VPC DNS

Product Documentation

Contents

VPC DNS 3
- Product Introduction 3
  - VPC DNS Overview 3
  - Strengths 5
  - Use Limits 6
  - Use Cases 7
- Quick Start 8
  - Getting Started 8
- Operation Guide 15
  - Description of Each Record Type 15
  - Adding a Forward Resolution Record 18
    - Creating Forward Resolution for Private Domain 18
    - A Record 20
    - AAAA Record 22
    - CNAME Record 24
    - MX Record 26
    - TXT Record 28
    - SRV Record 29
    - SPF Record 31
  - Adding a Reverse Resolution Record 33
    - Creating Reverse Resolution Private Domain 33
    - PTR Record 35
  - Deleting Resolution Record 37
  - Modifying Resolution Record 39
  - Importing DNS Records 40
  - Exporting DNS Records 41
  - Associating with VPC 42
  - Verifying DNS Record Validation 44
  - Deleting Private Domain 46
  - Enabling Subdomain Recursion 48
  - Forwarding List 50
- Cloud Access Management 53
  - Overview 53
  - Authorizable Resource Types 55
  - Policy Examples 57
- FAQs 59
  - FAQs 59
- Glossary 60
- API Documentation 62
  - VPC DNS (VPCDNS)(vpcdns) 62
    - version(2019-10-25) 62
      - API overview 62
      - CallingMethod 64
        - signaturev1 64
        - signaturev3 72
        - request 81
        - result 82
        - PublicParameter 85
      - vpcdns 87
        - BindVpcDnsDomain 87
        - CreateBackupRecord 89
        - CreateForwardRule 92
        - CreateVpcDnsDomain 94
        - CreateVpcDnsDomainRemark 96
        - CreateVpcDnsRecord 98
        - DeleteBackupRecord 101
        - DeleteForwardRule 103
        - DeleteVpcDnsDomain 105
        - DeleteVpcDnsRecord 107
        - DescribeBackupRecordList 109
        - DescribeExportFileUrl 111
        - DescribeForwardRuleList 113
        - DescribeImportTemplateUrl 115
        - DescribeUploadUrl 117
        - DescribeVpcDnsDomainList 119
        - DescribeVpcDnsRecordList 121
        - ImportRecords 123
        - ModifyForwardRule 125
        - ModifyVpcDnsDomain 127
        - ModifyVpcDnsRecord 129
      - Data structure 132
      - ErrorCode 135

Product Introduction
VPC DNS Overview

Overview

VPCDNS is a private DNS management service for the Virtual Private Cloud (VPC) environment. It allows you to map private domains to specific IP addresses in one or more custom VPCs.
With VPCDNS, you can easily create, delete, and modify private domains, manage VPC bindings, and support adding, deleting, and modifying records. These private domains cannot be accessed outside of the VPC. Moreover, you can connect your VPC to a traditional data center via methods such as dedicated lines or VPNs, allowing resources to be accessed mutually through private domains between the traditional data center and your Converge Cloud VPC.

Product Features

Private DNS

The private domain list contains the private domain resource records that you need to manage. You can create multiple private domains that you want to use and add the following types of resolution records for them:

Record Type	Description
A	It is used to specify the IPv4 address (for example 8.8.8.8) of a domain. If you want to point a domain to an IP address, you need to add an A record.
AAAA	It is used to specify the IPv6 address (for example ff06:0:0:0:0:0:0:c3) of a domain. If you want to point a domain to an IPv6 address, you need to add an AAAA record.
CNAME	Alias record. It points to another domain.
MX	If you want to set up your mailbox so that it can receive emails, you need to add an MX record.
TXT	You can enter anything in this record with a length limit of 255 characters. Most TXT records are used as SPF records (for anti-spam).
PTR	It reversely maps an IP address to a domain.

Record Type	Description
SRV	It is used to identify that a server is using a specific service, which is commonly found in the Microsoft system's directory management. The format is Service Name. Protocol Type, such as _sip._tcp.
SPF	Used to specify mail-sending servers. It's an efficient anti-spam solution.

Associating with VPC

You can associate a private domain with one or more VPCs that need to be configured so as to map it to IP addresses.

> Description
>
> Private domains with the same name cannot be associated with the same VPC. For example, when two example.com exist at the same time, it is not possible to associate both example.com with the same VPC.

Reverse Resolution

Reverse DNS refers to mapping an IP address to a domain; that is, the private domain pointed to by the IP address is obtained by querying the PTR record of the IP address.

Subdomain Recursive Resolution

When the subdomain recursive resolution setting is enabled, subdomains not configured will automatically be forwarded to the default recursive domain server for resolution.

Custom Private Domain

It plans CVM hostname management to make machine use and information more understandable and user-friendly.

Internal API Call

API calls are managed internally to avoid the troubles caused by IP address changes to the API use, which makes ops easier.

Internal Domain Security Isolation

The core system privacy protection feature ensures that the domain of internal core systems is not exposed to the Internet and thus improves their security.

Strengths

Secure and Reliable

- You can associate a custom resolution with a specific VPC. The resolution is effective only within that VPC and cannot be queried from any unassociated VPCs. This defines the access boundary of your internal systems and restricts core data access to the minimum scope.
- VPCDNS cannot be queried from the Internet, preventing external malicious probing of your internal service information and system architecture.
- By leveraging network tunneling features, your VPCDNS data is protected against malicious cracking.

High Scalability

The VPCDNS components are stateless and deployed on virtual machines, supporting rapid, smooth horizontal scale-out.

Unified Management

One private domain can be associated with multiple VPCs to facilitate unified management and deployment.

Comprehensive Recording

Various record types are supported, including A, CNAME, MX, AAAA, TXT, PTR, SRV, and SPF, catering to diverse DNS resolution configuration scenarios.

Reverse Resolution

You can set reverse resolution by creating a fixed private domain ending with in-addr.arpa and adding a PTR record.

Subdomain Recursive Resolution

When the subdomain recursive resolution setting is enabled, subdomains not configured will automatically be forwarded to the default recursive domain server for resolution.

Use Limits

For better use of the VPCDNS product, it is necessary to be aware of some operation limitations before use:

1. A VPC can only be associated with one of the same resolution domains/reverse resolution domains.
2. Up to 50 A, AAAA, and MX records are supported for the same subdomain.
3. Up to 10 TXT and SPF records are supported for the same subdomain.
4. PTR records can only be resolved to configured resolution domains.

Use Cases

Below are several common scenarios for VPCDNS:

Domain Privacy Protection Scenario

Core system privacy protection ensures that the domain of internal core systems is not exposed to the Internet, which is safer.

Reverse Resolution Scenario

Private IP address reverse resolution allows IP addresses to be reverse-mapped to a domain, making network troubleshooting more intuitive.

Private Network Access Hijacking

You can use VPCDNS to create a private domain and associate it with a VPC, add resolution records to the private domain, and set resource mapping to implement private network hijacking capabilities. When you access the private domain within the VPC, the returned data will be the mapped resource you have previously set.

Cloud Service Resource Management

You can use private domain records to manage CVM hostnames in the VPC. For example, you can plan CVM hostnames based on the cloud servers' region, business scenario, server information, etc., and use the hostname information to add private domain and resolution records for those servers. These private domains will be inaccessible outside the VPC, facilitating an intuitive management of cloud service resources.

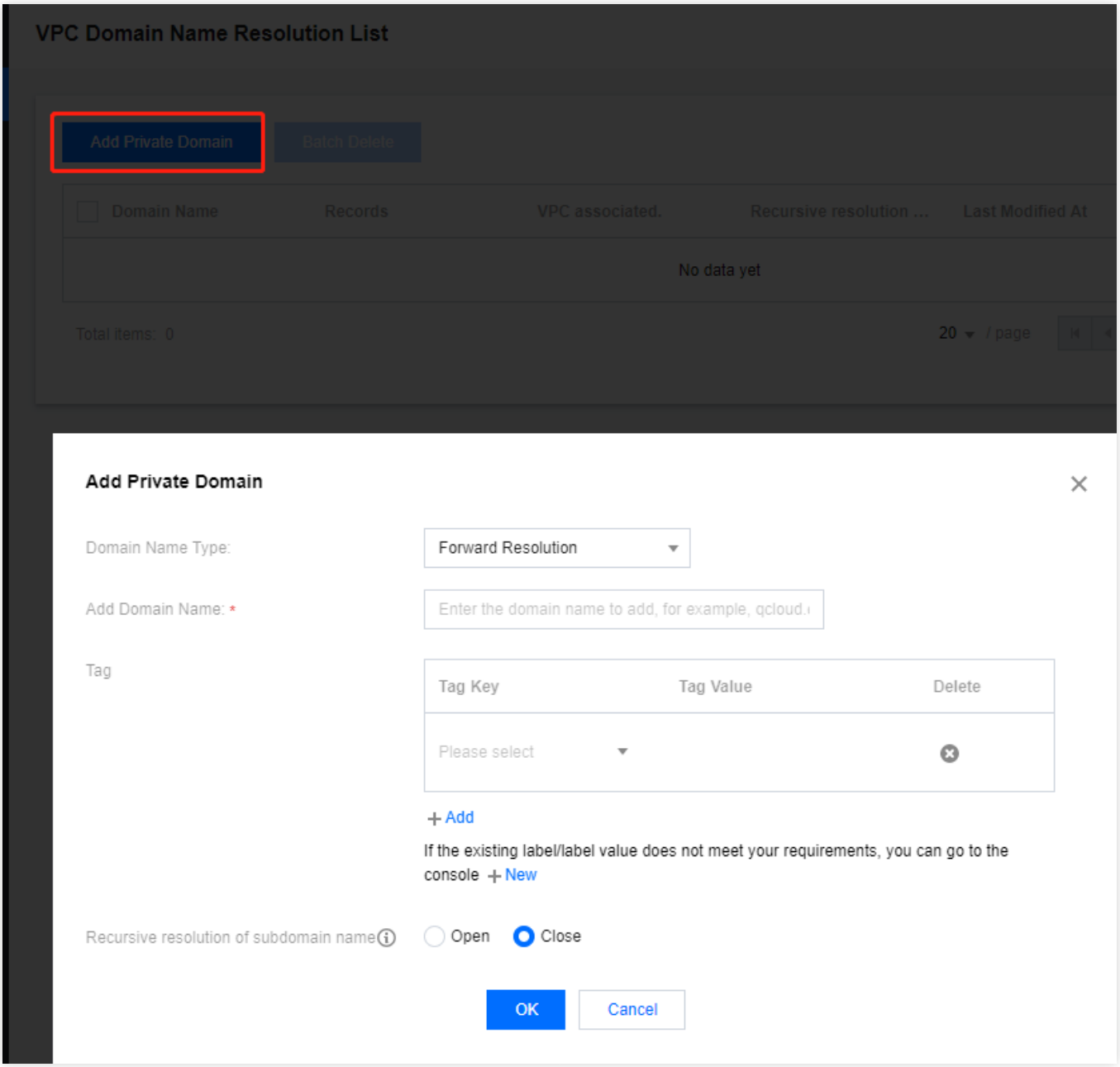
Cloud Service Resource Switching

Generally, to ensure the stable operation of a high-concurrency business, the business is distributed on multiple CVMs so they can share the pressure, and the same VPC can be established to enable mutual access between CVMs via private IPs. However, when a CVM is switched, its private IP will also change accordingly. Therefore, it is necessary to modify the business code and release the change, which is extremely inconvenient for the ops. In this case, you can create a private domain for each CVM in your VPC through private domain services and add resolution records pointing to the corresponding private IPs. The CVMs can access each other through the private network domain, and when a CVM is switched, you do not need to modify the CVM code. Instead, you only need to modify the resolution record of its domain.

Quick Start
Getting Started

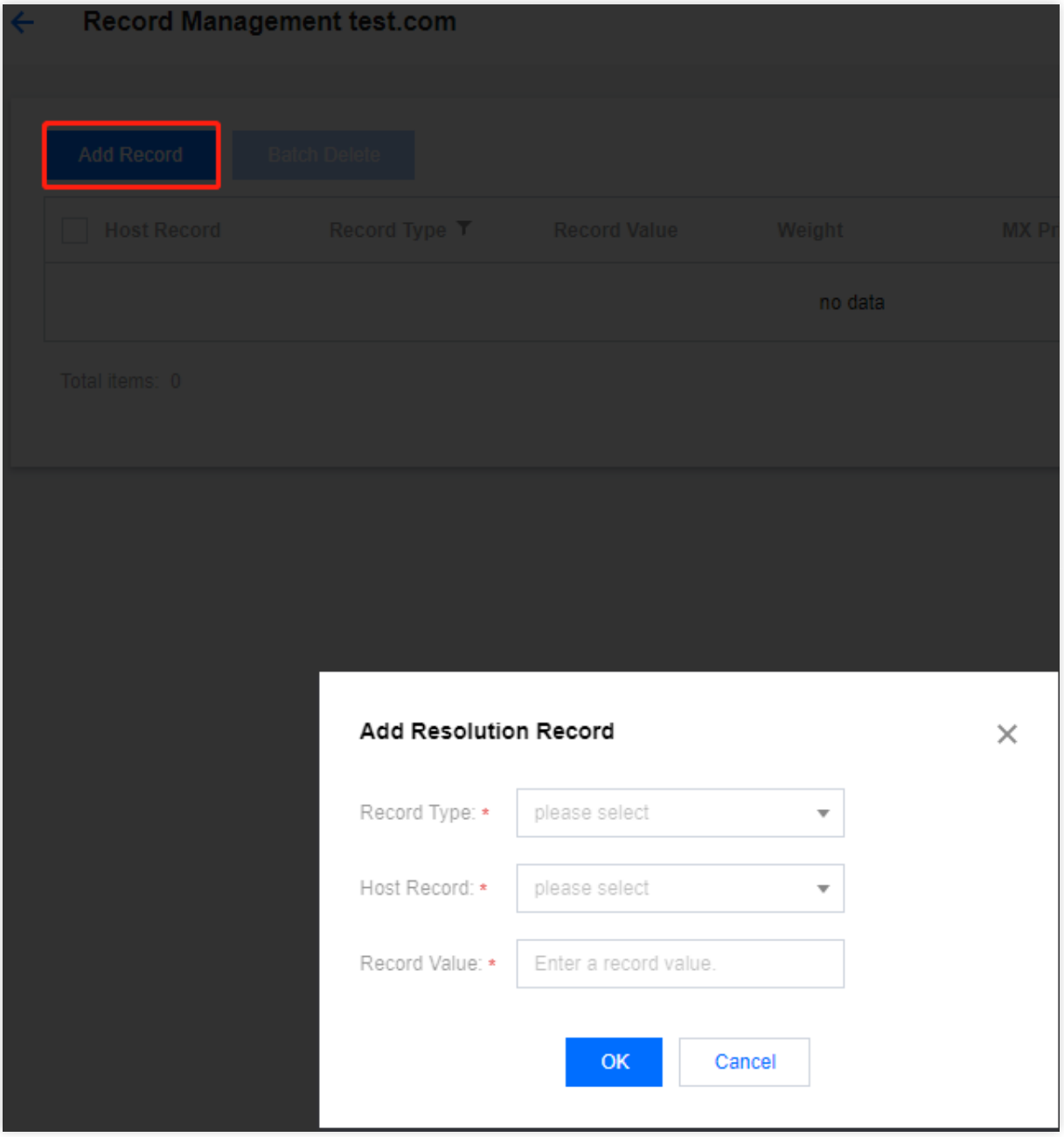
Adding a Private Domain

1. Click Add Private Domain to add the domain you want to resolve.



Adding a Resolution Record

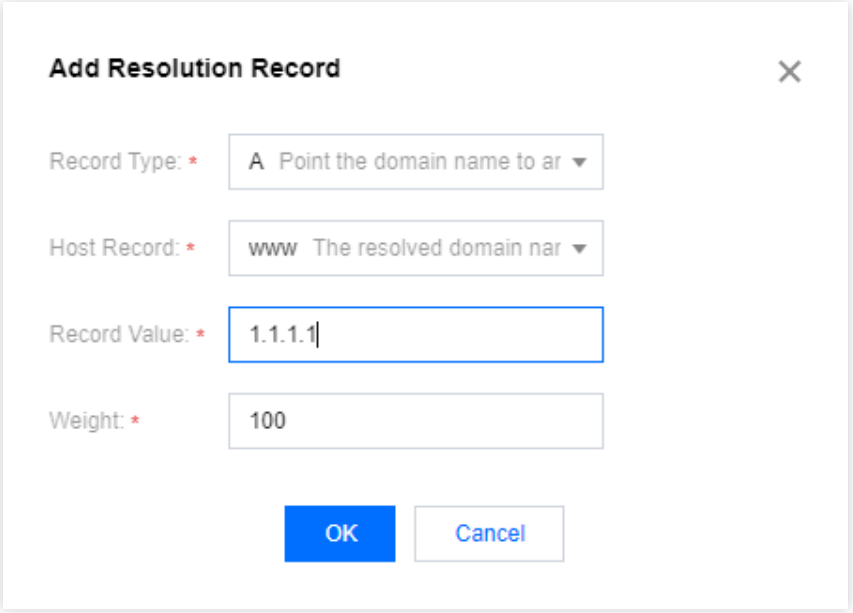
1. On the list page, click Private Domain to enter the Record Management page.
2. Click Add Record to add a resolution record for that domain.



Adding a Forward Resolution Record

Adding an A Record

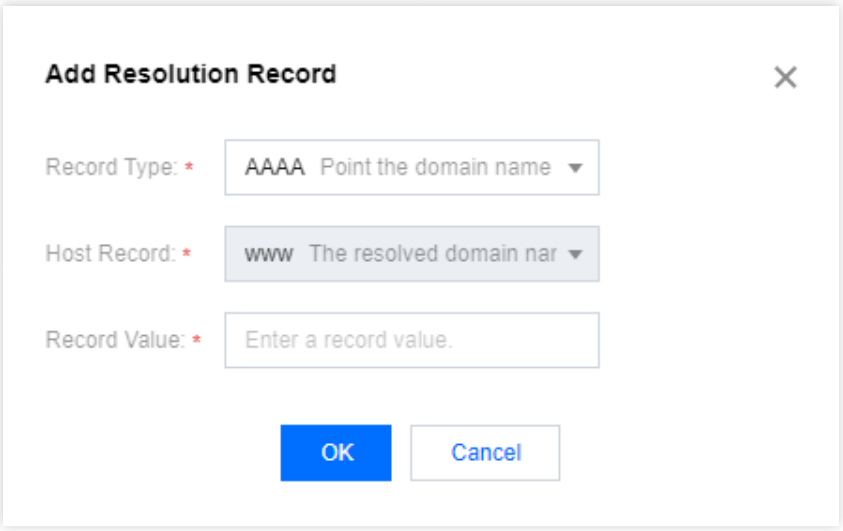
1. Add an A record for the domain.



> Note:
>
> - Weight range: 1–100 (default: 100)
> - Resolution proportion = record weight ÷ sum of all weights under the subdomain

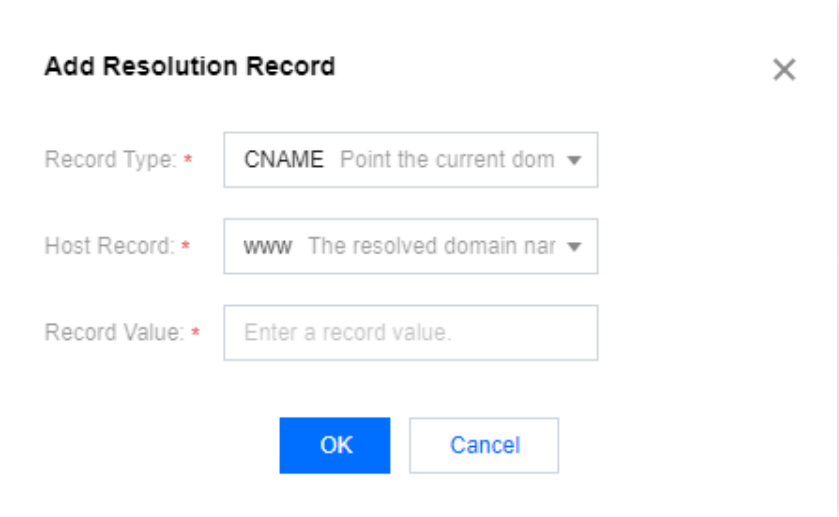
Adding an AAAA Record

1. Add an AAAA record for the domain.



Adding a CNAME Record

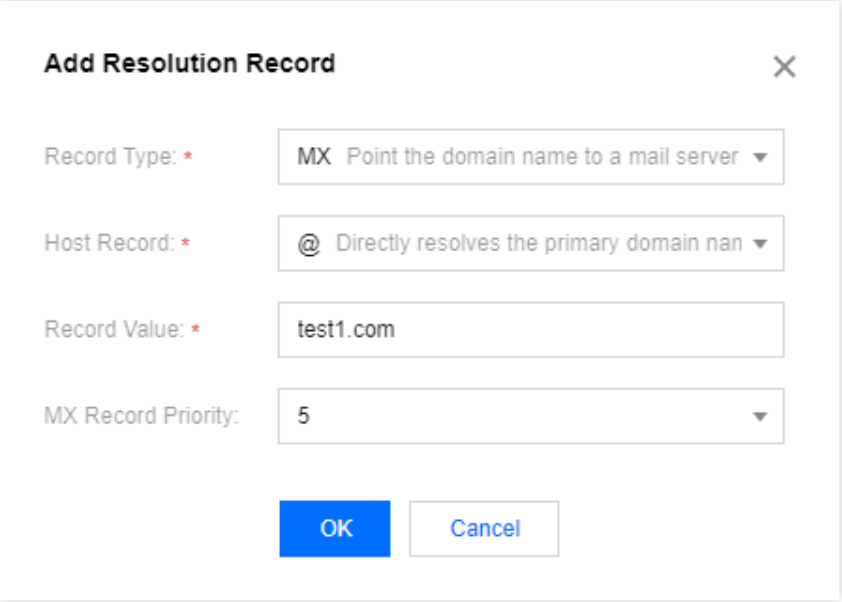
1. Add a CNAME record for the domain.



