Console 6. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 7. Check that traffic is available 8. Return to VM_2 console 9. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 021: Configure VPNaaS with default parameters between two tenants as admin user and destroy(shut down) primary controller**
Test Case ID	destructive_destroy_primary_controller
Prerequisites	Creation non-admin tenant and launching instances
Steps	1. Configure VPN with default parameters between two tenants as in the case 5 2. Find primary controller with command on the controllers: hiera role 3. Destroy this controller from lab (choose variant which your lab supports) a) virsh destroy <node> b) VBoxManage controlvm fuel-slave-2 poweroff 4. Wait 30 sec 5. Return to VM_1 Console 6. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 7. Check that traffic is available 8. Return to VM_2 console 9. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

Title	**TC 022: Configure VPNaaS with default parameters between two tenants as admin user and reset primary controller**
Test Case ID	destructive_reset_primary_controller
Prerequisites	Creation non-admin tenant and launching instances
Steps	1. Configure VPN with default parameters between two tenants as in the case 5 2. Find primary controller with command on the controllers: hiera role 3. Reset this controller from lab (choose variant which your lab supports) a) virsh reset <node> b) VBoxManage controlvm fuel-slave-2 reset 4. Wait 30 sec 5. Return to VM_1 Console 6. Try to send icmp traffic to VM_2 from test_tenant (ping <vm_2_ip>) 7. Check that traffic is available 8. Return to VM_2 console 9. Try to spend icmp traffic to VM_1 from test_tenant (ping <vm_1_ip>) 10. Check that traffic is available
Expected Result	All steps should be passed, we should have the ability to send ICMP traffic between VMs in different tenants.

System testing

Title	**TC 023: Install plugin and deploy environment**
Test Case ID	install_plugin_deploy_env
Prerequisites	1 non deployed cloud
Steps	1. Upload plugin to the master node 2. Install plugin: fuel plugins --install vpnaas-plugin-<x.x.x>.rpm 3. Ensure that plugin is installed successfully using cli: fuel plugins 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 1 node with Compute role 6. Apply network settings 7. Run network verification 8. Deploy the cluster

	9. Check plugin health using cli: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 10. Run OSTF
Expected Result	Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled and worked as expected.

Title	**TC 024: Modifying env with enabled plugin (removing/adding controller nodes)**
Test Case ID	modify_env_with_plugin_remove_add_controller
Prerequisites	1 non deployed cloud
Steps	1. Upload plugin to the master node 2. Install plugin: fuel plugins --install vpnaas-plugin-<x.x.x>.rpm 3. Ensure that plugin is installed successfully using cli: fuel plugins 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 1 node with Compute role 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 10. Run OSTF 11. Remove 1 nodes with Controller role /*remove node, where plugin's services available, to ensure that according to ha mode all plugins resources will be replaced and available on another live node and continue to work as expected*/ 12. Re-deploy cluster 13. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 14. Run OSTF 15. Add 1 new node with Controller role 16. Re-deploy cluster

	17. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 18. Run OSTF
Expected Result	Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled after migration in ha mode and worked as expected after modifying of environment.

Title	**TC 025: Modifying env with enabled plugin (removing/adding compute node)**
Test Case ID	modify_env_with_plugin_remove_add_compute
Prerequisites	1 non deployed cloud
Steps	1. Upload plugin to the master node 2. Install plugin: fuel plugins --install vpnaas-plugin-<x.x.x>.rpm 3. Ensure that plugin is installed successfully using cli: fuel plugins 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 2 nodes with compute roles 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 10. Run OSTF 11. Remove 1 compute node 12. Re-deploy cluster 13. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 14. Run OSTF 15. Add 1 compute node 16. Re-deploy cluster

	17. Check plugin health using cli on controller: pcs resource (p_neutron-vpn-agents are started, p_neutron-l3-agents are stopped) 18. Run OSTF
Expected Result	Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled and worked as expected after modifying of environment.

Title	**TC 023: Uninstall of plugin**
Test Case ID	positive_uninstall_plugin
Prerequisites	1 non deployed cloud
Steps	1. Install plugin: fuel plugins --install vpnaas-plugin-<x.x.x>.rpm 2. Check that it was successfully installed: fuel plugins 3. Remove plugin: fuel plugins --remove vpnaas-plugin==<version> 4. Check that it was successfully removed: fuel plugins
Expected Result	Plugin was installed and then removed successfully

Title	**TC 024: Negative: Uninstall of plugin with deployed env**
Test Case ID	negative_uninstall_plugin
Prerequisites	1 non deployed cloud
Steps	1. Install plugin: fuel plugins --install vpnaas-plugin-<x.x.x>.rpm 2. Deploy env with this plugin 3. Run OSTF 4. Try to delete plugin and ensure that present in cli alert: "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)" 5. Remove env 6. Remove plugin: fuel plugins --remove vpnaas-plugin==<version> 7. Check that it was successfully removed: fuel plugins

Expected Result	Plugin was installed successfully. Alert is present when we trying to delete plugin which is attached to enabled environment. When environment was removed, plugin is removed successfully too.

Appendix

№	**Resource title**
1	Guide to the VPNaaS Fuel Plugin
2	Creation non-admin tenant, network and launching instances