Tài liệu viết dựa trên bản thiết kế dành cho VNPaaS phiên bản đầu tiên ở bản phát hành Havana. Trong các bản cập nhập tiếp theo sẽ thêm tính năng và sửa một số bug, sẽ được cập nhập ở tài liệu.

Ở phiên bản đầu tiên, nhà phát hành thực hiện IKE với chế độ xác thực PSK thay vì sử dụng các chứng chỉ. Trong tương lai ở các bản cập nhập khác sẽ hỗ trợ xác thực dựa trên chứng chỉ.

**VPNaaS update:**

Bắt đầu từ phiên bản Havana:

**Liberty:** Start using reno to manage release notes.

**Mitaka:** Generation of sample Neutron VPNaaS configuration files.

### New Features

* Neutron VPNaaS no longer includes static example configuration files. Instead, use tools/generate\_config\_file\_samples.sh to generate them. The files are generated with a .sample extension.

**Newton:** Neutron VPNaaS is integrated with Guru Meditation Reports library.

### New Features

* Neutron VPNaaS services should respond to SIGUSR2 signal by dumping valuable debug information to standard error output.

**Pike:** Flavor framework integration.

### New Features

* Neutron VPNaaS is now integrated with Neutron flavor framework. Multiple VPN service providers might be configured at the same time. A flavor of service type VPN associated with a profile containing a driver is used to find the provider for a newly created VPN service.

### **Queens**: New Features

* VPNaaS support in neutron L3 agent is now implemented as L3 agent extension. We no longer have a separate binary neutron-vpn-agent. To enable VPNaaS support in L3 agent, ensure to specify vpnaas to the configuration extensions in [AGENT] section of the L3 agent config file.

### Upgrade Notes

* The separate L3 agent binary neutron-vpn-agent for VPNaaS support is dropped and VPNaaS support is now implemented as L3 agent extension. When upgrading your deployment to Queens, ensure to specify vpnaas to [AGENT] extensions configuration of the L3 agent config file and run neutron-l3-agent instead of neutron-vpn-agent.

### Bug Fixes

* The libreswan driver of neutron-vpnaas can now also work with Libreswan 3.19+ (bug [#1711456](https://launchpad.net/bugs/1711456)).

**Rocky:** Enable sha384 and sha512 auth algorithms for \*Swan drivers

### New Features

* Users can now specify sha384 and sha512 as the auth algorithm for both IKE policy and IPsec policy, when using \*Swan IPsec drivers.

### Upgrade Notes

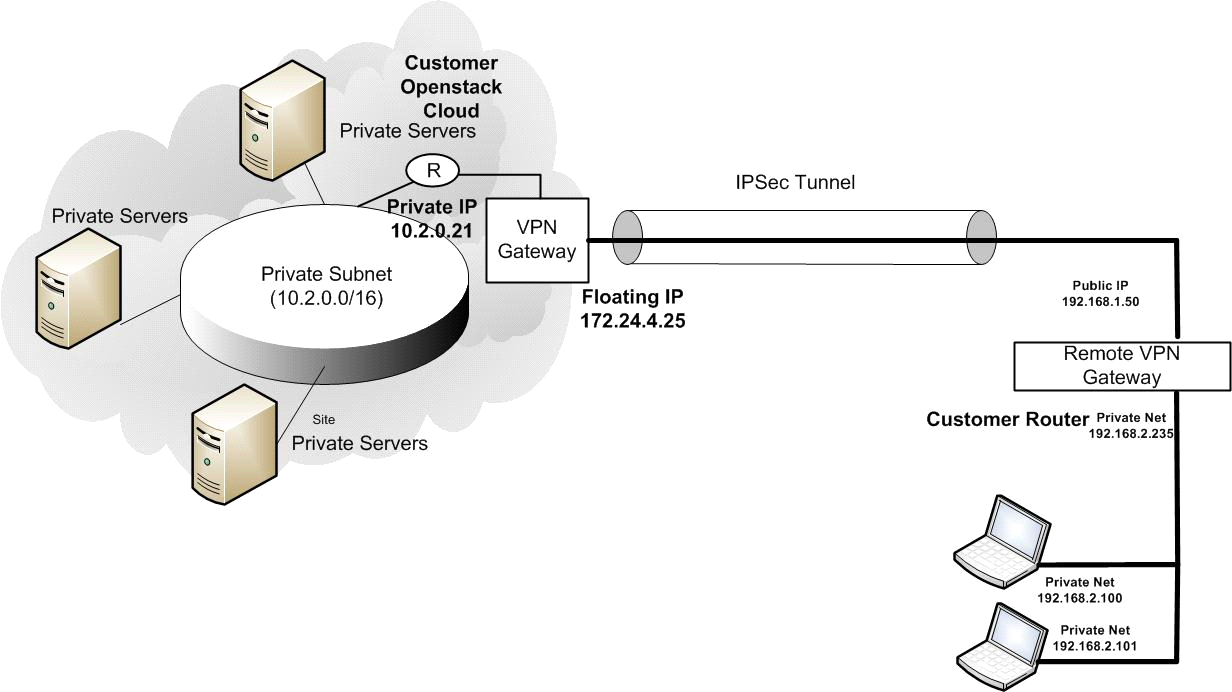
* The following drivers are removed due to the lack of maintainers of the drivers CiscoCsrIPsecDriver, FedoraStrongSwanDriver, VyattaIPsecDriver. Please refer the following [mailing list post](http://lists.openstack.org/pipermail/openstack-dev/2018-February/127793.html) for more detail.

