

OTC-VPC Subservices Terraform Provider and Gophercloud

Functional Spec

March 3, 2018

Sapan Vaswani

Email: sapan.vaswani@click2cloud.net

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description of Change** |
| 1.0 | 2018.03.07 | Sapan Vaswani | Initial Document Creation |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version Approved** | **Role** | **Date** |
| Sandeep Thakre |  | PM | 2018.03.07 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Introduction 4](#_Toc508725892)

[Objectives 4](#_Toc508725893)

[Project Scope 4](#_Toc508725894)

[Out of Scope 6](#_Toc508725895)

[Assumptions 6](#_Toc508725896)

[Architecture 7](#_Toc508725897)

[Architecture Overview for OTC-VPC-Subservices Gophercloud 7](#_Toc508725898)

[Component Structure 7](#_Toc508725899)

[Architecture Overview for OTC-VPC-Subservices Terraform provider 8](#_Toc508725900)

[Component Structure 8](#_Toc508725901)

[Prerequisite 9](#_Toc508725902)

[Operating Environment 9](#_Toc508725903)

[Software Used: 9](#_Toc508725904)

[Dependencies 9](#_Toc508725905)

[File Names and Structure 10](#_Toc508725906)

[Interfaces Definition – VPC Subservices 12](#_Toc508725907)

[Subnet: 12](#_Toc508725908)

[VPC Route: 12](#_Toc508725909)

[VPC Peering Connection: 13](#_Toc508725910)

[Disclaimer 14](#_Toc508725911)

[Appendix: Glossary 14](#_Toc508725912)

[Point of Contact 14](#_Toc508725913)

# Introduction

## Objectives

The objective of this document is to describe technical and functional specification on OTC-VPC Subservices i.e. Subnet, Route and VPC Peering Connection Terraform Provider and Gophercloud. This document covers scope and detailed on different parameters supported by interfaces. This deliverable will cover the implementation of OTC Subnet, Route and VPC Peering Connection interface in Gophercloud and relative terraform provider subservice for Virtual Private Cloud (VPC).

In a cloud environment, various users’ resources must be kept separate. This can be achieved by each user creating a Subnet, Route and VPC Peering Connection. A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another tenant. A subnet is a range of IP addresses in your VPC. You can launch resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet. A route is a mapping of an IP range to a destination. Routes tell the VPC network where to send packets destined for a particular IP address. By default, every network has routes that let instances in a network send traffic directly to each other, even across subnets.

In this project, our main goal is to automate provisioning of OTC-VPC service using terraform provider with support from gophercloud.

## Project Scope

Terraform is used to create, manage, and manipulate infrastructure resources. Examples of resources include physical machines, VMs, network switches, containers, etc. Almost any infrastructure noun can be represented as a resource in Terraform.

So as a part of project scope, Click2Cloud would develop and deliver following to automate the provisioning using Terraform Provider;

* 1 custom Gopher SDK for OTC Cloud Subnet Service Client with following API interactions.

|  |
| --- |
| **VPC Subnet** |
| Creating a Subnet |
| Querying Subnet Details |
| Querying Subnets |
| Updating Subnet Information |
| Deleting a Subnet |

|  |
| --- |
| **Terraform Data Sources** |
| otc\_vpc\_subnet (Query), otc\_vpc\_subnet\_ids |
| **Terraform Resources** |
| otc\_vpc\_subnet (Create, Update, Delete) |

* 2 Terraform Data Source and 1 Terraform Resource for OTC-Subnet service
* 1 custom Gopher SDK for OTC Cloud Route Service Client with following API interactions.

|  |
| --- |
| **VPC Route** |
| Creating a Route |
| Querying Route Details |
| Querying Route |
| Deleting a Route |

* 2 Terraform Data Source and 1 Terraform Resource for OTC-Route service

|  |
| --- |
| **Terraform Data Sources** |
| otc\_vpc\_route(Query), otc\_vpc\_route\_ids |
| **Terraform Resources** |
| otc\_vpc\_route(Create, Update, Delete) |

* 1 custom Gopher SDK for OTC Cloud Peering Connection Service Client with following API interactions.

|  |
| --- |
| **VPC Peering Connection** |
| Creating a VPC Peering Connection |
| Querying a VPC Peering Connection |
| Querying VPC Peering Connections |
| Updating a VPC Peering Connection |
| Deleting a VPC Peering Connection |
| Accepting a VPC Peering Connection |
| Refusing a VPC Peering Connection |

