

# 阿里云 Cloud Toolkit

开发应用

文档版本：20190911

# 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
	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。	 <b>禁止：</b> 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 <b>警告：</b> 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 <b>说明：</b> 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	单击 <b>确定</b> 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[ ]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand   slave}</code>

# 目录

---

法律声明.....	I
通用约定.....	I
1 Arthas 诊断.....	1
2 快速创建 Dubbo 工程.....	4

# 1 Arthas 诊断

---

通过 Cloud Toolkit 插件，您可以在本地 IDE 中使用 Arthas 来实现本地诊断或远程诊断。本文将介绍使用 Arthas 进行本地诊断和远程诊断的操作过程。

## 背景信息

[Arthas](#) 是 Alibaba 开源的 Java 诊断工具，可以帮助开发者进行在线排查问题、动态跟踪 Java 代码、实时监控 JVM 状态等。Arthas 支持 JDK 6+，支持 Linux、Mac 和 Windows 系统，采用命令行交互模式，同时提供丰富的 Tab 自动补全功能，方便进行问题的定位和诊断。

您可以通过 [Arthas 快速入门](#) 来学习 Arthas，更多高级功能请参见[进阶使用](#)。

## 前提条件

您已安装和配置 Cloud Toolkit，参见在 [IntelliJ IDEA 中安装和配置 Cloud Toolkit](#)

## 操作步骤

1. 打开 IntelliJ IDEA。

2. 根据您的需求选择一种方式进入诊断界面，开始诊断。

- 诊断本地的 Java 应用。

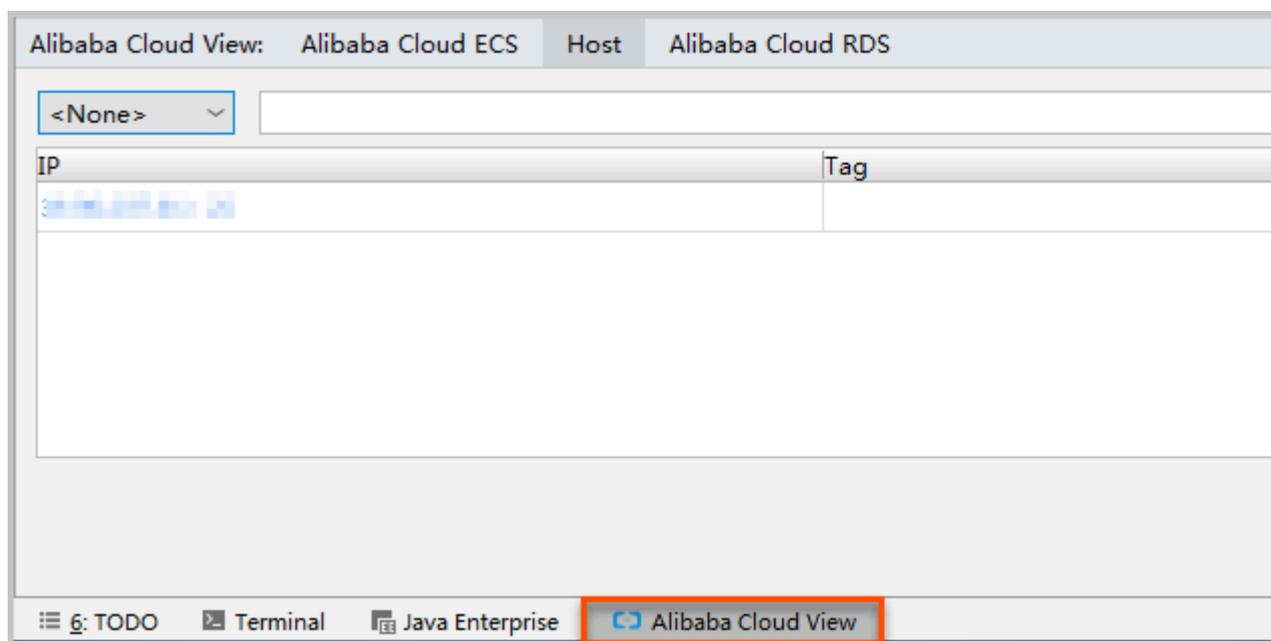
在 IntelliJ IDEA 编辑器顶部导航栏选择 Tools > Alibaba Cloud > Diagnostic Tools，进入本地诊断界面。

