ALIBABA CLOUD

阿里云

Serverless 应用引擎

应用部署

文档版本：20210902

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.应用部署概述 06
2.JAR包或WAR包部署 09
2.1. Java应用 09
2.1.1. Java运行环境说明 09
2.1.2. 在SAE控制台使用WAR包部署Java Web应用 10
2.1.3. 在SAE控制台使用JAR文件部署微服务应用 15
2.1.4. 在SAE控制台使用镜像方式部署Java应用 21
2.2. PHP应用 27
2.2.1. 在SAE控制台使用镜像部署PHP应用 27
2.3. 使用镜像部署多语言应用 34
2.4. 灰度发布应用 41
2.5. 管理灰度规则（Java） 46
2.6. 多语言应用 49
3.插件部署 50
3.1. 通过Maven插件自动部署应用 50
3.2. 通过IntelliJ IDEA插件部署应用 58
3.3. 通过Eclipse插件一键部署应用 64
4.CICD部署 71
4.1. 使用云效部署应用至SAE 71
4.1.1. 部署Java应用至SAE 71
4.1.2. 部署Golang应用至SAE 85
4.1.3. 部署Node.js、Python、PHP应用至SAE 100
4.2. 使用云效2020部署应用至SAE 105
4.2.1. 部署Java应用至SAE 105
4.2.2. 部署Golang应用至SAE 114
4.2.3. 部署Node.js应用至SAE 124

4.2.4. 部署Python应用至SAE 132
4.2.5. 部署PHP应用至SAE 140
4.3. 使用Jenkins创建应用的持续集成 149
4.3.1. 使用Jenkins创建Java应用的持续集成 149
4.3.2. 使用Jenkins创建多语言应用的持续集成 154
5.高级配置 162
5.1. 设置PHP应用监控 162
5.2. 设置PHP应用配置文件 163
5.3. 设置启动命令 165
5.4. 设置Java Tomcat参数 173
5.5. 设置环境变量 177
5.6. 设置Hosts绑定 179
5.7. 设置健康检查 181
5.8. 设置日志收集 190
5.9. 设置NAS存储 193
5.10. 设置OSS存储 196
5.11. 设置应用生命周期管理 200
5.12. 制作应用容器Docker镜像 205
5.13. 注入配置信息 211
6.设置限流降级 213
7.设置微服务无损下线 215

1.应用部署概述

SAE是面向应用的Serverless PaaS平台，向上抽象了应用的概念，您可以无需管理和维护集群与服务器，只需专注于设计和构建应用程序，将其部署在SAE。您可以通过SAE控制台部署应用，也可以通过toolkit-maven-plugin、Alibaba Cloud Toolkit for IntelliJ IDEA和Alibaba Cloud Toolkit for Eclipse等插件将应用部署在SAE上。

部署方式

如需升级应用，建议使用灰度发布应用。

应用的部署方式参见以下表格。

应用举例	部署方式	参考开发文档
原生Spring Cloud	WAR、JAR、镜像	将 Spring Cloud 应用托管到 SAE
原生Dubbo	WAR、JAR、镜像	将 Dubbo 应用托管到 SAE
HSF	WAR、JAR、镜像	无
多语言应用	镜像	无

使用Spring Cloud、Dubbo 和HSF框架来开发的应用都可以部署在SAE中，但部署方式不同其应用的运行环境也不同。

- Spring Cloud和Dubbo应用通过WAR包部署时，选择apache-tomcat相关版本的运行环境。
- Spring Cloud和Dubbo应用通过JAR包部署时，选择标准Java应用运行环境。
- HSF应用通过WAR或JAR包部署时，选择EDAS-Container相关版本的运行环境。

服务注册中心说明

将微服务应用托管到SAE，无需关注服务注册、发现和配置管理，SAE自带服务注册中心和配置管理，即SAE注册中心，您只需关注应用自身的逻辑。

您在SAE部署应用时，SAE服务注册中心以高优先级自动设置Nacos Server服务端地址和服务端口，以及namespace、access-key、secret-key、context-path等信息，无需进行任何额外的配置。

SAE服务注册中心，与Nacos、Eureka和Consul相比，具有以下优势：

- 共享组件，节省了您部署、运维Nacos、Eureka或Consul的成本。
- 对服务注册和发现的调用进行了链路加密，保护您的服务安全性，无需担心服务被未授权的应用发现。
- SAE服务注册中心与SAE其他组件紧密结合，提供了完整的微服务解决方案，包括环境隔离、平滑上下线、灰度发布等。

如果您坚持使用自建Nacos为服务注册中心，请参见如何搭建 Nacos 为服务注册中心（不推荐）进行搭建。

说明

- 使用自建Nacos时请确保SAE的网络与自建Nacos的网络互通。
- 使用自建Nacos为服务注册中心，在部署应用时建议使用镜像方式或者IAR包方式，并配置启动参数 `-Dnacos.use.endpoint.parsing.rule=false` 和 `-Dnacos.use.cloud.namespace.parsing=false` 。
  - 如采用镜像方式，请将 `-Dnacos.use.endpoint.parsing.rule=false` 和 `-Dnacos.use.cloud.namespace.parsing=false` 配置在镜像文件中。关于Docker镜像制作方法，请参见制作应用容器Docker镜像。
  - 如采用IAR包方式，请在控制台**启动命令设置**区域的**options设置**文本框输入 `-Dnacos.use.endpoint.parsing.rule=false -Dnacos.use.cloud.namespace.parsing=false` 。

控制台部署

使用控制台部署应用SAE。建议使用Chrome浏览器进行控制台操作。

如何设置启动命令

您可以在应用部署时为应用设置容器启动和运行时所需的命令，如Nginx。

如何设置环境变量

您可以为应用程序配置运行所需的特定环境变量，如Java_home及其Path，便于部署应用后灵活变更应用的配置。

如何设置Hosts绑定

SAE 支持应用实例级别的实例通过Host绑定对主机名进行解析，方便应用实例通过主机名进行访问。

如何设置应用健康检查

健康检查能帮助您了解集群环境下整个服务的运行状态，从而为审查与定位问题提供帮助。

如何设置日志收集

文件日志收集可以无限制行数查看日志、自行聚合分析日志，方便业务日志对接，按日志使用量计费 。

如何挂载NAS存储

通常容器销毁以后数据也将丢失，数据丢失意味着线上生产环境的灾难性事件。SAE的NAS存储功能，解决了应用实例数据持久化和实例间多读共享数据的问题。

如何设置PostStart和PreStop

通常应用运行前和停止前都会有一些动作，例如用于部署资源的运行前任务，停止前优雅下线应用和通知其他服务或者应用。SAE依托于K8s完美集成了该功能，支持配置PostStart和PreStop。

工具部署

除通过控制台方式进行应用部署，还可通过以下工具进行部署。

应用部署高级设置

在应用部署时，可以参考以下文档进行启动命令配置、环境变量配置、Host绑定配置、健康检查配置、日志收集配置和持久存储配置等。

> **说明** 以上应用的高级配置功能，您可以在创建、部署应用时设置，也可以在部署应用后依据实际情况进行设置。如果您选在应用部署后设置，应用将会重启生效该配置，请选择业务较少的时段进行高级设置。

2.JAR包或WAR包部署

2.1. Java应用

2.1.1. Java运行环境说明

本文介绍SAE应用在部署时的Java运行环境，您可以根据实际需求选择。

运行环境	更新时间	JDK版本	操作系统版本	是否开源	是否维护	说明
Dragonwell 8	2020-09-27	1.8.0_262	CentOS Linux release 7.8.2003	是	是	默认开启Wisp协程，Netty NIO兼容性问题自查等。更多信息，请参见开源版Dragonwell 8。
Dragonwell 11	2020-09-27	11.0.8.8-AJDK	CentOS Linux release 7.9.2009	否	是	具体信息，请参见开源版Dragonwell 11。
Open JDK 8	2019-06-19	1.8.0_191	CentOS Linux release 7.6.1810	是	否	无
Open JDK 7	2019-02-25	1.7.0_201	CentOS Linux release 7.6.1810	是	否	无
openjdk-8u191-jdk-alpine3.9	2019-02-25	1.8.0_191	alpine 3.9.2	是	否	alpine镜像部分命令与CentOS系统中命令不同。例如： • **telnet**：alpine镜像中**telnet**命令不支持 `-e` 参数。 • **curl** • **wget**

运行环境	更新时间	JDK版本	操作系统版本	是否开源	是否维护	说明
openjdk-7u201-jdk-alpine3.9	2019-02-25	1.7.0_201	alpine 3.9.2	是	否	alpine镜像部分命令与CentOS系统中命令不同。例如： • telnet：alpine镜像中telnet命令不支持 -e 参数。 • curl • wget

2.1.2. 在SAE控制台使用WAR包部署Java Web应用

应用开发完成后，您可以将应用部署到SAE进行托管。本文介绍如何在SAE控制台以WAR包方式部署Java Web应用。

前提条件

- 创建命名空间。
- 创建VPC。

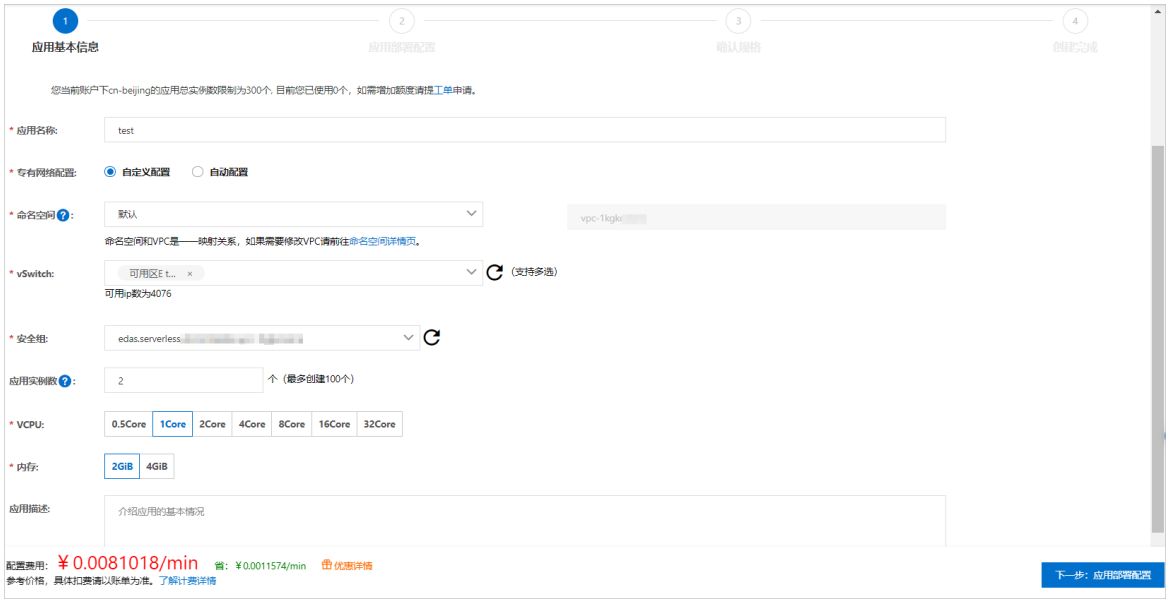
背景信息

在SAE中应用部署方式如下表所示。

应用举例	部署方式	参考文档
原生Spring Cloud	WAR、JAR、镜像	将Spring Cloud应用托管到SAE
原生Dubbo	WAR、JAR、镜像	将Dubbo应用托管到SAE
HSF	WAR、JAR、镜像	无
多语言应用	镜像	使用镜像部署多语言应用

操作步骤

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。

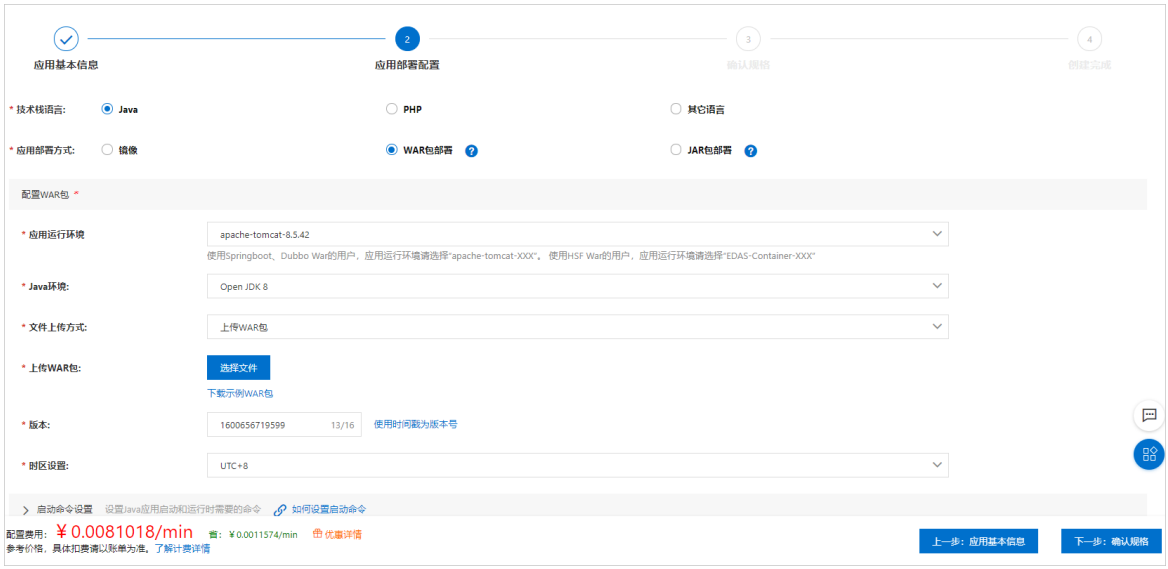


参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。

参数名	说明
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。
应用实例数	选择需要创建的实例个数。
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，配置相关参数。



参数说明如下表所示：

参数名	说明
技术栈语言	SAE支持Java、PHP及其它语言。 本文选择Java。

参数名	说明
应用部署方式	Java应用支持以下部署方式： ○ **镜像** ○ **WAR包部署** ○ **JAR包部署** 本文选择**WAR包部署**。
应用运行环境	○ **apache-tomcat-XXX**：适用于Spring Boot或Dubbo应用。 ○ **EDAS-Container-XXX**：适用于HSF应用。
Java环境	您可以根据需要选择Java环境： ○ **openjdk-XXXXX-jdk-alpine3.9**：基于Alphine操作系统，基础镜像小，实例交互速度快。 ○ **Open JDK X**：基于CentOS操作系统，基础镜像大，实例交互速度慢。 ○ **Dragonwell X**：支持最新OpenJDK 8版本，支持应用启动及运行时加速，提升GC效率等。更多信息，请参见设置启动命令。 **说明** 基于Alphine操作系统的Java环境附带的工具或命令少于基于CentOS操作系统的Java环境，如果您选择了**openjdk-XXXXX-jdk-alpine3.9**，可能会导致Webshell中部分命令无法执行，请根据实际情况选择环境。
文件上传方式	可选择**上传WAR包**或**WAR包地址**。 ○ **上传WAR包**：单击**选择文件**，选择待部署WAR包。 ○ **WAR包地址**：输入WAR包的存放地址。 **注意** 应用部署程序包名仅允许字母、数字、短划线（-）和下划线（_）。
版本	设置应用版本号，您可以选择输入版本号或者单击**使用时间戳为版本号**将时间戳作为应用版本号。
时区设置	选择当前应用所在时区，例如**UTC+8**。

5. （可选）您可以根据需要设置以下高级选项。
   - **（可选）启动命令设置**：请参见设置启动命令。
   - **（可选）Java Tomcat设置**：请参见设置Java Tomcat参数。
   - **（可选）环境变量设置**：请参见设置环境变量。
   - **（可选）Hosts绑定设置**：请参见设置Hosts绑定。

- （可选）**应用健康检查设置**：请参见设置健康检查。
- （可选）**应用生命周期管理设置**：请参见设置应用生命周期管理。
- （可选）**日志收集设置**：请参见设置日志收集。
- （可选）**日志收集设置**：请参见设置日志收集。
- （可选）**持久化存储**：请参见设置NAS存储。
- （可选）**配置管理**：请参见注入配置信息。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

结果验证

应用部署后，您可以通过以下两种方式验证应用的部署结果：

- 查看应用实例运行状态。
  i. 登录SAE控制台。
  ii. 在左侧导航栏单击**应用列表**，在**应用列表**页面单击具体应用。
  iii. 在**应用详情**页面的**实例部署信息**页签查看实例的运行状态。
  如果**运行状态**显示为绿色的Running或者Completed，表示应用部署成功。
- 配置公网负载均衡并访问应用。
  应用部署后，根据实际需要，通过配置负载均衡SLB在指定范围内开放应用访问，以便其它应用访问。
  负载均衡包括以下两种：
  - 私网负载均衡：在应用所在的VPC内提供应用的访问入口，保证应用能被同VPC内的其它应用访问。
  - 公网负载均衡：为该应用自动购买公网SLB服务，或者复用在SLB控制台已购买的实例，保证应用能被公网中的其它应用访问。

> **说明** 配置公网SLB访问和配置私网SLB访问的步骤相同，具体操作请参见为应用绑定SLB。SLB绑定完成后，在浏览器输入由SLB的IP地址、端口及应用部署包名组成的访问地址并访问，例如192.168.XX.XX/80/image，即可进入应用。

查看应用监控指标

应用成功部署后，您可以在**应用监控**模块中查看应用的关键性能指标。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏按需查看应用监控数据：
   - **应用总览**：选择**应用监控 > 应用总览**，查看应用的关键性能指标，例如总请求量、平均响应时间等。
   - **应用详情**：选择**应用监控 > 应用详情**，查看更全面的应用监控数据，例如JVM监控、异常分析等。
   - **接口调用**：选择**应用监控 > 接口调用**，查看以接口为维度的监控数据，例如调用拓扑图、链路上游和链路下游等。
   - **高级监控**：选择**应用监控 > 高级监控**，查看更高性能的监控管理数据，例如应用全局拓扑、异常SQL分析和本地调用堆栈等。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操

作，请参见管理应用生命周期。

- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.1.3. 在SAE控制台使用JAR文件部署微服务应用

本文以两个简单的示例JAR文件分别表示微服务的提供者和消费者。提供者提供简单的Echo服务，会返回消费者输入的消息。本文介绍在SAE控制台部署微服务应用的操作步骤及验证方式。

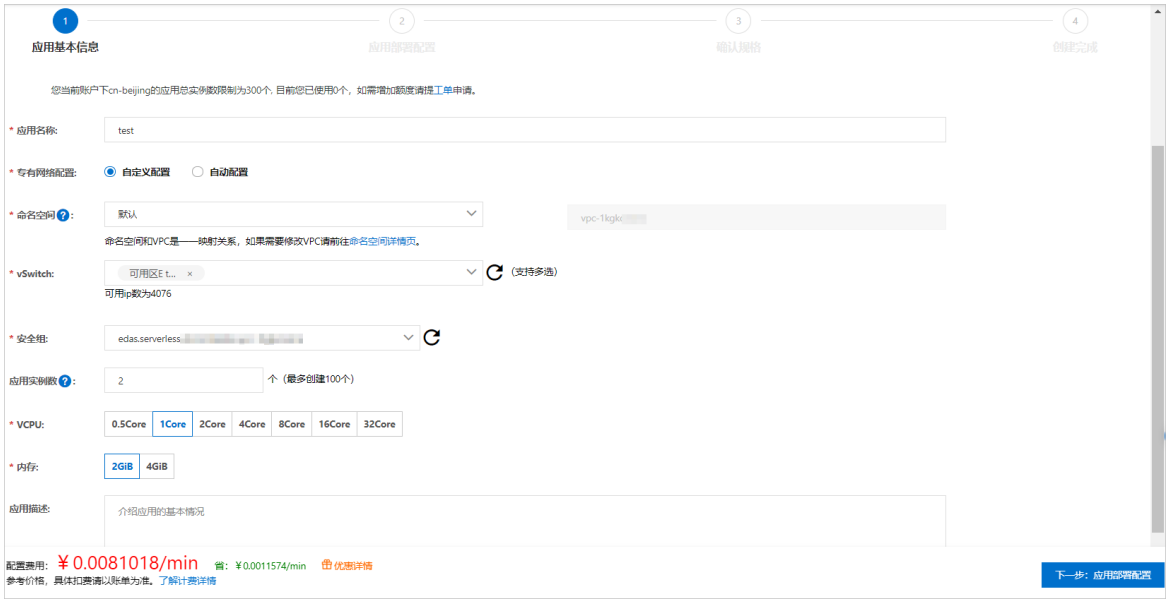
前提条件

- 创建命名空间。
- 创建VPC。

操作步骤

按照以下步骤将示例提供者微服务应用部署到SAE。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。

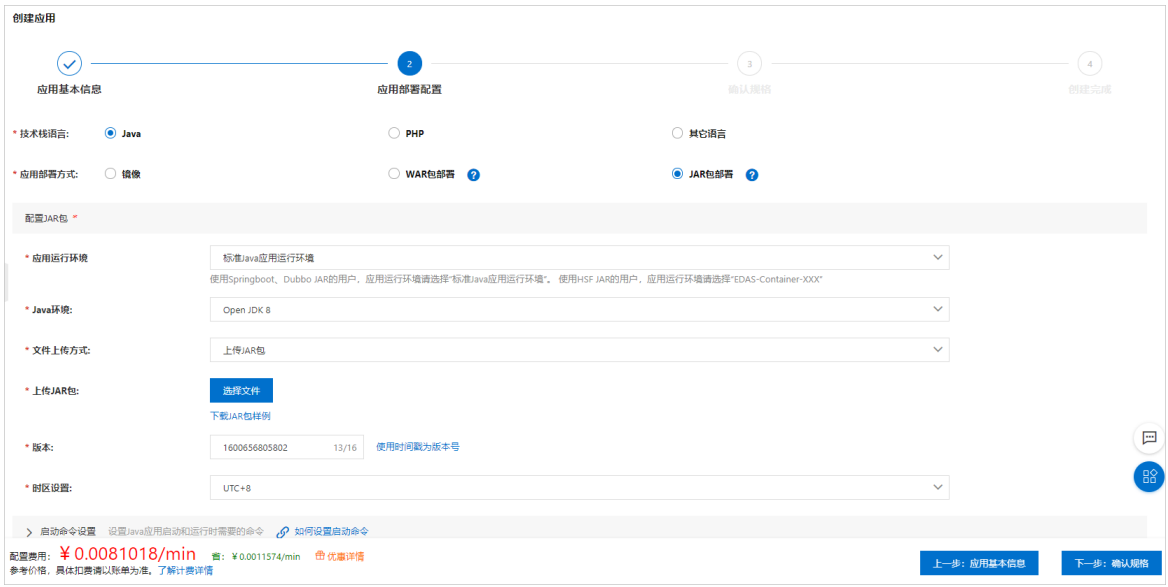


参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。

参数名	说明
应用实例数	选择需要创建的实例个数。
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，配置相关参数。



参数说明如下表所示：

参数名	说明
技术栈语言	SAE支持**Java**、**PHP**及**其它语言**。 本文选择**Java**。
应用部署方式	Java应用支持以下部署方式： ○ **镜像** ○ **WAR包部署** ○ **JAR包部署** 本文选择**JAR包部署**。
应用运行环境	○ **标准Java应用运行环境**：适用于Spring Boot或Dubbo应用。 ○ **EDAS-Container-XXX**：适用于HSF应用。 本文的示例应用为Dubbo应用，因此选择**标准Java应用运行环境**。

参数名	说明
Java环境	您可以根据需要选择Java环境： ○ **openjdk-XXXXX-jdk-alpine3.9**：基于Alphine操作系统，基础镜像小，实例交付速度快。 ○ **Open JDK X**：基于CentOS操作系统，基础镜像大，实例交付速度慢。 ○ **Dragonwell X**：支持最新OpenJDK 8版本，支持应用启动及运行时加速，提升GC效率等。更多信息，请参见设置启动命令。 **说明** 基于Alphine操作系统的Java环境附带的工具或命令少于基于CentOS操作系统的Java环境，如果您选择了**openjdk-XXXXX-jdk-alpine3.9**，可能会导致Webshell中部分命令无法执行，请根据实际情况选择环境。
文件上传方式	可选择**上传JAR包**或**JAR包地址**。 ○ **上传JAR包**：单击**选择文件**，选择待部署的JAR文件。 ○ **JAR包地址**：输入JAR文件的存放地址。 示例提供者JAR文件的地址如下： https://aliware-images.oss-cn-hangzhou.aliyuncs.com/demo/sae/spring-boot-provider-1.0-SNAPSHOT.jar **注意** 应用部署程序包名称仅可包含字母、数字、短划线（-）和下划线（_）。
版本	设置应用版本号，您可以选择输入版本号或者单击**使用时间戳为版本号**将时间戳作为应用版本号。
时区设置	选择当前应用所在时区，例如**UTC+8**。

5. （可选）您可以根据需要设置以下高级选项。
   - **（可选）启动命令设置**：请参见设置启动命令。
   - **（可选）环境变量设置**：请参见设置环境变量。
   - **（可选）Hosts绑定设置**：请参见设置Hosts绑定。
   - **（可选）应用健康检查设置**：请参见设置健康检查。
   - **（可选）应用生命周期管理设置**：请参见设置应用生命周期管理。
   - **（可选）日志收集设置**：请参见设置日志收集。

- （可选）**持久化存储**：请参见设置NAS存储。
- （可选）**配置管理**：请参见注入配置信息。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及费用配置情况，并单击**确认创建**。

部署消费者微服务应用

部署消费者微服务应用的步骤与部署提供者微服务的步骤相似，请参见操作步骤，在步骤4**JAR包地址**中输入存放示例消费者JAR文件的地址：

```
https://aliware-images.oss-cn-hangzhou.aliyuncs.com/demo/sae/spring-boot-consumer-1.0-SNAPSHOT.jar
```

验证部署是否成功

您可以通过提供者和消费者的服务列表页面判断部署是否成功。

1. 登录SAE控制台。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
4. 在左侧导航栏选择**微服务治理 > 服务列表**，查看服务信息。
   - 提供者应用：如果**发布的服务**页签上可看到提供者所发布的服务，则说明提供者应用部署成功。

   - 如果**消费的服务**页签上可看到提供者所发布的服务，则说明消费者应用部署成功。

验证微服务调用

您可以为消费者应用设置SLB公网访问，并在浏览器中发起调用请求来判断微服务调用是否能正常工作。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择地域，单击消费者应用的名称。
3. 在**基本信息**页签的**应用访问设置**区域，单击**添加公网SLB访问**。
4. 在**添加公网SLB访问**对话框的**请选择SLB**下拉框中选择**新建**，并在**TCP协议**页签的**SLB端口**区域输入80，在**容器端口**区域输入9091，单击**确定**。

> **说明**
>
> - 使用SLB会产生额外费用，SLB的计费方式请参见SLB计费说明。
> - **容器端口**区域必须填写应用中设置的后端端口。本文示例应用的后端端口为9091。更多信息，请参见为应用绑定SLB。

等待2分钟~3分钟后，SLB实例将绑定完毕，**公网访问地址**字段后将显示公网SLB访问地址及端口。

5. 在浏览器地址栏中输入以下URL。

```
<公网SLB访问地址>:<端口>/sayHello/<自定义字符串>
```

例如输入 `http://*.***.169.144/sayHello/HelloSAE` 。

如果浏览器返回以下结果，则说明微服务调用成功。

```
Hello, HelloSAE (from Spring Boot)
```

查看应用监控指标

应用成功部署后，您可以在**应用监控**模块中查看应用的关键性能指标。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏按需查看应用监控数据：
   - **应用总览**：选择**应用监控 > 应用总览**，查看应用的关键性能指标，例如总请求量、平均响应时间等。
   - **应用详情**：选择**应用监控 > 应用详情**，查看更全面的应用监控数据，例如JVM监控、异常分析等。
   - **接口调用**：选择**应用监控 > 接口调用**，查看以接口为维度的监控数据，例如调用拓扑图、链路上游和链路下游等。
   - **高级监控**：选择**应用监控 > 高级监控**，查看更高性能的监控管理数据，例如应用全局拓扑、异常SQL分析和本地调用堆栈等。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.1.4. 在SAE控制台使用镜像方式部署Java应用

您可以通过SAE控制台使用上传至阿里云容器镜像服务的镜像来部署应用，支持的镜像包括您阿里云账号下的镜像、Demo镜像、可通过公网访问的公有镜像和其它阿里云账号私有镜像。本文以部署Java应用为例，介绍如何在SAE控制台使用镜像部署应用。

前提条件

- 创建VPC
- 创建命名空间
- 制作应用镜像

背景信息

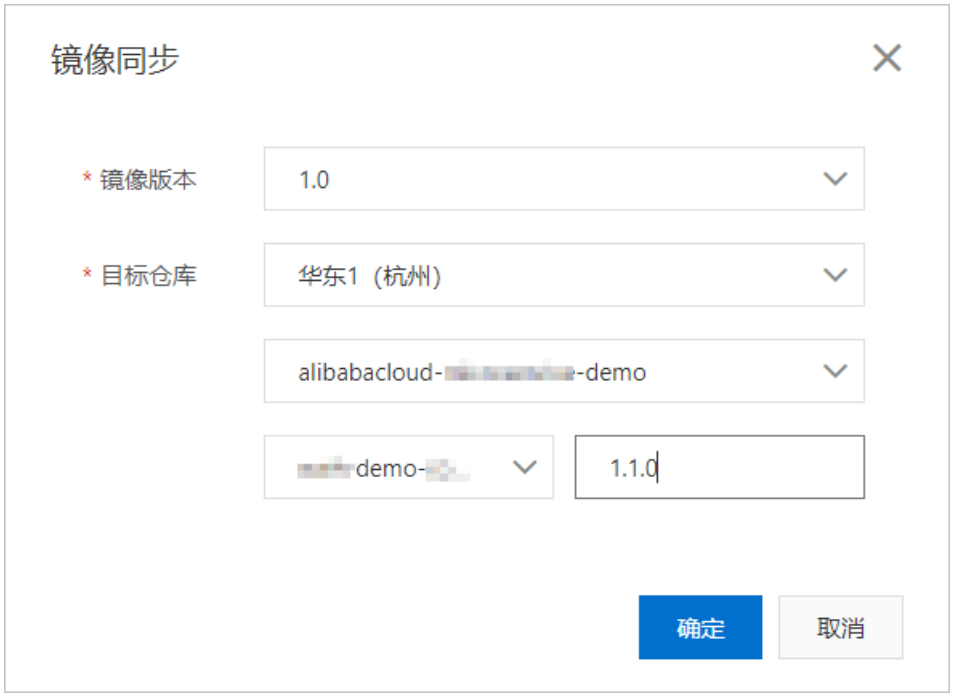
在SAE上部署应用的方式如下表所示。

应用举例	部署方式	参考文档
原生Spring Cloud	WAR、JAR、镜像	将Spring Cloud应用托管到SAE
原生Dubbo	WAR、JAR、镜像	将Dubbo应用托管到SAE
HSF	WAR、JAR、镜像	无
多语言应用	镜像	使用镜像部署多语言应用

可选：在容器镜像服务控制台配置镜像同步

如果要使用的阿里云镜像仓库与您的应用不在同一个地域，您需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域，再部署应用。

1. 登录容器镜像服务控制台。
2. 在顶部菜单栏，选择所需地域。
3. 在**实例列表**页面，单击**个人实例**，进入**个人实例**页面。
4. 在左侧导航栏选择**仓库管理 > 镜像仓库**，在**镜像仓库**页面单击目标仓库名称。
5. 在左侧导航栏单击**镜像同步**，在**镜像同步**页面左上角单击**创建同步任务**。
6. 在**镜像同步**对话框中输入以下信息并单击**确定**。

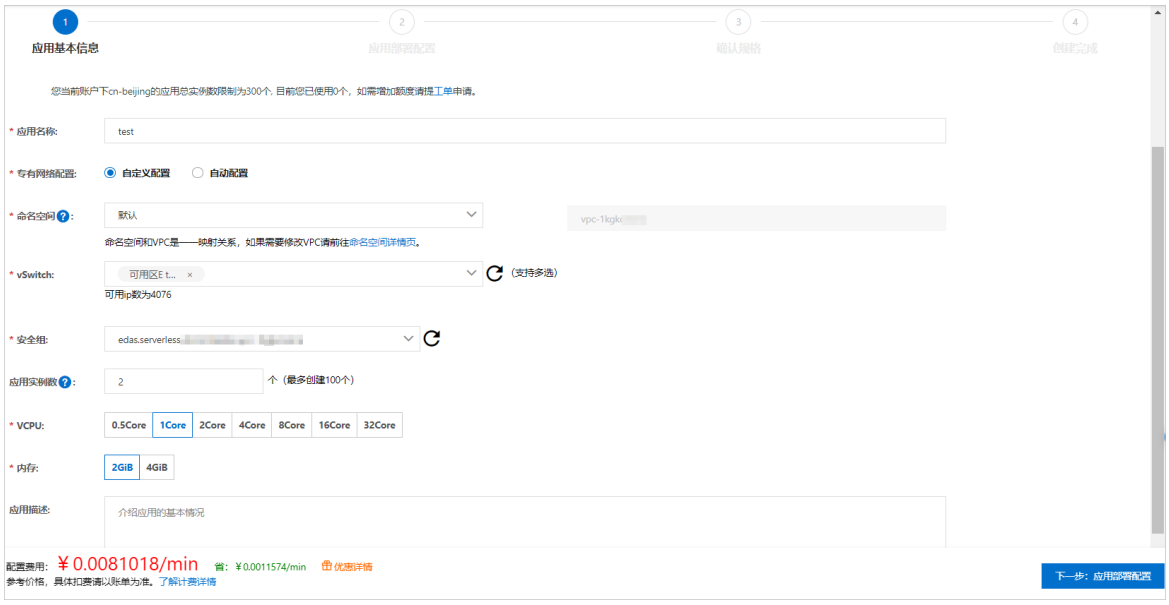


- **镜像版本**：当前镜像的版本。
- **目标仓库**：依次输入目标仓库的地域（即您的应用所在的地域）、命名空间、仓库名称和版本号。

镜像同步完成后，在容器镜像服务控制台上选择的镜像仓库将被同步至此处选择的目标仓库。随后您将可以在SAE控制台的**应用部署配置**页面上选择该镜像。

部署应用

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。



参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。
应用实例数	选择需要创建的实例个数。
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，配置相关参数。

参数说明如下所示。

参数名	说明
技术栈语言	SAE支持**Java**、**PHP**及**其它语言**。 本文选择**Java**。
应用部署方式	Java应用支持以下部署方式： ○ **镜像** ○ **WAR包部署** ○ **JAR包部署** 本文选择**镜像**。
	指定部署应用的镜像文件。 ○ **我的阿里云镜像**：选择您的阿里云账号下的且已上传至阿里云容器镜像服务的镜像文件及其版本。 ■ **镜像服务个人版**：通过**镜像仓库命名空间**下拉列表和**镜像仓库名**文本框来快速筛选目标镜像文件。 **说明** 如果要使用的阿里云镜像仓库与您的应用不在同一个地域，则需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域。具体操作，请参见可选：在容器镜像服务控制台配置镜像同步。

参数名	说明
	■ **镜像服务企业版**：在**镜像服务企业版**下拉列表选择目标企业版实例。
配置镜像	**说明** ■ 邀测体验中。如果您需要使用容器镜像服务企业版，请提交工单申请。 ■ 如果选择容器镜像服务企业版，您需要先在容器镜像服务控制台创建企业版实例并绑定VPC和vSwitch。具体操作，请参见创建企业版实例和配置专有网络的访问控制。 ■ 如果SAE应用的VPC和vSwitch未在容器镜像服务控制台绑定，您将无法在SAE控制台对应用进行部署。 ○ **Demo镜像**：在**请选择**下拉列表选择Demo镜像的版本。 ○ **公有镜像**：输入可通过公网访问的公共镜像的完整镜像地址。 **说明** 如需使用公有镜像，请确保VPC能够访问公网。 ○ **其它阿里云账号私有镜像**：输入**acrAssumeRoleArn**和**完整镜像仓库地址**。 **说明** 在RAM控制台获取acrAssumeRoleArn的具体操作，请参见跨账号拉取阿里云镜像。

5. （可选）您可以根据需要设置以下高级选项。
   - **（可选）启动命令设置**：请参见设置启动命令。
   - **（可选）环境变量设置**：请参见设置环境变量。
   - **（可选）Hosts绑定设置**：请参见设置Hosts绑定。
   - **（可选）应用健康检查设置**：请参见设置健康检查。
   - **（可选）应用生命周期管理设置**：请参见设置应用生命周期管理。
   - **（可选）日志收集设置**：请参见设置日志收集。
   - **（可选）持久化存储**：请参见设置NAS存储。
   - **（可选）配置管理**：请参见注入配置信息。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及费用配置情况，并单击**确认创建**。

结果验证

应用部署后，您可以通过以下两种方式验证应用的部署结果：

- 查看应用实例运行状态。
  i. 登录SAE控制台。
  ii. 在左侧导航栏单击**应用列表**，在**应用列表**页面单击具体应用。
  iii. 在**应用详情**页面的**实例部署信息**页签查看实例的运行状态。
  如果**运行状态**显示为绿色的Running或者Completed，表示应用部署成功。
- 配置公网负载均衡并访问应用。
  应用部署后，根据实际需要，通过配置负载均衡SLB在指定范围内开放应用访问，以便其它应用访问。
  负载均衡包括以下两种：
  - 私网负载均衡：在应用所在的VPC内提供应用的访问入口，保证应用能被同VPC内的其它应用访问。
  - 公网负载均衡：为该应用自动购买公网SLB服务，或者复用在SLB控制台已购买的实例，保证应用能被公网中的其它应用访问。

> **说明** 配置公网SLB访问和配置私网SLB访问的步骤相同，具体操作请参见为应用绑定SLB。SLB绑定完成后，在浏览器输入由SLB的IP地址、端口及应用部署包名组成的访问地址并访问，例如192.168.XX.XX/80/image，即可进入应用。

查看应用监控指标

应用成功部署后，您可以在**应用监控**模块中查看应用的关键性能指标。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏按需查看应用监控数据：
   - **应用总览**：选择**应用监控 > 应用总览**，查看应用的关键性能指标，例如总请求量、平均响应时间等。
   - **应用详情**：选择**应用监控 > 应用详情**，查看更全面的应用监控数据，例如JVM监控、异常分析等。
   - **接口调用**：选择**应用监控 > 接口调用**，查看以接口为维度的监控数据，例如调用拓扑图、链路上游和链路下游等。
   - **高级监控**：选择**应用监控 > 高级监控**，查看更高性能的监控管理数据，例如应用全局拓扑、异常SQL分析和本地调用堆栈等。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理

- 变更实例规格

- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.2. PHP应用

2.2.1. 在SAE控制台使用镜像部署PHP应用

您可以通过SAE控制台使用上传至阿里云容器镜像服务的镜像来部署应用，支持的镜像包括您阿里云账号下的镜像、Demo镜像、可通过公网访问的公有镜像和其它阿里云账号私有镜像。本文以部署PHP应用为例，介绍如何在SAE控制台使用镜像部署应用。

前提条件

- 创建VPC。
- 创建命名空间。
- 制作应用镜像。

背景信息

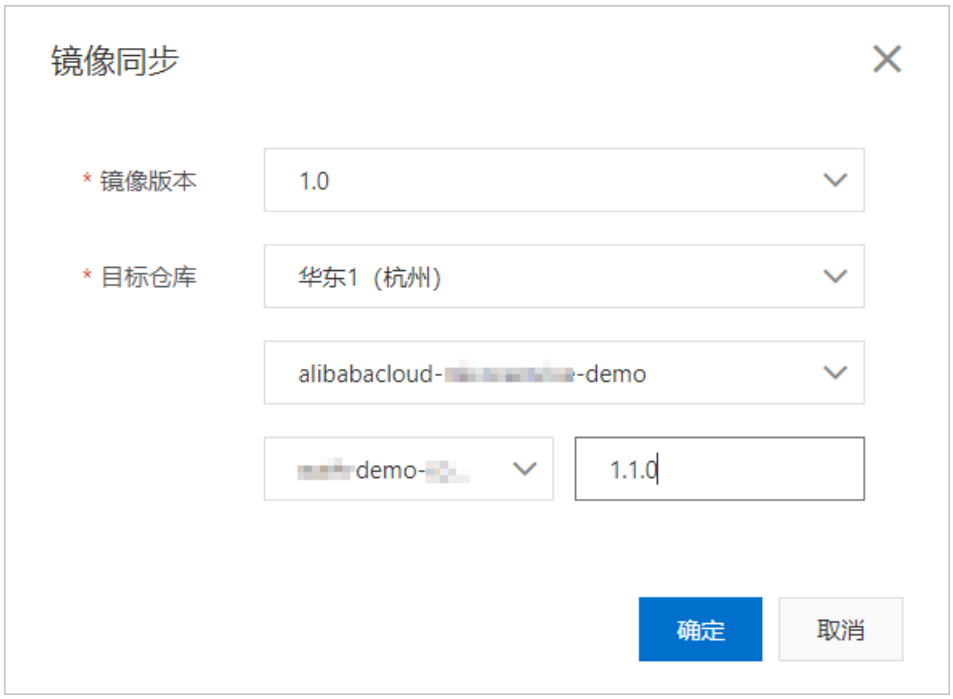
在SAE上部署应用的方式如下表所示。

应用举例	部署方式	参考文档
原生Spring Cloud	WAR、JAR、镜像	将Spring Cloud应用托管到SAE
原生Dubbo	WAR、JAR、镜像	将Dubbo应用托管到SAE
HSF	WAR、JAR、镜像	无
多语言应用	镜像	使用镜像部署多语言应用

