

ALIBABA CLOUD

阿里云

云插件 Cloud Toolkit
部署应用

文档版本：20210621

 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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.部署应用概述	06
2.部署应用到ECS	08
2.1. 使用IntelliJ IDEA部署应用到ECS	08
2.2. 使用Eclipse部署应用到ECS	10
2.3. 使用Visual Studio Code部署应用到ECS	11
3.部署应用到EDAS	14
3.1. 使用IntelliJ IDEA部署应用到EDAS	14
3.2. 使用Eclipse部署应用到EDAS	19
3.3. 使用Maven部署应用到EDAS	21
3.3.1. Maven部署说明	21
3.3.2. 使用toolkit-maven-plugin部署应用到EDAS	25
3.4. 使用Cloud Toolkit实现服务的端云互联	29
3.4.1. 端云互联简介	29
3.4.2. 本地开发及联调	29
3.4.3. 使用Cloud Toolkit实现端云互联（IntelliJ IDEA）	31
3.4.4. 使用Cloud Toolkit实现端云互联（Eclipse）	38
4.部署应用到SAE	43
4.1. 使用IntelliJ IDEA部署应用到SAE	43
4.2. 使用Eclipse部署应用到SAE	44
5.部署应用到镜像仓库	47
5.1. 使用IntelliJ IDEA部署应用到镜像仓库	47
5.2. 使用Eclipse部署应用到镜像仓库	51
6.部署应用到服务器	55
6.1. 使用IntelliJ IDEA部署应用到Linux服务器	55
6.2. 使用IntelliJ IDEA部署应用到Windows服务器	58
6.3. 使用Visual Studio Code部署应用到远程服务器	62

7.部署应用到Kubernetes	65
7.1. 部署应用到容器服务Kubernetes	65
7.1.1. 使用IntelliJ IDEA部署应用到容器服务Kubernetes	65
7.1.2. 使用Eclipse部署应用到容器服务Kubernetes	69
7.2. 部署应用到自建Kubernetes	73
7.2.1. 使用IntelliJ IDEA部署应用到自建Kubernetes	74
7.2.2. 使用Eclipse部署应用到自建Kubernetes	78
8.部署多模块工程	82
8.1. 使用IntelliJ IDEA部署多模块工程中的子模块	82
8.2. 使用Eclipse部署多模块工程中的子模块	86
9.部署应用到小程序云	89
9.1. 使用IntelliJ IDEA部署应用到小程序云	89
9.2. 使用Eclipse部署应用到小程序云	90
10.部署多工程应用	93

1.部署应用概述

Cloud Toolkit与主流IDE及阿里云其他产品无缝集成，帮助您大大简化应用部署到服务器的操作。



部署应用至ECS

- 使用IntelliJ IDEA部署应用到ECS
- 使用Eclipse部署应用到ECS
- 使用Visual Studio Code部署应用到ECS

部署应用至EDAS

- 使用IntelliJ IDEA部署应用到EDAS
- 使用Eclipse部署应用到EDAS
- 使用Maven部署应用到EDAS

部署应用至SAE

- 使用IntelliJ IDEA部署应用到SAE
- 使用Eclipse部署应用到SAE

部署应用至镜像仓库

- 使用IntelliJ IDEA部署应用到镜像仓库
- 使用Eclipse部署应用到镜像仓库

部署应用至服务器

- 部署应用到Linux服务器
- 部署应用到Windows服务器
- 使用Visual Studio Code部署应用到远程服务器

部署应用至容器服务Kubernetes

- 使用IntelliJ IDEA部署应用到容器服务Kubernetes
- 使用Eclipse部署应用到容器服务Kubernetes

部署多模块工程

- 使用IntelliJ IDEA部署多模块工程中的子模块
- 使用Eclipse部署多模块工程中的子模块



部署应用至小程序云

- 使用IntelliJ IDEA部署应用到小程序云
- 使用Eclipse部署应用到小程序云

2.部署应用到ECS

2.1. 使用IntelliJ IDEA部署应用到ECS

若您已购买阿里云ECS，可以使用Cloud Toolkit实现快速部署应用。本文将介绍如何在IntelliJ IDEA中使用Cloud Toolkit将应用部署到ECS。

前提条件

- 已在部署应用的地域中存在ECS实例请参见[购买ECS](#)。

 说明 若您于2017年12月01日之前购买的ECS，则需[安装云助手客户端](#)。

- 已在IntelliJ IDEA中安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

注意事项

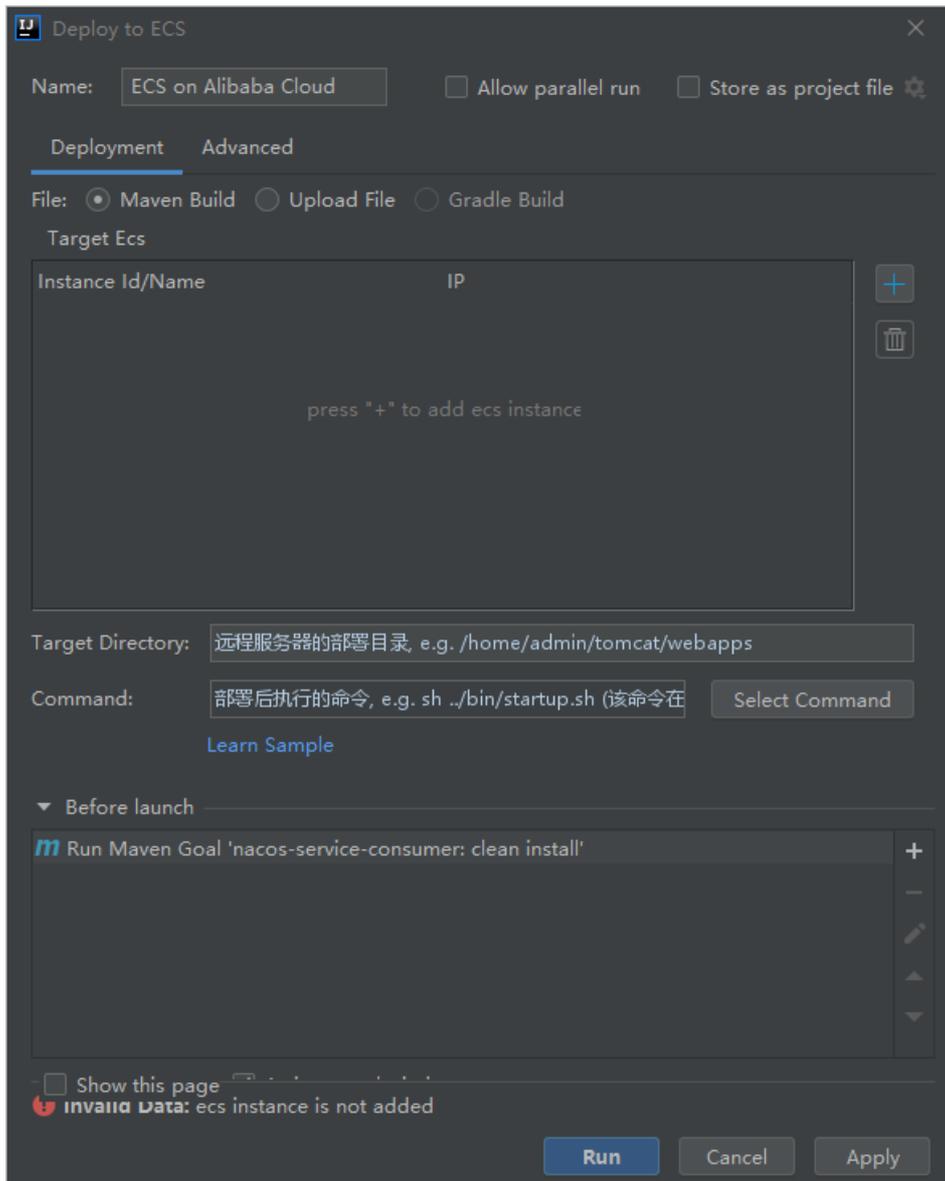
由于阿里云ECS云助手的限制，本教程中只能采用root账号来进行部署。若需切换其他账号部署，建议采用以下方式：

- 在Command脚本中切换账号。
- 使用应用服务器功能来完成部署，请参见[部署应用到Linux服务器](#)。

操作步骤

- 在IntelliJ IDEA中打开您的工程。
- 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to ECS...

3. 在Deploy to ECS对话框设置部署参数，然后单击Run。



部署参数说明：

- **Deploy File**：可选择**Maven Build**或**Upload File**。
 - **Maven Build**：若当前工程采用Maven构建，可以直接构建并部署。
 - **Upload File**：若当前工程不是采用Maven构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
 - **Gradle Build**：若当前工程采用Gradle构建，可以直接构建并部署。
- **Target Ecs**：选择目标ECS实例，操作如下：
 - a. 单击界面右侧的+。
 - b. 在Select Ecs对话框中选择目标账号、地域。
 - c. 在实例列表中勾选目标ECS实例，单击Select。
- **Target Directory**：输入ECS上的部署路径。示例如下：

Java Spring Boot Go Node.js

- **Command**：表示在完成应用包的部署后，需要执行的命令。通常是一句启动命令，示例如下：

```
sh /root/restart.sh
```

 说明 更多Command命令，请参见[Command命令编写说明](#)。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

2.2. 使用Eclipse部署应用到ECS

若您已购买阿里云ECS，可以使用Cloud Toolkit实现快速部署应用。本文将介绍如何在Eclipse中使用Cloud Toolkit将应用部署到ECS。

前提条件

- 需要部署应用的地域中存在ECS实例，请参见[购买ECS](#)。

 说明 若您是在2017年12月01日之前购买的ECS，则需[安装云助手客户端](#)。

- 已在Eclipse中安装和配置Cloud Toolkit，请参见[在Eclipse中安装和配置Cloud Toolkit](#)。

注意事项

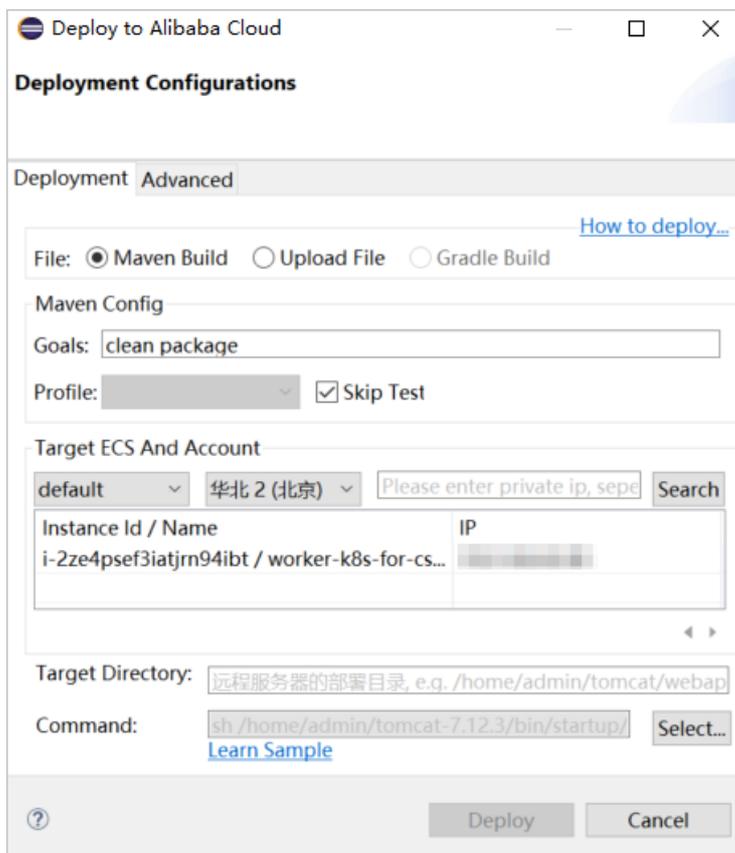
由于阿里云ECS云助手的限制，本教程中只能采用root账号来进行部署。若需切换其他账号部署，建议采用以下方式：

- 在Command脚本中切换账号。
- 使用将应用部署到服务器的功能，请参见[部署应用到Linux服务器](#)。

操作步骤

1. 在Eclipse中打开您的工程。
2. 右键单击您的应用工程名，在弹出的菜单中选择Alibaba Cloud > Deploy to ECS...

3. 在Deploy to Alibaba Cloud对话框设置部署参数，然后单击Deploy。



部署参数说明：

- **Deploy File**：可选择**Maven Build**或**Upload File**。
 - **Maven Build**：若当前工程采用Maven构建，可以直接构建并部署。
 - **Upload File**：若当前工程不是采用Maven构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
 - **Gradle Build**：若当前工程采用Gradle构建，可以直接构建并部署。
- **Target ECS And Account**：在列表中选择目标账号、地域，然后在地域中选择目标ECS实例。
- **Target Directory**：输入在ECS上的部署路径。示例如下：

Java Spring Boot Go Node.js

- **Command**：表示在完成应用包的部署后，需要执行的命令。通常是一句启动命令，示例如下：

```
sh /root/restart.sh
```

🔗 说明 更多Command命令，请参见[Command命令编写说明](#)。

结果验证

部署开始后，Eclipse的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

2.3. 使用Visual Studio Code部署应用到ECS

若您已购买阿里云ECS，可以使用Cloud Toolkit实现快速部署应用。本文将介绍如何在Visual Studio Code中使用Cloud Toolkit将应用部署到ECS。

前提条件

- 已在部署应用的地域中存在ECS实例，详情请参见[购买ECS](#)。

 **说明** 若您于2017年12月01日之前购买的ECS，则需 [安装云助手客户端](#)。

- 已在Visual Studio Code中安装和配置Cloud Toolkit，详情请参见[在Visual Studio Code中安装和配置Cloud Toolkit](#)。

注意事项

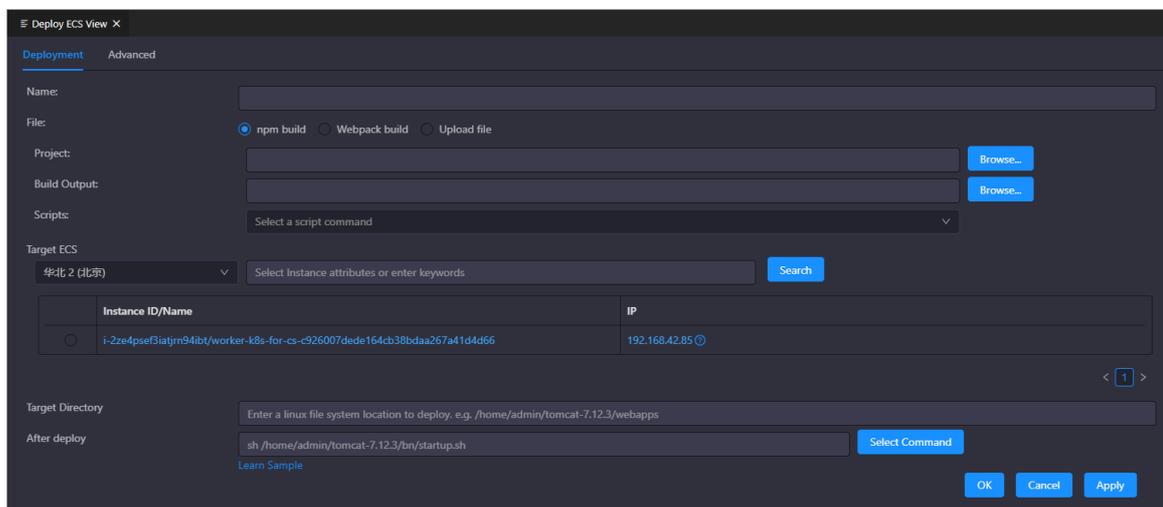
由于阿里云ECS云助手的限制，本教程中只能采用root账号来进行部署。若需切换其他账号部署，建议采用以下方式：

- 在Command脚本中切换账号。
- 使用应用服务器功能来完成部署，详情请参见[部署应用到Linux服务器](#)。

操作步骤

使用Visual Studio Code部署应用到ECS的操作步骤如下：

- 在Visual Studio Code中打开您的工程。
- 单击Cloud Toolkit的快捷图标。
- 在RUN CONFIGURATIONS区域单击，选择Deploy to ECS。
- 在Deploy to ECS页面设置部署参数，单击OK。



部署参数说明：

- **Name**：可自定义部署参数名称。
- **File**：可选择**npm build**、**Webpack build**或**Upload file**。
 - **npm build**：若当前工程采用npm构建，可以直接构建并部署。
 - **Webpack build**：若当前工程采用Webpack构建，可以直接构建并部署。
 - **Upload file**：若当前工程在本地已存在打包好的部署文件，可以选择并上传本地的部署文件。
- **Target ECS**：在下拉列表中选择**地域**，然后在地域中选择要部署的ECS实例。

- **Target Directory:** 输入ECS上的部署路径。示例如下：

Java Spring Boot Go Node.js

```
##部署目录为：  
/root/tomcat/webapps
```

- Command:** 表示在完成应用包的部署后，需要执行的命令。通常是一句启动命令，示例如下：
-

```
sh /root/restart.sh
```

 **说明** 更多Command命令，请参见[Command命令编写说明](#)。

结果验证

部署开始后，Visual Studio Code的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

3.部署应用到EDAS

3.1. 使用IntelliJ IDEA部署应用到EDAS

使用Cloud Toolkit可以将应用快速部署到EDAS，适用于快速迭代更新、分批部署应用等场景。在IntelliJ IDEA安装和配置Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

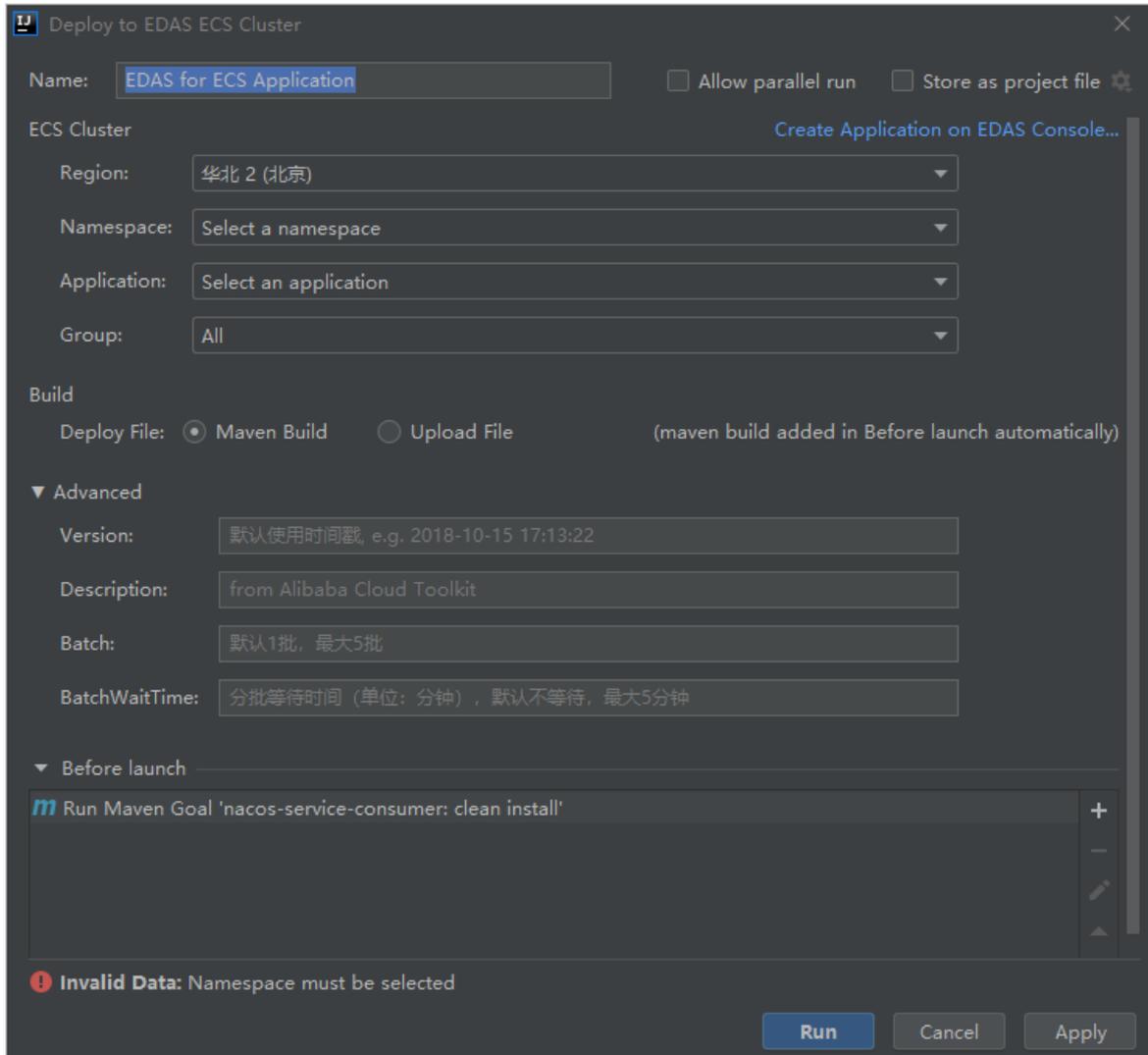
前提条件

- 已在EDAS中创建应用并完成首次部署，请参见：
 - [在ECS集群中创建Java Web应用](#)
 - [在中使用镜像方式部署Java应用](#)
- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

部署应用到ECS集群

这里以部署java_demo应用为例，具体步骤如下：

1. 在IntelliJ IDEA中打开java_demo工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to EDAS... > EDAS for ECS Application。
3. 在Deploy to EDAS ECS Cluster对话框中进行部署配置。



说明 若您尚未在EDAS上创建应用，可在对话框右上角单击Create application on EDAS Console，跳转到EDAS控制台创建应用。

部署参数说明如下。

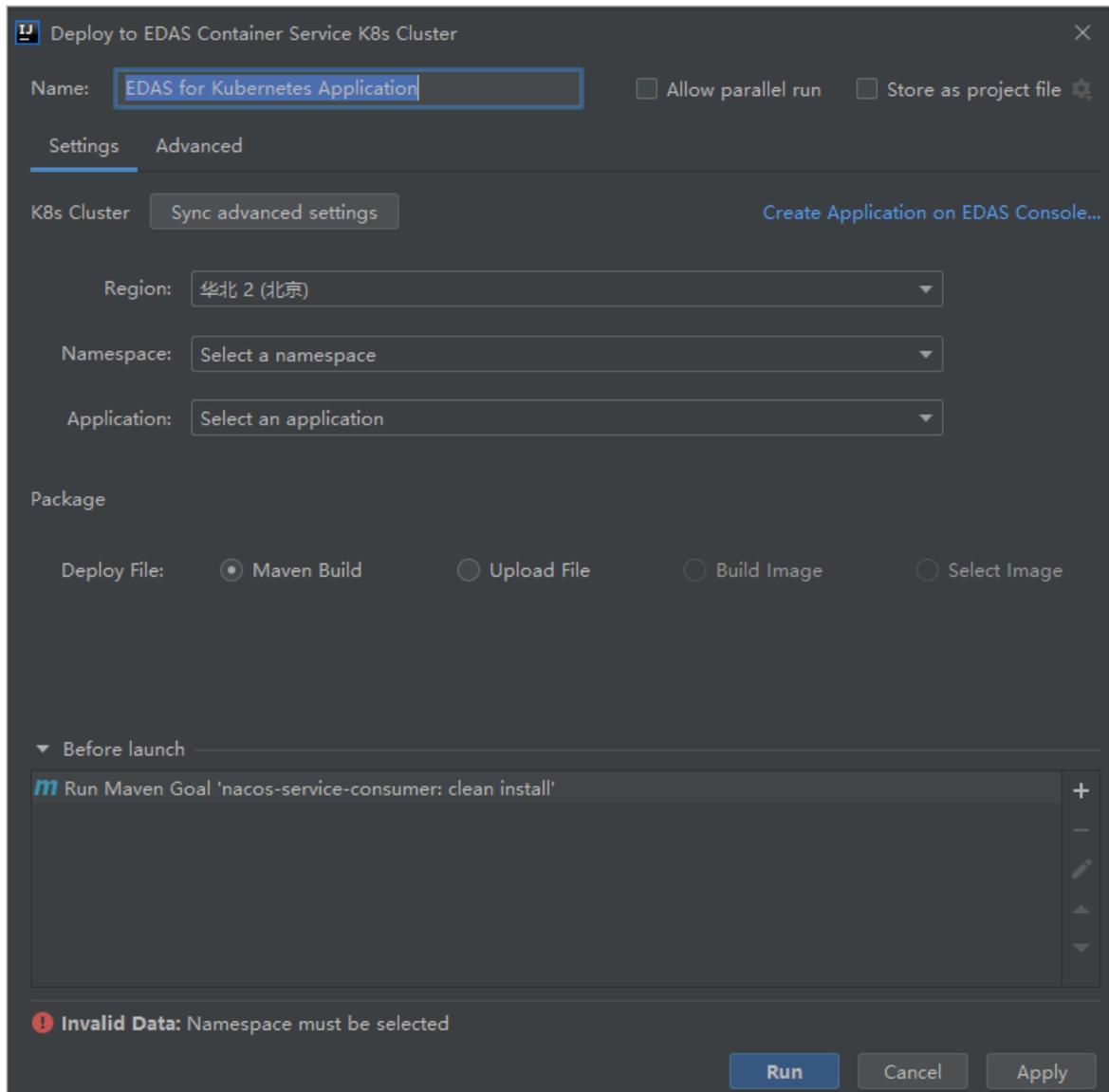
参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
	Group	应用分组，默认为All。
部署方式 (Deploy File)	Maven Build	若当前工程采用 Maven 构建，可以直接构建并部署。

参数	参数	描述
	Upload File	若当前工程不是采用 Maven 构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
分批部署（Advanced）	Version	部署版本。
	Description	部署信息描述。
	Batch	分批数。如果您的应用有多个分组，并且在部署时选择部署全部分组，那么将会自动按照分组粒度来分批，Batch值不用设置。
	BatchWaitTime	分批部署等待时间，单位为分钟。

4. 先单击Apply，然后单击Run。

部署应用到容器服务K8s集群

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to EDAS... > EDAS for Kubernetes Application。
3. 在Deploy to EDAS Container Service K8s Cluster对话框的Settings页签中进行部署参数配置。

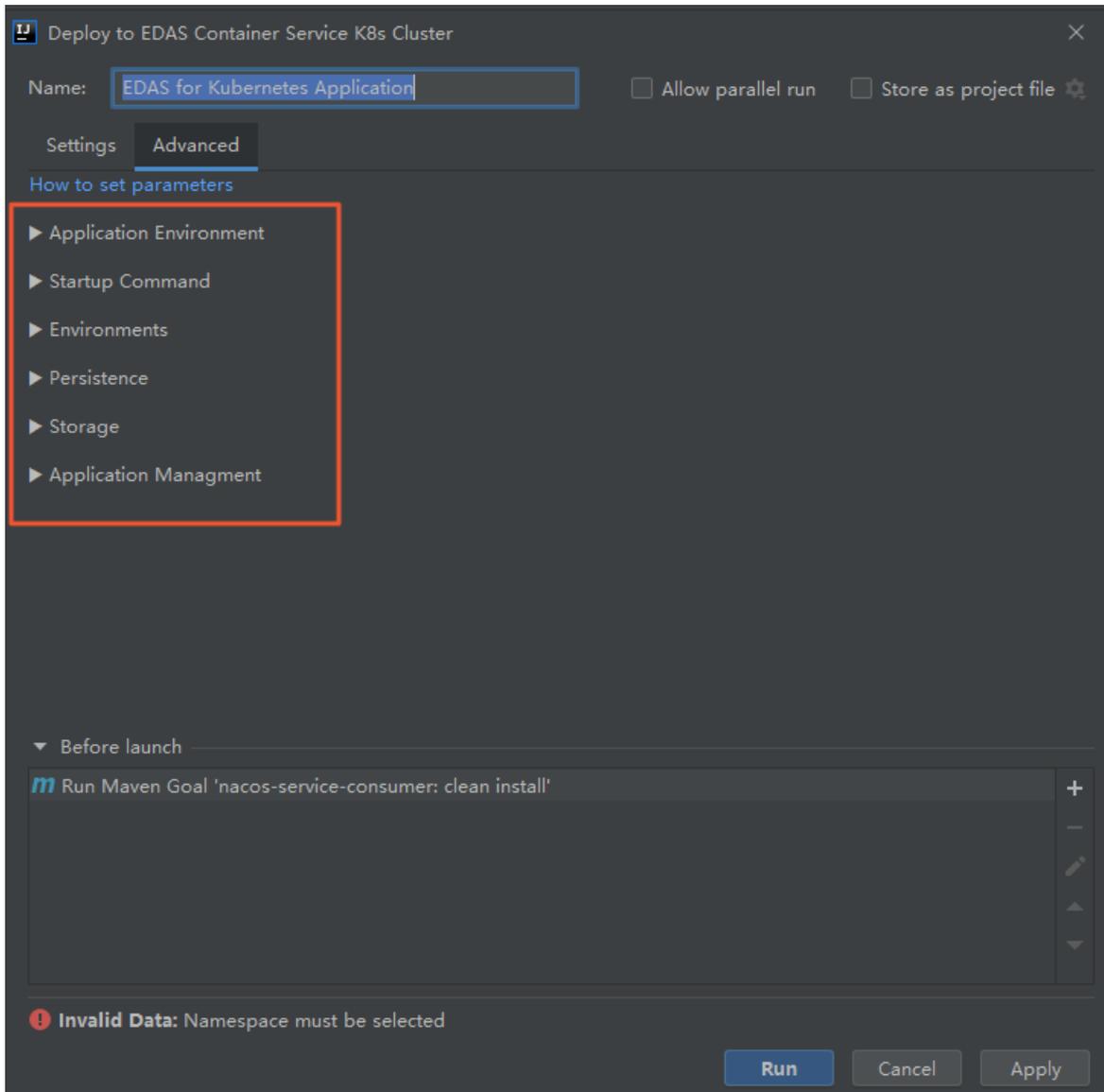


Settings页签部署参数配置说明如下。

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
	Maven Build	若当前工程采用Maven构建，可以直接构建并部署。
	Upload File	若当前工程不是采用Maven构建，或在本地已存在部署文件的压缩包，则上传本地的部署文件即可。