### Bug Fixes

* The libreswan driver of neutron-vpnaas can now also work with Libreswan 3.19+ (bug [#1711456](https://launchpad.net/bugs/1711456)).

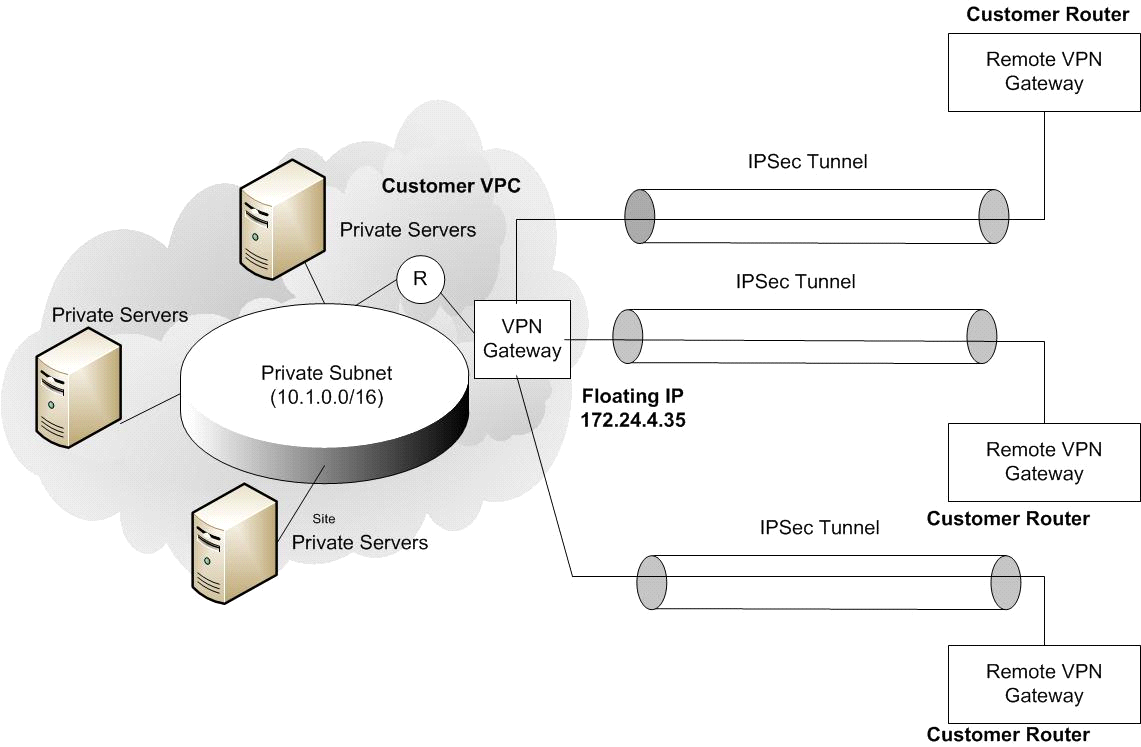
**IPsec Use Case**

**Single Connection**



*Trong trường hợp tạo một kết nối*

**Multiple Connections**



*Tạo nhiều kết nối VPN*

**Install:** <https://github.com/phamngocsonls/SVTT/blob/phamngocsonls/SONPN/NetWorking/VPNaaS_install.md>

**Setting:**

## vpn-ikepolicy

Là giao thức thực hiện quá trình trao đổi khóa và thỏa thuận các thông số bảo mật như: thuật toán mã hóa được áp dụng, khoảng thời gian khóa cần được thay đổi . Sau khi thỏa thuận xong thì sẽ thiết lập “hợp đồng” giữa 2 bên, khi đó IPSec SA (Security Association) được tạo ra. SA là những thông số bảo mật đã được thỏa thuận thành công, các thông số SA này sẽ được lưu trong cơ sở dữ liệu của SA Ngoài ra IKE còn dùng 2 giao thức khác để chứng thực đầu cuối và tạo khóa: ISAKMP (Internet Security Association and Key Management Protocol) và Oakley.

– ISAKMP: là giao thức thực hiện việc thiết lập, thỏa thuận và quản lý chính sách bảo mật SA.

– Oakley: là giao thức làm nhiệm vụ chứng thực khóa, bản chất là dùng thuật toán Diffie-Hellman để trao đổi khóa bí mật thông qua môi trường chưa bảo mật

Lưu ý: Giao thức IKE dùng UDP port 500.

VPNasaS hỗ trợ 2 loại IKE: IKE và IKEv2

Phiên bản IKEv1 được giới thiệu vào năm 1998 và bị thay thế bởi IKEv2 7 năm sau. Sự khác biệt lớn nhất giữa IKEv2 và IKEv1 là giảm băng thông.

Tóm tắt so sánh:

1.IKEv2 không tiêu thụ nhiều băng thông như IKEv1.   
2.IKEv2 hỗ trợ xác thực EAP trong khi IKEv1 thì không.   
3.IKEv2 hỗ trợ MOBIKE trong khi IKEv1 thì không.   