* 2 Terraform Data Source and 1 Terraform Resource for OTC- Peering Connection service

|  |
| --- |
| **Terraform Data Sources** |
| otc\_vpc\_peering\_connection |
| **Terraform Resources** |
| otc\_vpc\_peering\_connection (Create, Update, Delete), otc\_vpc\_peering\_connection\_accepter(Accept, Refuse) |

## Out of Scope

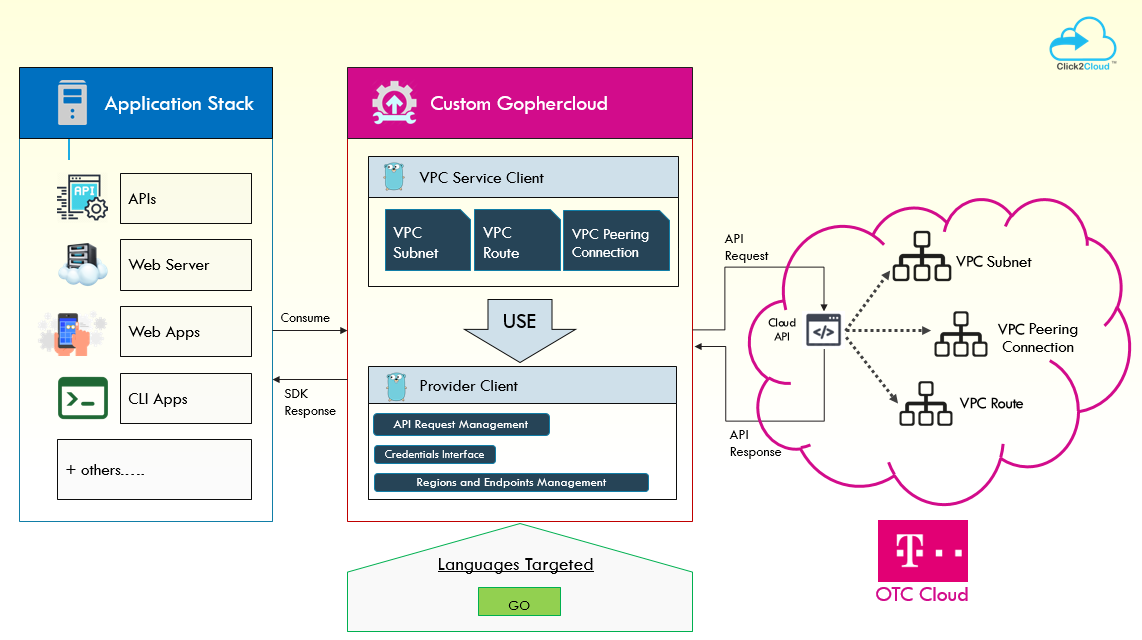
1. Other tasks than above task, for which development will be done, are not in the scope of project.
2. Gophercloud Authentication mechanism would not be customized as VPC subservice client is part of scope.
3. Resource and data source will be tested using open source terraform for windows. Scope does not cover testing on Terraform Enterprise.
4. Apart from English language, other languages such as Chinese is not considered.
5. Hashicorp approval is not a part of scope.

## Assumptions

1. The technical platform such as OTC authentication details, terraform and otc terraform provider and gophercloud is available.
2. Gophercloud Authentication mechanism will be used as is without any customization with username, password, tenant\_id, etc.
3. Change in any functional requirement documented below shall be treated as CR (Change Request).
4. This document to be freeze and sign-off before implementation start.
5. Tasks mentioned in the [Out of Scope](#_Out_of_Scope) are not part of requirement.
6. All interfaces which need to integrate with terraform provider, are finalized by Huawei Team under SOW.
7. Resource and data source will be tested using open source terraform for windows. Scope does not cover testing on Terraform Enterprise.
8. Any update within REST API should be informed to Click2Cloud Team. Click2Cloud will do required analysis and will identify the impact. If the impact is significant, then it will be communicated to stakeholders for further decision.
9. All the supporting document will be provided in English language only.