部署方式 (Deploy File)	参数	描述
	Build Image	选择此方式来构建应用时, 需要依次输入工作目录Content Directory、文件路径Dockerfile、镜像标签Version。
	Select Image	选择此方式来构建应用时, 可在 Select Image 的列表选择需要部署的镜像。

4. 单击Advanced页签, 配置相关参数。



Advanced页签部署参数配置说明如下。

参数	描述
Application Environment	应用环境。
Startup Command	应用启动Command命令。
Environments	环境变量。
Persistence	Nas盘挂载。
Storage	应用存储。
Appcalition Management	应用管理。

5. 先单击Apply，然后单击Run。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志，请根据日志信息检查部署结果。

您还可以登录EDAS控制台，在所部署应用的基本信息页面查看部署结果。

3.2. 使用Eclipse部署应用到EDAS

使用Cloud Toolkit可以将应用快速部署到EDAS，适用于快速迭代更新、分批部署应用等场景。在Eclipse中安装和配置Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

前提条件

- 已在EDAS中创建应用并完成首次部署，请参见：
 - 在ECS集群中创建Java Web应用
 - 在容器服务集群中创建应用
- 已安装和配置Cloud Toolkit，请参见在Eclipse中安装和配置Cloud Toolkit。

操作步骤

这里以部署Java_demo应用为例，具体步骤如下：

- 在Eclipse中导入Java_demo工程。
- 在Eclipse界面左侧的Project Explorer区域中右键单击Java_demo（工程名称），在菜单中选择Alibaba Cloud > Deploy to EDAS。
- 在Deploy to EDAS对话框中进行部署参数配置，然后单击Deploy。

说明 若您尚未在EDAS上创建应用，可在对话框右上角单击Create application on EDAS console，跳转到EDAS控制台创建应用。

部署参数说明如下。

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
	Group	应用分组，默认为All。
部署方式 (Deploy File)	Maven Build	若当前工程采用 Maven 构建，可以直接构建并部署。
	Upload File	若当前工程不是采用 Maven 构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
	Version	部署版本。
	Description	部署信息描述。

参数 分批部署 (Advanced)	参数	描述
	Batch	分批数。如果您的应用有多个分组，并且在部署时选择部署全部分组，那么将会自动按照分组粒度来分批，Batch值不用设置。
	BatchWaitTime	分批部署等待时间，单位为分钟。

结果验证

部署开始后，Eclipse的Console区域会打印部署日志，请根据日志信息检查部署结果。

您还可以登录[EDAS控制台](#)，在所部署应用的基本信息页面查看部署结果。

3.3. 使用Maven部署应用到EDAS

3.3.1. Maven部署说明

使用Maven将应用部署到EDAS需要创建配置文件、账号文件等，您可以根据实际自定义配置文件、账号文件的内容。本文将介绍使用Maven将应用部署到EDAS的配置项说明、指定配置文件说明、账号配置优先级说明及相关使用示例。

配置项

部署应用的配置项可分成三大类：

- 基本环境 (ENV)
- 应用配置 (APP)
- 存储配置 (OSS)

目前支持的配置项如下表所示：

类型	key	是否必须	说明
ENV	region_id	是	应用所在的区域ID。
	endpoint	否	用于专有云设置POP网关接入点。
	app_id	是	应用ID。
	package_version	否	部署包的版本。默认为pom文件的version加上当前机器构建的时间，格式如：“1.0 (2018-09-27 19:00:00)”。
	desc	否	部署的描述。
	group_id	否	部署分组ID。默认为所有分组。
	batch	否	部署分批。默认为1批，最大为5批。

APP 类型	key	是否必须	说明
	batch_wait_time	否	部署分批之间的等待时间，单位为分钟。默认不等待。
	stage_timeout	否	展示每个变更流程stage状态的超时时间，单位为分钟，默认为5分钟。如果同时设置了batch_wait_time，那么此参数在计算时会自动加上batch_wait_time。在运行时，如果某个stage等待时间超过此阈值，那么此插件会自动退出。
OSS	region_id	否	目标存储桶所在的区域ID。默认使用应用所在的区域ID。
	bucket	否	目标存储桶名称。默认使用EDAS提供的免费OSS存储空间。若指定了OSS配置，则必须指定bucket参数，否则使用EDAS自动分配的免费OSS存储空间。
	key	否	应用包上传到OSS的自定义路径，默认使用EDAS提供的免费OSS存储空间。若使用指定的OSS存储，则可通过该参数指明包存储路径，同时可以使用以下变量来进行参数化的路径配置 {region_id}, {app_id}, {version}，例如： pkgs/petstore/{version}/store.war该配置默认为 {region_id}/{app_id}/{version}。
	access_key_id	否	应用包上传到OSS的自定义账号ID。
	access_key_secret	否	应用包上传到OSS的自定义账号密钥。

示例一

用户A在Region为北京有一个ID为 eb20dc8a-xxx 的应用，若需将版本为1.2的应用部署在分组ID为 06923bb9-xxx 分组下，则配置文件如下：

```
env:
  region_id: cn-beijing
app:
  app_id: eb20dc8a-xxx
  package_version: 1.2
  group_id: 06923bb9-xxx
```

示例二

用户B需部署一个ID为eb20dc8a-xxx的应用，并将部署包上传到自己在北京Region的名为 `release-pkg` 的存储桶中的 `my.war` 文件下。OSS账号ID为ABC，OSS账号密钥为1234567890。则配置文件如下：

```
env:
  region_id: cn-beijing
app:
  app_id: eb20dc8a-xxx
oss:
  region_id: cn-beijing
  bucket: release-pkg
  key: my.war
  access_key_id: ABC
  access_key_secret: 1234567890
```

指定配置文件

指定配置文件的方式有以下两种：

- 在任意目录下新建配置文件，通过设置参数 `-Dedas_config={配置文件路径}` 来指定配置文件。
- 当未设置参数指定配置文件时，Cloud Toolkit会默认使用被打包的工程根目录下的 `.edas_config.yaml` 文件作为配置文件。若被打包的工程为一个Maven工程的子模块，则默认使用该子模块的根目录下的 `.edas_config.yaml` 文件。

 **注意** 如果既存在默认配置文件，也通过参数指定配置文件，Cloud Toolkit会优先使用参数指定的配置文件。

账号配置及优先级

使用Cloud Toolkit将应用部署到云端时，需要使用阿里云上的资源。因此在部署应用前，需要设置您的阿里云账号信息，确保拥有使用和管理相关资源、应用的权限。目前Cloud Toolkit支持多种配置方式，优先级从高到低如下：

 **说明** 当重复配置时，优先级高的配置方式会覆盖优先级低的配置方式。

- 使用命令行指定AccessKeyId和AccessKeySecret参数，有以下两种方式。
 - 在使用Maven命令打包时，通过命令 `-Daccess_key_id=xx -Daccess_key_secret=xx` 来指定。

- 在Pom文件中配置Cloud Toolkit时，插入AccessKeyId和AccessKeySecret参数配置，示例如下：

```
<plugin>
<groupId>com.aliyun</groupId>
<artifactId>edas-maven-plugin</artifactId>
<version>2.30.0</version>
<configuration>
<accessKeyId>abc</accessKeyId>
<accessKeySecret>1234567890</accessKeySecret>
</configuration>
</plugin>
```

- 命令行指定账号文件（推荐）。
在使用Maven命令打包时，通过 `-Daccess_key_file={账号文件路径}` 来指定yaml格式的账号文件。账号文件示例如下：

```
access_key_id: abc
access_key_secret: 1234567890
```

- 使用默认的阿里云账号文件。
如果没有通过以上两种方式指定账号，那么Cloud Toolkit会使用您曾经配置过的阿里云账号进行应用部署。
- 如果您使用过最新的aliyuncli工具并且配置过阿里云账号，那么阿里云会在您当前Home目录下生成一个 `.aliyuncli` 目录，并在 `.aliyuncli` 目录下创建 `credentials` 文件来保存您的账号信息。以Mac系统为例，在 `/Users/用户名/.aliyuncli/credentials` 文件中保存账号信息如下：

```
[default]
aliyun_access_key_secret = 1234567890
aliyun_access_key_id = abc
```

- 如果您使用过老的aliyun工具并且配置过阿里云账号，那么aliyun工具会在您当前Home目录下生成一个 `.aliyun` 目录，并且在 `.aliyun` 目录下创建一个 `config.json` 文件来保存您的账号信息。以Mac系统为例，在 `/Users/用户名/.aliyun/config.json` 文件中保存账号信息如下：

```
{
  "current": "",
  "profiles": [{
    "name": "default",
    "mode": "AK",
    "access_key_id": "",
    "access_key_secret": "",
    "sts_token": "",
    "ram_role_name": "",
    "ram_role_arn": "",
    "ram_session_name": "",
    "private_key": "",
    "key_pair_name": "",
    "expired_seconds": 0,
    "verified": "",
    "region_id": "",
    "output_format": "json",
    "language": "en",
    "site": "",
    "retry_timeout": 0,
    "retry_count": 0
  }, {
    "name": "",
    "mode": "AK",
    "access_key_id": "abc",
    "access_key_secret": "xxx",
    "sts_token": "",
    "ram_role_name": "",
    "ram_role_arn": "",
    "ram_session_name": "",
    "private_key": "",
    "key_pair_name": "",
    "expired_seconds": 0,
    "verified": "",
    "region_id": "cn-hangzhou",
    "output_format": "json",
    "language": "en",
    "site": "",
    "retry_timeout": 0,
    "retry_count": 0
  }],
  "meta_path": ""
}
```

- 系统环境变量。
若您未采用上述任何一种方式设置账号文件，Cloud Toolkit会尝试通过系统环境变量来获取 access key id和access key secret的值（即通过Java代码的 `System.getenv("access_key_id")` 和 `System.getenv("access_key_secret")` 来获取相应的值）。

3.3.2. 使用toolkit-maven-plugin部署应用到EDAS

Cloud Toolkit支持使用Maven部署应用到EDAS，优化了传统部署方式（先将应用打包成WAR包或JAR包再到控制台部署应用），使用命令行即可快速部署。本文将介绍如何部署应用到EDAS的ECS集群为例介绍如何使用 toolkit-maven-plugin。

前提条件

- 已下载并安装Maven。
- 已在EDAS控制台上创建应用并完成首次部署。
 - 在ECS集群中创建应用请参见[在ECS集群中创建并部署应用](#)。
 - 在容器服务K8s集群中创建应用请参见[在容器服务集群中创建应用](#)。

自动化部署应用

1. 在您的打包工程的pom.xml文件中增加如下的插件依赖。

```
<build>
<plugins>
<plugin>
  <groupId>com.alibaba.cloud</groupId>
  <artifactId>toolkit-maven-plugin</artifactId>
  <version>1.0.3</version>
</plugin>
</plugins>
</build>
```

 说明 您还可以从Maven官方中央仓库中获取该插件最新版本。

2. 在工程目录下创建一个文件格式为yaml的 .edas_config.yaml 文件。

说明

- 如果部署的工程为Maven的子模块，那么请在子模块的目录下创建该文件。
- 以下配置参数为示例参数，请替换成您的实际的应用参数。例如部署分批等更多配置参数请参见[配置项](#)。
- 关于多模块工程部署更多方式，请参考[部署多模块工程](#)。

```
env:
  region_id: cn-beijing # 应用实例所在的区域 ID
app:
  app_id: eb20****-e6ee-4f6d-a36f-5f6a5455**** # 您想部署应用的 ID
```

若您不知如何设置以上参数值，请参照以下步骤获取：

- i. 登录EDAS控制台，在页面左上角选择所需地域。
- ii. 在左侧导航栏中选择应用管理 > 应用列表，在应用列表页面单击待部署应用的应用名称。
- iii. 在基本信息页面右上角单击部署应用。
- iv. 在部署模式选择页面选择常规发布（单批/多批）开始部署卡片右上角的开始部署。

- v. 单击展开生成Maven插件配置，获取参数信息。

```

生成Maven插件配置

# 此配置根据您的输入自动生成，关于配置使用详情参考如下文档：
# //help.aliyun.com/document_detail/92242.html
---
env:
  region_id: cn-hangzhou
app:
  app_id: 
  group_id: all
  batch: 2
  batch_wait_time: 0

```

3. 在任意目录下新建一个YAML类的账号文件，并添加AccessKey ID和AccessKey Secret账号信息。

```

access_key_id:<yourAccessKeyId>
access_key_secret:<yourAccessKeySecret>

```

说明

- 该配置中Access Key ID和Access Key Secret只会用来生成请求签名，不会用于网络传输等其他用途。
- AccessKey ID和AccessKey Secret可登录阿里云[用户信息管理](#)查看，推荐使用RAM子账号（需已授予应用管理权限）来控制应用权限以提高应用安全。

4. 进入您的工程目录，执行以下打包命令。

说明 如果为Maven多模块则进入子模块目录。

```

mvn clean package toolkit:deploy -Daccess_key_file={账号文件路径}

```

- toolkit:deploy**：使用toolkit-maven-plugin在打包成功后进行应用部署。只有添加此参数才会进行应用部署。
- access_key_file**：阿里云账号文件。更多指定密钥对的方式请参考[账号配置及优先级](#)。

结果验证

执行命令后，终端开始输出日志。若输出以下日志则说明部署成功。

```

[INFO] Waiting...
[INFO] StageName:Enable SLB,
[INFO] ServiceStageName:Enable SLB,
[INFO] StageName:Process Complete,
[INFO] ServiceStageName:Process Complete,
[INFO] Deploy application successfully!
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 31.574 s
[INFO] Finished at: 2019-02-27T18:25:01+08:00
[INFO] -----

```

部署多模块工程

多模块工程是常见的项目组织形式，此Maven插件支持以下两种方式来部署您的多模块工程。

- 在父工程中执行打包及部署命令：在父工程中执行 `toolkit:deploy` 命令时，增加 `-Ddeploy_artifacts` 参数来指定需要部署的子模块Artifact ID。如果希望部署多个子模块，以半角逗号分隔。例如：一个CarShop工程存在以下示例的子模块：

```

carshop
itemcenter-api
itemcenter
detail

```

如果想同时部署itemcenter和detail子模块，您可以在carshop目录下，执行以下命令：

```

mvn clean package toolkit:deploy -Ddeploy_artifacts=itemcenter,detail

```

默认情况下，此插件会根据itemcenter和detail模块下的 `.edas_config.yaml` 文件来分别部署对应的应用。您也可以通过增加 `-Dedas_config` 参数来指定配置文件，相关操作请详见[指定配置文件](#)。

 **注意** 此方式要求toolkit-maven-plugin版本为1.0.3及以上。

- 在子模块中执行打包及部署命令：
 - 在父工程中执行 `install` 命令，将子模块依赖安装到本地的Maven仓库。

ii. 进入到需要部署的子模块目录下执行 `toolkit:deploy` 命令。

 说明 此方式支持 toolkit-maven-plugin 所有版本。

更多信息

更多配置项说明、指定配置文件说明和账号配置及优先级说明信息请参见 [Maven部署说明](#)。

3.4. 使用Cloud Toolkit实现服务的端云互联

3.4.1. 端云互联简介

在开发应用时，可以使用Alibaba Cloud Toolkit插件实现本地应用和部署在EDAS中的应用的相互调用，即端云互联，帮助您提升开发效率。

端云互联支持的IDE

您可以在IntelliJ IDEA和Eclipse这两种IDE中使用Cloud Toolkit的端云互联功能，实现本地应用和云上应用的相互调用。

- [使用Cloud Toolkit实现端云互联（IntelliJ IDEA）](#)
- [使用Cloud Toolkit实现端云互联（Eclipse）](#)

使用限制

端云互联目前支持Java应用，而且不同Java微服务框架还有以下限制：

微服务框架	使用限制
Spring Cloud	如果使用Nacos进行配置管理，请确保Spring Cloud为Spring CloudEdgware及以上版本。
Dubbo	<ul style="list-style-type: none"> • Dubbo 2.7.2及以上版本 • 依赖的服务注册及发现组件版本： <ul style="list-style-type: none"> ◦ dubbo-nacos-registry 2.7.2及以上版本 ◦ edas-dubbo-extension 2.0.2及以上版本
HSF	无

3.4.2. 本地开发及联调

使用Cloud Toolkit可以在本地开发环境中直接启动本地注册中心或加入自建的注册中心，更方便地与他人进行联调，免去修改配置文件、绑定Host等繁杂操作。您可以在IntelliJ IDEA中使用Cloud Toolkit的本地注册中心来实现本地和云上应用的相互调用，提升开发效率。

前提条件

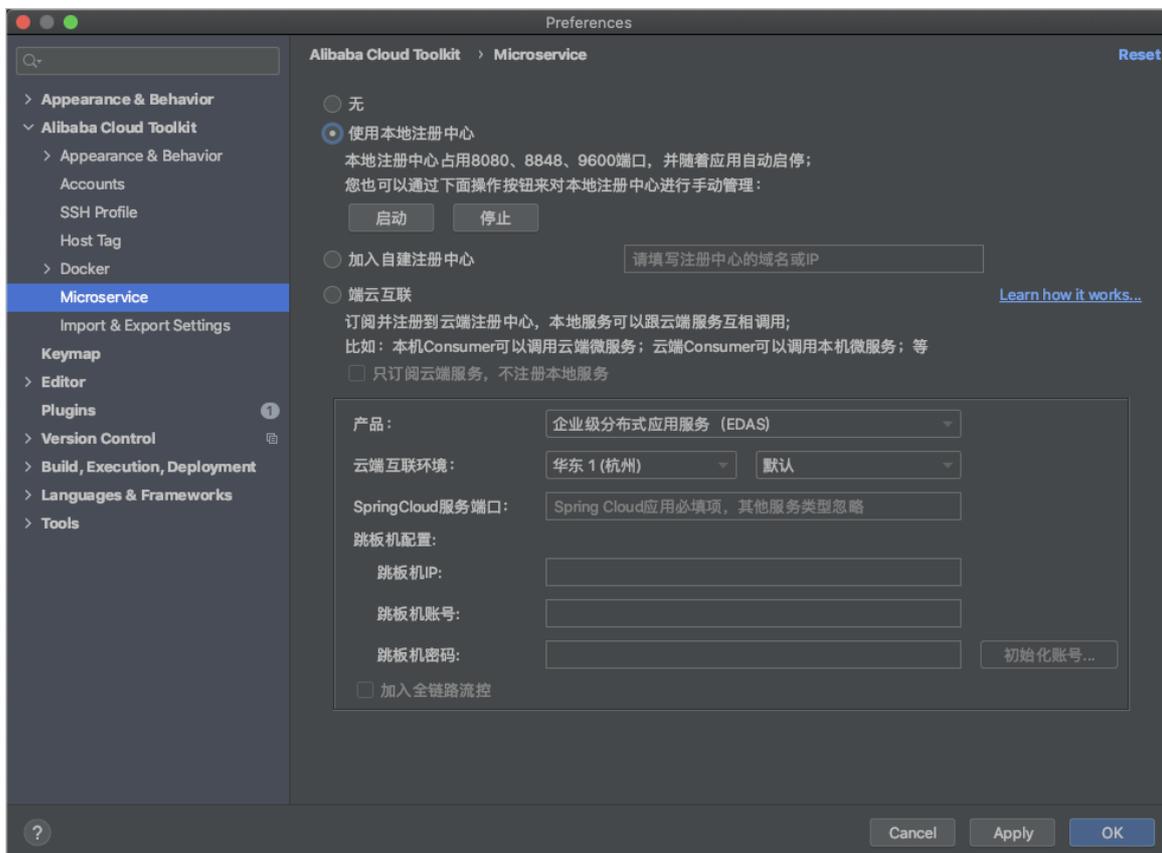
已安装和配置Cloud Toolkit，详情请参见 [在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

使用限制

该功能支持HSF、Spring Cloud和Dubbo等微服务框架，但只能使用本地注册中心，本地注册中心必须要占用8080、8848和9600端口。

在本地开发应用并完成联调

1. 下载nacos-service-provider和nacos-service-consumer工程，在IntelliJ IDEA中打开这两个工程，对这两个工程均进行步骤2-5的配置。
2. 在IntelliJ IDEA顶部菜单栏选择File > Settings。
3. 在Settings页面左侧导航栏选择Alibaba Cloud Toolkit > Microservice，选择注册方式。



配置参数说明如下：

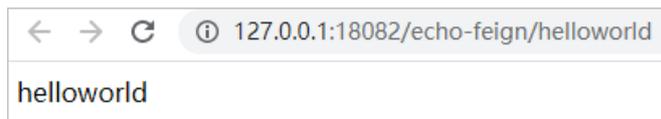
参数	描述
无	不需要注册中心。
使用本地注册中心	<p>使用本地轻量级配置及注册中心，轻量级配置及注册中心包含3个端口：</p> <ul style="list-style-type: none"> 8848：支持Nacos的服务注册和配置管理。 9600：HSF/Dubbo的服务注册及订阅。 8080：支持ANS的服务注册以及ACM的配置管理。 <p>在选择使用本地注册中心后，单击启动和停止即可启动或停止本地注册中心，然后通过 <code>http://localhost:<指定端口></code> 即可访问不同的注册中心控制台。</p>
加入自建注册中心	<p>填写您自建的注册中心的域名或IP，在本地运行应用时，EDAS会通过-D参数的形式为应用配置该注册中心并将应用注册到该注册中心。</p>
端云互联	<p>实现本地服务和云端服务的相互调用，用于联调测试，详情请参见端云互联简介。</p>

 **说明** 本文以选择使用本地注册中心为例进行结果验证说明。

4. 配置参数完成后，单击Apply和OK。
5. 运行nacos-service-provider和nacos-service-consumer工程。

执行结果

在浏览器中访问<http://127.0.0.1:18082/echo-feign/helloworld>，18082是consumer的端口。如果显示helloworld字符串，即表示实现了这两个工程的本地联调。



3.4.3. 使用Cloud Toolkit实现端云互联（IntelliJ IDEA）

您可以在IntelliJ IDEA中使用Cloud Toolkit的端云互联功能实现本地和云上应用的相互调用，提升开发效率。

前提条件

在使用Cloud Toolkit实现端云互联前，请完成以下工作：

- 确保IntelliJ IDEA为2018.3及以上版本。
- 2020.9.1及以上版本的Cloud Toolkit能够基于本地工程的项目（Project）和模块（Module）粒度配置端云互联，如果想使用配置粒度功能，请将Cloud Toolkit升级到2020.9.1及以上版本。

 **说明** 如果Cloud Toolkit已经是2020.9.1版本，但没有配置粒度选项，请您卸载再重新安装Cloud Toolkit插件。

- 创建一台可使用SSH登录的ECS，用于建立端云互联通道。具体操作，请参见[通过控制台使用ECS实例（快捷版）云服务器ECS快速入门](#)。

注意

- 请确保该ECS实例和需要互联的应用在同一个VPC内。
- SSH通道需要使用密码方式登录，暂不支持使用密钥对登录。

使用限制

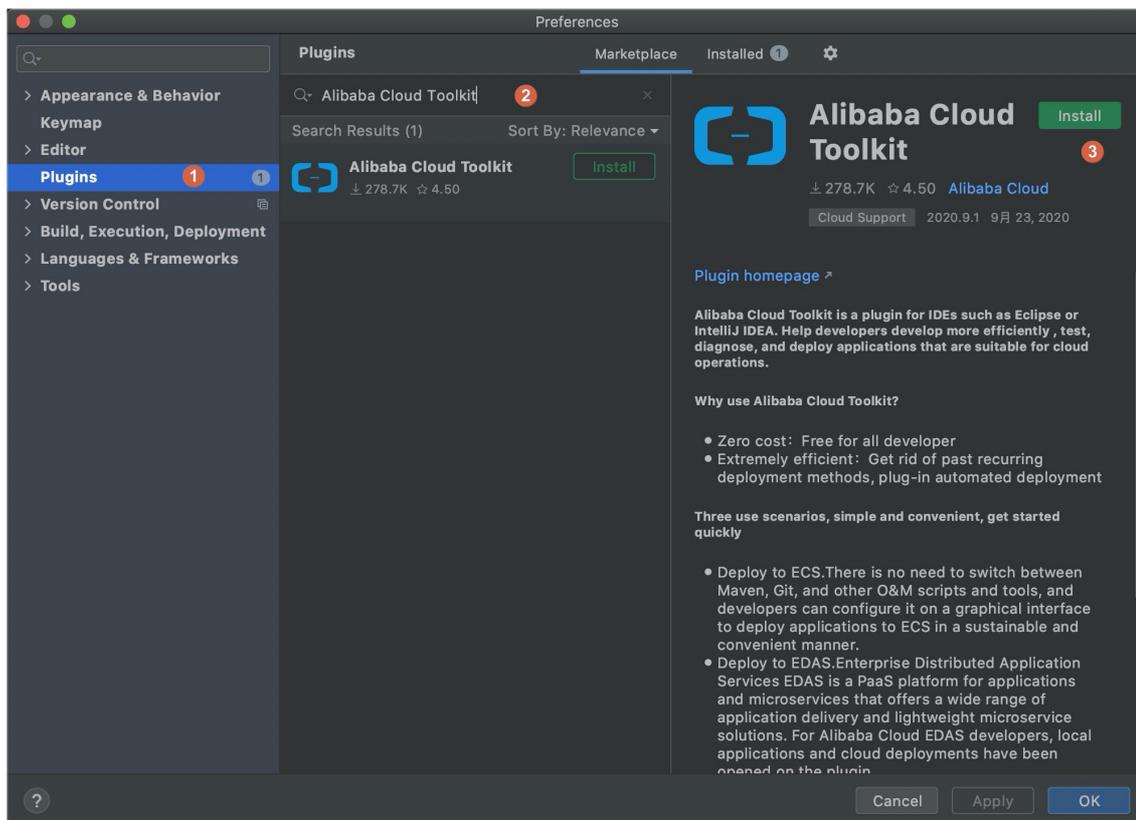
端云互联目前支持Java应用，而且不同Java微服务框架还有以下限制：

微服务框架	使用限制
Spring Cloud	如果使用Nacos进行配置管理，请确保Spring Cloud为Spring CloudEdgware及以上版本。
Dubbo	<ul style="list-style-type: none"> • Dubbo 2.7.2及以上版本 • 依赖的服务注册及发现组件版本： <ul style="list-style-type: none"> ◦ dubbo-nacos-registry 2.7.2及以上版本 ◦ edas-dubbo-extension 2.0.2及以上版本

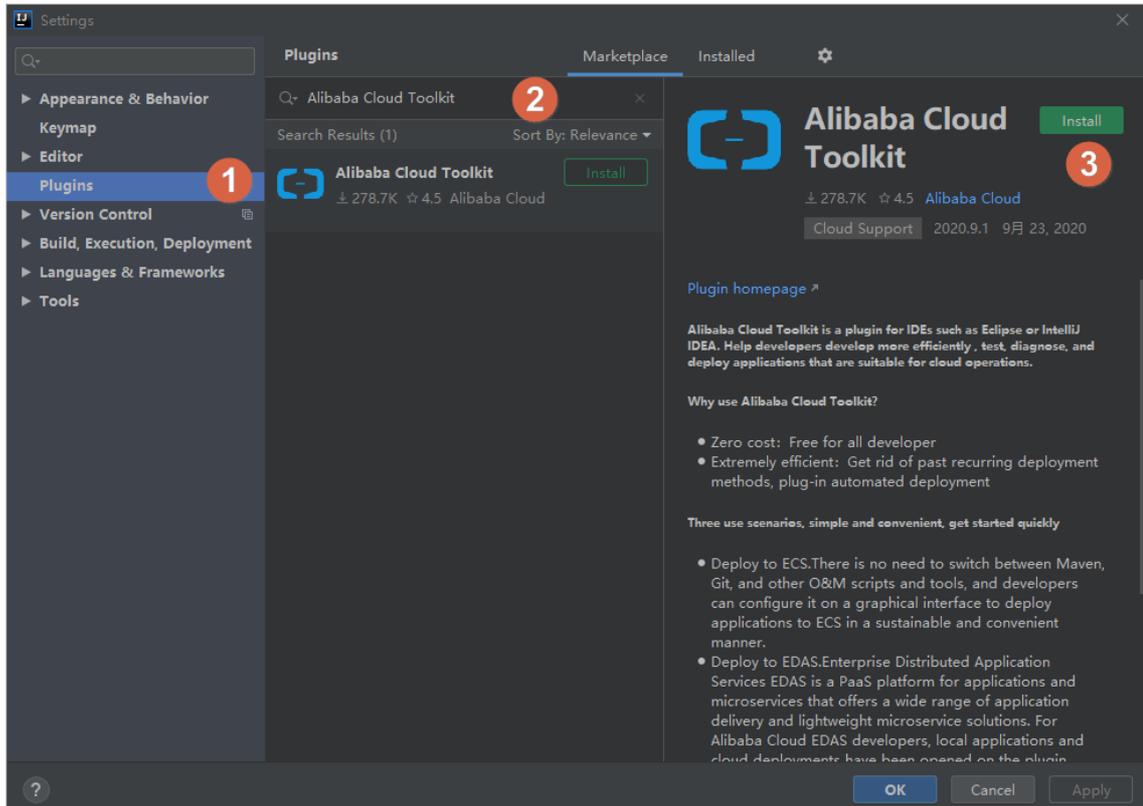
微服务框架	使用限制
HSF	无

步骤一：安装Cloud Toolkit

1. 启动IntelliJ IDEA。
2. 在IntelliJ IDEA中安装插件。
 - o **macOS系统**：在顶部菜单栏选择IntelliJ IDEA > Preference...，在Preference配置页面左边导航栏单击Plugins，搜索Alibaba Cloud Toolkit，并单击Install安装。



