ALIBABA CLOUD

Alibaba Cloud

阿里云公共DNS

常见问题

文档版本：20210305

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.iOS14原生加密DNS方案

本文档介绍了阿里云公共DNS 在 iOS 14原生加密DNS方案中的接入和开发方式。

概述

DNS解析是网络资源访问的第一跳，iOS 14 开始系统原生支持两种标准规范的 Encrypted DNS, 分别是 DNS over TLS 与 DNS over HTTPS，可以解决以下两个问题：

一、传统Local DNS的查询与回复均基于非加密UDP，发生我们常见的DNS劫持问题。

二、Local DNS Server本身不可信，或者本地Local DNS 服务不可用问题。

针对DNS解析过程中以上两个问题，阿里云公共DNS已经有了解决方案，即使用阿里云公共DNS SDK，但使用SDK会面临一些技术坑，比如302场景的IP直连处理、WebView下IP直连如何处理、以及iOS上的SNI问题等，而iOS 14 上的 Encrypted DNS 功能很好的解决了集成SDK的方案存在的问题，您可以参考Demo示例工程源码了解如何设置阿里云公共DNS为加密DNS默认解析器。

iOS 14原生加密DNS方案如何接入阿里云公共DNS

iOS 14 提供了两种设置加密DNS的方法。

第一种方式是选择一个DNS服务器作为系统全局所有App默认的DNS解析器，如果你提供的是一个公共DNS服务器，你可以使用NEDNSSettingsManager API编写一个NetworkExtension App完成系统全局加密DNS设置。

使用NetworkExtension设置系统域全局DNS服务器，使用DoH协议示例代码：

```
import NetworkExtension

NEDNSSettingsManager.shared().loadFromPreferences { loadError in
        if let loadError = loadError {
          // ...handle error...
          return
        }
        let dohSettings = NEDNSOverHTTPSSettings(servers: ["223.5.5.5","223.6.6.6","2400:3200:baba::1","240
0:3200::1"])
        dohSettings.serverURL = URL(string: "https://您在控制台注册应用时分配的AccountID.alidns.com/dns-qu
ery")
        NEDNSSettingsManager.shared().dnsSettings = dohSettings
        NEDNSSettingsManager.shared().saveToPreferences { saveError in
          if let saveError = saveError {
           // ...handle error...
           return
          }
        }
    }
```

使用NetworkExtension设置系统域全局DNS服务器，使用DoT协议示例代码：

```
import NetworkExtension

NEDNSSettingsManager.shared().loadFromPreferences { loadError in
        if let loadError = loadError {
          // ...handle error...
          return
        }
        let dotSettings = NEDNSOverTLSSettings(servers: ["223.5.5.5","223.6.6.6","2400:3200:baba::1","2400:3
200::1"])
        dotSettings.serverName = "您在控制台注册应用时分配的AccountID.alidns.com"
        NEDNSSettingsManager.shared().dnsSettings = dotSettings
        NEDNSSettingsManager.shared().saveToPreferences { saveError in
          if let saveError = saveError {
           // ...handle error...
           return
          }
        }
    }
```

一条DNS配置包括阿里公共DNS服务器地址、DoT/DoH协议、一组网络规则。网络规则确保DNS设置兼容不同的网络。

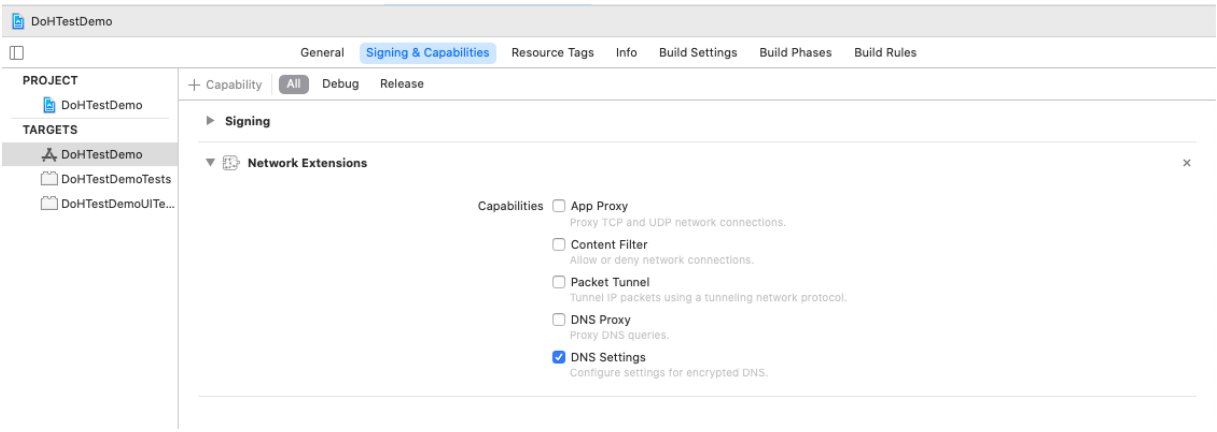
网络规则设置示例代码：

```
let workWiFi = NEOnDemandRuleEvaluateConnection()
   workWiFi.interfaceTypeMatch = .wiFi
   workWiFi.ssidMatch = ["MyWorkWiFi"]
   workWiFi.connectionRules = [NEEvaluateConnectionRule(matchDomains: ["enterprise.example"], andActi
on: .neverConnect)]

   let disableOnCell = NEOnDemandRuleDisconnect()
   disableOnCell.interfaceTypeMatch = .cellular

    let enableByDefault = NEOnDemandRuleConnect()
    NEDNSSettingsManager.shared().onDemandRules = [
          workWiFi,
          disableOnCell,
          enableByDefault
    ]
```