# Architecture

## Architecture Overview for OTC-VPC-Subservices Gophercloud

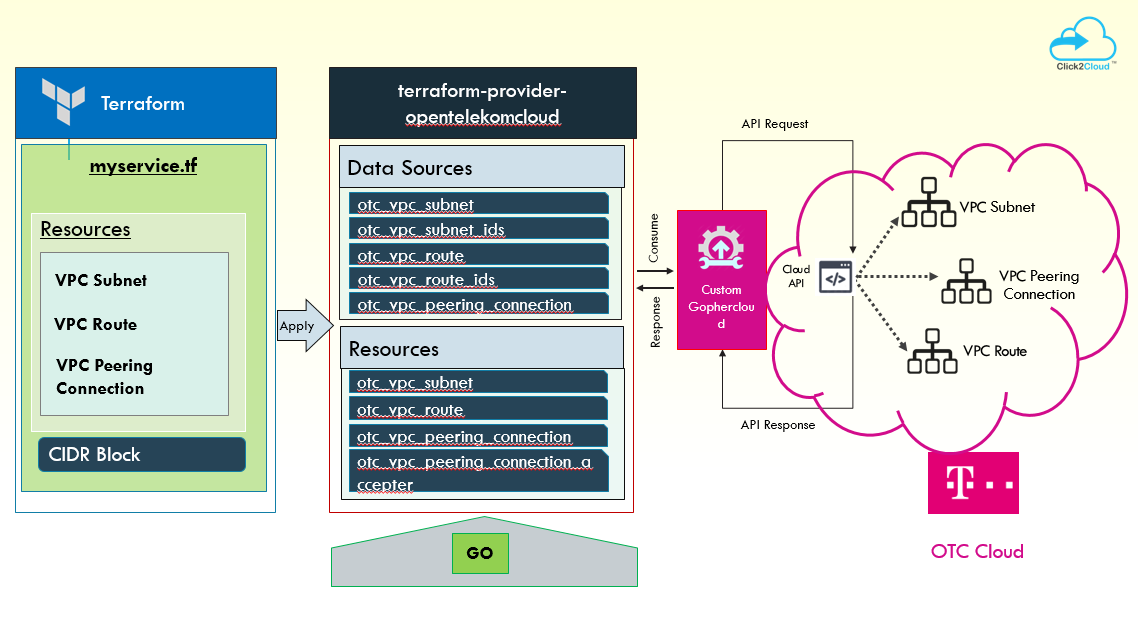


## Component Structure

As the image above:

1. Gophercloud is the OpenStack SDK for GoLang, which will handle the VPC subservices API interaction for OTC.
2. The service client will use provider client for authentication and API Request management.
3. It will be consumed by Terraform Provider to automate infrastructure provisioning.

## Architecture Overview for OTC-VPC-Subservices Terraform provider



## Component Structure

As the image above:

1. terraform-provider-opentelekomcloud is the extension of terraform to communicate with OTC services.
2. Gophercloud is the middle layer, the OpenStack SDK for GoLang, which will handle the VPC service API interaction for OTC.
3. The SDK will be consumed by Terraform Provider to automate infrastructure provisioning.