可选：在容器镜像服务控制台配置镜像同步

如果要使用的阿里云镜像仓库与您的应用不在同一个地域，您需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域，再部署应用。

1. 登录容器镜像服务控制台。
2. 在顶部菜单栏，选择所需地域。
3. 在**实例列表**页面，单击**个人实例**，进入**个人实例**页面。
4. 在左侧导航栏选择**仓库管理 > 镜像仓库**，在**镜像仓库**页面单击目标仓库名称。
5. 在左侧导航栏单击**镜像同步**，在**镜像同步**页面左上角单击**创建同步任务**。
6. 在**镜像同步**对话框中输入以下信息并单击**确定**。

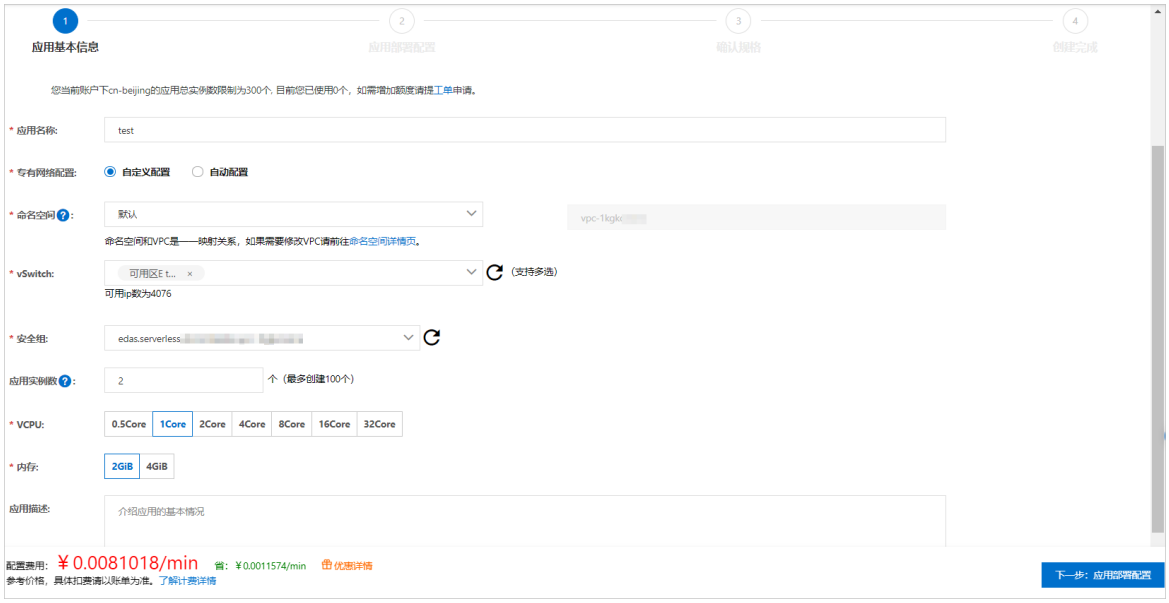


- **镜像版本**：当前镜像的版本。
- **目标仓库**：依次输入目标仓库的地域（即您的应用所在的地域）、命名空间、仓库名称和版本号。

镜像同步完成后，在容器镜像服务控制台上选择的镜像仓库将被同步至此处选择的目标仓库。随后您将可以在SAE控制台的**应用部署配置**页面上选择该镜像。

部署应用

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。

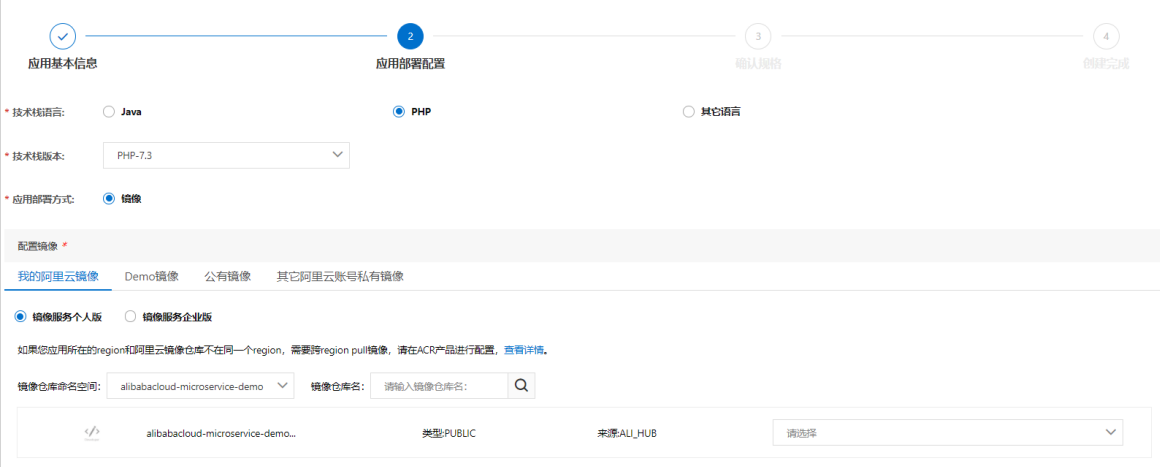


参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。
应用实例数	选择需要创建的实例个数。

参数名	说明
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，配置相关参数。



参数说明如下所示。

参数名	说明
技术栈语言	SAE支持**Java**、**PHP**及**其它语言**。 本文选择**PHP**。
技术栈版本	选择PHP应用的语言版本。仅**技术栈语言**为**PHP**时需要设置。 **注意** 请选择正确的**技术栈版本**，错误版本可能会导致PHP应用运行异常。
应用部署方式	PHP应用仅支持**镜像**部署方式。
	指定部署应用的镜像文件。 ○ **我的阿里云镜像**：选择您的阿里云账号下的且已上传至阿里云容器镜像服务的镜像文件及其版本。

参数名	说明
配置镜像	■ **镜像服务个人版**：通过**镜像仓库命名空间**下拉列表和**镜像仓库名**文本框来快速筛选目标镜像文件。 **说明** 如果要使用的阿里云镜像仓库与您的应用不在同一个地域，则需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域。具体操作，请参见可选：在容器镜像服务控制台配置镜像同步。 ■ **镜像服务企业版**：在**镜像服务企业版**下拉列表选择目标企业版实例。 **说明** ■ 邀测体验中。如果您需要使用容器镜像服务企业版，请提交工单申请。 ■ 如果选择容器镜像服务企业版，您需要先在容器镜像服务控制台创建企业版实例并绑定VPC和vSwitch。具体操作，请参见创建企业版实例和配置专有网络的访问控制。 ■ 如果SAE应用的VPC和vSwitch未在容器镜像服务控制台绑定，您将无法在SAE控制台对应用进行部署。 ○ **Demo镜像**：在**请选择**下拉列表选择Demo镜像的版本。 ○ **公有镜像**：输入可通过公网访问的公共镜像的完整镜像地址。 **说明** 如需使用公有镜像，请确保VPC能够访问公网。 ○ **其它阿里云账号私有镜像**：输入**acrAssumeRoleArn**和**完整镜像仓库地址**。 **说明** 在RAM控制台获取acrAssumeRoleArn的具体操作，请参见跨账号拉取阿里云镜像。

5. （可选）您可以根据需要设置以下高级选项。

高级设置	是否必选	参考文档
设置PHP应用监控	否	设置PHP应用监控 说明 仅技术栈语言为PHP时可以设置。
设置PHP应用配置文件	否	设置PHP应用配置文件 说明 仅技术栈语言为PHP时可以设置。
启动命令	否	设置启动命令
环境变量	否	设置环境变量
Hosts绑定	否	设置Hosts绑定
应用健康检查	否	设置健康检查
应用生命周期管理	否	设置应用生命周期管理
日志收集	否	设置日志收集
持久化存储	否	设置NAS存储
配置管理	否	注入配置信息

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及费用配置情况，并单击**确认创建**。

结果验证

应用部署后，您可以通过以下两种方式验证应用的部署结果：

- 查看应用实例运行状态。
  i. 登录SAE控制台。
  ii. 在左侧导航栏单击**应用列表**，在**应用列表**页面单击具体应用。
  iii. 在**应用详情**页面的**实例部署信息**页签查看实例的运行状态。
  如果**运行状态**显示为绿色的Running或者Completed，表示应用部署成功。
- 配置公网负载均衡并访问应用。
  应用部署后，根据实际需要，通过配置负载均衡SLB在指定范围内开放应用访问，以便其它应用访问。
  负载均衡包括以下两种：
  - 私网负载均衡：在应用所在的VPC内提供应用的访问入口，保证应用能被同VPC内的其它应用访问。
  - 公网负载均衡：为该应用自动购买公网SLB服务，或者复用在SLB控制台已购买的实例，保证应用能被公网中的其它应用访问。

> **说明** 配置公网SLB访问和配置私网SLB访问的步骤相同，具体操作请参见为应用绑定SLB。SLB绑定完成后，在浏览器输入由SLB的IP地址、端口及应用部署包名组成的访问地址并访问，例如192.168.XX.XX/80/image，即可进入应用。

查看应用监控指标

应用成功部署后，您可以在**应用监控**模块中查看应用的关键性能指标。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏按需查看应用监控数据：
   - **应用总览**：选择**应用监控 > 应用总览**，查看应用的关键性能指标，例如总请求量、平均响应时间等。
   - **应用详情**：选择**应用监控 > 应用详情**，查看更全面的应用监控数据，例如JVM监控、异常分析等。
   - **接口调用**：选择**应用监控 > 接口调用**，查看以接口为维度的监控数据，例如调用拓扑图、链路上游和链路下游等。
   - **高级监控**：选择**应用监控 > 高级监控**，查看更高性能的监控管理数据，例如应用全局拓扑、异常SQL分析和本地调用堆栈等。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.3. 使用镜像部署多语言应用

您可以通过SAE控制台使用上传至阿里云容器镜像服务的镜像来部署应用，支持的镜像包括您阿里云账号下的镜像、Demo镜像、可通过公网访问的公有镜像和其它阿里云账号私有镜像。本文介绍如何在SAE控制台使用镜像部署多语言应用。

前提条件

- 创建VPC
- 创建命名空间
- 制作应用镜像

背景信息

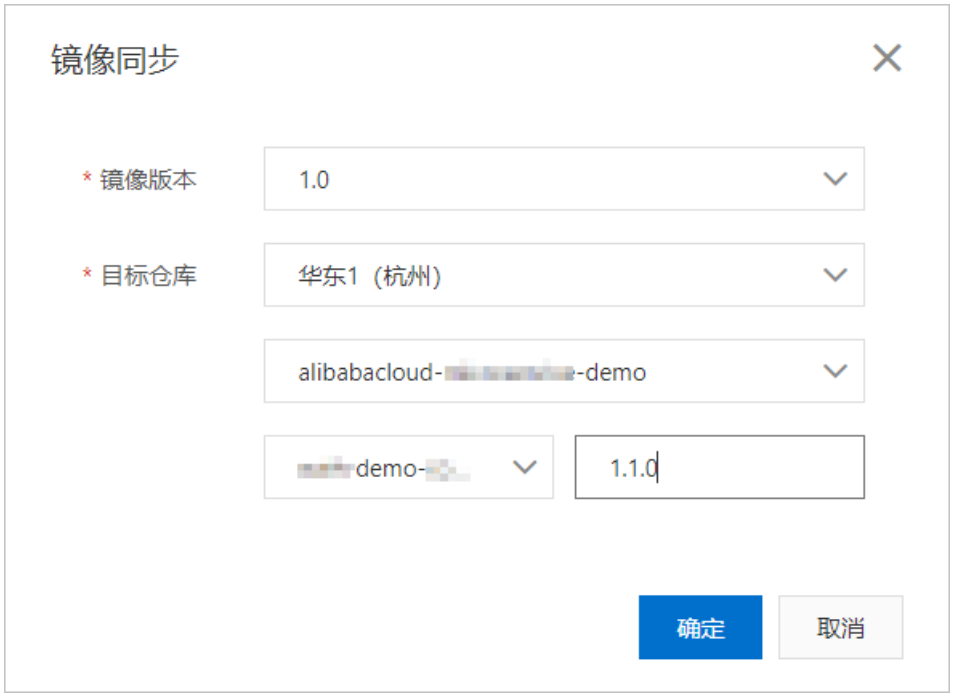
在SAE上部署应用的方式如下表所示。

应用举例	部署方式	参考文档
原生Spring Cloud	WAR、JAR、镜像	将Spring Cloud应用托管到SAE
原生Dubbo	WAR、JAR、镜像	将Dubbo应用托管到SAE
HSF	WAR、JAR、镜像	无
多语言应用	镜像	使用镜像部署多语言应用

可选：在容器镜像服务控制台配置镜像同步

如果要使用的阿里云镜像仓库与您的应用不在同一个地域，您需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域，再部署应用。

1. 登录容器镜像服务控制台。
2. 在顶部菜单栏，选择所需地域。
3. 在**实例列表**页面，单击**个人实例**，进入**个人实例**页面。
4. 在左侧导航栏选择**仓库管理 > 镜像仓库**，在**镜像仓库**页面单击目标仓库名称。
5. 在左侧导航栏单击**镜像同步**，在**镜像同步**页面左上角单击**创建同步任务**。
6. 在**镜像同步**对话框中输入以下信息并单击**确定**。

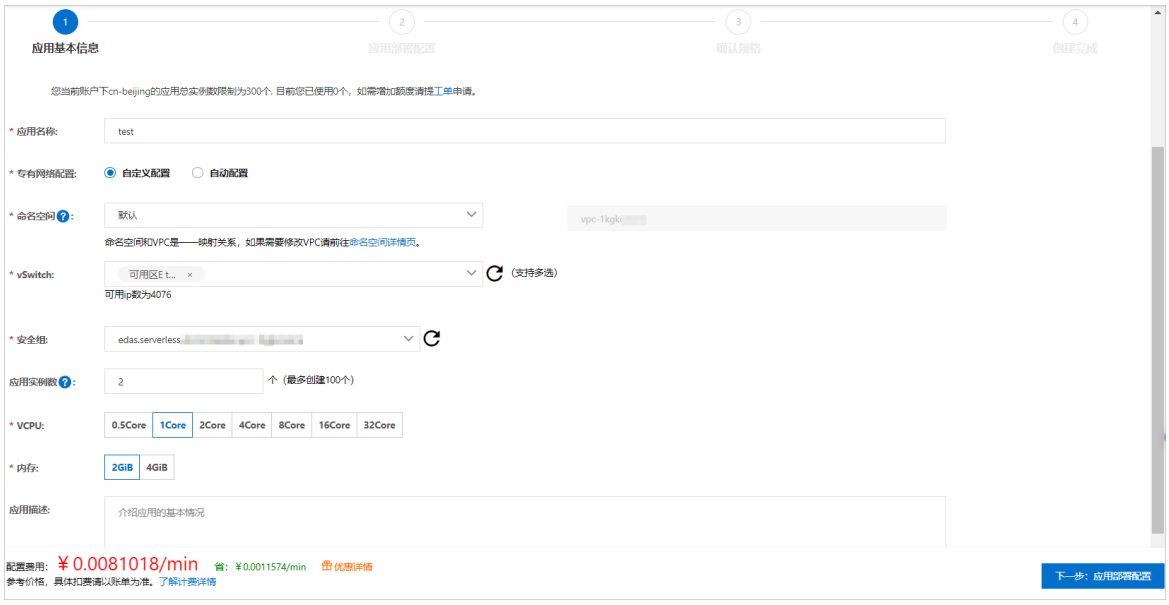


- **镜像版本**：当前镜像的版本。
- **目标仓库**：依次输入目标仓库的地域（即您的应用所在的地域）、命名空间、仓库名称和版本号。

镜像同步完成后，在容器镜像服务控制台上选择的镜像仓库将被同步至此处选择的目标仓库。随后您将可以在SAE控制台的**应用部署配置**页面上选择该镜像。

部署应用

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。



参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。
应用实例数	选择需要创建的实例个数。
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，配置相关参数。

参数说明如下所示。

参数名	说明
技术栈语言	SAE支持**Java**、**PHP**及**其它语言**。 本文选择**其它语言**，例如Go、Python、C++等语言。
应用部署方式	其它语言的应用仅支持**镜像**部署方式。

参数名	说明
配置镜像	指定部署应用的镜像文件。 ○ **我的阿里云镜像**：选择您的阿里云账号下的且已上传至阿里云容器镜像服务的镜像文件及其版本。 ■ **镜像服务个人版**：通过**镜像仓库命名空间**下拉列表和**镜像仓库名**文本框来快速筛选目标镜像文件。 说明 如果要使用的阿里云镜像仓库与您的应用不在同一个地域，则需要先使用容器镜像服务ACR的镜像同步功能将该镜像仓库同步至应用所在的地域。具体操作，请参见可选：在容器镜像服务控制台配置镜像同步。 ■ **镜像服务企业版**：在**镜像服务企业版**下拉列表选择目标企业版实例。 说明 ■ 邀测体验中。如果您需要使用容器镜像服务企业版，请提交工单申请。 ■ 如果选择容器镜像服务企业版，您需要先在容器镜像服务控制台创建企业版实例并绑定VPC和vSwitch。具体操作，请参见创建企业版实例和配置专有网络的访问控制。 ■ 如果SAE应用的VPC和vSwitch未在容器镜像服务控制台绑定，您将无法在SAE控制台对应用进行部署。 ○ **Demo镜像**：在**请选择**下拉列表选择Demo镜像的版本。 ○ **公有镜像**：输入可通过公网访问的公共镜像的完整镜像地址。 说明 如需使用公有镜像，请确保VPC能够访问公网。 ○ **其它阿里云账号私有镜像**：输入**acrAssumeRoleArn**和**完整镜像仓库地址**。 说明 在RAM控制台获取acrAssumeRoleArn的具体操作，请参见跨账号拉取阿里云镜像。

5. （可选）您可以根据需要设置以下高级选项。

高级设置	是否必选	参考文档
设置PHP应用监控	否	设置PHP应用监控 说明 仅技术栈语言为PHP时可以设置。
设置PHP应用配置文件	否	设置PHP应用配置文件 说明 仅技术栈语言为PHP时可以设置。
启动命令	否	设置启动命令
环境变量	否	设置环境变量
Hosts绑定	否	设置Hosts绑定
应用健康检查	否	设置健康检查
应用生命周期管理	否	设置应用生命周期管理
日志收集	否	设置日志收集
持久化存储	否	设置NAS存储
配置管理	否	注入配置信息

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及费用配置情况，并单击**确认创建**。

结果验证

应用部署后，您可以通过以下两种方式验证应用的部署结果：

- 查看应用实例运行状态。
  i. 登录SAE控制台。
  ii. 在左侧导航栏单击**应用列表**，在**应用列表**页面单击具体应用。
  iii. 在**应用详情**页面的**实例部署信息**页签查看实例的运行状态。
  如果**运行状态**显示为绿色的Running或者Completed，表示应用部署成功。
- 配置公网负载均衡并访问应用。
  应用部署后，根据实际需要，通过配置负载均衡SLB在指定范围内开放应用访问，以便其它应用访问。
  负载均衡包括以下两种：
  - 私网负载均衡：在应用所在的VPC内提供应用的访问入口，保证应用能被同VPC内的其它应用访问。
  - 公网负载均衡：为该应用自动购买公网SLB服务，或者复用在SLB控制台已购买的实例，保证应用能被公网中的其它应用访问。

> **说明** 配置公网SLB访问和配置私网SLB访问的步骤相同，具体操作请参见为应用绑定SLB。SLB绑定完成后，在浏览器输入由SLB的IP地址、端口及应用部署包名组成的访问地址并访问，例如192.168.XX.XX/80/image，即可进入应用。

查看应用监控指标

应用成功部署后，您可以在**应用监控**模块中查看应用的关键性能指标。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏按需查看应用监控数据：
   - **应用总览**：选择**应用监控 > 应用总览**，查看应用的关键性能指标，例如总请求量、平均响应时间等。
   - **应用详情**：选择**应用监控 > 应用详情**，查看更全面的应用监控数据，例如JVM监控、异常分析等。
   - **接口调用**：选择**应用监控 > 接口调用**，查看以接口为维度的监控数据，例如调用拓扑图、链路上游和链路下游等。
   - **高级监控**：选择**应用监控 > 高级监控**，查看更高性能的监控管理数据，例如应用全局拓扑、异常SQL分析和本地调用堆栈等。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.4. 灰度发布应用

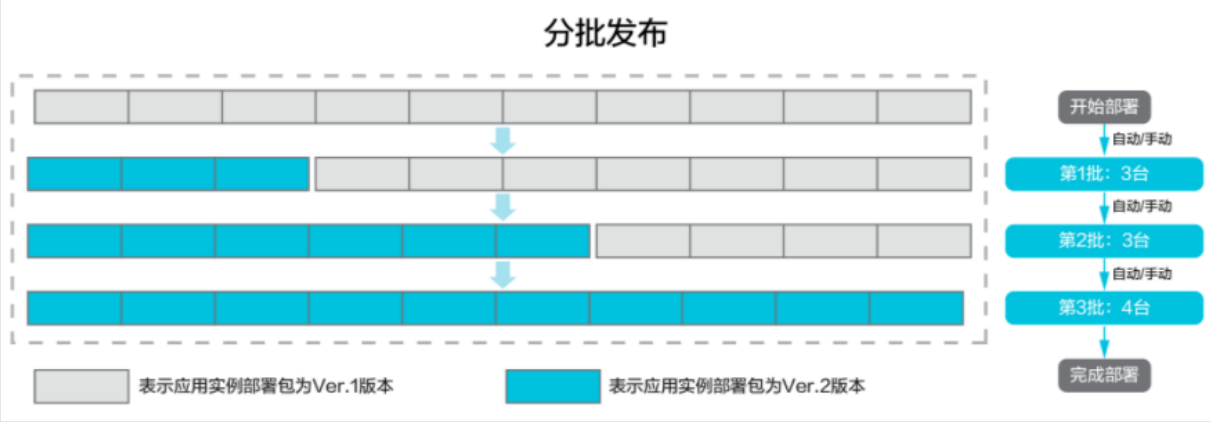
如果您的应用有多个应用实例且需要升级，那么您可以按需通过灰度或分批发布方式升级应用实例版本。本文介绍灰度发布方式的背景信息、场景示例和操作步骤；此外，还涉及应用回滚的说明。

背景信息

- 灰度发布：又名金丝雀发布，在原有部署版本可用的情况下，同时部署新版本应用作为金丝雀，测试新版本的性能，在保证整体系统稳定的情况下，帮助您尽早发现问题和修复问题。
  应用进行灰度发布时，为了保证应用稳定性，灰度发布的应用实例数不能超过应用实例总数的50%，剩余的应用实例按照指定的批次分批发布。
- 分批发布：按批次进行应用部署，每次仅对应用的一部分实例进行升级。分批发布过程中如果出现故障，则可以及时终止并回退，待问题修复后重新发布。
  应用进行分批发布时，应用内的实例数会平均分配到每个批次进行部署。如果无法平均分配，则靠前的批次分配到的实例数少，靠后的批次分配到的实例数多。更多信息，请参见分批发布应用。

场景示例

某应用包含10个应用实例，每个应用实例的部署版本为Ver.1版本，现需将每个应用实例升级为Ver.2版本。假设将所有应用实例分3批进行部署，灰度发布流程如下图所示。

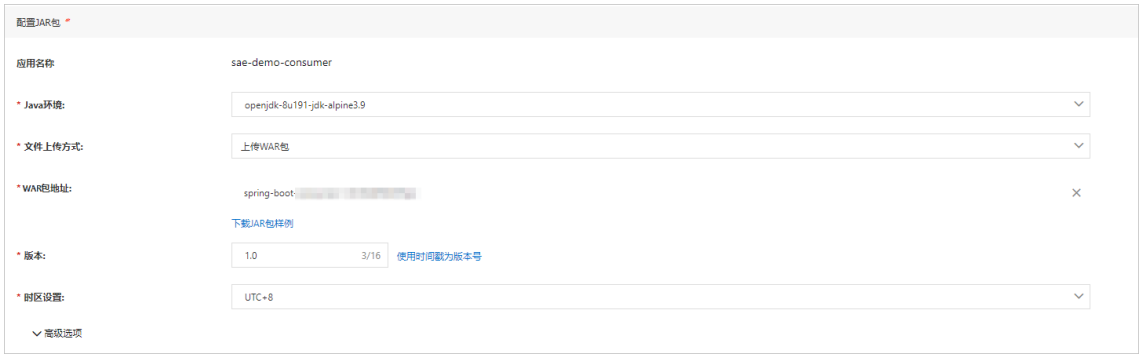


操作步骤

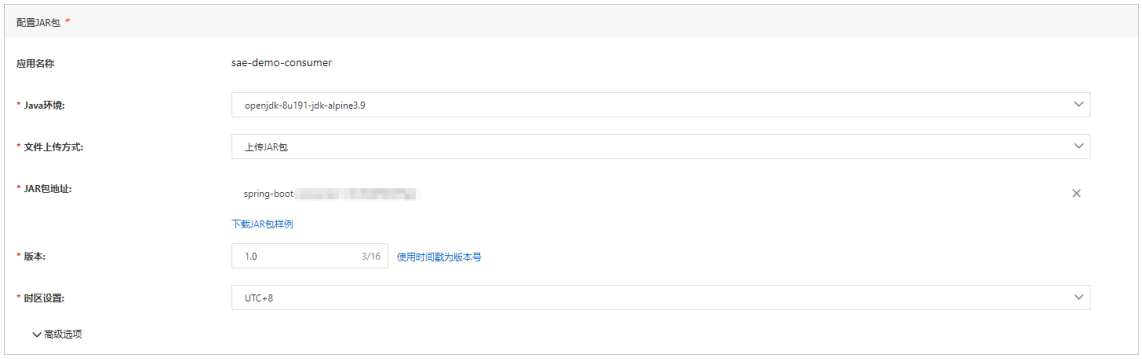
1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 配置部署参数。

> **说明** 部署方式由应用首次部署方式决定，请根据所需的部署方式设置参数。

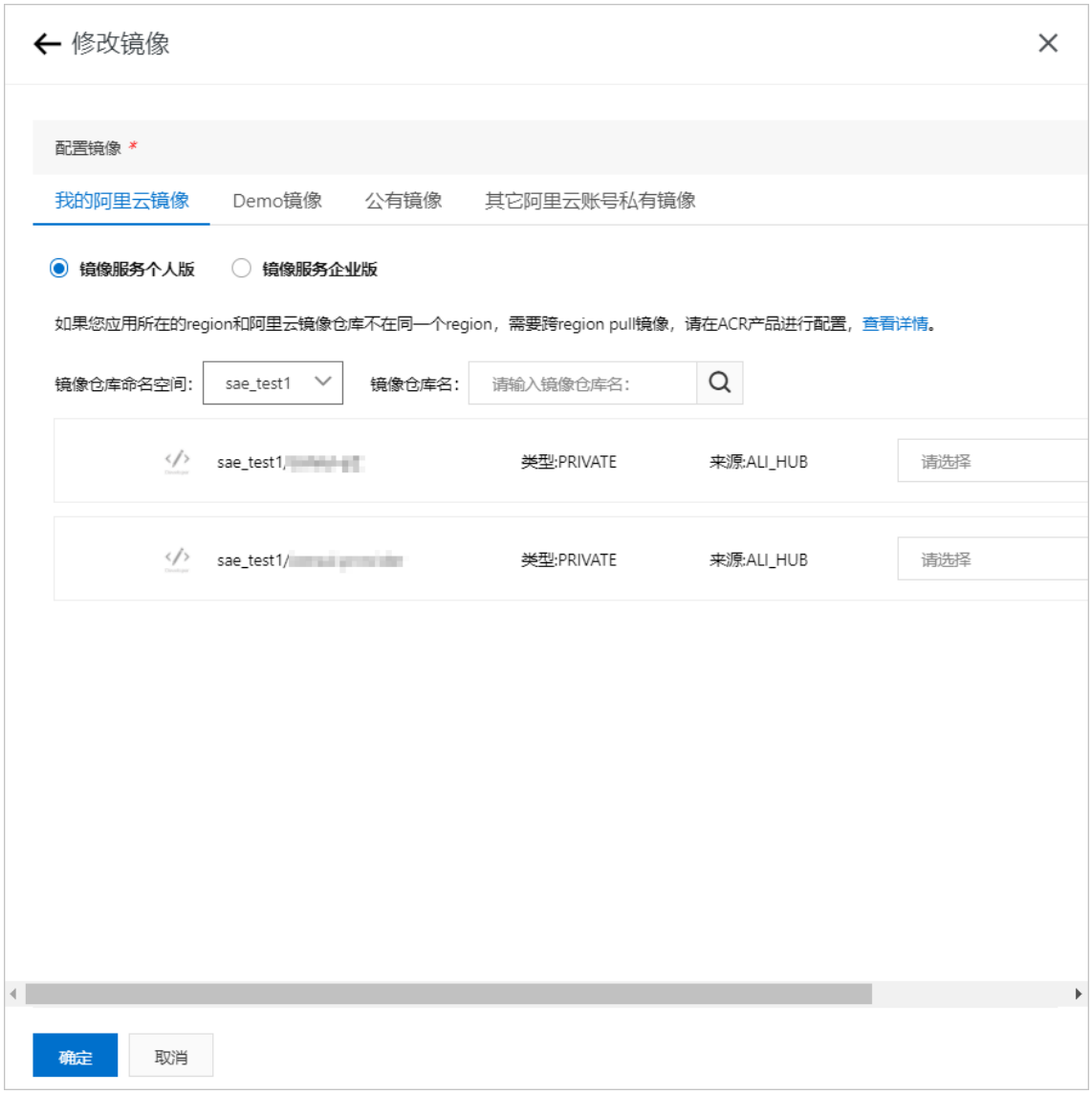
- WAR包部署：重新上传WAR包或者输入新部署WAR包的地址，并完成相关环境和参数设置。



- JAR包部署：重新上传JAR包或者输入新部署JAR包的地址，并完成相关环境和参数设置。



- 镜像：在**部署应用**区域单击**修改镜像**，在**修改镜像**面板重新选择镜像仓库或镜像版本。



5. 在**发布策略设置**区域内配置灰度发布。

灰度发布参数说明如下。

配置	是否必选	说明
发布策略	是	选择**灰度发布**。
灰度数量	是	设置首先需要进行灰度发布的应用实例数量。
灰度后剩余批次	是	灰度发布后，剩余的应用实例按照设定的批次完成发布。
最小存活实例数	是	每次滚动升级最小存活的实例数。选择**使用系统推荐值**后SAE将根据您的需求为应用设置最佳的最小存活实例数。
启用微服务灰度规则	否	您为Spring Cloud或Dubbo应用创建的灰度规则。具体操作，请参见管理灰度规则（Java）。

6. （可选）高级配置。

高级设置	是否必选	参考文档
设置PHP应用监控	否	设置PHP应用监控 说明 仅技术栈语言为PHP时可以设置。
设置PHP应用配置文件	否	设置PHP应用配置文件 说明 仅技术栈语言为PHP时可以设置。
启动命令	否	设置启动命令
环境变量	否	设置环境变量
Hosts绑定	否	设置Hosts绑定
应用健康检查	否	设置健康检查
应用生命周期管理	否	设置应用生命周期管理
日志收集	否	设置日志收集
持久化存储	否	设置NAS存储
配置管理	否	注入配置信息

7. 单击**确认**完成发布设置。
8. 您可以选择以下任一方式验证发布结果。
   - 方法一：在应用的**变更记录**页面中查看应用变更详情，查看发布状态，如果所有批次都执行成功，则说明应用更新成功。
   - 方法二：在应用**基本信息**页面的**实例部署信息**页签查看实例的运行状态。如果**执行状态**显示为**Running**，且实例的版本已变为Ver.2，表示应用部署成功。

应用回滚

采用灰度发布或者分批发布方式升级应用实例时，如果应用实例中存在未完成升级的实例，则当前应用升级状态处于**进行中**。

在实时跟踪应用升级时，如果首批应用实例升级突发异常停止响应，为了保证业务不受影响，请在**变更详情**页面中单击**立即回滚**，将已升级的实例回退至升级前版本，并将配置还原为升级前原有配置。

在应用变更过程中，如果出现部署包不可用、健康检查失败等变更流程异常情况，将导致应用升级失败，SAE将停止当前应用并进行回退。

在SAE上进行应用升级耗时约为30分钟，超出该时间后SAE将上报超时异常并暂停变更流程，请在**变更详情**页面手动终止发布流程并回滚应用。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

2.5. 管理灰度规则（Java）

对于部署在SAE的Spring Cloud或Dubbo微服务应用，为了确保升级操作的安全性，您可以通过启用灰度发布（即金丝雀发布）的灰度规则进行小规模验证，验证通过后再全量升级。本文介绍如何在SAE控制台上创建、查看、编辑和删除灰度规则。

前提条件

您已在SAE创建并部署Spring Cloud或Dubbo微服务应用。具体操作，请参见以下文档：

- 使用Spring Cloud开发微服务应用并部署至SAE
- 使用Spring Boot开发Dubbo应用并部署至SAE

背景信息

如果您的应用有多个应用实例且需要升级，那么您可以按需通过灰度或分批发布方式升级应用实例版本。其中，针对Spring Cloud和Dubbo微服务的灰度发布需要启动灰度规则才能实现。

更多信息，请参见灰度发布应用。

创建灰度规则

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏，选择**微服务治理 > 灰度规则**。
4. 在**灰度规则**页面，单击**新建灰度规则**。
5. 在**新建灰度规则**面板，执行以下操作，然后单击**确定**。

   SAE支持的灰度规则是按内容灰度，您所需配置的参数会因应用框架而异。

i. 按需配置以下参数。

参数	说明
规则名称	灰度规则的名称。支持以小写字母开头，以数字或小写字母结尾，仅可包含小写字母、中文字符、数字及短划线（-）。不超过64个字符。
规则描述	灰度规则的描述信息。不超过64个字符。
框架类型	配置灰度规则应用的框架类型，支持以下框架： ■ Spring Cloud：需要设置Path。 ■ Dubbo：需要选择**服务方法**。
条件模式	选择**同时满足下列条件**或**满足下列任一条件**。

ii. 单击**添加新的规则条件**，按需配置灰度规则条件。

参数	说明
条件列表	■ Spring Cloud：根据Cookie、Header或Parameter参数类型，设置相应的**参数**、**条件**以及**值**。 ■ Dubbo：根据应用实际情况，设置**参数**、**参数值获取表达式**、**条件**以及**值**。

iii. （可选）单击**新建流量规则**创建多个入口流量规则，多个规则可以同时生效。

Spring Cloud应用灰度规则示例

创建成功后，您将在**灰度规则**页面查看到刚创建的灰度规则。同时，**新建灰度规则**按钮已置灰。

查看规则详情

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏，选择**微服务治理 > 灰度规则**。
4. 在**灰度规则**页面，找到您需查看的灰度规则，单击规则名称。
5. 在**灰度规则详情**面板，查看规则内容。

编辑灰度规则

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏，选择**微服务治理 > 灰度规则**。
4. 在**灰度规则**页面，找到您需编辑的灰度规则，在其**操作**列，单击**编辑**。
5. 在**编辑灰度规则**面板，按需更新规则内容。
6. 单击**确定**。

删除灰度规则

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在左侧导航栏，选择**微服务治理 > 灰度规则**。
4. 在**灰度规则**页面，找到您需删除的灰度规则，在其**操作**列，单击**删除**。
5. 在弹出的**提示**对话框，单击**确认**。

2.6. 多语言应用

3.插件部署

3.1. 通过Maven插件自动部署应用

您可以通过Maven的*toolkit-maven-plugin*插件完成SAE应用的自动化部署。本文介绍通过Maven自动化部署应用的操作步骤及典型场景示例。

前提条件

- 已下载并安装Maven。更多信息，请参见Maven。
- 已在SAE控制台上部署应用。更多信息，请参见以下文档：
  - 在SAE控制台使用WAR包部署Java Web应用
  - 在SAE控制台使用JAR包部署微服务应用
  - 在SAE控制台使用镜像部署PHP应用

背景信息

Cloud Toolkit的更多信息，请参见什么是Alibaba Cloud Toolkit。

典型场景示例

本章节介绍通过Maven插件自动部署应用的典型部署场景及相关配置文件示例。关于自动部署应用的操作步骤，请参见操作步骤。

- 场景一：本地构建WAR或FatJAR包并部署
  例如您在北京环境有WAR类型的应用，期望在本地使用WAR包部署应用。文件打包和部署配置如下。
  - 打包文件

```
apiVersion: V1
kind: AppPackage
spec:
 packageType: War
```

  - 部署文件

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appId:     #部署应用的ID。如果配置了该参数则无需配置namespaceId和appName。
  namespaceId: #应用所属命名空间ID。如果您不清楚应用ID，可使用应用所属命名空间及应用名称进行部署。
  appName:    #应用名称。如果您不清楚应用ID，可使用应用名称及命名空间进行部署。
```

- 场景二：使用已有镜像地址部署镜像类型应用
  例如您在北京环境有一个镜像类型应用，期望使用已有的镜像部署应用。文件打包和部署配置如下。

- 打包文件

```
apiVersion: V1
kind: AppPackage
spec:
 packageType: Image
 imageUrl: registry.cn-beijing.aliyuncs.com/test/gateway:latest  #镜像地址。
 acrInstanceId: #ACREE实例ID，如果需要使用ACREE才填写。1.1.7新版本支持的功能。
```

- 部署文件

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appId:     #部署应用的ID。如果配置了该参数则无需配置namespaceId和appName。
  namespaceId: #应用所属命名空间ID。如果您不清楚应用ID，可使用应用所属命名空间及应用名称进行部署。
  appName:    #应用名称。如果您不清楚应用ID，可使用应用名称及命名空间进行部署。
```

- 场景三：本地构建镜像上传至仓库并部署应用
  例如您在北京环境有镜像类型应用，期望在本地编译并构建为镜像，然后上传到阿里云镜像仓库部署应用。文件打包和部署配置如下。
  - 打包文件

```
apiVersion: V1
kind: AppPackage
spec:
 packageType: Image
 build:
  docker:
   dockerfile: Dockerfile #指定Dockerfile。
   imageRepoAddress:    #镜像仓库地址。
   imageTag:          #镜像Tag。
   imageRepoUser:      #镜像仓库用户名。
   imageRepoPassword:   #镜像仓库密码。
 acrInstanceId: #ACREE实例ID，如果需要使用ACREE才填写。1.1.7新版本支持的功能。
```

  - 部署文件

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appId:     #部署应用的ID。如果配置了该参数则无需配置namespaceId和appName。
  namespaceId: #应用所属命名空间ID。如果您不清楚应用ID，可使用应用所属命名空间及应用名称进行部署。
  appName:    #应用名称。如果您不清楚应用ID，可使用应用名称及命名空间进行部署。
```

操作步骤

1. 在您的打包工程的*pom.xml*文件中添加插件依赖。

```
<build>
  <plugins>
    <plugin>
      <groupId>com.alibaba.cloud</groupId>
      <artifactId>toolkit-maven-plugin</artifactId>
      <version>1.1.7</version>
    </plugin>
  </plugins>
</build>
```

> **说明** 建议您使用1.1.7版本的toolkit-maven-plugin。

2. 配置插件信息。

   配置插件信息主要包括配置账号、打包和部署信息。更多信息，请参见更多配置项。

   i. 配置账号信息。

   在打包工程的根目录下创建文件格式为 `YAML` 的账号配置文件，命名为 `toolkit_profile.yaml` 并输入以下信息。

```
regionId:     #应用所在地域。
accessKeyId:   #访问阿里云资源的AccessKey ID，建议您使用RAM用户的AccessKey ID降低安全风险。
accessKeySecret: #访问阿里云资源的AccessKey Secret，建议您使用RAM用户的AccessKey Secret降低安全风险。
```

   ii. 配置打包信息。

   在打包工程的根目录下创建文件格式为 `YAML` 的打包配置文件。如果打包工程为Maven的子模块，则需要在子模块的目录下创建该文件，并命名为 `toolkit_package.yaml` ，输入以下信息。