> Note:
>
> - Only one CNAME record per host record; cannot coexist with other records
> - Supports CNAME only to private domains under the current account

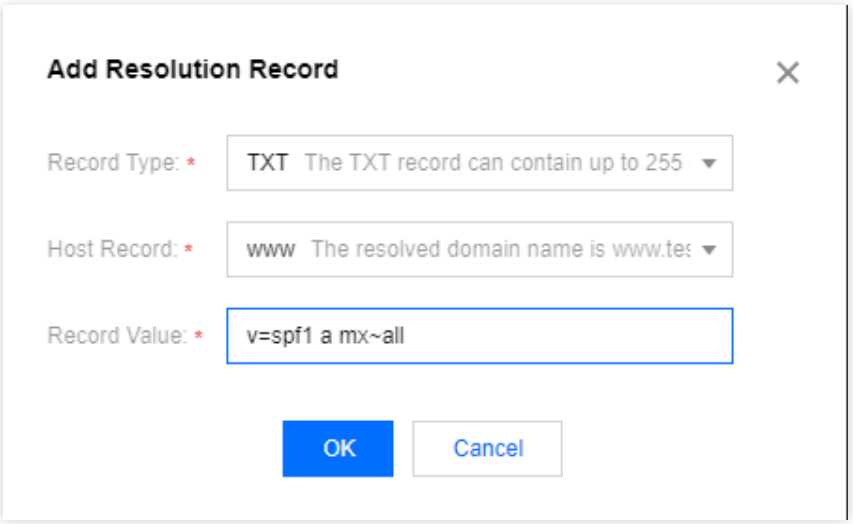
Adding an MX Record

1. Add an MX record for the domain.



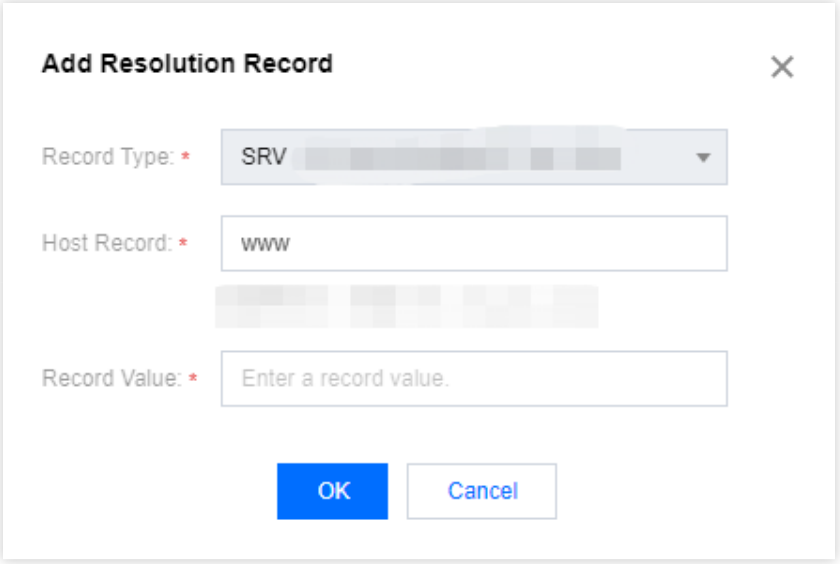
Adding a TXT Record

1. Add a TXT record for the domain.



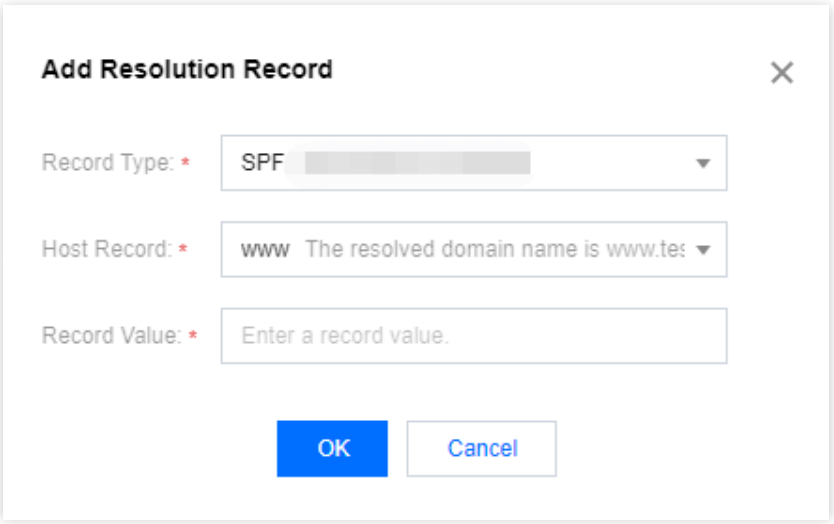
Adding an SRV Record

1. Add an SRV record for the domain.



Adding an SPF Record

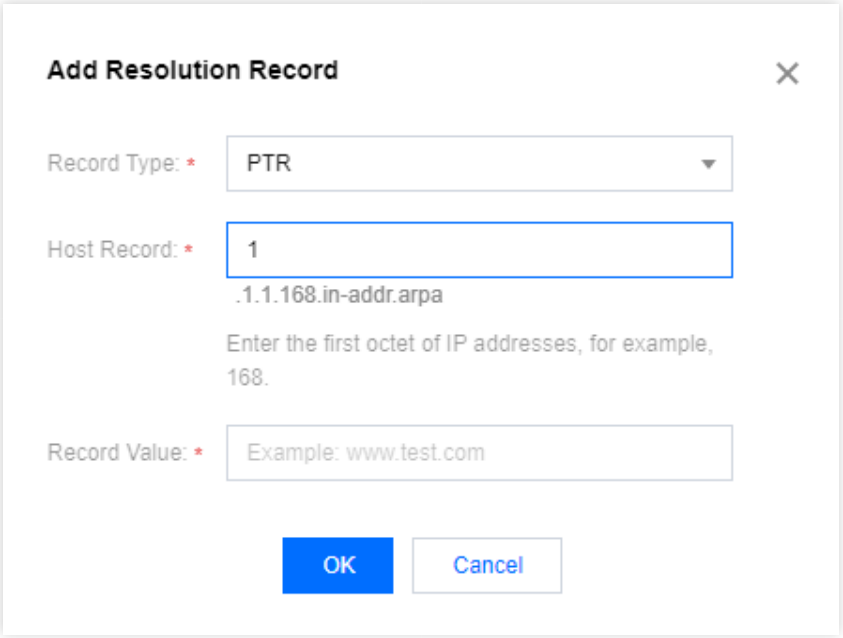
1. Add an SPF record for the domain.



Adding a Reverse Resolution Record

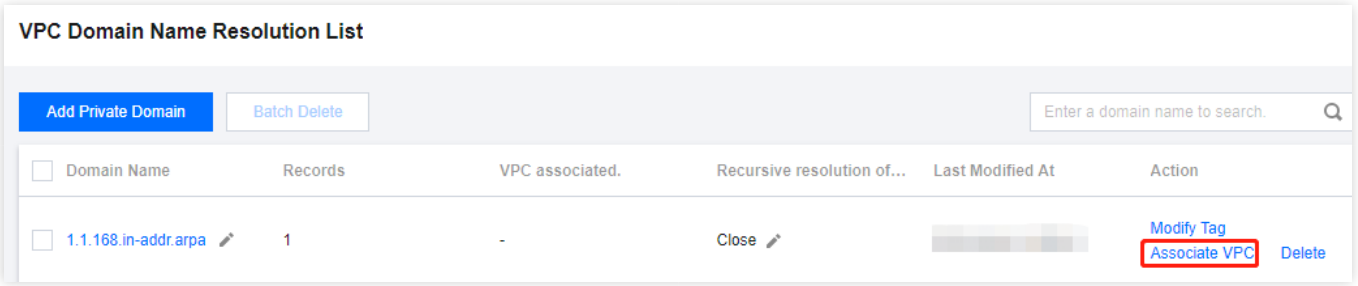
Adding a PTR Record

1. Add a PTR record for the domain.

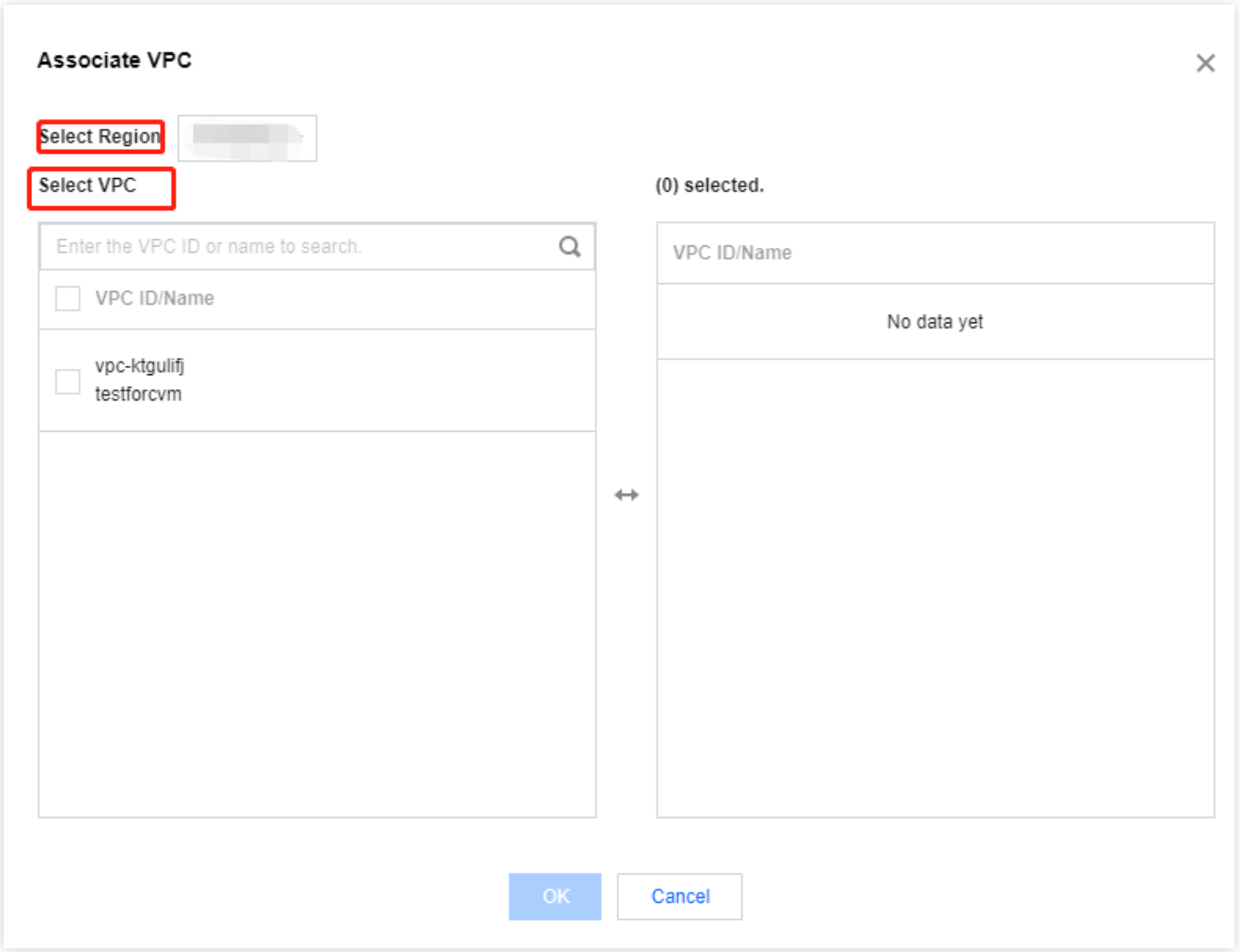


Associating with a VPC

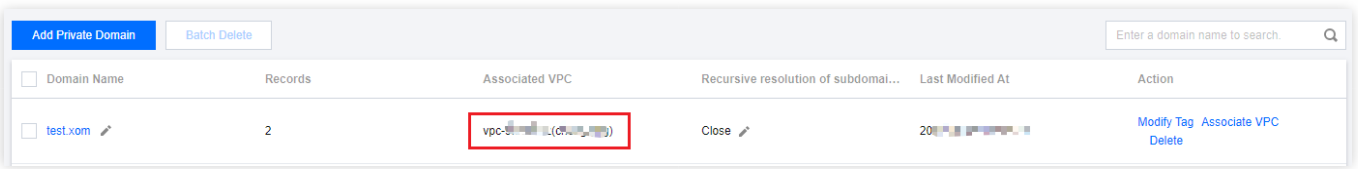
1. After adding the domain, click Associate with VPC.



2. Select a region and VPC, then click OK to associate.



3. The resolution takes effect immediately.



Operation Guide
Description of Each Record Type

VPCDNS Supported Resolution Record Types

Record Type	Description	Host Record Description	Record Value Description	MX Priority Description	Weight Description
A Record	Points the domain to an IPv4 address.	Select a subdomain. For example, for www.123.com, select www . To resolve 123.com, select @ .	Enter an IPv4 address (e.g., 1.1.1.1).	-	Weight range: 1–100 (default: 100). Resolution proportion = record weight ÷ total weight of all records for the subdomain.
AAAA Record	Points the domain to an IPv6 address.	Select a subdomain (e.g., www) or @ for the root domain.	Enter an IPv6 address (e.g., 1030::C9B4:FF12:48AA:1A2B).	-	-
CNAME Record	Points the domain to another domain. Cannot coexist with other record types.	Enter a subdomain (e.g., www) or @ for the root domain.	Enter the target domain.	-	-

Record Type	Description	Host Record Description	Record Value Description	MX Priority Description	Weight Description
MX Record	Points the domain to an email server address. Priority can be set from 1–50; lower value = higher priority.	Enter a subdomain (e.g., mail) or @ for the root domain.	Domain or IP address of the mail server.	Set priority from 1–50. Lower value = higher priority.	-
TXT Record	Text record (≤255 characters), commonly used for SPF.	Enter a subdomain (e.g., www) or @ for the root domain.	Free format. Example SPF: "v=spf1 a mx ~all" .	-	-
PTR Record	Reverse resolution for an IP address. Only for reverse-resolution domains.	For reverse-resolution domain 1.168.192.in-addr.arpa and IP 192.168.1.1, enter 1 .	Target domain for reverse resolution.	-	-
SRV Record	Identifies a server offering a specific service, e.g., Microsoft directory services.	Enter service name and protocol type (e.g., _sip._tcp).	Priority, weight, port, and hostname. A dot is added at the end automatically (e.g., 5 0 5269 xmpp-server.l.test.com).	-	-

Record Type	Description	Host Record Description	Record Value Description	MX Priority Description	Weight Description
SPF Record	Specifies authorized mail-sending servers, helps prevent spam.	Enter a subdomain (e.g., www) or @ for the root domain.	Example: v=spf1 include:spf.mail.test.com ~all — only IPs in A and MX records are allowed to send emails.	-	-

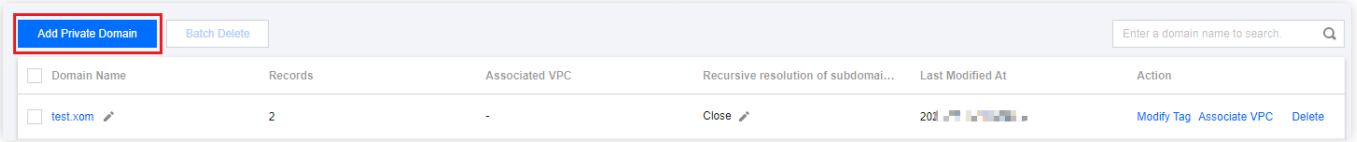
Adding a Forward Resolution Record Creating Forward Resolution for Private Domain

Operation Scenarios

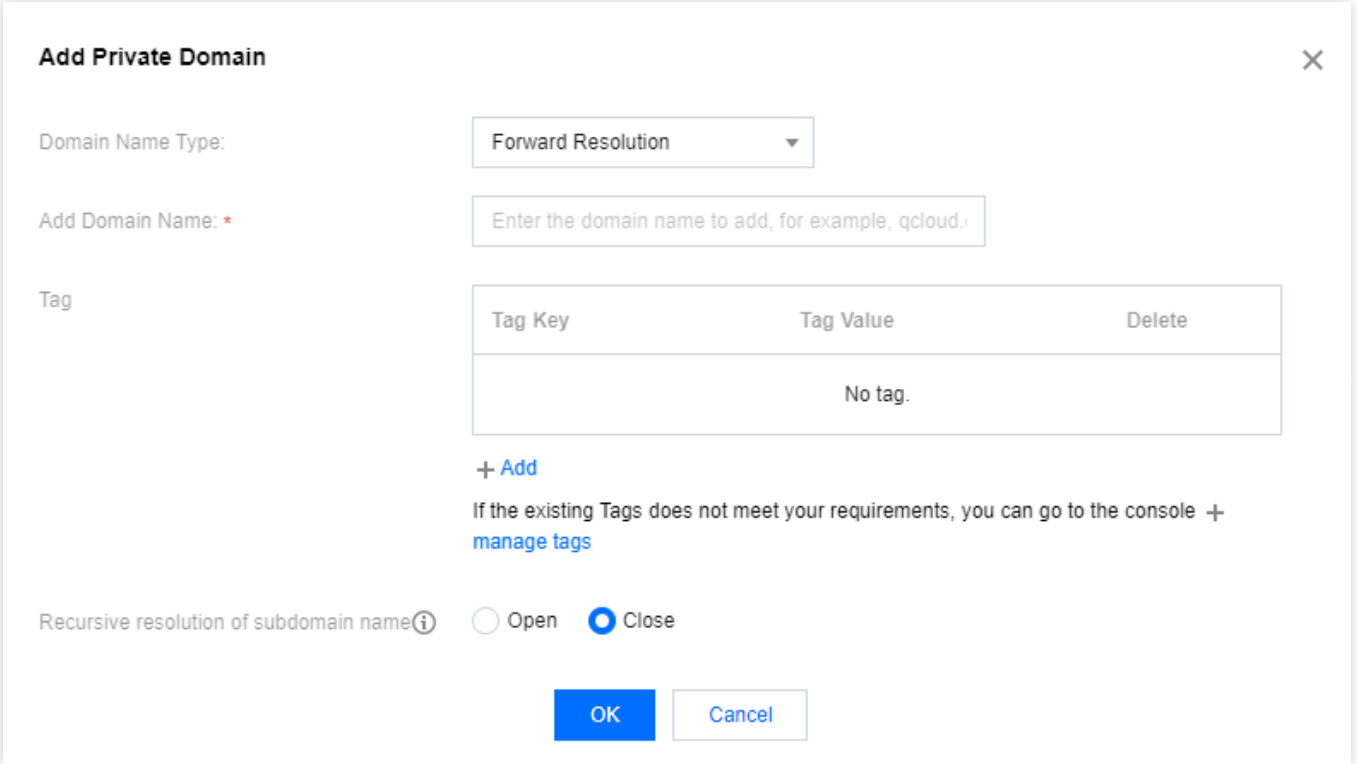
This document describes how to create a private domain. After the private domain is created, you need to set resolution records first before you can associate this domain with a VPC. Once the domain is associated with a VPC, the domain's private network resolution record will overwrite its public network resolution record and prioritize resolution to the VPC network.

Directions

1. Log in to the cloud console and select Cloud Services > VPCDNS to enter the VPCDNS console.
2. Click Add Private Domain.



3. In the pop-up Add Private Domain dialog box, fill in the relevant information for the private domain.



   - Domain Type: Select forward resolution.
   - Add Domain: Custom your domain.

- Tag: Select a tag. You can categorize, search for, and aggregate cloud resources through tags.
- Subdomain Recursive Resolution: Select according to your actual needs; the default state is Off. For more details, see Subdomain Recursive Resolution Description.