- o **Windows系统**：在顶部菜单栏选择File > Settings，在Settings页面的左侧导航栏单击Plugins，搜索Alibaba Cloud Toolkit，并单击Install安装。

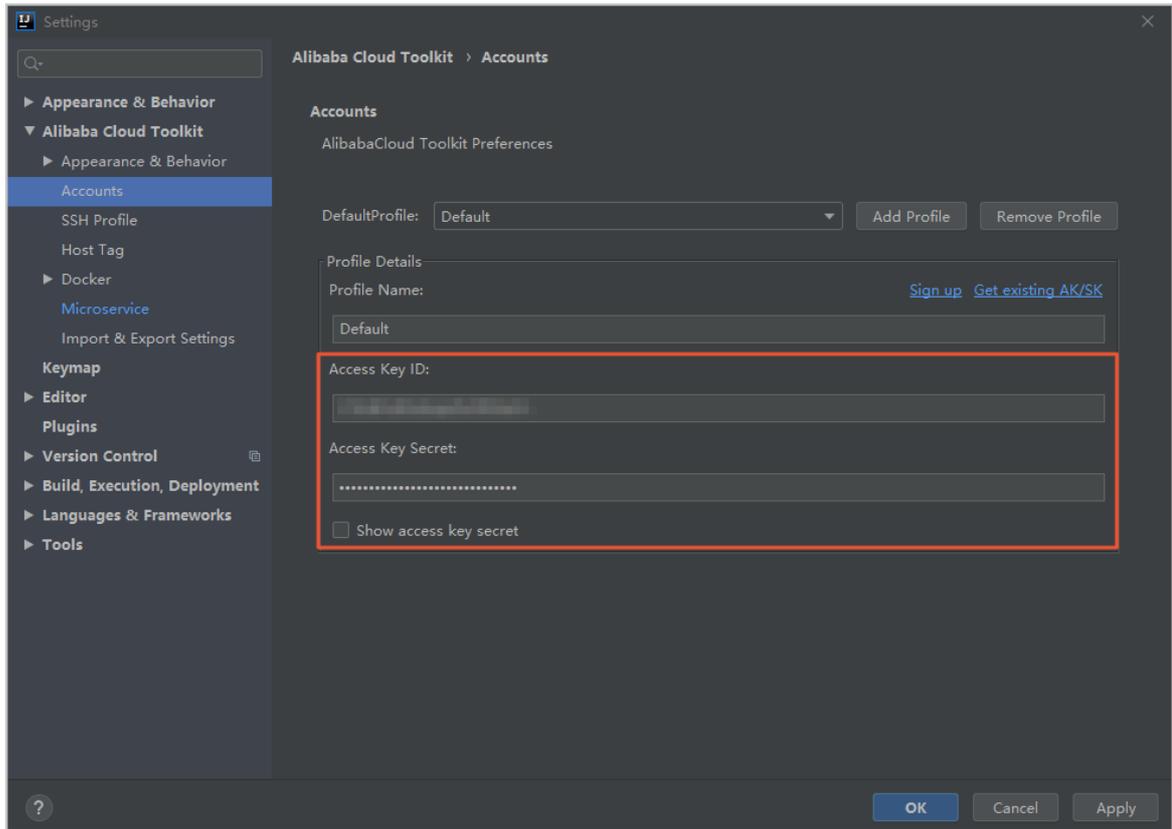


3. 在IntelliJ IDEA中插件安装成功后，重启IntelliJ IDEA，您可以在工具栏看到Alibaba Cloud Toolkit的图标。

步骤二：配置Cloud Toolkit账号

在安装完Alibaba Cloud Toolkit后，您需使用AccessKey ID和AccessKey Secret来配置Cloud Toolkit的账号。

1. 启动IntelliJ IDEA。
2. 在顶部菜单栏中选择Tools > Alibaba Cloud > Preferences...
3. 在Settings对话框中选择Alibaba Cloud Toolkit > Accounts。
4. 在Accounts界面中设置AccessKey ID和AccessKey Secret，然后单击OK。如果您使用子账号的AccessKey ID和AccessKey Secret，请确认该子账号至少拥有部署应用的权限，具体操作，请参见为RAM用户授权。

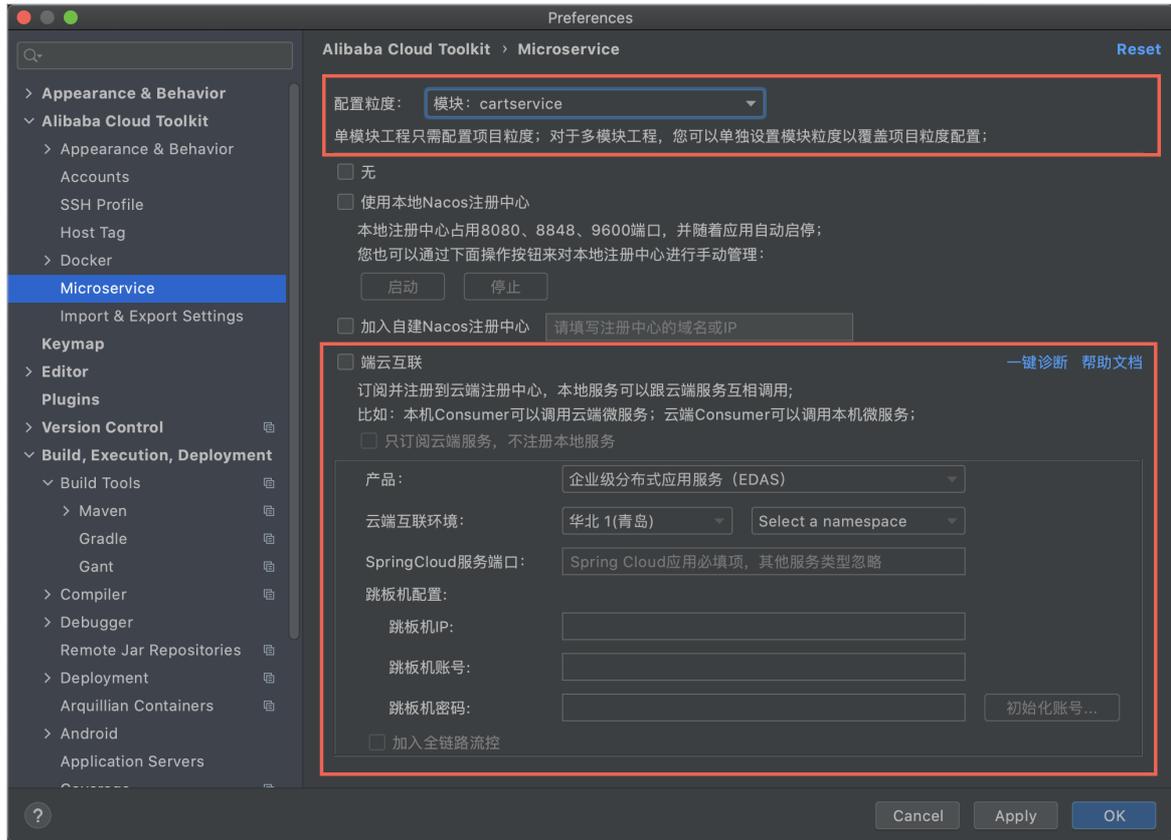


关于阿里云账号说明如下：

- 如果您已经注册过阿里云账号，在Accounts界面中单击Get existing AK/SK，进入阿里云登录页面。用已有账号登录后，跳转至安全管理页面，获取AccessKey ID和AccessKey Secret。
- 如果您还没有阿里云账号，在Accounts界面中单击Sign up，进入阿里云账号注册页面，注册账号。注册完成后按照上述方式获取AccessKey ID和AccessKey Secret。

步骤三：端云互联配置

1. 启动IntelliJ IDEA。
2. 在顶部菜单栏中选择Tools > Alibaba Cloud > Preferences...
3. 在Settings对话框中选择Alibaba Cloud Toolkit > Microservice。
4. 在Microservice界面中配置端云互联相关参数。



参数	描述
配置粒度	<p>本地工程需要使用端云互联功能的粒度，包含项目和模块。</p> <ul style="list-style-type: none"> 项目：当前工程使用统一的端云互联配置。适用于单模块工程或多模块工程的模块间无配置差异。 模块：当前工程包含多个模块，其中仅某个模块需要使用端云互联或不同模块的端云互联配置需求不同。如果需要为不同模块配置端云互联，选择具体模块，完成各自的端云互联配置。
端云互联	<p>选中端云互联启用端云互联功能。</p> <p>启用端云互联功能后，本地应用默认注册到云端注册中心并订阅云端注册中心的服务，本地应用中的服务可以和云端服务相互调用。</p> <p>如果仅需要调用云端服务，不希望云端服务调用本地服务，可以选中只订阅云端服务，不注册本地服务。</p>
产品	选择企业级分布式应用服务（EDAS）。

参数	描述
端云互联环境	<p>设置需要调用的云端应用所在的地域和命名空间。除了默认命名空间外，其他命名空间下的应用需按照以下步骤手动打开允许远程调试开关。</p> <ol style="list-style-type: none"> 登录EDAS控制台。 在左侧导航栏选择应用管理 > 命名空间，然后在顶部菜单栏选择地域。 在命名空间列表中目标命名空间的操作列单击编辑按钮。 在编辑命名空间对话框中打开允许远程调试开关。
SpringCloud服务端口	<p>如果是Spring Cloud应用，则需在SpringCloud服务端口文本框内添加该应用的服务端口，其他类型应用不需要填写。</p>
跳板机配置	<ul style="list-style-type: none"> 跳板机IP：您创建的跳板机（ECS实例）的公网IP。 跳板机账号：输入用于建立端云互联通道的用户名。 跳板机密码：输入用于建立端云互联通道的密码。 <p> 说明 您可以直接输入用于建立端云互联通道的ECS实例的用户名和密码，也可以输入新的用户名和密码，然后单击下面的初始化账号...进行配置。</p>
初始化账号...	<ul style="list-style-type: none"> 如果输入的是ECS实例的root用户名和密码，则会使用此root账号进行配置，如果成功则会出现配置已添加成功的提示弹窗。 如果使用新账号或其他非root账号，那么需要root权限来对此账号进行代理配置，在Add SSH Rule对话框中输入Password，然后单击Add即可。 <p> 注意</p> <ul style="list-style-type: none"> 此处使用ECS实例的密码只是用来创建一个网络代理，不会将ECS实例的用户名和密码用于其他用途。 推荐使用新账号或其他非root账号进行互联，后续可将此新账号或非root账号直接共享给其他需要端云互联的团队人员使用，避免泄漏root信息。

参数	描述
一键诊断	端云互联过程中，如果遇到问题，可以单击 一键诊断 ，排查 <i>etrans</i> 通道启动异常和服务连接不通等问题。

5. 先单击**Apply**，然后单击**OK**。

注意 如果使用EDAS专有云企业版，还需要按以下步骤在Cloud Toolkit中配置Endpoint。Endpoint请联系EDAS技术支持获取。

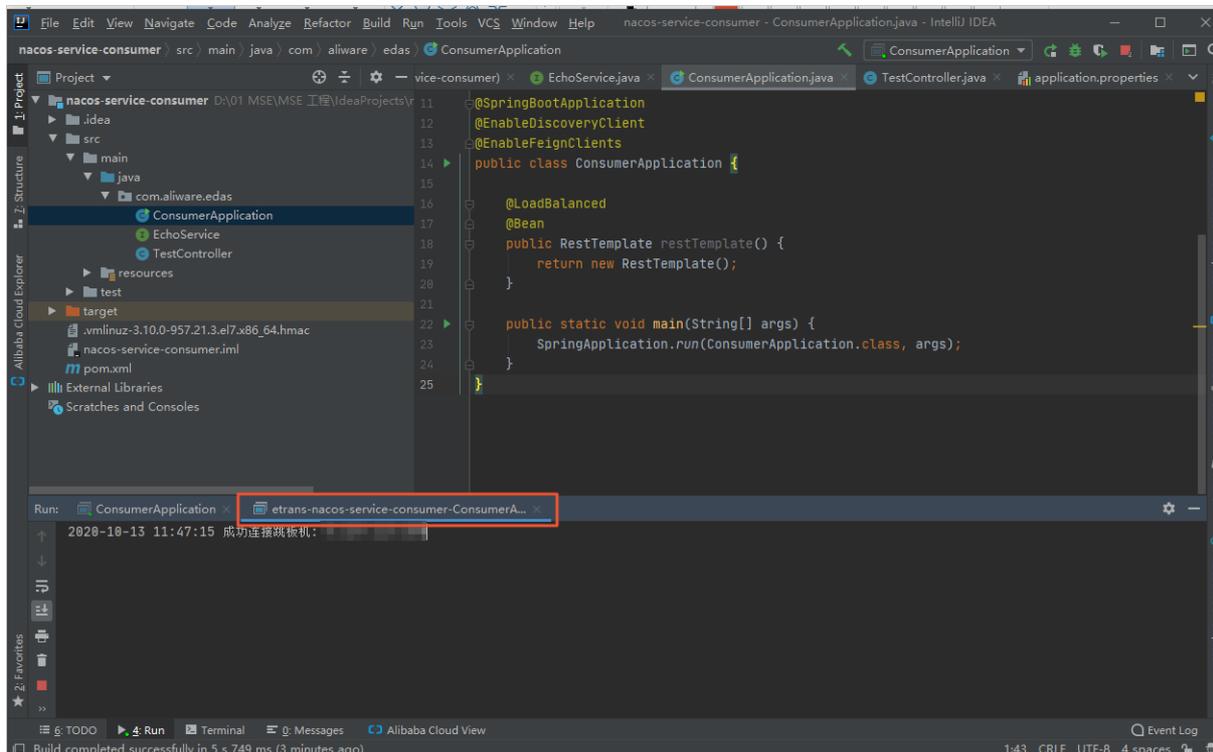
- i. 在**Preference (Filtered)**对话框的左侧导航栏中选择**Appearance & Behavior > Endpoint**。
- ii. 在**Endpoint**界面中设置Endpoint，配置完成后，单击**Apply and Close**。

步骤四：启动本地应用进行端云互联

启动本地应用，如果当前状态处于端云互联状态，那么会有如下提示：



并且，在启动应用之后会启动一个etrans的进程：



说明

端云互联过程中，如果遇到问题，可以单击**一键诊断**，排查 *etrans* 通道启动异常和服务连接不通等问题。

更多信息

- 在使用Cloud Toolkit实现端云互联时，如果遇到相关问题，请参见[端云互联问题](#)。
- 您可以在EDAS上代理购买ECS，具体操作，请参见[创建ECS实例](#)。
- 如果您想使用IntelliJ IDEA插件快速在EDAS上部署应用，请参见[使用IntelliJ IDEA部署应用到EDAS](#)。

3.4.4. 使用Cloud Toolkit实现端云互联（Eclipse）

您可以在Eclipse中使用Cloud Toolkit的端云互联功能实现本地和云上应用的相互调用，提升开发效率。

前提条件

- 确保Eclipse为4.5.0及以上版本。
- 2020.9.1及以上版本的Cloud Toolkit能够基于本地工程的项目（Project）和模块（Module）粒度配置端云互联，如果想使用配置粒度功能，请将Cloud Toolkit升级到2020.9.1及以上版本。

 **说明** 如果Cloud Toolkit已经是2020.9.1版本，但没有配置粒度选项，请您卸载再重新安装Cloud Toolkit插件。

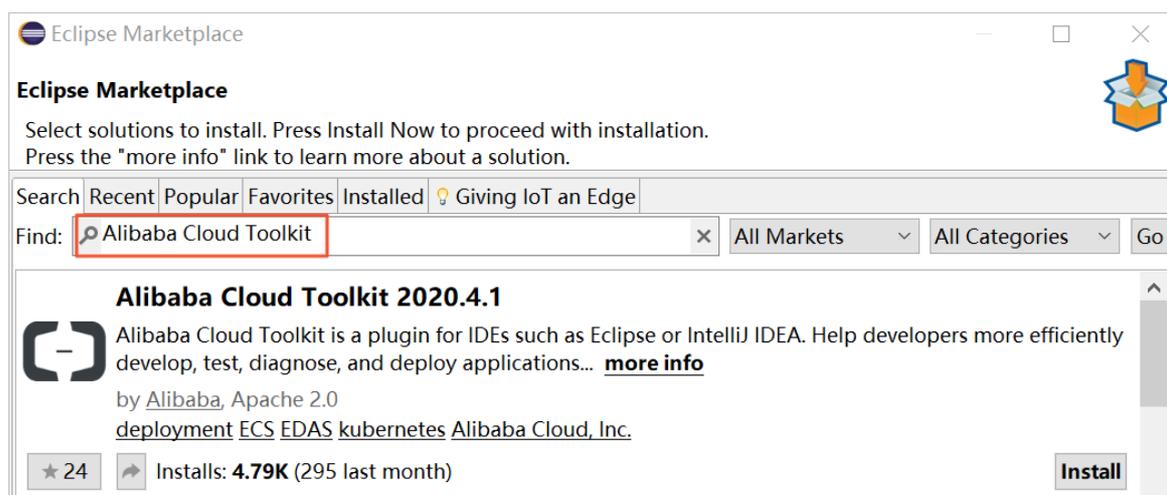
- 创建一台可使用SSH登录的ECS，用于建立端云互联通道。具体操作，请参见[通过控制台使用ECS实例（快捷版）云服务器ECS快速入门](#)。

注意

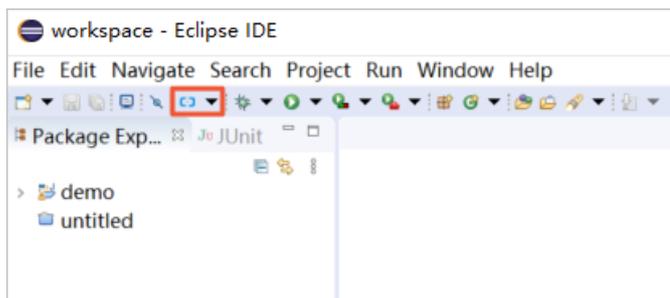
- 请确保该ECS实例和需要互联的应用在同一个VPC内。
- SSH通道需要使用密码方式登录，暂不支持使用密钥对登录。

步骤一：安装Cloud Toolkit

1. 启动Eclipse。
2. 在顶部菜单栏中选择Help > Eclipse Market place。
3. 在Eclipse Market place对话框中Find右侧的文本框中输入Alibaba Cloud Toolkit。



4. 在搜索结果中单击Alibaba Cloud Toolkit 区域右下角的Install。
5. 按照Eclipse安装页面的提示，完成后续安装步骤。
Cloud Toolkit插件安装完成后，重启Eclipse，您可以在工具栏看到Alibaba Cloud Toolkit的图标。

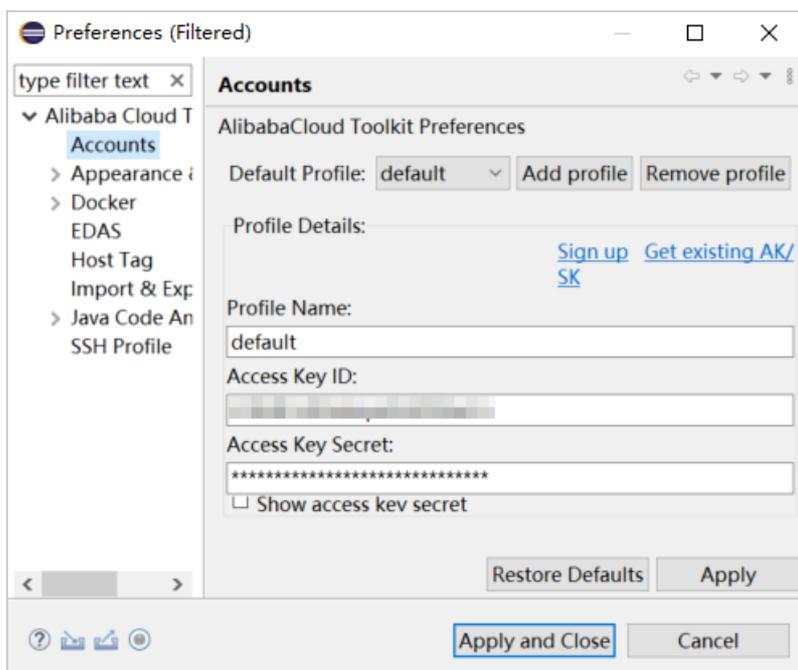


步骤二：配置Cloud Toolkit账号

您需使用AccessKey ID和AccessKey Secret来配置Cloud Toolkit的账号。

1. 启动Eclipse。
2. 在工具栏单击Alibaba Cloud Toolkit图标右侧的下拉按钮，在下拉菜单中单击Preferences...
3. 在Preferences (Filtered) 对话框的左侧导航栏中单击Accounts。
4. 在Accounts界面中设置Access Key ID和Access Key Secret，然后单击Apply。

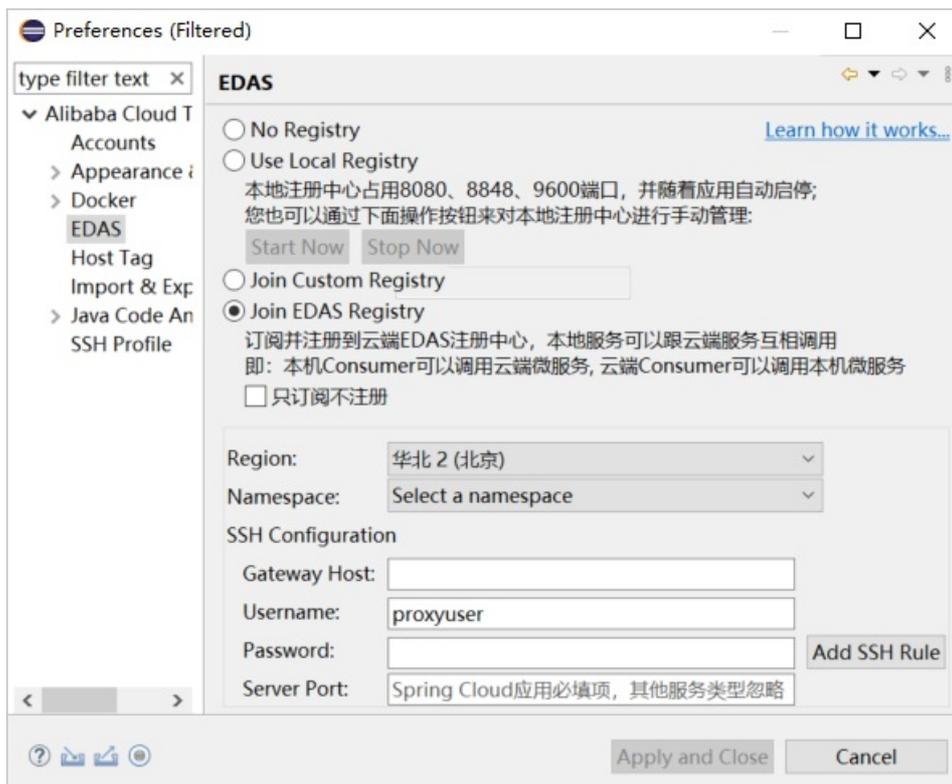
 **注意** 如果您使用RAM用户的AccessKey ID和AccessKey Secret，请确认该RAM用户至少拥有部署应用的权限，具体操作，请参见[RAM账号授权](#)。



- 如果您已经注册过阿里云账号，在Accounts界面中单击Get existing AK/SK，进入阿里云登录页面。用已有账号登录后，跳转至安全信息管理页面，获取Access Key ID和Access Key Secret。
- 如果您还没有阿里云账号，在Accounts界面中单击Sign up，进入阿里云账号注册页面，注册账号。注册完成后按照上述方式获取Access Key ID和Access Key Secret。

步骤三：配置端云互联

1. 在Eclipse中单击工具栏Alibaba Cloud Toolkit的图标 (☁️)，在下拉菜单中单击Preferences...
2. 在Preferences (Filtered) 对话框的左侧导航栏中选择Alibaba Cloud Toolkit > EDAS，在页面右侧设置区域进行端云互联相关配置。



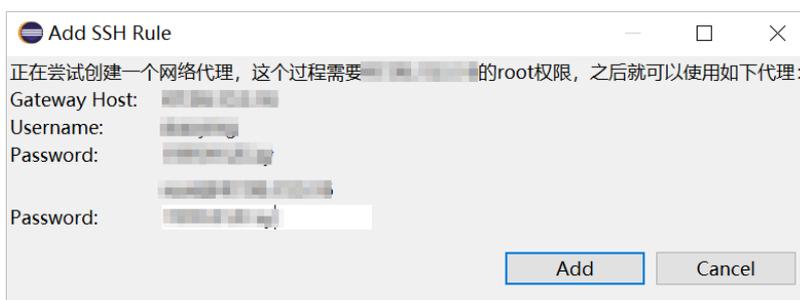
- i. 选中Join EDAS Register开启端云互联功能。
- ii. 设置Region和Namespace为端云互联应用所在的地域和命名空间。
除了默认命名空间外，其他命名空间需手动打开允许远程调试开关。
 - a. 登录EDAS控制台。
 - b. 选择地域，进入应用管理 > 命名空间。
 - c. 在命名空间列表中单击待选择的命名空间操作列的编辑按钮。
 - d. 在编辑命名空间对话框中打开允许远程调试开关。
- iii. 在SSH Configuration区域执行以下操作。
 - a. 在Gateway Host文本框内输入您创建的ECS实例的公网IP。
 - b. 在Username和Password文本框内输入用于建立SSH端云互联通道的用户名和密码：您可以直接输入您用于建立SSH端云互联通道的ECS实例的用户名和密码，也可以在这里填入新的用户名和密码，然后通过下面的Add SSH Rule来增加此新用户及密码。
 - c. 如果是Spring Cloud应用，则您需在Server Port文本框内添加该应用的服务端口，其他类型应用不需要填写。

iv. 单击Add SSH Rule完成配置。

- 如果您输入的是ECS实例的root用户名和密码，则会使用此root账号进行配置，如果成功则会出现配置已添加成功的提示弹窗。



- 如果使用新账号或其他非root账号进行互联，那么需要root权限来对此账号进行代理配置，在Add SSH Rule对话框中输入Password，然后单击Add。



注意

- 此处使用ECS实例的密码只是用来创建一个网络代理，不会将ECS实例的用户名和密码用于其他用途。
- 推荐使用新账号或其他非root账号进行互联，后续可将此新账号或非root账号直接共享给其他需要端云互联的团队成员使用，避免泄漏root信息。

3. 先单击Apply，然后单击OK。

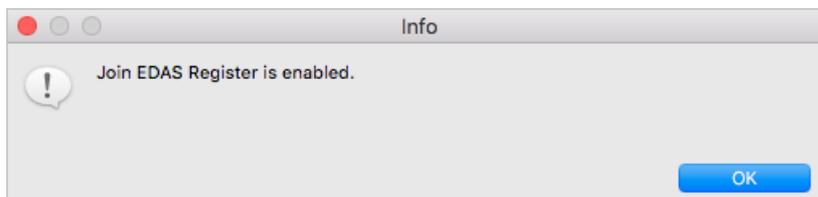
注意 如果使用EDAS专有云企业版，还需要按以下步骤在Cloud Toolkit中配置Endpoint。Endpoint请联系EDAS技术支持获取。

- 在Preference (Filtered) 对话框的左侧导航栏中选择Appearance & Behavior > Endpoint。
- 在Endpoint界面中设置Endpoint，配置完成后，单击Apply and Close。