```
apiVersion: V1
kind: AppPackage
spec:
 packageType:  #应用部署包类型，支持War、FatJar、Image、url。您只有在该处配置了url，那么packageUrl才能生效。
 packageUrl:   #部署包地址，War、FatJar类型应用可配置。不填则使用当前Maven构建的包进行部署。
 imageUrl:    #如果部署包类型为Image，可配置此字段。Image类型也可以在本地构建Docker镜像进行部署，更多信息，请参见更多配置项。
 acrInstanceId: #ACREE实例ID，如果需要使用ACREE才填写。1.1.7新版本支持的功能。
```

   iii. 配置部署信息。

   在打包工程的根目录下创建文件格式为 `YAML` 的部署文件，命名为 `toolkit_deploy.yaml` ，并输入以下信息。

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appId:     #部署应用的ID。如果配置了该参数则无需配置namespaceId和appName。
  namespaceId: #应用所属命名空间ID。如果您不清楚应用ID，可使用应用所属命名空间及应用名称进行部署。
  appName:    #应用名称。如果您不清楚应用ID，可使用应用名称及命名空间进行部署。
```

3. 进入*pom.xml*所在的目录，执行以下命令构建部署工程。

> **注意** 如果部署的是Maven子模块，则需要进入子模块*pom.xml*文件所在的目录执行部署工程。

```
mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml
```

命令参数说明如下。

参数	说明
toolkit:deploy	打包完成后进行应用部署。
-Dtoolkit_profile	指定账号配置文件。如果账号文件与*pom.xml*在同一个目录下，且名字为 `.toolkit_profile.yaml` ，可不填写该参数，插件会自动获取。 **注意** 插件自动获取的 `.toolkit_profile.yaml` 文件名最前面包含英文句号。
-Dtoolkit_package	指定打包文件。如果打包文件与*pom.xml*在同一个目录下，且名字为 `.toolkit_package.yaml` ，可不填写该参数，插件会自动获取。 **注意** 插件自动获取的 `.toolkit_package.yaml` 文件名最前面包含英文句号。
-Dtoolkit_deploy	指定部署文件。如果部署文件与*pom.xml*在同一个目录下，且名字为 `.toolkit_deploy.yaml` ，可不填写该参数，插件会自动获取。 **注意** 插件自动获取的 `.toolkit_deploy.yaml` 文件名最前面包含英文句号。
-Ddeploy version	指定部署的版本号，优先级高于 `toolkit_deploy.yaml` 文件中的Version配置。 **说明** *toolkit-maven-plugin*插件1.1.2及以上版本支持该配置参数。

执行该打包命令后，系统显示如下结果，当返回信息中显示 `BUILD SUCCESS` 表示部署成功。

```
D:\Demo\[redacted]>mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml
[INFO] Scanning for projects...
[INFO]
[INFO] --------------< com.aliware.edas:eureka-service-provider >--------------
[INFO] Building eureka-service-provider 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ eureka-service-provider ---
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ eureka-service-provider ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ eureka-service-provider ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to D:\Demo\[redacted]\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ eureka-service-provider ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Demo\[redacted]\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ eureka-service-provider ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ eureka-service-provider ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:3.1.1:jar (default-jar) @ eureka-service-provider ---
[INFO] Building jar: D:\Demo\[redacted]\target\eureka-service-provider-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- toolkit-maven-plugin:1.1.7:deploy (default-cli) @ eureka-service-provider ---
[WARNING] Product settings only support edas and sae, using edas as default
[INFO] Image url is set, skip building image.
[INFO] Begin to trace change order: 9f226035-2d75-4e8c-a32a-[redacted] currently trace interval is 2 second(s).
[INFO] PipelineName:Batch 1 Change, PipelineId:b2446e8b-8957-4dfe-bb83-[redacted]
[INFO] Waiting...
[INFO] Waiting...
[INFO] Waiting...
[INFO] Waiting...
[INFO] Waiting...
[INFO] Deploy application successfully!
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  22.104 s
[INFO] Finished at: 2021-08-26T22:01:56+08:00
[INFO] ------------------------------------------------------------------------
```

更多配置项

- 打包参数
  打包文件支持的参数如下所示。

```
apiVersion: V1
kind: AppPackage
spec:
 packageType:  #应用部署包类型，支持War、FatJar、Image、url。您只有在该处配置了url，那么packageUrl
才能生效。
 imageUrl:   #镜像地址，Image包类型应用可设置。如果设置了spec.build.docker，使用本地构建的镜像部署，
则不需要设置该参数。
 acrInstanceId: #ACREE实例ID，如果需要使用ACREE才填写。1.1.7新版本支持的功能。
 packageUrl:  #部署包地址，War、FatJar类型应用可填入。
 build:
  docker:
   dockerfile:    #Docker镜像构建文件。如果您希望在本地构建镜像部署，需填入此字段，例如：Dockerfile。
   imageRepoAddress: #阿里云镜像仓库地址。如果您希望在本地构建镜像部署，需填入此字段，例如：registry
.cn-beijing.aliyuncs.com/edas_demo/demo。
   imageTag:     #镜像Tag。如果您希望在本地构建镜像部署，需填入此字段，例如：test。
   imageRepoUser:  #阿里云镜像仓库用户名。如果您希望在本地构建镜像部署，需填入此字段，例如：***@***
。
   imageRepoPassword: #阿里云镜像仓库密码。如果您希望在本地构建镜像部署，需填入此字段，例如：passw
ord。
  oss:
   bucket:      #目标存储桶名称。如果您希望使用自定义的OSS仓库存储部署包，需填入此字段，例如：bucket
-name01。
   key:        #OSS自定义路径。如果您希望使用自定义的OSS仓库存储部署包，需填入此字段，例如：test1/te
st.jpg。
   accessKeyId:    #OSS账号。如果您希望使用自定义的OSS仓库存储包，需填入此字段。
   accessKeySecret:  #OSS密码。如果您希望使用自定义的OSS仓库存储包，可填入此字段。
   useVpcEndpoint: true #是否通过内网上传文件，true表示通过内网上传，false表示不通过内网上传。
   accessTimeout:30  #部署包临时访问地址的过期时间，单位分钟，默认为30。如果您希望使用自定义的过期时
间，可修改该字段。
```

- 部署参数
  部署文件当前支持的完整参数如下所示。

> **说明**
> - 如果您有新的参数需要支持，请提交工单或者扫描本文下方的二维码或搜索群号23198618，加入钉钉群与我们反馈。
> - 以下部分参数的使用需要1.1.7版本的toolkit-maven-plugin插件支持。为了能够完整地使用以下参数部署应用，建议您将插件升级到1.1.7版本。

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appName:  #应用名称。
  namespaceId: #应用所在命名空间。
  appId:   #应用ID。插件会使用此应用进行部署，如果未填入则使用namespaceId和appname来查找应用进行部
署。
 version:   #部署版本号，默认使用日时分秒格式。
 jdk:     #部署的包依赖的JDK版本，支持以下6个选项：Open JDK 8、Open JDK 7、Dragonwell 11、Dragonw
ell 8、openjdk-8u191-jdk-alpine3.9、openjdk-7u201-jdk-alpine3.9。镜像不支持。
 webContainer: #部署的包依赖的Tomcat版本，支持以下2个选项：apache-tomcat-7.0.91、apache-tomcat-8.
```

```
5.42。
 batchWaitTime: #分批等待时间。
 command:    #镜像启动命令。该命令必须为容器内存在的可执行的对象。例如：sleep。设置该命令将导致镜像原本的启动命令失效。
 commandArgs:  #镜像启动命令参数。上述启动命令所需参数。
 - 1d
 updateStrategy:
  type: GrayBatchUpdate #部署类型，BatchUpdate表示分批部署，GrayBatchUpdate表示灰度部署。
  batchUpdate:
   batch: 2 #分批数，如果是灰度部署，表示灰度批次后的分批数。
   releaseType: manual #分批类型。manual表示手动分批，auto表示自动分批。
   batchWaitTime: 0 #分批类型为auto时使用，表示分批间间隔时间，单位为分钟。
  grayUpdate: #灰度部署时需要配置。
   gray: 1 #灰度的实例数。
 envs:  #容器环境变量参数。
  - name: envtmp0
   value: '0'
  - name: envtmp1
   value: '1'
 customHostAlias:  #容器内自定义host映射。
  - hostName: 'samplehost1'
   ip: '127.X.X.X'
  - hostName: 'samplehost2'
   ip: '127.X.X.X'
 jarStartOptions:  #JAR包启动应用选项。
 jarStartArgs:   #JAR包启动应用参数。
 liveness:  #容器健康检查，健康检查失败的容器将被重启。
  exec:
   command:
    - sleep
    - 1s
  initialDelaySeconds: 5
  timeoutSeconds: 11
  periodSeconds: 30 #1.1.7新版本支持的功能。
 readiness:  #应用启动状态检查，多次健康检查失败的容器将被重启。不通过健康检查的容器将不会有SLB流量进入。
  exec:
   command:
    - sleep
    - 1s
  initialDelaySeconds: 5
  timeoutSeconds: 11
  periodSeconds: 30  #1.1.7新版本支持的功能。
 minReadyInstances: 1  #最小存活实例数。在滚动升级过程中或者滚动升级失败时，可用实例数都将尽可能不小于该值，为0则没有限制。弹性扩缩容的范围不能小于该值，否则将触发异常。
#1.1.7新版本支持的功能如下所示。
 timezone: Asia/Shanghai #时区设置。
 warStartOptions: CATALINA_OPTS=\"$CATALINA_OPTS $Options\" catalina.sh run #WAR包部署的启动命令。
 tomcatConfig: #以WAR包部署的Tomcat配置。
  port: 8082
  maxThreads: 399
  contextPath: /test/
  uriEncoding: UTF-8
```