4. After setting the parameters, click OK to complete adding the forward resolution of the private domain.

A Record

Operation Scenarios

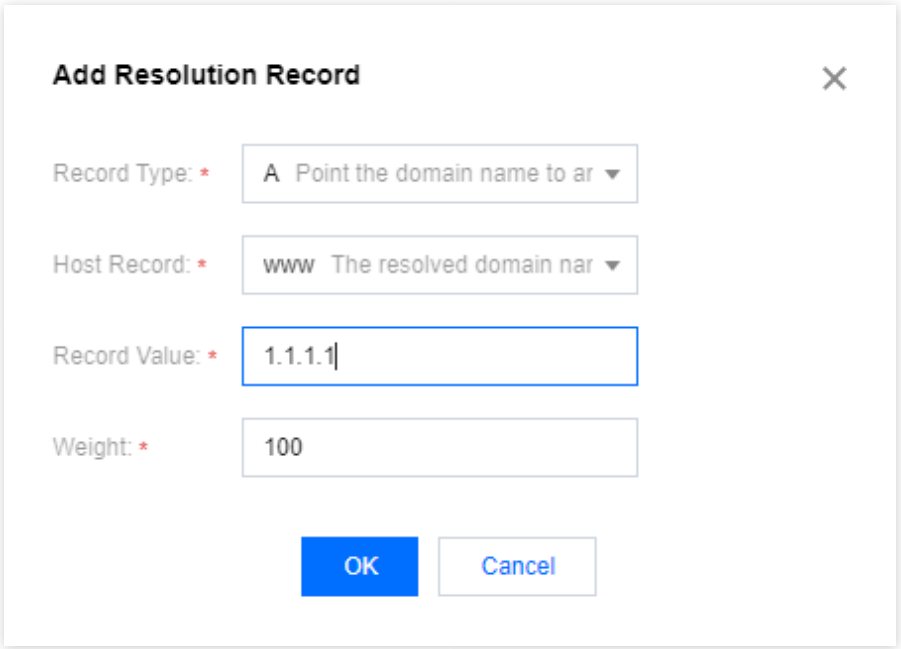
You can add an A record to point a domain to an IP address (private network address). This document describes how to add an A record.

Prerequisites

The corresponding private domain has been created. For details, see Creating Forward DNS Resolution for Private Domain.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the VPCDNS List section, click the name of the private domain for which you need to create an A record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Select A.
   - Host Record: Select a subdomain.

- Record Value: Enter the server's private network address.
- Weight: The weight value is set to an integer within the range of 1–100, with the default value being 100.

4. Click Save to complete the addition.

AAAA Record

Operation Scenarios

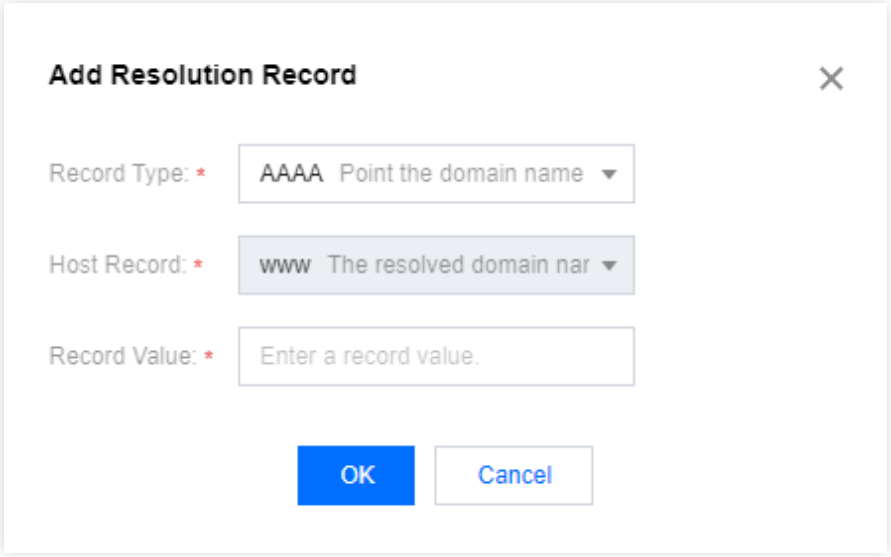
Adding an AAAA record can point a domain to an IPv6 address. This document describes how to add an AAAA record.

Prerequisites

The corresponding private domain has been created. For details, see Creating Forward DNS Resolution for Private Domain.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the VPCDNS List section, click the name of the private domain for which you need to create an AAAA record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Select AAAA.
   - Host Record: Enter the subdomain.
   - Record Value: Enter the IP address, you can only enter an IPv6 address.
4. Click Save to complete the addition.

CNAME Record

VPCDNS Supported Resolution Record Types

Record Type	Description	Host Record Description	Record Value Description	MX Priority Description	Weight Description
A Record	Points the domain to an IPv4 address.	Select a subdomain. For example, for www.123.com, select www. To resolve 123.com, select @.	Enter an IPv4 address (e.g., 1.1.1.1).	-	Weight range: 1–100 (default: 100). Resolution proportion = record weight ÷ total weight of all records for the subdomain.
AAAA Record	Points the domain to an IPv6 address.	Select a subdomain (e.g., www) or @ for the root domain.	Enter an IPv6 address (e.g., 1030::C9B4:FF12:48AA:1A2B).	-	-
CNAME Record	Points the domain to another domain. Cannot coexist with other record types.	Enter a subdomain (e.g., www) or @ for the root domain.	Enter the target domain.	-	-
MX Record	Points the domain to an email server address. Priority can be set from 1–50; lower	Enter a subdomain (e.g., mail) or @ for the root domain.	Domain or IP address of the mail server.	Set priority from 1–50. Lower value = higher priority.	-

Record Type	Description	Host Record Description	Record Value Description	MX Priority Description	Weight Description
	value = higher priority.				
TXT Record	Text record (≤255 characters), commonly used for SPF.	Enter a subdomain (e.g., www) or @ for the root domain.	Free format. Example SPF: "v=spf1 a mx ~all" .	-	-
PTR Record	Reverse resolution for an IP address. Only for reverse-resolution domains.	For reverse-resolution domain 1.168.192.in-addr.arpa and IP 192.168.1.1, enter 1 .	Target domain for reverse resolution.	-	-
SRV Record	Identifies a server offering a specific service, e.g., Microsoft directory services.	Enter service name and protocol type (e.g., _sip._tcp).	Priority, weight, port, and hostname. A dot is added at the end automatically (e.g., 5 0 5269 xmpp-server.l.test.com).	-	-
SPF Record	Specifies authorized mail-sending servers, helps prevent spam.	Enter a subdomain (e.g., www) or @ for the root domain.	Example: v=spf1 include:spf.mail.test.com ~all — only IPs in A and MX records are allowed to send emails.	-	-

MX Record

Operation Scenarios

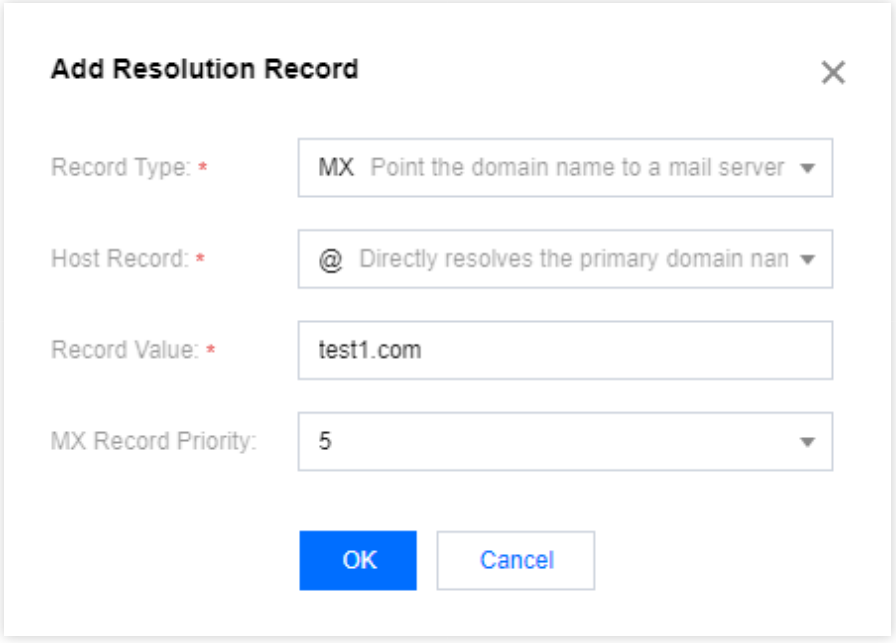
If you want to set up your mailbox so that it can receive emails, you need to add an MX record. This document describes how to add an MX record.

Prerequisites

You have created the corresponding private domain. For more details, see Creating Private Domain.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the "VPC Domain Resolution List", click the private domain name for which you need to create an MX record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Select MX.

- Host Record: Enter the subdomain.
- Record Value: It can be a domain or an IP address.
  - If the record value is a domain, the domain pointed to must have an A record, and a dot will be automatically added at the end of the domain after the record is generated.
  - If the record value is an IP address, directly enter the mail server's IP address, and a dot will be automatically added at the end after the record is generated.
- MX Priority: The lower the value, the higher the priority.

> Note:
> The value of MX priority can only be a positive multiple of 5 and must be ≤ 50. Lower values indicate higher priority. Ensure the domain used in the record value has an A record if it's a domain, or enter a valid IP address.

4. Click Save to complete the addition.

TXT Record

Operation Scenarios

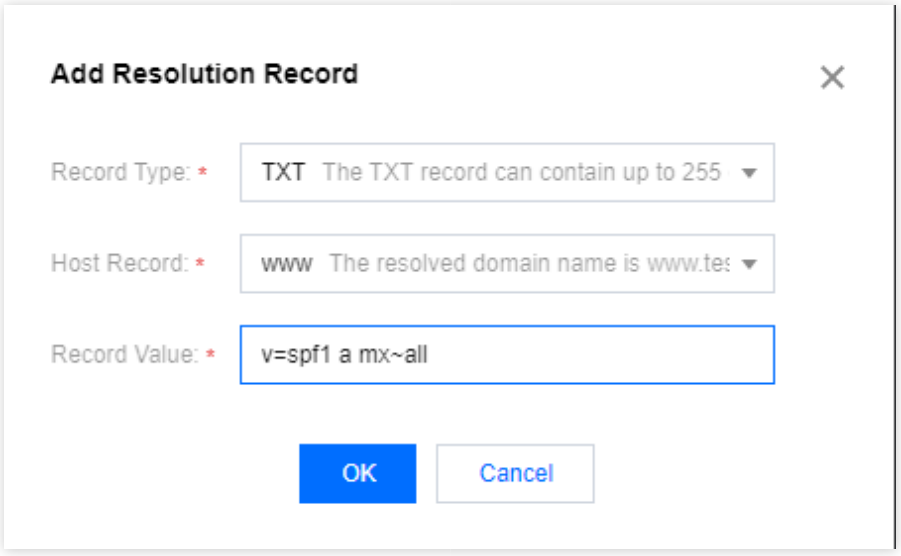
If you want to identify and describe a domain, you can use TXT records. This document describes how to add a TXT record.

Prerequisites

The corresponding private domain has been created. For details, refer to.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the VPCDNS List section, click the name of the private domain for which you need to create a TXT record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Select TXT.
   - Host Record: Enter the subdomain.
   - Record Value: There is no fixed format.
4. Click Save to complete the addition.

SRV Record

Operation Scenarios

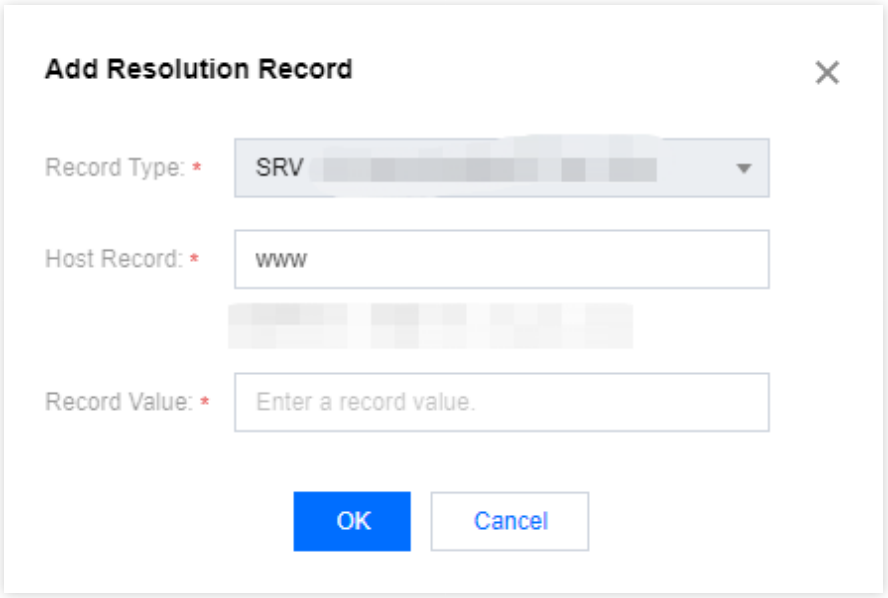
SRV records are used to identify a server that is using a specific service, which is commonly found in the Microsoft system's directory management. This document describes how to add an SRV record.

Prerequisites

The corresponding private domain has been created. For details, see Creating Forward DNS Resolution for Private Domain.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the VPCDNS List section, click the name of the private domain for which you need to create an SRV record.
3. Go to the Resolution Record Management page and click Add Record. In the pop-up Add Resolution Record window, fill in the relevant record value information, as shown below:



   - Record Type: Select SRV.
   - Host Record: The service's name. The type of protocol.
   - Record Value: Priority, weight, port, and hostname.

4. Click Save to complete the addition.

SPF Record

Operation Scenarios

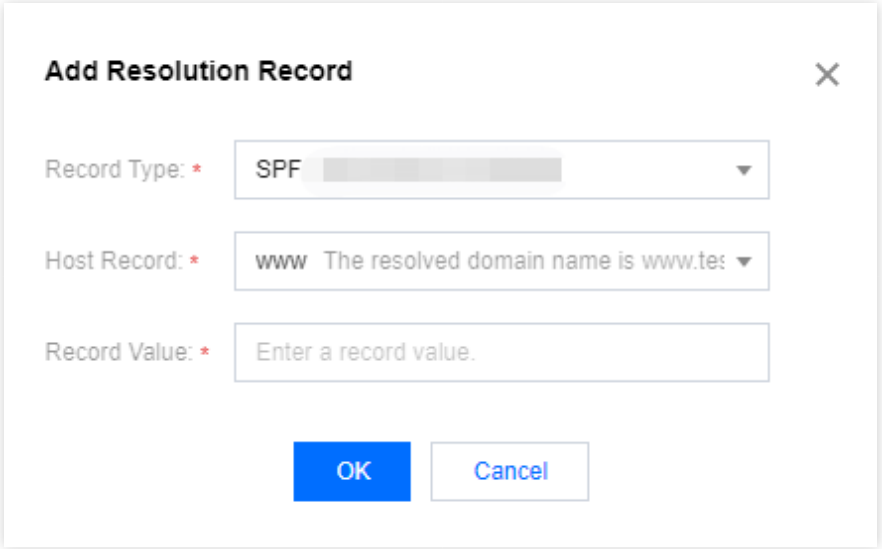
SPF records are used to specify the servers that send emails and represent an efficient anti-spam solution. This document describes how to add an SPF record.

Prerequisites

The corresponding private domain has been created. For details, see Creating Forward DNS Resolution for Private Domain.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the "VPC Domain Resolution List", click the private domain name for which you need to create an SPF record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Select SPF.
   - Host Record: Enter the subdomain.
   - Record Value: For example, v=spf1 include:spf.mail.test.com ~all , which indicates that only the IP addresses in A and MX records of the domain name are authorized to send emails using this

domain name.

4. Click Save to complete the addition.

Adding a Reverse Resolution Record
Creating Reverse Resolution Private Domain

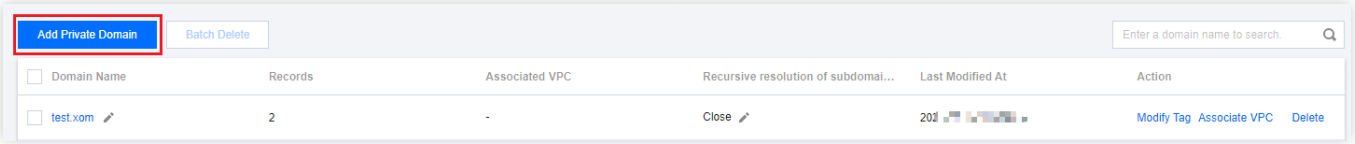
Reverse resolution maps an IP address to a domain, equivalent to how forward resolution maps a domain to an IP address.

Operation Scenarios

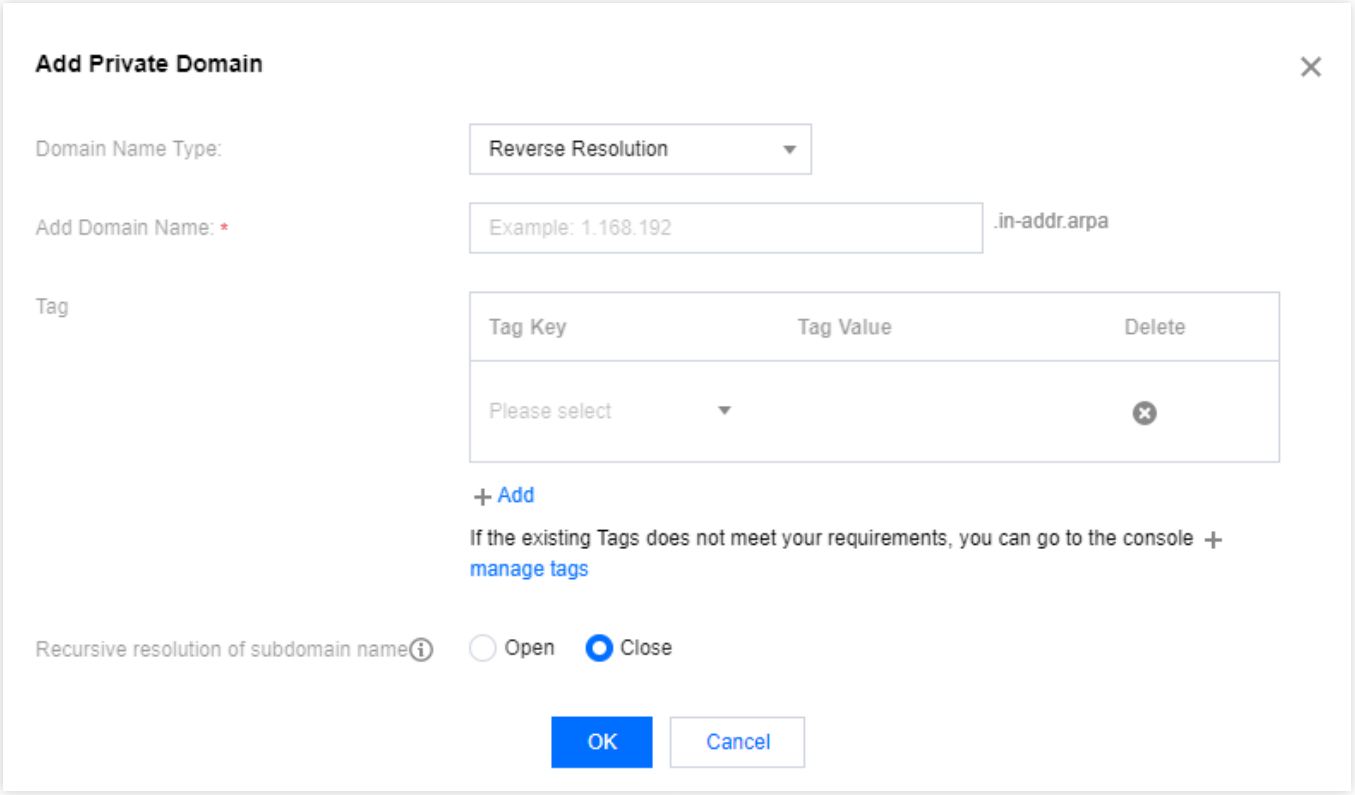
This document describes how to create a private domain. After the private domain is created, you need to set resolution records first before you can associate this domain with a VPC. Once the domain is associated with a VPC, the domain's private network resolution record will overwrite its public network resolution record and prioritize resolution to the VPC network.

Directions

1. Log in to the cloud console and select Cloud Services > VPCDNS to enter the VPCDNS console.
2. Click Add Private Domain, as shown below:



3. In the pop-up Add Private Domain dialog box, fill in the relevant information for the private domain.



   - Domain Type: Select reverse resolution.

- Add Domain: Enter the domain to be resolved.
- Tag: Select a tag. You can categorize, search for, and aggregate cloud resources through tags.
- Subdomain Recursive Resolution: Select according to your actual needs; the default state is Off. For more details, see Subdomain Recursive Resolution Description.