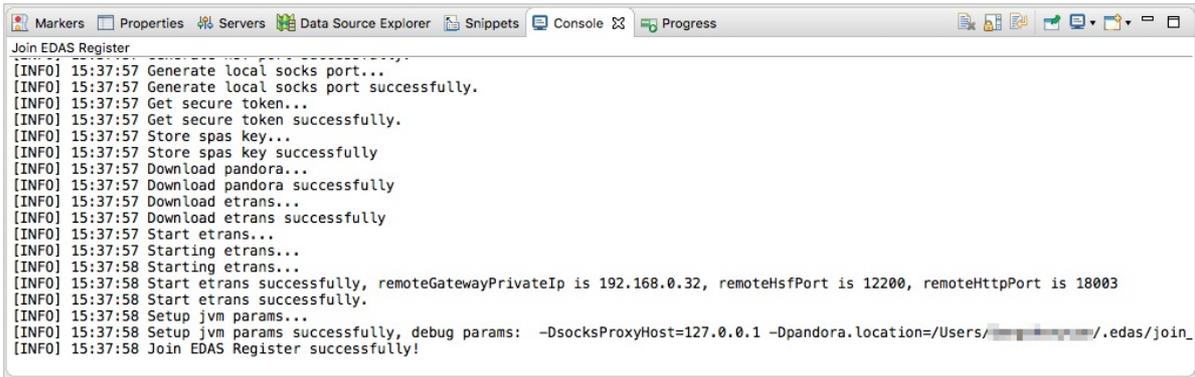
步骤四：启动本地应用进行端云互联

在项目列表中选中工程项目的根目录，然后启动应用。如果当前状态处于端云互联状态，那么会有如下提示：

- 端云互联可用的提示框。



- 在Console面板中会有一个标题为Join EDAS Register的控制台打印初始化端云互联环境的日志。



```
Join EDAS Register
[INFO] 15:37:57 Generate local socks port...
[INFO] 15:37:57 Generate local socks port successfully.
[INFO] 15:37:57 Get secure token...
[INFO] 15:37:57 Get secure token successfully.
[INFO] 15:37:57 Store spas key...
[INFO] 15:37:57 Store spas key successfully
[INFO] 15:37:57 Download pandora...
[INFO] 15:37:57 Download pandora successfully
[INFO] 15:37:57 Download etrans...
[INFO] 15:37:57 Download etrans successfully
[INFO] 15:37:57 Start etrans...
[INFO] 15:37:57 Starting etrans...
[INFO] 15:37:58 Starting etrans...
[INFO] 15:37:58 Start etrans successfully, remoteGatewayPrivateIp is 192.168.0.32, remoteHsfPort is 12200, remoteHttpPort is 18003
[INFO] 15:37:58 Start etrans successfully.
[INFO] 15:37:58 Setup jvm params...
[INFO] 15:37:58 Setup jvm params successfully, debug params: -DsocksProxyHost=127.0.0.1 -Dpandora.location=/Users/.../.edas/join_
[INFO] 15:37:58 Join EDAS Register successfully!
```

更多信息

- 在使用Cloud Toolkit实现端云互联时，如果遇到相关问题，请参见[端云互联问题](#)。
- 您可以在EDAS上代理购买ECS，详情请参见[在ECS集群中代购ECS实例](#)。
- 如果您想使用Eclipse插件快速在EDAS上部署应用，详情请参见[使用Eclipse部署应用到EDAS](#)。

4.部署应用到SAE

4.1. 使用IntelliJ IDEA部署应用到SAE

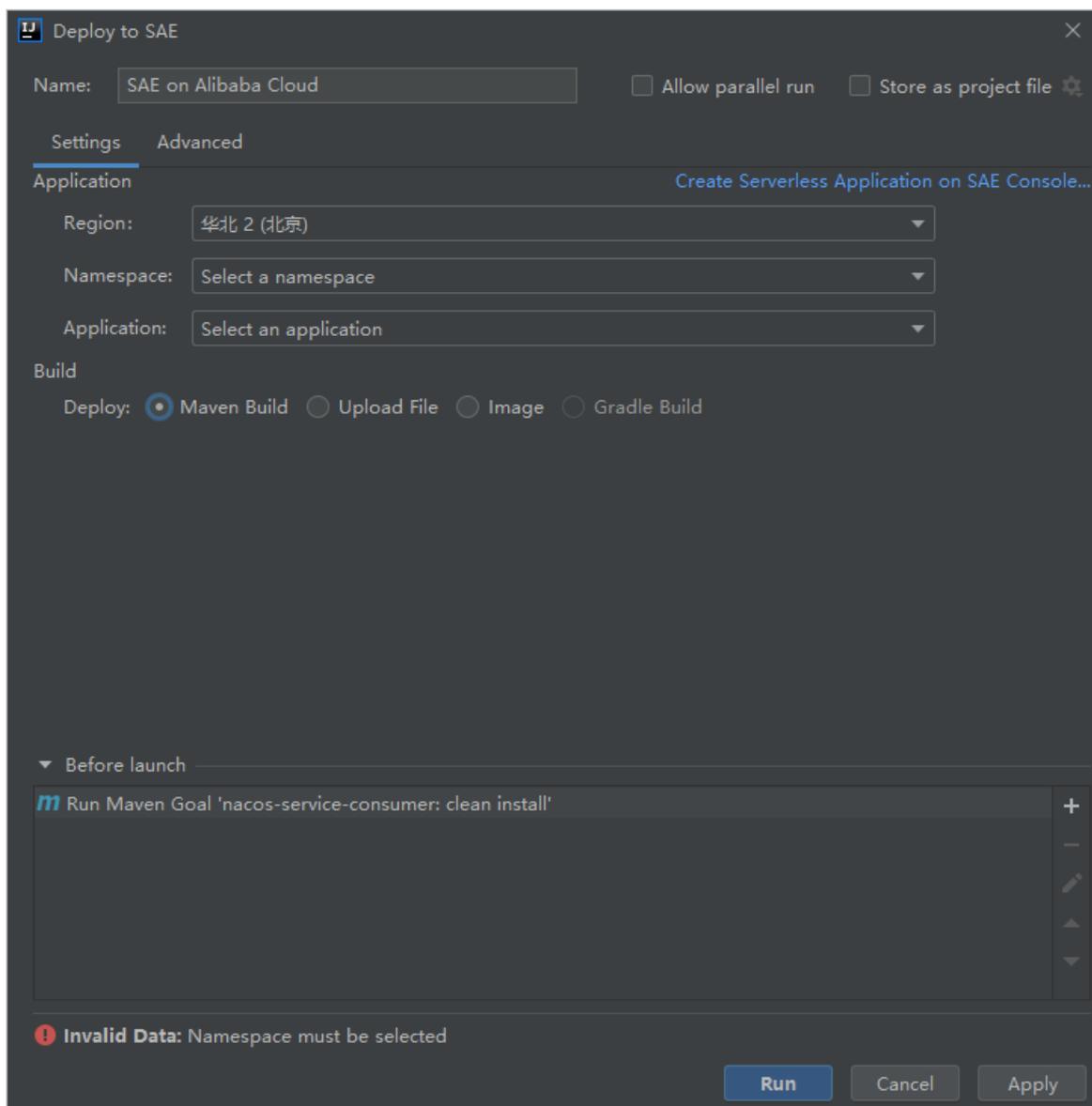
使用Cloud Toolkit可以将应用快速部署到SAE，适用于快速迭代更新应用的场景。在IntelliJ IDEA安装和配置Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

前提条件

- 已成功部署SAE应用，请参见[在控制台部署应用](#)。
- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

操作步骤

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to SAE...
3. 在Deploy to SAE对话框中设置部署参数。



 **说明** 若您尚未在SAE上创建应用，可在对话框右上角单击Create Serverless Application on SAE Console，跳转到SAE控制台创建应用。

部署参数说明如下。

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
部署方式 (Deploy File)	Maven Build	选择Maven Build方式来构建应用时，系统会默认添加一个Maven任务来构建部署包。如果您需要部署多模块工程中的一个子模块，请参见 部署多模块工程中的子模块 。
	Upload File	选择Upload File方式来构建应用时，选择上传您的WAR包或者JAR包，然后进行部署。
	Image	选择Image方式来构建应用时，需要填入一个镜像地址，然后进行部署。
	Gradle Build	选择Gradle Build方式来构建应用时，可以直接构建并部署。

 **说明** 若您已使用Jar/War包部署应用，使用Cloud Toolkit部署应用时只能选择Maven Build或Upload File两种部署方式；若您已使用镜像部署应用，使用Cloud Toolkit部署应用时只能选择Image部署方式。

4. 先单击Apply，然后单击Run。

验证结果

开始部署后，IntelliJ IDEA的Console区域会打印部署日志，可以根据日志信息检查部署结果。

您还可以登录[SAE控制台](#)，在应用详情的变更记录页面查看更新记录。

4.2. 使用Eclipse部署应用到SAE

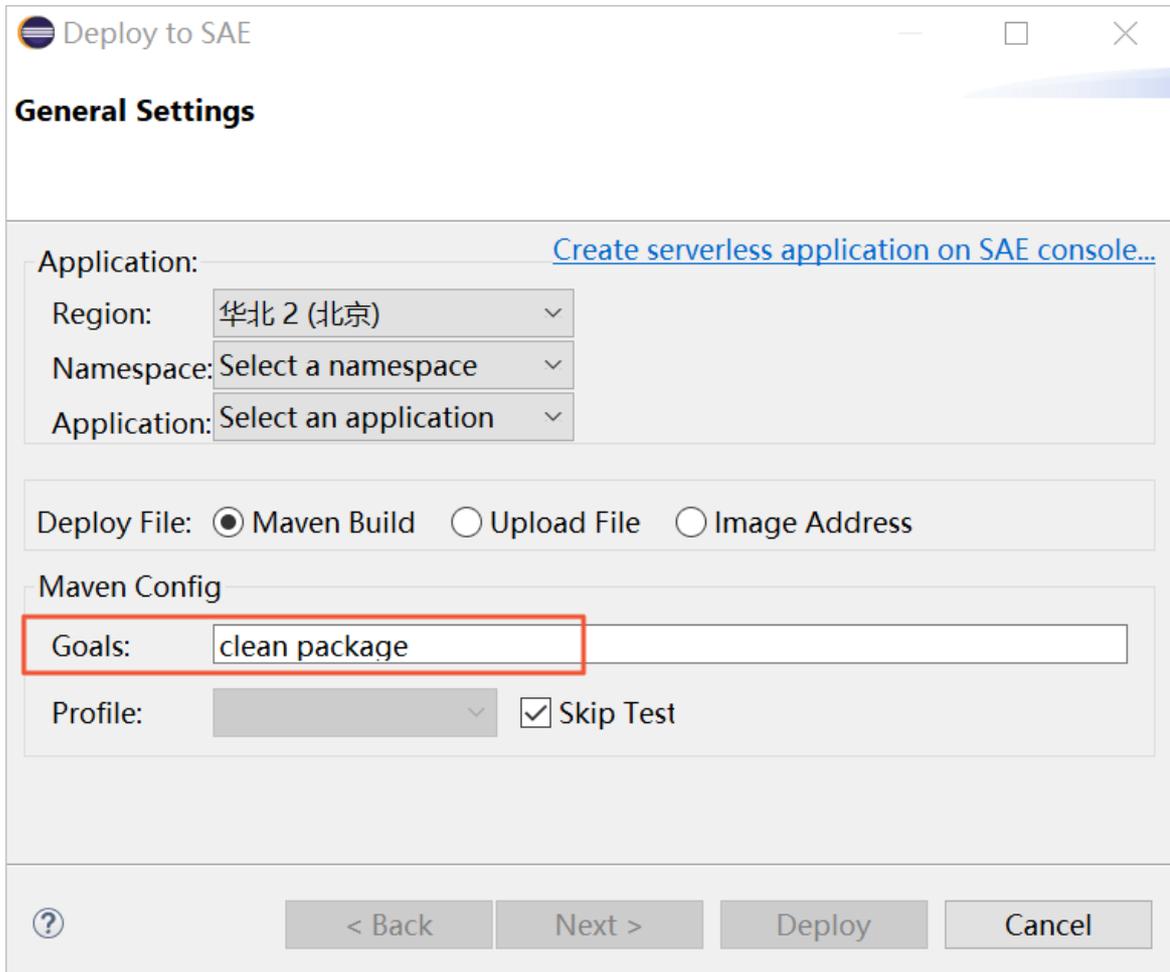
使用Cloud Toolkit可以将应用快速部署到SAE，适用于快速迭代更新应用的场景。在Eclipse中安装和配置Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

前提条件

- 已成功部署SAE应用，请参见[在控制台部署应用](#)。
- 已安装和配置Cloud Toolkit，请参见[在Eclipse中安装和配置Cloud Toolkit](#)。

操作步骤

1. 在Eclipse界面左侧的Project Explorer中右键单击待部署的工程名，在快捷菜单中选择 **Alibaba Cloud > Deploy to SAE...**。
2. 在Deploy to SAE对话框中设置部署参数。



说明 若您尚未在SAE上创建应用，可在对话框右上角单击Create Serverless Application on SAE console，跳转到SAE控制台创建应用。

部署参数说明如下表所示：

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
	Maven Build	选择Maven Build方式来构建应用时，系统会默认添加一个Maven任务来构建部署包。如果您需要部署多模块工程中的一个子模块，请参见 使用Eclipse部署多模块工程中的子模块 。

部署方式 (Deploy File)	参数	描述
	Upload File	选择Upload File方式来构建应用时，选择上传您的WAR包或者JAR包，然后进行部署。
	Image Address	选择Image方式来构建应用时，需要填入一个镜像地址，然后进行部署。

 **说明** 若您已使用Jar/War包部署应用，使用Cloud Toolkit部署应用时只能选择Maven Build或Upload File两种部署方式；若您已使用镜像部署应用，使用Cloud Toolkit部署应用时只能选择Image部署方式。

3. 单击**Deploy**。

结果验证

部署开始后，Eclipse的**Console**区域会打印部署日志，可以根据日志信息检查部署结果。

您还可以登录**SAE控制台**，在应用详情的**变更记录**页面查看更新记录。

5.部署应用到镜像仓库

5.1. 使用IntelliJ IDEA部署应用到镜像仓库

Alibaba Cloud Toolkit 可以将应用快速部署到镜像仓库，提供安全的镜像托管能力。在IntelliJ IDEA安装和配置Alibaba Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

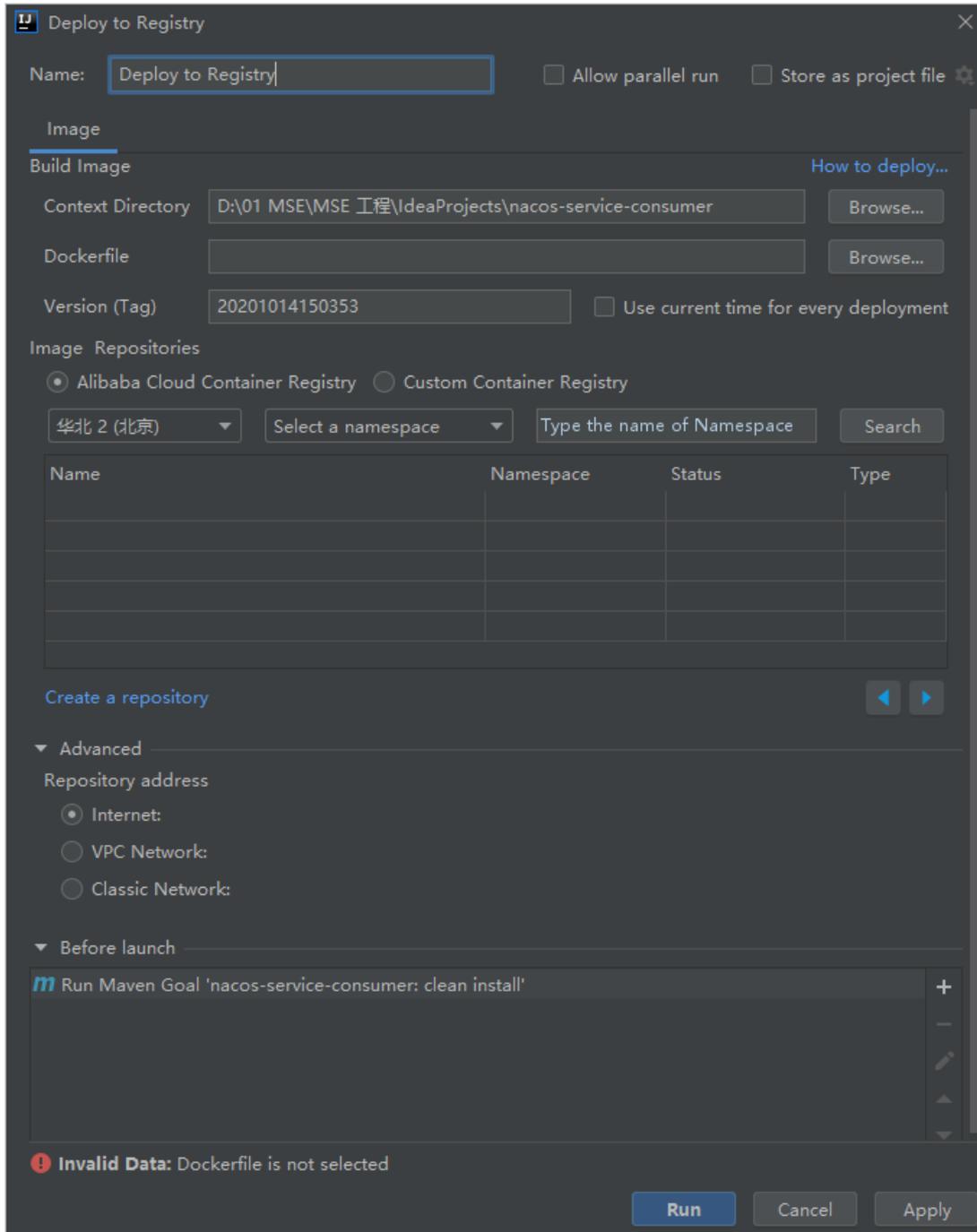
前提条件

- 已安装和配置[Docker](#)。
- 已安装和配置Alibaba Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

镜像仓库配置

使用Cloud Toolkit将应用部署到镜像仓库具体步骤如下：

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to Registry / Kubernetes > Deploy to Registry。
3. 在Deploy to Registry对话框设置部署参数。



部署镜像仓库参数说明如下。

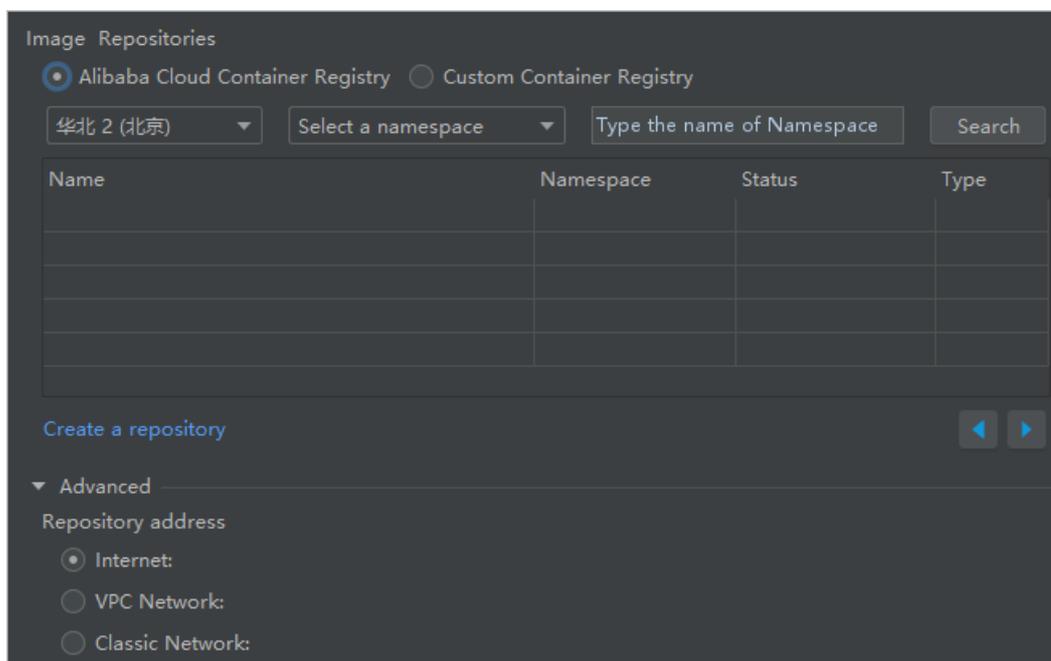
参数	描述
Build Image	<ul style="list-style-type: none">◦ Context Directory: 文件目录。◦ Dockerfile: Docker文件。◦ Version: 镜像版本号。

参数	描述
Image Repositories	<ul style="list-style-type: none"> Alibaba Cloud Container Registry: 阿里云镜像仓库, 详情请参见部署应用到ACR。 Custom Container Registry: 自建镜像仓库, 详情请参见部署应用到其它镜像仓库。

 说明 Context Directory和Dockerfile通常会根据您的本地应用工程自动识别并设置。

部署应用到ACR

1. 在部署参数页面选择Alibaba Cloud Container Registry。



2. 选择地域、命名空间和镜像仓库。
3. 在Advanced下拉选项中选择网络类型。
 - o Internet: 公有网络。
 - o VPC Network: VPC网络。
 - o Classic Network: 经典网络。
4. 先单击Apply, 然后单击Run。

部署应用到其它镜像仓库

1. 在部署参数页面选择Custom Container Registry。
2. 单击界面右侧的Add, 配置Registry信息。
3. 在Registry页面配置镜像仓库Name、Address、Username和Password, 单击Apply, 然后单击OK。

Alibaba Cloud Toolkit > Docker > Registry Reset

Registry

Registry: test Add Remove

Registry Details

Name: test

Address: registry.cn-hangzhou.aliyuncs.com

Username: test

Password: Show

Test Connection

OK Cancel Apply

? 说明 建议单击Test Connection，测试远程仓库是否连接成功。

4. 在Repository对话框填入您的镜像地址。
5. 先单击Apply，然后单击Run。

结果验证

以阿里云容器镜像服务平台为例，可通过查看您的镜像版本更新时间来确认镜像是否推送成功。

版本	镜像ID	状态	Digest	镜像大小	最后更新时间	操作
20200422164720	...	● 正常	...	84.224 MB	2020-04-22 16:47:32	安全扫描 层信息 同步 删除
20200422155306	...	● 正常	...	84.224 MB	2020-04-22 16:42:13	安全扫描 层信息 同步 删除

5.2. 使用Eclipse部署应用到镜像仓库

Alibaba Cloud Toolkit可以将应用快速部署到镜像仓库，提供安全的镜像托管能力。在Eclipse安装和配置Alibaba Cloud Toolkit后，只需在配置界面设置部署参数即可实现自动化部署。

前提条件

- 已安装和配置Docker。
- 已安装和配置Alibaba Cloud Toolkit，请参见[在Eclipse中安装和配置Cloud Toolkit](#)。

镜像仓库配置

使用Cloud Toolkit将应用部署到镜像仓库具体步骤如下：

1. 在Eclipse界面左侧的Project Explorer中右键单击待部署的工程名，在快捷菜单中选择Alibaba Cloud > Deploy to Registry / Kubernetes > Deploy to Registry。
2. 在Select a package method页面选择项目路径。

 说明 Context Directory和Dockerfile通常会根据您的本地应用工程自动识别并设置。

3. 在Select a Repository对话框设置部署参数。

Deploy to CS Kubernetes

Select a Repository

Repositories [Create a new repositories](#)

Alibaba Cloud Container Registry Custom Container Registry

华北 2 (北京) Select a name! Type the name of Namespace Search

Name	Namespace	Status	Type

▼ Image

Version(Tag): User current time for every deloyment

▼ Advanced

Repository address

Internet

VPC Network

Classic Network

部署镜像仓库参数说明：

- Repositories
 - Alibaba Cloud Container Registry: 阿里云镜像仓库，详情请参见[部署应用到ACR](#)。
 - Custom Container Registry: 自建镜像仓库，详情请参见[部署应用到其它镜像仓库](#)。
- Image
 - Version: 镜像版本号。

部署应用到ACR

1. 在部署参数页面选择Alibaba Cloud Container Registry。

Repositories [Create a new repositories](#)

Alibaba Cloud Container Registry Custom Container Registry

华北 2 (北京) Select a name! Type the name of Namespace Search

Name	Namespace	Status	Type

Image

Advanced

Repository address

Internet

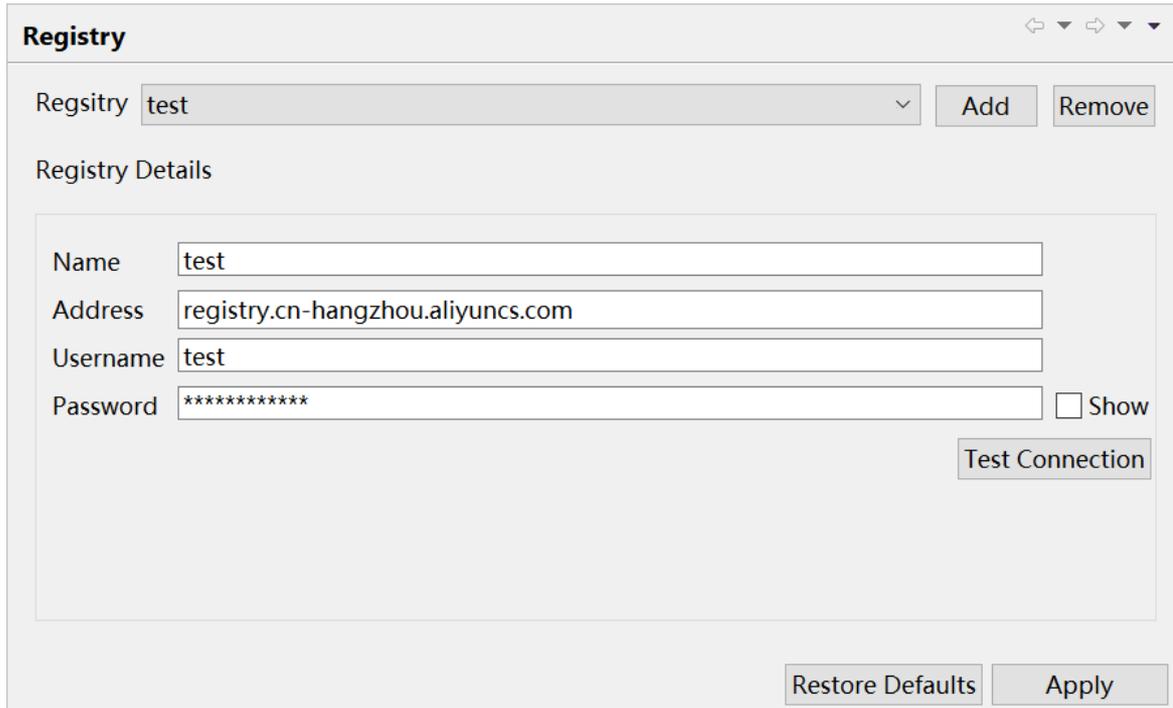
VPC Network

Classic Network

2. 选择地域。
3. 选择命名空间。
4. 选择镜像仓库。
5. 在Advanced下拉选项中选择网络类型。
 - o Internet：公有网络。
 - o VPC Network：VPC网络。
 - o Classic Network：经典网络。
6. 单击Finish。

部署应用到其它镜像仓库

1. 在部署参数页面选择Custom Container Registry。
2. 单击Add，配置Registry信息。
3. 在Registry页面配置镜像仓库Name、Address、Username和Password。



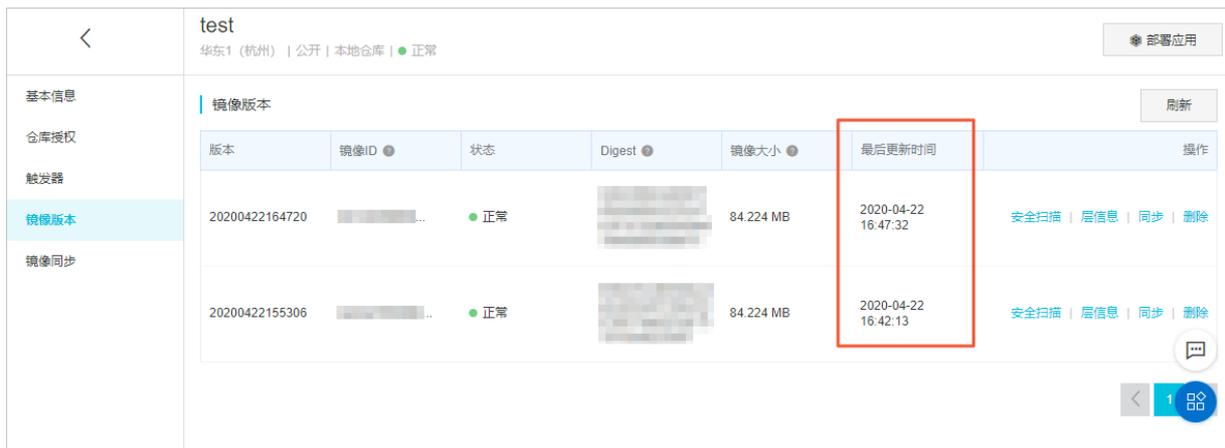
The image shows a 'Registry' configuration dialog box. At the top, there is a dropdown menu for 'Registry' set to 'test', with 'Add' and 'Remove' buttons. Below this is the 'Registry Details' section with input fields for 'Name' (test), 'Address' (registry.cn-hangzhou.aliyuncs.com), 'Username' (test), and 'Password' (masked with asterisks). A 'Show' checkbox is next to the password field. A 'Test Connection' button is located at the bottom right of the details section. At the very bottom of the dialog are 'Restore Defaults' and 'Apply' buttons.