```
  useBodyEncodingForUri: false
nasConfig: #Nas配置。
  nasId: 3fc4*****
  mountHost: 3fc4*****.cn-shanghai.nas.aliyuncs.com
  mountDesc:
    - mountPath: /nas
      nasPath: /
postStart: #启动后处理。
  exec:
    command:
      - sh
      - -c
      - echo helloPostStart
preStop: #停止前处理。
  exec:
    command:
      - sh
      - -c
      - echo helloPreStop
terminationGracePeriodSeconds: 9 #优雅下线时间，单位为秒。
changeOrderDesc: maven test #发布说明。
autoEnableApplicationScalingRule: true #是否自动启用弹性规则。
configMapMountDesc: #ConfigMap配置。
  - configMapId: 123**
    key: config1
    mountPath: /configMap
enableAhas: true #是否启用AHAS。
ossConfig: #OSS配置。
  ossAkId: ******
  ossAkSecret: ******
  ossMountDecs:
    - bucketName: ******-cn-shanghai-s-*****
      bucketPath: /
      mountPath: /ossPath
      readOnly: true
```

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理

- 应用事件查看
- 变更记录查看
- 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

3.2. 通过IntelliJ IDEA插件部署应用

除了通过SAE控制台进行应用部署，您还可以通过Alibaba Cloud Toolkit for IntelliJ IDEA（简称Cloud Toolkit）插件部署应用。

前提条件

1. 下载并安装JDK1.8或更高版本。
2. 下载并安装IntelliJ IDEA（2018.3或更高版本）。

   > **说明** 由于JetBrains插件官方服务器设立在海外，如果因访问缓慢导致无法下载安装，请加入文末交流群，从Cloud Toolkit产品运营部获取离线安装包。

3. 在IntelliJ IDEA中安装和配置Cloud Toolkit。

背景信息

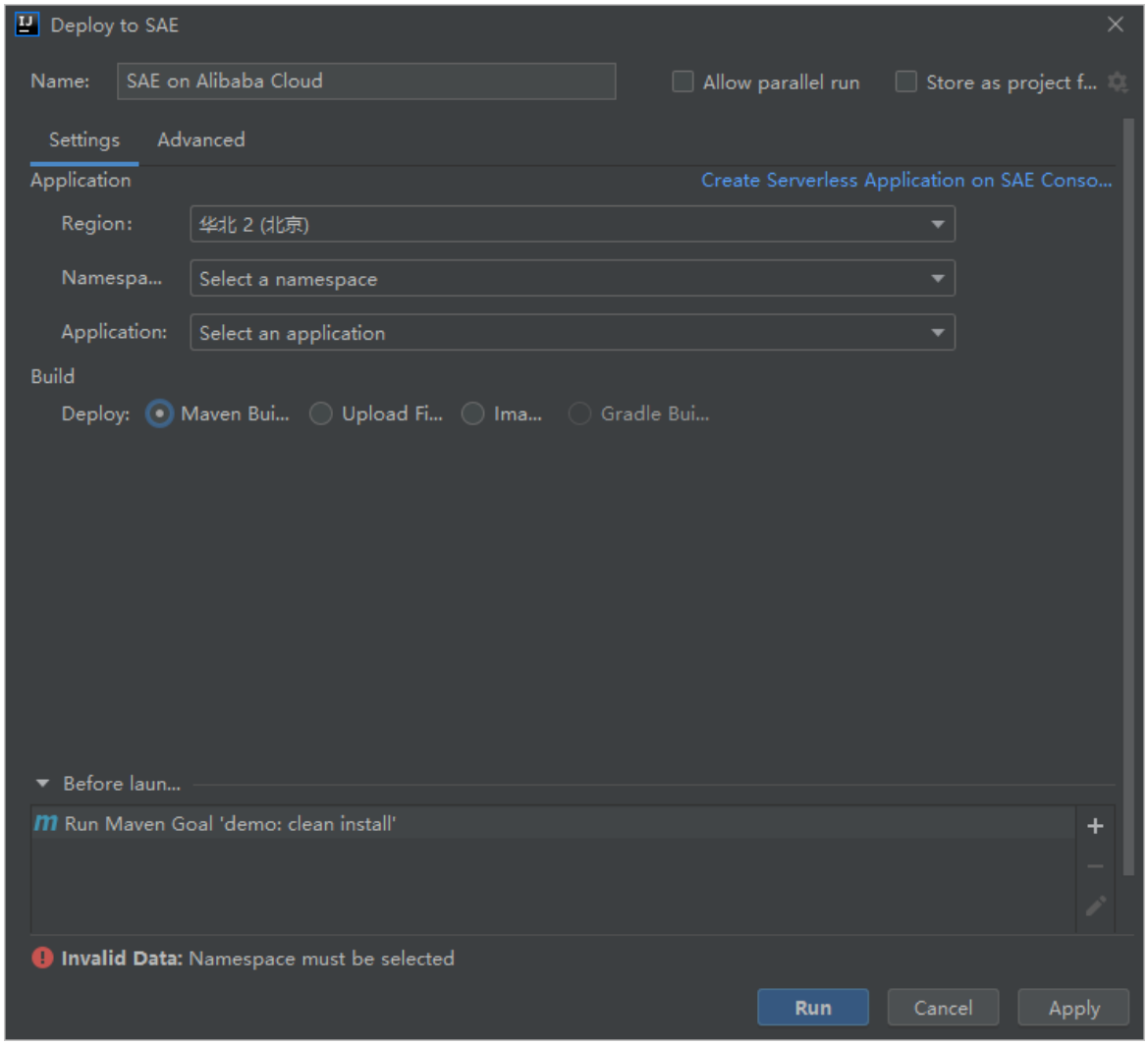
Cloud Toolkit是阿里巴巴提供的免费IDE插件。您可以注册或使用已有的账号免费下载Cloud Toolkit，下载完成后，将其安装在IntelliJ IDEA中。

在本地完成应用程序的开发、调试及测试后，您可以通过本插件将应用程序快速部署到SAE。

部署应用到SAE

Cloud Toolkit插件目前仅支持将应用以WAR包、JAR包或镜像方式部署到SAE。

1. 在IntelliJ IDEA界面左侧的**Project**区域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，配置应用部署参数，配置完成后单击**Apply**保存设置。

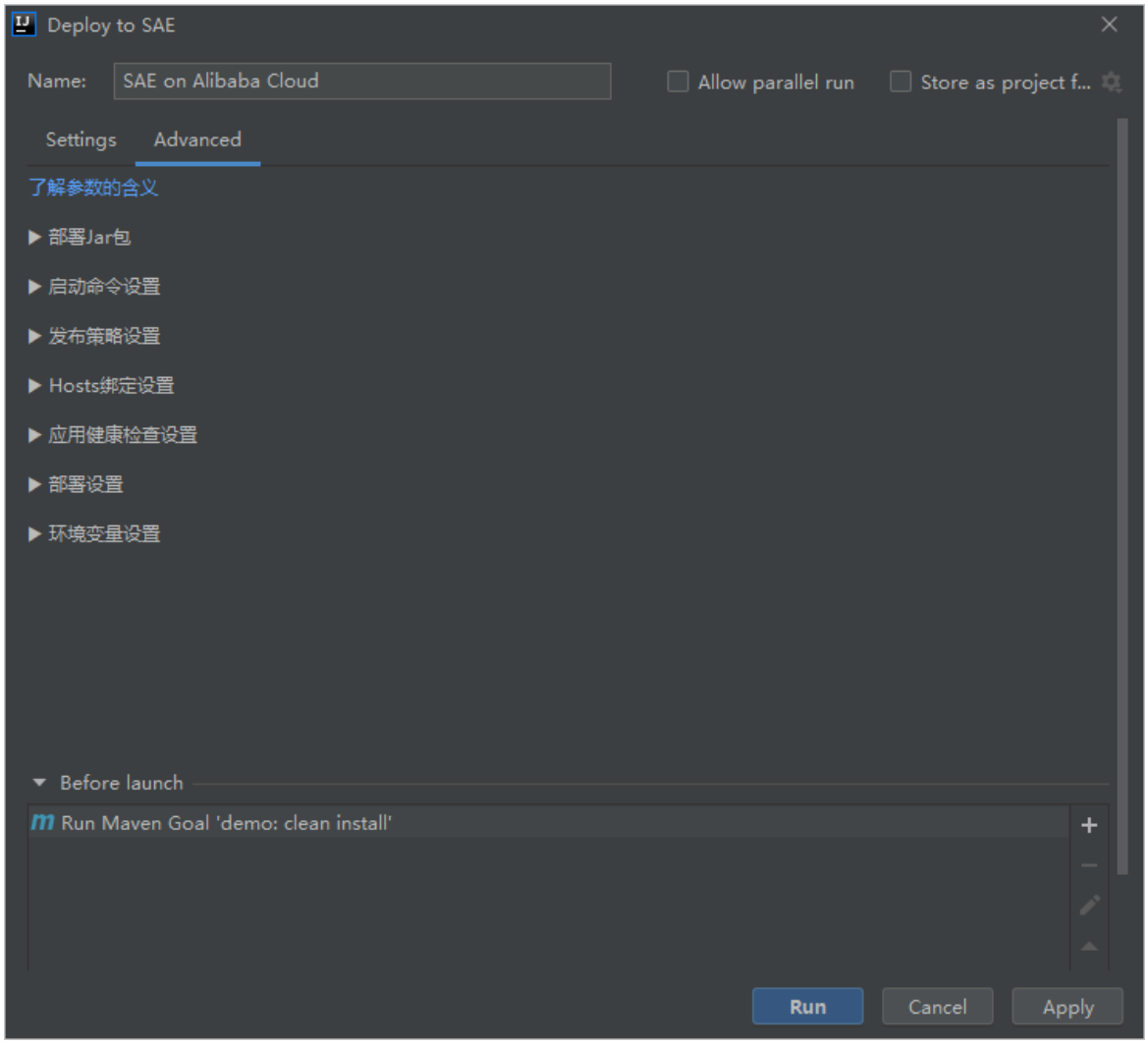


参数说明如下：

参数	说明
Region	选择应用所在地域。
Namespace	选择应用所在命名空间。
Application	选择应用名称。
Build	○ **Maven Build**：Maven Build方式构建应用，默认添加Maven任务构建部署包。如果需要部署子模块，请参见使用IntelliJ IDEA部署多模块工程中的子模块。 ○ **Upload File**：JAR包或者WAR包构建应用，上传WAR包或者JAR包后部署应用。 ○ **Image**：镜像方式构建应用，需要设置镜像地址后部署应用。 ○ **Gradle Build**：暂不支持。

3. （可选）在Deploy to SAE对话框，单击Advanced，在Advanced区域配置应用高级参数。

> **说明** 如果您未配置高级参数，部署时将默认使用SAE控制台上的值。



参数说明如下：

类型	参数	示例值	说明
部署JAR包	Package Version	1.0.1	部署的应用版本号。
	JDK	Open JDK 8	部署的应用依赖的JDK版本。
	Web Container	apache-tomcat-7.0.91	部署的应用依赖的Tomcat版本。

类型	参数	示例值	说明
启动命令设置	Jar Start Options	custom-option	JAR包启动应用选项。应用默认启动命令如下： `$JAVA_HOME/bin/java $JarStartOptions -jar $CATALINA_OPTS "$package_path" $JarStartArgs`
	Jar Start Args	custom-args	JAR包启动应用参数。应用默认启动命令如下： `$JAVA_HOME/bin/java $JarStartOptions -jar $CATALINA_OPTS "$package_path" $JarStartArgs`

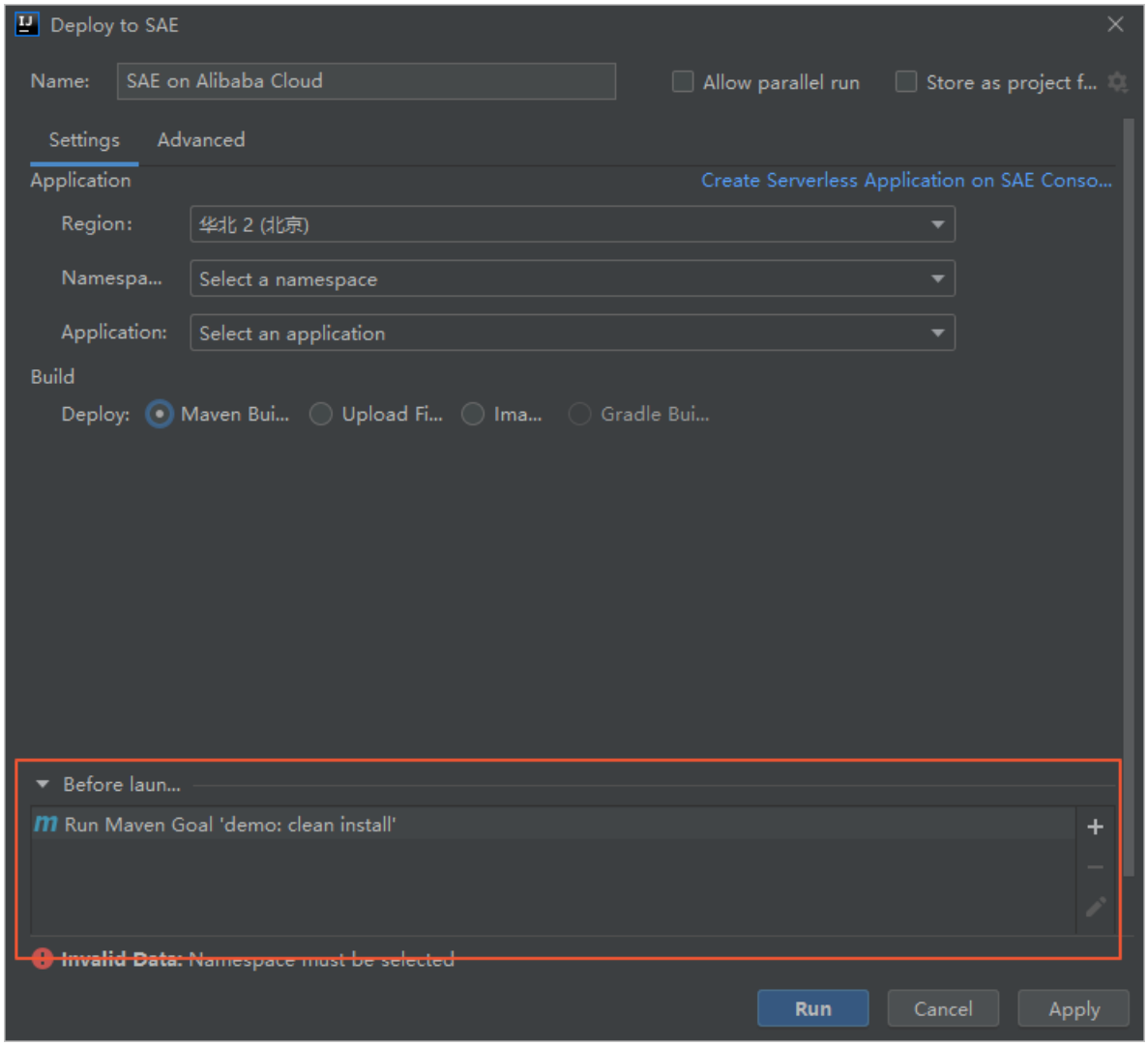
发布策略设置	Update Strategy	{"type":"GrayBatchUpdate","batchUpdate":{"batch":2,"releaseType":"auto","batchWaitTime":1},"grayUpdate":{"gray":1}}	部署策略，可参考以下示例：
			○ 示例1：灰度1台+后续分2批+自动分批+分批间隔1分钟
			`{"type":"GrayBatchUpdate","batchUpdate":{"batch":2,"releaseType":"auto","batchWaitTime":1},"grayUpdate":{"gray":1}}`
			○ 示例2：分2批发布+自动分批+分批间隔0分钟
			`{"type":"BatchUpdate","batchUpdate":{"batch":2,"releaseType":"auto","batchWaitTime":0}}`

类型	参数	示例值	说明
Hosts绑定设置	Custom Host Alias	[{"hostName":"sample host","ip":"127.0.0.1"}]	容器内自定义Host映射。
应用健康检查设置	Liveness	{"exec":{"command":["sleep","5s"]},"initialDelaySeconds":10,"timeoutSeconds":11}	容器健康检查，健康检查失败的容器将被杀死并恢复。目前仅支持容器内下发命令的方式： {"exec":{"command":["sleep","5s"]},"initialDelaySeconds":10,"timeoutSeconds":11}
	Readiness	{"exec":{"command":["sleep","6s"]},"initialDelaySeconds":15,"timeoutSeconds":12}	应用启动状态检查，多次健康检查失败的容器将被杀死并重启。不通过健康检查的容器将不会有SLB流量进入。
部署设置	Min Ready Instances	1	应用的最小存活实例数。
	Batch Wait Time	10	分批发布等待时间，单位为秒。
环境变量设置	Envs	[{"name":"envtmp","value":"0"}]	容器环境变量参数。

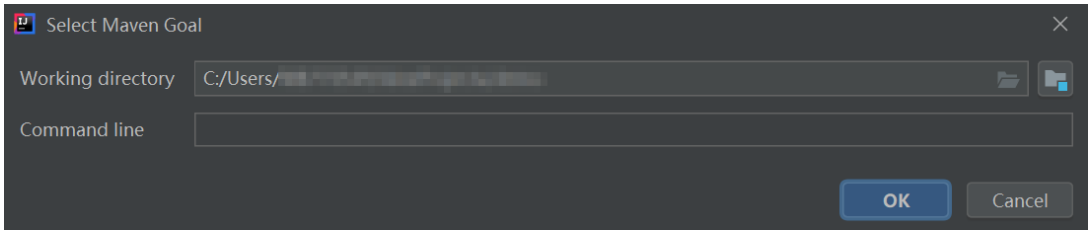
4. 单击**Run**。
   您可以通过以下两种方式证明应用已部署成功：
   - IntelliJ IDEA的**Console**区域打印的运行日志中显示 `BUILD SUCCESS` 。
   - SAE控制台上的应用变更记录显示**执行成功**。

管理Maven构建任务

在Deploy to SAE页面的Before launch区域，您可以对Maven构建任务执行添加、删除、修改和移动操作。



1. 在IntellIj IDEA界面左侧的Project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页面的Before launch区域，管理Maven任务。
   - 添加任务
     a. 单击Before launch区域右侧的+，在下拉框中选择Run Maven Goal。
     b. 在Select Maven Goal对话框中，选择当前工程可用的模块，在Command line区域中输入构建命令。



     c. 单击OK。

- 删除任务：选择需要删除的任务，单击Before launch区域右侧的➖。
- 修改任务：选择需要修改的任务，单击Before launch区域右侧的✎，在Select Maven Goal对话框中修改任务信息，单击OK。
- 移动任务：选择需要移动的任务，单击Before launch区域右侧的▲或▼，调整任务顺序。

部署多模块工程

多模块工程是各个模块独立开发，模块之间存在调用关系的项目工程。Cloud Toolkit可以用于部署多模块工程中的某个子模块的场景。

如果您需要部署多模块Maven工程的子模块，您需要在Deploy to SAE页面的Before launch区域中，将待部署的子模块任务设置为最后执行，具体操作请参见使用Intellij IDEA部署多模块工程中的子模块 。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

3.3. 通过Eclipse插件一键部署应用

您除了通过控制台方式将应用部署到SAE，还可以通过Alibaba Cloud Toolkit for Eclipse插件进行部署。

前提条件

- 下载并安装JDK1.8或更高版本。
- 下载并安装适用于Java EE开发的Eclipse IDE、4.5.0（代号：Mars）或更高版本。

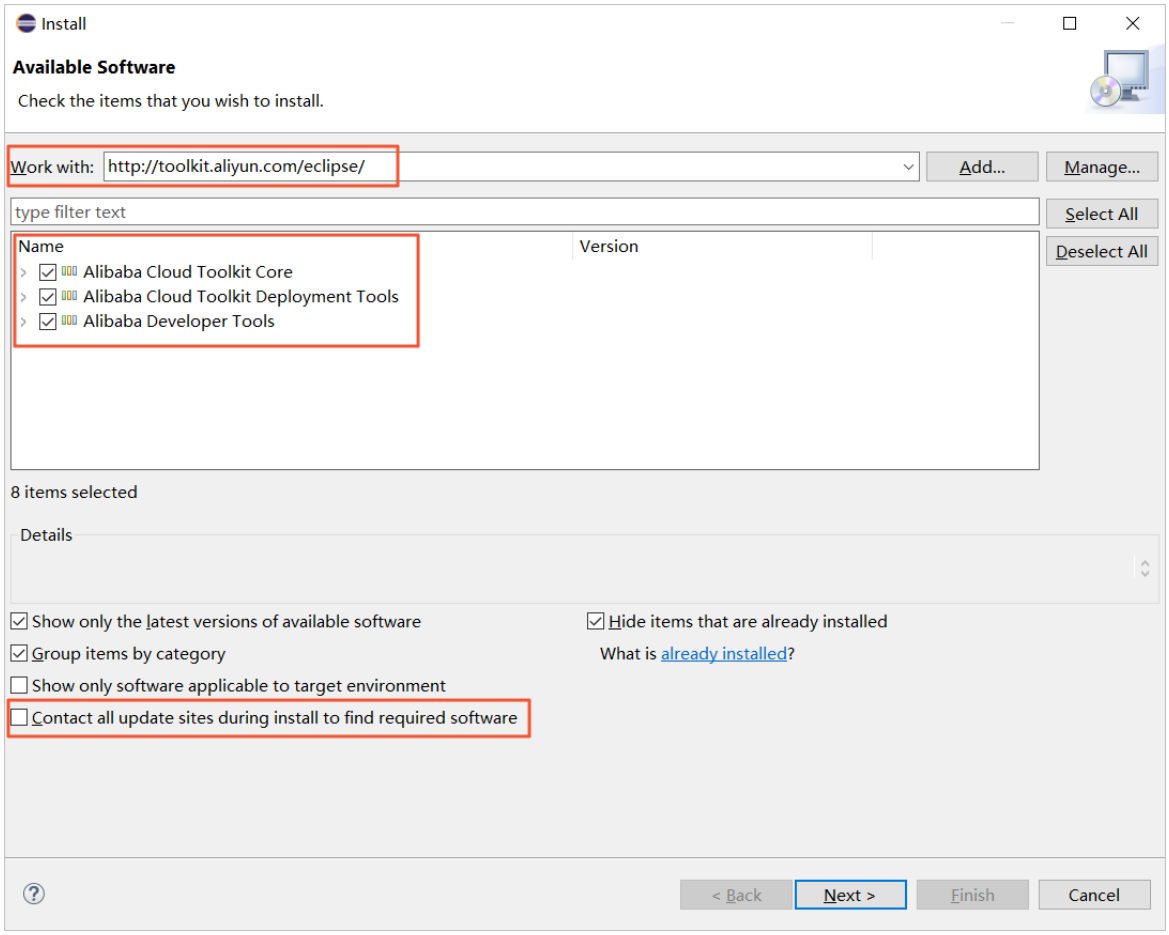
背景信息

Cloud Toolkit是阿里巴巴提供的免费IDE插件。您可以注册或使用已有的账号免费下载Cloud Toolkit，下载完成后，将其安装在IntelliJ IDEA中。

在本地完成应用程序的开发、调试及测试后，您可以通过本插件将应用程序快速部署到SAE。

安装Cloud Toolkit

1. 启动Eclipse。
2. 在菜单栏中选择**Help > Install New Software**。
3. 在**Available Software**对话框的**Work with**文本框中，输入*Cloud Toolkit for Eclipse*的URL*http://toolkit.aliyun.com/eclipse/*，然后回车。
4. 组件配置。



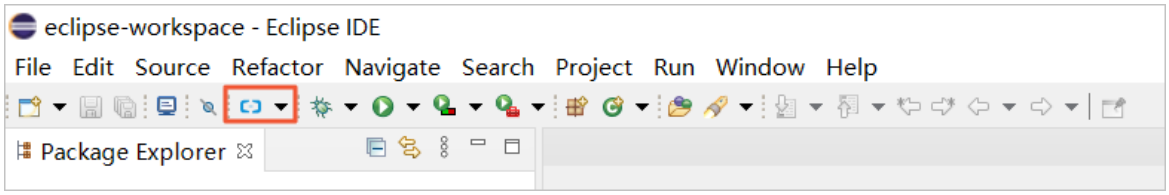
   i. 在**type filter text**列表区域中，勾选需要的组件。
   ii. 在下方**Details**区域中，清除勾选**Connect all update sites during install to find required software**。
   iii. 单击**Next**。

5. 按照Eclipse安装页面的提示，完成后续安装步骤。

> **注意** 如果安装过程中弹出没有数字签名的提示信息，请选择Install anyway。

6. 重启Eclipse。

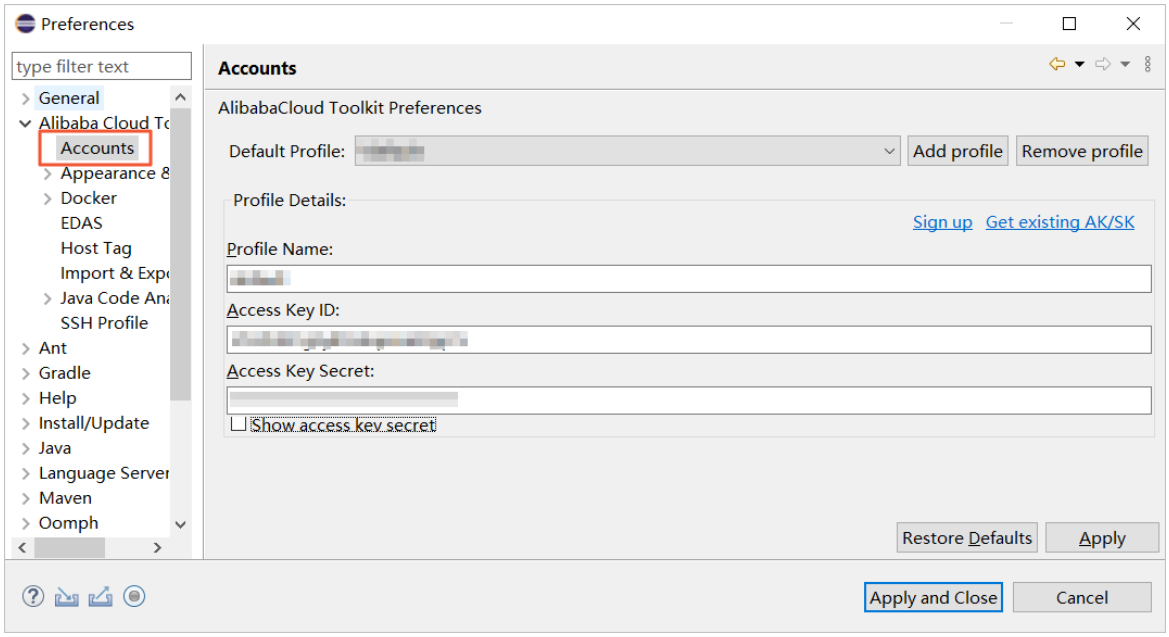
   Cloud Toolkit插件安装完成后，重启Eclipse。重启后在工具栏显示Alibaba Cloud Toolkit 图标。



配置Cloud Toolkit账号

使用Cloud Toolkit部署应用到云端时，需要调用阿里云的API，调用API时需要使用访问密钥（AccessKey，包括AccessKey ID和AccessKey Secret）进行云端身份验证。因此在部署应用之前，需要先在Cloud Toolkit中配置账户信息。

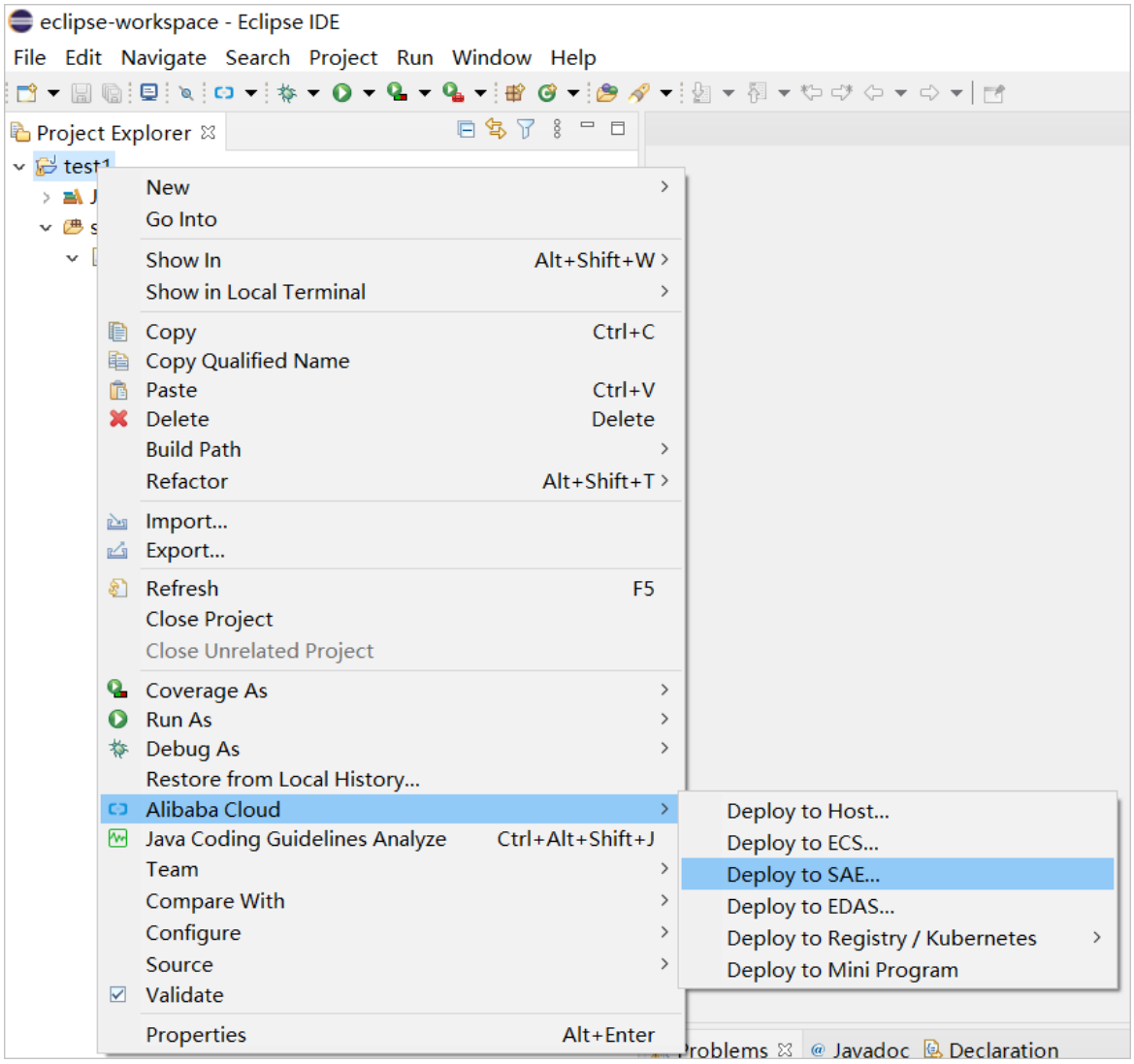
1. 获取AccessKey。
2. 启动Eclipse。
3. 在顶部菜单栏，选择 **Windows > Preferences**。
4. 在**Preferences**页面的左侧导航栏，选择**Alibaba Cloud Toolkit>Accounts**。
5. 在**Accounts**页面，输入**Access Key ID**和**Access Key Secret**，并单击**Apply and Close**。



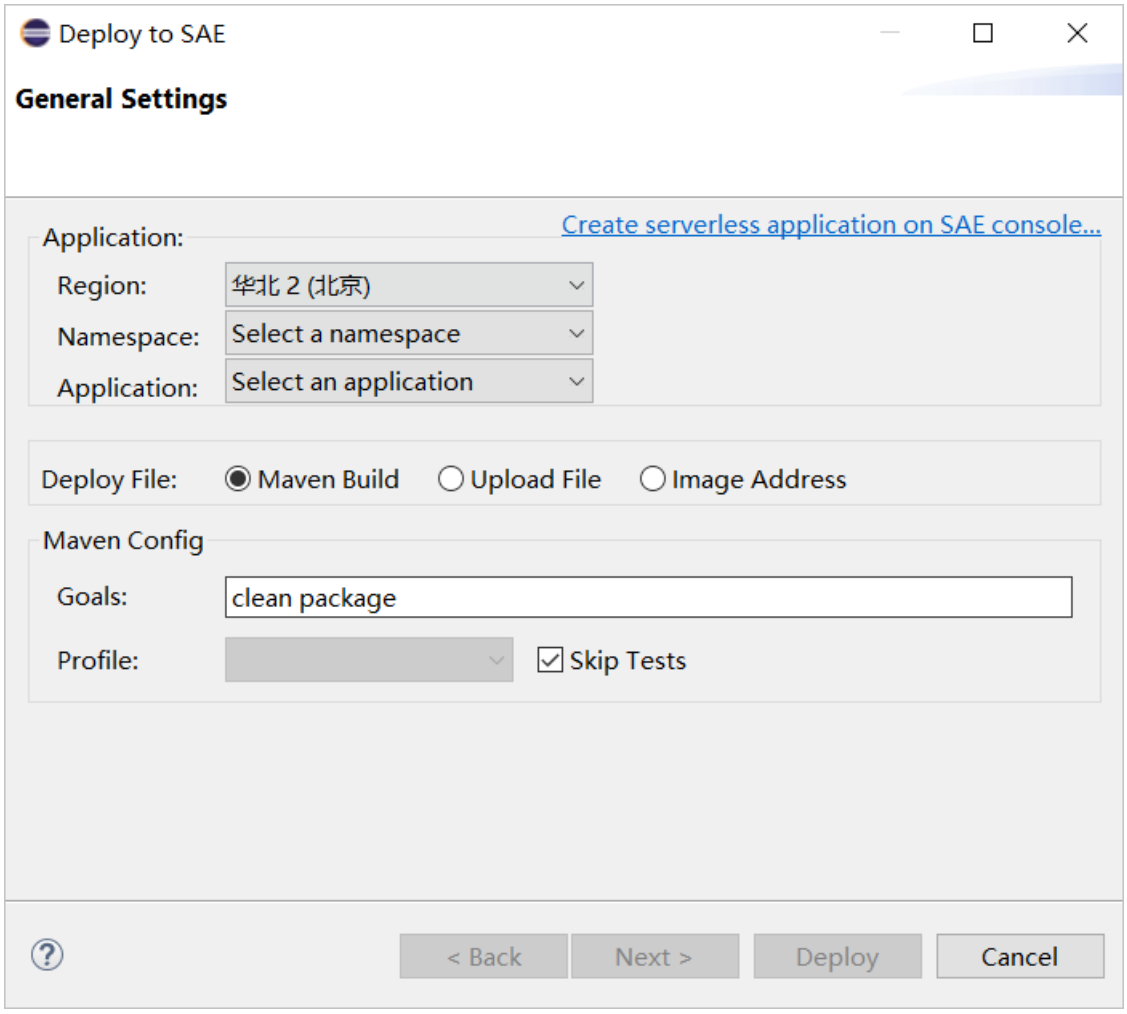
将应用部署到SAE

Cloud Toolkit插件支持将应用以WAR包、JAR包或镜像方式部署到SAE。

1. 在Eclipse页面左侧的**Package Explorer**区域，右键单击待部署的工程名，并在弹出的菜单栏中选择**Alibaba Cloud > Deploy to SAE...**。



2. 在**Deploy to SAE**对话框中，依据需求选择应用的**Region**、**Namespace**和**Application**，并设置部署方式。



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，请参见使用Eclipse部署多模块工程中的子模块。

参数	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. 配置完成后，单击**Deploy**。
   - 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。

终止Cloud Toolkit插件运行

在插件运行过程中，如果现场需要运行其他插件，请在**Progress**页面终止SAE-deploy进程。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.CICD部署

4.1. 使用云效部署应用至SAE

4.1.1. 部署Java应用至SAE

本文以Java应用为例，介绍如何使用云效以镜像方式将应用部署至SAE。云效现已支持分批发布的发布策略。

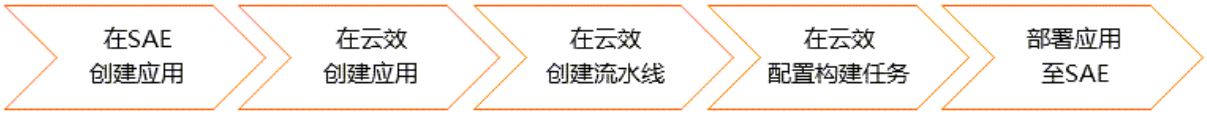
前提条件

- 注册阿里云账号。
- 将业务代码上传至阿里云Code。

> **说明** 业务代码中须包含应用的Dockerfile文件，具体制作步骤请参见制作应用容器Docker镜像。

- 将镜像文件上传至阿里镜像库。
- 开通云效。

部署流程



1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果第一次部署应用到SAE，则需要在SAE控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在SAE创建应用完成后，需要在云效控制台创建对应的应用，应用的部署方式（例如JAR、WAR和镜像）两边需要一致。
3. 在云效控制台完成应用创建后，需要使用云效提供的流水线模板创建出应用集成发布的基础流水线。
4. 基础流水线创建完成后，需要更改流水线中的构建任务。例如：如果您在SAE控制台使用镜像部署应用，那么在该环节您需要将原流水线**构建**任务删掉，添加为供镜像方式部署使用的**Java镜像构建**。整个流水线运行后，本环节的构建产物将供部署环节发布使用。
5. 构建任务配置完成后，需要指定上环节（即构建任务环节）构建产物和应用的部署目的地，即应用在SAE上的区域和应用名称。整个流水线运行成功后，应用成功部署到SAE上。

步骤一：在SAE上创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上使用Demo应用创建相应的应用。

本文以镜像部署为例，具体操作请参见在SAE控制台使用镜像部署PHP应用。

> **注意** 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。简而言之，在SAE上使用镜像方式部署，那么在云效设置流水线时，构建环节必须是镜像相关配置。

步骤二：在云效配置应用基本信息

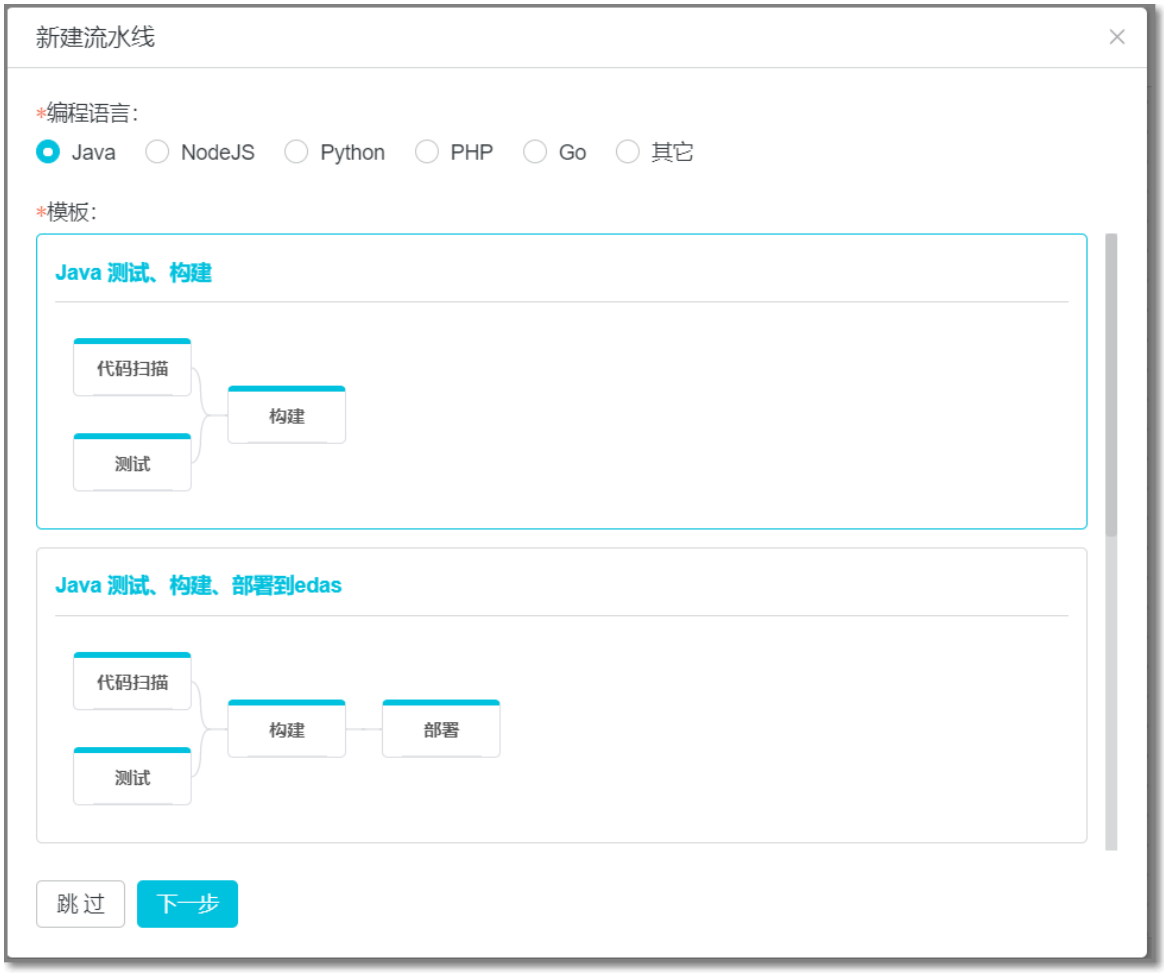
1. 登录云效控制台应用列表页面。
2. 在菜单栏中选择**项目 > 项目列表**，并在**项目列表**页面单击右上角的**新建项目**，然后在弹出的页面中设置项目信息，并单击**确定**。

   项目信息有**项目类型**、**项目名称**、**公开性**和**项目背景**等，请依据实际情况设置。

3. 在菜单栏中选择**研发 > 应用**，并在**我的应用**页面中，单击左上角的**创建新应用**，然后在**创建新应用**对话框设置应用基本信息并单击**创建**。

步骤三：在云效使用模板构建基础流水线

1. 在菜单栏中选择**研发 > 流水线**，在**流水线**页面单击右上角的**新建流水线**。
2. 在**新建流水线**页面设置**编程语言**和**流程模板**，并单击**下一步**。

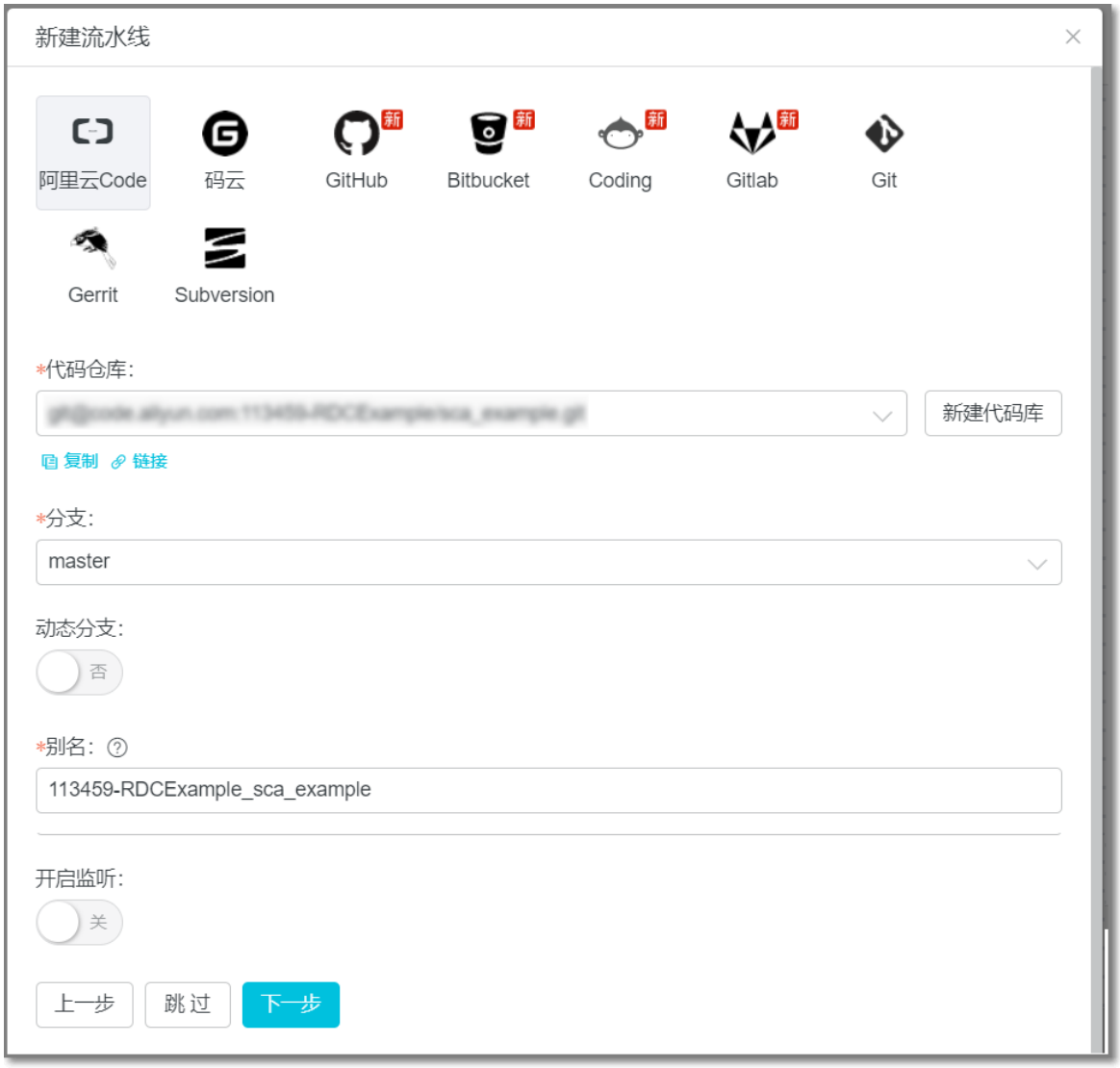


- **编程语言**：选择Java。

- **模板**：选择**Java测试、构建**。

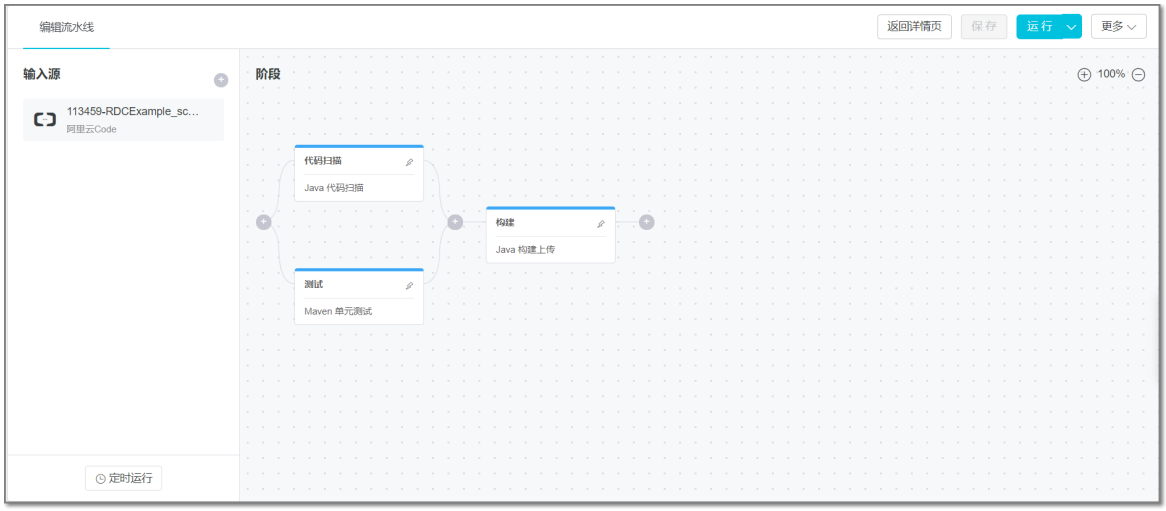
> **说明** 如果您的应用编程语言非Java，请根据实际情况选择。

3. 在弹出的流水线配置页面中，进行代码库设置，并单击**下一步**。



- 代码库类型：选择**阿里云Code**。
- **代码仓库**：选择您的代码仓库地址。
- **分支**：选择代码分支。
- **别名**：代码的别名，后续用于执行组件代码克隆路径，使用数字、字母或下划线。

4. 在弹出的流水线配置页面中，设置流水线的基本信息如流水线名称、管理员等，并单击**创建**。



步骤四：在云效中配置构建任务

1. 在**编辑流水线**页面单击**构建**，在**阶段：构建**页面单击 。

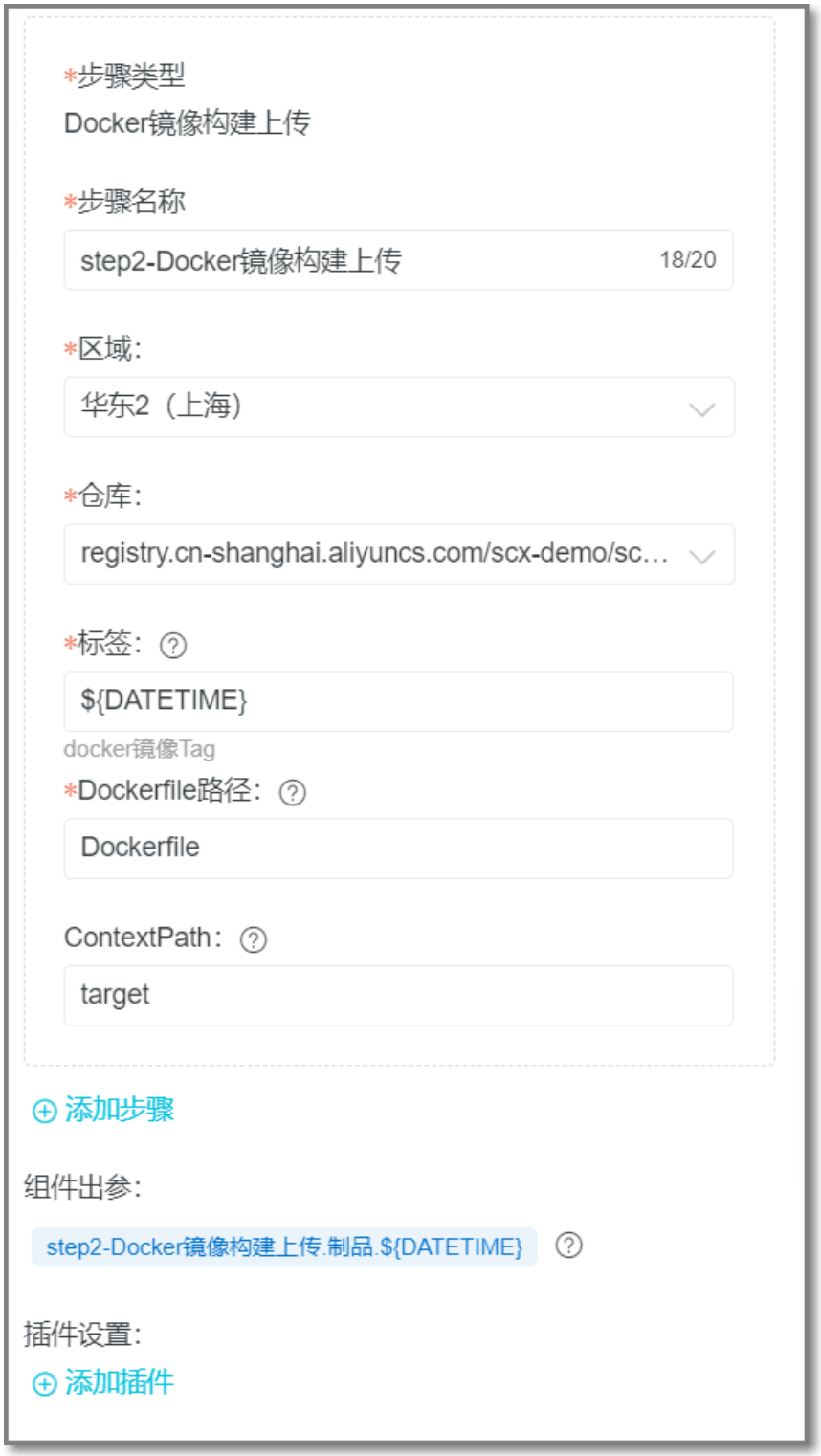
2. 在**编辑流水线**页面单击 ，并在弹出的**阶段模板**中选择**Java镜像构建**。

阶段模板
构建
Java 构建
Java 构建，压缩包上传，用于脚本部署
使 用
发布maven二方库
maven构建，并上传到maven私服
使 用
Java 镜像构建
Java 构建Docker镜像并推送镜像仓库
使 用
Java JAR/WAR 包构建
Java JAR/WAR 包构建上传，常用于EDAS和SAE部署
使 用
Python 镜像构建
Python 镜像构建
使 用
Python 构建
Python 构建 上传到仓库
使 用
Node.js 镜像构建
Node.js 构建后生成Docker镜像
使 用
Node.js 构建
Node.js 构建
使 用
PHP 镜像构建
PHP 镜像构建
使 用
PHP 构建
PHP 构建上传到仓库
使 用
Go 镜像构建
Go 镜像构建
使 用
Go 构建
Go 构建
使 用

阶段
100%
代码扫描
Java 代码扫描
测试
Maven 单元测试
新增阶段3
Java构建Docker镜像...
阶段: 新增阶段3
*阶段名称:
新增阶段3
5/20
*流转配置:
自动流转
任务列表:
Java构建Docker镜像并推送镜像仓库
未完成
新建任务

3. 配置构建任务的基本信息。
   i. 在弹出的**阶段：新增阶段**面板中设置**阶段名称**和**流转配置**。

4. 配置构建任务的执行步骤。
   i. 单击**任务列表**中具体任务名称。
   ii. 在**任务：Java构建Docker镜像并推送镜像仓**页面的**Docker镜像构建上传**区域，设置镜像信息。



- **步骤名称**：保持默认，也可以自定义。
- **区域**：Docker镜像文件所在的地域。
- **仓库**：Docker镜像文件所在地址。
- **标签**：Docker镜像Tag，支持固定参数例如1.0，或者动态参数${DATETIME}。
- **Dockerfile路径**：Dockerfile相对于代码库根目录所在路径，如META/config/Dockerfile或Dockerfile。该文件云效会自动为您创建，无需自建。

- **ContextPath**：填写相对于代码根目录的路径，如target，如果不填则为Dockerfile文件所在目录。
- **组件出参**：为本构建环节产生的结果文件，供部署环节使用。

步骤五：在云效部署应用至SAE

步骤一所创建应用不包含您的任何业务代码，在本环节将应用代码推送至SAE上，即将SAE上的应用升级为含有您业务代码的应用。

> **说明** 云效现已支持分批发布的发布策略，详情参见步骤4。

1. 在**编辑流水线**页面的**阶段**区域中，单击 + 。
2. 在**阶段模板**面板中选择**部署到 SAE**。

3. 单击新增的流水线阶段，在右侧面板上设置阶段的基本信息，并单击**任务列表**下的**部署到 SAE**。

4. 在**任务：部署到SAE**面板中配置部署任务。

- **任务名称**：自定义的任务名称。不可超过20个字符。
- **区域**：选择步骤一所创建的应用所在地域。
- **SAE应用**：选择步骤一所创建的应用。
- **构建产物**：为步骤四中**组件出参**产生的结果。
- **发布策略**：可选择**分批发布**或**灰度发布**。
- **分批方式**：可选择**手动确认**或**自动确认**。例如，如需在完成第一批发布时先观察发布结果再决定后

续操作，则可选择**手动确认**。

- **灰度台数**：要执行灰度发布的机器数量。

> **说明** 此字段仅在**发布策略**为**灰度发布**时显示。

- **发布批次**：发布分批的数量。
- **分批等待时间**：相邻发布批次之间的等待时间。

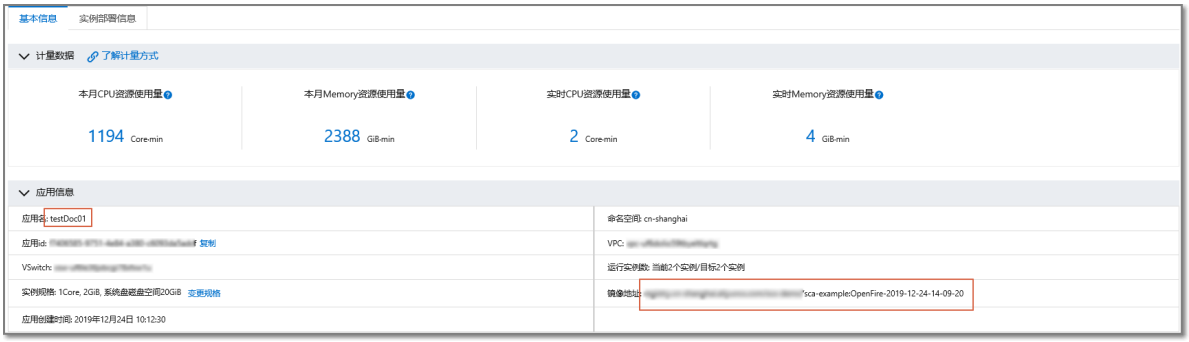
5. 配置完毕后，单击页面右上角的**保存**和**运行**。

云效开始按照配置的流水线工作，最终将应用部署至SAE。

结果验证

- 方法一：
  云效显示部署成功后，在SAE控制台查看应用的变更记录，是否产生应用重新部署的变更记录。

- 方法二：
  云效显示部署成功后，在SAE控制台查看应用的基本信息，查看镜像地址是否与在云效中设置相同。



常见问题

- 如何使用云效以JAR或者WAR包方式部署应用到SAE?
  i. 在SAE控制台以JAR或者WAR包方式创建应用。

ii. 在步骤四：在云效中配置构建任务的步骤2中构建**阶段模板**选择**Java JAR/WAR包构建**，并在弹出的任务步骤设置页面中，依据提示设置JAR或者WAR包构建物信息。

iii. 设置部署阶段任务并将应用至SAE。

- 其他编程语言如何使用云效将应用部署到SAE?
  在流水线构建任务环节，您选择所需的编程语言模板，具体部署操作请参见部署Node.js应用至SAE和部署Golang应用至SAE。

- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用那种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。

- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.1.2. 部署Golang应用至SAE

本文以Golang应用为例，指导您如何使用云效将应用以镜像的方式部署至SAE。

前提条件

- 注册阿里云账号
- 将业务代码上传至阿里云Code

> 注意　业务代码中须包含应用的Dockerfile文件。

- 将镜像文件上传至阿里镜像库
- 开通云效

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE

控制台创建应用，以便云效将业务代码推送到该应用中。

2. 在SAE创建应用完成后，需要在云效创建对应的应用，应用的部署方式（如JAR、WAR和镜像）两边需要一致。
3. 在云效应用创建完成后，可以使用云效提供的流水线模板创建应用集成发布的基础流水线。
4. 基础流水线创建完成，需要更改流水线中的构建任务。例如：如果您在SAE控制台创建的应用采用的是镜像方式，那么在该环节您需要将构建任务中的构建步骤配置为**Docker镜像构建上传**。整个流水线运行后，本环节的构建产物将供部署环节发布使用。
5. 构建任务配置完成后，需要指定上一环节（即构建任务环节）构建产物和应用的部署目的地，即应用在SAE上的区域和应用名称。整个流水线运行成功后，应用成功部署到SAE上。

步骤一：在SAE上创建应用

如果您第一次使用SAE托管应用，需要预先在SAE上使用Demo应用创建相应的应用。

本文以镜像部署为例，具体操作请参见在SAE控制台使用镜像部署PHP应用。

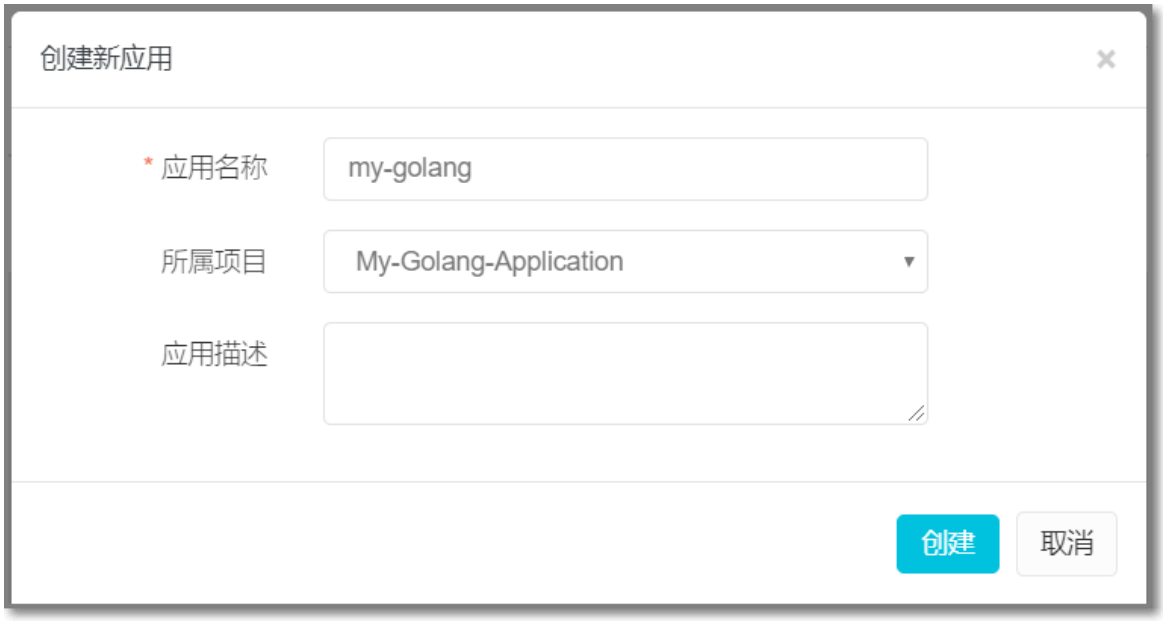
> **注意** 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。简而言之，在SAE上使用镜像方式部署，在云效设置流水线时，构建环节必须是镜像相关配置。

步骤二：在云效配置应用基本信息

1. 登录云效控制台应用列表页面。
2. 在菜单栏中选择**项目 > 项目列表**，并在**项目列表**页面单击右上角的**新建项目**，然后在弹出的页面中设置项目信息并单击**确定**。

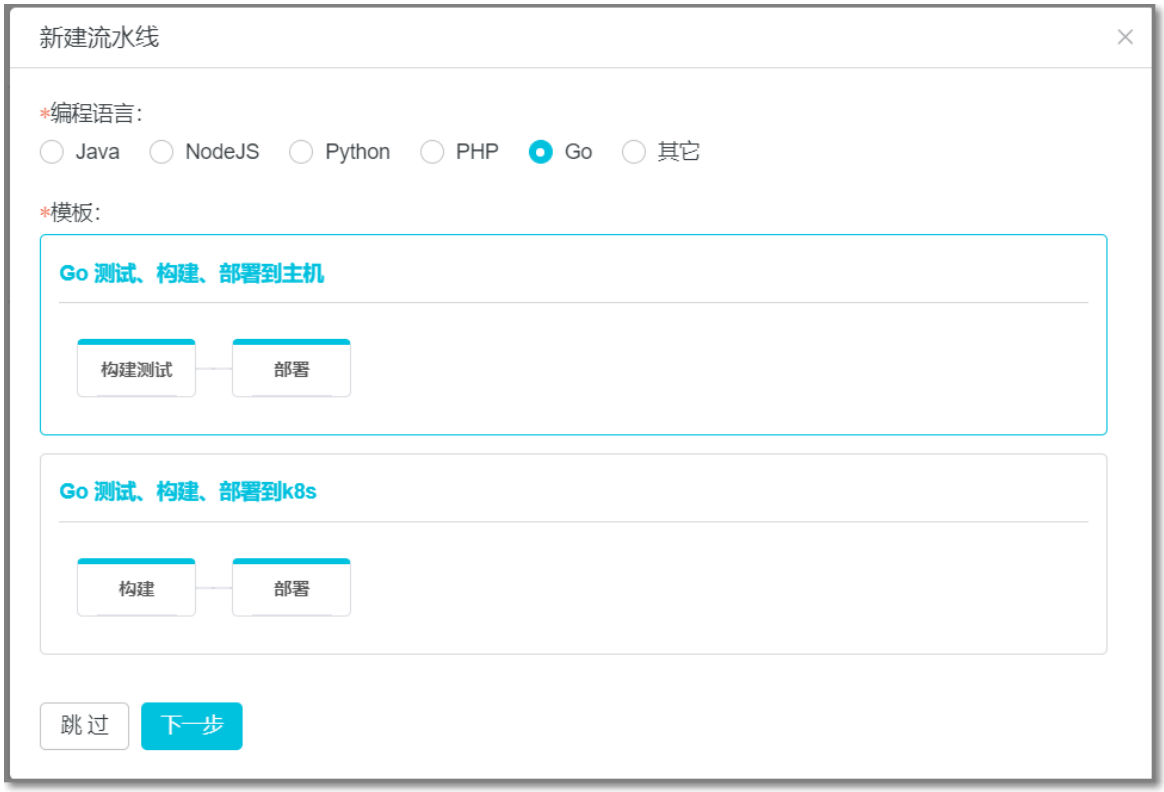
   项目信息有**项目类型**、**项目名称**、**公开性**和**项目背景**等，请依据实际情况设置。

3. 在菜单栏中选择**研发 > 应用**，并在**我的应用**页面中，单击左上角的**创建新应用**，然后在**创建新应用**对话框设置应用基本信息并单击**创建**。



步骤三：在云效使用模板构建基础流水线

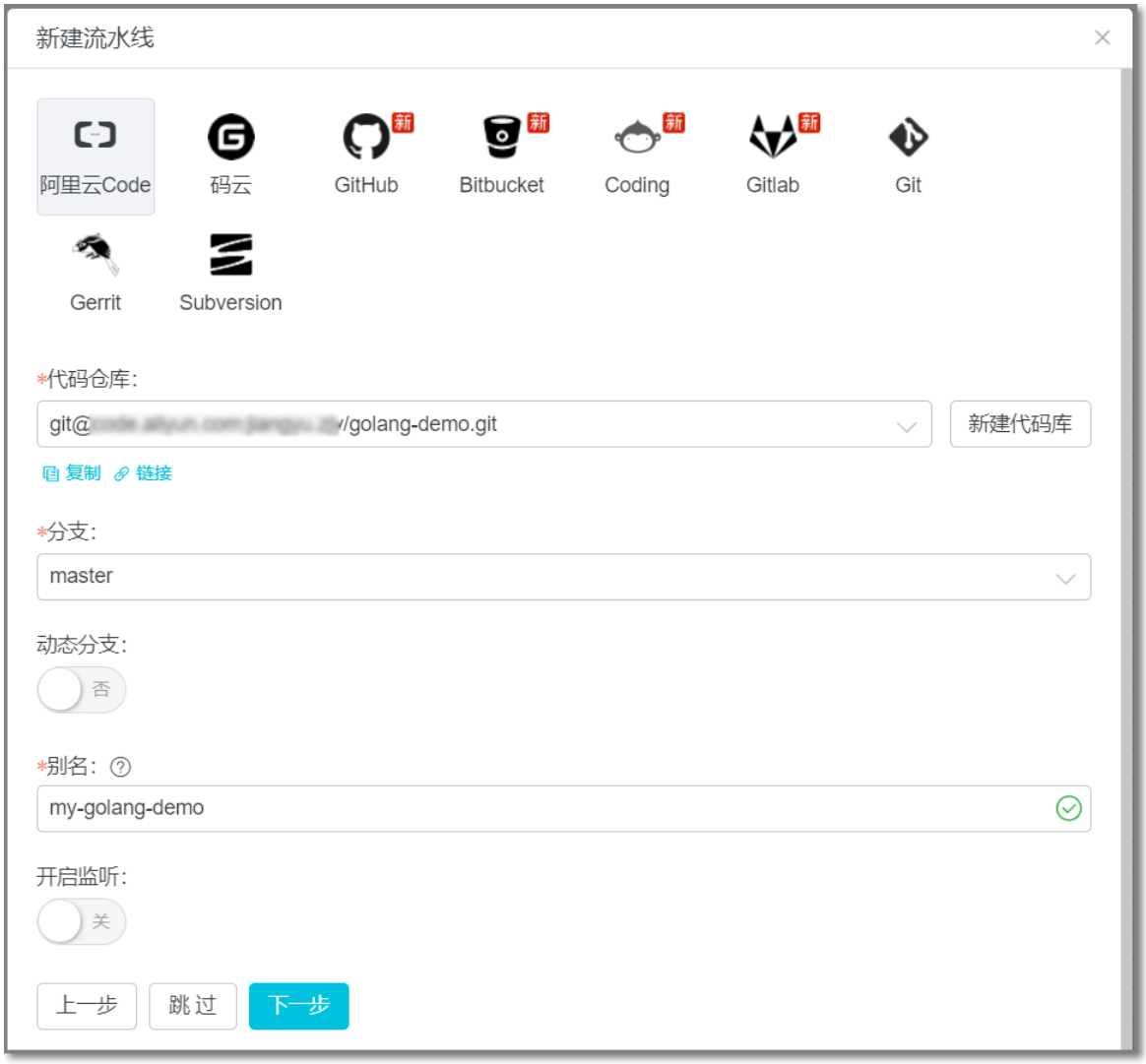
1. 在菜单栏中选择**研发 > 流水线**，在**流水线**页面单击右上角的**新建流水线**。
2. 在**新建流水线**页面设置**编程语言**和**流程模板**，并单击**下一步**。



- **编程语言**：选择**Go**。
- **模板**：选择**Go 测试、构建、部署到主机**。

> **说明** 如果您的应用编程语言非Go，请根据实际情况选择。

3. 在弹出的流水线配置页面中，进行代码库设置，并单击**下一步**。



- 代码库类型：选择**阿里云Code**。
- **代码仓库**：选择您的代码仓库地址。
- **分支**：选择代码分支。
- **别名**：代码的别名，后续用于执行组件代码克隆路径，使用数字、字母或下划线。

4. 在弹出的流水线配置页面中，设置流水线的基本信息如流水线名称、管理员等，并单击**创建**。

步骤四：在云效中配置构建任务

1. 配置构建任务的基本信息。
   i. 在**编辑流水线**页面单击**构建测试**。

ii. 在弹出的**阶段：构建**面板中设置**阶段名称**和**流转配置**。

2. 配置构建任务的执行步骤。

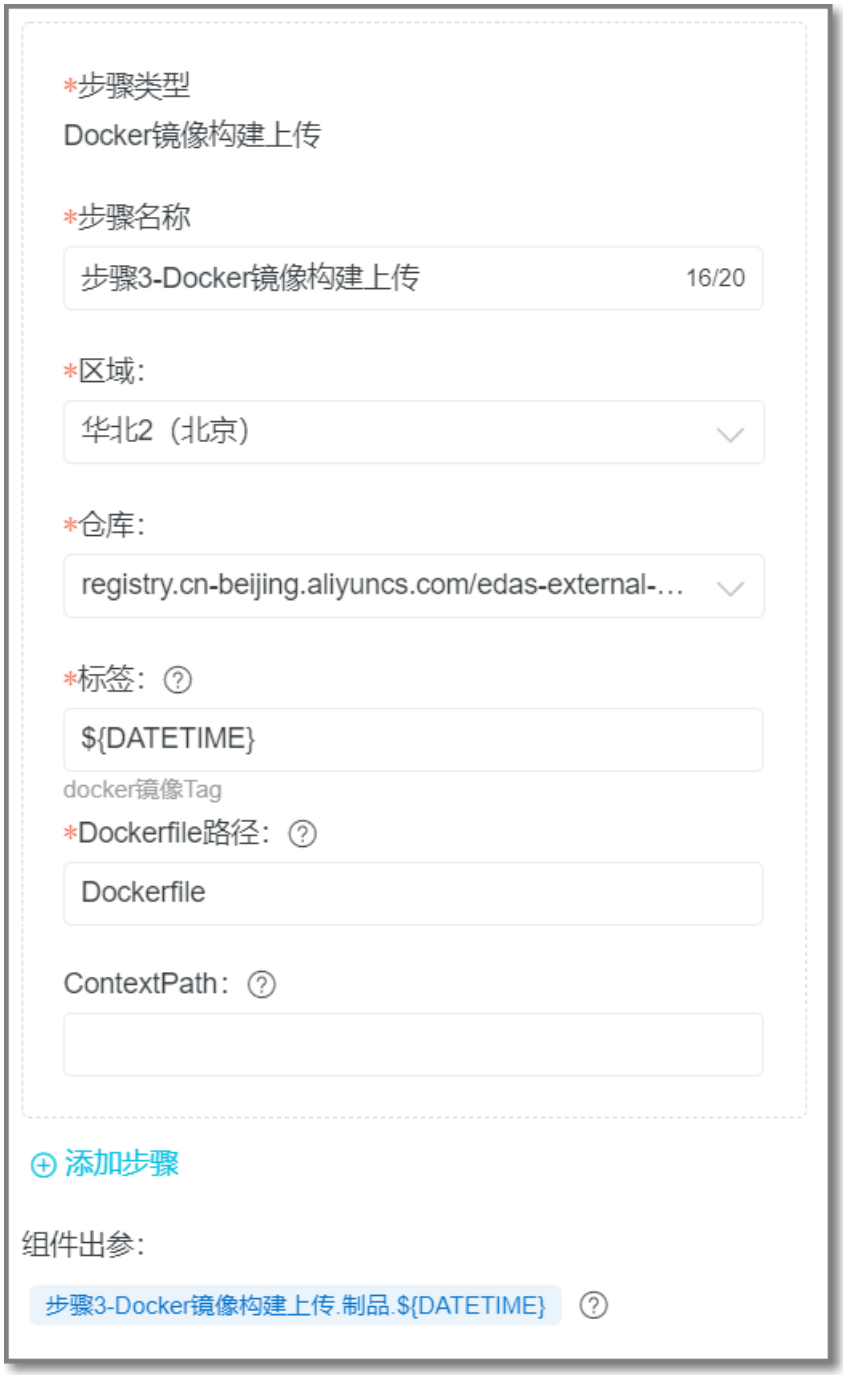
i. 单击**任务列表**中具体任务，在**任务：Go构建Docker**页面删除**构建上传**模块，单击**添加步骤**。

> **说明** 本文以镜像为例，原模板中构建配置不适合镜像方式，需要将其删除并改为镜像配置。

删除原模板中**构建物上传**如下图所示。

ii. 在弹出框中选择**Docker镜像构建上传**。

iii. 在**Docker镜像构建上传**步骤区域，设置镜像信息。



- **步骤名称**：保持默认，也可以自定义。
- **区域**：Docker镜像文件所在的地域。
- **仓库**：Docker镜像文件所在地址。
- **标签**：Docker镜像Tag，支持固定参数例如1.0，或者动态参数${DATETIME}。
- **Dockerfile路径**：Dockerfile相对于代码库根目录所在路径，如META/config/Dockerfile或Dockerfile。该文件云效会自动为您创建，无需自建。
- **ContextPath**：填写相对于代码根目录的路径，如target，如果不填则为Dockerfile文件所在目录。
- **组件出参**：为本构建环节产生的结果文件，供部署环节使用。

步骤五：在云效部署应用至SAE

步骤一所创建应用不包含您的任何业务代码，在本环节将应用代码推送至SAE上。即将SAE上的应用升级为含有您业务代码的应用。

1. 在**编辑流水线**页面的**阶段**区域中，删除原流水线的**部署**任务，然后单击添加。

2. 在弹出的**阶段模板**面板中选择**部署到SAE**。

3. 单击新增的流水线任务，并设置基本信息，设置完成后单击**部署到SAE**。

4. 在**任务：部署到SAE**面板中设置部署任务信息。

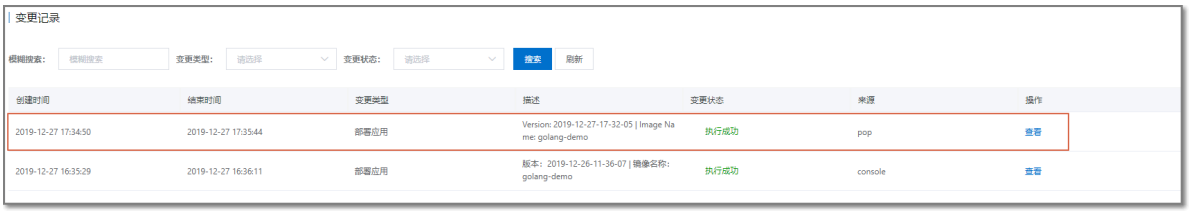
- **区域**：选择步骤一中所创建的应用所在区域。
- **SAE应用**：选择步骤一中所创建应用的应用名称。
- **构建产物**：为步骤四中**组件出参**产生的结果。

5. 单击右上角的**运行**。

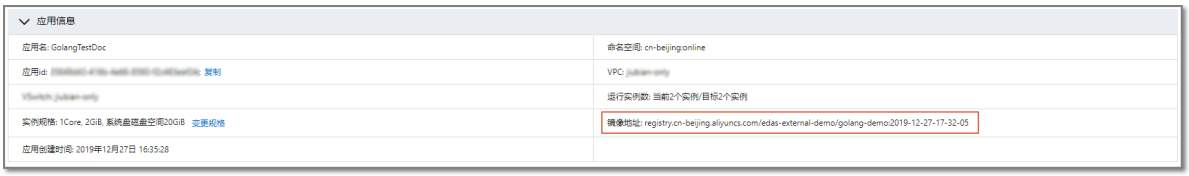
   云效开始依据配置的流水线将应用部署至SAE。

步骤六：结果验证

- 方法一
  云效显示部署成功后，在SAE控制台查看应用的变更记录，是否产生应用重新部署的变更记录。



- 方法二
  云效显示部署成功后，在SAE控制台查看应用的基本信息，查看镜像地址是否与在云效中设置相同。



常见问题

- 其他编程语言怎么使用云效将应用部署到SAE?
  在流水线构建任务环节，您可以选择所需的编程语言模板。具体部署配置操作请参见云效快速入门。

- 除Golang应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用镜像方式。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用

- 配置管理
- 变更实例规格

- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.1.3. 部署Node.js、Python、PHP应用至SAE

SAE支持如Node.js、Python、PHP等多种编程语言开发的应用，如果您的Node.js应用、Python应用或者PHP应用等想要部署到SAE，那么您可以使用云效部署。本文介绍如何使用云效部署应用至SAE。

前提条件

- 注册阿里云账号
- 将业务代码上传至阿里云Code

> **注意** 业务代码中须包含应用的Dockerfile文件。

- 将镜像文件上传至阿里镜像库
- 开通云效

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在SAE创建应用完成后，需要在云效创建对应的应用，应用的部署方式（如JAR、WAR和镜像）两边需要一致。
3. 在云效应用创建完成后，可以使用云效提供的流水线模板创建应用集成发布的基础流水线。

4. 基础流水线创建完成，需要更改流水线中的构建任务。例如：如果您在SAE控制台创建的应用采用的是镜像方式，那么在该环节您需要将构建任务中的构建步骤配置为**Docker镜像构建上传**。整个流水线运行后，本环节的构建产物将供部署环节发布使用。
5. 构建任务配置完成后，需要指定上一环节（即构建任务环节）构建产物和应用的部署目的地，即应用在SAE上的区域和应用名称。整个流水线运行成功后，应用成功部署到SAE上。

操作指导

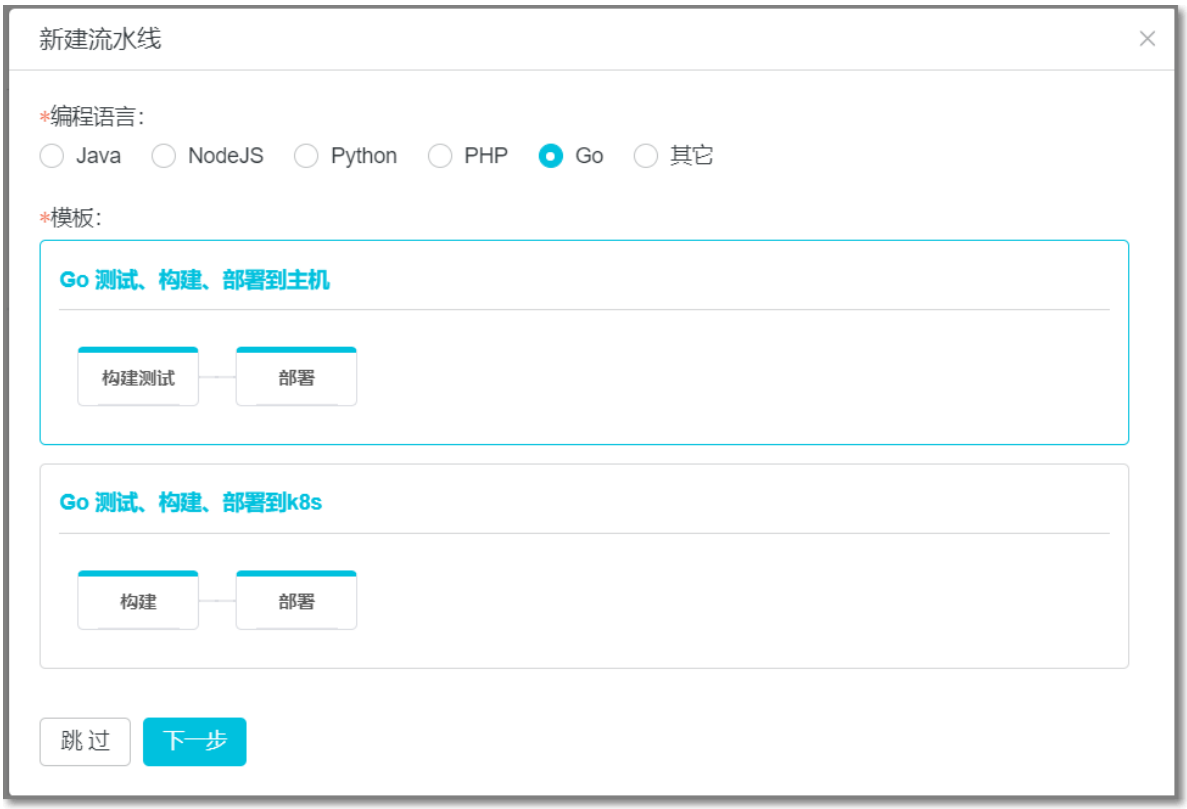
使用云效将Node.js应用、Python应用或者PHP应用等部署到SAE的流程和操作步骤，与使用云效将Java应用、Golang应用部署到SAE相同，详细操作请参见部署Java应用至SAE或者部署Golang应用至SAE。本文仅描述操作过程中的关键配置。

如果您对云效有一定的了解，那么可以依据部署流程和关键配置内容，将应用部署到SAE。

关键配置：

- 在SAE控制台以镜像方式创建并部署Demo应用。

- 在新建流水线时选择相应的编程语言和流水线模板。

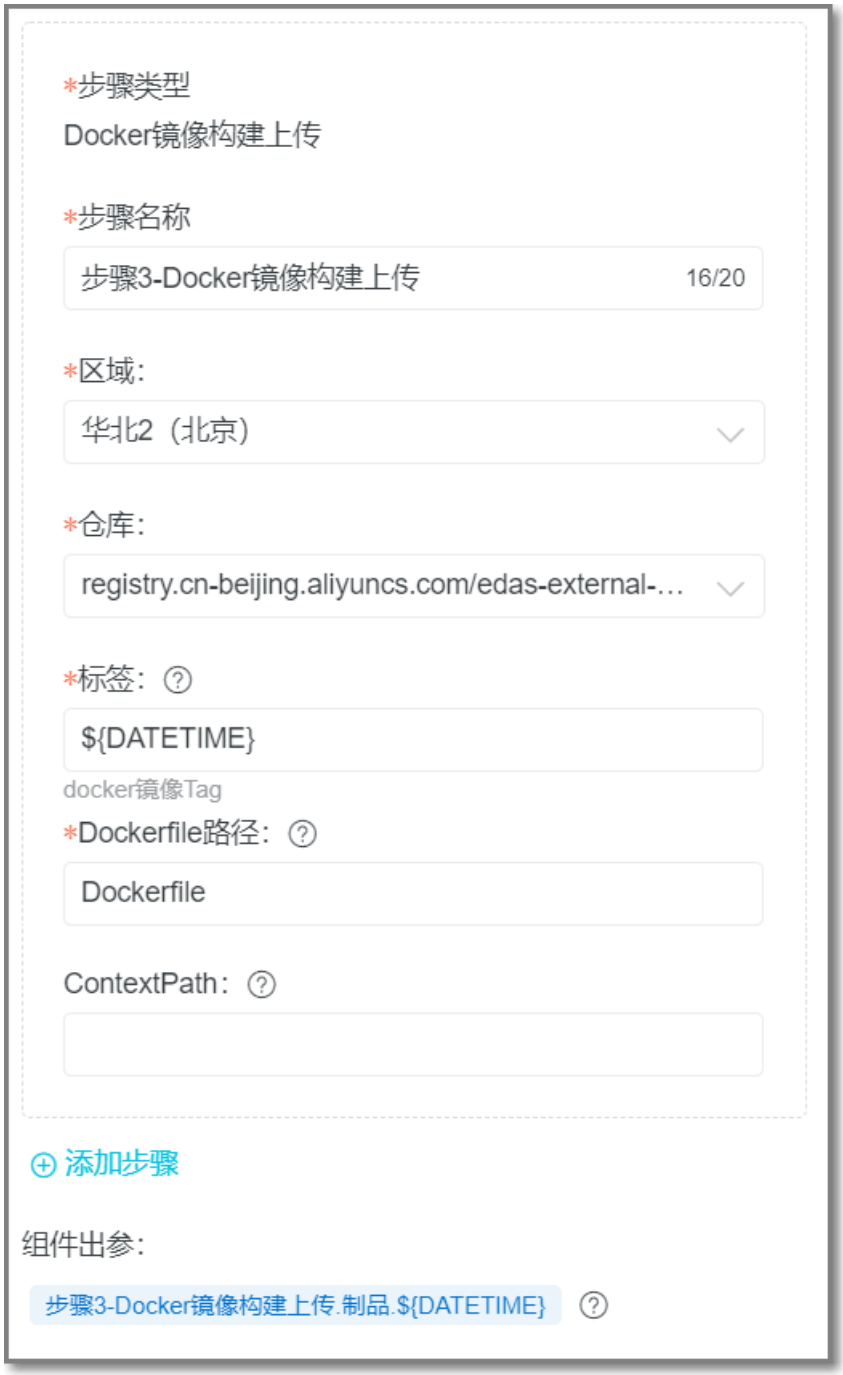


- 在配置构建任务时，将原步骤**构建物上传**删除，新建**Docker镜像构建上传**步骤，然后配置您的镜像信息。

  构建步骤模板选择示意图

Python 单元测试
构建
获取 Git 版本
镜像构建并推送至自定义镜像仓库
Docker镜像构建上传
执行命令
Golang 构建
Java 构建
Node.js 构建
PHP 构建
Python 构建
代码
代码合并
静态扫描

配置镜像信息示意图



- 在部署任务配置时，将部署任务模板选择**部署到 SAE**，并设置部署信息。

更多信息

- 您在SAE部署完应用后，可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作。具体操作，请参见管理应用生命周期。
- 您在SAE部署完应用后，可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作。具体操作，请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用

- 配置管理
- 变更实例规格

- 您在SAE部署完应用后，还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作。具体操作，请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.2. 使用云效2020部署应用至SAE

4.2.1. 部署Java应用至SAE

本文介绍如何使用云效以JAR包或者WAR包的方式将Java语言的应用部署至SAE。云效现已支持分批发布的发布策略。

前提条件

- 注册阿里云账号。
- 开通云效。

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在SAE创建应用完成后，在云效上创建企业并进入流水线FLOW。
3. 根据SAE应用类型配置应用集成发布的流水线信息。
4. 流水线配置完成后，运行流水线完成SAE应用部署。

步骤一：在SAE创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上创建相应的应用。SAE应用支持3种类型，后续流水线配置需要跟应用类型保持一致。本文以JAR包或者WAR包部署为例。

- 在SAE控制台使用WAR包部署Java Web应用
- 在SAE控制台使用JAR文件部署微服务应用

> **注意**
>
> - 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。
> - 除Java应用外，其他语言的应用仅支持**镜像方式**部署，详细操作请参见在SAE控制台使用镜像部署PHP应用。

步骤二：在云效创建企业

如果您是第一次使用云效Flow，则需要在云效上创建您的企业。

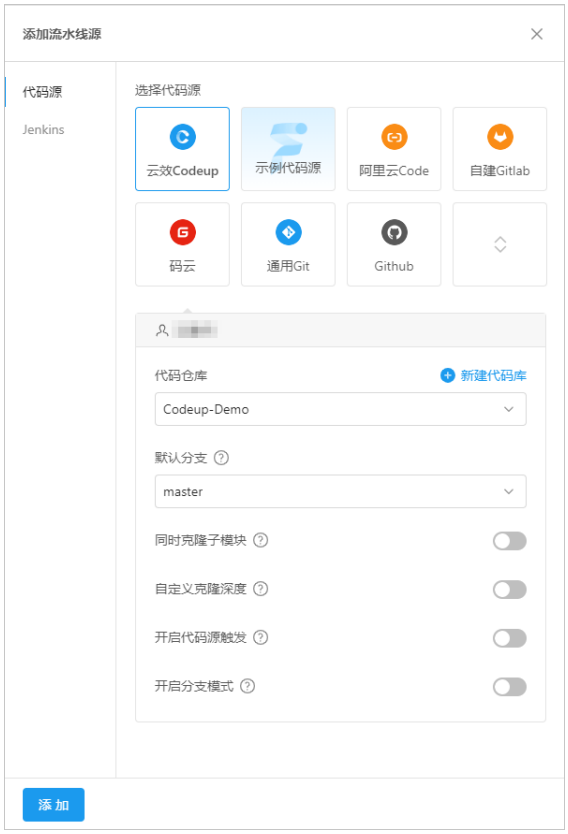
1. 登录云效Flow。
2. 设置企业名称并选择使用规模，单击**完成创建**。

步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在**流水线Flow**页面右上角，单击**新建流水线**。
3. 在**选择流水线模板**对话框，选择**空模板**，单击**创建**。

创建完成后，页面左下角弹出提示框。

4. 流水线创建成功后，默认进入流程配置界面，配置代码源信息，单击**添加**。



- **选择代码源**：选择您业务代码所在的代码源。

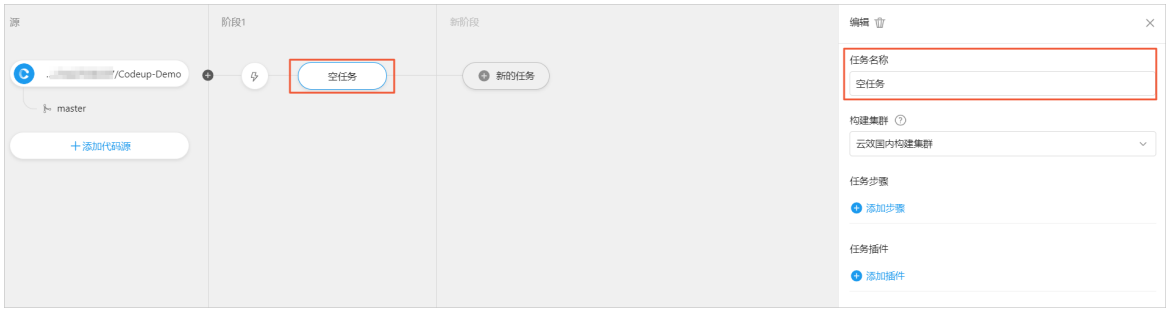
> **说明** 不同代码源需要配置的参数不同，请根据实际界面进行配置。

- **代码仓库**：选择您的代码仓库地址。
- **默认分支**：选择代码分支。

步骤四：在云效部署应用至SAE

本步骤将SAE上的应用升级为含有您业务代码的应用。

1. 在**阶段1**区域单击**空任务**，设置**任务名称**。



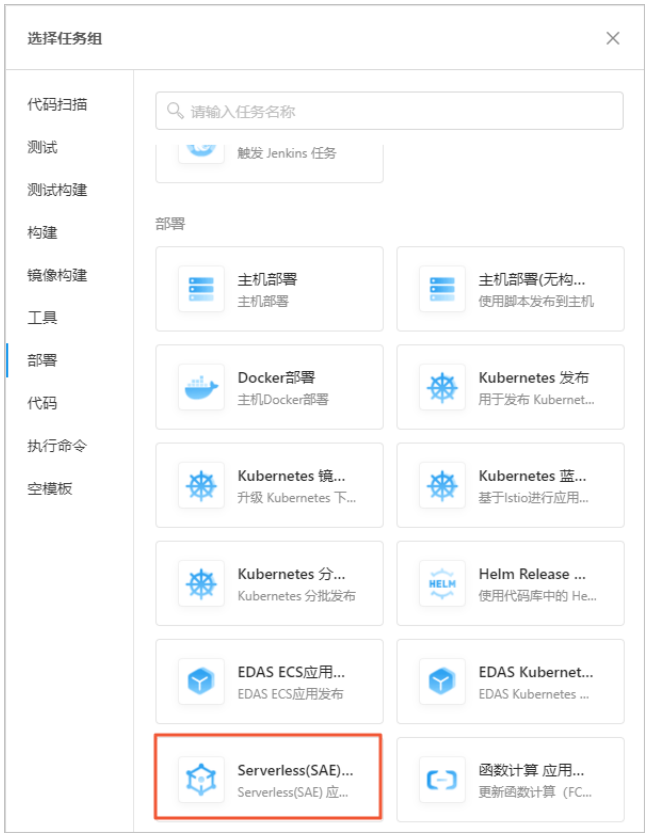
2. 单击**添加步骤**，在下拉列表中选择**构建 > Java构建**，配置Java信息。

- **请选择Java版本**：您可以在下拉列表中选择需要的Java版本。
- **构建命令**：自定义的构建命令。

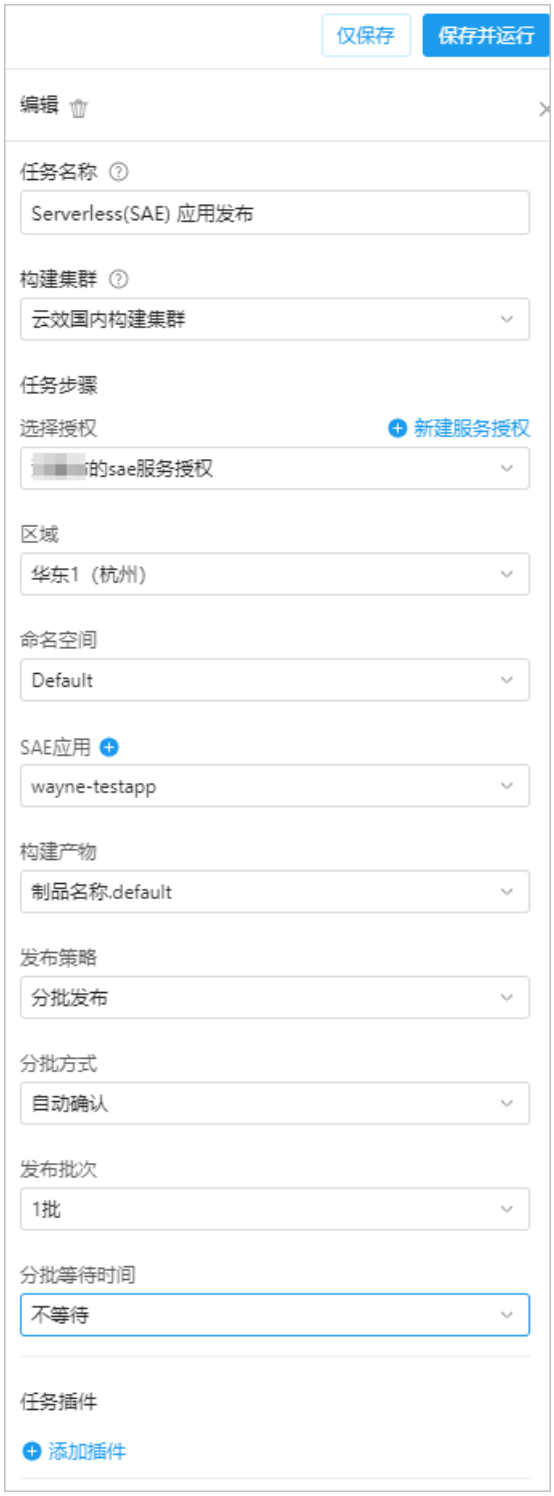
3. 单击**添加步骤**，在下拉列表中选择**上传 > 构建物上传（EDAS/SAE使用）**，上传构建物。

- **制品名称**：自定义的产出物名称。
- **上传文件**：设置需要上传的JAR包或者WAR包所在的路径。

4. 在**新阶段**区域单击**新的任务**，选择**Serverless（SAE）应用发布**。



5. 在编辑面板，配置任务信息，单击**保存并运行**。



配置的任务信息说明如下表所示。

参数名	说明
任务名称	自定义的任务名称。不可超过20字符。
构建集群	可为任务选择不同的构建集群。

参数名	说明
选择授权	选择任务的服务授权，使云效能在SAE上部署应用。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
命名空间	选择步骤一所创建的应用所在命名空间。
SAE应用	选择步骤一所创建的应用。
构建产物	选择步骤四所创建的产出物名称。
发布策略	可选择**分批发布**或**灰度发布**。
分批方式	可选择**手动确认**或**自动确认**。例如，如果您需在完成第一批发布时先观察发布结果再决定后续操作，则可选择**手动确认**。
灰度台数	要执行灰度发布的机器数量。 **说明** 此字段仅在**发布策略**为**灰度发布**时显示。
发布批次	发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

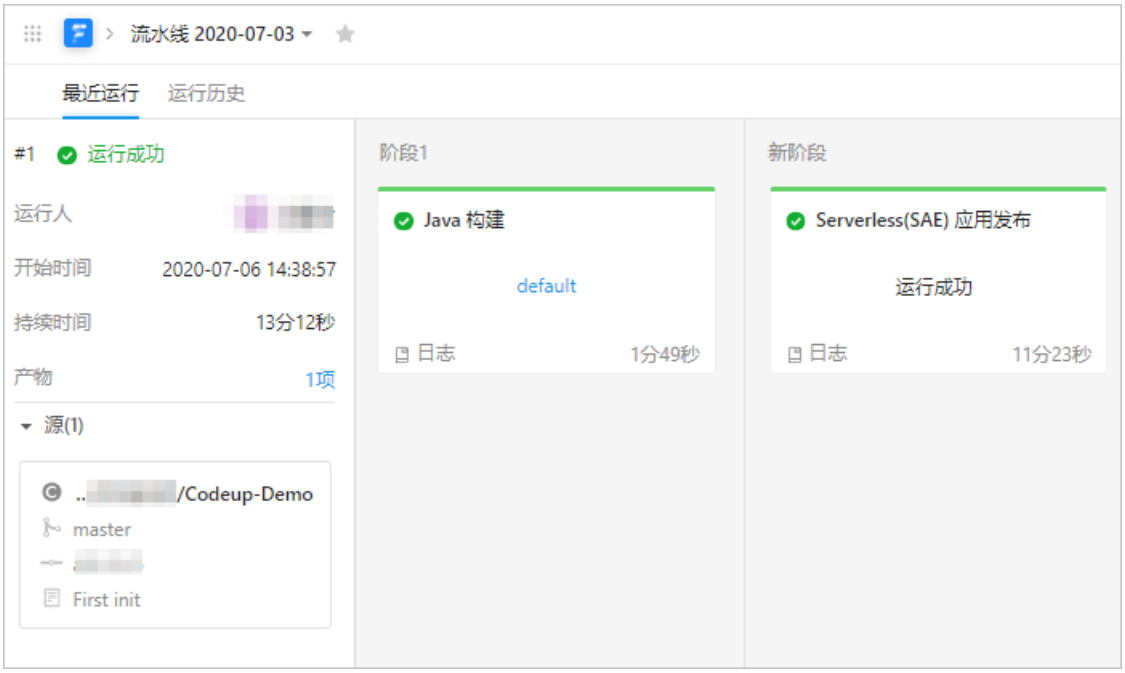
参数名	说明
任务插件（可选）	您可以根据需要配置任务插件来发送流水线通知。 ○ **钉钉机器人通知插件** ■ **webhook地址**：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见钉钉机器人配置。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **邮件通知** ■ **邮件地址**：配置需要发送通知的邮件地址。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **Webhook通知插件** ■ **webhook地址**：配置需要发送通知的webhook地址，地址必须公网可访问。 ■ **运行时机**：您可以选择任务插件发送通知的时机。

6. 在弹出的**运行配置**框中单击**运行**。

云效开始按照配置的流水线工作，最终将应用部署至SAE。

结果验证

- 方法一
  云效开始部署后，默认进入最近运行界面，可查看流水线运行结果。



- 方法二
  云效显示部署成功后，在SAE控制台查看应用的变更记录，是否产生应用重新部署的变更记录。

常见问题

- 如何使用云效以镜像方式部署应用到SAE?
  在SAE控制台以镜像方式创建应用，部署步骤请参见部署Golang应用至SAE。
- 其他编程语言如何使用云效将应用部署到SAE?
  具体部署操作请参见以下文档：
  - 部署Golang应用至SAE
  - 部署Node.js应用至SAE
  - 部署Python应用至SAE
  - 部署PHP应用至SAE
- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 在SAE部署完成后，您可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作，具体操作方式请参见管理应用生命周期。
- 在SAE部署完成后，您可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作，具体操作方式请参见以下文档：
  - 绑定SLB

- 配置弹性伸缩策略
- 一键启停应用
- 配置管理
- 变更实例规格

- 在SAE部署完成后，您还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作，具体操作方式请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE的过程中有任何疑问和建议，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.2.2. 部署Golang应用至SAE

本文介绍如何使用云效以镜像方式将Golang语言的应用部署至SAE。云效现已支持分批发布的发布策略。

前提条件

- 注册阿里云账号。
- 将业务代码上传至代码库。

> 说明
>
> - 云效支持的代码库请参见代码源。
> - 业务代码中须包含应用的Dockerfile文件，具体制作步骤请参见制作应用容器Docker镜像。

- 开通阿里云镜像仓库服务。
- 开通云效。

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE

控制台创建应用，以便云效将业务代码推送到该应用中。

2. 在SAE创建应用完成后，在云效上创建企业并进入流水线FLOW。
3. 根据SAE应用类型配置应用集成发布的流水线信息。
4. 流水线配置完成后，运行流水线完成SAE应用部署。

步骤一：在SAE创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上创建相应的应用。SAE应用支持3种类型，后续流水线配置需要跟应用类型保持一致。 本文以镜像部署为例，镜像部署应用的操作请参见在SAE控制台使用镜像部署PHP应用。

> **注意**
>
> - 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。
> - 除Java应用外，其他语言的应用仅支持**镜像方式**部署。

步骤二：在云效创建企业

如果您是第一次使用云效Flow，则需要在云效上创建您的企业。

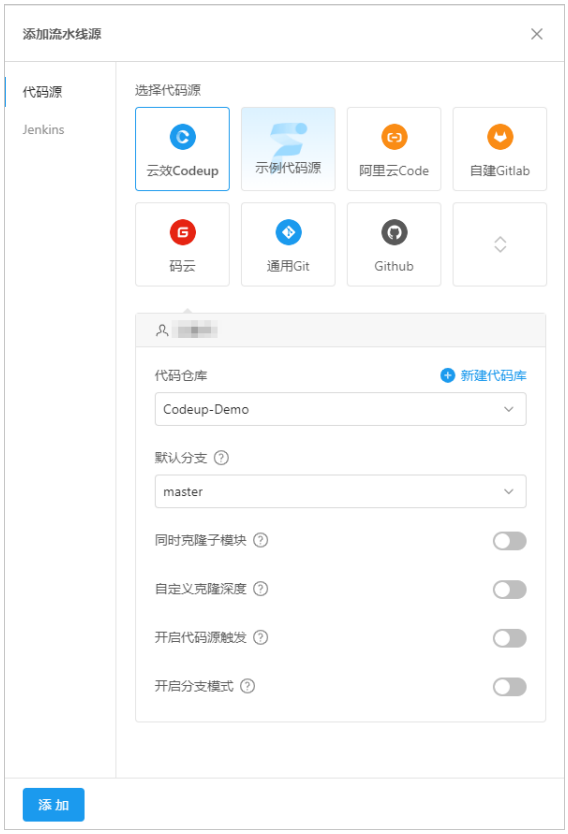
1. 登录云效Flow。
2. 设置企业名称并选择使用规模，单击**完成创建**。

步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在**流水线Flow**页面右上角，单击**新建流水线**。
3. 在**选择流水线模板**对话框，选择**空模板**，单击**创建**。

创建完成后，页面左下角弹出提示框。

4. 流水线创建成功后，默认进入流程配置界面，配置代码源信息，单击**添加**。



- **选择代码源**：选择您业务代码所在的代码源。

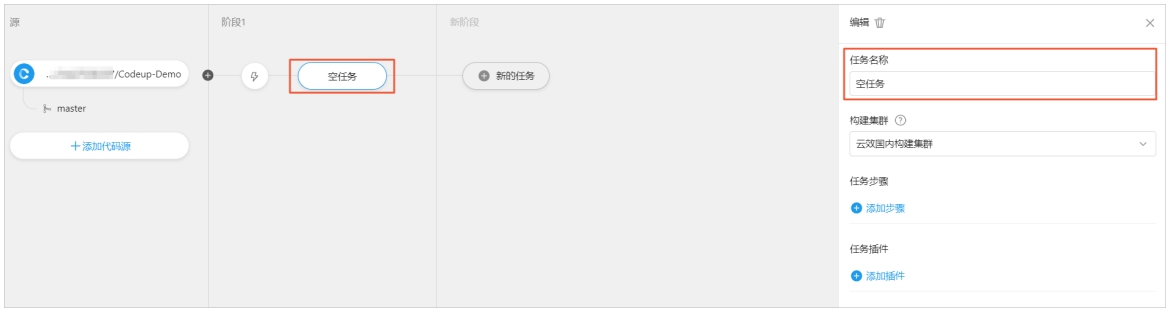
> **说明** 不同代码源需要配置的参数不同，请根据实际界面进行配置。

- **代码仓库**：选择您的代码仓库地址。
- **默认分支**：选择代码分支。

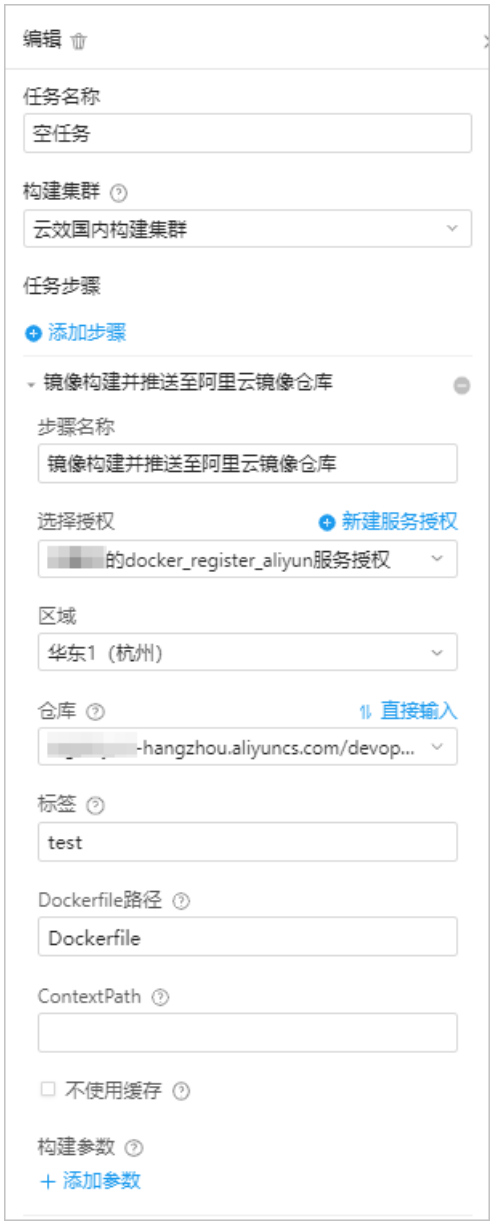
步骤四：在云效部署应用至SAE

本步骤将SAE上的应用升级为含有您业务代码的应用。

1. 在**阶段1**区域单击**空任务**，设置**任务名称**。



2. 单击**添加步骤**，在下拉列表中选择**构建 > 镜像构建并推送至阿里云镜像仓库**，配置镜像信息。

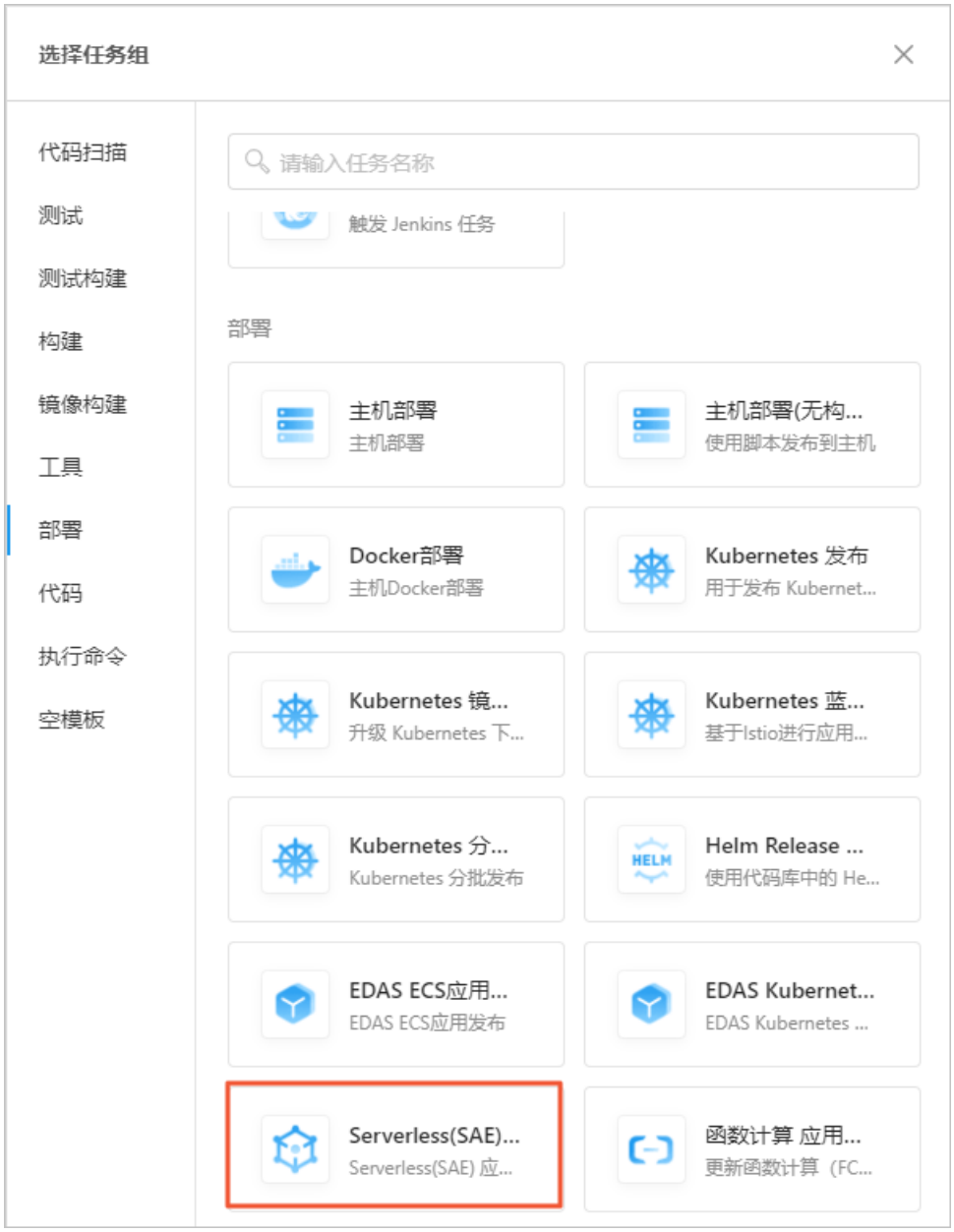


配置的镜像信息说明如下表所示。

参数名	说明
选择授权	选择服务授权，使ACR能够储存镜像。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
仓库	选择镜像构建完成或推送的仓库地址。
标签	自定义的标签名称。

参数名	说明
Dockerfile路径	Dockerfile文件所在路径。
ContextPath	Docker build命令执行上下文路径。不填则默认为Dockerfile路径。
不使用缓存	选择后，docker build将使用no-cache=true参数进行镜像构建。
构建参数	运行构件时会以-build-arg的形式传递到build命令中的一组参数，您可以根据实际需要配置。

3. 在**新阶段**区域单击**新的任务**，选择**Serverless（SAE）应用发布**。



4. 在编辑面板，配置任务信息，单击**保存并运行**。

配置的任务信息说明如下表所示。

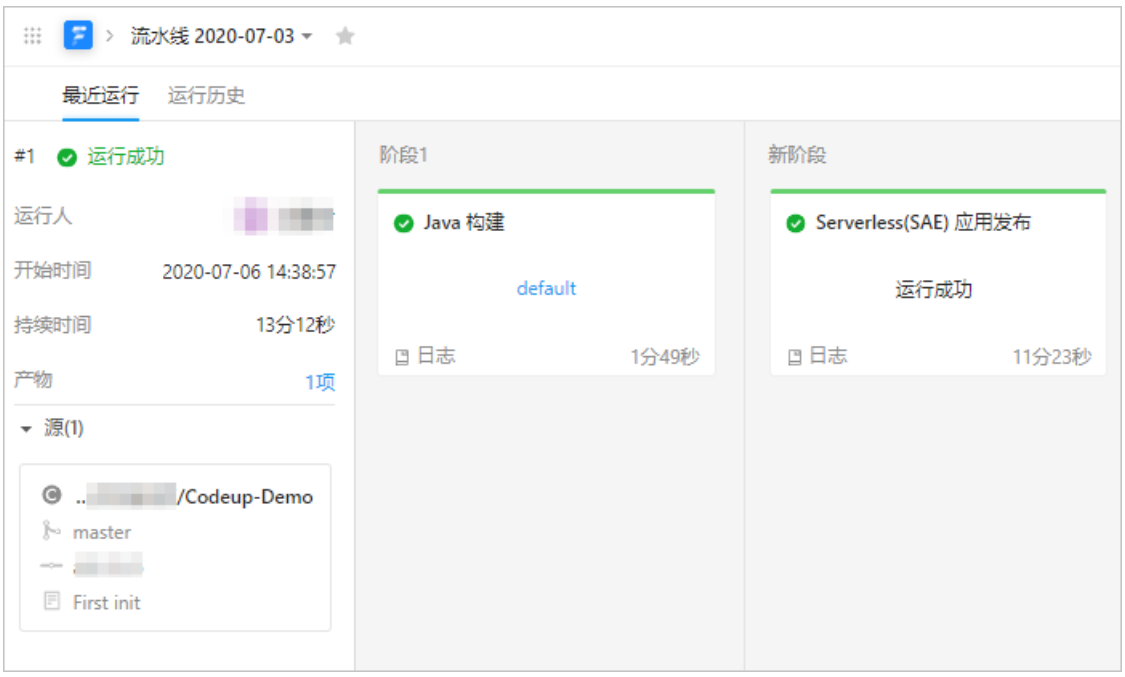
参数名	说明
任务名称	自定义的任务名称。不可超过20字符。
构建集群	可为任务选择不同的构建集群。

参数名	说明
选择授权	选择任务的服务授权，使云效能在SAE上部署应用。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
命名空间	选择步骤一所创建的应用所在命名空间。
SAE应用	选择步骤一所创建的应用。
构建产物	选择步骤四所创建的标签名称。
发布策略	可选择**分批发布**或**灰度发布**。
分批方式	可选择**手动确认**或**自动确认**。例如，如需在完成第一批发布时先观察发布结果再决定后续操作，则可选择**手动确认**。
灰度台数	要执行灰度发布的机器数量。 **说明** 此字段仅在**发布策略**为**灰度发布**时显示。
发布批次	发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

参数名	说明
任务插件	您可以根据需要配置任务插件来发送流水线通知。 ○ **钉钉机器人通知插件** ■ **webhook地址**：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见钉钉机器人配置。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **邮件通知** ■ **邮件地址**：配置需要发送通知的邮件地址。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **Webhook通知插件** ■ **webhook地址**：配置需要发送通知的webhook地址，地址必须公网可访问。 ■ **运行时机**：您可以选择任务插件发送通知的时机。

结果验证

- 方法一
  云效开始部署后，默认进入最近运行界面，可查看流水线运行结果。



- 方法二
  云效显示部署成功后，在SAE控制台查看应用的变更记录，是否产生应用重新部署的变更记录。

常见问题

- 其他编程语言如何使用云效将应用部署到SAE?
  具体部署操作请参见以下文档：
  - 部署Java应用至SAE
  - 部署Node.js应用至SAE
  - 部署Python应用至SAE
  - 部署PHP应用至SAE
- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 在SAE部署完成后，您可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作，具体操作方式请参见管理应用生命周期。
- 在SAE部署完成后，您可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作，具体操作方式请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 在SAE部署完成后，您还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作，具体操作方式请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE的过程中有任何疑问和建议，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.2.3. 部署Node.js应用至SAE

本文介绍如何使用云效以镜像方式将Node.js语言的应用部署至SAE。云效现已支持分批发布的发布策略。

前提条件

- 注册阿里云账号。
- 将业务代码上传至代码库。

> 说明
> - 云效支持的代码库请参见代码源。
> - 业务代码中须包含应用的Dockerfile文件，具体制作步骤请参见制作应用容器Docker镜像。

- 开通阿里云镜像仓库服务。
- 开通云效。

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在SAE创建应用完成后，在云效上创建企业并进入流水线FLOW。
3. 根据SAE应用类型配置应用集成发布的流水线信息。
4. 流水线配置完成后，运行流水线完成SAE应用部署。

步骤一：在SAE创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上创建相应的应用。SAE应用支持3种类型，后续流水线配置需要跟应用类型保持一致。 本文以镜像部署为例，镜像部署应用的操作请参见在SAE控制台使用镜像部署PHP应用。

注意

- 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。
- 除Java应用外，其他语言的应用仅支持**镜像方式**部署。

步骤二：在云效创建企业

如果您是第一次使用云效Flow，则需要在云效上创建您的企业。

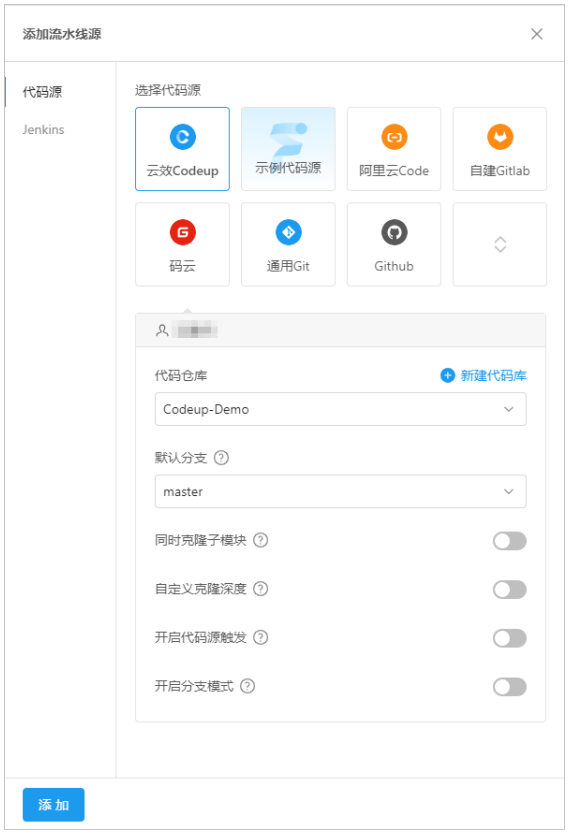
1. 登录云效Flow。
2. 设置企业名称并选择使用规模，单击**完成创建**。

步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在**流水线Flow**页面右上角，单击**新建流水线**。
3. 在**选择流水线模板**对话框，选择**空模板**，单击**创建**。

创建完成后，页面左下角弹出提示框。

4. 流水线创建成功后，默认进入流程配置界面，配置代码源信息，单击**添加**。



- **选择代码源**：选择您业务代码所在的代码源。

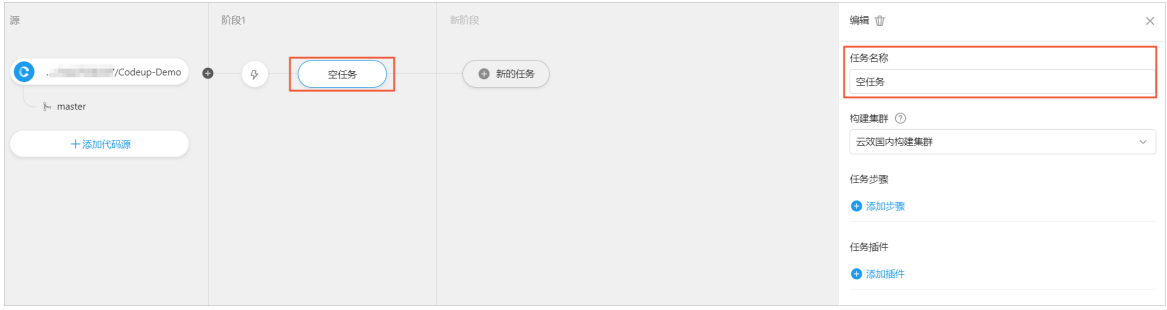
> **说明** 不同代码源需要配置的参数不同，请根据实际界面进行配置。

- **代码仓库**：选择您的代码仓库地址。
- **默认分支**：选择代码分支。

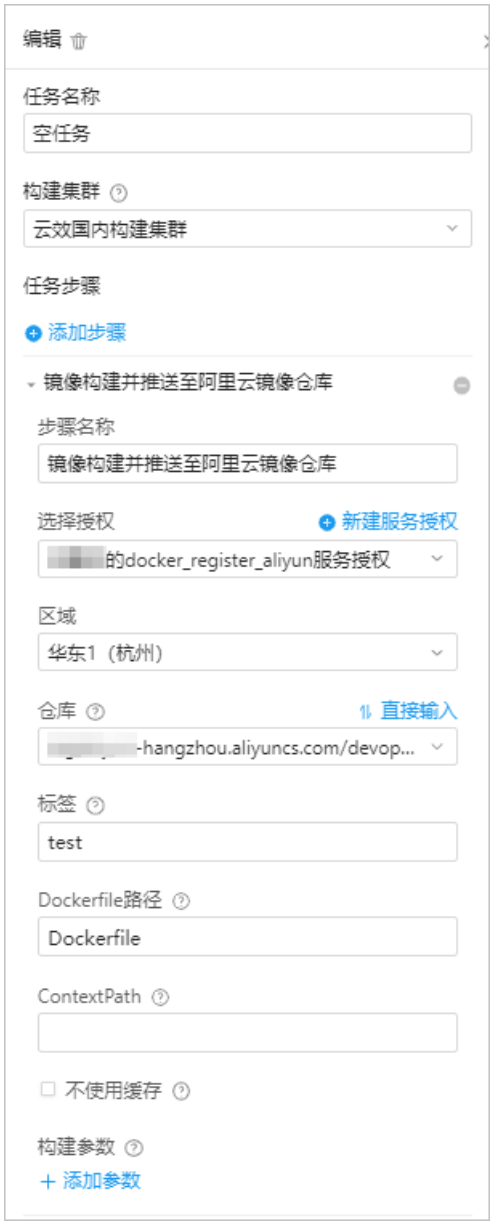
步骤四：在云效部署应用至SAE

本步骤将SAE上的应用升级为含有您业务代码的应用。

1. 在**阶段1**区域单击**空任务**，设置**任务名称**。



2. 单击**添加步骤**，在下拉列表中选择**构建 > 镜像构建并推送至阿里云镜像仓库**，配置镜像信息。

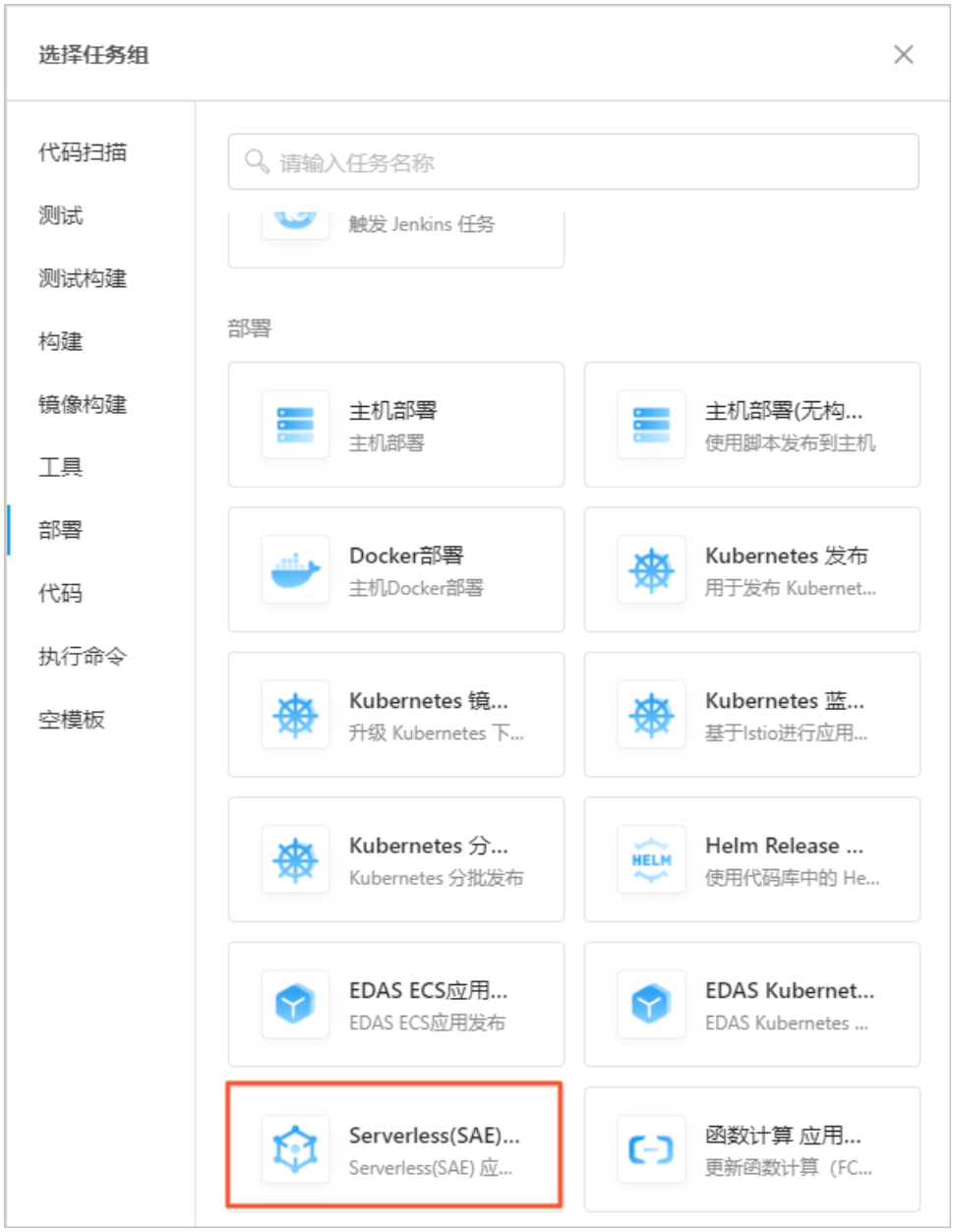


配置的镜像信息说明如下表所示。

参数名	说明
选择授权	选择服务授权，使ACR能够储存镜像。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
仓库	选择镜像构建完成或推送的仓库地址。
标签	自定义的标签名称。

参数名	说明
Dockerfile路径	Dockerfile文件所在路径。
ContextPath	Docker build命令执行上下文路径。不填则默认为Dockerfile路径。
不使用缓存	选择后，docker build将使用no-cache=true参数进行镜像构建。
构建参数	运行构件时会以-build-arg的形式传递到build命令中的一组参数，您可以根据实际需要配置。

3. 在**新阶段**区域单击**新的任务**，选择**Serverless（SAE）应用发布**。



4. 在编辑面板，配置任务信息，单击**保存并运行**。

配置的任务信息说明如下表所示。

参数名	说明
任务名称	自定义的任务名称。不可超过20字符。
构建集群	可为任务选择不同的构建集群。

参数名	说明
选择授权	选择任务的服务授权，使云效能在SAE上部署应用。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
命名空间	选择步骤一所创建的应用所在命名空间。
SAE应用	选择步骤一所创建的应用。
构建产物	选择步骤四所创建的标签名称。
发布策略	可选择**分批发布**或**灰度发布**。
分批方式	可选择**手动确认**或**自动确认**。例如，如需在完成第一批发布时先观察发布结果再决定后续操作，则可选择**手动确认**。
灰度台数	要执行灰度发布的机器数量。 **说明** 此字段仅在**发布策略**为**灰度发布**时显示。
发布批次	发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

参数名	说明
任务插件	您可以根据需要配置任务插件来发送流水线通知。 ○ **钉钉机器人通知插件** ■ **webhook地址**：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见钉钉机器人配置。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **邮件通知** ■ **邮件地址**：配置需要发送通知的邮件地址。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **Webhook通知插件** ■ **webhook地址**：配置需要发送通知的webhook地址，地址必须公网可访问。 ■ **运行时机**：您可以选择任务插件发送通知的时机。

常见问题

- 其他编程语言如何使用云效将应用部署到SAE?
  具体部署操作请参见以下文档：
  - 部署Java应用至SAE
  - 部署Golang应用至SAE
  - 部署Python应用至SAE
  - 部署PHP应用至SAE
- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 在SAE部署完成后，您可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作，具体操作方式请参见管理应用生命周期。
- 在SAE部署完成后，您可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作，具体操作方式请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 在SAE部署完成后，您还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作，具体操作方式请参见以下文档：

- 日志管理
- 监控管理
- 应用事件查看
- 变更记录查看
- 使用Webshell诊断应用

问题反馈

如果您在使用SAE的过程中有任何疑问和建议，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.2.4. 部署Python应用至SAE

本文介绍如何使用云效以镜像方式将Python语言的应用部署至SAE。云效现已支持分批发布的发布策略。

前提条件

- 注册阿里云账号。
- 将业务代码上传至代码库。

> **说明**
>
> - 云效支持的代码库请参见代码源。
> - 业务代码中须包含应用的Dockerfile文件，具体制作步骤请参见制作应用容器Docker镜像。

- 开通阿里云镜像仓库服务。
- 开通云效。

步骤一：在SAE创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上创建相应的应用。SAE应用支持3种类型，后续流水线配置需要跟应用类型保持一致。 本文以镜像部署为例，镜像部署应用的操作请参见在SAE控制台使用镜像部署PHP应用。

> **注意**
>
> - 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。
> - 除Java应用外，其他语言的应用仅支持**镜像方式**部署。

步骤二：在云效创建企业

如果您是第一次使用云效Flow，则需要在云效上创建您的企业。

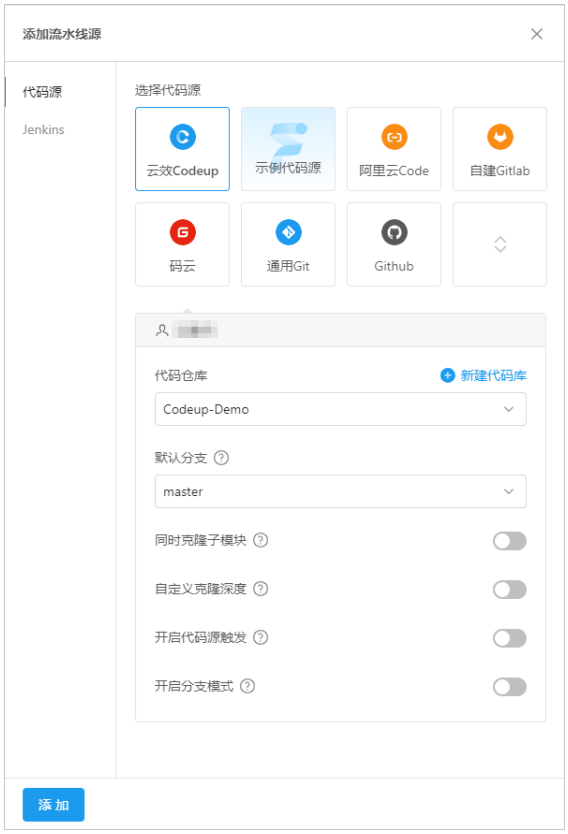
1. 登录云效Flow。
2. 设置企业名称并选择使用规模，单击**完成创建**。

步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在**流水线Flow**页面右上角，单击**新建流水线**。
3. 在**选择流水线模板**对话框，选择**空模板**，单击**创建**。

创建完成后，页面左下角弹出提示框。

4. 流水线创建成功后，默认进入流程配置界面，配置代码源信息，单击**添加**。



- **选择代码源**：选择您业务代码所在的代码源。

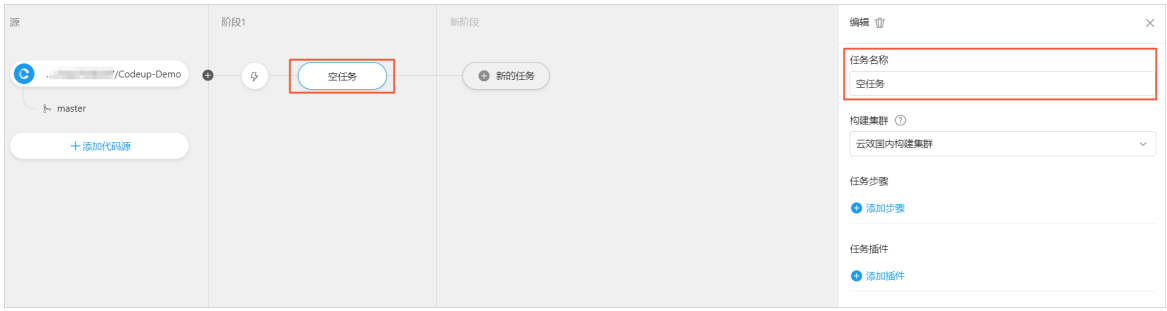
> **说明** 不同代码源需要配置的参数不同，请根据实际界面进行配置。

- **代码仓库**：选择您的代码仓库地址。
- **默认分支**：选择代码分支。

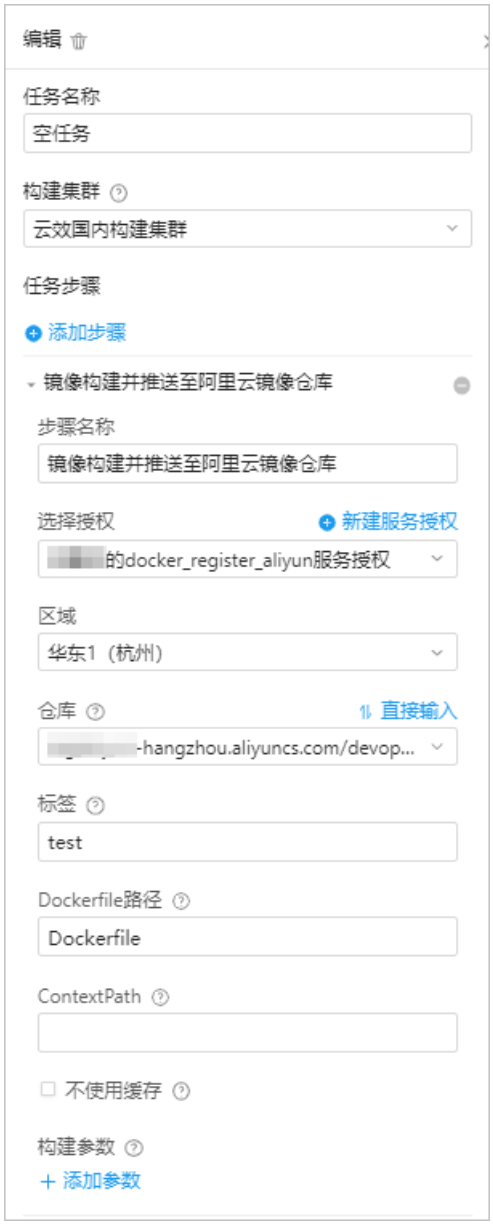
步骤四：在云效部署应用至SAE

本步骤将SAE上的应用升级为含有您业务代码的应用。

1. 在**阶段1**区域单击**空任务**，设置**任务名称**。



2. 单击**添加步骤**，在下拉列表中选择**构建 > 镜像构建并推送至阿里云镜像仓库**，配置镜像信息。

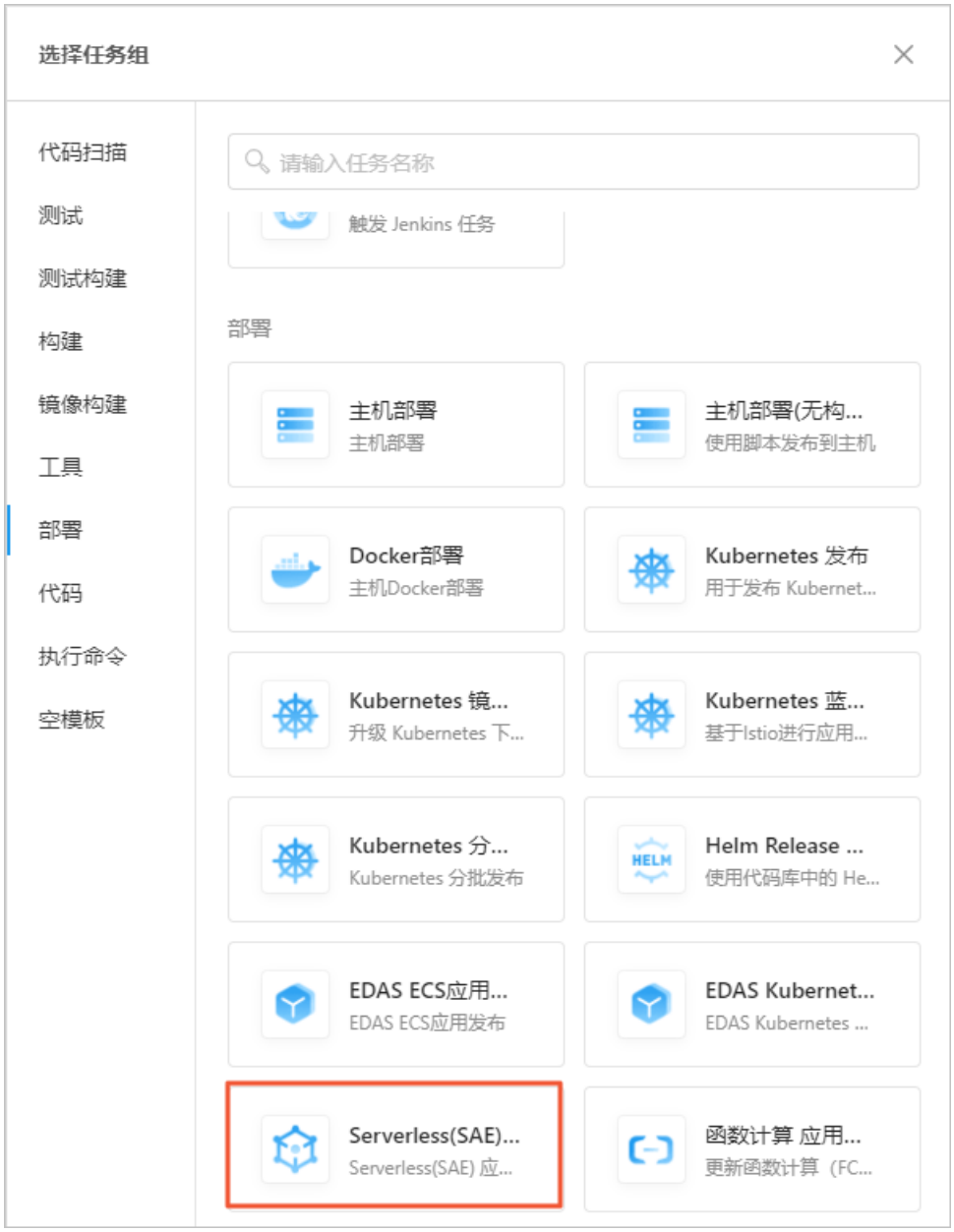


配置的镜像信息说明如下表所示。

参数名	说明
选择授权	选择服务授权，使ACR能够储存镜像。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
仓库	选择镜像构建完成或推送的仓库地址。
标签	自定义的标签名称。

参数名	说明
Dockerfile路径	Dockerfile文件所在路径。
ContextPath	Docker build命令执行上下文路径。不填则默认为Dockerfile路径。
不使用缓存	选择后，docker build将使用no-cache=true参数进行镜像构建。
构建参数	运行构件时会以-build-arg的形式传递到build命令中的一组参数，您可以根据实际需要配置。

3. 在**新阶段**区域单击**新的任务**，选择**Serverless（SAE）应用发布**。



4. 在编辑面板，配置任务信息，单击**保存并运行**。

配置的任务信息说明如下表所示。

参数名	说明
任务名称	自定义的任务名称。不可超过20字符。
构建集群	可为任务选择不同的构建集群。

参数名	说明
选择授权	选择任务的服务授权，使云效能在SAE上部署应用。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
命名空间	选择步骤一所创建的应用所在命名空间。
SAE应用	选择步骤一所创建的应用。
构建产物	选择步骤四所创建的标签名称。
发布策略	可选择**分批发布**或**灰度发布**。
分批方式	可选择**手动确认**或**自动确认**。例如，如需在完成第一批发布时先观察发布结果再决定后续操作，则可选择**手动确认**。
灰度台数	要执行灰度发布的机器数量。 **说明** 此字段仅在**发布策略**为**灰度发布**时显示。
发布批次	发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

参数名	说明
任务插件	您可以根据需要配置任务插件来发送流水线通知。 ○ **钉钉机器人通知插件** ■ **webhook地址**：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见钉钉机器人配置。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **邮件通知** ■ **邮件地址**：配置需要发送通知的邮件地址。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **Webhook通知插件** ■ **webhook地址**：配置需要发送通知的webhook地址，地址必须公网可访问。 ■ **运行时机**：您可以选择任务插件发送通知的时机。

常见问题

- 其他编程语言如何使用云效将应用部署到SAE?
  具体部署操作请参见以下文档：
  - 部署Java应用至SAE
  - 部署Golang应用至SAE
  - 部署Node.js应用至SAE
  - 部署PHP应用至SAE
- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 在SAE部署完成后，您可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作，具体操作方式请参见管理应用生命周期。
- 在SAE部署完成后，您可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作，具体操作方式请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 在SAE部署完成后，您还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作，具体操作方式请参见以下文档：

- 日志管理
- 监控管理
- 应用事件查看
- 变更记录查看
- 使用Webshell诊断应用

4.2.5. 部署PHP应用至SAE

本文介绍如何使用云效以镜像方式将PHP语言的应用部署至SAE。云效现已支持分批发布的发布策略。

前提条件

- 注册阿里云账号。
- 将业务代码上传至代码库。

> 说明
> - 云效支持的代码库请参见代码源。
> - 业务代码中须包含应用的Dockerfile文件，具体制作步骤请参见制作应用容器Docker镜像。

- 开通阿里云镜像仓库服务。
- 开通云效。

部署流程

1. 如果您的应用已经部署在SAE上，则无需创建新的应用。如果您是第一次部署应用到SAE，则需要在SAE控制台创建应用，以便云效将业务代码推送到该应用中。
2. 在SAE创建应用完成后，在云效上创建企业并进入流水线FLOW。
3. 根据SAE应用类型配置应用集成发布的流水线信息。
4. 流水线配置完成后，运行流水线完成SAE应用部署。

步骤一：在SAE创建应用

如果您是第一次使用SAE托管应用，需要预先在SAE上创建相应的应用。SAE应用支持3种类型，后续流水线配置需要跟应用类型保持一致。 本文以镜像部署为例，镜像部署应用的操作请参见在SAE控制台使用镜像部署PHP应用。

> 注意
> - 在SAE控制台创建应用时使用的部署方式（JAR、WAR和镜像），必须与在云效的流水线设置保持一致。
> - 除Java应用外，其他语言的应用仅支持**镜像方式**部署。

步骤二：在云效创建企业

如果您是第一次使用云效Flow，则需要在云效上创建您的企业。

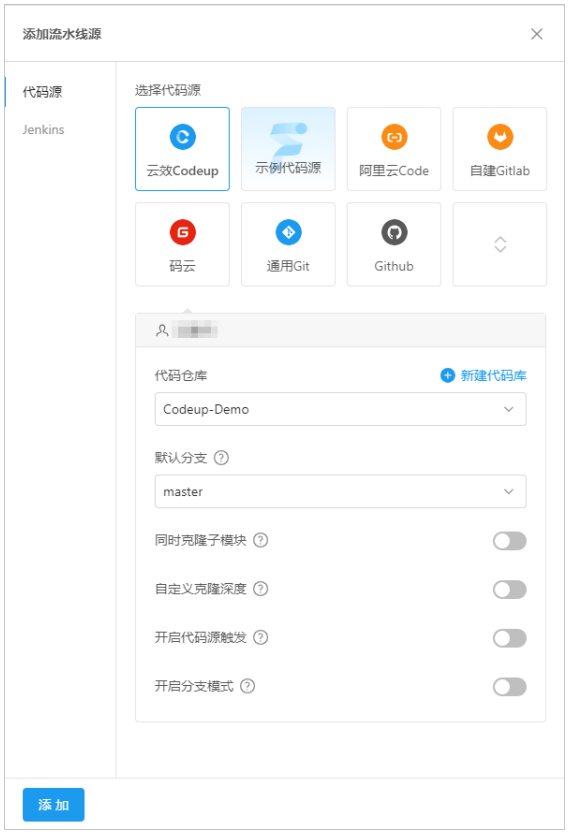
1. 登录云效Flow。
2. 设置企业名称并选择使用规模，单击**完成创建**。

步骤三：在云效创建流水线

1. 登录云效Flow。
2. 在**流水线Flow**页面右上角，单击**新建流水线**。
3. 在**选择流水线模板**对话框，选择**空模板**，单击**创建**。

创建完成后，页面左下角弹出提示框。

4. 流水线创建成功后，默认进入流程配置界面，配置代码源信息，单击**添加**。



- **选择代码源**：选择您业务代码所在的代码源。

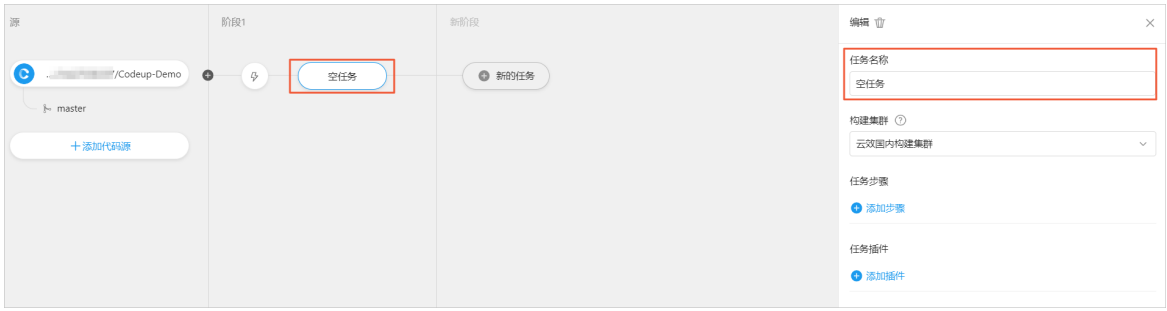
> **说明** 不同代码源需要配置的参数不同，请根据实际界面进行配置。

- **代码仓库**：选择您的代码仓库地址。
- **默认分支**：选择代码分支。

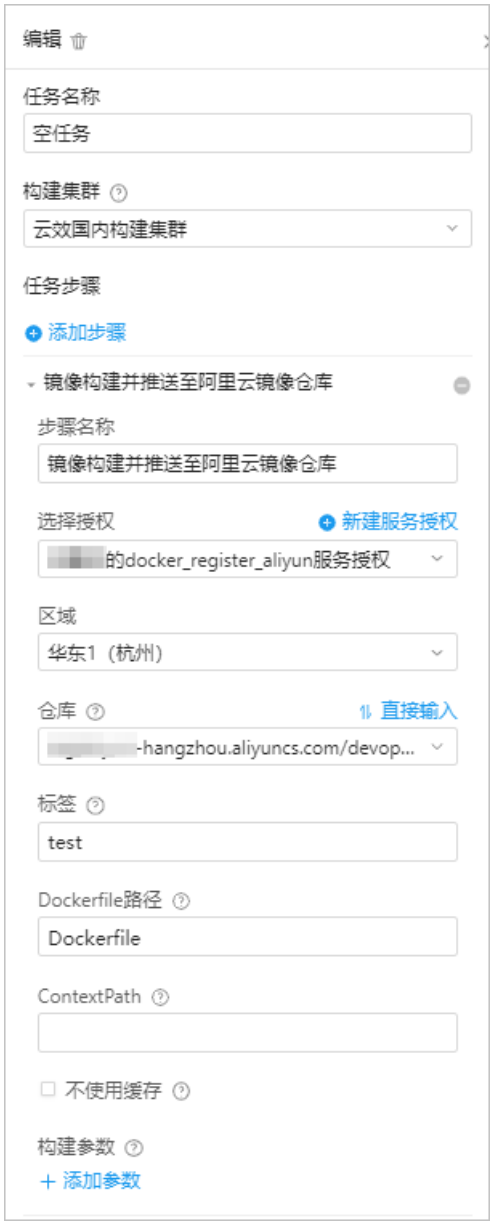
步骤四：在云效部署应用至SAE

本步骤将SAE上的应用升级为含有您业务代码的应用。

1. 在**阶段1**区域单击**空任务**，设置**任务名称**。



2. 单击**添加步骤**，在下拉列表中选择**构建 > 镜像构建并推送至阿里云镜像仓库**，配置镜像信息。

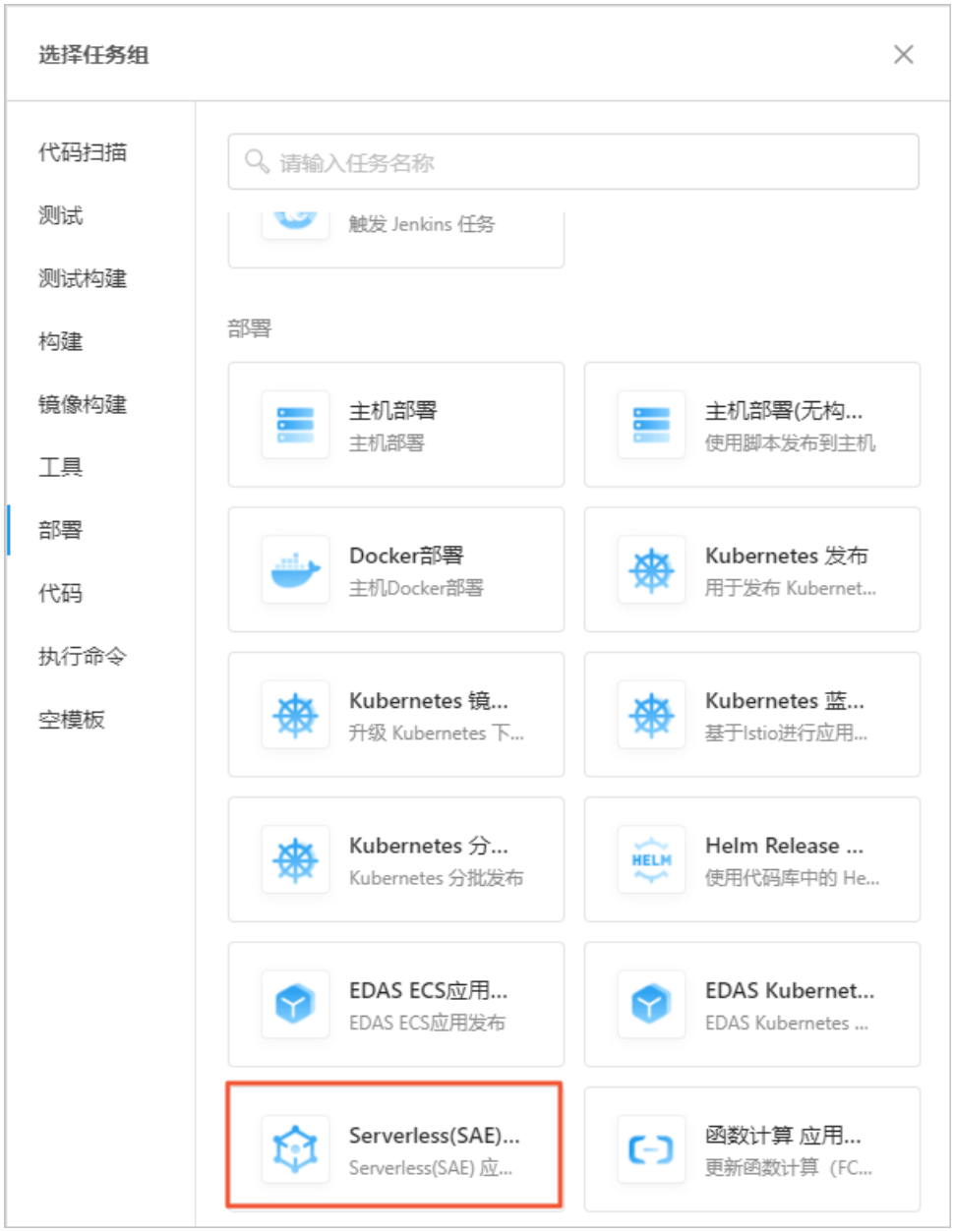


配置的镜像信息说明如下表所示。

参数名	说明
选择授权	选择服务授权，使ACR能够储存镜像。 **说明** 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
仓库	选择镜像构建完成或推送的仓库地址。
标签	自定义的标签名称。

参数名	说明
Dockerfile路径	Dockerfile文件所在路径。
ContextPath	Docker build命令执行上下文路径。不填则默认为Dockerfile路径。
不使用缓存	选择后，docker build将使用no-cache=true参数进行镜像构建。
构建参数	运行构件时会以-build-arg的形式传递到build命令中的一组参数，您可以根据实际需要配置。

3. 在**新阶段**区域单击**新的任务**，选择**Serverless（SAE）应用发布**。



4. 在编辑面板，配置任务信息，单击**保存并运行**。

配置的任务信息说明如下表所示。

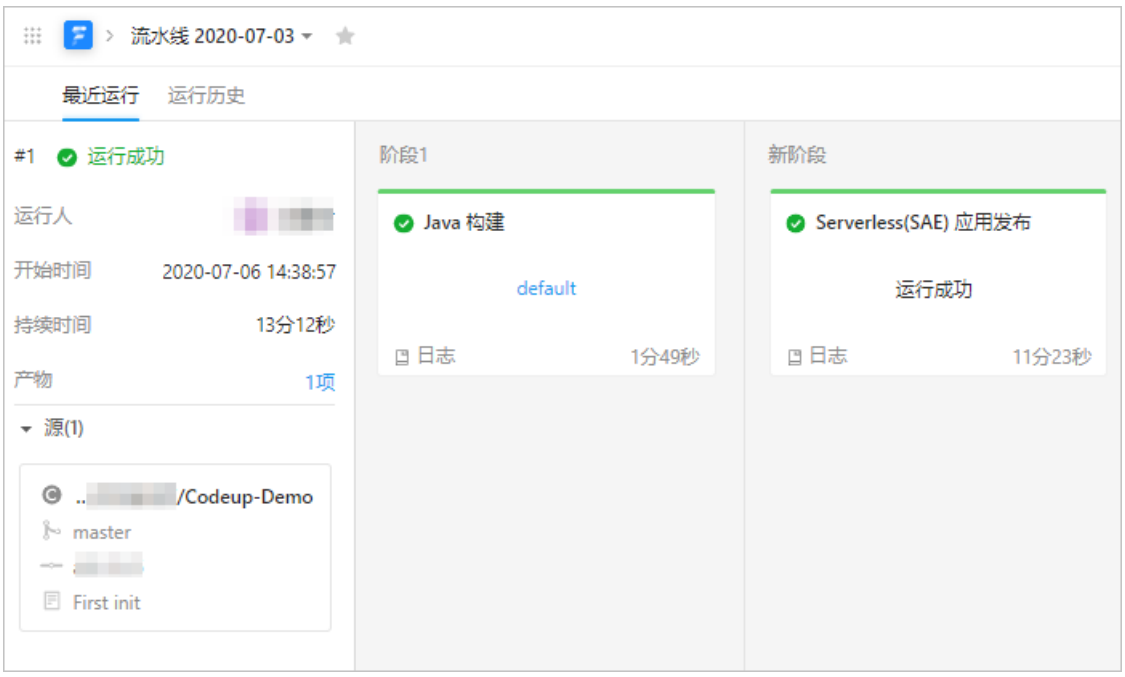
参数名	说明
任务名称	自定义的任务名称。不可超过20字符。
构建集群	可为任务选择不同的构建集群。

参数名	说明
选择授权	选择任务的服务授权，使云效能在SAE上部署应用。 说明 如果您从未授权过，请先单击**新建服务授权**，根据跳转完成阿里云RAM授权后再进行相应配置。
区域	选择步骤一所创建的应用所在地域。
命名空间	选择步骤一所创建的应用所在命名空间。
SAE应用	选择步骤一所创建的应用。
构建产物	选择步骤四所创建的标签名称。
发布策略	可选择**分批发布**或**灰度发布**。
分批方式	可选择**手动确认**或**自动确认**。例如，如需在完成第一批发布时先观察发布结果再决定后续操作，则可选择**手动确认**。
灰度台数	要执行灰度发布的机器数量。 说明 此字段仅在**发布策略**为**灰度发布**时显示。
发布批次	发布分批的数量。
分批等待时间	相邻发布批次之间的等待时间。

参数名	说明
任务插件	您可以根据需要配置任务插件来发送流水线通知。 ○ **钉钉机器人通知插件** ■ **webhook地址**：配置钉钉机器人的webhook地址，钉钉机器人的创建步骤请参见钉钉机器人配置。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **邮件通知** ■ **邮件地址**：配置需要发送通知的邮件地址。 ■ **运行时机**：您可以选择任务插件发送通知的时机。 ○ **Webhook通知插件** ■ **webhook地址**：配置需要发送通知的webhook地址，地址必须公网可访问。 ■ **运行时机**：您可以选择任务插件发送通知的时机。

结果验证

- 方法一
  云效开始部署后，默认进入最近运行界面，可查看流水线运行结果。



- 方法二
  云效显示部署成功后，在SAE控制台查看应用的变更记录，是否产生应用重新部署的变更记录。

常见问题

- 其他编程语言如何使用云效将应用部署到SAE?
  具体部署操作请参见以下文档：
  - 部署Java应用至SAE
  - 部署Golang应用至SAE
  - 部署Node.js应用至SAE
  - 部署Python应用至SAE
- 除Java应用外其他编程语言的应用想要部署在SAE上，在SAE创建应用时使用哪种部署方式（JAR、WAR和镜像）？
  使用**镜像方式**。使用云效部署时，切记您应用程序代码中须包含应用的Dockerfile文件。

更多信息

- 在SAE部署完成后，您可以对应用进行更新、扩缩容、启停、删除应用等生命周期管理操作，具体操作方式请参见管理应用生命周期。
- 在SAE部署完成后，您可以对应用进行自动弹性伸缩、SLB绑定和批量启停等提升应用性能的操作，具体操作方式请参见以下文档：
  - 绑定SLB
  - 配置弹性伸缩策略
  - 一键启停应用
  - 配置管理
  - 变更实例规格
- 在SAE部署完成后，您还可以对应用进行日志管理、监控管理、应用事件查看和变更记录查看等聚焦应用运行状态的操作，具体操作方式请参见以下文档：
  - 日志管理
  - 监控管理
  - 应用事件查看
  - 变更记录查看
  - 使用Webshell诊断应用

问题反馈

如果您在使用SAE的过程中有任何疑问和建议，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

4.3. 使用Jenkins创建应用的持续集成

4.3.1. 使用Jenkins创建Java应用的持续集成

本文介绍使用Jenkins构建SAE Java应用的持续集成。

前提条件

在开始持续集成之前，您需要完成以下准备工作：

- 获取阿里云的AccessKey ID和AccessKey Secret。具体操作，请参见获取AccessKey。
- 在SAE控制台中创建一个可以部署的应用。具体步骤，请参见在SAE控制台使用WAR包部署Java Web应用。
- 使用Git Lab托管您的代码。您可以自行搭建Git Lab或者使用阿里云Code。
  本文使用通过自行搭建的Git Lab做演示，更多信息，请参见Git Lab。
- 安装Maven。具体操作，请参见Maven。
- 安装Jenkins。具体操作，请参见Jenkins官网。

背景信息

您可以使用Jenkins构建SAE应用的持续集成方案。本文适用于对以下语言或工具有一定了解的开发人员。

工具	说明
Maven	Maven是一个项目管理和构建的自动化工具。
Jenkins	Jenkins是一个可扩展的持续集成引擎。
Git Lab	Git Lab是一个利用Ruby on Rails开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。 它拥有与Git Hub类似的功能，能够浏览源代码，管理缺陷和注释。

步骤一：配置项目

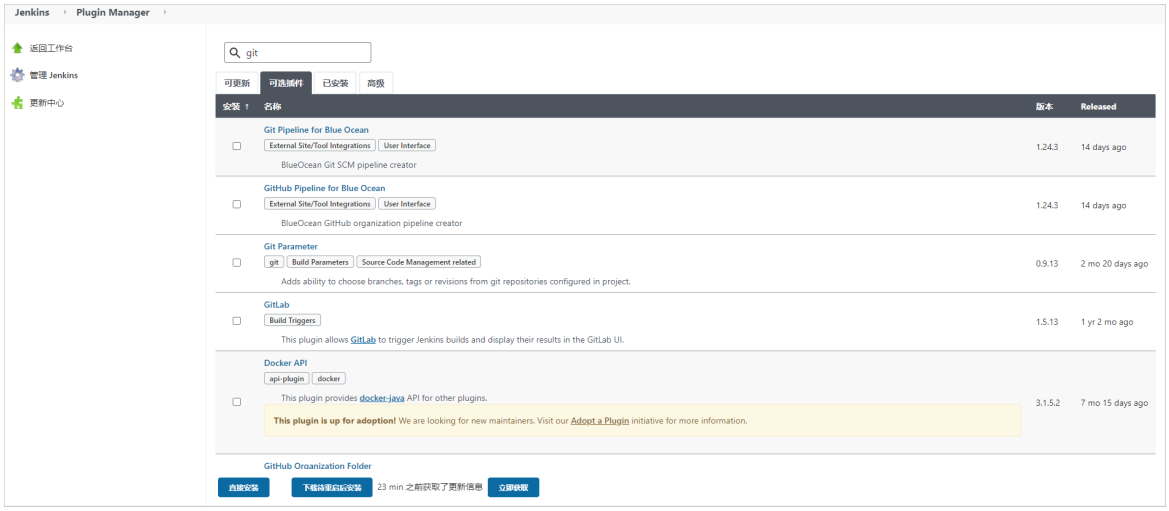
修改Maven项目配置，添加toolkit-maven-plugin及部署信息，具体操作，请参见通过Maven插件自动化部署应用。

> **说明** 修改项目配置后，建议您在本地使用Maven构建验证配置是否正确。

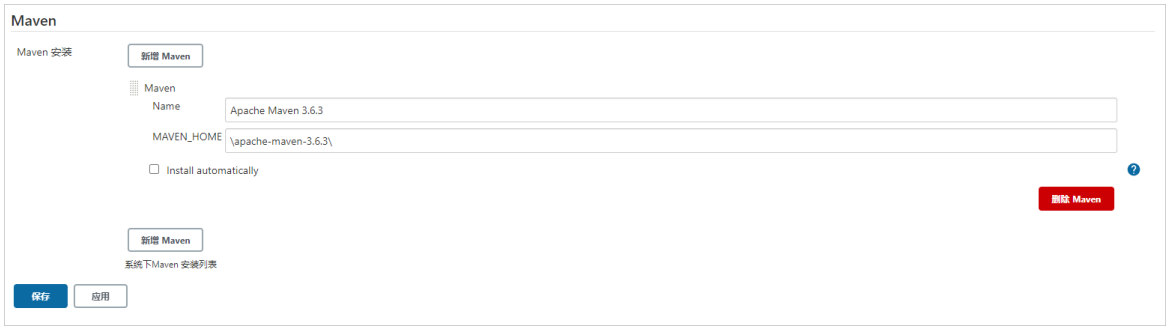
步骤二：配置Jenkins

1. 在Jenkins控制台的菜单栏中选择**Manage Jenkins > Manage Plugins**，安装Git和Git Lab插件。

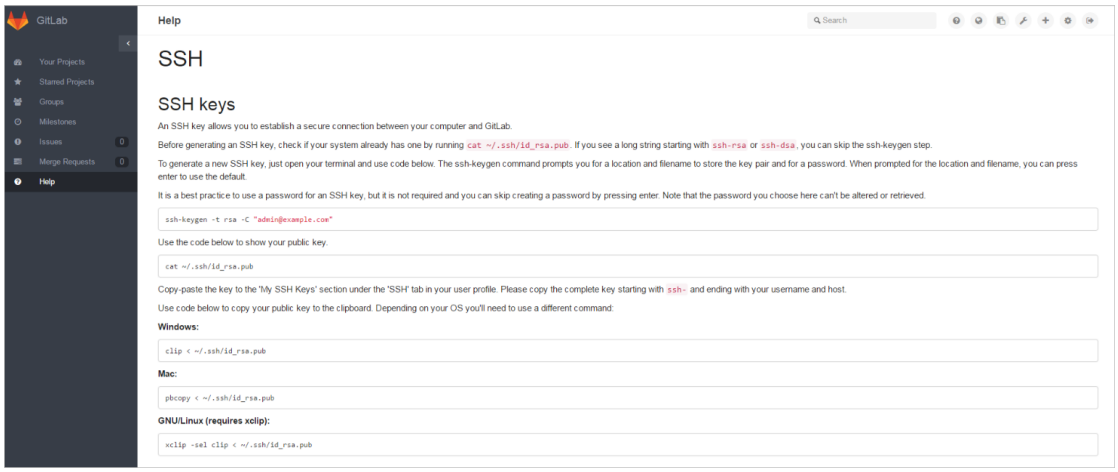
> **说明**
>
> - GIT Client Plugin和GIT Plugin插件可以帮助Jenkins拉取Git仓库中的代码。
> - Gitlab Hook Plugin插件可以帮助Jenkins在收到Gitlab发来的Hook后触发一次构建任务。



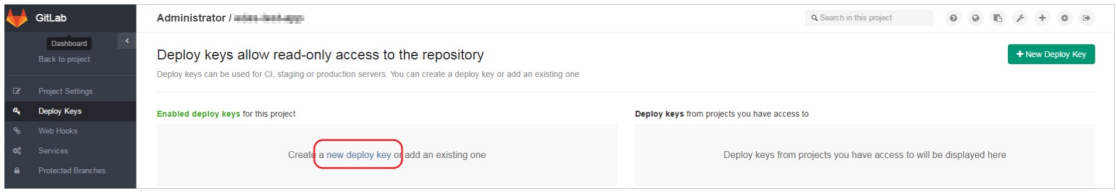
2. 在Jenkins控制台的菜单栏中选择**Manage Jenkins > Global Tool Configuration**，设置Maven版本名称并配置路径，单击**保存**。



3. 在Jenkins服务器上生成SSH RSA密钥对，并将公钥导入GitLab，实现Jenkins拉取GitLab代码时的自动认证。
   i. 在Jenkins服务器生成SSH RSA密钥对。具体信息，请参见GitLab文档。

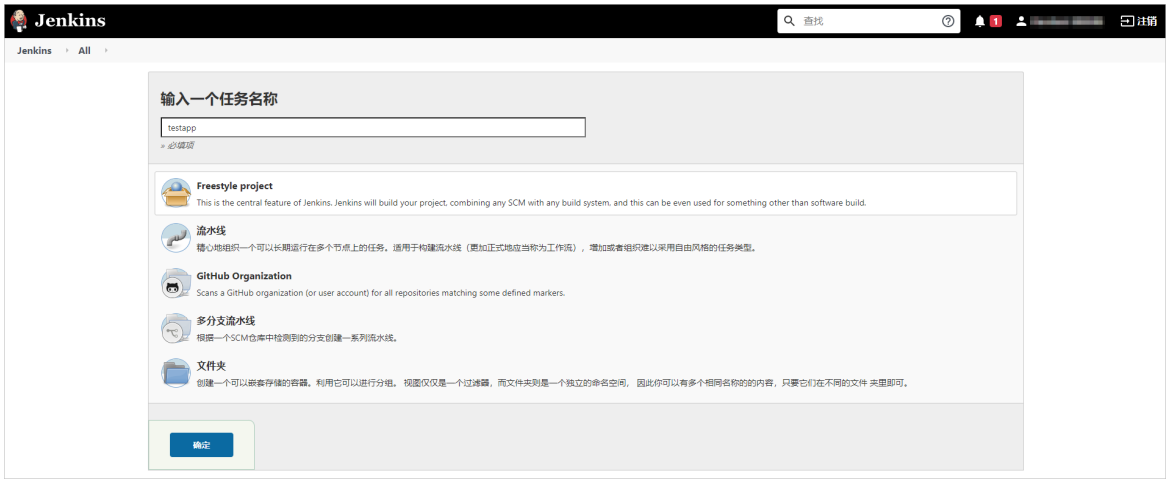


ii. 进入GitLab首页，在菜单栏选择**Settings > Deploy Keys**，并单击**New Deploy Key**，导入在Jenkins服务器上创建的SSH RSA公钥。



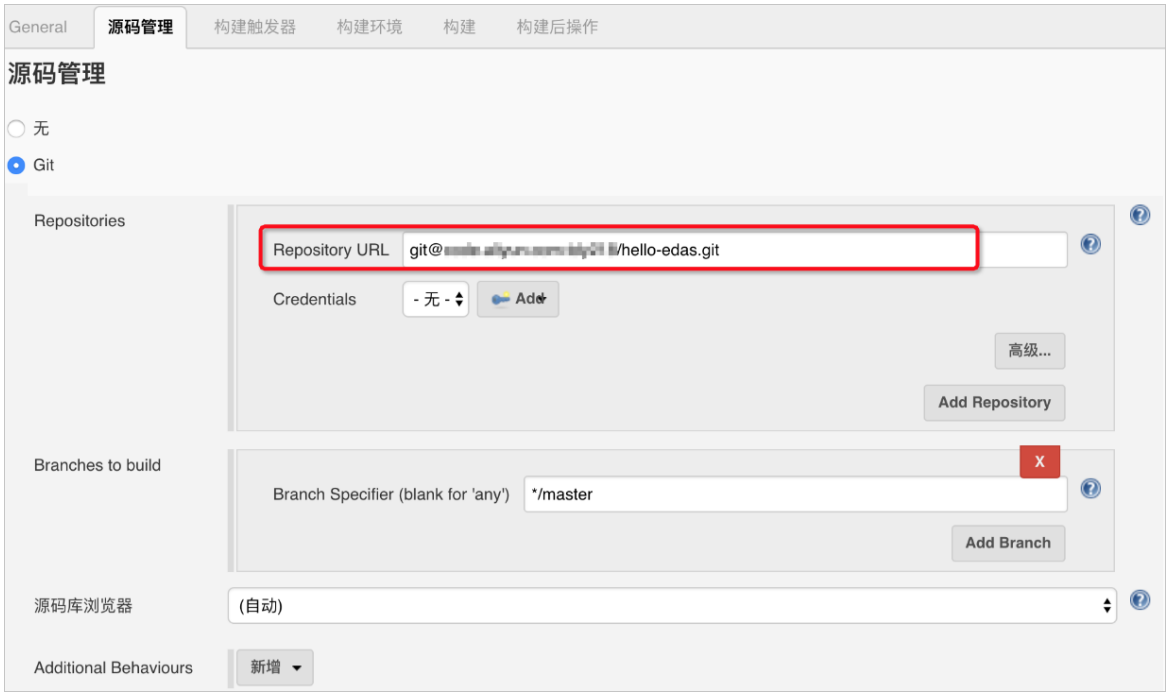
步骤三：创建Jenkins任务

1. 在Jenkins首页左侧导航栏中单击**新建Item**，在创建任务界面输入任务名称，并选择**Freestyle project**，单击**确定**，配置任务信息。

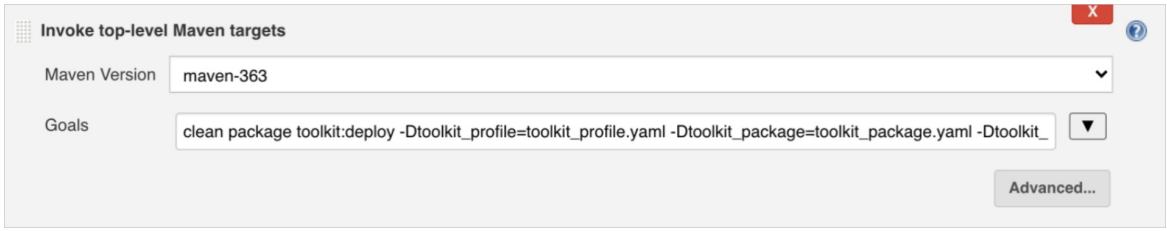


2. 单击**源码管理**，在**源码管理**页签中选择**Git**，并设置相关参数。
   - **Repository URL**：您的项目的Git协议地址。
   - **Credentials**：安全凭证，选择**无**即可。

> **注意** 请确保您的SSH RSA公匙已添加到该Git项目所在的GitLab中，否则将会报错。



3. 单击**构建触发器**，在**构建触发器**页签选中GitHub hook trigger for GITScm polling。
4. 单击**构建环境**，在**构建环境**页签选中Add timestamps to the Console Output，为控制台输出的信息添加时间戳。
5. 单击**构建**，在**构建**页签单击**增加构建步骤**，在下拉列表中选择Invoke top-level Maven targets。
6. 在Invoke top-level Maven targets区域设置Maven Version和Goals。如果您想部署多模块工程，请参见（可选）创建多模块工程的Jenkins任务。
   - **Maven Version**：单击该选项后面的下拉框，选择在**全局工具配置**里配置的**Maven版本名称**。
   - **Goals**：输入mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml（如有其它参数，请根据实际情况输入）。

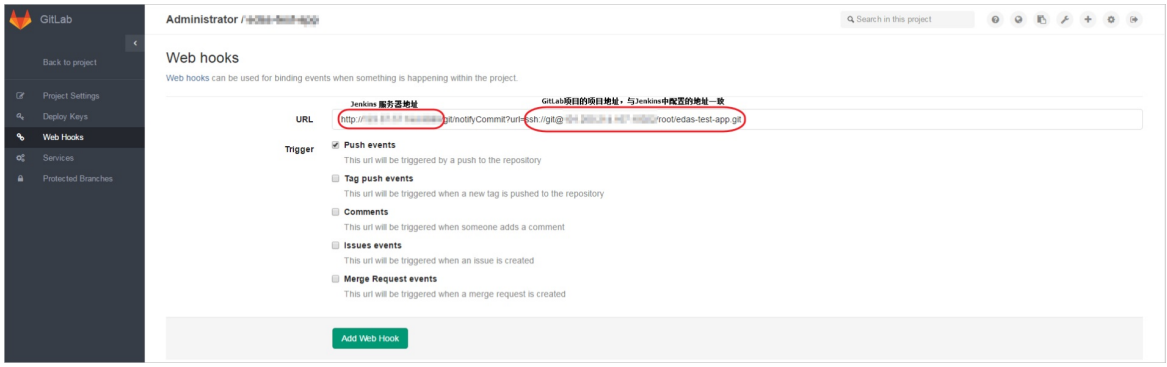


> **说明** Maven配置完成后，在步骤五：提交变更到GitLab步骤中便可通过POP API方式触发应用部署。

步骤四：配置GitLab的Web Hook

1. 在Gitlab首页右键单击GitLab工程，然后选择**Setting > Web Hooks**。
2. 在**Web Hooks**页面的**URL**区域中输入 `http://jenkins服务器地址:jenkins服务器监听端口/git/notifyCommit?url=本项目的git协议地址` 。

   图中表示的Jenkins服务器地址为您的Jenkins服务器的Web访问地址如 `192.168.XX.XX:8080` 。



3. 配置完成后，单击**Test Hook**，测试配置结果。

步骤五：提交变更到GitLab

如果上述步骤配置正确，提交后将会触发一次Git Lab Hook。Jenkins在接收到该Hook后会构建您的Maven项目，并在构建结束时调用SAE POP API脚本触发部署。

> **说明** 构建的Maven项目中配置了通过SAE POP API方式部署应用的脚本。

提交部署成功输出的日志信息（**Build Number > 控制台输出**）。