4. After setting the parameters, click OK to complete adding the reverse resolution of the private domain.

PTR Record

Operation Scenarios

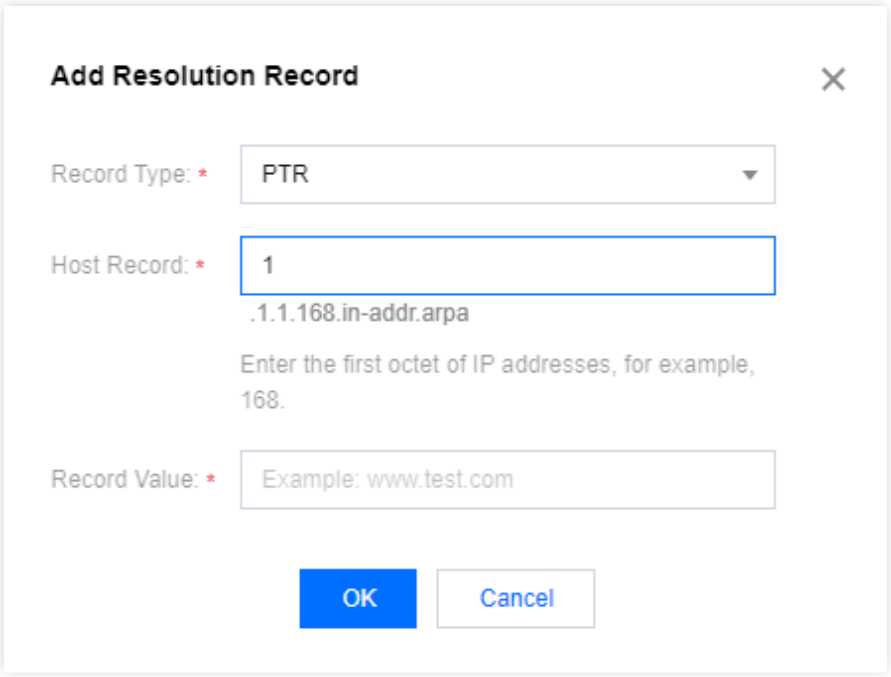
If you want to reversely resolve the corresponding private network domain by private IP address, you can use the PTR record type. This document describes how to add a PTR record.

Prerequisites

- The corresponding reverse private DNS has been created. For details, refer to Creating Reverse Resolution for Private Domains.
- You have configured reverse private DNS.

Directions

1. Log in to the VPCDNS console and enter the VPCDNS List management page.
2. In the VPCDNS List section, click the name of the private domain for which you need to create a PTR record.
3. Enter the Resolution Record Management Page, click Add Record, and in the pop-up "Add DNS Record" window, fill in the relevant record value information.



   - Record Type: Only PTR is supported.

- Host Record: The host record, combined with the created domain (excluding in-addr.arpa), is in the fixed IPv4 format. You can only enter integers from 0 to 255 for each IP range.
- Record Value: Fill in the private domain corresponding to the private IP address.

4. Click Save to complete the addition.

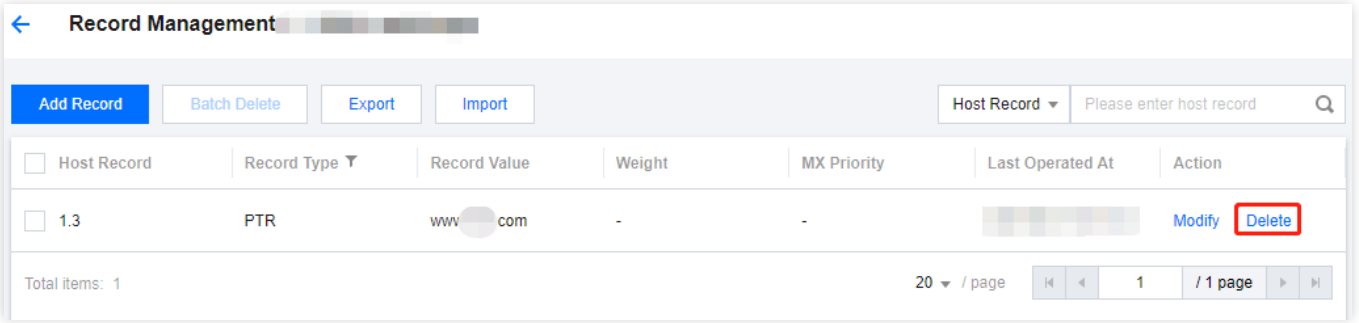
Deleting Resolution Record

Operation Scenarios

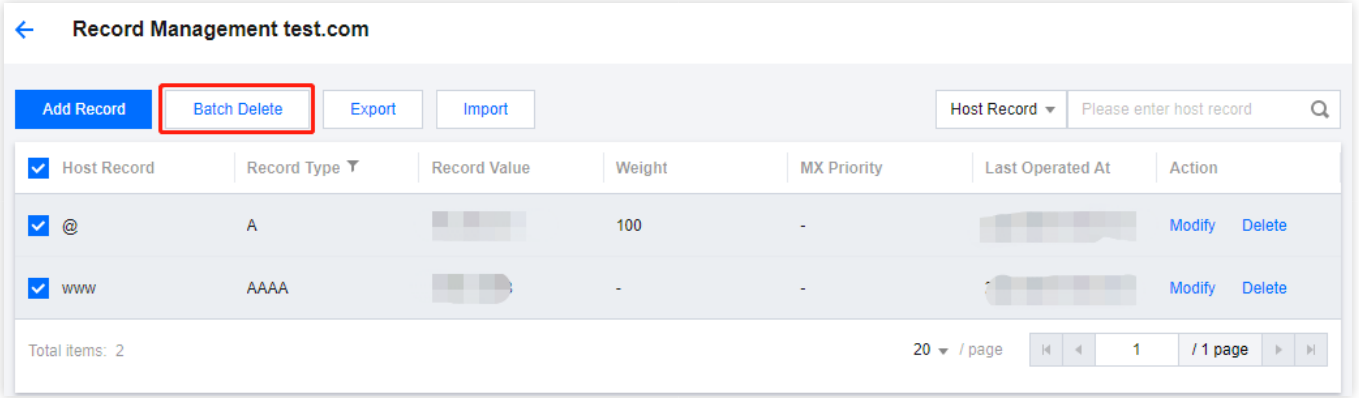
When you don't need a record any longer, you can delete it. Once deleted, the record will stop taking effect. Proceed with caution.

Directions

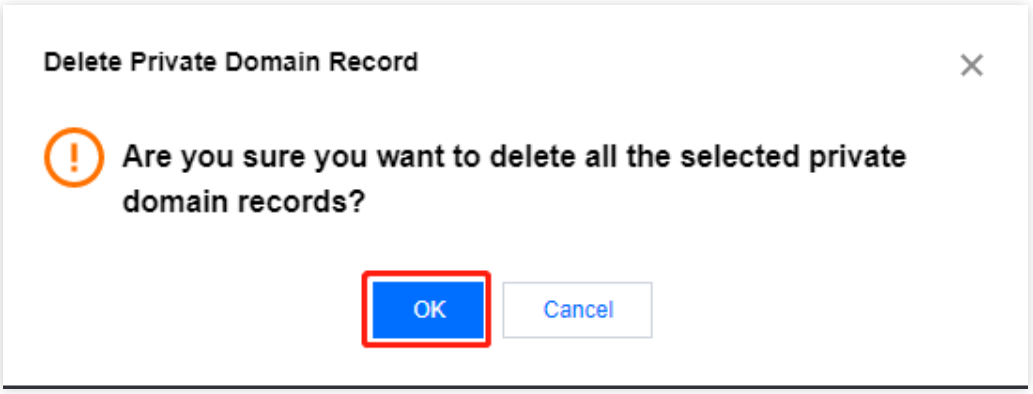
1. Log in to the Converge Cloud VPC DNS console and enter the VPC DNS List management page.
2. In the "VPC Domain Name Resolution List", select the private domain name for which you need to delete records, and enter the Resolution Record page of that domain.
3. Please select the resolution records you need to delete and click Delete in the action column to delete them.



4. If you want to delete multiple resolution records, you can select multiple records and click the Batch Deletion button to delete them.



5. In the pop-up dialog box, click OK to complete the deletion.



Note:
Deleted records cannot be restored. Proceed with caution.

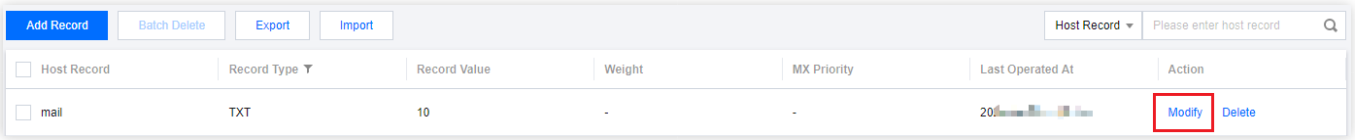
Modifying Resolution Record

Operation Scenarios

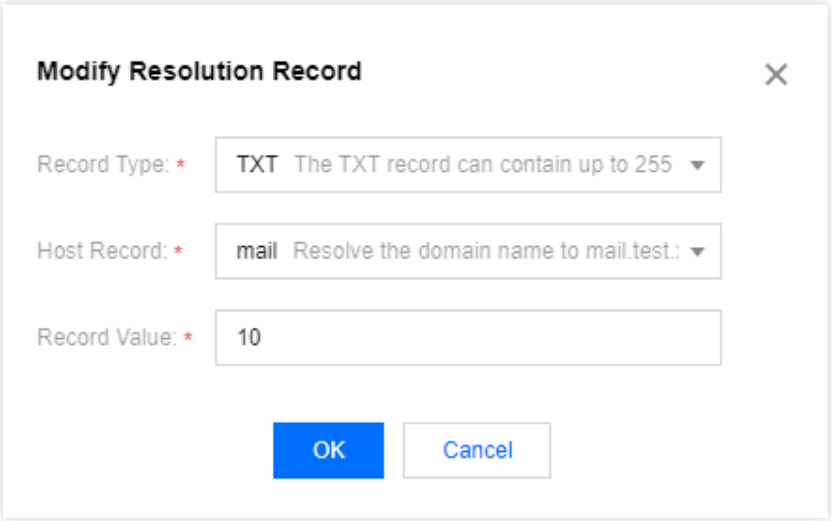
In general, if your resolution record needs to be changed, you can modify it directly. This document describes how to modify your record as needed.

Directions

1. Log in to the VPC DNS console and enter the VPC DNS List management page.
2. In the "VPC Domain Resolution List", select the private domain you need to modify the record for, and go to the Resolution Record page of that domain.
3. Please select the resolution record you need to modify and click Modification in the Operation Column.



4. In the pop-up Edit Resolution Record window, after completing the modifications, click OK to save the changes.



Importing DNS Records

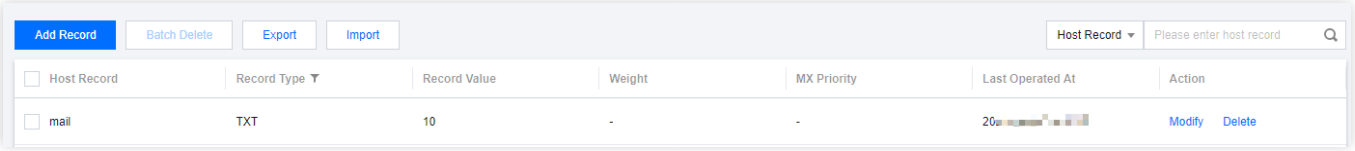
Operation Scenarios

This document describes how to quickly import resolution records.

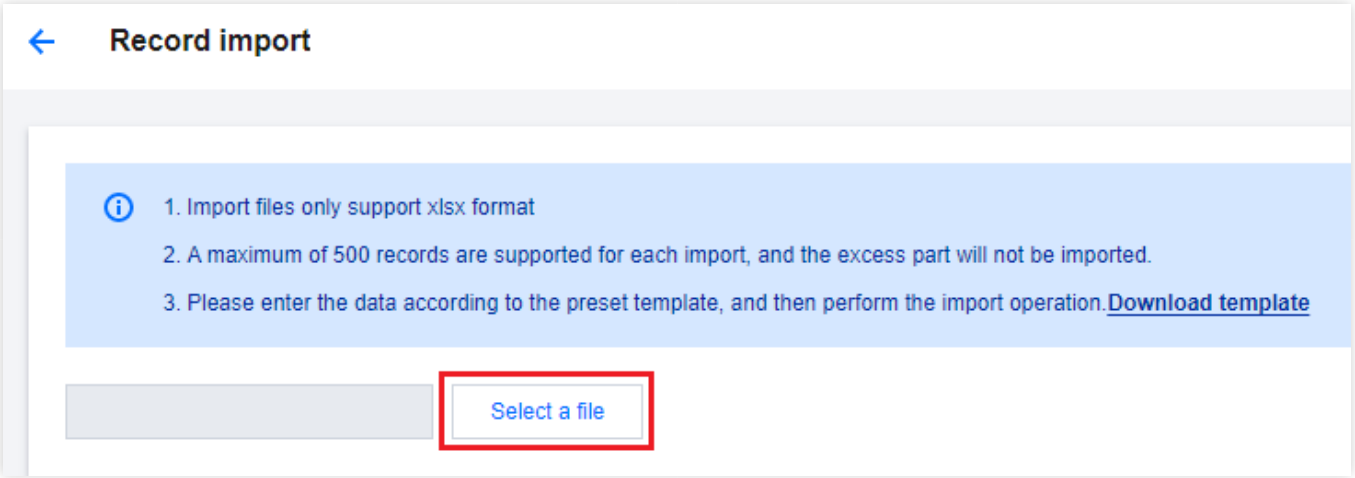
Directions

> Note:
> A maximum of 500 resolution records can be imported at a time; records exceeding this number will not be imported.

1. Log in to the Converge Cloud VPC DNS console and enter the VPCDNS List management page.
2. In the "VPC Domain Name Resolution List", click the private domain name you want to import.
3. Enter the Resolution Record Management Page, and click Import.



4. On the "Import Records" page, click Select file to import the resolution file.



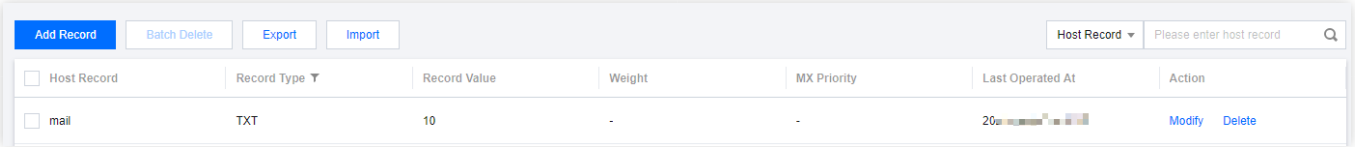
Exporting DNS Records

Operation Scenarios

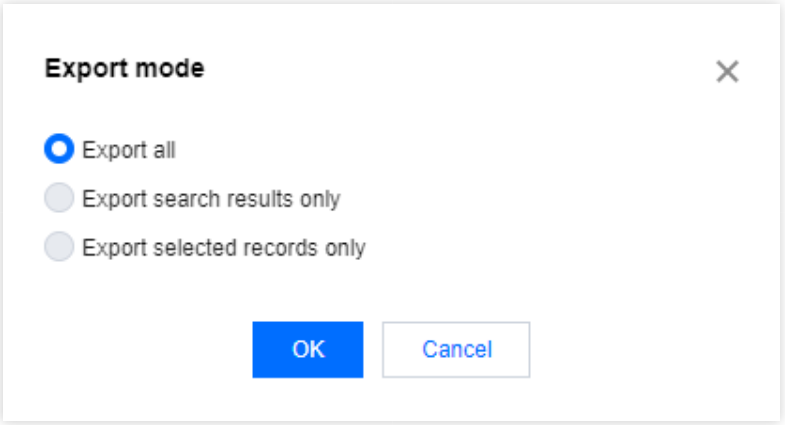
This document describes how to quickly export resolution records.

Directions

1. Log in to the VPC DNS console and enter the VPC DNS List management page.
2. In the "VPC Domain Name Resolution List", click the private domain name you need to export.
3. Enter the Resolution Record Management Page, click Export.



4. In the pop-up "Export Method" window, select the method you need to export.

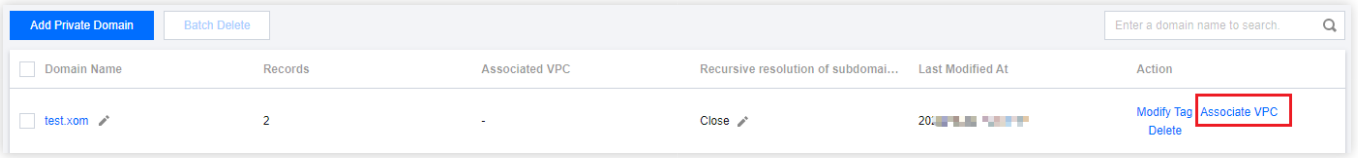


5. Click OK to export the resolution records.

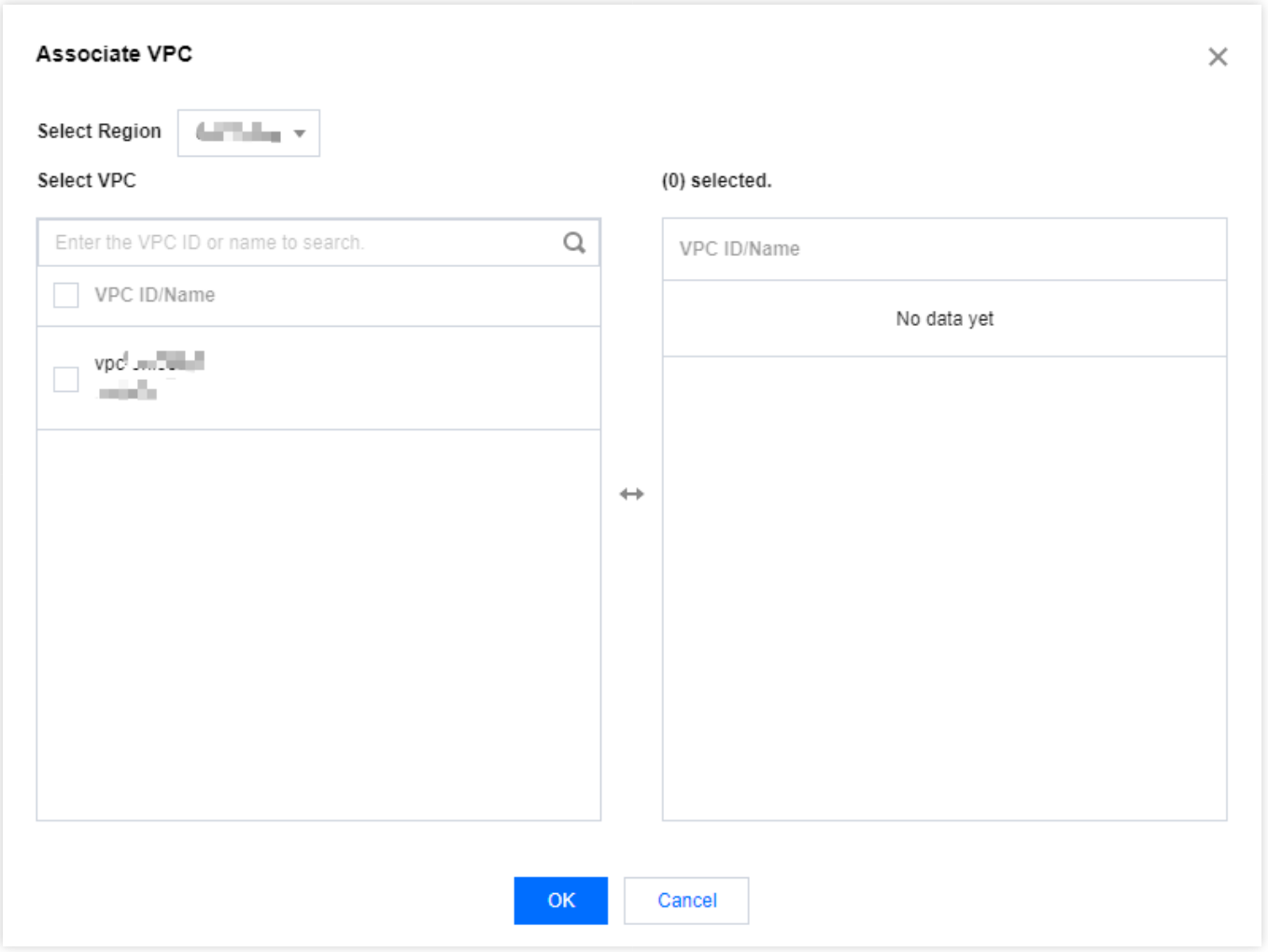
Associating with VPC

Associating with VPC

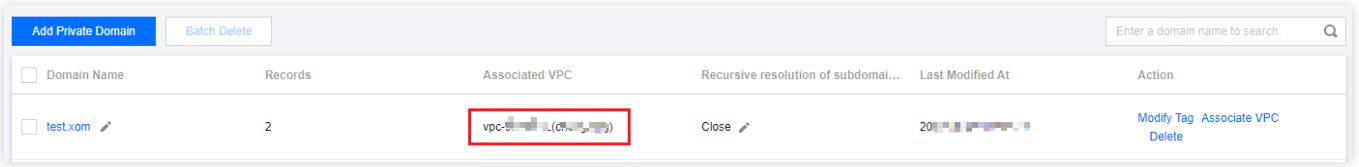
1. After adding a domain, click Associating with VPC.