说明 建议单击Test Connection，测试远程仓库是否连接成功。

- 4. 选择Apply > Apply and Close。
- 5. 在Repository对话框中填入您的镜像地址。
- 6. 单击Finish。

结果验证

以阿里云容器镜像服务平台为例，可通过查看您的镜像版本更新时间来确认镜像是否推送成功。



The screenshot shows the 'test' repository page in the console. The page title is 'test' with subtext '华东1 (杭州) | 公开 | 本地仓库 | 正常'. There is a '部署应用' button in the top right. The left sidebar has '镜像版本' selected. The main content area shows a table of image versions. The '最后更新时间' column for the first version is highlighted with a red box.

版本	镜像ID	状态	Digest	镜像大小	最后更新时间	操作
20200422164720	...	正常	...	84.224 MB	2020-04-22 16:47:32	安全扫描 层信息 同步 删除
20200422155306	...	正常	...	84.224 MB	2020-04-22 16:42:13	安全扫描 层信息 同步 删除

6.部署应用到服务器

6.1. 使用IntelliJ IDEA部署应用到Linux服务器

Cloud Toolkit支持将应用部署到采用SSH标准协议的服务器，您无需在一系列运维工具之间切换，只需在图形界面上选择目标服务器即可快速部署。本文为您介绍在IntelliJ IDEA中使用Cloud Toolkit部署本地应用到Linux服务器。

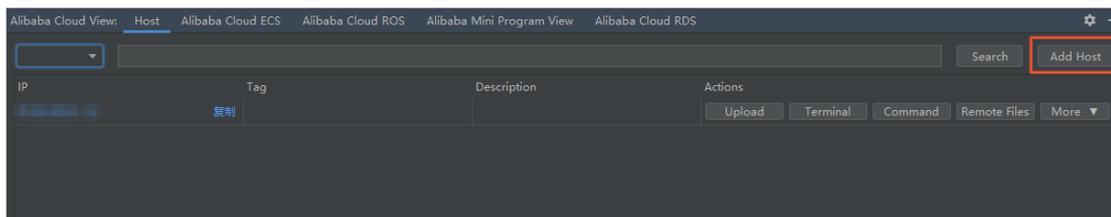
前提条件

- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。
- 已准备好支持标准SSH标准协议的Linux服务器。

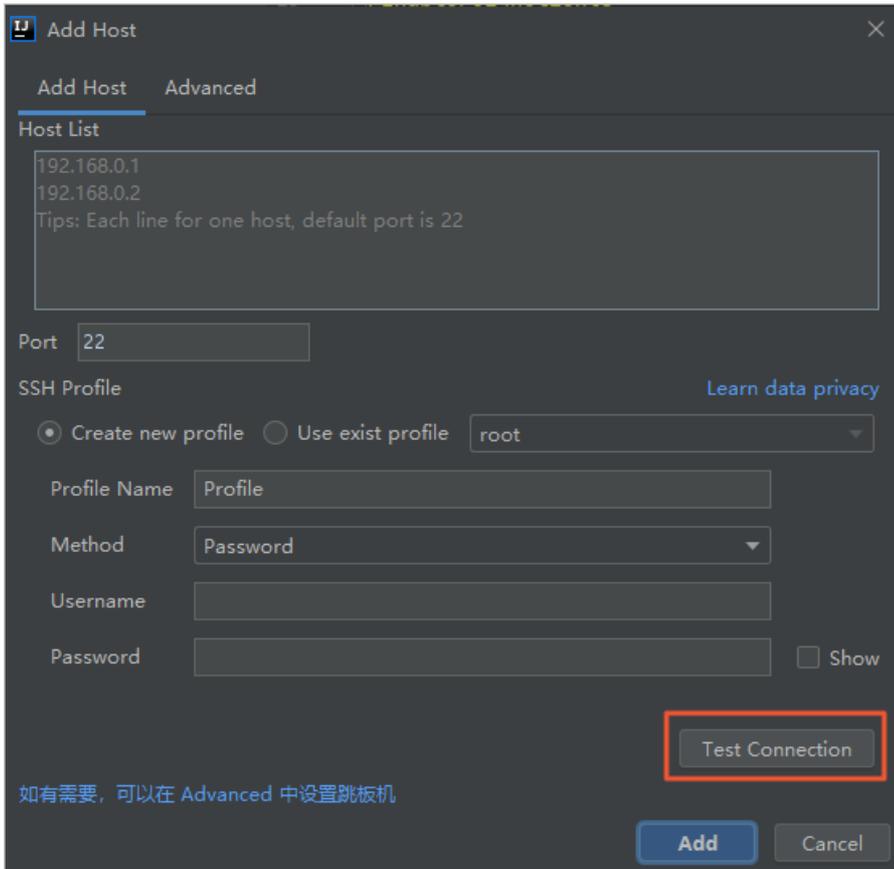
部署应用到Linux服务器

您可以下载[java_demo](#)（Java_demo用于在Web页面上打印 Hello World 字符串）来完成部署工作，具体步骤如下：

1. 在IntelliJ IDEA中导入Java_demo工程。
2. 添加服务器。
 - i. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Alibaba Cloud View > Host。
 - ii. 在弹出的Host页签中单击Add Host。

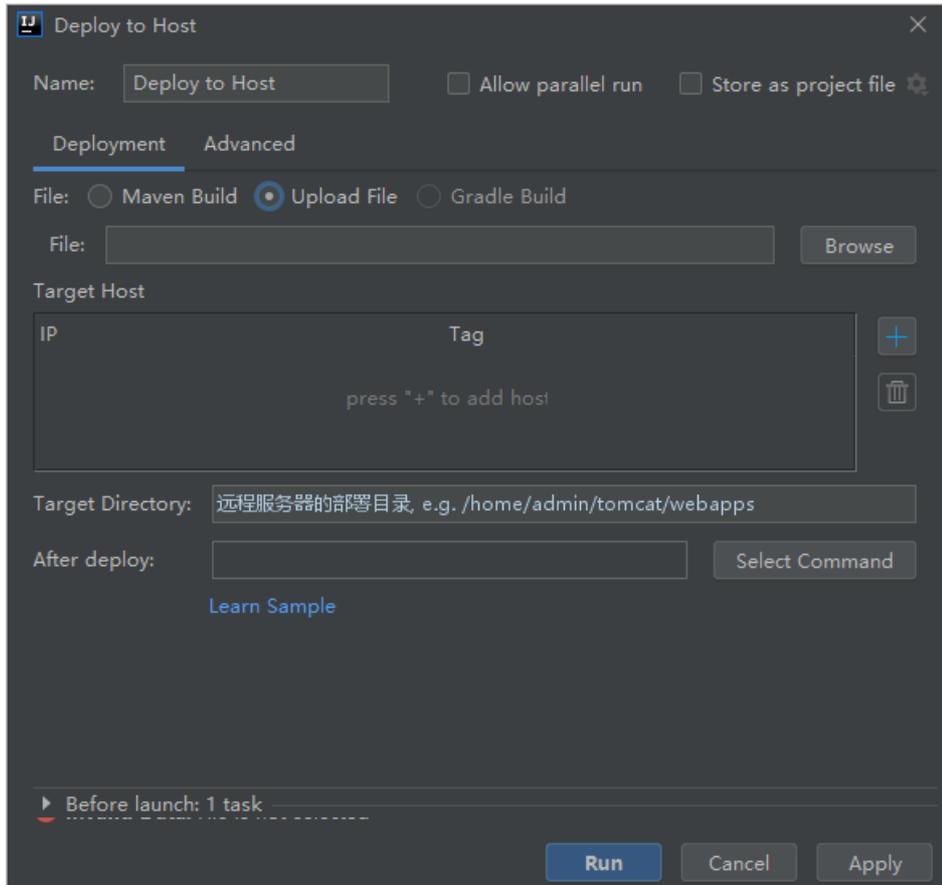


- iii. 在Add Host对话框中设置Host List、Username、Password和Tag等参数，完成后单击Add。



 说明 完成参数设置后，可以单击Test Connection来测试是否能成功连接服务器。

- 3. 部署应用。
 - i. 在IntelliJ IDEA顶部菜单中选择Alibaba Cloud > Deploy to Host...
 - ii. 在Deploy to Host对话框设置部署参数，然后单击Run。



部署参数说明如下表所示：

参数	描述
File	<ul style="list-style-type: none"> ■ Maven Build：若当前工程采用Maven构建，可以直接构建并部署。 ■ Upload File：若当前工程不是采用Maven构建，或在本地已存在打包好的部署文件，可以选择并上传本地的部署文件。 ■ Gradle Build：若当前工程采用Gradle构建，选择此项来构建并部署。
Target Host	在下拉列表中选择Tag，然后在该Tag中选择要部署的服务器。
Target Directory	输入在服务器上的部署路径，如 <code>/root/tomcat/webapps</code> 。
Command	输入应用启动命令，如 <code>sh /root/restart.sh</code> 。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志，您可以根据日志信息检查部署结果。

您还可以通过内置Terminal终端登录远程服务器，`cd` 至部署径下，然后执行 `ls` 命令检查是否部署成功。登录远程服务器操作步骤如下：

1. 在IntelliJ IDEA编辑器底部单击Alibaba Cloud View页签，并选择Host页签。
2. 在Host页签中单击目标远程服务器Actions区域的Terminal，登录远程服务器。

了解更多关于如何查看应用部署的日志详情请参见[使用Cloud Toolkit查看远程服务器按日滚动的日志文件](#)。

6.2. 使用IntelliJ IDEA部署应用到Windows服务器

Cloud Toolkit支持将应用部署到采用SSH标准协议的服务器，您无需在一系列运维工具之间切换，只需在图形界面上选择目标服务器即可快速部署。本文为您介绍在IntelliJ IDEA中使用Cloud Toolkit部署本地应用到Windows服务器，您可选择OpenSSH Server或freeSSHd进行部署配置。

前提条件

已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

准备工作

需配置支持SSH标准协议的Windows服务器，具体操作步骤如下：

 说明 您可任意选择OpenSSH Server或freeSSHd进行安装。

- 安装OpenSSH Server。如果您的远程服务器操作系统为Windows 10，可能已自带OpenSSH服务端，可跳过此步骤，检查方法请参见[官方文档](#)。
在Windows服务器中安装PowerShell，详情请参见[官方教程](#)。若您的服务器中已安装PowerShell，可跳过此步骤。
 - 在线安装请参见[官方教程](#)。
 - 离线安装的具体操作步骤如下：
 - a. 下载[OpenSSH安装包](#)。

 说明 64位操作系统和32位操作系统的服务器对应的安装包不同，请下载与您的服务器操作系统对应的安装包。

- b. 解压安装包至 `C:\Program Files\OpenSSH-Win64` 路径下。
- c. 在 `C:\Program Files\OpenSSH-Win64` 路径下运行cmd，然后执行以下命令安装OpenSSH。

```
powershell.exe -ExecutionPolicy Bypass -File install-sshd.ps1
```

- d. 执行以下命令将OpenSSH设置为自动启动模式，并启动OpenSSH。

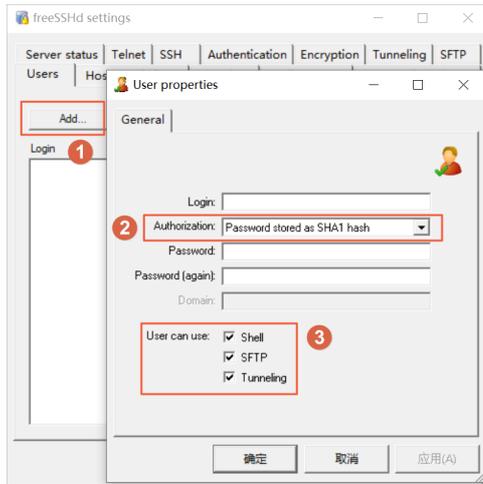
```
sc config sshd start= auto  
net start sshd
```

- e. 在PowerShell中执行以下命令修改默认Shell为PowerShell。

```
New-ItemProperty -Path "HKLM:\SOFTWARE\OpenSSH" -Name DefaultShell -Value "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -PropertyType String -Force
```

- 安装freeSSHd。具体操作步骤如下：
 - i. 下载[freeSSHd安装包](#)。
 - ii. 在C盘创建文件夹，重命名为ActTemp。

- iii. 打开freeSSHd，单击SFTP页签，在SFTP home path路径中选择C:\ActTemp。
- iv. 在Users页签下单击Add。在User properties对话框中配置账户信息。



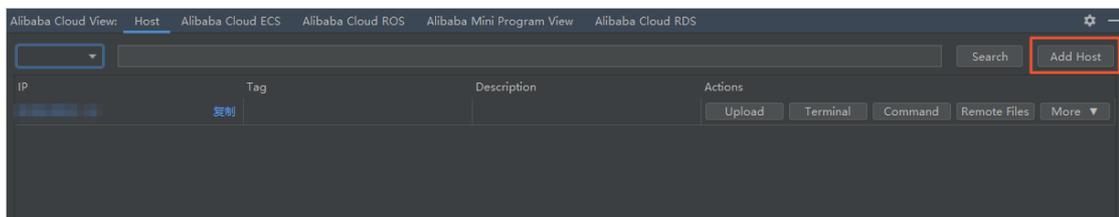
通用参数配置如下：

- **Login**：设置Login name。
 - **Authorization**：选择Password stored as SHA1 hash。
 - **Password**：设置密码。
 - **Password (again)**：再次确认设置密码。
 - **User can use**：勾选shell、SFTP和Tunneling。
- v. 配置默认shell。单击SSH页签，在Command shell路径中选择C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe。
 - vi. 启动SSH服务。单击Server status页签，启动Telnet server is running和SSH server is running。

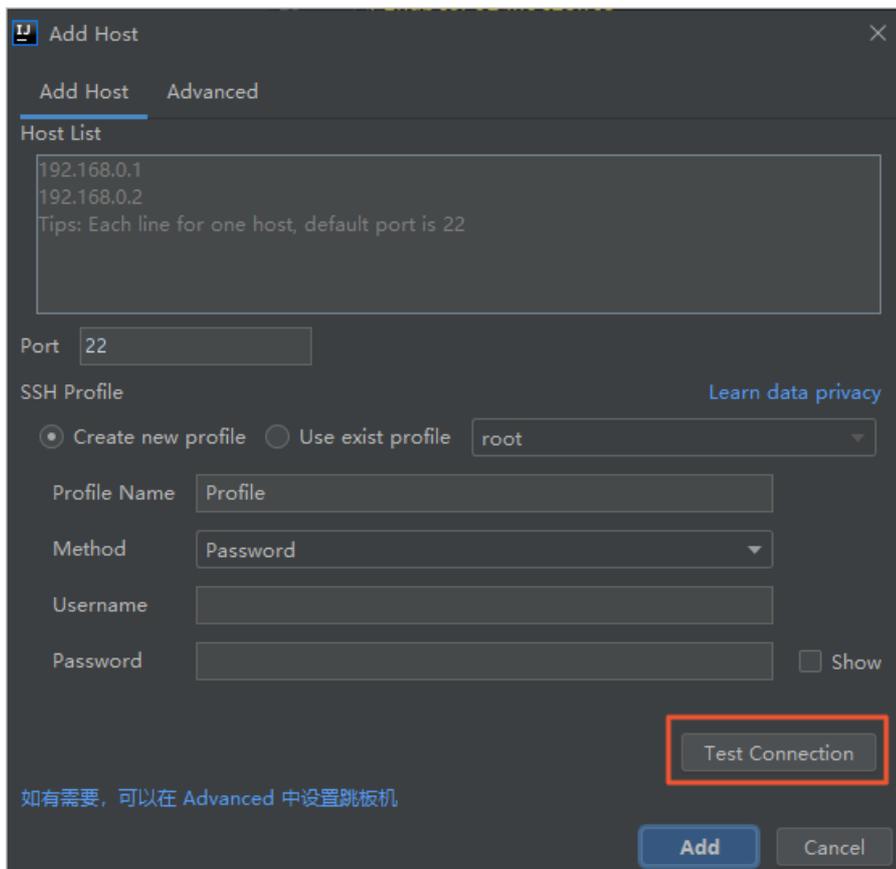
部署应用到Windows服务器

您可以下载java_demo（Java_demo用于在Web页面上打印 Hello World 字符串）来完成部署工作，具体步骤如下：

1. 在IntelliJ IDEA中导入Java_demo工程。
2. 添加服务器。
 - i. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Alibaba Cloud View > Host。
 - ii. 在弹出的Host页签中单击Add Host。

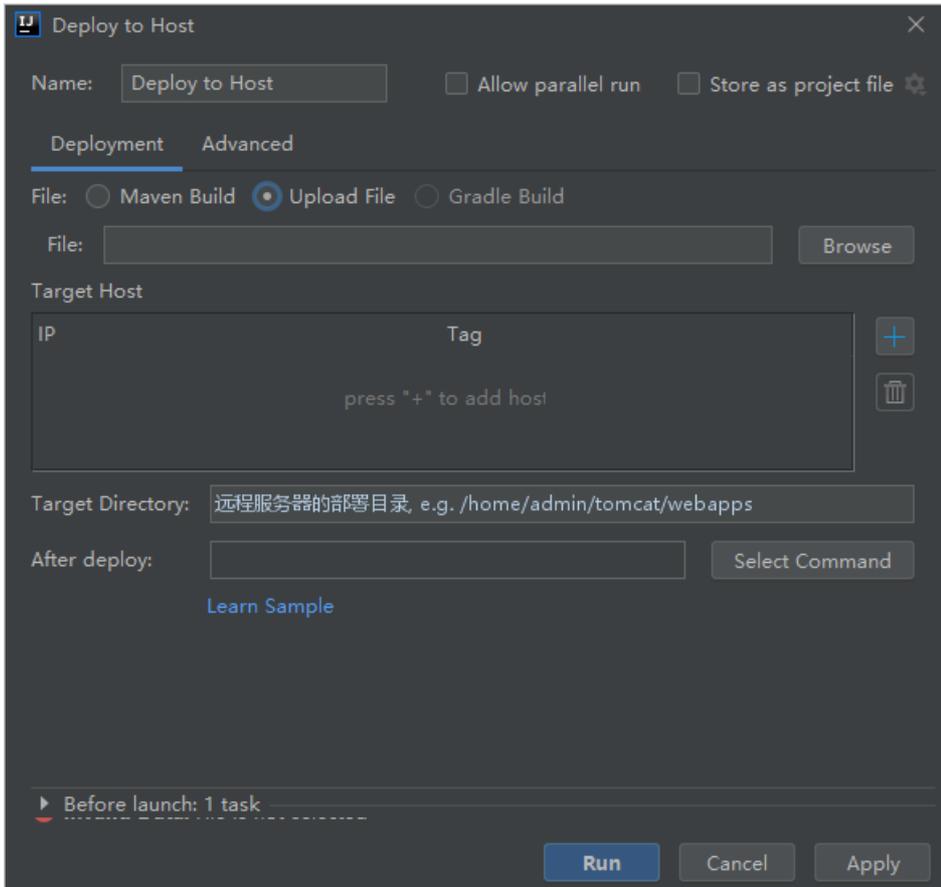


- iii. 在Add Host对话框中设置Host List、Username、Password和Tag等参数，完成后单击Add。



 说明 完成参数设置后，可以单击Test Connection来测试是否能成功连接服务器。

3. 部署应用。
 - i. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to Host...
 - ii. 在Deploy to Host对话框设置部署参数，然后单击Run。



部署参数说明如下。

参数	描述
File	<ul style="list-style-type: none"> ■ Maven Build：若当前工程采用Maven构建，可以直接构建并部署。 ■ Upload File：若当前工程不是采用Maven构建，或在本地已存在打包好的部署文件，可以选择并上传本地的部署文件。 ■ Gradle Build：若当前工程采用Gradle构建，选择此项来构建并部署。
Target Host	在下拉列表中选择Tag，然后在该Tag中选择要部署的服务器。
Target Directory	输入在服务器上的部署路径，如 <code>c:\server\tomcat\deploy</code> 。
Command	输入应用启动命令，如 <code>c:\server\tomcat\deploy\start.bat</code> 。

说明 Command命令默认支持所有PowerShell命令，暂不能直接使用DOS命令。若需使用DOS命令，请用批处理脚本（.bat/.cmd）来执行。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志，您可以根据日志信息检查部署结果。

您还可以通过内置Terminal终端登录远程服务器，执行 `cd` 命令至部署的路径下，然后执行 `ls` 命令检查是否部署成功。登录远程服务器操作步骤如下：

1. 在IntelliJ IDEA编辑器底部单击Alibaba Cloud View页签，并选择Host View页签。
2. 在Host View页签中单击目标远程服务器Actions区域的Terminal，登录远程服务器。

6.3. 使用Visual Studio Code部署应用到远程服务器

Cloud Toolkit支持将应用部署到采用SSH标准协议的服务器，您无需在一系列运维工具之间切换，只需在图形界面上选择目标服务器即可快速部署。

前提条件

- 已安装和配置Cloud Toolkit，请参见在[Visual Studio Code中安装和配置Cloud Toolkit](#)。
- 已准备好支持标准SSH标准协议的服务器。

部署应用到远程服务器

您可以下载java_demo（Java_demo用于在Web页面上打印 Hello World 字符串）来完成部署工作，具体步骤如下：

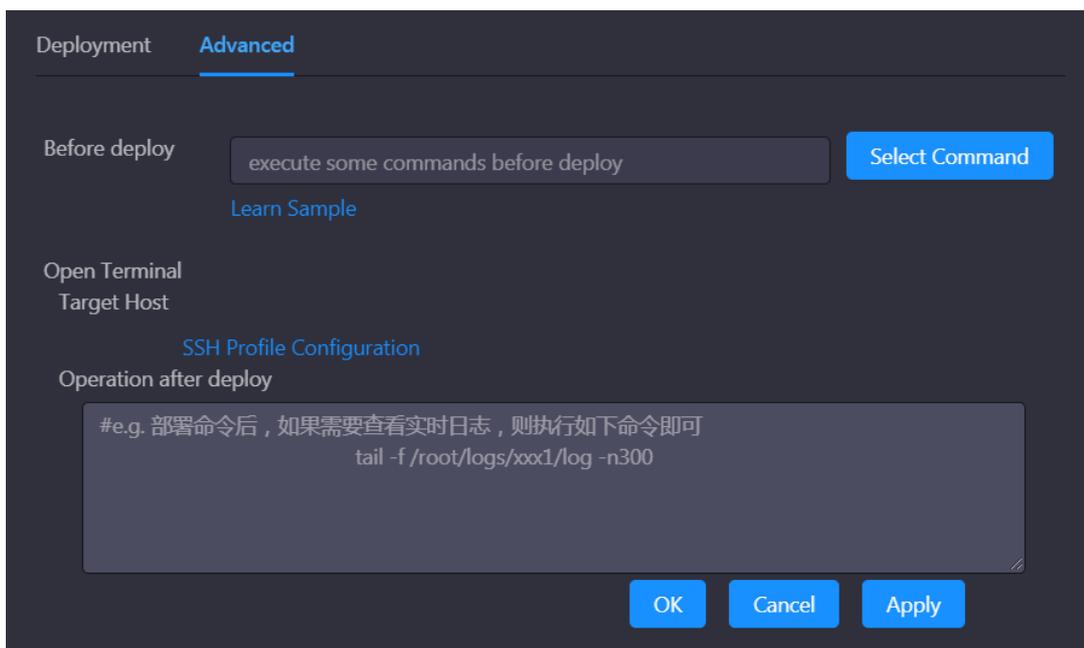
1. 在Visual Studio Code中导入Java_demo工程。
2. 添加服务器。
 - i. 在左侧菜单栏中单击阿里云图标，然后在ALIBABA CLOUD TOOLKIT侧边栏中选择ALIBABA CLOUD VIEW > Host View。
 - ii. 在弹出的Host页签中单击Add Host。
 - iii. 在Add Host对话框中设置Host List、Port和SSH Profile后，单击Test Connection测试是否能成功连接服务器。
3. 部署应用。
 - i. 单击ALIBABA CLOUD TOOLKIT侧边栏中RUN CONFIGURATIONS旁边的...图标，选择Deploy to Host。
 - ii. 在Deploy HOST View页签中设置部署参数，然后单击Apply。
 - Deployment（必选）：

部署参数说明如下表所示：

参数	描述
Name	为此次配置定义一个名字，以便区分多个配置。
File	在以下打包方式中选择： npm build 、 Webpack build 或 Upload file 。
Project	选择待部署工程的根目录。打包方式选择 npm build 和 Webpack build 时需要配置。
Build Output	打包之后的Output目录。打包方式选择 npm build 和 Webpack build 时需要配置。
Scripts	选择脚本命令，仅当打包方式选择 npm build 时需配置。
Webpack	Webpack配置，选择一个Webpack脚本，仅当打包方式选择 Webpack build 时需配置。
File	选择要上传文件的文件夹，仅当打包方式选择 Upload file 时需设置。

参数	描述
Target Host	此次部署的远程目标服务器。
Target Directory	远程目标服务器上的指定部署目录。
After deploy	输入应用启动命令，表示在完成应用包的部署后，需要执行的命令。 以Java程序为例，通常是一句Tomcat的启动命令，如 <code>sh /root/restart.sh</code> 。

- **Advanced**（可选）：在设置页面完成启动命令和部署后命令的设置，设置完成后单击**Apply**。



结果验证

部署开始后，Visual Studio Code的Console区域会打印部署日志，您可以根据日志信息检查部署结果。

您还可以通过内置Terminal终端登录远程服务器，`cd` 至部署径下，然后执行 `ls` 命令检查是否部署成功。登录远程服务器操作步骤如下：

1. 在左侧菜单栏中单击阿里云图标，然后在ALIBABA CLOUD TOOLKIT侧边栏中选择ALIBABA CLOUD VIEW > Host View。
2. 在Host View页签中单击目标远程服务器Actions列的Terminal，登录远程服务器。

7.部署应用到Kubernetes

7.1. 部署应用到容器服务Kubernetes

7.1.1. 使用IntelliJ IDEA部署应用到容器服务Kubernetes

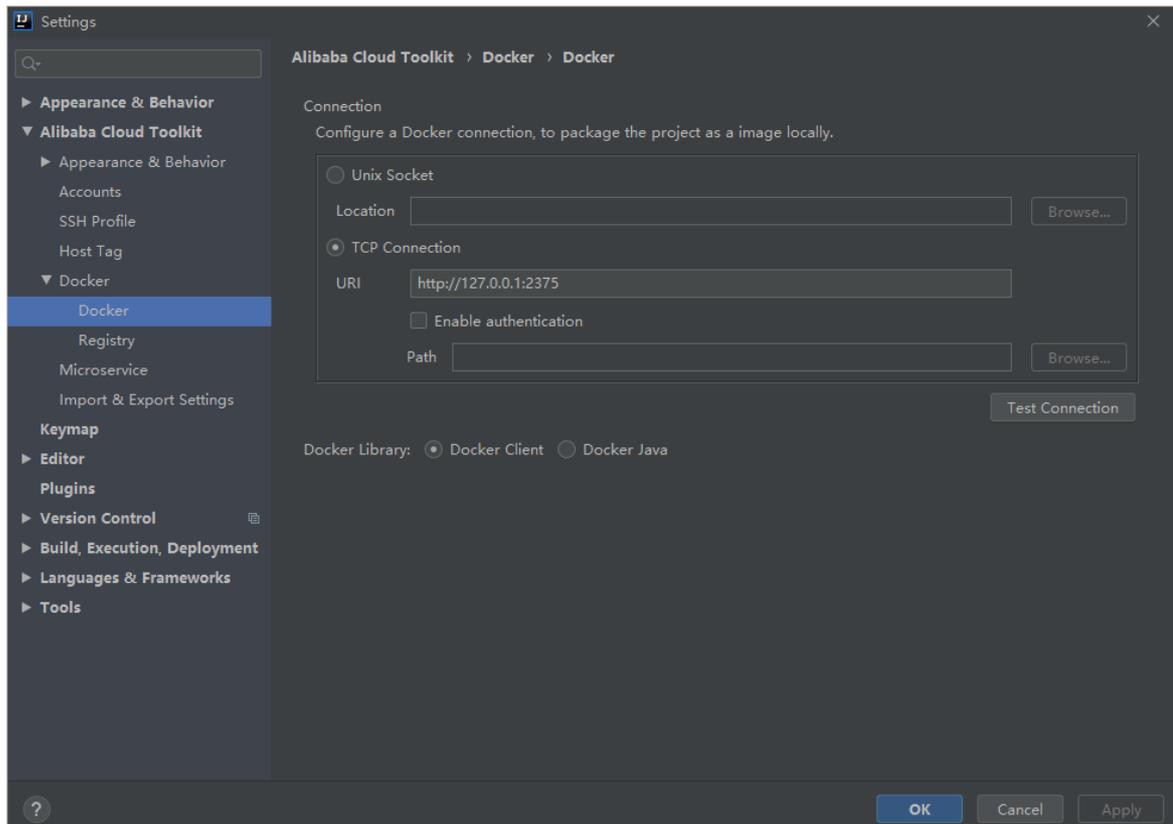
若您已使用阿里云容器服务Kubernetes，可以使用Cloud Toolkit来快速部署应用。本文将介绍如何在IntelliJ IDEA中使用Cloud Toolkit部署应用到容器服务Kubernetes。