```
15:58:51 [INFO] Deploy application successfully!
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 [INFO] BUILD SUCCESS
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 [INFO] Total time: 24.330 s
15:58:51 [INFO] Finished at: 2018-12-25T15:58:51+08:00
15:58:51 [INFO] Final Memory: 23M/443M
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 Finished: SUCCESS
```

如果部署失败，您可以登录SAE控制台，查看此次部署任务的执行过程。具体步骤，请参见查看变更记录。

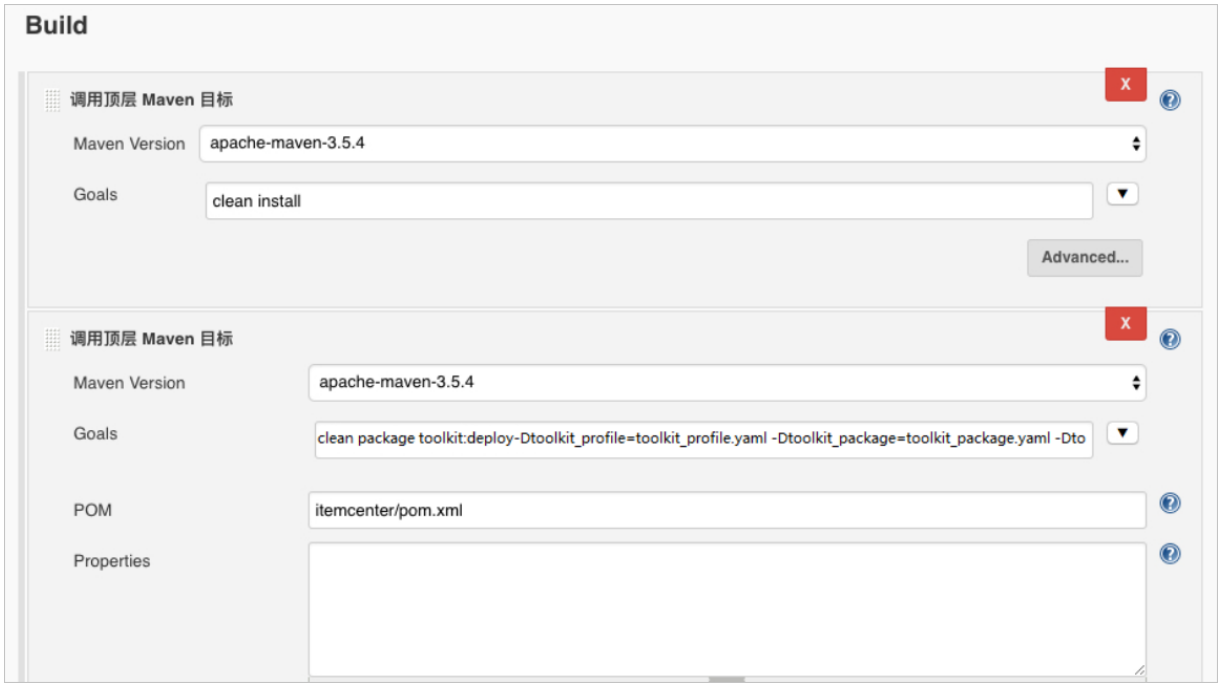
（可选）创建多模块工程的Jenkins任务

如果您需要创建多模块工程的Jenkins任务，您可以参考以下内容设置。

创建多模块工程的Jenkins任务和步骤二：配置Jenkins的第5步基本相同，只需要修改**调用顶层Maven目标**。如果工程为多模块工程，想在Jenkins中部署子模块的话，那么需要在父模块中调用 `mvn clean install` 命令，然后在子模块中调用 `mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml` 命令。以Demo工程为例，工程结构如下。