# Prerequisite

## Operating Environment

### Software Used:

**Gopher SDK** - go1.9.2 windows/amd64

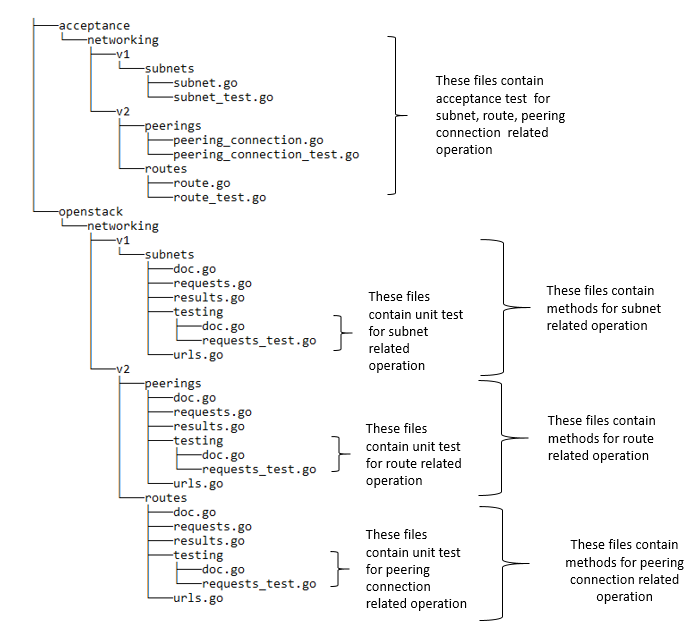
**Terraform** - Terraform v0.11.1

## Dependencies

* Access/Credentials to OTC Cloud environment to create, update, delete VPC Subservices.
* Forked/cloned repository of official Gopher cloud repository would be shared by client with admin/full access.
* Access to OTC Terraform provider repository where the data source and resource need to be developed
* The timelines specified are tentative and can change based on the priority of items to be delivered, delay in any approvals required from Huawei, infrastructure issues from Huawei.

# File Names and Structure

1. Gophercloud File Structure



1. OTC Terraform Provider File Structure



# Interfaces Definition – VPC Subservices

Following are the list of interfaces will be integrated with gopher cloud and terraform

## Subnet:

### Creating a Subnet

Create an subnet according to below parameters.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| subnet | Yes | *Dictionary data structure* | Specifies the subnet objects. A hash/dictionaries of subnet is given in [Description of Creating subnet fields](#_Description_of_) |

#### Description of Creating **subnet** fields

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| id | Yes | String | Specifies a resource  ID in UUID format. |
| name | Yes | String | Specifies the subnet  name. |
| cidr | Yes | String | Specifies the subnet network  segment. |
| gateway\_ip | Yes | String | Specifies the subnet gateway  address. |
| dhcp\_enable | No | Boolean | Specifies whether the DHCP function  is enabled for the subnet. |
| primary\_dns | No | String | Specifies the IP address of DNS  server 1 on the subnet. |
| secondary\_dns | No | String | Specifies the IP address of DNS  server 2 on the subnet. |
| dnsList | No | List | Specifies the IP address list of DNS  servers on the subnet. |
| availability\_zone | No | String | Identifies the AZ to which the  subnet belongs. |
| vpc\_id | Yes | String | Specifies the ID of the VPC to which  the subnet belongs. |
| status | Yes | String | Specifies the status of the subnet.  The value can be **ACTIVE**, **DOWN**, **UNKNOWN**, or **ERROR**. |

### Querying Subnet Details

This interface is used to query details about a subnet.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| subnet | Yes | *Dictionary data structure* | Specifies the subnet objects. A hash/dictionaries of subnet is given in [Description of Querying subnet fields](#_Description_of_Querying) |

#### Description of Querying **subnet** fields

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| id | Yes | String | Specifies a resource  ID in UUID format. |
| name | Yes | String | Specifies the subnet  name. |
| cidr | Yes | String | Specifies the subnet network  segment. |
| gateway\_ip | Yes | String | Specifies the subnet gateway  address. |
| dhcp\_enable | No | Boolean | Specifies whether the DHCP function  is enabled for the subnet. |
| primary\_dns | No | String | Specifies the IP address of DNS  server 1 on the subnet. |
| secondary\_dns | No | String | Specifies the IP address of DNS  server 2 on the subnet. |
| dnsList | No | List | Specifies the IP address list of DNS  servers on the subnet. |
| availability\_zone | No | String | Identifies the AZ to which the  subnet belongs. |
| vpc\_id | Yes | String | Specifies the ID of the VPC to which  the subnet belongs. |
| status | Yes | String | Specifies the status of the subnet.  The value can be **ACTIVE**, **DOWN**, **UNKNOWN**, or **ERROR**. |

### Querying Subnets

This interface is used to query subnets using search criteria and to display the subnets in a list.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| subnet | Yes | *Dictionary data structure* | Specifies the subnet objects. A hash/dictionaries of subnet is given in [Description of Querying subnet fields](#_Description_of_Querying_1) |

#### Description of Querying **subnet** fields

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| id | Yes | String | Specifies a resource  ID in UUID format. |
| name | Yes | String | Specifies the subnet  name. |
| cidr | Yes | String | Specifies the subnet network  segment. |
| gateway\_ip | Yes | String | Specifies the subnet gateway  address. |
| dhcp\_enable | No | Boolean | Specifies whether the DHCP function  is enabled for the subnet. |
| primary\_dns | No | String | Specifies the IP address of DNS  server 1 on the subnet. |
| secondary\_dns | No | String | Specifies the IP address of DNS  server 2 on the subnet. |
| dnsList | No | List | Specifies the IP address list of DNS  servers on the subnet. |
| availability\_zone | No | String | Identifies the AZ to which the  subnet belongs. |
| vpc\_id | Yes | String | Specifies the ID of the VPC to which  the subnet belongs. |
| status | Yes | String | Specifies the status of the subnet.  The value can be **ACTIVE**, **DOWN**, **UNKNOWN**, or **ERROR**. |