前提条件

- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。
- 已在容器镜像服务控制台创建容器镜像仓库，并使用镜像创建应用，请参见[创建容器镜像仓库](#)和[使用镜像创建应用](#)。

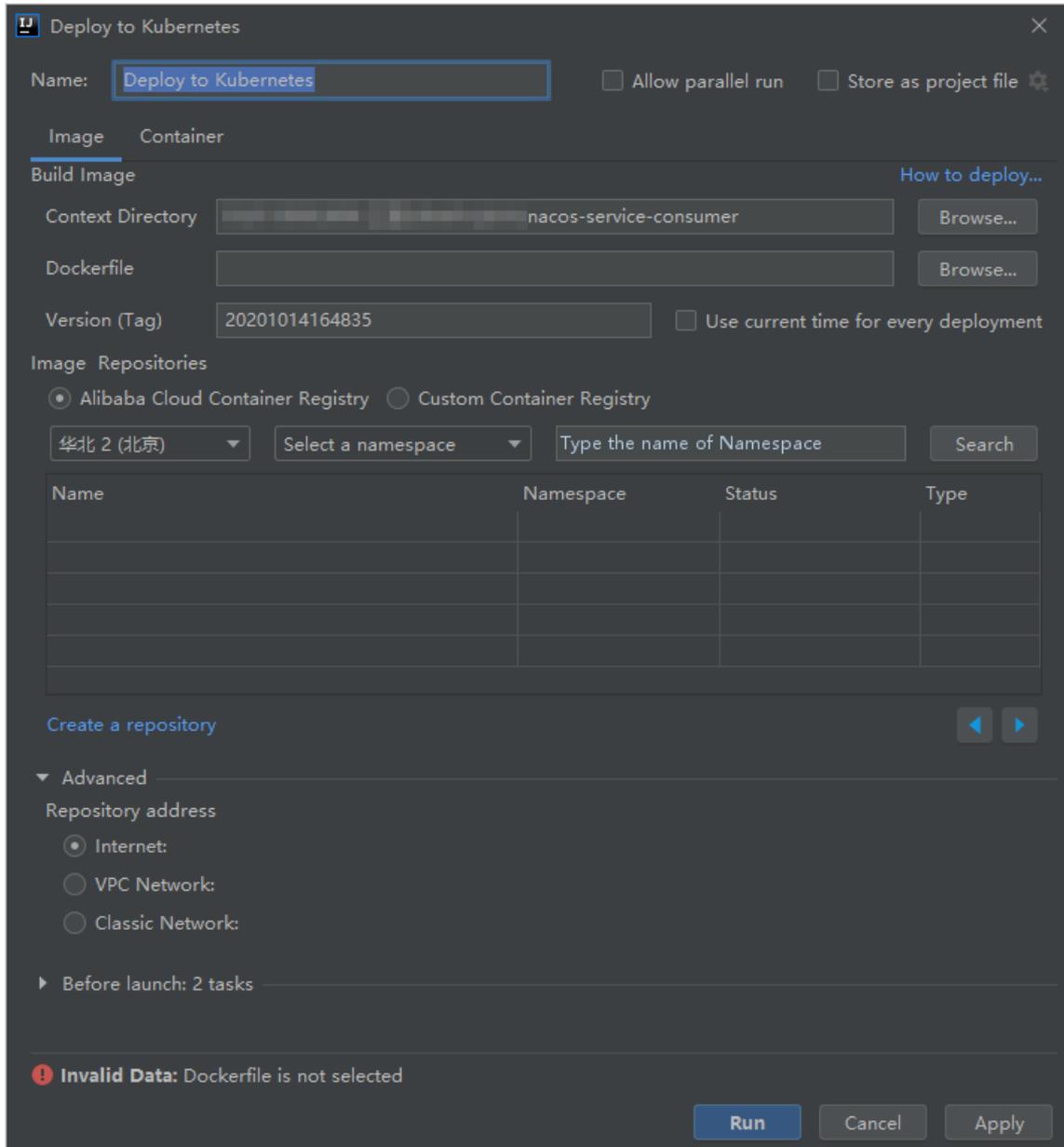
设置Docker环境

1. 在IntelliJ IDEA顶部菜单栏单击Tools > Alibaba Cloud > Preferences...。
2. 在Settings对话框的左侧导航栏中选择Alibaba Cloud Toolkit > Docker。
3. 在Docker界面中设置可连接的Docker环境，包括本地和远程两种方式，然后单击OK。
 - 本地Docker环境：
 - 如果您本地为Mac或Linux操作系统，勾选Unix Socket，然后在Location输入框中输入 `unix:///var/run/docker.sock`。
 - 如果您本地为Windows操作系统，勾选TCP Connection，然后在URI右侧文档框输入本地Docker的URI，如 `http://127.0.0.1:2375`。
 - 远程Docker环境：勾选TCP Connection，在URI右侧的文本框里输入远端的Docker环境的URI（包括IP地址和端口），如 `http://x.x.x.x:2375`，并确保远程主机的HTTP服务开启。



部署应用到容器服务Kubernetes

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to Registry/Kubernetes > Deploy to Kubernetes。
3. 在Deploy to Kubernetes对话框中进行以下配置。
 - 在Image页签中配置Build Image和Image Repositories参数。



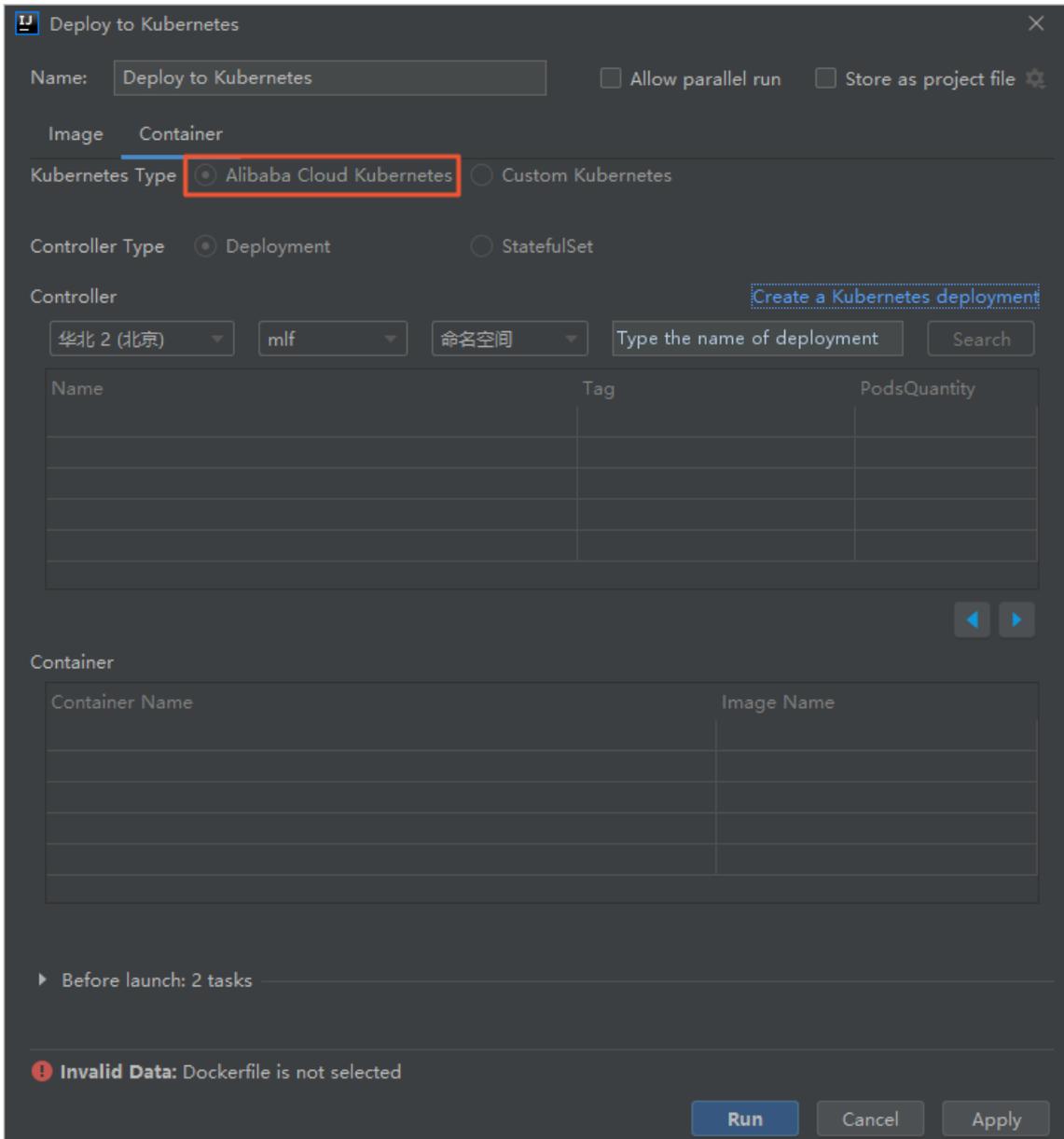
Image页签中的部署参数说明如下。

参数	描述
Name	部署名，建议以英文命名。
Build Image	<ul style="list-style-type: none"> ▪ Context Directory: 上下文目录。 ▪ Dockerfile: Dockerfile文件。 ▪ Version: 镜像版本号。

参数	描述
Image Repositories	选择容器镜像服务的地域、命名空间和镜像仓库。 <ul style="list-style-type: none">Alibaba Cloud Container Registry: 阿里云镜像仓库, 详情请参见部署应用到ACR。Custom Container Registry: 自建镜像仓库, 详情请参见部署应用到其它镜像仓库。

 说明 Context Directory和Dockerfile通常会根据您本地的应用工程自动识别并设置。

- 在Container页签中配置Kubernetes Type、Controller Type、Controller 和Container参数。



Container页签中的部署参数说明如下。

参数	描述
Kubernetes Type	选择Alibaba Cloud Kubernetes，即部署到阿里云容器服务Kubernetes。
Controller Type	<ul style="list-style-type: none"> ▪ Deployment：无状态应用。 ▪ StatefulSet：有状态应用。
Controller	选择区域、命名空间和对应的应用。
Container	选择对应的容器。

4. 先单击Apply，然后单击Run。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

如果您在使用Cloud Toolkit过程中有任何疑问，欢迎您[联系我们](#)。

7.1.2. 使用Eclipse部署应用到容器服务Kubernetes

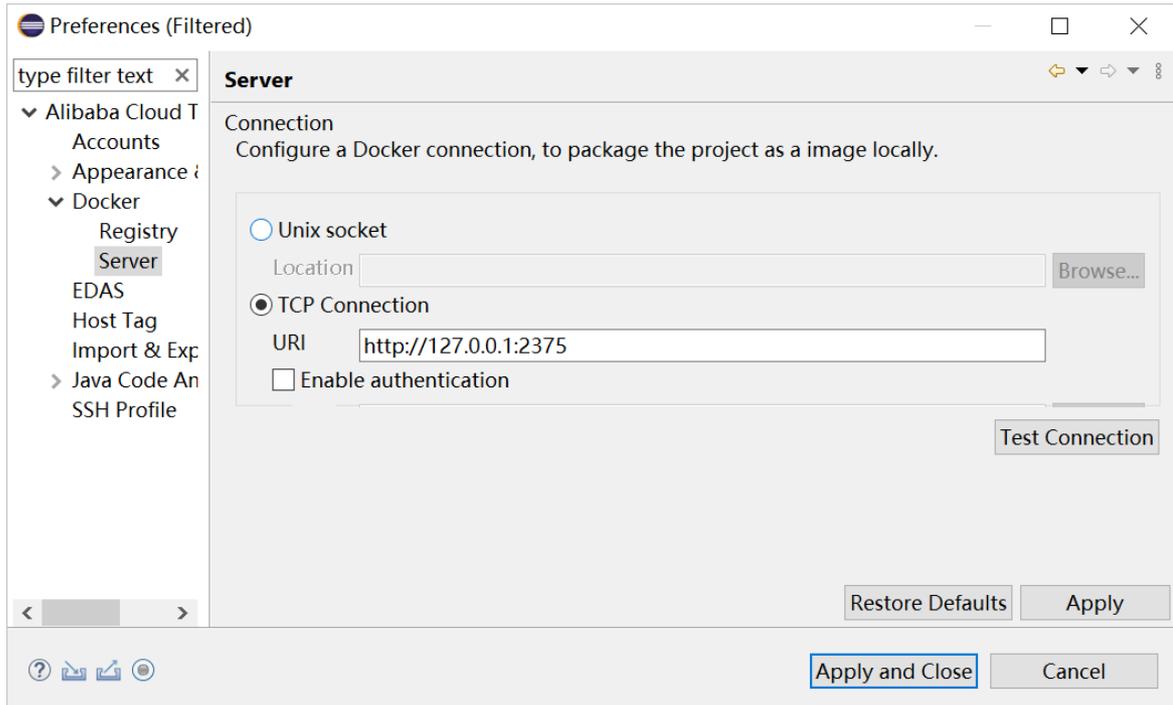
若您已使用阿里云容器服务Kubernetes，可以使用Cloud Toolkit来快速部署应用。本文将介绍如何在Eclipse中使用Cloud Toolkit将应用部署到容器服务Kubernetes。

前提条件

- 您已在Eclipse中安装和配置Cloud Toolkit，请参见在[Eclipse 中安装和配置 Cloud Toolkit](#)。
- 您已在容器镜像服务控制台创建容器镜像仓库，并使用镜像创建应用，请参见[创建容器镜像仓库和使用镜像创建应用](#)。

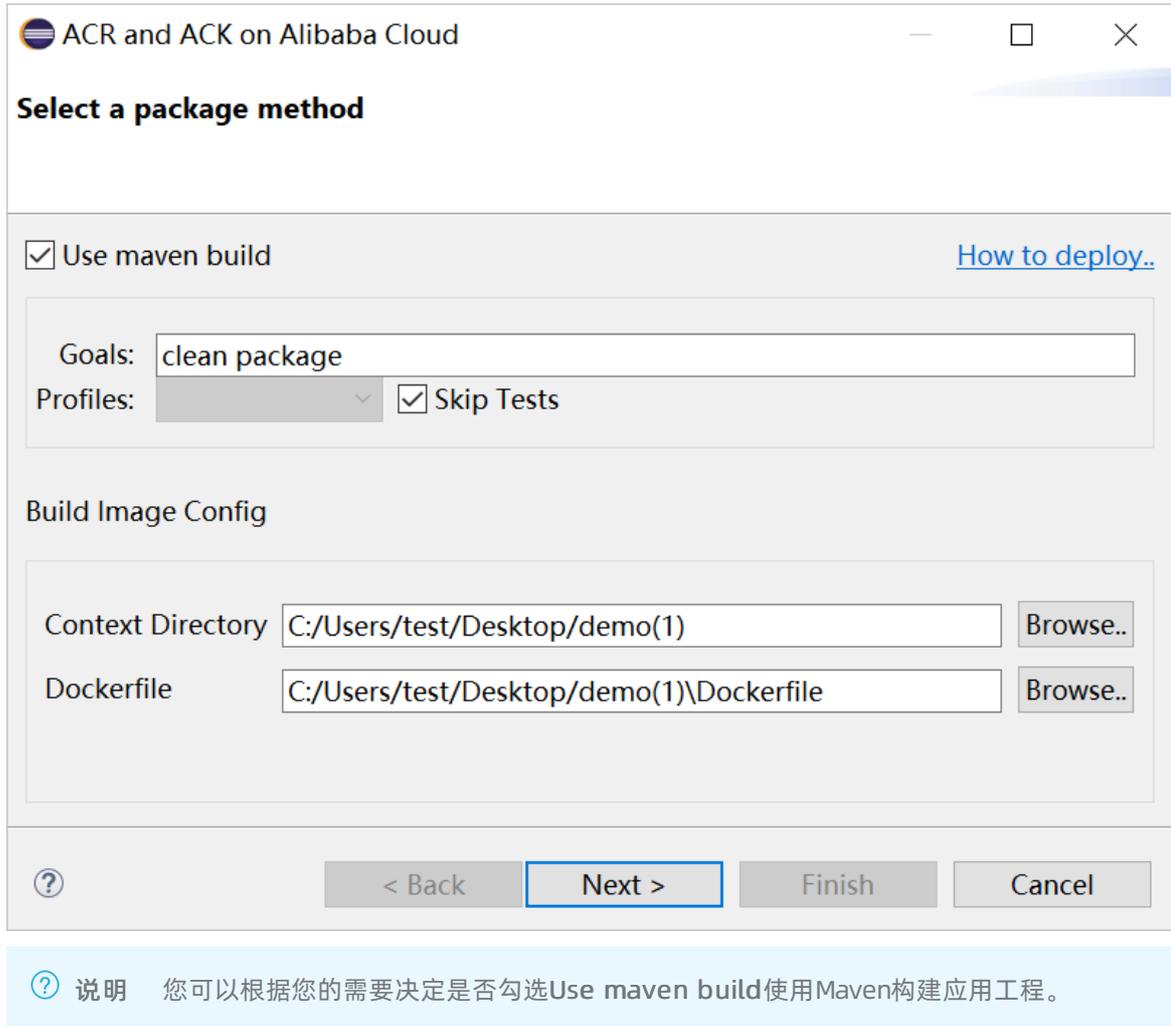
设置Docker环境

1. 在Eclipse工具栏单击Cloud Toolkit图标，在下拉菜单中单击Preference。
2. 在Preference（Filtered）对话框的左侧导航栏中选择Docker > Server。
3. 在Server界面中设置可连接的Docker环境，然后单击Apply。
 - 本地Docker环境：
 - 如果您本地为Mac或Linux操作系统，勾选Unix Socket，然后单击Browse，选择本地的Docker安装目录。
 - 如果您本地为Windows操作系统，勾选Tcp Connection，然后在URI右侧文档框输入本地Docker的URI，如`http://127.0.0.1:2375`。
 - 远程Docker环境：勾选TCP Connection，在URI右侧的文本框里输入远端的Docker环境的URI（包括IP地址和端口），如`http://x.x.x.x:2375`，并确保远程主机的HTTP服务开启。

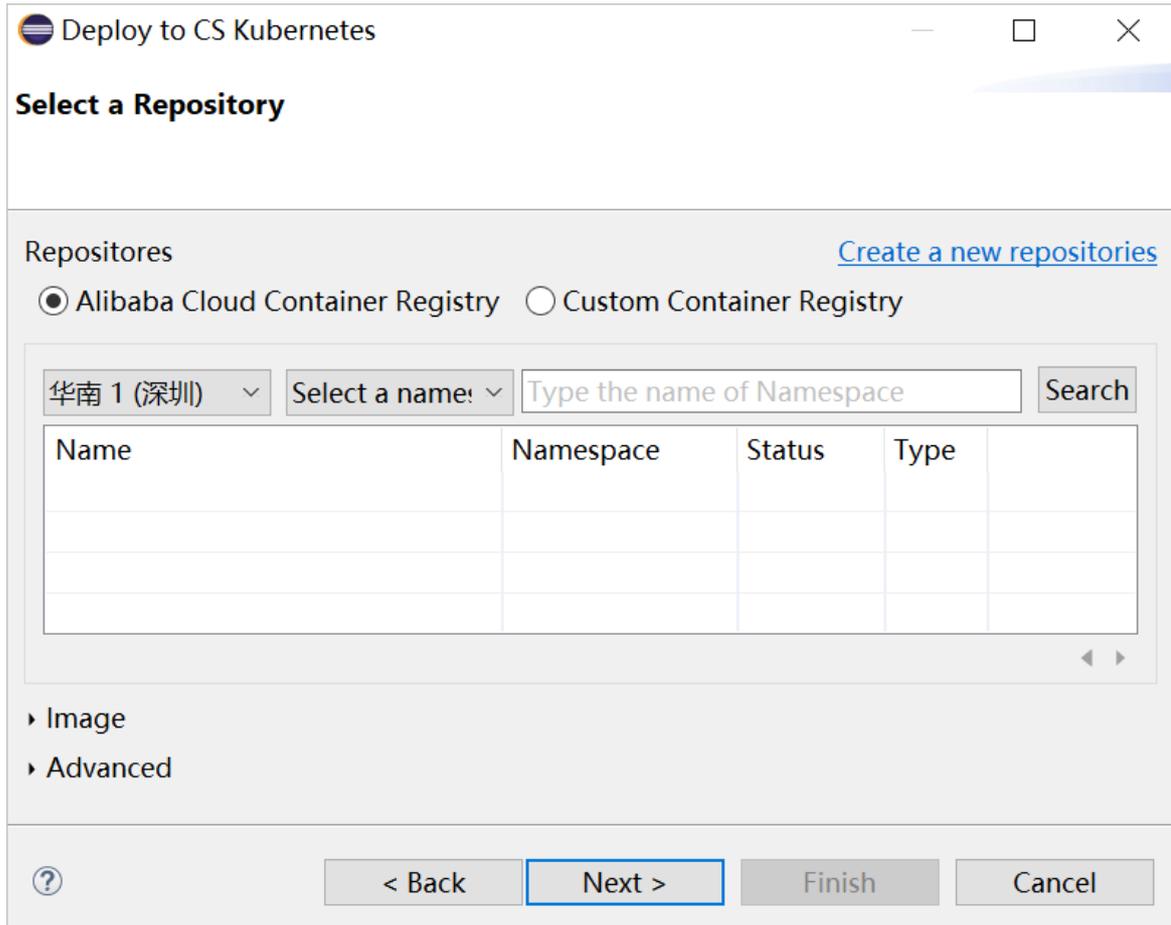


部署应用到容器服务Kubernetes

1. 在Eclipse界面左侧的Package Explorer中右键单击您的Docker应用工程名，在菜单中选择 **Alibaba Cloud > Deploy to Registry/Kubernetes > Deploy to Kubernetes**。
2. 在Deploy to CS Kubernetes对话框Select a package method，选择本地应用程序的Context Directory和Dockerfile（通常会根据您本地的应用工程自动识别并设置），然后单击 **Next**。

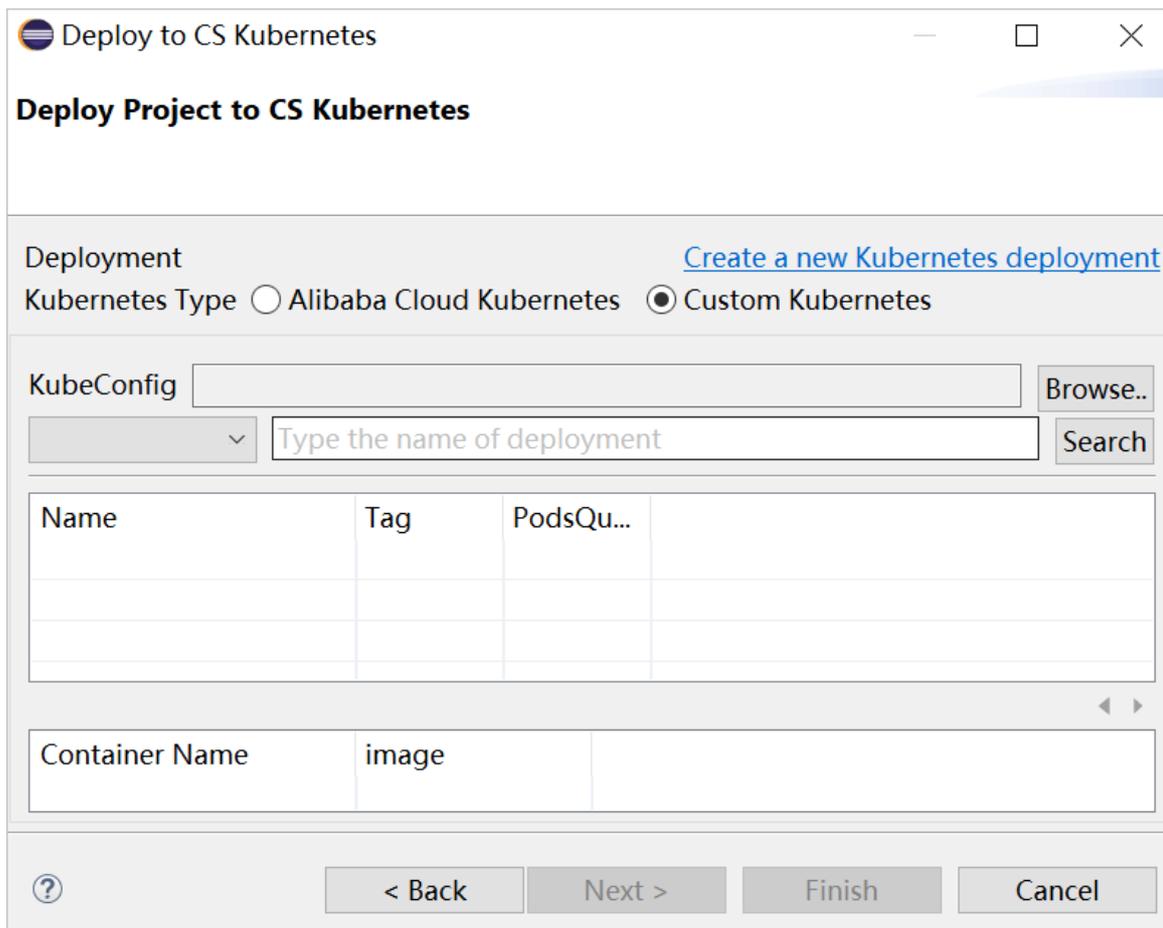


3. 在Deploy to CS Kubernetes对话框Select a Repository，选择容器镜像服务的类型、地域、命名空间和镜像仓库，然后单击 Next。关于镜像仓库部署详情请参见使用Eclipse部署应用到镜像仓库。



说明 如果您还没有镜像仓库，在对话框右上角单击Create a new repositories跳转到容器镜像仓库创建镜像仓库。创建步骤请参见[容器镜像仓库](#)。

4. 在Deploy to CS Kubernetes对话框Deploy Project to CS Kubernetes部署容器服务相关参数，然后单击Finish。



容器部署参数说明如下表所示：

参数	描述
Deployment Kubernetes Type	选择Alibaba Cloud Kubernetes，即部署到阿里云容器服务Kubernetes。
地域	选择地域。
集群	选择集群。
命名空间	选择命名空间。

说明 如果您还没有容器服务，在对话框右上角单击Create a new Kubernetes deployment 跳转到容器服务控制台创建容器服务。创建步骤请参见[首次使用容器服务Kubernetes版](#)。

结果验证

部署开始后，Eclipse的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

如果您在使用Cloud Toolkit过程中有任何疑问，欢迎您[联系我们](#)。

7.2. 部署应用到自建Kubernetes

7.2.1. 使用IntelliJ IDEA部署应用到自建Kubernetes

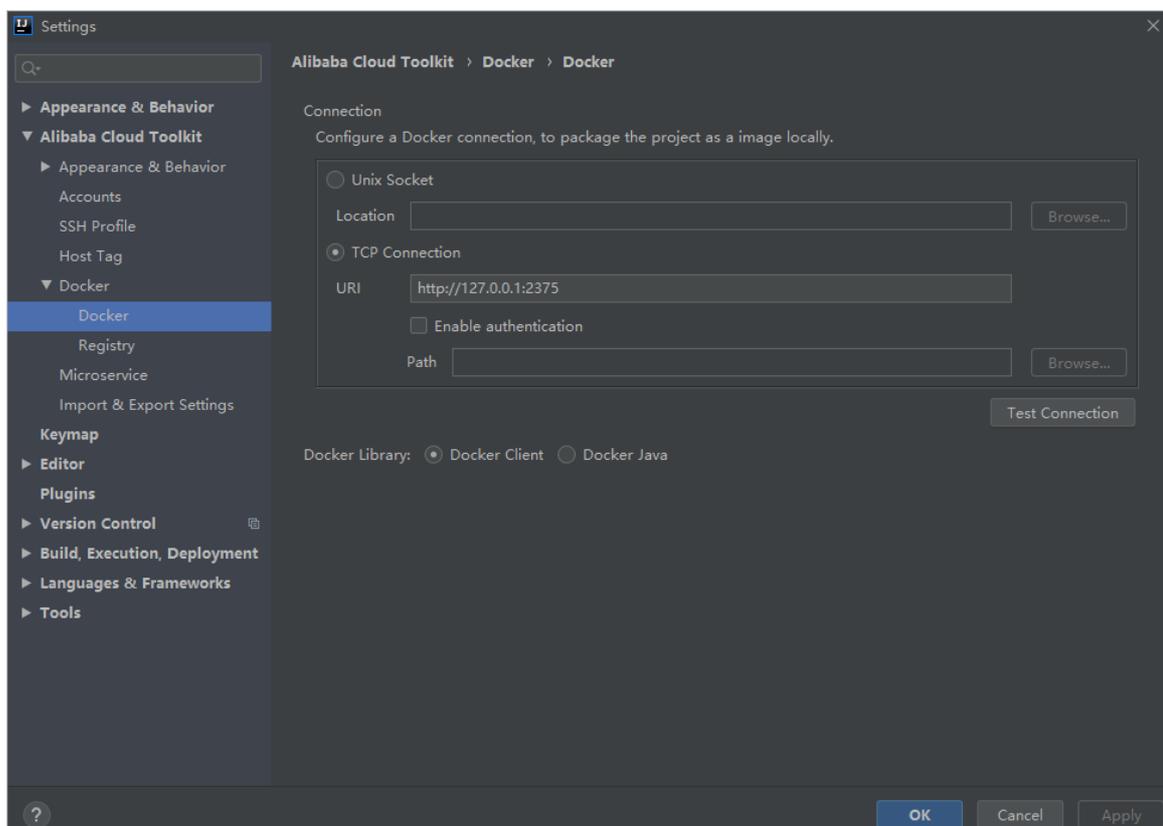
若您已使用自建容器服务Kubernetes，可以使用Cloud Toolkit来快速部署应用。本文将介绍如何在IntelliJ IDEA中使用Cloud Toolkit部署应用到容器服务Kubernetes。