```
sh-3.2# tree -L 1 carshop
carshop
├── detail
├── itemcenter
├── itemcenter-api
└── pom.xml
```

其中，detail、itemcenter、itemcenter-api为子模块，如果您想部署itemcenter模块的话，需要在父工程中设置一个clean install构建目标，然后在itemcenter模块中设置 `clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml` 构建目标。



4.3.2. 使用Jenkins创建多语言应用的持续集成

本文介绍使用Jenkins通过镜像方式创建SAE Golang、Node.js、Python、PHP等多语言应用的持续集成。

前提条件

在开始持续集成之前，您需要完成以下准备工作：

- 获取阿里云的AccessKey ID和AccessKey Secret。具体操作，请参见获取AccessKey。
- 在SAE控制台中创建一个可以部署的应用。具体步骤，请参见在SAE控制台使用WAR包部署Java Web应用。
- 使用GitLab托管您的代码。您可以自行搭建GitLab或者使用阿里云Code。
  本文使用通过自行搭建的GitLab做演示，更多信息，请参见GitLab。
- 安装Maven。具体操作，请参见Maven。
- 安装Jenkins。具体操作，请参见Jenkins官网。

背景信息

您可以使用Jenkins构建SAE应用的持续集成方案。本文适用于对以下语言或工具有一定了解的开发人员。

工具	说明
Maven	Maven是一个项目管理和构建的自动化工具。
Jenkins	Jenkins是一个可扩展的持续集成引擎。

工具	说明
GitLab	GitLab是一个利用Ruby on Rails开发的开源应用程序，实现一个自托管的Git项目仓库，可通过Web界面进行访问公开的或者私人项目。 它拥有与GitHub类似的功能，能够浏览源代码，管理缺陷和注释。

步骤一：配置项目

1. 修改Maven项目配置，填写Dockerfile路径以及仓库地址。具体操作，请参见通过Maven插件自动化部署应用。

   配置文件示例如下：

   - 打包文件