- 诊断远程服务器上的应用。

a. 在 IntelliJ IDEA 中添加服务器。请参见[添加服务器](#)。

b. 在 IntelliJ IDEA 编辑器底部单击 Alibaba Cloud View 页签，并选择 Host 页签，然后在目标机器右 Actions 列单击 Diagnostic，进入远程诊断界面。

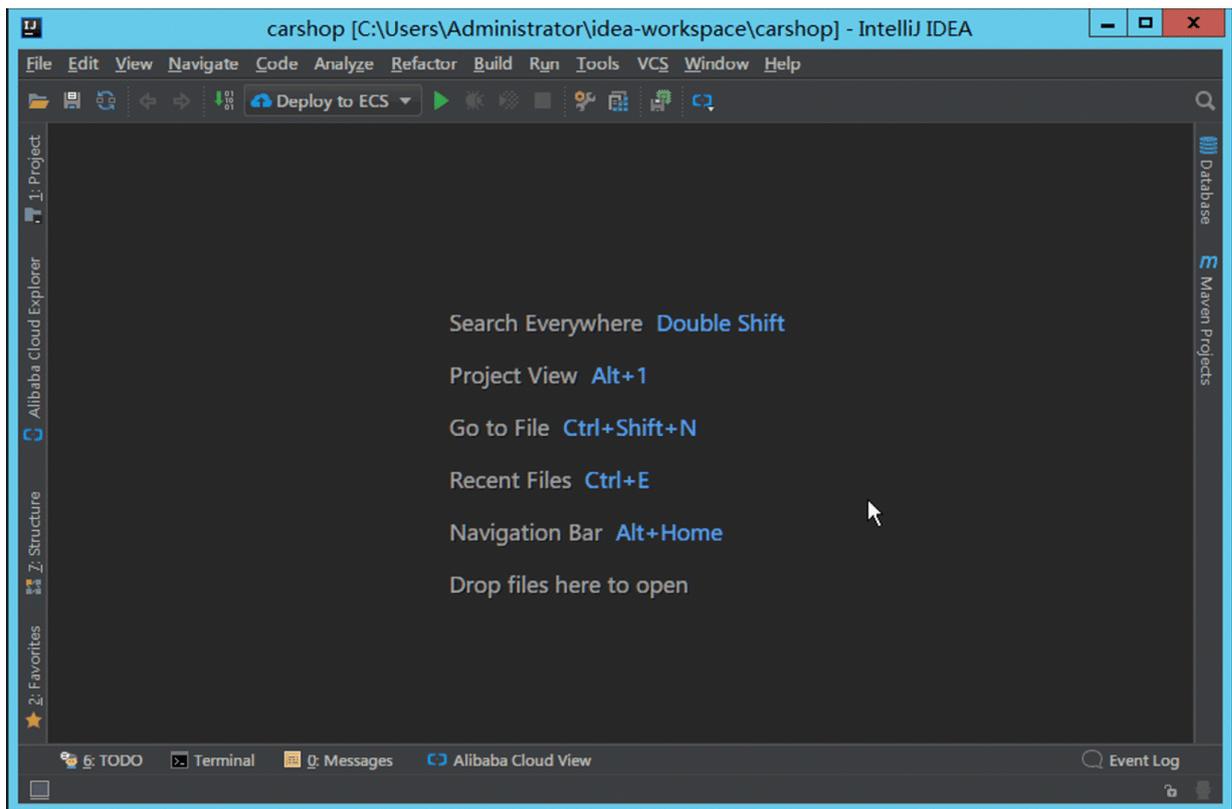
若您购买阿里云 ECS 来添加服务器，则需在 Alibaba Cloud View 页签顶部选择 Alibaba Cloud ECS 页签，然后在目标机器右册 Actions 列单击 Diagnostic。



说明:

第一次使用 Diagnostic Tools 诊断时，会自动安装并初始化 Arthas。您需要等待约一分钟后即可开始使用。

### 操作演示



## 2 快速创建 Dubbo 工程

---

使用 Cloud Toolkit 可以快速创建 Dubbo 工程，然后您可以将 Dubbo 工程部署到 EDAS 上。本文将介绍如何在 IntelliJ IDEA 使用 Cloud Toolkit 创建 Apache Dubbo 应用样例工程（包含一个 Provider 和 Consumer），并完成本地调用验证。

### 前提条件

确保您在 IntelliJ IDEA 中安装和配置 Cloud Toolkit 版本在 2019.6.2 及以上，安装步骤请参见在 [IntelliJ IDEA 中安装和配置 Cloud Toolkit](#)。

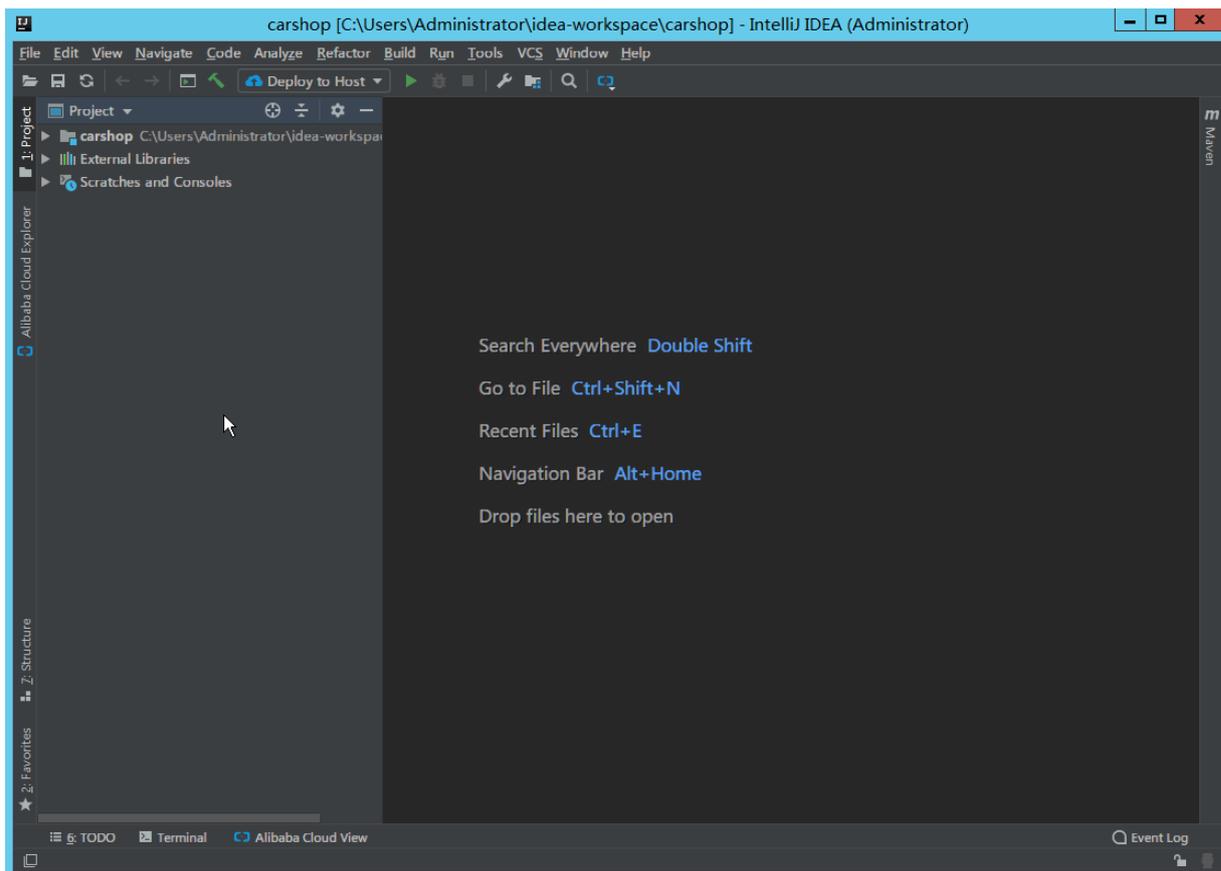
### 操作步骤

1. 打开 IntelliJ IDEA，在菜单栏选择 File > New > Project。
2. 在 New Project 对话框左侧的导航栏中单击 Apache Dubbo，在右侧界面中选择 JDK 版本，然后单击 Next。
3. 设置工程基本配置，然后单击 Next。