前提条件

- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。
- 已在容器镜像服务控制台创建容器镜像仓库，并使用镜像创建应用，请参见[创建容器镜像仓库](#)和[使用镜像创建应用](#)。

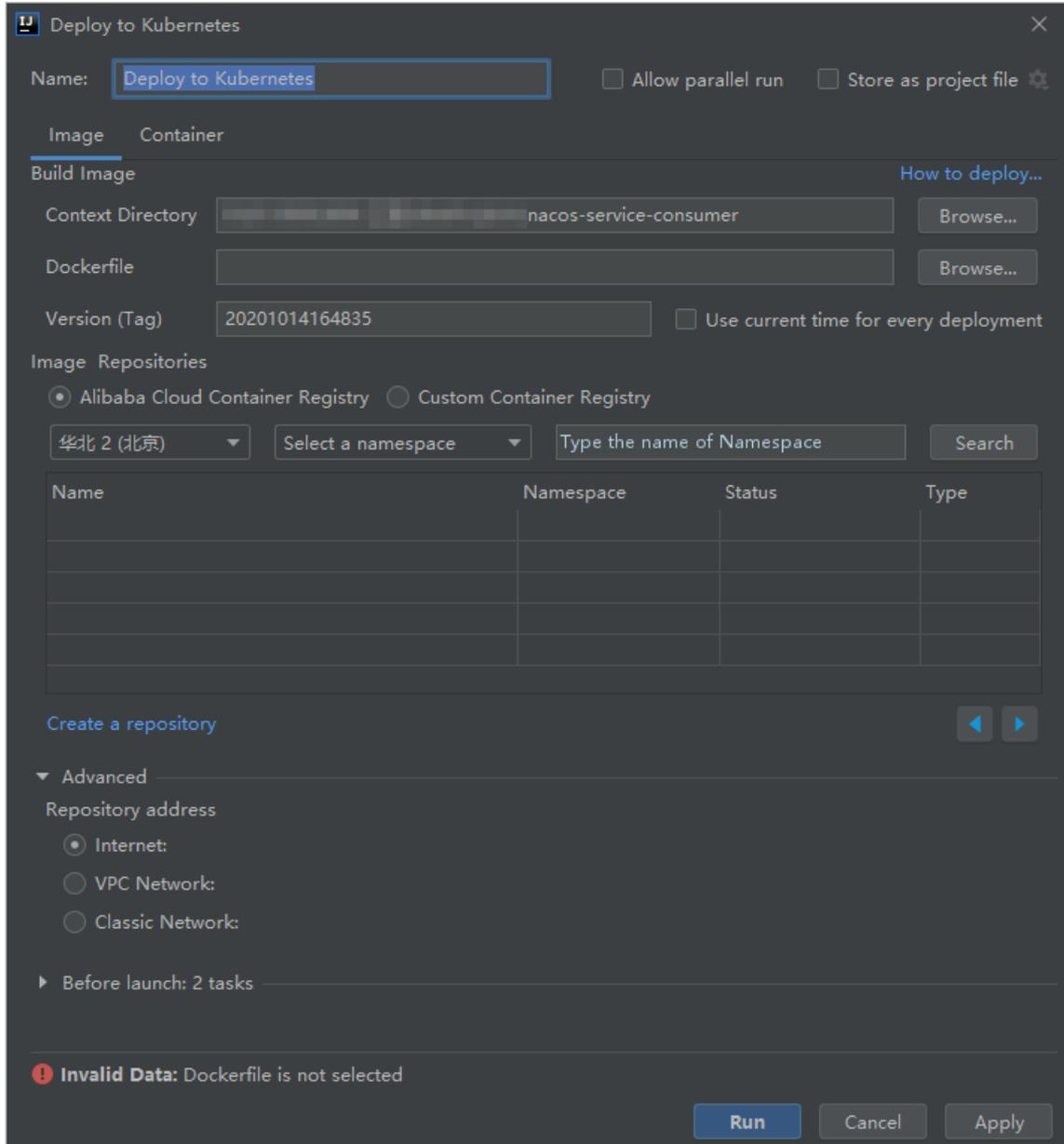
设置Docker环境

1. 在IntelliJ IDEA工具栏单击Tools > Alibaba Cloud > Preferences...
2. 在Settings对话框的左侧导航栏中选择Alibaba Cloud Toolkit > Docker。
3. 在Docker界面中设置可连接的Docker环境，包括本地和远程两种方式，然后单击OK。
 - 本地Docker环境：
 - 如果您本地为Mac或Linux操作系统，勾选Unix Socket，然后在Location输入框中输入 `unix:///var/run/docker.sock`。
 - 如果您本地为Windows操作系统，勾选TCP Connection，然后在URI右侧文档框输入本地Docker的URI，如 `http://127.0.0.1:2375`。
 - 远程Docker环境：勾选TCP Connection，在URI右侧的文本框里输入远端的Docker环境的URI（包括IP地址和端口），如 `http://x.x.x.x:2375`，并确保远程主机的HTTP服务开启。



部署应用到自建Kubernetes

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to Registry/Kubernetes > Deploy to Kubernetes。
3. 在Deploy to Kubernetes对话框中进行以下配置。
 - o 在Image页签中配置Build Image和Image Repositories参数。



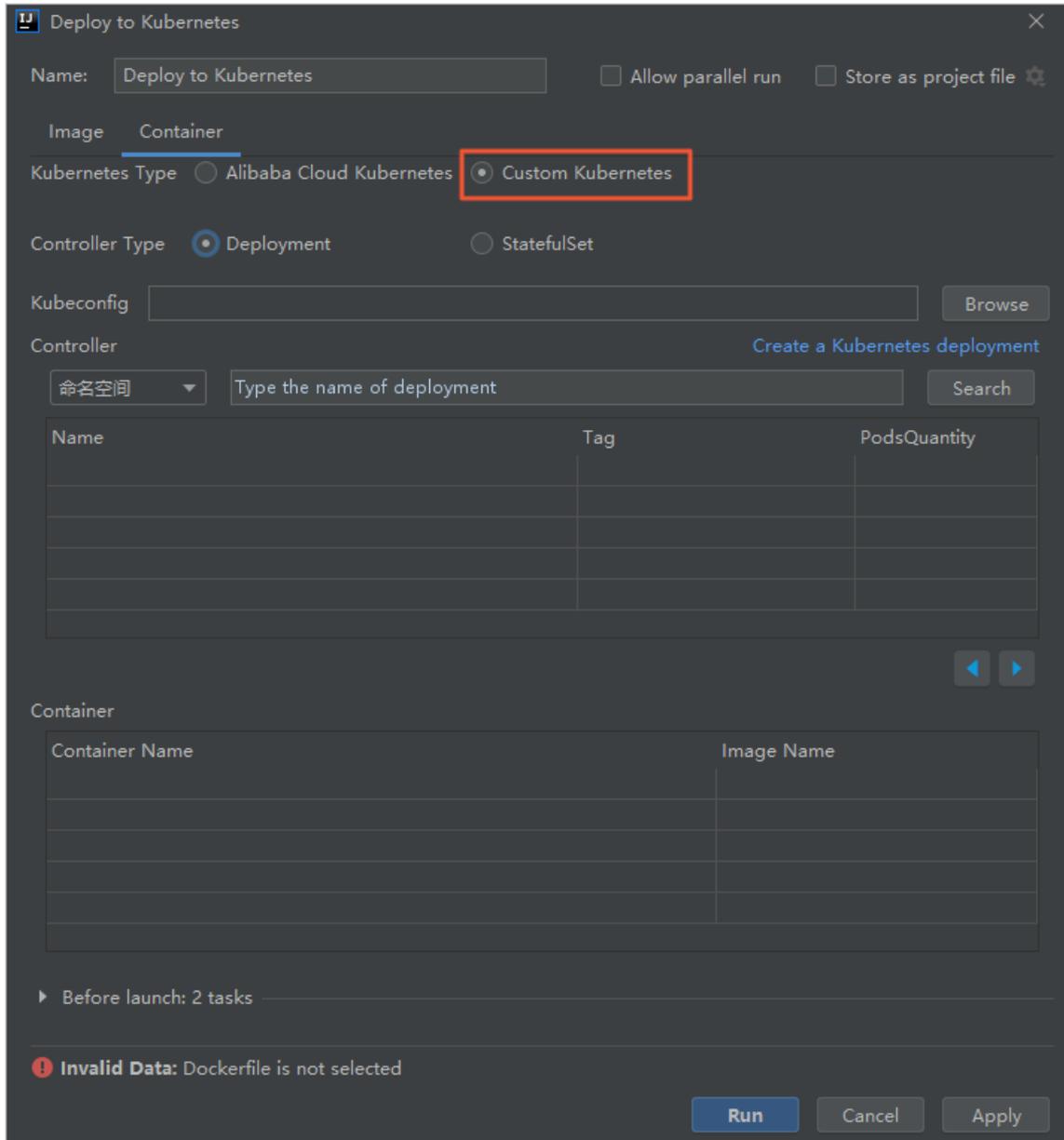
Image页签中的部署参数说明如下。

参数	描述
Name	部署名，建议以英文命名。

参数	描述
Build Image	<ul style="list-style-type: none">▪ Context Directory: 上下文目录。▪ Dockerfile: Dockerfile文件。▪ Version: 镜像版本号。
Image Repositories	<p>选择容器镜像服务的地域、命名空间和镜像仓库。</p> <ul style="list-style-type: none">▪ Alibaba Cloud Container Registry: 阿里云镜像仓库, 详情请参见部署应用到ACR。▪ Custom Container Registry: 自建镜像仓库, 详情请参见部署应用到其它镜像仓库。

 **说明** Context Directory和Dockerfile通常会根据您本地的应用工程自动识别并设置。

- 在Container页签中配置Kubernetes Type、Kubeconfig、Controller Type、Controller和Container参数。



Container页签中的部署参数说明如下。

参数	描述
Kubernetes Type	选择Custom Kubernetes，即部署到自建容器服务Kubernetes。
Kubeconfig	选择自建部署文件。
Controller Type	<ul style="list-style-type: none"> Deployment：无状态应用。 StatefulSet：有状态应用。
Controller	选择区域、命名空间和对应的应用。
Container	选择对应的容器。

4. 先单击Apply，然后单击Run。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

如果您在使用Cloud Toolkit过程中有任何疑问，欢迎您[联系我们](#)。

7.2.2. 使用Eclipse部署应用到自建Kubernetes

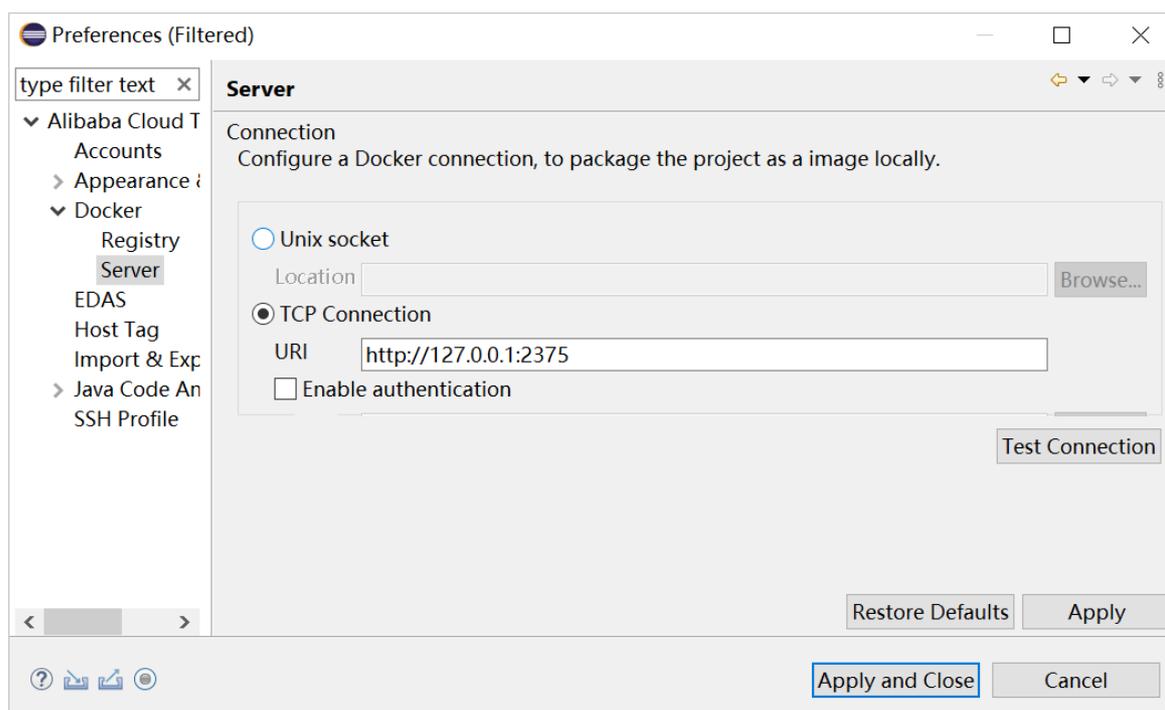
若您使用自建容器服务Kubernetes，可以使用Cloud Toolkit来快速部署应用。本文将介绍如何在Eclipse中使用Cloud Toolkit将应用部署到容器服务Kubernetes。

前提条件

- 您已在Eclipse中安装和配置Cloud Toolkit，请参见在[Eclipse 中安装和配置 Cloud Toolkit](#)。
- 您已在容器镜像服务控制台创建容器镜像仓库，并使用镜像创建应用，请参见[创建容器镜像仓库和使用镜像创建应用](#)。

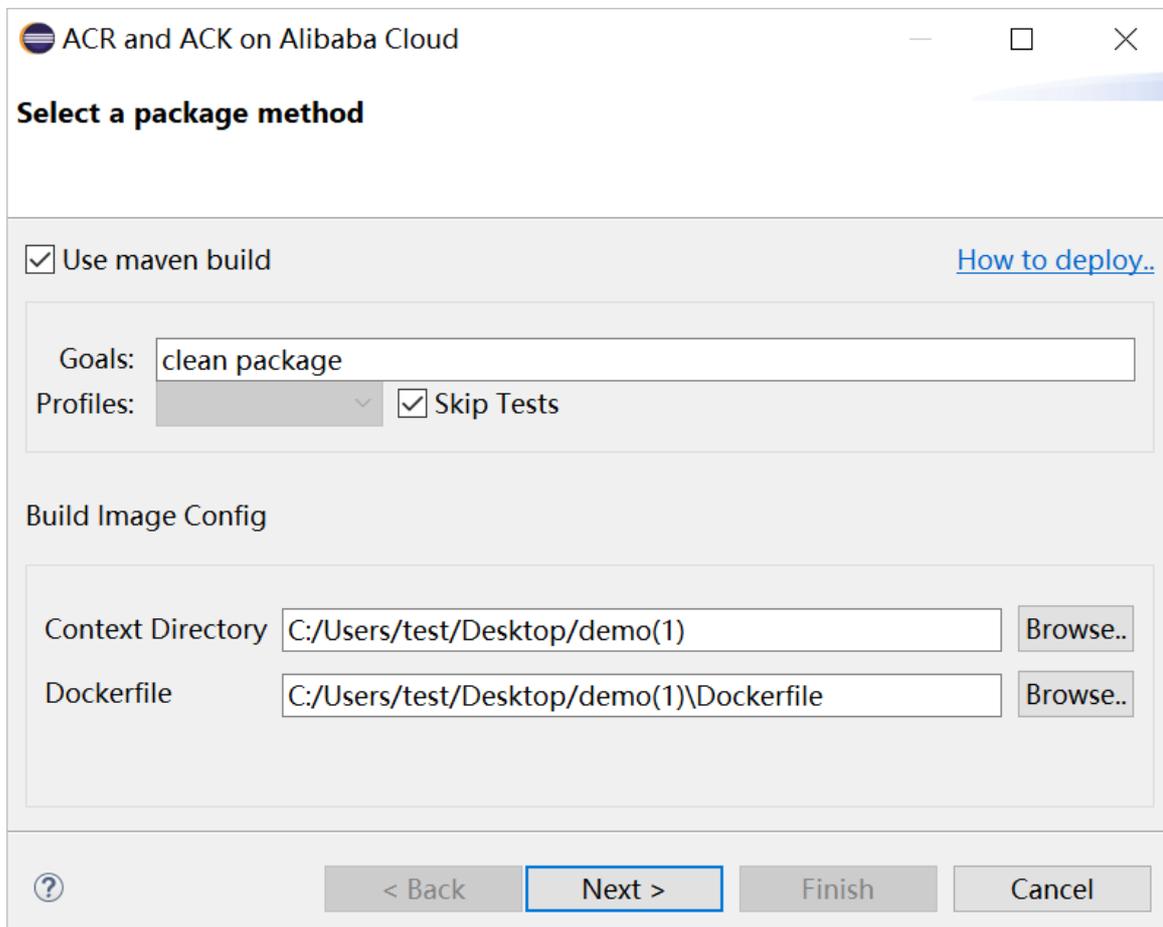
设置Docker环境

1. 在Eclipse工具栏单击Cloud Toolkit图标，在下拉菜单中单击Preference。
2. 在Preference (Filtered) 对话框的左侧导航栏中选择Docker > Server。
3. 在Server界面中设置可连接的Docker环境，然后单击Apply。
 - o 本地Docker环境：
 - 如果您本地为Mac或Linux操作系统，勾选Unix Socket，然后单击Browse，选择本地的Docker安装目录。
 - 如果您本地为 Windows 操作系统，勾选Tcp Connection，然后在URI右侧文档框输入本地Docker的URI，如`http://127.0.0.1:2375`。
 - o 远程Docker环境：勾选TCP Connection，在URI右侧的文本框里输入远端的Docker环境的URI（包括IP地址和端口），如`http://x.x.x.x:2375`，并确保远程主机的HTTP服务开启。



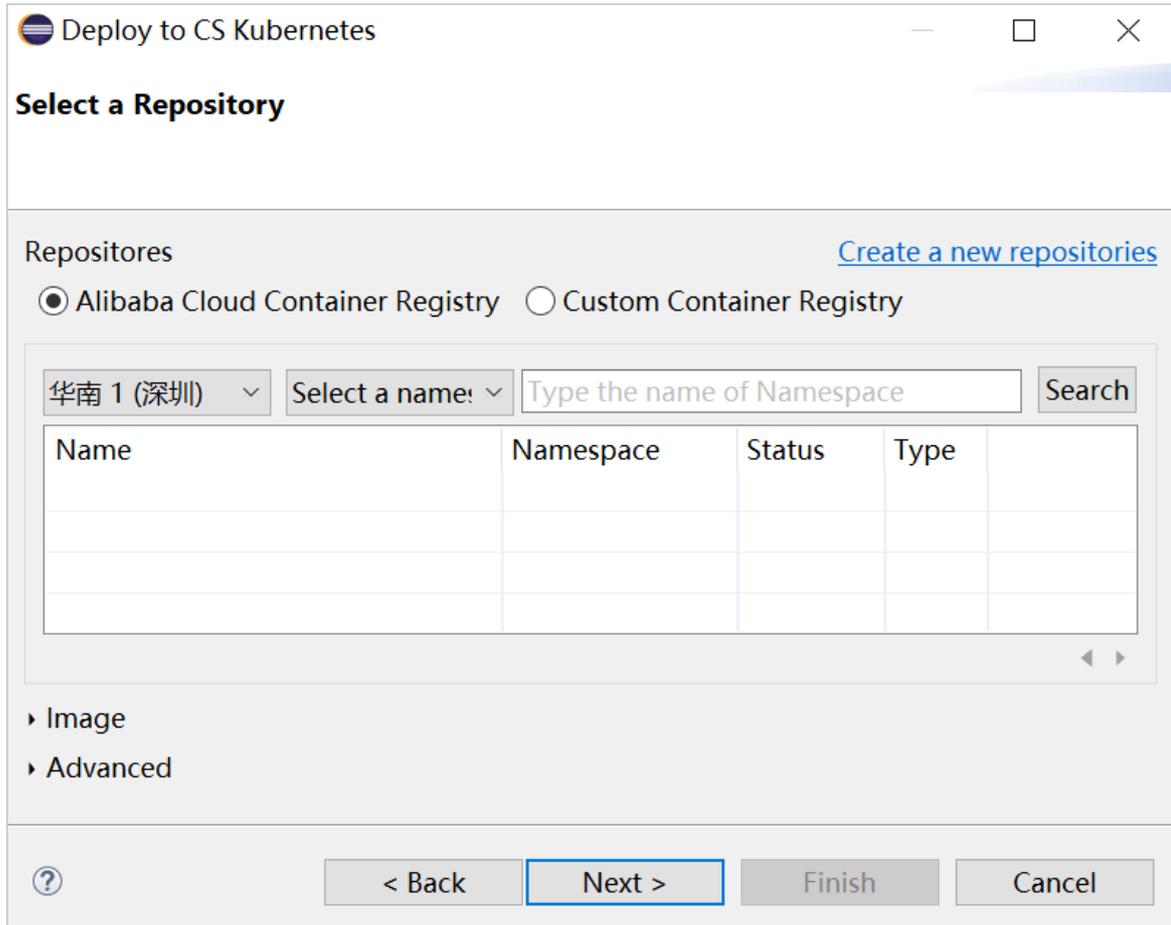
部署应用到容器服务Kubernetes

1. 在Eclipse界面左侧的Package Explorer中右键单击您的Docker应用工程名，在菜单中选择 **Alibaba Cloud > Deploy to Registry/Kubernetes > Deploy to Kubernetes**。
2. 在Deploy to CS Kubernetes对话框Select a package method，选择本地应用程序的Context Directory和Dockerfile（通常会根据您本地的应用工程自动识别并设置），然后单击 **Next**。



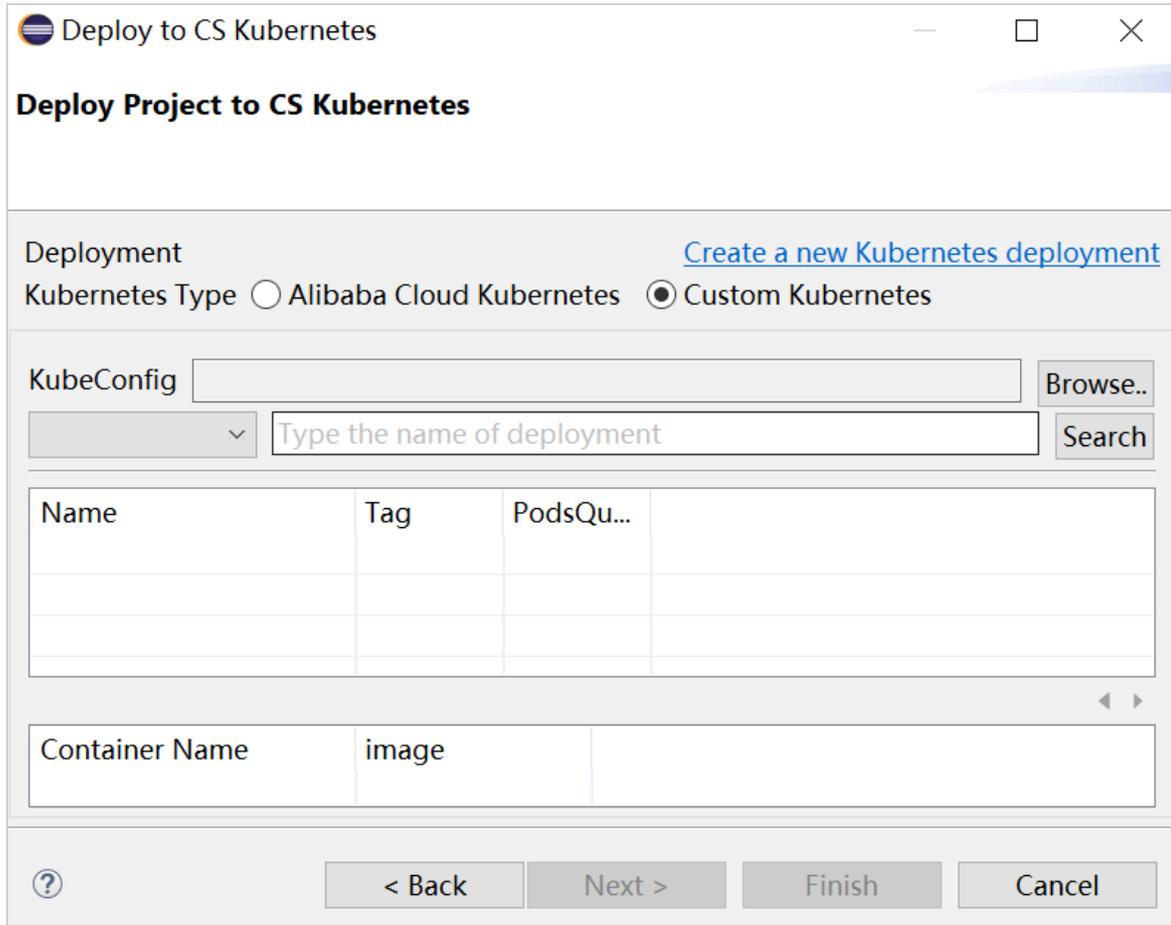
 说明 您可以根据您的需要决定是否勾选Use maven build使用Maven构建应用工程。

3. 在Deploy to CS Kubernetes对话框Select a Repository，选择容器镜像服务的类型、地域、命名空间和镜像仓库，然后单击 **Next**。关于镜像仓库部署详情请参见[使用Eclipse部署应用到镜像仓库](#)。



说明 如果您还没有镜像仓库，在对话框右上角单击Create a new repositories跳转到容器镜像仓库创建镜像仓库。创建步骤请参见[容器镜像仓库](#)。

4. 在Deploy to CS Kubernetes对话框Deploy Project to CS Kubernetes部署容器服务相关参数，然后单击Finish。



容器部署参数说明如下表所示：

参数	描述
Deployment Kubernetes Type	选择Custom Kubernetes，即部署到自建容器Kubernetes。
KubeConfig	选择自建部署KubeConfig文件。
地域	选择地域。
集群	选择集群。
命名空间	选择命名空间。

 **说明** 如果您还没有容器服务，在对话框右上角单击Create a new Kubernetes deployment 跳转到容器服务控制台创建容器服务。创建步骤请参见[首次使用容器服务Kubernetes版](#)。

结果验证

部署开始后，Eclipse的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

如果您在使用Cloud Toolkit过程中有任何疑问，欢迎您[联系我们](#)。

8.部署多模块工程

8.1. 使用IntelliJ IDEA部署多模块工程中的子模块

Cloud Toolkit可以用于部署多模块工程中的某个子模块的场景。本文档将以在IntelliJ IDEA中部署Meetup多模块工程中的Consumer子模块到SAE为例介绍部署方法。