```
apiVersion: V1
kind: AppPackage
spec:
 packageType: Image
 build:
  docker:
    dockerfile: Dockerfile #指定Dockerfile。
    imageRepoAddress:    #镜像仓库地址。
    imageTag:          #镜像Tag。
    imageRepoUser:      #镜像仓库用户名。
    imageRepoPassword:   #镜像仓库密码。
```

   - 部署文件

```
apiVersion: V1
kind: AppDeployment
spec:
 type: serverless
 target:
  appId:     #部署应用的ID。如果配置了该参数则无需配置namespaceId和appName。
  namespaceId: #应用所属命名空间。如果您不清楚应用ID，可使用应用所属命名空间及应用名称进行部署。
  appName:    #应用名称。如果您不清楚应用ID，可使用应用名称及命名空间进行部署。
```

2. 在您的打包工程中添加*pom.xml*文件，以便添加插件依赖。

   *pom.xml*文件示例如下：

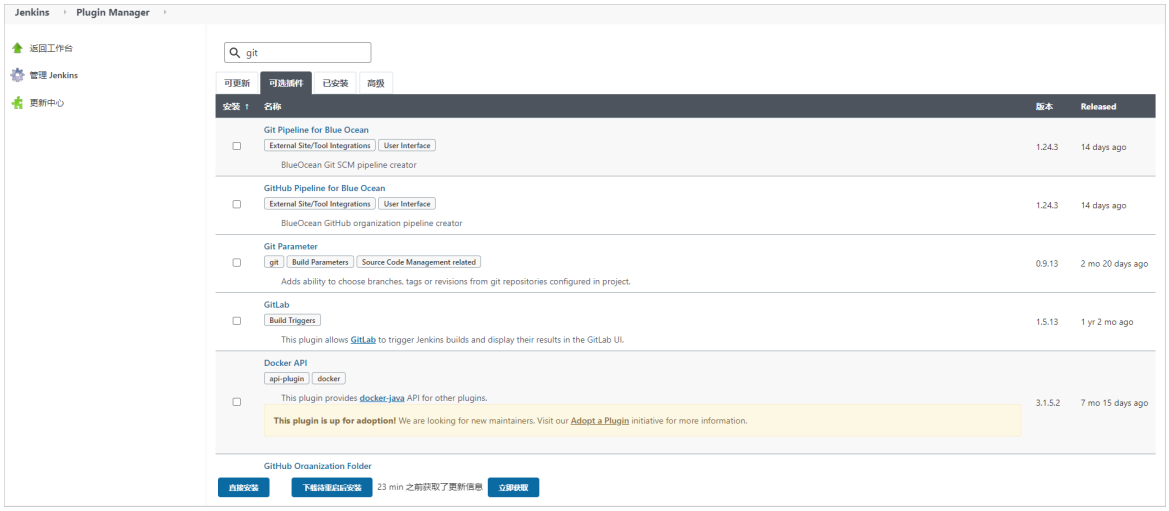
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.
xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>demo</groupId>
  <artifactId>sae-demo</artifactId>
  <version>1.0.0</version>
  <build>
    <plugins>
          <plugin>
            <groupId>com.alibaba.cloud</groupId>
            <artifactId>toolkit-maven-plugin</artifactId>
            <version>1.1.2</version>
          </plugin>
    </plugins>
  </build>
</project>
```

说明 修改项目配置后，建议您在本地使用Maven构建验证配置是否正确。

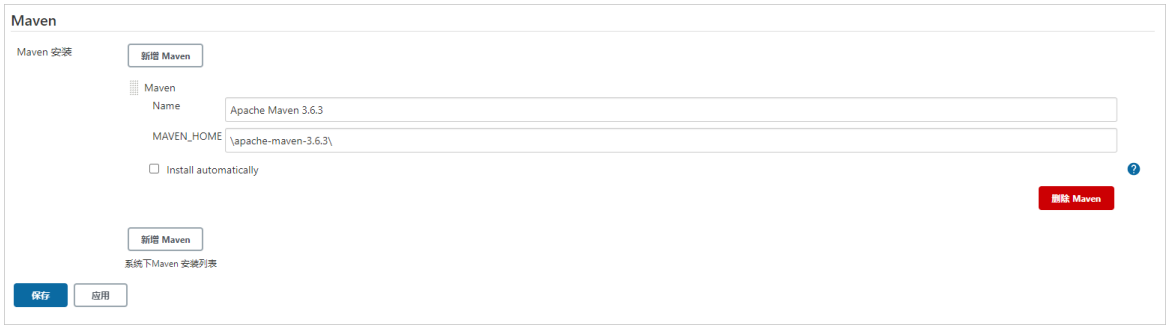
步骤二：配置Jenkins

1. 在Jenkins控制台的菜单栏中选择**Manage Jenkins > Manage Plugins**，安装Git和Git Lab插件。

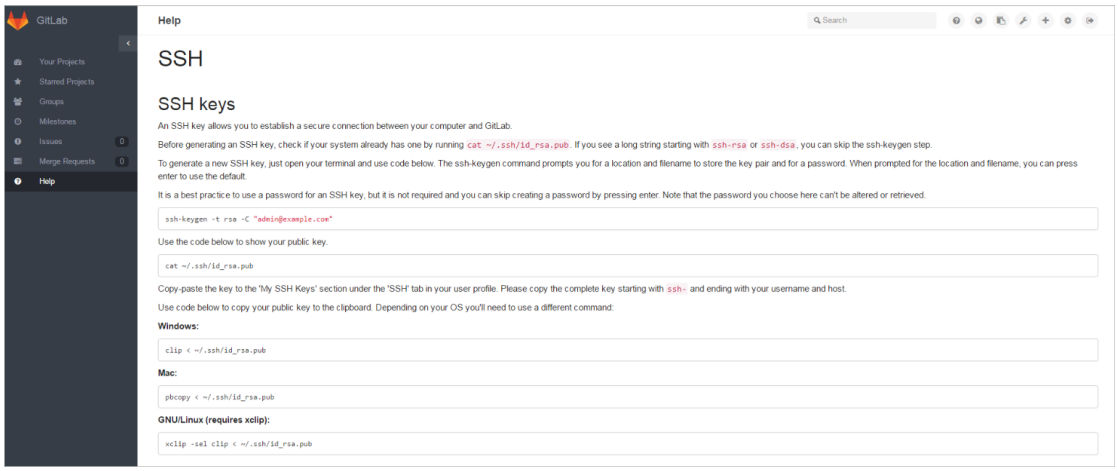
   说明
   - GIT Client Plugin和GIT Plugin插件可以帮助Jenkins拉取Git仓库中的代码。
   - Gitlab Hook Plugin插件可以帮助Jenkins在收到Gitlab发来的Hook后触发一次构建任务。



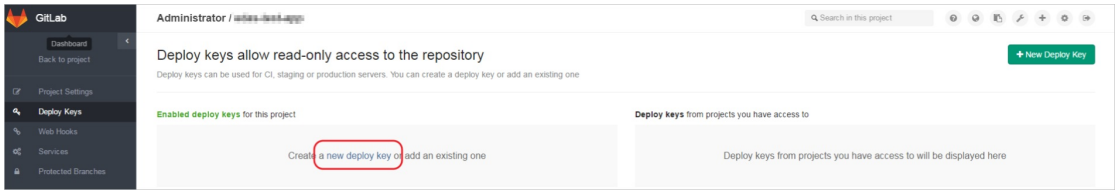
2. 在Jenkins控制台的菜单栏中选择**Manage Jenkins > Global Tool Configuration**，设置Maven版本名称并配置路径，单击**保存**。



3. 在Jenkins服务器上生成SSH RSA密钥对，并将公钥导入GitLab，实现Jenkins拉取GitLab代码时的自动认证。
   i. 在Jenkins服务器生成SSH RSA密钥对。具体信息，请参见GitLab文档。

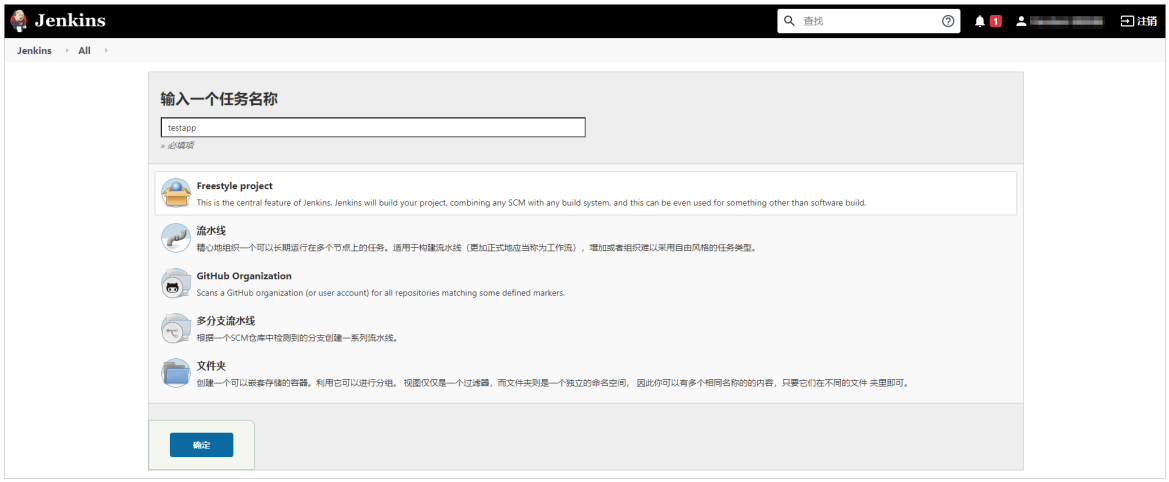


   ii. 进入GitLab首页，在菜单栏选择**Settings > Deploy Keys**，并单击**New Deploy Key**，导入在Jenkins服务器上创建的SSH RSA公钥。