4.IKEv2 đã tích hợp NAT traversal trong khi IKEv1 thì không.   
5.IKEv2 có thể phát hiện xem một đường hầm có còn sống hay không trong khi IKEv1 không thể.  
6. IKEv2 an toàn và nhiều tính năng hơn, đáng tin cậy hơn.

Xem thêm so sánh tại: <http://rockhoppervpn.sourceforge.net/techdoc_ikev1vsikev2.html>

Phương thức xác thực. Với các phiên bản cũ IKEpolicy chỉ hỗ trợ SHA1, ở các phiên bản gần đây bổ sung thêm sha384 và sha512

Với tùy chỉnh mặc định : neutron vpn-ikepolicy-create <name>

neutron vpn-ikepolicy-create [-h] [-f {shell,table}] [-c COLUMN]

[--variable VARIABLE] [--prefix PREFIX]

[--request-format {json,xml}]

[--tenant-id TENANT\_ID]

[--description DESCRIPTION]

[--auth-algorithm AUTH-ALGORITHM]

[--encryption-algorithm ENCRYPTION-ALGORITHM]

[--phase1-negotiation-mode PHASE1-NEGOTIATION-MODE]

[--ike-version IKE-VERSION]

[--pfs PFS]

[--lifetime unit=UNITS,value=VALUE]

NAME

* **NAME**: Friendly name of the IKEPolicy used in IPsec VPN Service Connections
* **description**: Friendly description of the IKEPolicy used in IPsec VPN Service Connections
* **tenant-id**: ID of the Tenant that owns the VPN Service.
* **auth-algorithm**: Authentication algorithm used in the IKEPolicy.
* **encryption-algorithm**: Encryption algorithm used in the IKEPolicy.
* **phase1-negotiation-mode**: Phase1 negotiation mode for IKE either 'Main' or 'Aggressive'.
* **lifetime**: String with lifetime specific parameters example: --lifetime "units=seconds,value=3600"
* **units**: Units for lifetime ('seconds' or 'kilobytes')
* **value**. Value for lifetime (non-negative integer).
* **ike-version**: Specify the ike\_version.
* **pfs**: Specify the Perfect Forward Secrecy.