上述代码设置了三个网络规则，第一个规则表示DNS配置应该在SSID=“MyWorkWiFi”的WiFi网络生效，但对私有企业域名enterprise.example.net不开启；第二个规则表示规则在蜂窝网下应该被禁止使用；第三个NEOnDemandRuleConnect表示DNS配置应该默认开启；因为配置DNS是系统支持的，所以在编写NetworkExtension App时不需要实现Extension程序，只需要在Network Extensions中勾选DNS Settings选项。



运行NetworkExtension App，DNS配置将会被安装到系统，为了让DNS配置生效，需要前往设置->通用->VPN & Network->DNS手动启用。

第二种方式是针对单个App的所有连接或部分连接启用加密DNS。

如果你只想为你的App使用加密DNS，而非涉及整个系统域。你可以适配Network.framework的PrivacyContext，对你的整个App开启加密DNS，或者仅对某一连接开启。

对单个连接使用加密DNS，使用DoH协议示例代码：

```
import Network

let privacyContext = NWParameters.PrivacyContext(description: "EncryptedDNS")
    if let url = URL(string: "https://您在控制台注册应用时分配的AccountID.alidns.com/dns-query") {
        let address1 = NWEndpoint.hostPort(host: "223.5.5.5", port: 443)
        let address2 = NWEndpoint.hostPort(host: "223.6.6.6", port: 443)
        let address3 = NWEndpoint.hostPort(host: "2400:3200::1", port: 443)
        let address4 = NWEndpoint.hostPort(host: "2400:3200:baba::1", port: 443)
        privacyContext.requireEncryptedNameResolution(true, fallbackResolver: .https(url, serverAddresses:
[address1,address2,address3,address4]))
    }
```

对单个连接使用加密DNS，使用DoT协议示例代码：

```
import Network

 let privacyContext = NWParameters.PrivacyContext(description: "EncryptedDNS")
    let alidnsHost = NWEndpoint.hostPort(host: "您在控制台注册应用时分配的AccountID.alidns.com", port: 85
3)
    let address1 = NWEndpoint.hostPort(host: "223.5.5.5", port: 853)
    let address2 = NWEndpoint.hostPort(host: "223.6.6.6", port: 853)
    let address3 = NWEndpoint.hostPort(host: "2400:3200::1", port: 853)
    let address4 = NWEndpoint.hostPort(host: "2400:3200:baba::1", port: 853)
    privacyContext.requireEncryptedNameResolution(true, fallbackResolver: .tls(alidnsHost, serverAddress
es: [address1,address2,address3,address4]))
```

如果你想在App范围内使用加密DNS，你可以配置默认的PrivacyContext；App内发起的每个DNS解析都会使用这个配置。

在App范围内使用加密DNS，使用DoH协议示例代码：

```
import Network

if let aliUrl = URL(string: "https://您在控制台注册应用时分配的AccountID.alidns.com/dns-query"){
        let address1 = NWEndpoint.hostPort(host: "223.5.5.5", port: 443)
        let address2 = NWEndpoint.hostPort(host: "223.6.6.6", port: 443)
        let address3 = NWEndpoint.hostPort(host: "2400:3200::1", port: 443)
        let address4 = NWEndpoint.hostPort(host: "2400:3200:baba::1", port: 443)
        NWParameters.PrivacyContext.default.requireEncryptedNameResolution(true, fallbackResolver: .http
s(aliUrl, serverAddresses: [address1,address2,address3,address4]))
        }
```

在App范围内使用加密DNS，使用DoT协议示例代码：

```
import Network

let alidnsHost = NWEndpoint.hostPort(host: "您在控制台注册应用时分配的AccountID.alidns.com", port: 853)
    let address1 = NWEndpoint.hostPort(host: "223.5.5.5", port: 853)
    let address2 = NWEndpoint.hostPort(host: "223.6.6.6", port: 853)
    let address3 = NWEndpoint.hostPort(host: "2400:3200::1", port: 853)
    let address4 = NWEndpoint.hostPort(host: "2400:3200:baba::1", port: 853)
    NWParameters.PrivacyContext.default.requireEncryptedNameResolution(true, fallbackResolver: .tls(ali
dnsHost, serverAddresses: [address1,address2,address3,address4]))
```