### Updating Subnet Information

This interface is used to update information about a subnet.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| subnet | Yes | *Dictionary data structure* | Specifies the subnet objects. A hash/dictionaries of subnet is given in Description of Updating Subnet Info |

#### Description of Updating Subnet Info

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| name | Yes | String | Specifies the subnet name.  The value is a string of 1 to 64 characters that can contain letters, digits, underscores (\_), and hyphens (-). |
| dhcp\_enable | No | Boolean | Specifies whether the DHCP function is enabled for the subnet.  The value can be **true** or **false**.  If this parameter is left blank, it is set to **true** by default. |
| primary\_dns | No | String | Specifies the IP address of DNS server 1 on the subnet.  The value must be a valid IP address. |
| secondary\_dns | No | String | Specifies the IP address of DNS server 2 on the subnet.  The value must be a valid IP address. |
| dnsList | No | List | Specifies the DNS server address list of a subnet. This field is required if you need to use more than two DNS servers.  This parameter value is the superset of both DNS server address 1 and DNS server address 2. |

### Deleting Subnet

This interface is used to delete a subnet.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| tenant\_id | Yes | String | Specifies the tenant ID  of the operator. |
| vpc\_id | Yes | String | Specifies the ID of the  subnet VPC. |
| subnet\_id | Yes | String | Specifies the subnet ID, which  uniquely identifies the subnet. |

## VPC Route:

### Creating a Route

This interface is used to create a route.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| route | Dict | Yes | Specifies the route object list. A hash/dictionaries of route is given in [Description of Creating Route fields](#_Description_of_Querying_1) |

#### Description of Creating Route fields

| **Attribute** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| id | String | Yes | Specifies the route ID. |
| destination | String | Yes | Specifies the destination IP address  or CIDR block. |
| nexthop | String | Yes | Specifies the next hop. If the route  type is **peering**, enter the VPC peering connection ID. |
| type | String | Yes | Specifies the route type. |
| vpc\_id | String | Yes | Specifies the VPC for which a  route is to be added. |
| tenant\_id | String | No | Specifies the tenant ID. Only the administrator  can specify the tenant ID of  other tenants. |

### Querying VPC Routes

This interface is used to query details about a VPC Route.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| routes | List(route) | Yes | Specifies the route object list. For details see [Description of Querying VPC Routes](#_Description_of_Querying_2) |

#### Description of Querying VPC Routes

| **Attribute** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| id | String | Yes | Specifies the route ID. |
| destination | String | Yes | Specifies the destination IP address  or CIDR block. |
| nexthop | String | Yes | Specifies the next hop. If the route  type is **peering**, enter the VPC peering connection ID. |
| type | String | Yes | Specifies the route type. |
| vpc\_id | String | Yes | Specifies the VPC for which a  route is to be added. |
| tenant\_id | String | No | Specifies the tenant ID. Only the administrator  can specify the tenant ID of  other tenants. |

### Querying a VPC Route

This interface is used to query Route using search criteria and to display the Roue in a list.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| routes | List(route) | Yes | Specifies the route object list. For details see [Description of Querying a VPC Routes](#_Description_of_Querying_3) |

#### Description of Querying a VPC Route

| **Attribute** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| id | String | Yes | Specifies the route ID. |
| destination | String | Yes | Specifies the destination IP address  or CIDR block. |
| nexthop | String | Yes | Specifies the next hop. If the route  type is **peering**, enter the VPC peering connection ID. |
| type | String | Yes | Specifies the route type. |
| vpc\_id | String | Yes | Specifies the VPC for which a  route is to be added. |
| tenant\_id | String | No | Specifies the tenant ID. Only the administrator  can specify the tenant ID of  other tenants. |

### Deleting a VPC Route

This interface is used to delete a Route.

| **Name** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| id | Yes | String | Deletes a route to which the specified tenant has access. |

## VPC Peering Connection:

Create, Update, Delete and Query a VPC Peering Connection according to below input and output parameters.