2. Select a region and VPC, and after selecting, click OK to associate.

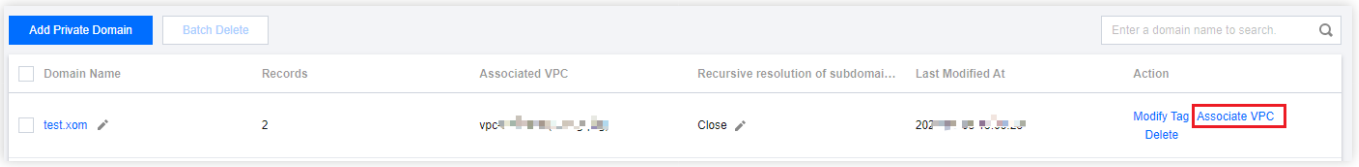


3. After adding, the resolution takes effect immediately.

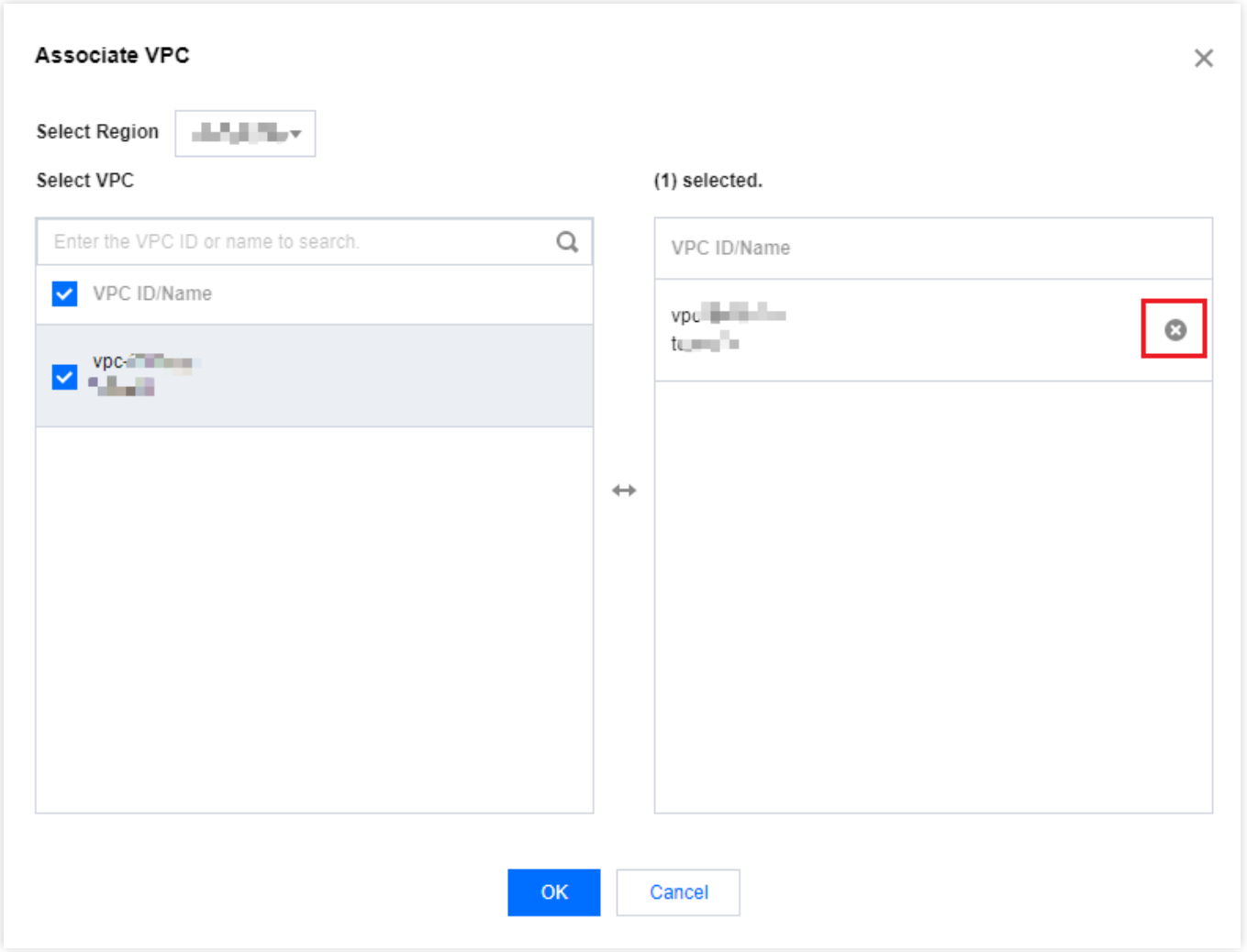


Disassociate with VPC

1. Click Associate with VPC.



2. Locate the VPC to delete, click the × button, then click OK to delete.

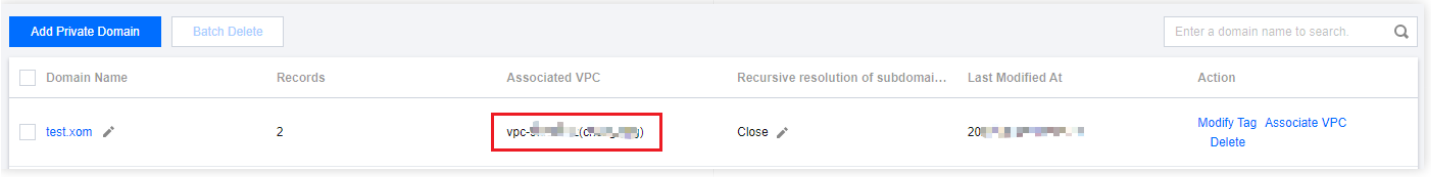


Verifying DNS Record Validation

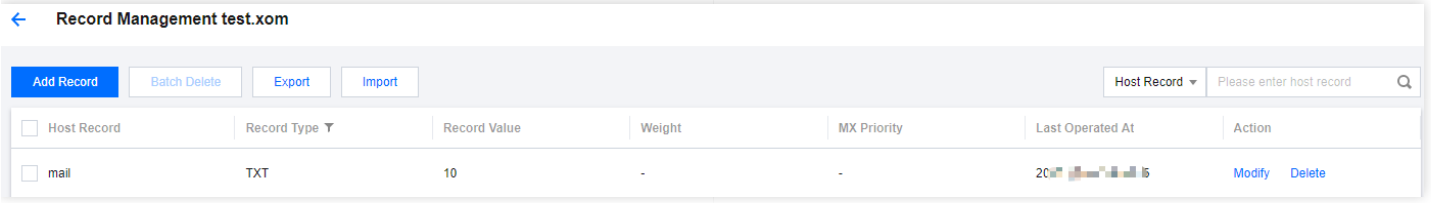
Prerequisites

After adding a domain, setting up resolution records, and associating the domain with a VPC, you can follow the steps below to verify if the settings have taken effect.

For example, your domain, yourdomain.com, is associated with a VPC in the Caloocan region.



The domain yourdomain.com has the following resolution records set.



Directions

You can check the domain's resolution on a server within the corresponding VPC in Caloocan. Log in to the server in the Caloocan VPC and use the dig command to verify. The verification result is shown in the figure below.

```
[root@VM_17_14_centos ~]# dig aa.yehao.com

; <<>> DiG 9.9.4-RedHat-9.9.4-74.el7_6.1 <<>> aa.yehao.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1084
;; flags: qr rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;aa.yehao.com.                  IN      A

;; ANSWER SECTION:
aa.yehao.com.           600     IN      A       2.2.2.2

;; Query time: 0 msec
;; SERVER: 183.60.83.19#53(183.60.83.19)
;; WHEN: Tue Nov 05 15:53:12 CST 2019
;; MSG SIZE  rcvd: 57
```

As seen on the server within the VPC, yehao.com can be resolved to a private network address.

Deleting Private Domain

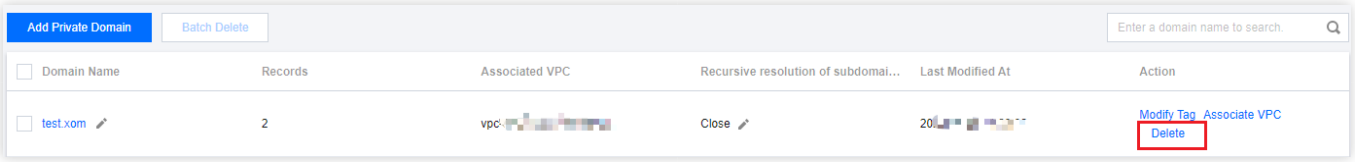
Overview

This document describes how to delete a private domain.

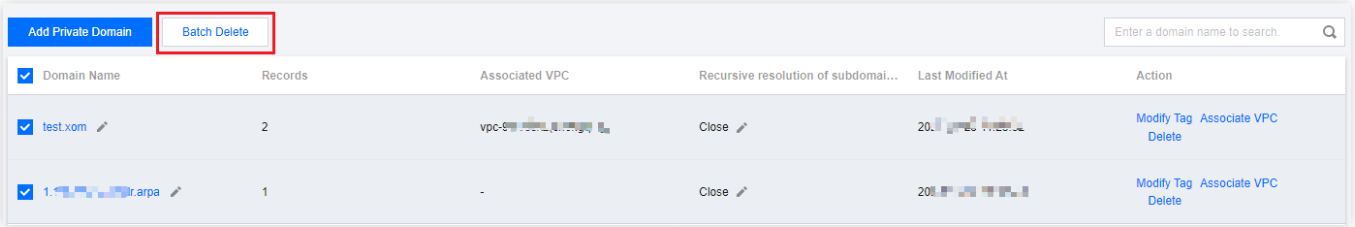
Directions

> Note:
> - After a private domain is deleted, the associated VPC is also unbound. Proceed with caution.
> - After a private domain is deleted, all resolution records under that domain are also deleted. Proceed with caution.

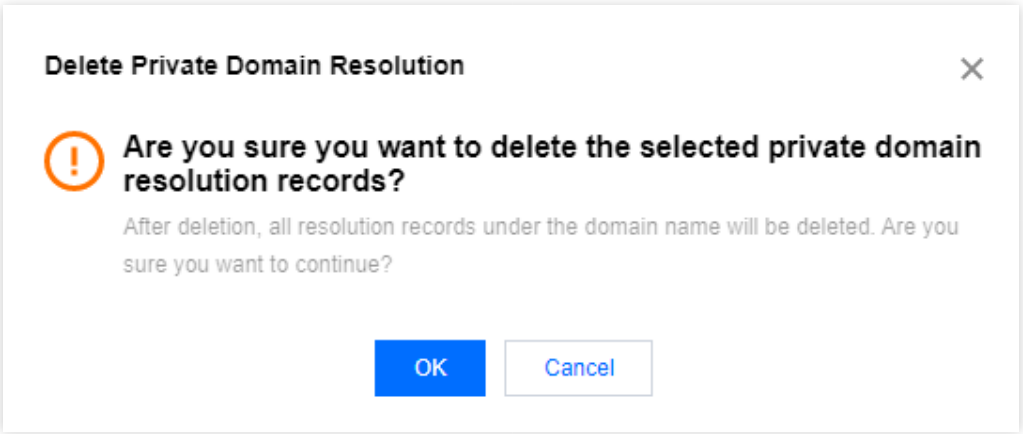
1. Log in to the Converge Cloud VPC DNS console and enter the VPC DNS List management page.
2. In the "VPC Domain Name Resolution List", select the private domains you want to delete, and click the Delete option in the action column.



3. If you want to delete multiple private domains, you can select multiple records and click the Batch Delete button to delete them.



4. In the pop-up "Delete Private Domain" window, click Confirm to complete the deletion.


Delete Private Domain Resolution
Are you sure you want to delete the selected private domain resolution records?
After deletion, all resolution records under the domain name will be deleted. Are you sure you want to continue?
OK
Cancel

Enabling Subdomain Recursion

Overview

Converge Cloud VPC DNS supports the subdomain recursive resolution feature. After enabling this feature, when a client initiates a DNS request, if the corresponding subdomain resolution record is not configured within the private domain, Converge Cloud VPC DNS will return the corresponding record based on whether subdomain recursive resolution is enabled or not.

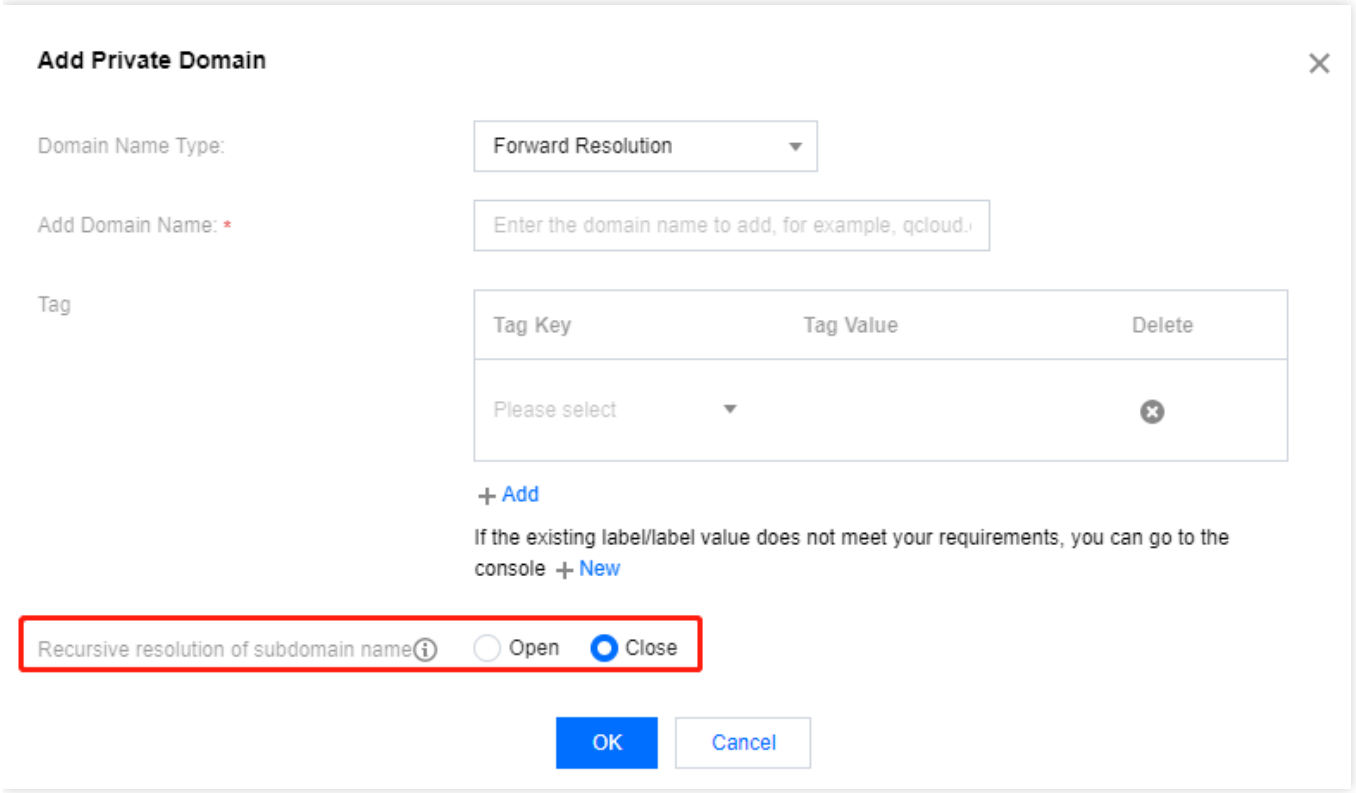
> Note:
> After enabling the subdomain recursive resolution feature for a Converge Cloud VPC DNS private domain, if a record in the domain does not hit, the resolution result of the record on the public network will be returned.

Directions

Enabling Subdomain Recursive Resolution

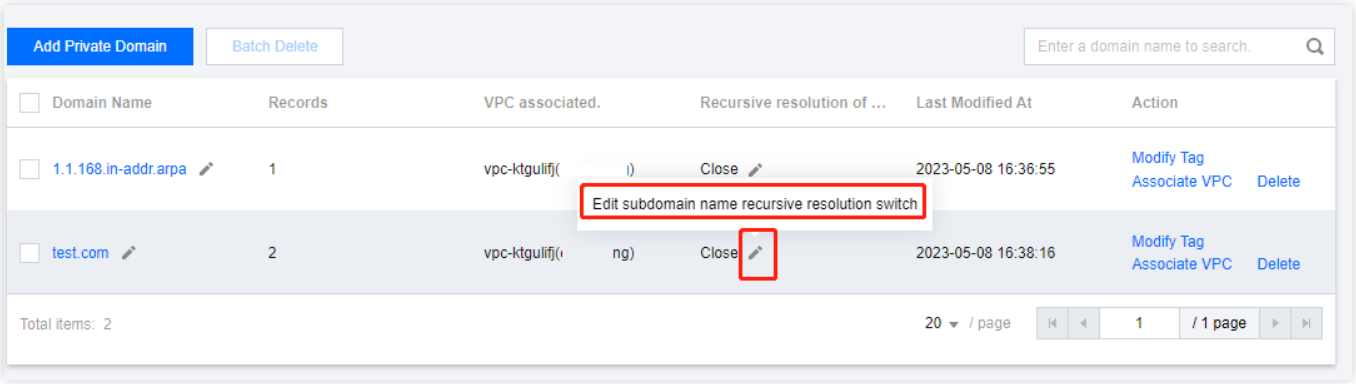
Method 1

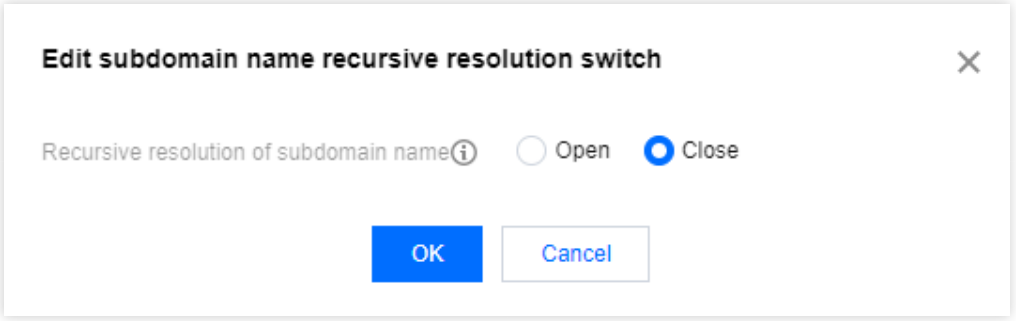
When first adding a private domain, you can choose to enable subdomain recursive resolution. For detailed instructions, please refer to Create Private Domain.



Method 2

1. In the Private Domain List, select the private domain you want to enable subdomain recursive resolution for and click the Subdomain Recursive Resolution button. In the pop-up window, select Enable subdomain recursion.





Forwarding List

Overview

VPCDNS supports forwarding the DNS request for a domain name to a designated DNS server for resolution. After a new forwarding rule is created, when a client initiates a DNS request for that domain name:

- VPCDNS first determines whether a resolution record is configured. If yes, VPCDNS intercepts the request for resolution; if not, VPCDNS determines whether a forwarding rule is configured.
- If a forwarding rule is configured, the forwarded DNS performs the resolution; if no forwarding rule is configured and subdomain recursion is enabled, the default recursive resolution is used; otherwise, the request is discarded.

Prerequisites

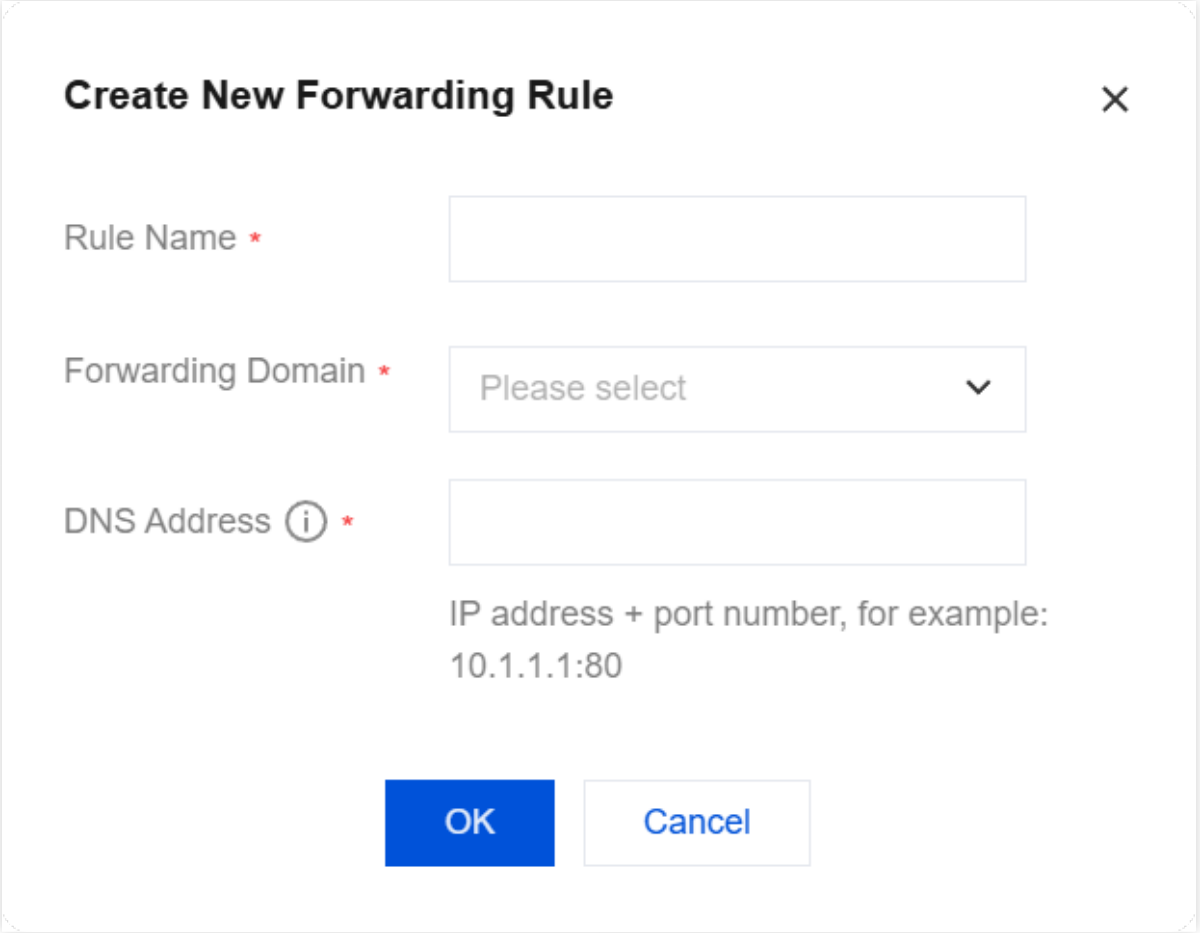
A private domain name has been created.