基本参数设置说明及示例：

- Project：选择 Maven Project。
  - groupId：输入相应的 Group ID，如 com.dubbo.example。
  - ArtifactId：输入相应的 Artifact ID，如 dubbo-demo。
  - Version：应用工程的版本，如 1.0-SNAPSHOT。
  - Dubbo：选择 Dubbo 的版本，如 2.7.2。
  - Type：工程的类型，勾选 Consumer 和 Provider，则会创建服务提供者和服务消费者的工程 Demo。
  - Spring Boot：Spring Boot 的版本，如 2.1.5.RELEASE。
4. 设置 Project name 和 Project location，然后单击 Finish。

具体操作过程如下：



## 验证 Apache Dubbo 应用样例工程

通过 Cloud Toolkit 创建的 Apache Dubbo 应用样例工程是一个 Spring boot + Dubbo 的工程。服务端 (Provider) 提供了一个服务 `com.dubbo.example.DemoService`，通过 Dubbo 协议暴露在 12345 端口。



### 说明:

由于默认配置为 `dubbo.registry.address=N/A`，说明服务没有注册到任何注册中心，需要客户端 (Consumer) 通过直连的方式来发现服务。

### 步骤一：启动 Provider

1. 在 IntelliJ IDEA 中运行 (run) `com.dubbo.example.provider.DubboProviderBootstrap` 的 main 函数。
2. 观察标准输出。

出现以下字段，说明服务端 (Provider) 成功启动。

```
2019-07-03 16:05:50.585 INFO 19246 --- [           main] c.d.e.
provider.DubboProviderBootstrap : Started DubboProviderBootstrap
in 36.512 seconds (JVM running for 42.004)2019-07-03 16:05:50.587
INFO 19246
```

```
--- [pool-1-thread-1] .b.c.e.AwaitingNonWebApplicationListener : [Dubbo] Current Spring Boot Application is await...
```

## 步骤二：启动 Consumer 并验证调用

在客户端（Consumer）的 `com.dubbo.example.consumer.DubboConsumerBootstrap` 中有如下代码：

```
@Reference(version = "1.0.0", url = "dubbo://127.0.0.1:12345")
private DemoService demoService;
```

这表明客户端（Consumer）通过指定服务端（Provider）地址 `dubbo://127.0.0.1:12345` 的直连方式调用服务。

1. 在 IntelliJ IDEA 中运行（Run） `com.dubbo.example.consumer.DubboConsumerBootstrap` 的 `main` 函数。
2. 观察服务端（Provider）的打印日志，出现以下字段：

```
Hello mercyblitz, request from consumer: /30.5.124.39:59553
```

3. 观察客户端（Consumer）打印日志，出现以下字段，则说明调用成功。

```
Hello mercyblitz, response from provider: 30.5.124.39:12345
```



### 说明：

- 注意：若要使用注册中心进行服务注册发现，请修改 `dubbo-demo/dubbo-demo-provider/src/main/resources/application.properties` 和 `dubbo-demo/dubbo-demo-consumer/src/main/resources/application.properties` 文件中的 `dubbo.registry.address` 为对应的注册中心的地址，如：`zookeeper://localhost:2181`。
- 如果使用 ZooKeeper 为注册中心，还需要在 Provider 和 Consumer 各自的样例工程的 `pom.xml` 文件中添加 ZooKeeper 的依赖。

## 后续操作

在完成 Apache Dubbo 样例工程的创建和调用验证后，可以将该样例工程打包（JAR 包）并部署到 EDAS 的不同集群（主要为 ECS 集群和容器服务 Kubernetes 集群）中，详情请参见[应用部署概述](#)。