OpenStack Networking API v2.0 (neutron) Reference

API v2.0 (neutron) and extensions (2014-09-25)
Copyright © 2011-2014 OpenStack Foundation All rights reserved.

This document is for software developers who develop applications by using the OpenStack Networking API v2.0.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this fi e except in comp iance with the License. You may obtain a copy of the License at

http://www.apache.org/ icenses/LICENSE 2.0

Un ess required by app icab e aw or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or imp ied. See the License for the specific anguage governing permissions and imitations under the License.

Table of Contents

Preface iv
Intended audience iv
Document change history v
Resources vi
1. Overview 1
Glossary 2
Concepts 3
High-level task flow 10
Plug-ins 12
2. General API information 13
Authentication and authorization 13
Request and response formats 14
Filtering and column selection 14
Synchronous versus asynchronous plug-in behavior 15
Bulk-create 15
Pagination 16
Sorting 19
Extensions 20
Faults 21
3. API operations and extensions 22

Preface

Intended audience iv
Document change history v
Resources vi

The OpenStack Networking (neutron) project provides virtual networking services among devices managed by the OpenStack compute service.

This document describes the Networking (neutron) API v2.0 features.

We welcome feedback, comments, and bug reports at bugs.launchpad.net/Neutron.

Intended audience

This guide is for software developers who create applications by using the Networking (neutron) API v2.0. To use this information, you should have a general understanding of the OpenStack Networking service, the OpenStack compute service, and the integration between the two. You should also have access to a plug-in that implements the Networking (neutron) API v2.0.

You should also be familiar with:

- ReSTful web services
- HTTP/1.1
- JSON and XML data serialization formats

Document change history

This version of the document replaces and obsoletes all previous versions. The following table describes the most recent changes:

Revision Date	Summary of Changes
August 8, 2014	• Removed and rep aced WADL references with inks to *Networking API v2.0 (CURRENT)*.
June 29, 2014	• Added `binding:vnic_type`.
Apri 29, 2014	• Changed `binding:capabilities` to `binding:vif_details`.
December 20, 2013	• Updated book to source information from WADL fi es.
October 11, 2013	• Added the Networking API ports binding extension.
May 22, 2013	• Updated the tit e of the book and project name to "OpenStack Networking." • Updated the tit e of the book to *Reference* from *Developer Guide* for consistency.
March 23, 2013	• Updated incorrect nova command examp es in the section ca ed "High eve task f ow" [10].
September 5, 2012	• Removed XML as a va id request or response type.
August 17, 2012	• First edition of this book.

Resources

Use the following resources in conjunction with this guide:

Resource	See
Re ated documents	OpenStack Documentation
OpenStack Neutron Wiki	http://wiki.openstack.org/Neutron

1. Overview

Glossary 2
Concepts 3
High-level task flow 10
Plug-ins 12

The Neutron project provides virtual networking services among devices that are managed by the OpenStack compute service.

The Networking API v2.0 combines the Quantum API v1.1 with some essential Internet Protocol Address Management (IPAM) capabilities from the Melange API.

These IPAM capabilities enable you to:

- Associate IP address blocks and other network configuration settings required by a network device, such as a default gateway and dns-servers settings, with an OpenStack Networking network.
- Allocate an IP address from a block and associate it with a device that is attached to the network through an OpenStack Networking port.

To do this, the Networking API v2.0 introduces the subnet entity. A subnet can represent either an IP version 4 or version 6 address block. Each OpenStack Networking network commonly has one or more subnets. When you create a port on the network, an available fixed IP address is allocated to it from one the designated subnets for each IP version. When you delete the port, the allocated addresses return to the pool of available IPs on the subnet. Networking API v2.0 users can choose a specific IP address from the block or let OpenStack Networking choose the first available IP address.



Note

The Quantum API v1.x was removed from the source code tree. To use the Quantum API v1.x, install the Quantum Essex release.

Glossary

Term	Description
CRUD	Create, read, update, and de ete (CRUD) are the basic functions of persistent storage in com puter programming.
entity	Any piece of hardware or software that can connect to the network services provided by OpenStack Networking. An entity can use OpenStack Networking services by imp ementing a VIF.
IPAM	Internet Protoco Address Management (IPAM) is a means of p anning, tracking, and manag ing the Internet Protoco (IP) address space that is used in a network.
ayer 2 network	A virtua Ethernet network that is managed by the OpenStack Networking service. Current y, OpenStack Networking manages on y Ethernet networks.
network	An iso ated virtua ayer 2 broadcast domain that is typica y reserved for the tenant who creat ed it un ess the network is configured to be shared. Tenants can create mu tip e networks unti they reach the thresho ds specified by per tenant quotas.
p ug in	A software component that imp ements Networking API v2.0.
port	A virtua switch port on a ogica network switch. Virtua instances attach their interfaces into ports. The ogica port a so defines the MAC address and the IP addresses to be assigned to the interfaces p ugged into them. When creating a port, any una ocated IP in the subnet can be stated specifica y, even if the IP address is not in the a ocation poo . Enab ing users to specify the IP exp icit y a ows them to retain a particu ar subset of the subnet IPs for static a ocation.
subnet	An IP address b ock that can be used to assign IP addresses to virtua instances. Each subnet must have a CIDR and must be associated with a network. IPs can be either se ected from the who e subnet CIDR or from a ocation poo s that can be specified by the user.
VM	A virtua machine (VM) is a comp ete y iso ated guest operating system insta ation within a norma host operating system.

Concepts

Use the Networking API v2.0 to manage the following entities:

- **Network**. An isolated virtual layer-2 domain. A network can also be a virtual, or logical, switch. See the section called "Network" [4].
- **Subnet**. An IP version 4 or version 6 address block from which IP addresses that are assigned to VMs on a specified network are selected. See the section called "Subnet" [6].
- **Port**. A virtual, or logical, switch port on a specified network. See the section called "Port" [8].

These entities have auto-generated unique identifiers and support basic create, read, update, and delete (CRUD) functions with the **POST**, **GET**, **PUT**, and **DELETE** verbs.

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/378012137064006104