Directions

Creating a Forwarding Rule

1. Log in to the VPCDNS console and go to the Forwarding List management page.

2. Click Create New Forwarding Rule, and configure the rule name, forwarding domain name, and corresponding DNS address in the dialog box.

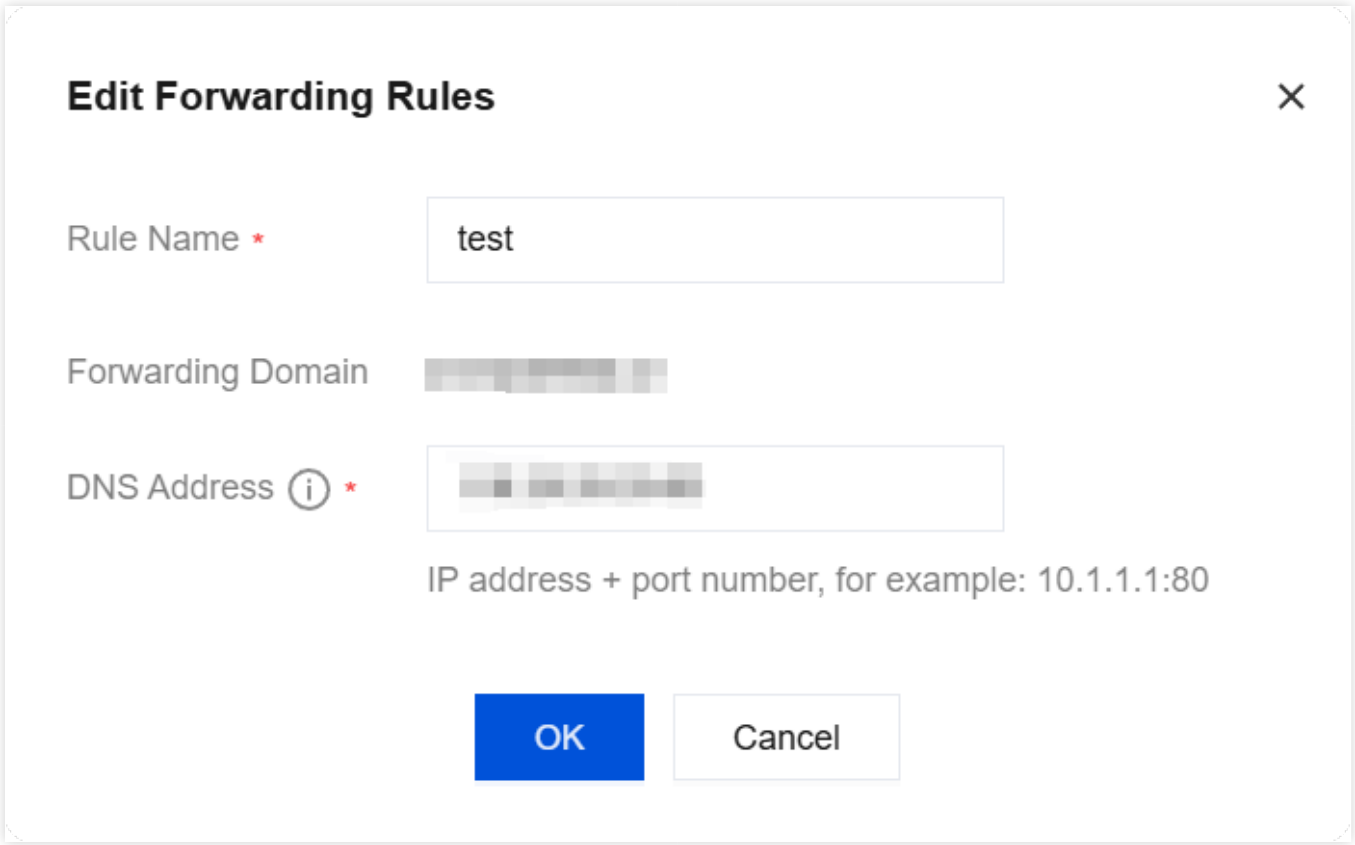


> Note:
>
> Up to five DNS addresses can be configured.

3. Click OK. The forwarding rule is created.

Modifying a Forwarding Rule

1. On the Forwarding List management page, click Modify in the Operation column.
2. Modify the rule name and corresponding DNS address.



Deleting a Forwarding Rule

1. On the Forwarding List management page, click Delete in the Operation column.
2. Click OK in the pop-up dialog box. The forwarding rule is deleted.

Cloud Access Management Overview

Cloud Access Management (CAM) is used to manage the access permissions for resources under Converge Cloud accounts. Through CAM, you can control which sub-accounts have access to which resources using identity management and policy management.

For example, multiple VPC DNS resources are under your account and deployed in different projects. In that case, you can bind an authorization policy to the administrator of project A to strengthen permission control and authorize resources. This policy specifies that only this administrator can manage the VPC DNS resources under project A.

If you do not need to manage CAM related to VPCDNS resources for sub-accounts, you can skip this section. Skipping these parts will not affect your understanding and use of the rest of the document.

Basic Concepts of CAM

The root account authorizes sub-accounts by binding policies. The policy settings can be precise to the API, Resource, User/User Group, Allow/Deny, and Condition dimensionality.

1. Account

- Root Account
  The fundamental entity for cloud resource ownership and pay-as-you-go resource usage. It is capable of logging in to Converge Cloud services.
- Sub-account
  Created by the root account, with a specific identity ID and credential, and able to log in to the Converge Cloud console. A root account can create multiple sub-accounts (users). A sub-account, by default, does not own resources and must be authorized by its associated root account.
- Identity Credential
  Includes logging in credentials and access certificates. Logging in Credentials refers to a user's login name and password. Access Certificates refer to Cloud API keys (SecretId and SecretKey).

2. Resource and Permission

- Resource
  An object operated within cloud services, such as a cloud server instance, COS bucket, and VPC instance.
- Permission

Refers to allowing or denying certain users' execution of certain actions. By default, the root account has access permissions to all resources under its name, while sub-accounts do not have access to any resources under the root account.

- Policy
  A syntax rule that defines and describes one or more permissions. The root account performs authorization by associating policies with users/user groups.

For more information, see the CAM Overview.

Documentation

Target	Link
Understand the Relationship Between Policy and User	Policy management
Understand the Basic Policy Structure	Policy syntax

Authorizable Resource Types

Authorizable VPC DNS Resource Types in CAM

Resource-level permissions refer to the ability to specify which resources a user can perform operations on. Most VPC DNS APIs support resource-level authorization, which means that VPC DNS operations allow resource-level permissions to control which specific resources are permissible for user use.

Resource Type	Resource Description Method in Authorization Policies
Private Domain	qcs::vpcdns::$account:privatezone/$zoneid

Where:

- All $account should be the AccountId of the resource owner, which defaults to ownerUin and can be left blank.
- All $zoneid should be the ID of a private domain, or "*".
  And so on.

APIs Authorizable for VPC DNS in CAM

In CAM, you can authorize the following actions for a VPC DNS resource.

Related to Private Domain

API Operations	Resource Description	API Description
DescribeVpcDnsDomainList	Retrieve VPC DNS domain list.	qcs::vpcdns::$account:privatezone/$zoneid
CreateVpcDnsDomain	Create VPC DNS domain.	* authenticate only the API.
DeleteVpcDnsDomain	Delete VPC DNS domain.	qcs::vpcdns::$account:privatezone/$zoneid
CreateVpcDnsDomainRemark	Create VPC DNS domain remarks.	qcs::vpcdns::$account:privatezone/$zoneid
DescribeVpcDnsRecordList	Retrieve VPC DNS record list.	qcs::vpcdns::$account:privatezone/$zoneid

API Operations	Resource Description	API Description
CreateVpcDnsRecord	Create VPC DNS record.	qcs::vpcdns::$account:privatezone/$zoneid
ModifyVpcDnsRecord	Modify VPC DNS record.	qcs::vpcdns::$account:privatezone/$zoneid
DeleteVpcDnsRecord	Delete VPC DNS record.	qcs::vpcdns::$account:privatezone/$zoneid
BindVpcDnsDomain	Associate with VpcId.	qcs::vpcdns::$account:privatezone/$zoneid
ModifyVpcDnsDomain	Modify VPC DNS domain.	qcs::vpcdns::$account:privatezone/$zoneid

Policy Examples

Full Read-Write Policy of VPCDNS

- Authorize sub-account full management permission (creating, managing, etc.) for the VPC DNS service.

- Policy name: QcloudVPCDNSFullAccess

```
{
  "version": "2.0",
  "statement": [
     {
        "action": [
           "vpcdns:*"
        ],
        "resource": "*",
        "effect": "allow"
     }
  ]
}
```

Read-only Policy of VPC DNS

- Authorize sub-account read-only access to VPC DNS (i.e., the permission to view all resources under VPC DNS), but the sub-account cannot create, update, or delete them. In the console, the prerequisite to manipulate a resource is the ability to view that resource; therefore, we recommend you grant the sub-account full read access to VPC DNS.

- Policy name: QcloudVPCDNSReadOnlyAccess

```
{
  "version": "2.0",
  "statement": [
     {
        "action": [
           "name/vpcdns:Describe*"
        ],
        "resource": "*",
```

```
            "effect": "allow"
        }
    ]
}
```

FAQs
FAQs

Do I need to modify the DNS service configuration when using VPCDNS?

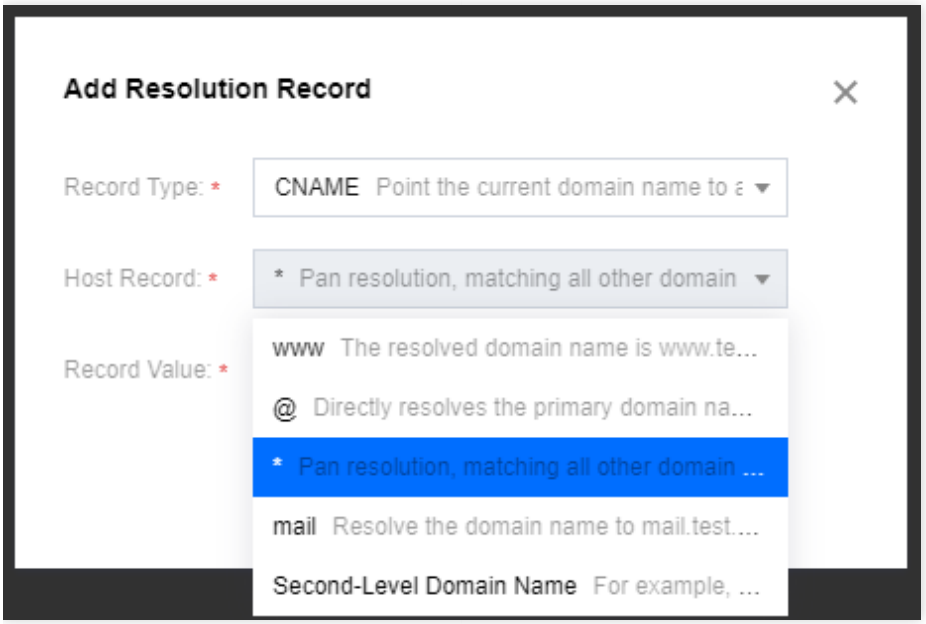
No. Tenant endpoint customers can enter the VPCDNS console to add a private domain, set resolution records, and associate with the VPC that needs to take effect.

If both VPCDNS and public network DNS use the same domain, will the resolution be overwritten?

When your domain is not associated with any VPC, it will not overwrite the existing domain on the public network. Once the domain is associated with a VPC, accessing the domain within the corresponding VPC will overwrite the existing DNS in the public network.

Does VPCDNS support wildcard resolution?

Yes. Currently, except for the MX record type, other record types support setting wildcard resolution. The specific operation is to select for the host record.



Glossary

A Record

Points the domain to an IPv4 address.

AAAA Record

Points the domain to an IPv6 address.

MX Record

Points the domain to an email server address.

TXT Record

The text length is less than 255, and it is commonly used for SPF Record (anti-spam).

CNAME Record

Points the domain to another domain.

PTR Record

Points an IP address to a domain.

Host

It generally refers to the prefix of the domain. For example, if your registered domain is example.com, then www in www.example.com is the prefix of your domain example.com.

Record Value

When setting DNS records, the specific address to which the domain points has different formats for different record types.

MX Priority

Priority can be set for different email servers, with values ranging from 1–50. The smaller the value, the higher the priority.

Weight

When a domain is set with multiple records pointing to different addresses, you can distribute the proportion of each line's resolution through weights. The proportion resolved to a certain record: the weight of that record / the sum of the weights of all records for that subdomain.

VPC

VPC (Virtual Private Cloud) is a dedicated cloud network space built on Converge Cloud, providing network services for your resources on Converge Cloud. Different VPCs are completely logically isolated. You can customize your network environment, route tables, security policies, etc. Additionally, VPC supports multiple ways to connect to the Internet, other VPCs, and your local data center, helping you effortlessly deploy cloud networks.

API Documentation
VPC DNS (VPCDNS)(vpcdns) version(2019-10-25)
API overview

API version

V3

vpcdns

Interface name	Interface function
BindVpcDnsDomain	BindVpcDnsDomain
CreateBackupRecord	CreateBackupRecord
CreateForwardRule	CreateForwardRule
CreateVpcDnsDomain	CreateVpcDnsDomain
CreateVpcDnsDomainRemark	CreateVpcDnsDomainRemark
CreateVpcDnsRecord	CreateVpcDnsRecord
DeleteBackupRecord	DeleteBackupRecord
DeleteForwardRule	DeleteForwardRule
DeleteVpcDnsDomain	DeleteVpcDnsDomain
DeleteVpcDnsRecord	DeleteVpcDnsRecord
DescribeBackupRecordList	DescribeBackupRecordList
DescribeExportFileUrl	DescribeExportFileUrl
DescribeForwardRuleList	DescribeForwardRuleList
DescribeImportTemplateUrl	DescribeImportTemplateUrl

Interface name	Interface function
DescribeUploadUrl	DescribeUploadUrl
DescribeVpcDnsDomainList	DescribeVpcDnsDomainList
DescribeVpcDnsRecordList	DescribeVpcDnsRecordList
ImportRecords	ImportRecords
ModifyForwardRule	ModifyForwardRule
ModifyVpcDnsDomain	ModifyVpcDnsDomain
ModifyVpcDnsRecord	ModifyVpcDnsRecord

CallingMethod signaturev1

Converge Cloud authenticates each access request, i.e. each request needs to include authentication information (Signature) in the common parameters to verify the identity of the requester.
The Signature is generated by the security credentials which include SecretId and SecretKey. If you don't have the security credentials yet, go to the API Key page to apply for them; otherwise, you cannot invoke the API.

1. Applying for Security Credentials

Before using the API for the first time, go to the API Key page to apply for security credentials.
Security credentials consist of SecretId and SecretKey:

- SecretId is used to identify the API requester.
- SecretKey is used to encrypt the signature string and verify it on the server.
- You must keep your security credentials private and avoid disclosure.

You can apply for the security credentials through the following steps:

1. Log in to the Converge Cloud Console.
2. Go to the Cloud API Key page.
3. On the API Key Management page, click Create Key to create a SecretId/SecretKey pair.

Note: Each account can have up to two pairs of SecretId/SecretKey.

2. Generating a Signature

With the SecretId and SecretKey, a signature can be generated. The following describes how to generate a signature:

Assume that the SecretId and SecretKey are:

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

Note: This is just an example. For actual operations, please use your own SecretId and SecretKey.

Take the Cloud Virtual Machine's request to view the instance list (DescribeInstances) as an example. When

you invoke this API, the request parameters may be as follows:

Parameter name	Description	Parameter value
Action	Method name	DescribeInstances
SecretId	Key ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	luzon
InstanceIds.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

First, sort all the request parameters in an ascending lexicographical order (ASCII code) by their names. Notes: (1) Parameters are sorted by their names instead of their values; (2) The parameters are sorted based on ASCII code, not in an alphabetical order or by values. For example, InstanceIds.2 should be arranged after InstanceIds.12. You can complete the sorting process using a sorting function in a programming language, such as the ksort function in PHP. The parameters in the example are sorted as follows:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'luzon',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

When developing in another programming language, you can sort these sample parameters and it will work as long as you obtain the same results.

2.2. Concatenating a Request String

This step generates a request string.
Format the request parameters sorted in the previous step into the form of "parameter name"="parameter value". For example, for the Action parameter, its parameter name is "Action" and its parameter value is "DescribeInstances", so it will become Action=DescribeInstances after formatted.
Note: The "parameter value" is the original value but not the value after URL encoding.

Then, concatenate the formatted parameters with "&". The resulting request string is as follows:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. Concatenating the Signature Original String

This step generates a signature original string.
The signature original string consists of the following parameters:

1. HTTP method: POST and GET modes are supported, and GET is used here for the request. Please note that the method name should be in all capital letters.
2. Request host: the domain name of the request to view the list of instances (DescribeInstances) is cvm.convergecloud.com. The actual request domain name varies by the module to which the API belongs. For more information, see the instructions of the specific API.
3. Request path: The request path in the current version of Cloud API is fixed to / .
4. Request string: the request string generated in the previous step.

The concatenation rule of the signature original string is: Request method + request host + request path + ? + request string

The concatenation result of the example is:

```
GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. Generating a Signature String

This step generates a signature string.
First, use the HMAC-SHA1 algorithm to sign thesignature original stringobtained in the previous step, and then encode the generated signature using Base64 to obtain the final signature.

The specific code is as follows with the PHP language being used as an example:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit
=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPL
E&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The final signature is:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

When developing in another programming language, you can sign and verify the original in the example above and it works as long as you get the same results.

3. Encoding a Signature String

The generated signature string cannot be directly used as a request parameter and must be URL encoded.

For example, if the signature string generated in the previous step is EliP9YW3pW28FpsEdkXt/+WcGeI= , the final signature string request parameter (Signature) is EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d, which will be used to generate the final request URL.

**Note: If your request method is GET, or the request method is POST and the Content-Type is application/x-www-form-urlencoded, then all the request parameter values need to be URL encoded (except the parameter key and the symbol of =) when sending the request. Non-ASCII characters need to be encoded with UTF-8 before URL encoding. **

Note: The network libraries of some programming languages automatically URL encode all parameters, in which case there is no need to URL encode the signature string; otherwise, two rounds of URL encoding will cause the signature to fail.

Note: Other parameter values also need to be encoded using RFC 3986. Use %XY in percent-encoding for special characters such as Chinese characters, where "X" and "Y" are hexadecimal characters (0-9 and uppercase A-F), and using lowercase will cause an error.

4. Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error code	Error description
AuthFailure.SignatureExpire	The signature is expired

Error code	Error description
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Converge Cloud SDK 3.0, Swhich encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- Node

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: https://cvm.convergecloud.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=luzon&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.convergecloud.com/?");
        // When signing, the parameters need to be sorted in lexicographical order. TreeMap is used here
to guarantee the correct order.
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.convergecloud.com/?");
        // There is no requirement for the order of the parameters in the actual request URL.
        for (String k : params.keySet()) {
            // The request string needs to be URL encoded. As the Key is all in English letters, only the valu
e is URL encoded here.
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
```

```
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automati
c sorting
        // A random number should be used when actually calling, for example: params.put("Nonce", new
Random().nextInt(java.lang.Integer.MAX_VALUE));
        params.put("Nonce", 11886); // Common parameter
        // The current time of the system should be used when actually calling, for example:   params.put
("Timestamp", System.currentTimeMillis() / 1000);
        params.put("Timestamp", 1465185768); // Common parameter
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // Common parameter
        params.put("Action", "DescribeInstances"); // Common parameter
        params.put("Version", "2017-03-12"); // Common parameter
        params.put("Region", "luzon"); // Common parameter
        params.put("Limit", 20); // Business parameter
        params.put("Offset", 0); // Business parameter
        params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // Common parameter
        System.out.println(getUrl(params));
    }
}
```

Python

Note: If running in a Python 2 environment, the following requests dependency package must be installed first: pip install requests .