背景信息

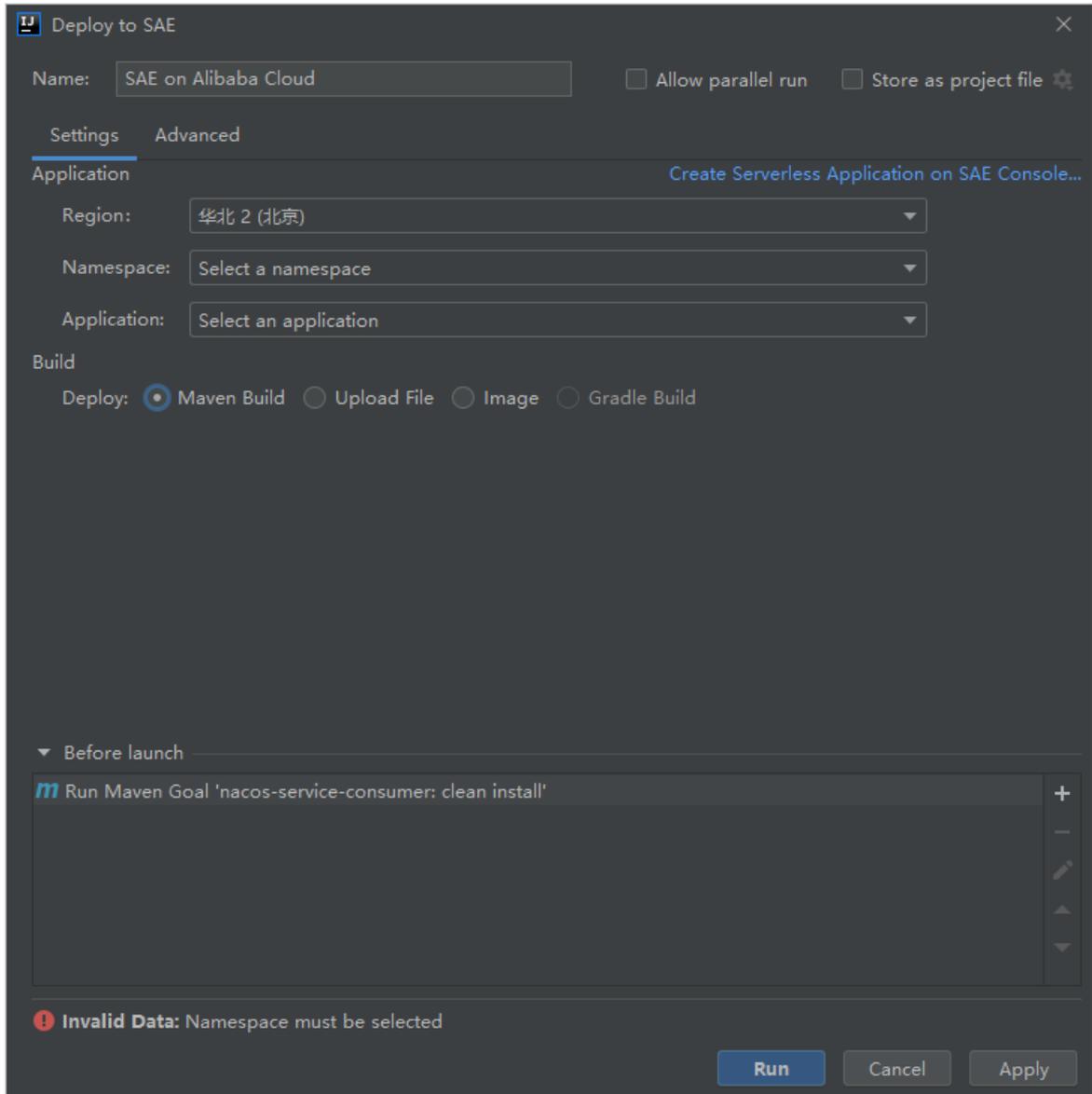
若您有一个Meetup多模块工程，结构为：

- Consumer
- Provider
- Provider-api

其中Consumer模块和Provider模块均为Meetup工程的子模块，且都依赖于Provider-api模块。

部署多模块工程中的子模块

1. 在IntelliJ IDEA界面左侧的Project中右键单击Meetup工程，在快捷菜单中选择Alibaba Cloud > Deploy to SAE...。
2. 在Deploy to SAE对话框中设置部署参数。



说明 若您尚未未在SAE上创建应用，可在对话框右上角单击Create Serverless Application on SAE Console，跳转到SAE控制台创建应用。

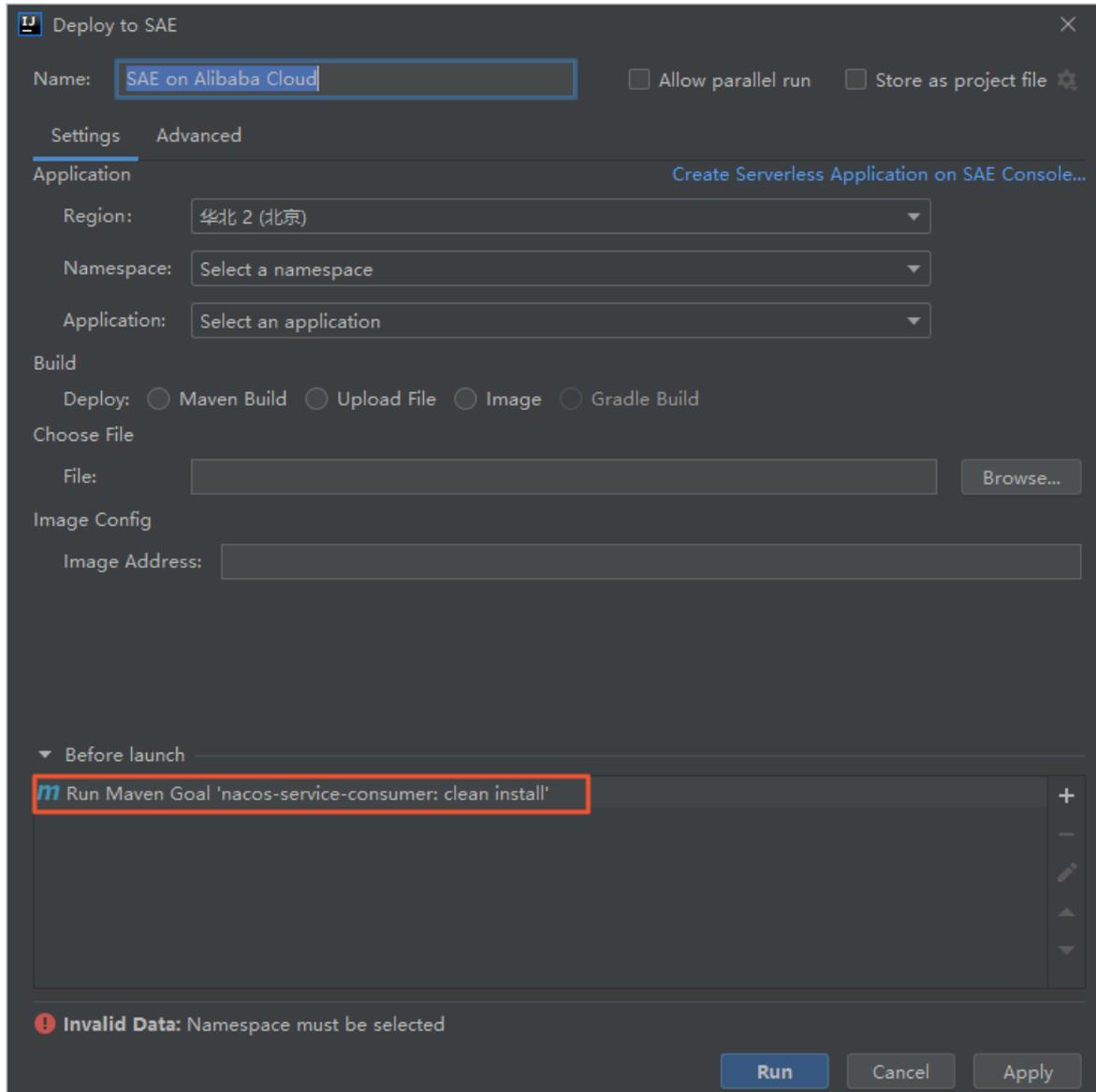
部署参数说明如下。

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。

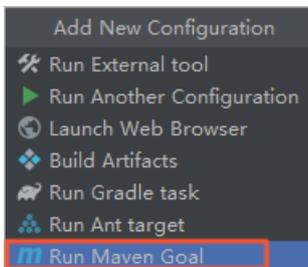
参数	参数	描述
部署方式 (Deploy File)	Maven Build	选择Maven Build方式来构建应用时，系统会默认添加一个Maven任务来构建部署包。如果您需要部署多模块工程中的一个子模块，请参见 部署多模块工程中的子模块 。
	Upload File	选择Upload File方式来构建应用时，选择上传您的WAR包或者JAR包，然后进行部署。
	Image	选择Image方式来构建应用时，需要填入一个镜像地址，然后进行部署。
	Gradle Build	选择Gradle Build方式来构建应用时，可以直接构建并部署。

 **说明** 若您已使用 Jar/War 包部署应用，使用 Cloud Toolkit 部署应用时只能选择 Maven Build 或 Upload File 两种部署方式；若您已使用镜像部署应用，使用 Cloud Toolkit 部署应用时只能选择 Image 部署方式。

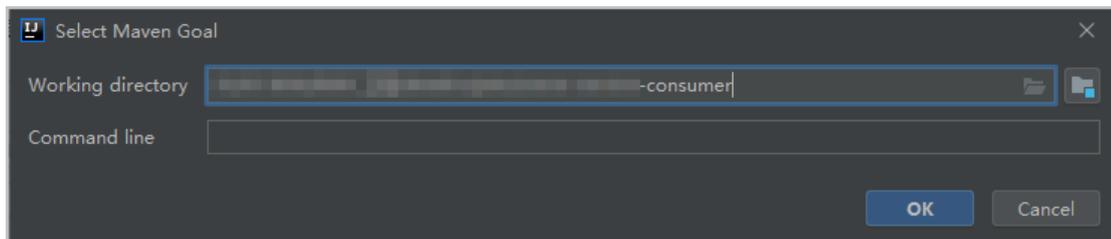
3. 对Meet up父工程执行 `mvn clean install` 命令（默认执行）。



- 4. 对Consumer子工程执行 `mvn clean package` 命令。
 - i. 在Deploy to EDAS对话框的Before launch区域单击+。
 - ii. 在Add New Configuration菜单中选择Run Maven Goal。



- iii. 在Select Maven Goal对话框中单击文件夹图标选择Consumer子模块，在Command line栏输入 `clean package`，然后单击OK。



5. 先单击Apply，然后单击Run。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

您还可以登录SAE控制台，在部署应用的基本信息页面查看部署结果。

8.2. 使用Eclipse部署多模块工程中的子模块

Cloud Toolkit可以用于部署多模块工程中的某个子模块的场景。本文档将以在Eclipse中部署Meetup多模块工程中的Consumer子模块到SAE为例介绍部署方法。

背景信息

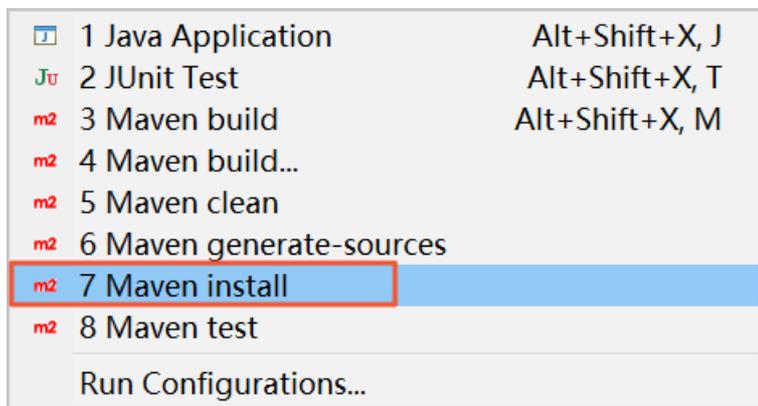
若您有一个Meetup多模块工程，结构为：

- Consumer
- Provider
- Provider-api

其中Consumer模块和Provider模块均为Meetup工程的子模块，且都依赖于Provider-api模块。

部署多模块工程中的子模块

1. 对Meetup父工程执行 `mvn clean install` 命令。以下任意一种方法均可：
 - 在Eclipse界面左侧的Project Explorer中右键单击Meetup工程，在快捷菜单中选择Run As > Maven install。



- cd至Meetup工程所在目录下执行 `mvn clean install` 命令。
2. 对Consumer子工程执行 `mvn clean package` 命令。
 - i. 在Eclipse界面左侧的Project Explorer中右键单击Consumer工程，在快捷菜单中选择Alibaba Cloud > Deploy to SAE...

- ii. 在Deploy to SAE对话框中设置部署参数。在Maven Config区域的Goals栏输入`clean package`命令。

The screenshot shows the 'Deploy to SAE' dialog box with the following settings:

- Application:** [Create serverless application on SAE console...](#)
- Region:** 华北 2 (北京)
- Namespace:** Select a namespace
- Application:** Select an application
- Deploy File:** Maven Build Upload File Image Address
- Maven Config:**
 - Goals:** clean package
 - Profile:** Skip Test

说明 若您尚未在SAE上创建应用，可在对话框右上角单击Create Serverless Application on SAE console，跳转到SAE控制台创建应用。

部署参数说明如下表所示：

参数	参数	描述
应用信息 (Application)	Region	应用所在地域。
	Namespace	应用所在命名空间。
	Application	应用名称。
部署方式 (Deploy File)	Maven Build	选择Maven Build方式来构建应用时，可直接构建并部署。
	Upload File	选择Upload File方式来构建应用时，选择上传您的WAR包或者JAR包，然后进行部署。

参数	参数	描述
	Image Address	选择Image方式来构建应用时，需要填入一个镜像地址，然后进行部署。

 **说明** 若您已使用Jar/War包部署应用，使用Cloud Toolkit部署应用时只能选择Maven Build或Upload File两种部署方式；若您已使用镜像部署应用，使用Cloud Toolkit部署应用时只能选择Image部署方式。

iii. 单击Deploy。

结果验证

部署开始后，Eclipse的Console区域会打印部署日志。您可以根据日志信息检查部署结果。

您还可以登录SAE控制台，在部署应用的基本信息页面查看部署结果。

9.部署应用到小程序云

9.1. 使用IntelliJ IDEA部署应用到小程序云

如果您是阿里云小程序云用户，可以使用Cloud Toolkit来部署小程序应用，适用于小程序的首次部署、快速迭代、版本回滚等场景。本文将介绍如何在IntelliJ IDEA中使用Cloud Toolkit部署小程序应用到小程序云。

前提条件

- 已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。
- 已在小程序云控制台创建云应用并构建环境，请参见[创建云应用](#)和[构建环境](#)。

背景信息

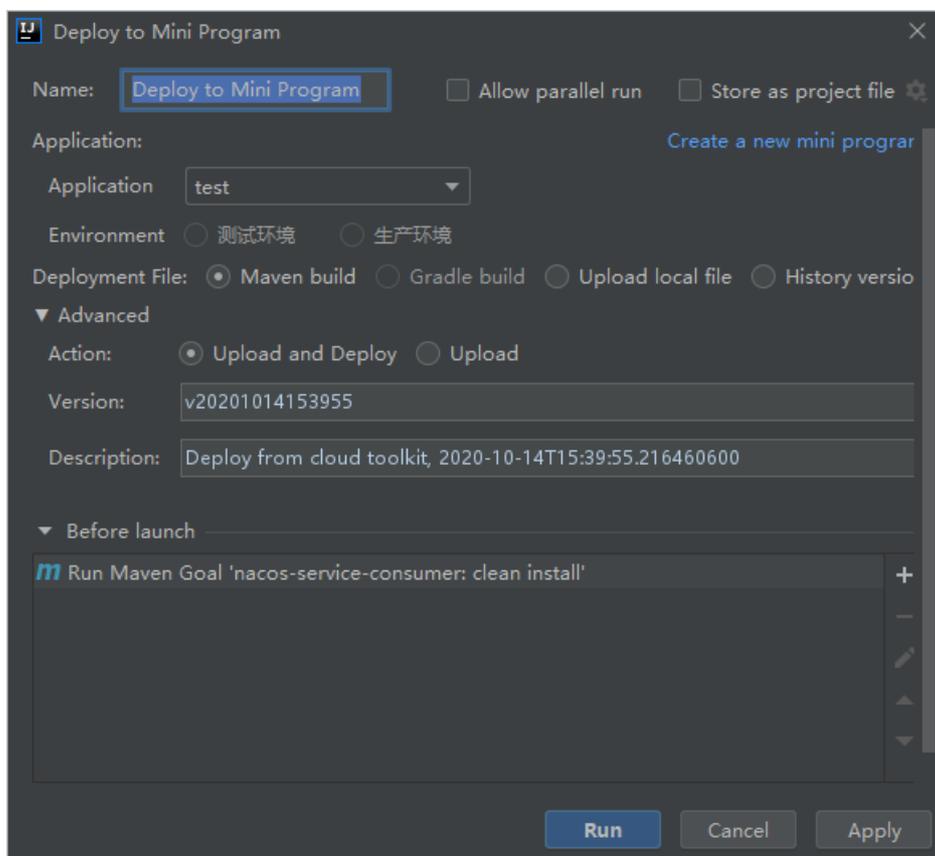
小程序云是阿里云面向小程序场景提供的一站式云服务，帮助开发者实现一云多端的业务战略。开发者可通过小程序云支撑各类小程序前端，在一朵云内实现统一的资源管理、统一的数据运营和统一的业务设计。详情请参见[什么是小程序云?](#)。

小程序云应用是面向小程序的应用场景，为开发者提供的一键构建后端应用运行环境、后端服务部署、运维监控等能力的一站式小程序部署服务。

使用Cloud Toolkit可以快速部署小程序云应用至小程序云，暂不支持部署小程序Serverless。

操作步骤

1. 在IntelliJ IDEA中打开您的小程序工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Deploy to Mini Program...
3. 在Deploy to Mini Program对话框中进行部署配置。



 **说明** 若您尚未在小程序云上创建应用，可在对话框右上角单击Create a new mini program，跳转到小程序云控制台创建应用。

部署参数说明如下。

参数	参数	描述
应用信息 (Application)	Application	应用名称，选择部署应用。
	Environment	应用环境，分为测试环境和生产环境。
部署方式 (Deploy File)	Maven Build	若当前工程采用Maven构建，可以直接构建并部署。
	Gradle Build	若当前工程采用Gradle构建，选择此项来构建并部署。
	Upload local file	若当前工程不是采用Maven构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
	History version	若需回滚到历史版本，可选择此项再选择版本信息。
分批部署 (Advanced)	Action	根据实际选择为上传并部署或者仅上传。
	Version	会默认生成本次部署的版本信息，可自行修改。
	Description	默认生成本息部署的说明信息，可自行修改。

4. 先单击Apply，然后单击Run。

结果验证

部署开始后，IntelliJ IDEA的Console区域会打印部署日志，请根据日志信息检查部署结果。

您还可以登录阿里云小程序云控制台，在所部署应用的发布信息页签查看部署结果。

后续步骤

在IntelliJ IDEA底部选择Alibaba Cloud View > Alibaba Mini Program View页签。在Alibaba Mini Program View页签中展示了当前账号下所有的小程序云应用，您可以对其执行查看、部署历史版本、重启、停止等操作。

9.2. 使用Eclipse部署应用到小程序云

如果您是阿里云小程序云用户，可以使用Cloud Toolkit来部署小程序应用，适用于小程序的首次部署、快速迭代、版本回滚等场景。本文将介绍如何在Eclipse中使用Cloud Toolkit部署小程序应用到小程序云。

前提条件

- 已安装和配置Cloud Toolkit，请参见[在Eclipse中安装和配置Cloud Toolkit](#)。

- 已在小程序云控制台创建云应用并构建环境，请参见[创建云应用](#)和[构建环境](#)。

背景信息

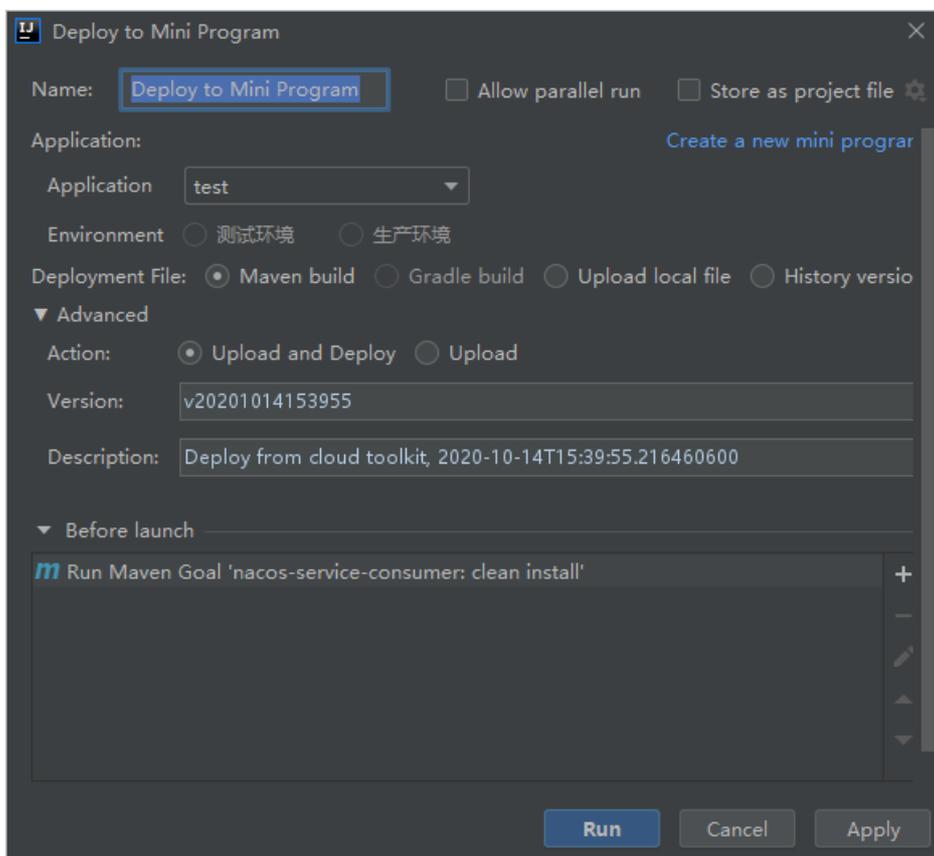
小程序云是阿里云面向小程序场景提供的一站式云服务，帮助开发者实现一云多端的业务战略。开发者可通过小程序云支撑各类小程序前端，在一朵云内实现统一的资源管理、统一的数据运营和统一的业务设计。详情请参见[什么是小程序云?](#)。

小程序云应用是面向小程序的应用场景，为开发者提供的一键构建后端应用运行环境、后端服务部署、运维监控等能力的一站式小程序部署服务。

使用Cloud Toolkit可以快速部署小程序云应用至小程序云，暂不支持部署小程序Serverless。

操作步骤

- 在Eclipse中打开您的小程序工程。
- 在Eclipse界面左侧的Project Explorer中右键单击待部署的工程名，在快捷菜单中选择Alibaba Cloud > Deploy to Mini Program。
- 在Deploy to Mini Program对话框中进行部署配置。



说明 若您尚未在小程序云上创建应用，可在对话框右上角单击Create a new mini program，跳转到小程序云控制台创建应用。

部署参数说明如下表所示：

参数	参数	描述
	Application	应用名称，选择部署应用。

参数信息 (Application)	参数	描述
部署方式 (Deploy File)	Environment	应用环境，分为测试环境和生产环境。
	Maven Build	若当前工程采用 Maven 构建，可以直接构建并部署。
	Gradle Build	若当前工程采用 Gradle 构建，选择此项来构建并部署。
	Upload local file	若当前工程不是采用 Maven 构建，或本地已存在部署文件的压缩包，则上传本地的部署文件即可。
分批部署 (Advanced)	History version	若需回滚到历史版本，可选择此项再选择版本信息。
	Action	根据实际选择为上传并部署或者仅上传。
	Version	会默认生成本次部署的版本信息，可自行修改。
	Description	默认生成本次部署的说明信息，可自行修改。

4. 先单击 **Apply**，然后单击 **Run**。

结果验证

部署开始后，Eclipse 的 **Console** 区域会打印部署日志，请根据日志信息检查部署结果。

您还可以登录阿里云小程序云控制台，在所部署应用的 **发布信息** 页签查看部署结果。

后续步骤

在 Eclipse 顶部单击 Alibaba Cloud Toolkit 图标，然后选择 **Alibaba Cloud Explorer > Alibaba Cloud Mini Program**。在 **Alibaba Cloud Mini Program** 页签中展示了当前账号下所有的小程序云应用，您可以对其执行查看、部署历史版本、重启、停止等操作。

10.部署多工程应用

本文为您介绍如何在IntelliJ IDEA中使用Cloud Toolkit同时部署多个工程应用。

前提条件

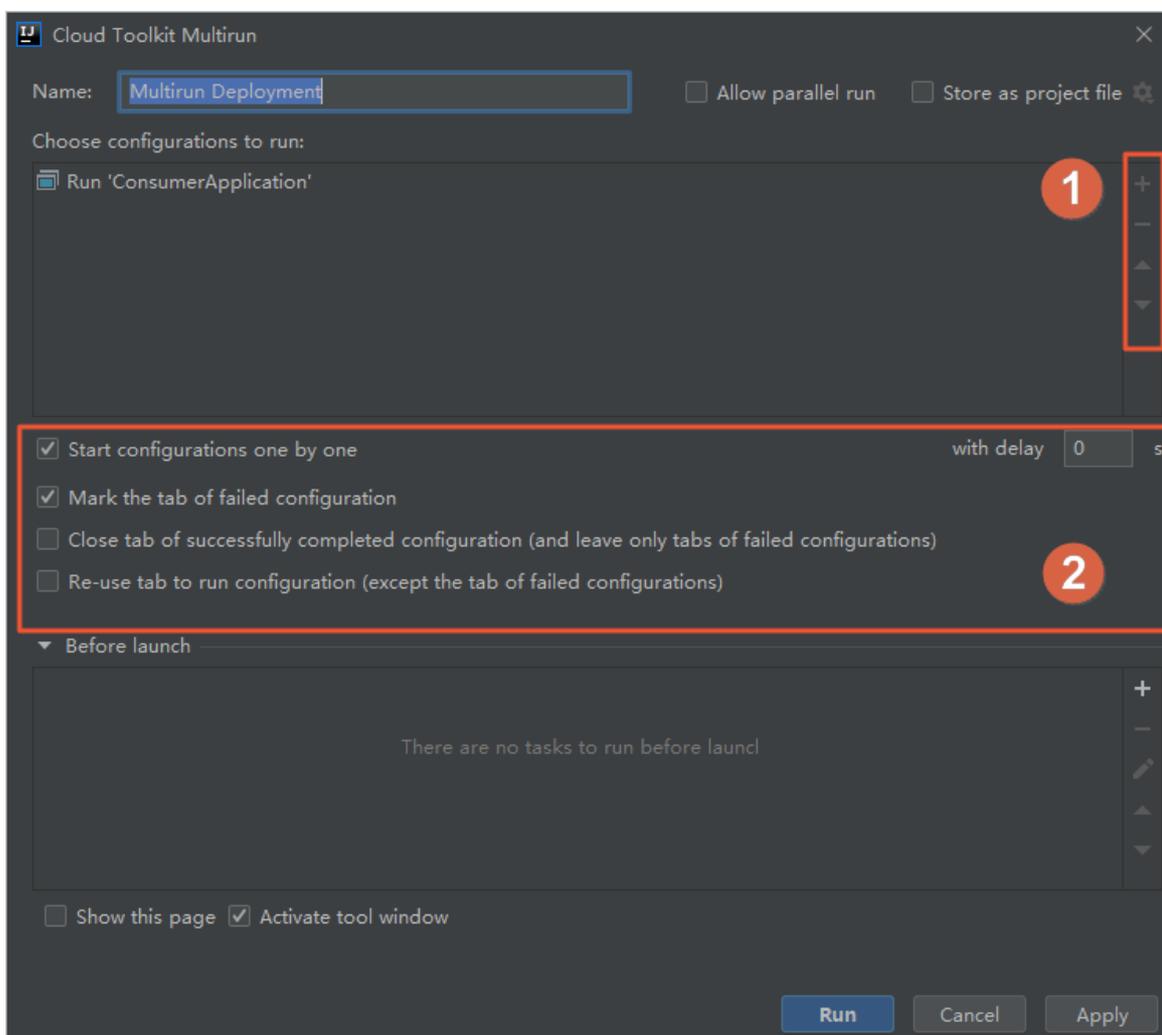
已安装和配置Cloud Toolkit，请参见[在IntelliJ IDEA中安装和配置Cloud Toolkit](#)。

背景信息

开发者在本地或微服务应用中，会遇到需要同时部署多个应用（服务）的场景，希望将本地多个JAR包同时部署。Cloud Toolkit中的Multirun Deployment功能能帮助您同时部署多个应用。

操作步骤

1. 在IntelliJ IDEA中打开您的工程。
2. 在IntelliJ IDEA顶部菜单栏中选择Tools > Alibaba Cloud > Multirun Deployment ...。
3. 在Cloud Toolkit Multirun对话框中单击右侧的+，选择需要同时部署的应用。



部署区域功能说明如下。

区域	描述
区域1	<ul style="list-style-type: none"> ◦ : 添加部署应用。 ◦ : 删除部署应用。 ◦ : 上移部署应用。 ◦ : 下移部署应用。
区域2	<ul style="list-style-type: none"> ◦ Start configurations one by one: 按照应用先后顺序依次部署应用。 <div style="border: 1px solid #ccc; background-color: #e0f2f7; padding: 5px; margin: 5px 0;"> <p> 说明 with delay: 可设置每个部署应用之间的间隔时间, 单位为s。</p> </div> <ul style="list-style-type: none"> ◦ Mark the tab of failed configuration: 标记部署中的应用, 使用*标记。 ◦ Close tab of successfully completed configuration (and leave only tab of failed configurations): 自动关闭已成功完成部署的应用, 部署失败的应用除外。 ◦ Re-use tab to run configuration (except the tab of failed configurations): 重利用部署成功的应用, 部署失败的应用除外。

4. 先单击Apply, 然后单击Run。