### Creating a VPC Peering Connection

Create an VPC Peering Connection according to below parameters.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peering | Dict | Yes | Specifies the VPC peering connection object list. If you create a VPC peering connection with a VPC of another tenant, you must specify **tenant\_id** of the peer tenant.  For details see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

#### Description of Creating a VPC Peering Connection

| **Attribute** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| id | String | Yes | Specifies the VPC peering connection ID. |
| name | String | Yes | Specifies the VPC peering connection  name. |
| status | String | No | Specifies the VPC peering connection status. The value can be **PENDING\_ACCEPTANCE**, **REJECTED**, **EXPIRED**,  **DELETED**, or **ACTIVE**. |
| request\_vpc\_info | Dict | Yes | Specifies information about the local VPC. A hash/dictionaries of route is given in [Local/Peer VPC](#_Local/Peer_VPC) |
| accept\_vpc\_info | Dict | Yes | Specifies information about the peer VPC.  A hash/dictionaries of route is given in [Local/Peer VPC](#_Local/Peer_VPC) |

#### Local/Peer VPC

| **Attribute** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| vpc\_id | String | Yes | Specifies the ID of a VPC involved in a VPC peering connection. |
| tenant\_id | String | No | Specifies the ID of the tenant to which a VPC involved in the VPC peering connection belongs. |

### Querying VPC Peering Connections

This interface is used to query Peering Connection using search criteria and to display the Peering Connection in a list.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peerings | List(peering) | Yes | Specifies the VPC peering connection object list. For details, see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

### Querying a VPC Peering Connection

This interface is used to query details about a VPC Peering Connection.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peering | Dict | Yes | Specifies the VPC peering connection object list. If you create a VPC peering connection with a VPC of another tenant, you must specify **tenant\_id** of the peer tenant.  For details see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

### Accepting a VPC Peering Connection

This interface is used by a tenant to accept a VPC peering connection request initiated by another tenant.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peering | Dict | Yes | Specifies the VPC peering connection object list. If you create a VPC peering connection with a VPC of another tenant, you must specify **tenant\_id** of the peer tenant.  For details see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

### Refusing a VPC Peering Connection

This interface is used by a tenant to refuse a VPC peering connection request initiated by another tenant.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peering | Dict | Yes | Specifies the VPC peering connection object list. If you create a VPC peering connection with a VPC of another tenant, you must specify **tenant\_id** of the peer tenant.  For details see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

### Updating a VPC Peering Connection

This interface is used to update information about a VPC Peering Connection.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| peering | Dict | Yes | Specifies the VPC peering connection object list. If you create a VPC peering connection with a VPC of another tenant, you must specify **tenant\_id** of the peer tenant.  For details see [Description of Creating a VPC Peering Connection](#_Description_of_Creating) |

### Deleting a VPC Peering Connection

This interface is used to delete a VPC Peering Connection.

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| Peering\_id | String | Yes | Deletes a VPC peering connection. |

# Disclaimer

This **DRAFT** Specification is being forwarded to you strictly for informational purposes and sign-off requirement before development starts. The specification is "AS IS," "WITH ALL FAULTS" and Click2Cloud makes no warranties, and disclaims all warranties, express, implied, or statutory related to the specifications. THE CORPORATIONS ARE NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES. THE CORPORATIONS ARE NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE.

# Appendix: Glossary

|  |  |  |
| --- | --- | --- |
| No. | Initial Name | Description |
| 1 | CR | Change Request |
| 2 | FSD | Functional Specification Document |
| 3 | VPC | Virtual Private Cloud |
| 4 | VPC Subservice | Subservices i.e. Subnet, Route and VPC Peering Connection |
| 5 | OTC | Open Telekom Cloud |

# Point of Contact

|  |  |  |
| --- | --- | --- |
| **Name** | **Designation** | **Email** |
| Sandeep Thakre | Project Manager | [sandeept@click2cloud.net](mailto:sandeept@click2cloud.net) |
| Sapan Vaswani | Lead Software Design Engineer | [sapan.vaswani@click2cloud.net](mailto:sapan.vaswani@click2cloud.net) |
|  |  |  |
|  |  |  |
|  |  |  |