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
```

```
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.convergecloud.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'luzon',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # n actual invocation would occur here, which may incur fees after success
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

signaturev3

Converge Cloud The API will authenticate each access request, meaning that each request must include signature information (Signature) in the public request parameters to verify the requester's identity.
The signature information is generated by the security credentials, which include SecretId and SecretKey . If the user does not have security credentials, please apply for them on the Cloud API Key page, otherwise they cannot call the Cloud API interfaces.

1. Apply for security credentials.

Before using the Cloud API for the first time, please go to the Cloud API Key page to apply for a security credential.
Security credentials include SecretId and SecretKey:

- SecretId The security credentials are used to identify the identity of the API caller.
- SecretKey The secret key is used to encrypt and sign the string for server-side verification of the signature.
- Users must strictly protect their security credentials and avoid disclosure them.

The specific steps to apply for security credentials are as follows:

1. Log in to Converge Cloud Administration Center Console
2. Go to the console page for cloud API keys
3. On the Cloud API Key page, click New to create a pair of SecretId/SecretKey

Note: Developer accounts can have up to two pairs SecretId / SecretKey.

2. TC3-HMAC-SHA256 signature method

Note: For GET method, only Content-Type: application/x-www-form-urlencoded protocol format is supported. For POST method, Content-Type: application/json and Content-Type: multipart/form-data are currently supported. By default, all business interfaces are supported in json format. Only specific business interfaces are supported in multipart format. In this case, the interface cannot be called in json format. Refer to the specific business interface document description.

The following takes Cloud Virtual Machine querying Region-1 instance list as an example to introduce the signature calculation process step by step. We only use two parameters for the query instance list: Limit and Offset, called using the GET method.

Assume that the SecretId and SecretKey of the user are AKIDz 8krbsJ 5yKBZQpn 74WFkmLPx 3EXAMPLE and Gu 5t9xGARNpq 86cd 98joQYCN3EXAMPLE, respectively.

2.1. splice specification request string

Concatenate CanonicalRequest strings in the following format:

```
CanonicalRequest =
  HTTPRequestMethod + '\n' +
  CanonicalURI + '\n' +
  CanonicalQueryString + '\n' +
  CanonicalHeaders + '\n' +
  SignedHeaders + '\n' +
  HashedRequestPayload
```

- HTTPRequestMethod: HTTP request methods (GET, POST), GET in this example;
- CanonicalURI: URI parameter, API 3.0 fixed to forward slash (/);
- CanonicalQueryString: Query string in the URL that initiates the HTTP request. For POST requests, it is fixed as an empty string. For GET requests, it is the question mark (?) in the URL. The following string content, this example value is: Limit=10&Offset=0. Note: CanonicalQueryString needs to be URL encoded.
- CanonicalHeaders: The header information of participating signature, at least including host and content-type, can also be added with customized header to improve the uniqueness and security of its own request. Splicing rules: 1) head key and value are uniformly converted to lowercase, and the first and last spaces are removed, and spliced according to the format of key:value\n;2) multiple heads are spliced according to the dictionary sorting of head key (lowercase). In this example: content-type: application/x-www-form-urlencoded\nhost: cvm.convergecloud.com\n
- SignedHeaders: The header information involved in the signature indicates which headers participate in the signature of this request, and the header content contained in CanonicalHeaders is one-to-one corresponding. content-type and host are required headers. constructed as: 1) The head key is uniformly converted to lowercase; 2) Multiple header keys (lowercase) are concatenated according to dictionary sorting and separated by semicolons (;). In this case: content-type;host
- HashedRequestPayload: The hash value of the request body, calculated as Lowercase(HexEncode(Hash.SHA256(RequestPayload))), SHA256 hash the entire payload of the HTTP request body, then hexadecimal encoding, and finally converting the encoding string to lowercase letters. Note: For GET requests, RequestPayload is fixed as an empty string. For POST requests, RequestPayload is the payload of the HTTP request body.

According to the above rules, the resulting canonical request string in the example is as follows (for clarity, \nnewlines are replaced by a new line printed separately):

```
GET
```

```
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.convergecloud.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. Concatenate the string to be signed

Concatenate the string to be signed in the following format:

```
StringToSign =
  Algorithm + \n +
  RequestTimestamp + \n +
  CredentialScope + \n +
  HashedCanonicalRequest
```

- Algorithm: Signature algorithm, currently fixed as TC3-HMAC-SHA256;
- RequestTimestamp: Request timestamp, i.e. the X-TC-Timestamp value of the request header. The request in the example above is 1539084154;
- CredentialScope: Credential range in the format Date/service/tc3_request, containing date, requested service and termination string (tc3_request). **Date is the date of UTC standard time, the value needs to be consistent with UTC standard time converted by common parameter X-TC-Timestamp **;service is the product name, which must be consistent with the product domain name called, such as cvm. The above example request, the value is 2018-10-09/cvm/tc3_request;
- HashedCanonicalRequest: The hash value of the canonical request string obtained by concatenating the above steps is calculated by Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))).

note:

> 1. Date must be calculated from timestamp X-TC-Timestamp and time zone UTC+0. Adding local time zone information to the system, such as East Zone 8, will cause calls to succeed during the day and night, but calls to fail in the early hours of the morning. Assuming the timestamp is 1551113065, the time in the East Eight District is 2019-02-26 00:44:25, but the calculated Date with UTC+0 should be 2019-02-25 instead of 2019-02-26.
> 2. Timestamp must be the current system time and ensure that the system time and standard time are synchronized. If the difference exceeds five minutes, it will fail. If it is not synchronized with the standard time for a long time, it may cause the request to fail after running for a period of time (return signature expiration error).

According to the above rules, the string to be signed in the example is as follows (for clarity,\nnewlines are

replaced by new lines printed):

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. calculate the signature

1. Compute the derived signature key, pseudocode as follows

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey: Original SecretKey;
- Date: That is, the Date field information in Credential, as shown in the example above, is 2018-10-09.;
- Service: That is, the Service field information in Credential, as shown in the example above, is cvm;

2. Calculate the signature, pseudocode as follows

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning: That is, the derivative signature key calculated above;
- StringToSign: That is, the character string to be signed calculated in step 2;

2.4. Splicing Authorization

Splice as follows Authorization:

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm: Signature method, fixed as TC3-HMAC-SHA256;
- SecretId: SecretId in key pair;
- CredentialScope: See above, Scope of Voucher;
- SignedHeaders: See above, header information for participating signatures;
- Signature: signature value

According to the above rules, the values obtained in the example are:

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

The final complete invocation information is as follows:

```
https://cvm.convergecloud.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.convergecloud.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: luzon
```

3. signing failed

According to the actual situation, there are the following signature failure error codes, please handle according to the actual situation

error code	error description
AuthFailure.SignatureExpire	signature expiration
AuthFailure.SecretIdNotFound	key is not present
AuthFailure.SignatureFailure	signature error
AuthFailure.TokenFailure	token error
AuthFailure.InvalidSecretId	Illegal key (not cloud API key type)

4. Signature Demo

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.convergecloud.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.convergecloud.com";
        String region = "luzon";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // Pay attention to the time zone, otherwise it is easy to make mistakes.
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* Step 1: Assemble the specification request string *************
```

```
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* Step 2: Stitching the string to be signed *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* Step 3: Calculate the signature *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* Step 4: Splicing Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

```
# -*- coding: utf-8 -*-
```

```
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# key parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.convergecloud.com"
endpoint = "https://" + host
region = "luzon"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* Step 1: Assemble the specification request string *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* Step 2: Stitching the string to be signed *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)
```

```
# ************* Step 3: Calculate the signature *************
# Compute signature digest function
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* Step 4: Splicing Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# Common parameters added to request header
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

request

1. Service Address

Region refers to the geographical area of a physical data center. Converge Cloud ensures complete isolation between different regions, providing maximum stability and fault tolerance. To reduce access latency and improve download speed, it is recommended to choose the region closest to your customers.

You can view the complete region list by using the API interface "DescribeRegions".

2. Communications Protocol

All the Converge Cloud APIs communicate via HTTPS, providing highly secure communication tunnels.

3. Request Methods

Supported HTTP request methods:

- POST (recommended)
- GET

PThe Content-Type types supported by POST requests:

- application/json (recommended). The TC3-HMAC-SHA256 signature algorithm must be used.
- application/x-www-form-urlencoded. The HmacSHA1 or HmacSHA256 signature algorithm must be used.
- multipart/form-data (only supported by certain APIs). You must use TC3-HMAC-SHA256 to calculate the signature.

The size of a GET request packet is up to 32 KB. The size of a POST request is up to 1 MB when the HmacSHA1 or HmacSHA256 signature algorithm is used, and up to 10 MB when TC3-HMAC-SHA256 is used.

4. Character Encoding

Only UTF-8 encoding is used.