## vpn-ipsecpolicy

## vpn-ipsecpolicy hỗ trợ nhiều phương thức mã hóa mạnh: 3des, aes 128, aes 192, aes 256

Với tùy chỉnh mặc định : neutron vpn-ipsecpolicy-create <name>

neutron vpn-ipsecpolicy-create [-h] [-f {shell,table}] [-c COLUMN]

[--variable VARIABLE] [--prefix PREFIX]

[--request-format {json,xml}]

[--tenant-id TENANT\_ID]

[--description DESCRIPTION]

--transform-protocol TRANSFORM-PROTOCOL

[--auth-algorithm AUTH-ALGORITHM]

[--encryption-algorithm ENCRYPTION-ALGORITHM]

[--encapsulation-mode ENCAPSULATION-MODE]

[--pfs PFS]

[--lifetime units=UNITS,value=VALUE]

NAME

* **NAME**: Friendly name of the IPsecPolicy used in IPsec VPN Service Connections
* **description**: Friendly description of the IPsecPolicy used in IPsec VPN Service Connections
* **tenant-id**: ID of the Tenant that owns the VPN Service.
* **auth-algorithm**: Authentication algorithm used in the IPsecPolicy.
* **encryption-algorithm**: Encryption algorithm used in the IPsecPolicy.
* **encapsulation-mode**: Encapsulation mode for IPsec tunnel either 'tunnel' or 'transport'.
* **transfrom-protocol**: IPsec Transform Protocol either 'ESP' or 'AH'.
* **lifetime**: String with lifetime specific parameters example: --lifetime "units=seconds,value=3600"
* **units**: Units for lifetime ('seconds' or 'kilobytes')
* **value**. Value for lifetime (non-negative integer).
* **pfs**: Specify the Perfect Forward Secrecy.

## vpn-service-create

Create a new vpnservice

neutron vpn-service-create [-h] [-f {shell,table}] [-c COLUMN]

[--variable VARIABLE] [--prefix PREFIX]

[--request-format {json,xml}]

[--tenant-id TENANT\_ID]

[--admin-state-down] [--name NAME]

[--description DESCRIPTION]

ROUTER

SUBNET

* **tenant-id**: ID of the Tenant that owns the VPN Service.
* **router**: Unique identifier of the Router (either 'name' or 'id') to which the VPN will be attached to.
* **subnet**: Unique identifier of the Subnet (either 'name' or 'id') to which the VPN will provide service. (\*)

## ipsec-site-connection-create

Create a new ipsec-site-connection object

neutron ipsec-site-connection-create [-h] [-f {shell,table}]

[-c COLUMN]

[--variable VARIABLE]

[--prefix PREFIX]

[--request-format {json,xml}]

[--tenant-id TENANT\_ID]

[--admin-state-down] --name NAME

[--description DESCRIPTION]

--peer-address PEER-ADDRESS

--peer-id PEER-ID --peer\_cidr

PEER-CIDRS

[--mtu MTU]

[--psk PSK]

[--initiator INITIATOR]

[--dpd DPD]

--vpnservice-id VPNSERVICE

--ikepolicy-id IKEPOLICY

--ipsecpolicy-id IPSECPOLICY

* **peer-address**: Remote Peer IP Address for the VPN Connection.
* **tenant-id**: ID of the Tenant that owns the VPN Service.
* **peer-id**: Peer identifier string.
* **peer\_cidr**: Remote Peer Subnet with mask in CIDR format.
* **mtu**: MTU for fragmentation
* **dpd**: String with the dpd attributes. Example: --dpd "action=hold,interval=30,timeout=120"
* **action**: Dead peer detection actions (action=hold, restart etc.,).
* **interval**: Dead peer detection interval.(interval=30)
* **timeout**: Dead peer detection timeout.(timeout=120)
* **route-mode**: Routing mode either 'static' or 'dynamic' - for first release only 'static supported.
* **auth-mode**: Authentication mode either 'PSK' or 'CERTS'
* **psk**: Peer identifier string.
* **initiator**: Initiator mode either 'bi-directional' or 'responder'.
* **vpnservice-id**: Unique Identifier to the VPN Service Object.
* **ikepolicy-id**: Unique Identifier to the IKE Policy Object.
* **ipsecpolicy-id**: Unique Identifier to the IPsec Policy Object.

Use case, setting, install, advan/disadvantage, các case có thể sử dụng được, biện pháp để tăng tính an toàn, luồng triển khai thực tế với hệ thống