result

Response for Successful Requests

For example, when calling CAM API (version: 2017-03-12) to view the status of instances (DescribeInstancesStatus), if the request has succeeded, you may see the response as shown below:

```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```

- The API will return Response, which contains RequestId, as long as it processes the request. It does not matter if the request is successful or not.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.
- Except for the fixed fields, all fields are action-specified. For the definitions of action-specified fields, see the corresponding API documentation. In this example, TotalCount and InstanceStatusSet are the fields specified by the API DescribeInstancesStatus. 0 TotalCount means that the requester owns 0 CVM instance so the InstanceStatusSet is empty.

Response for Failed Requests

If the request has failed, you may see the response as shown below:

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

- The presence of the Error field indicates that the request has failed. A response for a failed request will include Error, Code and Message fields.

- Code is the code of the error that helps you identify the cause and solution. There are two types of error codes so you may find the code in either common error codes or API-specified error codes.
- Message explains the cause of the error. Note that the returned messages are subject to service updates. The information the messages provide may not be up-to-date and should not be the only source of reference.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.

Common Error Codes

If there is an Error field in the response, it means that the API call failed. The Code field in Error indicates the error code. The following table lists the common error codes that all actions can return.

Error Code	Description
AuthFailure.InvalidSecretId	Invalid key (not a API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	The key does not exist.
AuthFailure.SignatureExpire	Signature expired.
AuthFailure.SignatureFailure	Signature error.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.

Error Code	Description
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	hHTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

PublicParameter

Common parameters are parameters used to identify users and interface authentication purposes. If not necessary, these parameters are not described in the separate interface document of each interface, but these parameters need to be carried in each request before the request can be normally initiated.

signature method v3

When using TC3-HMAC-SHA256 signature method, the common parameters need to be uniformly placed in the HTTP Header request header, as follows:

parameter name	type	required	described
X-TC-Action	String	Yes	The name of the operation interface. Refer to the description of the public parameter Action in the input parameters of the interface documentation for the value. For example, the value for the query instance list interface of the cloud server is DescribeInstances.
X-TC-Region	String	Yes	region parameters, Data used to identify which region you wish to operate on. The region value accepted by the interface refers to the description of the input parameter common parameter Region in the interface document. Note: Some interfaces do not need to pass this parameter, which is specifically stated in the interface documentation. In this case, even passing this parameter will not take effect.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp, which records the time when the API request was initiated. for example 1529223702. Note: If the time difference from the server exceeds 5 minutes, it will cause a signature expiration error.
X-TC-Version	String	Yes	Version of API for operation. Refer to the description of the public parameter Version in the interface document for the value. For example, the version of the cloud server. 2017-03-12.
Authorization	String	Yes	HTTP The standard authentication header field is "Authorization". For example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here, - TC3-HMAC-SHA256: Signature method, currently fixed as this value, - Credential: Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example,cvm; - SignedHeaders: The headers that contains the authentication information; content-type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for temporary certificates needs to be used in conjunction with a temporary key. Temporary keys and tokens need to be obtained by calling the access management service interface. Long-term keys do not require tokens.

signature method v1

When using HmacSHA1 and HmacSHA256 signature methods, common parameters need to be placed uniformly in the request string, as follows

parameter name	type	required	described
Action	String	Yes	The name of the interface to be operated. The value refers to the explanation of the public parameter Action in the interface documentation. For example, for the cloud server's query instance list interface, the value is DescribeInstances .
Region	String	Yes	The region parameter is used to identify which region's data you want to operate on. The region value accepted by the interface is specified in the input parameter of the public parameter Region in the interface documentation. Note: Some interfaces do not require this parameter to be passed, which will be specifically mentioned in the interface documentation. Even if this parameter is passed, it will not take effect in this case.
Timestamp	Integer	Yes	The current UNIX timestamp, which can record the time when the API request is initiated. For example, 1529223702. If the difference between the current time is too large, it will cause a signature expiration error.
Nonce	Integer	Yes	A random positive integer combined with Timestamp is used to prevent replay attacks.
SecretId	String	Yes	The SecretId is an identifier applied for the cloud API key to identify the user's identity. A SecretId corresponds to a unique SecretKey , and the SecretKey is used to generate the request signature Signature .
Signature	String	Yes	Request signature is used to verify the legitimacy of this request. Users need to calculate it based on the actual input parameters. The specific calculation method can be found in the interface authentication documentation.
Version	String	Yes	The version of the API being operated on. The value refers to the description of the public parameter Version in the interface documentation. For example, the version of the cloud server. 2017-03-12.
SignatureMethod	String	No	Signature method, currently supports HmacSHA256 and HmacSHA1. HmacSHA256 algorithm is used to verify signatures only if this parameter is specified as HmacSHA 256, otherwise HmacSHA1 is used to verify signatures.
Token	String	No	Token used for temporary certificate needs to be used together with temporary key. The temporary key and Token need to be obtained from the Access Management service invocation interface. Long-term keys do not require tokens.

zone list

A region (Region) refers to the geographical area of a physical data center. Converge Cloud Translation: To deliver verification that completely isolates different regions, ensuring maximum stability and fault tolerance between regions. In order to reduce access latency and improve download speed, it is recommended that you choose the region closest to your customers.

You can view the complete region list by using the API interface Query Region List.

vpcdns
BindVpcDnsDomain

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Associated VPC ID API

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 05:01:07.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: BindVpcDnsDomain
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	The domain name ID. Example: 3
VpcInfos	No	Yes	Array of VpcInfos	VPC information. Example:View

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.IllegalVpcInfo	Invalid VPC.
InternalError.UndefiendError	Undefined error.
InvalidParameterValue.VpcBinded	The VPC has bound to other resolution domains.
InvalidParameter.	Parameter error.
InvalidParameterValue.BackupRecordOverLimit	The number of backup records of this tenant exceeds the limit.

CreateBackupRecord

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to create a backup record.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-07-02 20:40:21.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateBackupRecord
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	Domain ID Example: 2
RecordType	Yes	No	String	Record type. Valid values: A and MX. Example: A
SubDomain	Yes	No	String	The subdomain name. Example: www
Value	Yes	No	String	The record value. Example: 3.2.4.1
MasterValue	Yes	No	String	Value of the primary record. Example: 4.3.1.1

3. Output Parameters

Parameter name	Type	Description
BackUpRecordId	Uint64	Backup record ID. Example: 3
Value	String	record value Example: 3.2.4.1
RentBackupRecordCount	Uint64	Number of disaster recovery records. Example: 34
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.	Parameter error.
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.IllegalCidr	Invalid CIDR block.
InvalidParameter.IllegalDomain	Incorrect domain name.
InvalidParameter.RecordLevelExceed	The level recorded exceeds the limit.
InvalidParameter.IllegalRecord	Invalid records.
InvalidParameter.IllegalDomainTld	Top-level domain name.
InvalidParameter.IllegalPTRRecord	Invalid PTR record.
InvalidParameterValue.CnameNotPrivateZone	You can only use CNAMEs to point to the private domains under the current account.
InvalidParameter.IllegalRecordValue	Invalid record value.

Error Code	Description
InvalidParameterValue.IllegalWeightValue	The weight value must be a number from 1 to 100.
InvalidParameterValue.IllegalTXTValue	After Punycode encoding, the length of the txt format record value must be between 1 and 255.
InvalidParameter.WeightNotSupported	The current record does not support weight settings.
FailedOperation.CreateRecordFailed	Failed to create the record.
InternalError.UndefiendError	Undefined error.
InvalidParameterValue.BackupRecordNotSupport	The current data does not support backup records.
InvalidParameterValue.RecordConflict	Record conflict.
InvalidParameterValue.RecordExist	Record already exists.
InvalidParameterValue.BackupRecordExceed	Setting more than three alternative DNS IP addresses is not supported.
InvalidParameterValue.BackupRecordOverLimit	The number of backup records of this tenant exceeds the limit.

CreateForwardRule

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Create forwarding rule

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:56:48.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateForwardRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainIdList	Yes	No	Array of String	Private domain ID Example: 3
ForwardAddress	Yes	Yes	Array of String	Forwarding IP Example: ["1.1.1.2:80", "1.2.3.4:80"]
Remark	No	No	String	Description. Example: This is a forwarding rule remark.

3. Output Parameters

Parameter name	Type	Description
RuleIdList	Array of String	Forwarding rule ID Example: ["40","41"]
CreatedAt	Datetime	Call Time Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.	Parameter value error.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.
InvalidParameterValue.RecordNotExist	This record does not exist.
UnknownParameter.	Unknown parameter.
InvalidParameterValue.RecordExist	Record already exists.
InvalidParameterValue.ForwardRuleOverLimit	Number of forwarding rules exceeds the limit of 200.

CreateVpcDnsDomain

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for creating VPC DNS domain names

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:37:27.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateVpcDnsDomain
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Domain	Yes	No	String	Domain name. Example: test.com
Tags	No	No	Array of Tag	List of tags. Example:View
DnsForwardStatus	No	No	String	Enable subdomain recursion, DISABLED: off, ENABLED: on Example: ENABLED

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
CreatedAt	Datetime	The return time of the creation request. Example: 2014-08-03 12:00:00
DomainId	Uint64	Domain ID Example: 3
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.IllegalSubnet	Invalid subnet.
InvalidParameter.IllegalDomain	Incorrect domain name.
InvalidParameter.IllegalRecord	Invalid records.
InternalError.UndefiendError	Undefined error.
InvalidParameter.IllegalDomainTld	Top-level domain name.
InvalidParameter.IllegalCidr	Invalid CIDR block.
InvalidParameter.	Parameter error.

CreateVpcDnsDomainRemark

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for creating remarks of the VPC DNS domain name

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:38:14.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateVpcDnsDomainRemark
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	The domain name ID. Example: 3
Remark	Yes	No	String	Description. Example: Remarks

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

CreateVpcDnsRecord

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for creating VPC DNS records

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:39:04.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: CreateVpcDnsRecord
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	The domain name ID. Example: 3
SubDomain	Yes	No	String	The subdomain name. Example: www
RecordType	Yes	No	String	Record types: A, AAAA, MX, CNAME, TXT, PTR, SRV, SPF Example: A
Value	Yes	No	String	The record value. Example: 1.1.1.1
Mx	No	No	Uint64	The MX priority. Example: 45

Parameter name	Required	Allow NULL	Type	Description
Weight	No	No	String	Weight. Example: 100

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The return time of the creation request. Example: 2014-08-03 12:00:00
Data	RecordId	Record ID. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.IllegalPTRRecord	Invalid PTR record.
InvalidParameterValue.RecordAAAACountExceed	The number of AAAA records for CLB exceeds 50.
InternalError.UndefiendError	Undefined error.
InvalidParameter.IllegalRecordValue	Invalid record value.
InvalidParameterValue.CnameNotPrivateZone	You can only use CNAMEs to point to the private domains under the current account.
InvalidParameterValue.RecordExist	Record already exists.
InvalidParameter.IllegalRecord	Invalid records.
InvalidParameterValue.RecordUnsupportWeight	The current record type does not support the weight.

Error Code	Description
FailedOperation.CreateRecordFailed	Failed to create the record.
InvalidParameterValue.RecordConflict	Record conflict.
InvalidParameterValue.RecordMXCountExceed	The number of MX records for CLB exceeds 50.
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameterValue.RecordTXTCountExceed	The number of TXT records for CLB exceeds 10.
InvalidParameter.	Parameter error.
InvalidParameterValue.RecordACountExceed	The number of A records for CLB exceeds 50.
InvalidParameterValue.IllegalTXTValue	After Punycode encoding, the length of the txt format record value must be between 1 and 255.

DeleteBackupRecord

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to delete a backup record.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-07-02 20:43:06.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteBackupRecord
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RecordId	Yes	No	Uint64	Collection of IDs of deleted backup records. Example: 3
DomainId	Yes	No	Uint64	Domain ID Example: 34

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.RecordNotExist	This record does not exist.
InvalidParameter.	Parameter error.
InvalidParameter.IllegalDomain	Incorrect domain name.
FailedOperation.DeleteRecordFailed	Failed to delete the record.
ResourceInUse.	The resource is in use.

DeleteForwardRule

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to delete forwarding rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2024-03-06 03:10:11.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteForwardRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RuleIdList	Yes	No	Array of String	Forwarding rule ID Example: ["12", "14"]

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00

Parameter name	Type	Description
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.	Parameter value error.
InternalError.UndefiendError	Undefined error.
InvalidParameterValue.RecordNotExist	This record does not exist.
FailedOperation.DeleteRecordFailed	Failed to delete the record.

DeleteVpcDnsDomain

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for deleting VPC DNS domain names

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:50:12.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteVpcDnsDomain
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainIds	Yes	No	String	The domain name IDs, which are separated with commas. Example: 3,2,4

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
FailedOperation.DeleteDomainFailed	Failed to delete the domain name.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

DeleteVpcDnsRecord

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for deleting VPC DNS records

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:49:11.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DeleteVpcDnsRecord
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	The domain name ID. Example: 3
RecordIds	Yes	No	String	The record IDs, which are separated with commas. Example: 3,4,5

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
FailedOperation.DeleteRecordFailed	Failed to delete the record.
InvalidParameterValue.RecordNotExist	This record does not exist.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

DescribeBackupRecordList

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to query the backup record list.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-07-02 20:38:38.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeBackupRecordList
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	Domain ID Example: 39
RecordType	Yes	No	String	Record type. Valid value: A. Example: A
SubDomain	Yes	No	String	The subdomain name. Example: www

3. Output Parameters

Parameter name	Type	Description
BackupRecords	Array of BackupRecord	Backup record. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameter.	Parameter error.

DescribeExportFileUrl

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Asynchronous API. This API needs to be called in polling. If an exception or a normal URL is returned, the polling can be stopped.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 05:05:04.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeExportFileUrl
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	String	The domain name ID. Example: 3
FileType	Yes	No	String	File types: xlsx, xls, csv Example: csv
Filters	No	No	Array of DomainListFilters	Filter parameters Example:View

3. Output Parameters

Parameter name	Type	Description
Url	String	Download link for the exported file. If it is an empty string, continue to poll. Example: vpcdns-records-storage-1255000002.cos.region1.csp.xx.cn/export/zone-110000000056_15_test2.com_GDPvMrElxX.csv?sign=???
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.
InternalError.	Internal Error
InvalidParameter.InvalidZoneFileFormat	The file type is not supported.

DescribeForwardRuleList

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to pull the forwarding rule list.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:58:19.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeForwardRuleList
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Filters	No	No	Array of RuleListFilters	DomainName, ForwardIp and DomainId. Example:View
Offset	Yes	No	Int64	Offset. Example: 0
Limit	Yes	No	Int64	Length. Example: 20

3. Output Parameters

Parameter name	Type	Description
Total	Int64	Total number Example: 3
ForwardRuleList	Array of ForwardRuleList	Forwarding rule list. Example:View
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.	Parameter value error.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

DescribeImportTemplateUrl

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Querys file import record template

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 05:06:01.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeImportTemplateUrl
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.

3. Output Parameters

Parameter name	Type	Description
TemplateUrl	String	Template file URL Example: https://-x?q-sign-algorithm=???-time=1624415309;1624422509&q-key-time=1624415309;1624422509&q-header-list=&q-url

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.UndefiendError	Undefined error.
InternalError.	Internal Error

DescribeUploadUrl

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Obtains the upload link for the imported file

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 05:07:45.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeUploadUrl
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	String	The domain name ID. Example: 4
FileType	Yes	No	String	File types: xlsx, xls, csv Example: xlsx

3. Output Parameters

Parameter name	Type	Description

Parameter name	Type	Description
SignedUrl	String	Signed upload link address Example: vpcdns-records-storage-1255000002.cos.region1.csp.yfm4.fsphere.cn/import/zone-110000000056_15_test2.com_0lFxMGLw.xlsx?sign=???
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.	Parameter error.

DescribeVpcDnsDomainList

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for pulling a VPC DNS list of domain names

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:32:24.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeVpcDnsDomainList
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Limit	No	No	Uint64	Length. Example: 20
Offset	No	No	Uint64	Offset. Example: 0
Filters	No	No	Array of DomainListFilters	Filters. Example:View

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
Info	DomainCountInfo	The number of domain names. Example:View
Domains	Array of DomainDetail	The domain. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

DescribeVpcDnsRecordList

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for pulling a VPC DNS list of records

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:34:40.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: DescribeVpcDnsRecordList
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
Limit	No	No	Uint64	Length. Example: 20
Offset	No	No	Uint64	Offset. Example: 0
Filters	No	No	Array of RecordListFilters	Filters. Example:View
DomainId	Yes	No	Uint64	The domain name ID. Example: 3

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
Info	RecordListCountInfo	The number of records. Example:View
Records	Array of RecordDetail	The resolution records. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.

ImportRecords

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

Complete the import operation and return the import result. Importing is an asynchronous process: (1) Call DescribeUploadUrl to get the upload link for the imported file. -> (2) The frontend puts the file into the link. -> (3) Call ImportRecords to complete the import.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 05:13:12.

The interface is both signature verification and authentication.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ImportRecords
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	String	The domain name ID. Example: 3
FileType	Yes	No	String	File types: xlsx, xls, csv Example: xlsx

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
SuccessfulCount	Uint64	Success count Example: 12
FailedRecords	Array of FailedRecordDetail	Adds details. Example:View
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InternalError.UndefiendError	Undefined error.
FailedOperation.CreateRecordFailed	Failed to create the record.
InvalidParameter.	Parameter error.
InvalidParameter.ImportedFileExpired	The file is expired
InvalidParameterValue.InvalidZoneFileSize	The file size exceeds the limit.
InternalError.FetchCOSFileError	Pulling the COS file failed.
InternalError.	Internal Error

ModifyForwardRule

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

This API is used to modify forwarding rules.

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:59:40.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyForwardRule
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
RuleId	Yes	No	String	Forwarding rule ID Example: 20
ForwardAddress	Yes	Yes	Array of String	Forwarding rule ID Example: ["1.1.1.2:80", "1.2.3.4:80"]
Remark	No	Yes	String	Description. Example: This is a remark.

3. Output Parameters

Parameter name	Type	Description
RuleId	String	Forwarding rule ID Example: 12
ForwardAddress	Array of String	Forwarding address IP address. Example: ["1.1.1.2:80", "1.2.3.4:80"]
Remark	String	Remarks Example: This is a remark.
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:0
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.	Parameter value error.
InternalError.UndefiendError	Undefined error.
InvalidParameter.	Parameter error.
InvalidParameterValue.RecordNotExist	This record does not exist.

ModifyVpcDnsDomain

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for modifying VPC DNS domain names

Default interface request rate limit: 20 times per second.

Interface update time: 2022-01-11 05:02:35.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyVpcDnsDomain
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainIds	Yes	No	String	The domain name IDs, which are separated with commas. Example: 1,3,4
DnsForwardStatus	No	No	String	Enable subdomain recursion, DISABLED: off, ENABLED: on Example: ENABLED

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The return time of the creation request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.	Parameter error.
FailedOperation.ModifyDomainFailed	Failed to modify the domain name.

ModifyVpcDnsRecord

1. Interface Description

Interface request domain name: vpcdns.api3.convergecloud.com.

API for modifying VPC DNS records

Default interface request rate limit: 20 times per second.

Interface update time: 2025-06-24 04:47:03.

The interface only verifies the signature but does not authenticate.

2. Input Parameters

The following list of request parameters only lists interface request parameters and some public parameters, For a complete list of public parameters see public request parameters.

Parameter name	Required	Allow NULL	Type	Description
Action	Yes	No	String	Public parameter, The value of this interface: ModifyVpcDnsRecord
Version	Yes	No	String	Public parameter, The value of this interface: 2019-10-25
Region	No	No	String	Public parameter,Regional information This interface does not need to pass this parameter.
DomainId	Yes	No	Uint64	The domain name ID. Example: 3
RecordId	Yes	No	Uint64	The record ID. Example: 4
SubDomain	Yes	No	String	The subdomain name. Example: www
RecordType	Yes	No	String	Record types: A, AAAA, MX, CNAME, TXT, PTR, SRV, SPF Example: A
Value	Yes	No	String	The record value. Example: 1.1.1.1

Parameter name	Required	Allow NULL	Type	Description
Mx	No	No	Uint64	The MX priority. Example: 45
Weight	Yes	No	String	Weight. Example: 100

3. Output Parameters

Parameter name	Type	Description
CreatedAt	Datetime	The response time of the binding request. Example: 2014-08-03 12:00:00
RequestId	String	Unique request id, Each request will return. The RequestId of the request needs to be provided when locating the problem.

4. Error Code

The following lists only the error codes related to the interface business logic, For other error codes, see public ErrorCode.

Error Code	Description
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.IllegalPTRRecord	Invalid PTR record.
InvalidParameter.IllegalRecordValue	Invalid record value.
InvalidParameterValue.RecordUnsupportWeight	The current record type does not support the weight.
InvalidParameterValue.RecordConflict	Record conflict.
InvalidParameterValue.CnameNotPrivateZone	You can only use CNAMEs to point to the private domains under the current account.
InvalidParameterValue.	Parameter value error.
InvalidParameterValue.RecordAAAACountExceed	The number of AAAA records for CLB exceeds 50.

Error Code	Description
InvalidParameterValue.RecordACountExceed	The number of A records for CLB exceeds 50.
InvalidParameter.IllegalDomain	Incorrect domain name.
InvalidParameter.IllegalRecord	Invalid records.
InvalidParameterValue.RecordNotExist	This record does not exist.
InvalidParameterValue.RecordTXTCountExceed	The number of TXT records for CLB exceeds 10.
InternalError.UndefiendError	Undefined error.
InvalidParameterValue.RecordMXCountExceed	The number of MX records for CLB exceeds 50.
InvalidParameter.IllegalCidr	Invalid CIDR block.
InvalidParameter.RecordLevelExceed	The level recorded exceeds the limit.
FailedOperation.UpdateRecordFailed	Failed to update the record.
InvalidParameter.	Parameter error.
InvalidParameterValue.RecordExist	Record already exists.

Data structure

Tag

A tag key-value pair.

Referenced by the following interfaces: CreateVpcDnsDomain

name	required	NULL allowed	type	description
Key	yes	no	String	Tag key Example: a
Value	yes	no	String	Tag value Example: tagValue1

VpcInfos

VpcInfos includes the VPC ID and region ID.

Referenced by the following interfaces: BindVpcDnsDomain, DescribeForwardRuleList, DescribeVpcDnsDomainList

name	required	NULL allowed	type	description
VpcId	yes	no	Int64	VpcId Example: 4
RegionId	yes	no	Int64	RegionId Example: 4
UnVpcId	yes	no	String	UnVpcId Example: vpc-9a5ycnih

BackUpRecordInfo

Backup record.

Referenced by the following interfaces: DescribeVpcDnsRecordList

name	required	NULL allowed	type	description
RecordType	yes	no	String	Record type. It can only be A or MX. Example: A
SubDomain	yes	no	String	The subdomain name. Example: www
RecordValue	yes	no	String	The record value. Example: 1.2.3.1
AliveList	yes	no	Array of AliveInfo	Liveness status Example:View

DomainDetail

The domain name details.

Referenced by the following interfaces: DescribeVpcDnsDomainList

name	required	NULL allowed	type	description
DomainId	no	no	Int64	Domain ID. Example: 4
OwnerUin	no	no	Int64	The UIN of the domain name owner. Example: 909619400
Domain	no	no	String	Domain name. Example: test.com
CreatedOn	no	no	Datetime	Creation time. Example: 2014-08-03 12:00:00
UpdatedOn	no	no	Datetime	Last modification time. Example: 2014-08-03 12:00:00
RecordCount	no	no	Int64	The number of domain name records. Example: 3
Remark	no	yes	String	The domian name remarks. Example: Remarks
DnsForwardStatus	no	no	String	Whether to enable the sub-domain name recursion. Valid values: "ENABLED" and "DISABLED". Example: ENABLED
ForwardRuleStatus	no	no	String	Whether domain name forwarding is enabled (1: disabled, 0: enabled). Example: 1
VpcInfos	no	yes	Array of VpcInfos	VPC information. Example:View

RuleListFilters

Query the forwarding rule list.

Referenced by the following interfaces: DescribeForwardRuleList

name	required	NULL allowed	type	description
Name	yes	no	String	Filter type. Example: DomainName
Values	yes	no	Array of String	Filter value. Example: test.com

RecordDetail

Record details.

Referenced by the following interfaces: DescribeVpcDnsRecordList

name	required	NULL allowed	type	description

name	required	NULL allowed	type	description
RecordId	yes	no	Int64	The record ID. Example: 4
DomainId	yes	no	Int64	The ID of the domain name corresponding to the record. Example: 3
SubDomain	yes	no	String	The host record. Example: www
RecordType	yes	no	String	Record types: A, AAAA, MX, CNAME, TXT, PTR, SRV, SPF Example: A
Value	yes	no	String	The record value. Example: 1.1.1.1
Ttl	yes	no	Int64	TTL Example: 600
Mx	yes	yes	Int64	The MX priority. Example: 45
Enabled	yes	no	Int64	Whether the record is available. 0: unavailable; 1: available. Example: 0
Status	yes	no	String	Record status, enabled, suspended or other. Valid values: new, pending, enabled, disabled and lock. Example: enabled
Extra	yes	yes	String	Other information. Example: weight:100
CreatedOn	yes	no	Datetime	Creation time of the record. Example: 2014-08-03 12:00:00
UpdatedOn	yes	no	Datetime	Last modification time of the record. Example: 2014-08-03 12:00:00
Weight	yes	yes	Int64	The weight of the record. Example: 100
AliveList	no	yes	Array of AliveInfo	Liveness status Example:View
BackUpRecordList	no	yes	Array of BackUpRecordInfo	Backup record. Example:View
EnableValue	no	no	String	Record value that is being used currently. Example: 1.2.3.1
HealthStatus	no	no	Int64	Response parameter of the health status. 0: abnormal; 1: healthy; 2: not enabled; 3: unknown; 4: partially abnormal. Example: 2
BackUpCam	no	no	Bool	Backup record CAM status. Example: true

RecordListCountInfo

The number of resolution records.

Referenced by the following interfaces: DescribeVpcDnsRecordList

name	required	NULL allowed	type	description
AllTotal	no	no	Int64	The total number of domain name records. Example: 3
RecordTotal	no	no	Int64	The number of returned records. Example: 3

RecordListFilters

The fuzzy query of the VPC DNS record list.

Referenced by the following interfaces: DescribeVpcDnsRecordList

name	required	NULL allowed	type	description
Name	yes	no	String	Filter types: SubDomain, Value, RecordId, RecordType Example: SubDomain
Values	yes	no	Array of String	Filter value. Example: ["www"]

ForwardRuleList

Forwarding rule list.

Referenced by the following interfaces: DescribeForwardRuleList

name	required	NULL allowed	type	description
DomainId	no	no	String	Domain ID. Example: 4
DomainName	no	no	String	Domain name. Example: test.com
RuleId	no	no	String	Forwarding rule ID Example: 12
ForwardAddress	no	no	Array of String	Forwarding IP address. Example: ["1.1.1.2:80", "1.2.3.4:80"]
Remark	no	yes	String	Description. Example: This is a remark.
CreatedOn	no	no	Datetime_iso	Creation time. Example: 2014-08-03 12:00:00
UpdatedOn	no	no	Datetime_iso	Last modification time. Example: 2014-08-03 12:00:00
VpcInfos	no	yes	Array of VpcInfos	VPC information. Example:View

BackupRecord

Backup record.

Referenced by the following interfaces: DescribeBackupRecordList

name	required	NULL allowed	type	description
RecordId	yes	no	Uint64	Backup record ID. Example: 421
DomainId	yes	no	Uint64	Domain ID. Example: 39
SubDomain	yes	no	String	The subdomain name. Example: www
RecordType	yes	no	String	The record type. Example: A
Value	yes	no	String	The record value. Example: 1.2.4.12

DomainCountInfo

Number of domain names.

Referenced by the following interfaces: DescribeVpcDnsDomainList

name	required	NULL allowed	type	description
AllTotal	no	no	Int64	Total number of domain names. Example: 4
DomainTotal	no	no	Int64	The number of returned domain names. Example: 4

AliveInfo

Liveness information.

Referenced by the following interfaces: DescribeVpcDnsRecordList

name	required	NULL allowed	type	description
VpcId	yes	no	String	VpcId Example: 6981
AliveStatus	yes	no	Uint64	Liveness status. 0: unknown; 1: dead; 2: live. Example: 0

RecordId

Record ID.

Referenced by the following interfaces: CreateVpcDnsRecord

name	required	NULL allowed	type	description
RecordId	yes	no	Int64	Record ID. Example: 3

DomainListFilters

The fuzzy query of VPC DNS domain names.

Referenced by the following interfaces: DescribeExportFileUrl, DescribeVpcDnsDomainList

name	required	NULL allowed	type	description
Name	yes	no	String	Filter criteria: Domain query Domain parsing domain; VpcInfo VPC ID or instance ID; Account user OwnerUin. Record query: SubDomain subdomain, Value record value, RecordId record ID, RecordType record type Example: Domain
Values	yes	no	Array of String	Filter value. Example: ["test"]

FailedRecordDetail

Details of the failed file import parse record

Referenced by the following interfaces: ImportRecords

name	required	NULL allowed	type	description
Type	no	yes	String	The record type. Example: A
Subdomain	no	yes	String	The host record. Example: www
Weight	no	yes	Uint64	Weight Example: 100
MX	no	yes	Uint64	MX value Example: 45
Value	no	yes	String	Parse record value Example: 1.1.1.1
Reason	no	yes	String	Reason for addition failure Example: Record not valid

ErrorCode

Feature Description

If there is an Error field in the response, it means that the API call failed. For example:

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

Code in Error indicates the error code, and Message indicates the specific information of the error.

Error Code List

Common Error Codes

Error codes	Description
AuthFailure.InvalidSecretId	Invalid key (not a Cloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	Key does not exist. Check if the key has been deleted or disabled in the console, and if not, check if the key is correctly entered. Note that whitespaces should not exist before or after the key.
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes. Please ensure your current local time matches the standard time.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you've followed the signature calculation process described in the Signature API documentation.
AuthFailure.TokenFailure	Token error.

Error codes	Description
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Service Error Codes

Error codes	Description
InvalidParameterValue.IllegalWeightValue	The weight value must be a number from 1 to 100.

Error codes	Description
InvalidParameterValue.RecordConflict	Record conflict.
InvalidParameter.	Parameter error.
InvalidParameterValue.RecordExist	Record already exists.
InvalidParameterValue.DomainNotExist	The domain name does not exist.
InvalidParameter.IllegalDomain	Incorrect domain name.
InvalidParameterValue.	Parameter value error.
InvalidParameter.IllegalRecord	Invalid records.
InvalidParameter.IllegalCidr	Invalid CIDR block.
InvalidParameter.IllegalDomainTld	Top-level domain name.
InvalidParameterValue.RecordNotExist	This record does not exist.
InvalidParameter.InvalidZoneFileFormat	The file type is not supported.
InvalidParameterValue.IllegalTXTValue	After Punycode encoding, the length of the txt format record value must be between 1 and 255.
InvalidParameter.IllegalPTRRecord	Invalid PTR record.
InvalidParameterValue.RecordTXTCountExceed	The number of TXT records for CLB exceeds 10.
InvalidParameterValue.InvalidZoneFileSize	The file size exceeds the limit.
InvalidParameter.ImportedFileExpired	The file is expired
InvalidParameterValue.BackupRecordExceed	Setting more than three alternative DNS IP addresses is not supported.
UnknownParameter.	Unknown parameter.
InternalError.	Internal Error
InvalidParameterValue.BackupRecordNotSupport	The current data does not support backup records.
ResourceInUse.	The resource is in use.
FailedOperation.UpdateRecordFailed	Failed to update the record.
InvalidParameter.IllegalVpcInfo	Invalid VPC.

Error codes	Description
InvalidParameterValue.RecordUnsupportWeight	The current record type does not support the weight.
InvalidParameterValue.RecordAAAACountExceed	The number of AAAA records for CLB exceeds 50.
InternalError.UndefiendError	Undefined error.
FailedOperation.ModifyDomainFailed	Failed to modify the domain name.
InternalError.FetchCOSFileError	Pulling the COS file failed.
InvalidParameter.IllegalSubnet	Invalid subnet.
InvalidParameterValue.RecordACountExceed	The number of A records for CLB exceeds 50.
FailedOperation.CreateRecordFailed	Failed to create the record.
InvalidParameterValue.ForwardRuleOverLimit	Number of forwarding rules exceeds the limit of 200.
FailedOperation.DeleteRecordFailed	Failed to delete the record.
InvalidParameterValue.RecordMXCountExceed	The number of MX records for CLB exceeds 50.
InvalidParameterValue.CnameNotPrivateZone	You can only use CNAMEs to point to the private domains under the current account.
InvalidParameter.RecordLevelExceed	The level recorded exceeds the limit.
InvalidParameter.IllegalRecordValue	Invalid record value.
InvalidParameterValue.VpcBinded	The VPC has bound to other resolution domains.
FailedOperation.DeleteDomainFailed	Failed to delete the domain name.
InvalidParameter.WeightNotSupported	The current record does not support weight settings.
InvalidParameterValue.BackupRecordOverLimit	The number of backup records of this tenant exceeds the limit.