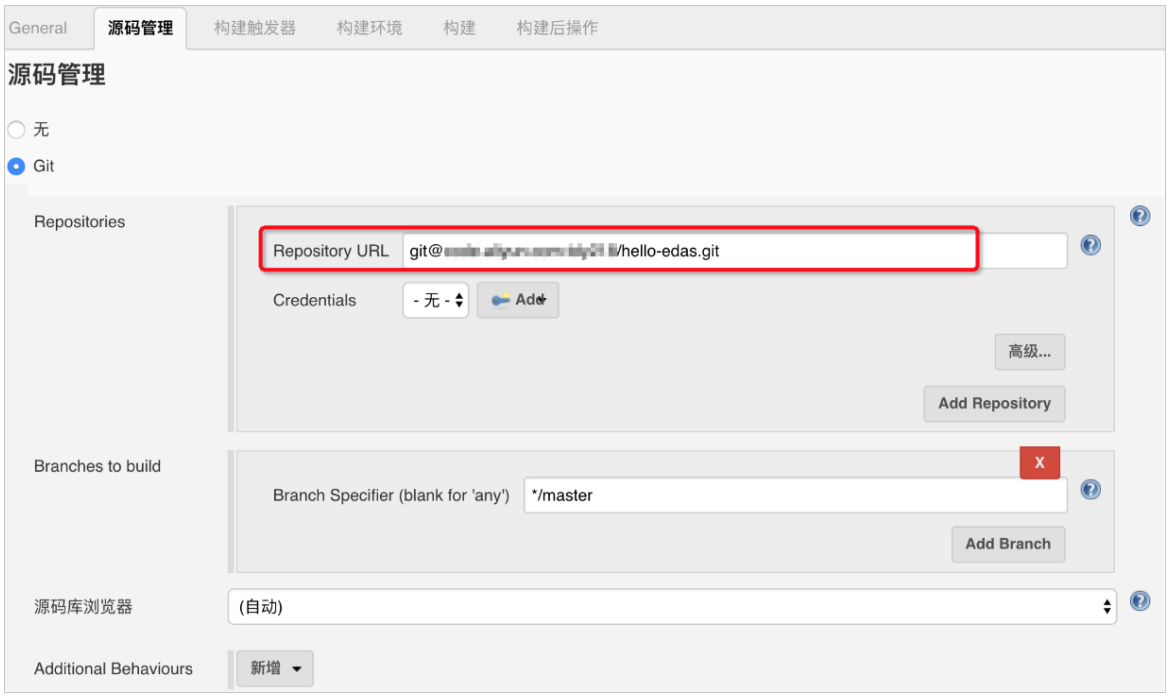
步骤三：创建Jenkins任务

1. 在Jenkins首页左侧导航栏中单击**新建Item**，在创建任务界面输入任务名称，并选择**Freestyle project**，单击**确定**，配置任务信息。



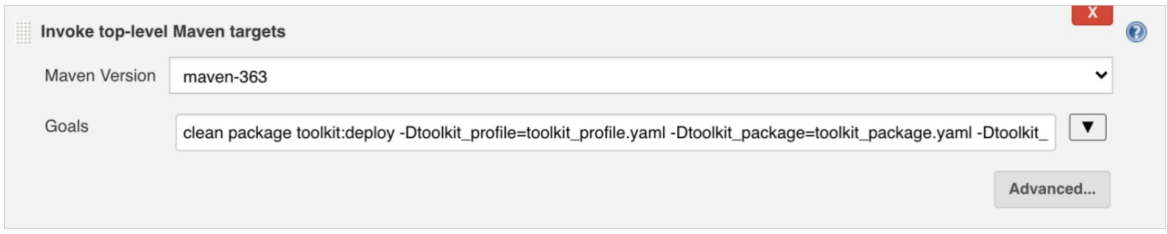
2. 单击**源码管理**，在**源码管理**页签中选择Git，并设置相关参数。
   - Repository URL：您的项目的Git协议地址。
   - Credentials：安全凭证，选择**无**即可。

> **注意** 请确保您的SSH RSA公匙已添加到该Git项目所在的GitLab中，否则将会报错。



3. 单击**构建触发器**，在**构建触发器**页签选中GitHub hook trigger for GITScm polling。
4. 单击**构建环境**，在**构建环境**页签选中Add timestamps to the Console Output，为控制台输出的信息添加时间戳。
5. 单击**构建**，在**构建**页签单击**增加构建步骤**，在下拉列表中选择Invoke top-level Maven targets。
6. 在Invoke top-level Maven targets区域设置Maven Version和Goals。如果您想部署多模块工程，请参见（可选）创建多模块工程的Jenkins任务。
   - Maven Version：单击该选项后面的下拉框，选择在**全局工具配置**里配置的**Maven版本名称**。
   - Goals：输入mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml （如有其它参

数，请根据实际情况输入）。

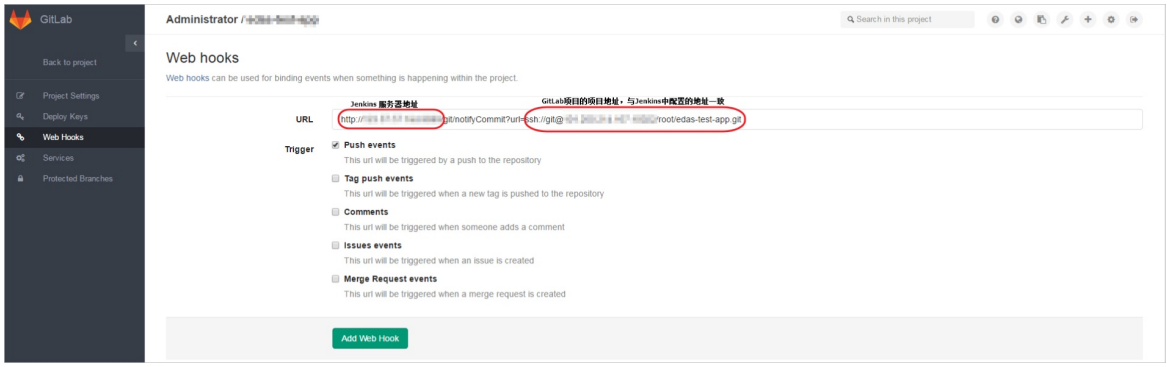


> **说明** Maven配置完成后，在步骤五：提交变更到GitLab中便可通过POP API方式触发应用部署。

步骤四：配置GitLab的Web Hook

1. 在Gitlab首页右键单击GitLab工程，然后选择**Setting > Web Hooks**。
2. 在**Web Hooks**页面的**URL**区域中输入 `http://jenkins服务器地址:jenkins服务器监听端口/git/notifyCommit?url=本项目的git协议地址` 。

   图中表示的Jenkins服务器地址为您的Jenkins服务器的Web访问地址如 `192.168.XX.XX:8080` 。



3. 配置完成后，单击**Test Hook**，测试配置结果。

步骤五：提交变更到GitLab

如果上述步骤配置正确，提交后将会触发一次GitLab Hook。Jenkins在接收到该Hook后会构建您的Maven项目，并在构建结束时调用SAE POP API脚本触发部署。

> **说明** 构建的Maven项目中配置了通过SAE POP API方式部署应用的脚本。

提交部署成功输出的日志信息（**Build Number > 控制台输出**）。

```
15:58:51 [INFO] Deploy application successfully!
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 [INFO] BUILD SUCCESS
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 [INFO] Total time: 24.330 s
15:58:51 [INFO] Finished at: 2018-12-25T15:58:51+08:00
15:58:51 [INFO] Final Memory: 23M/443M
15:58:51 [INFO] ------------------------------------------------------------------------
15:58:51 Finished: SUCCESS
```

如果部署失败，您可以登录SAE控制台，查看此次部署任务的执行过程。具体步骤，请参见查看变更记录。

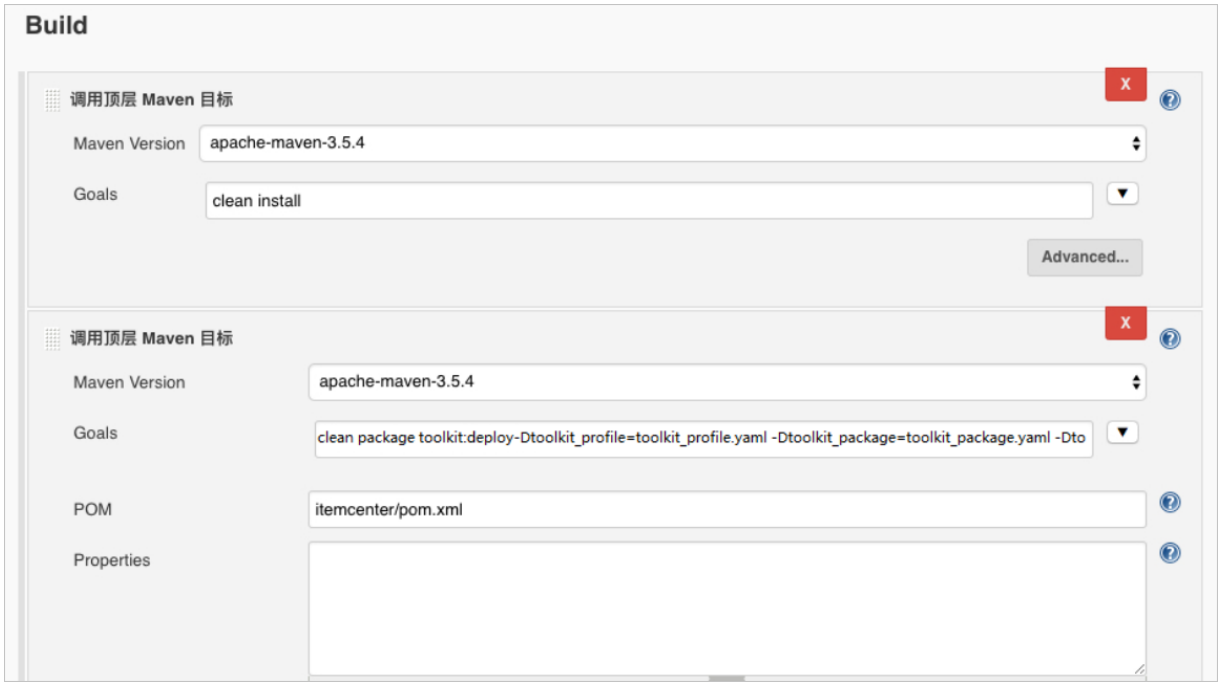
（可选）创建多模块工程的Jenkins任务

如果您需要创建多模块工程的Jenkins任务，您可以参考以下内容设置。

创建多模块工程的Jenkins任务和步骤二：配置Jenkins的第5步基本相同，只需要修改**调用顶层Maven目标**。如果工程为多模块工程，想在Jenkins中部署子模块的话，那么需要在父模块中调用 `mvn clean install` 命令，然后在子模块中调用 `mvn clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml` 命令。以Demo工程为例，工程结构如下。

```
sh-3.2# tree -L 1 carshop
carshop
├── detail
├── itemcenter
├── itemcenter-api
└── pom.xml
```

其中，detail、itemcenter、itemcenter-api为子模块，如果您想部署itemcenter模块的话，需要在父工程中设置一个clean install构建目标，然后在itemcenter模块中设置 `clean package toolkit:deploy -Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dtoolkit_deploy=toolkit_deploy.yaml` 构建目标。


Build
调用顶层 Maven 目标
Maven Version
apache-maven-3.5.4
Goals
clean install
Advanced...
调用顶层 Maven 目标
Maven Version
apache-maven-3.5.4
Goals
clean package toolkit:deploy-Dtoolkit_profile=toolkit_profile.yaml -Dtoolkit_package=toolkit_package.yaml -Dto
POM
itemcenter/pom.xml
Properties

5.高级配置

5.1. 设置PHP应用监控

如果您想要开启PHP应用的监控功能，您需要将SAE内置的应用监控配置文件挂载到PHP的配置加载目录。

在应用创建过程中设置PHP应用监控

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页签，展开**PHP应用监控设置**区域，在**挂载目录**区域设置应用监控配置文件的挂载目录。

> **注意** 请确保挂载的应用监控配置文件能被进程加载。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后设置PHP应用监控

PHP应用监控可以在创建应用过程中设置，也可以在应用部署完成后进行配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**PHP应用监控设置**区域，在**挂载目录**区域设置应用监控配置文件的挂载目录。

> **注意** 请确保挂载的应用监控配置文件能被进程加载。

5. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

结果验证

您可以通过以下方式验证PHP应用监控是否设置成功。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用**基本信息**页面单击**实例部署信息**，在**实例部署信息**页签，单击具体实例**操作**列的**Webshell**，在命令框中执行以下命令。
   - 执行以下命令获取ARMS信息。

     ```
     cat /usr/local/etc/php/conf.d/arms.ini # /usr/local/etc/php/conf.d/arms.ini为配置文件挂载目录，请根据实际配置填写。
     ```

     如果返回结果如下，说明PHP应用监控设置成功。

     ```
     extension=/usr/local/arms/arms-php-agent/arms-x.x.so
     [ARMS]
     arms.enable=1
     arms.app_id=<yourAppName>
     arms.license_key=<yourLicenseKey>
     arms.network_type=tcp
     arms.tcp_host=127.0.0.0
     arms.tcp_port=11234
     ```

   - 执行以下命令查看应用是否加载监控配置文件。

     ```
     php -i | grep ini
     ```

     如果返回结果如下，说明PHP应用监控设置成功。

     ```
     Scan this dir for additional .ini files => /usr/local/etc/php/conf.d
     Additional .ini files parsed => /usr/local/etc/php/conf.d/arms.ini
     ```

问题反馈

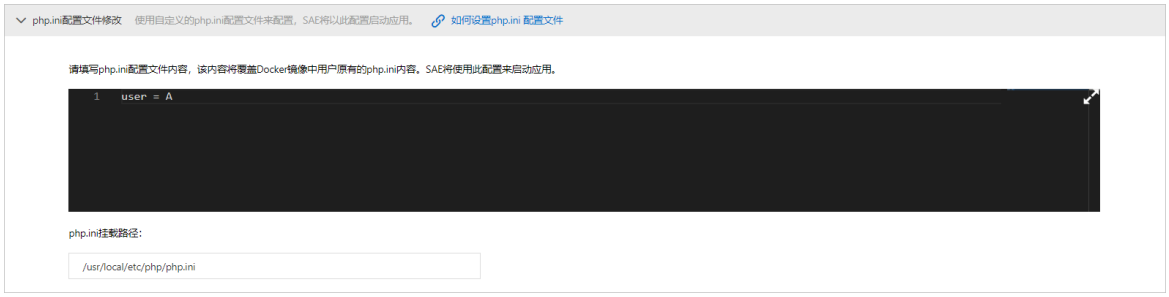
如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.2. 设置PHP应用配置文件

SAE支持自定义设置PHP应用的配置文件，设置完成后，SAE将使用您自定义的配置文件启动应用。本文介绍设置PHP应用配置文件及验证配置文件是否加载成功的操作步骤。

在应用创建过程中设置PHP应用配置文件

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页签，展开**php.ini配置文件修改**区域，填写配置文件内容，并在**php.ini挂载路径**区域设置配置文件挂载的路径。

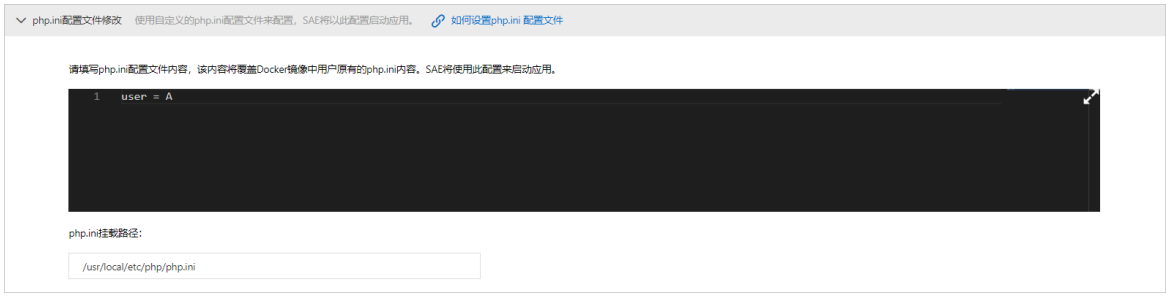


6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后设置PHP应用配置文件

PHP应用配置文件可以在创建应用过程中设置，也可以在应用部署完成后设置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**php.ini配置文件修改**区域，填写配置文件内容，并在**php.ini挂载路径**区域设置配置文件挂载的路径。



5. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

结果验证

您可以通过以下步骤验证应用配置文件是否加载成功。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用**基本信息**页面单击**实例部署信息**，在**实例部署信息**页签，单击具体实例**操作**列的**Webshell**，

在命令框中执行以下命令查看应用配置文件是否加载成功。

```
php -i | grep ini
```

如果返回结果如下，说明应用配置文件已加载成功。

```
Loaded Configuration File => /usr/local/etc/php/php.ini
```

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.3. 设置启动命令

通过SAE控制台使用镜像、JAR包和WAR包方式部署应用时，SAE通过容器镜像、JAR包和WAR包中预设的启动参数启动容器。如果在启动前需要进行特殊配置，例如Nginx，或者不采用预设的启动参数，您可以在SAE设置容器启动命令，进行特殊配置或者覆盖镜像的启动默认值。本配置适用于精通Docker和JVM的工程师。

背景信息

在制作镜像时容器的启动配置已经在Dockerfile文件中的ENTRYPOINT或CMD进行了配置，容器启动时Dockerfile文件中的内容会被优先执行。

例如Dockerfile中设置的以下命令，在容器启动时将被第一个执行。

```
FROM ubuntu
ENTRYPOINT [nginx, '-g', 'daemon off;']
```

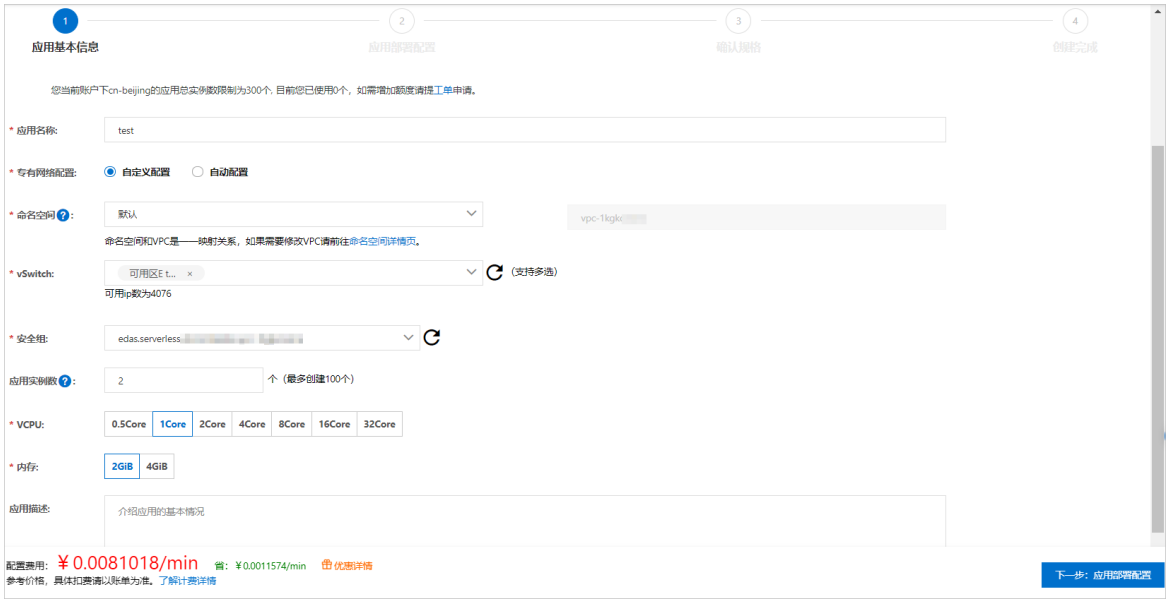
使用说明

启动命令可以在部署应用过程中设置，也可以在应用部署完成后设置。不同部署方式及应用环境可配置的启动命令不同，请您参考以下文档，根据实际界面进行配置：

- 设置镜像部署应用的启动命令
- 设置JAR包部署应用的启动命令
- 设置WAR包部署应用的启动命令
- 应用部署完成后配置启动命令

设置镜像部署应用的启动命令

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签，设置应用相关信息，配置完成后单击**下一步：应用部署配置**。



参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线（_）以及短划线（-）组合，仅允许字母开头，不超过36个字符。
专有网络配置	选择专有网络的配置方式。 ○ **自定义配置**：选中后，您可以为创建的应用选择所需的命名空间、VPC、vSwitch及安全组。 ○ **自动配置**：选中后，SAE将自动帮您配置命名空间、VPC、vSwitch及安全组，无需手动选择。
命名空间	在下拉菜单中选择创建好的命名空间。仅**自定义配置**专有网络时需要选择。 **说明** 命名空间和VPC是一一映射关系，关于修改VPC的更多信息，请参见使用专有网络。
vSwitch	在下拉菜单中选择vSwitch。仅**自定义配置**专有网络时需要选择。 vSwitch数量至少需要配置1个，建议不超过3个，且每个vSwitch至少匹配1个应用实例。
安全组	在下拉菜单中选择安全组。仅**自定义配置**专有网络时需要选择。 **注意** 如果您的VPC内没有创建安全组，请单击**创建安全组**，根据提示创建安全组。更多信息，请参见创建安全组。

参数名	说明
应用实例数	选择需要创建的实例个数。
VCPU	选择需要创建的实例CPU规格。
内存	选择需要创建的实例内存规格。
应用描述	填写应用的基本情况，输入的描述信息不超过100个字符。

4. 在**应用部署配置**页签，选择**技术栈语言**为**Java**，**应用部署方式**为**镜像**，展开**启动命令设置**区域并输入相关配置项。

> **注意**
> - 如果对原有Dockerfile镜像的ENTRYPOINT和CMD配置内容不熟悉，请勿自定义或者修改启动命令和启动参数，错误的启动命令将导致应用创建失败。
> - 由于Docker运行时仅支持一条ENTRYPOINT命令，所以SAE控制台中设置的启动命令，将会覆盖制作应用容器Docker镜像时Dockerfile中所设置的ENTRYPOINT和CMD命令。

以 `ENTRYPOINT: [nginx, '-g', 'daemon off;']` 为例，参数说明如下。

参数	说明
启动命令	输入启动命令，例如 `nginx` 。
启动参数	输入启动参数，例如 `-g` 。 单击**+添加**，可以在新的参数行中增加输入参数 `daemon off` 。

5. 单击**下一步：确认规格**。
6. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
7. 您可以通过以下方式验证配置是否生效。
   - 方法一：
     在应用的**变更记录**页面中查看应用变更详情，如果显示**执行成功**，则表示部署成功，即配置已生效。

- 方法二：
  在应用**基本信息**页面的**实例部署信息**页签查看实例的**运行状态**。如果**运行状态**显示为绿色的**Running**，表示应用部署成功，即配置已生效。

设置JAR包部署应用的启动命令

Java应用启动和运行时需要定义启动命令和参数，如Java的JVM参数、GC策略等。

SAE将上传的JAR包自动编译为镜像，并上传至镜像仓库，且以容器方式运行应用。在编译过程中，内置了启动命令和参数，指定了JAR包的存放路径，支持在SAE控制台上修改JAR包默认的启动命令及命令参数。

> **注意** 如果您对原有Dockerfile镜像的ENTRYPOINT和CMD配置内容不熟悉，请勿自定义或者修改启动命令和启动参数，错误的启动命令将导致应用创建失败。

设置JAR包部署应用的启动命令操作步骤与镜像部署的启动命令配置操作相同。具体操作，请参见设置镜像部署应用的启动命令，在步骤4中选择**应用部署方式**为**JAR包部署**，启动命令示例如下。

参数说明如下。

参数	说明
系统默认启动命令	SAE默认的启动命令。

参数	说明
options设置	配置JVM参数。关于参数详情，请参见JVM参数配置说明、Tuning Java Virtual Machines和JVM Tuning: How to Prepare Your Environment for Performance Tuning。 如果您需要使用应用的远程调试功能，请配置以下命令： `-agentlib:jdwp=transport=dt_socket,address=9000,server=y,suspend=n` • transport：远程调试间的数据传输方式。 • address：远程调试的地址。与开启远程调试时设置的调试端口保持一致，远程调试的详细说明，请参见Java远程调试。
args设置	配置标准输出和错误输出的重定向命令，例如 `1>>/tmp/std.log>&1` 。
options快捷设置：只有**Java环境**为**Dragonwell**时可以设置。	
开启微服务性能提升（Wisp 2协程）	默认开启，开启后可以提升运行时多线程性能。
开启应用内存优化（G1）	默认开启，开启后可以针对多CPU与大容量内存场景，降低GC时间，适用于GC需要优化、大数据等场景。
开启应用启动加速（Quickstart）	只有**Java环境**为**Dragonwell 11**时可以设置。选中**开启应用启动加速（Quickstart）**并设置**持久化目录**后，可以提升应用启动效率。 **注意** 开启应用启动加速前，需要先设置NAS存储。具体操作，请参见设置NAS存储。
持久化目录	开启应用加速后需要设置，设置的**持久化目录**推荐为NAS存储的目录或者子目录，达到跨实例间的加速效果。关于如何设置NAS存储，请参见设置NAS存储。

设置WAR包部署应用的启动命令

设置WAR包部署应用的启动命令操作步骤与镜像部署的启动命令配置操作相同。具体操作，请参见设置镜像部署应用的启动命令，在步骤4中选择**应用部署方式**为**WAR包部署**，启动命令示例如下。

参数说明如下。

参数	说明
系统默认启动命令	SAE默认的启动命令。
options设置	配置JVM参数。关于参数详情，请参见JVM参数配置说明、Tuning Java Virtual Machines和JVM Tuning: How to Prepare Your Environment for Performance Tuning。 如果您需要使用应用的远程调试功能，请配置以下命令： `-agentlib:jdwp=transport=dt_socket,address=9000,server=y,suspend=n` • transport：远程调试间的数据传输方式。 • address：远程调试的地址。与开启远程调试时设置的调试端口保持一致，远程调试的详细说明，请参见Java远程调试。 **注意** 以WAR包方式部署应用的启动命令与JAR包方式部署应用的启动命令不同，请您不要共用启动命令。
options快捷设置：只有**Java环境**为**Dragonwell**时可以设置。	
开启微服务性能提升（Wisp 2协程）	默认开启，开启后可以提升运行时多线程性能。
开启应用内存优化（G1）	默认开启，开启后可以针对多CPU与大容量内存场景，降低GC时间，适用于GC需要优化、大数据等场景。

参数	说明
开启应用启动加速（Quickstart）	只有**Java环境**为**Dragonwell 11**时可以设置。选中**开启应用启动加速（Quickstart）**并设置**持久化目录**后，可以提升应用启动效率。 **注意** 开启应用启动加速前，需要先设置NAS存储。具体操作，请参见设置NAS存储。
持久化目录	开启应用加速后需要设置，设置的**持久化目录**推荐为NAS存储的目录或者子目录，达到跨实例间的加速效果。关于如何设置NAS存储，请参见设置NAS存储。

应用部署完成后配置启动命令

如果您在创建应用时没有配置启动命令，可以在应用部署完成后配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面下方，展开**启动命令设置**区域，根据您的需求配置参数，然后单击**确认**。

   本章节以JAR包部署的应用为例说明启动命令参数，镜像及WAR包部署时的启动命令的参数，请参见设置镜像部署应用的启动命令和设置WAR包部署应用的启动命令。

参数说明如下。

参数	说明
系统默认启动命令	SAE默认的启动命令。

参数	说明
options设置	配置JVM参数。关于参数详情，请参见JVM参数配置说明、Tuning Java Virtual Machines和JVM Tuning: How to Prepare Your Environment for Performance Tuning。 如果您需要使用应用的远程调试功能，请配置以下命令： `-agentlib:jdwp=transport=dt_socket,address=9000,server=y,suspend=n` ○ transport：远程调试间的数据传输方式。 ○ address：远程调试的地址。与开启远程调试时设置的调试端口保持一致，远程调试的详细说明，请参见Java远程调试。
args设置	配置标准输出和错误输出的重定向命令，例如 `1>>/tmp/std.log>&1`。
options快捷设置：只有**Java环境**为**Dragonwell**时可以设置。	
开启微服务性能提升（Wisp 2协程）	默认开启，开启后可以提升运行时多线程性能。
开启应用内存优化（G1）	默认开启，开启后可以针对多CPU与大容量内存场景，降低GC时间，适用于GC需要优化、大数据等场景。
开启应用启动加速（Quickstart）	只有**Java环境**为**Dragonwell 11**时可以设置。选中**开启应用启动加速（Quickstart）**并设置**持久化目录**后，可以提升应用启动效率。 **注意** 开启应用启动加速前，需要先设置NAS存储。具体操作，请参见设置NAS存储。

参数	说明	
持久化目录	开启应用加速后需要设置，设置的**持久化目录**推荐为NAS存储的目录或者子目录，达到跨实例间的加速效果。关于如何设置NAS存储，请参见设置NAS存储。	

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.4. 设置Java Tomcat参数

当您使用WAR包部署应用时，您可以根据需要自定义Tomcat参数，包括应用端口、应用访问路径以及最大线程数等信息。

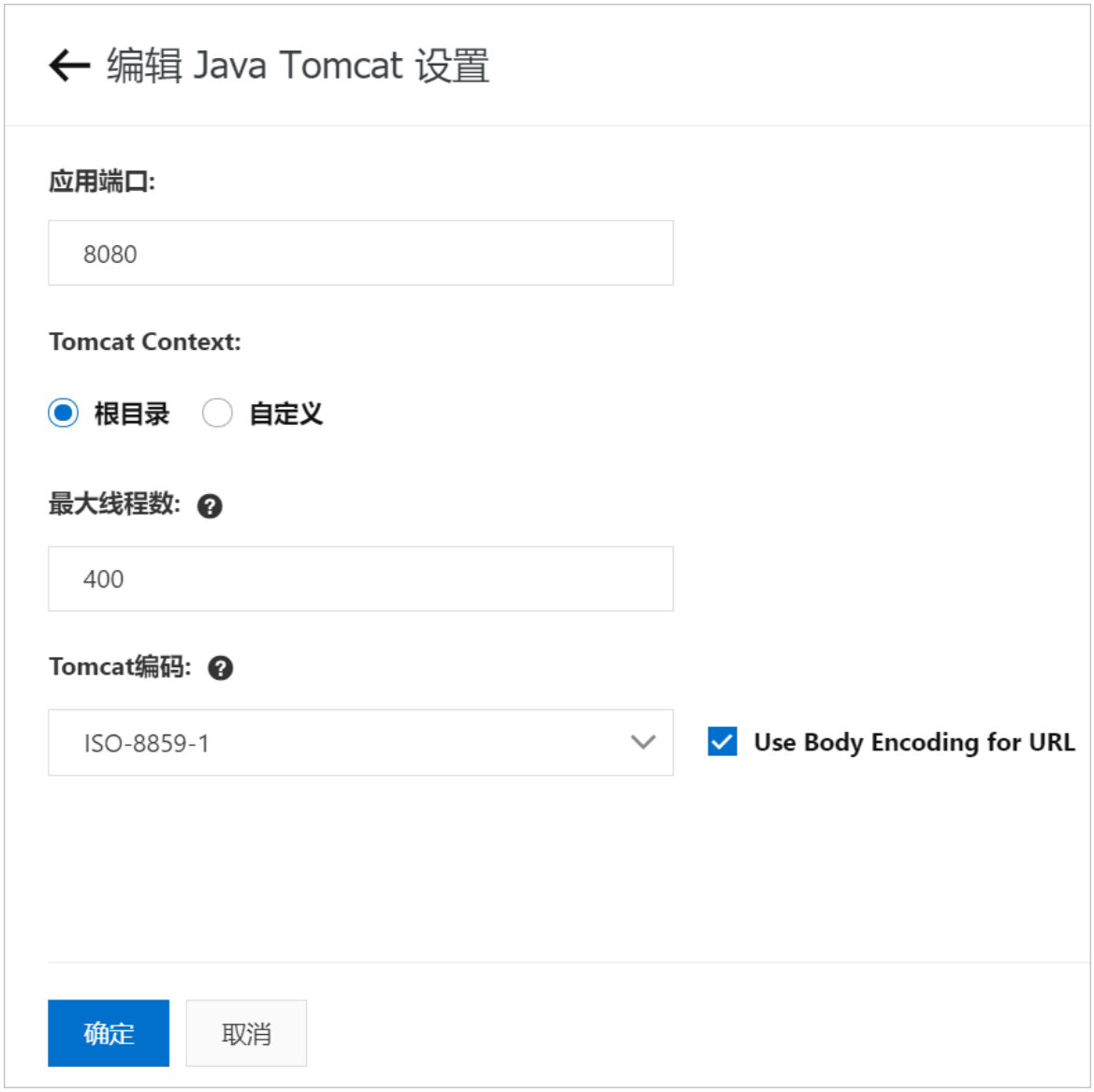
在创建应用过程中配置Tomcat

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，设置相关信息，展开**Java Tomcat设置**区域。

   > **说明** 只有**应用部署方式**为**WAR包部署**时才能设置Tomcat。

5. 在**Java Tomcat设置**区域，选中**自定义配置**，单击**编辑**。

6. 在**编辑Java Tomat设置**面板中设置参数，单击**确定**。



参数说明如下。

参数	说明
应用端口	默认为8080，取值范围[1024,49151]。

参数	说明
Tomcat Context	选择应用访问的路径。 ○ **程序包名字**：无需填写自定义路径，应用的访问路径是WAR包名称。**文件上传方式**为**上传WAR包**时可以选择。 ○ **根目录**：应用的访问路径是/。 ○ **自定义**：在**自定义路径**区域设置应用需要访问的路径。
最大线程数	设置并发处理请求的最大线程数。默认为400，取值范围[10, 1024]。 **注意** **最大线程数**会直接影响应用性能，请谨慎设置。
Tomcat编码	Tomcat的编码格式，包括**UTF-8**、**ISO-8859-1**、**GBK**和**GB2312**，默认为**ISO-8859-1**。
Use Body Encoding for URL	默认选中，选中后可以根据HTTP Body设置的字符集解码查询参数（QueryString）。

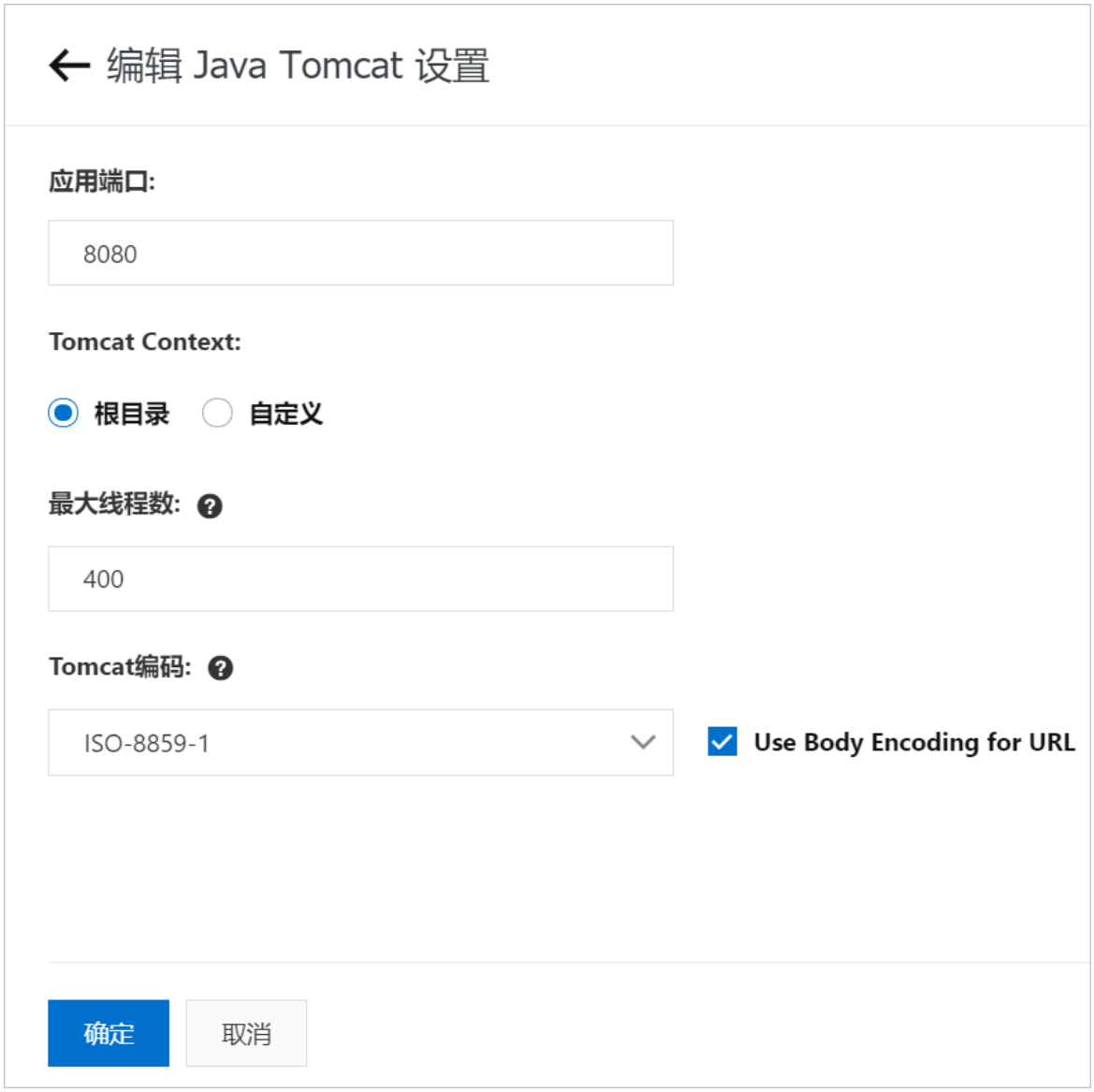
7. 单击**下一步：确认规格**。
8. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
9. 您可以通过以下方式验证配置是否生效。
   - 方法一：
     在应用的**变更记录**页面中查看应用变更详情，如果显示**执行成功**，则表示部署成功，即配置已生效。
   - 方法二：
     在应用**基本信息**页面的**实例部署信息**页签查看实例的**运行状态**。如果**运行状态**显示为绿色的**Running**，表示应用部署成功，即配置已生效。

应用部署完成后配置Tomcat

如果您在创建应用时没有配置Tomcat，可以在应用部署成功后配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面下方，展开**Java Tomcat设置**区域。
5. 在**Java Tomcat设置**区域，选中**自定义配置**，单击**编辑**。

6. 在**编辑Java Tomat设置**面板中设置参数，单击**确定**。



参数说明如下。

参数	说明
应用端口	默认为8080，取值范围[1024,49151]。

参数	说明
Tomcat Context	选择应用访问的路径。 ○ **程序包名字**：无需填写自定义路径，应用的访问路径是WAR包名称。**文件上传方式**为**上传WAR包**时可以选择。 ○ **根目录**：应用的访问路径是/。 ○ **自定义**：在**自定义路径**区域设置应用需要访问的路径。
最大线程数	设置并发处理请求的最大线程数。默认为400，取值范围[10, 1024]。 **注意** **最大线程数**会直接影响应用性能，请谨慎设置。
Tomcat编码	Tomcat的编码格式，包括**UTF-8**、**ISO-8859-1**、**GBK**和**GB2312**，默认为**ISO-8859-1**。
Use Body Encoding for URL	默认选中，选中后可以根据HTTP Body设置的字符集解码查询参数（QueryString）。

7. 单击**确认**，完成Tomcat配置。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.5. 设置环境变量

应用在系统中运行需要配置特定的环境变量，如Java应用程序在配置Java_home及其Path后，其相关命令才可以被执行。本文介绍如何在SAE控制台上为应用设置环境变量，包括自定义环境变量和从命名空间配置项中引入的环境变量。

在应用创建过程中设置环境变量

1. 登录SAE控制台。

2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页签，展开**环境变量设置**区域，并执行以下操作之一来添加环境变量：
   - 添加自定义环境变量：在**类型**下拉框中选择**自定义**，并输入**变量名称**和**变量值**。

   - 引用配置项作为环境变量：在**类型**下拉框中选择**引用配置项**，输入**变量名称**，并在最右侧的下拉框中选择已创建的配置项名称和该配置项中的键。

> **说明** 环境变量名称的长度建议不超过256个字符。SAE将环境变量作为属性存储在应用部署属性中，允许配置多个环境变量。

6. （可选）如需添加其他环境变量，单击**添加**，并重复上一步的操作。
7. 单击**下一步：确认规格**。
8. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

示例

某应用集成了MySQL，在应用使用时需为其配置MySQL运行所需的环境变量。在SAE控制台进行部署时为其配置如下所示环境信息：

- **MYSQL_ROOT_PASSWORD**：用于设置MySQL的Root密码。如果不设置该项，则MySQL容器无法正常启动。
- （可选）**MYSQL_USER**和**MYSQL_PASSWORD**：用于添加除Root之外的账号及其密码。
- （可选）**MYSQL_DATABASE**：用于设置生成容器时需要新建的数据库。

应用部署完成后配置环境变量

环境变量配置可以在创建应用过程中设置，也可以在应用部署完成后进行配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**环境变量设置**区域，并执行以下操作之一来添加环境变量。
   - 添加自定义环境变量：在**类型**下拉框中选择**自定义**，并输入**变量名称**和**变量值**。

- 引用配置项作为环境变量：在**类型**下拉框中选择**引用配置项**，输入**变量名称**，并在最右侧的下拉框中选择已创建的配置项名称和该配置项中的键。

> **说明** 环境变量名称的长度建议不超过256个字符。SAE将环境变量作为属性存储在应用部署属性中，允许配置多个环境变量。

-

> **说明** 环境变量名称的长度建议不超过256个字符。SAE将环境变量作为属性存储在应用部署属性中，允许配置多个环境变量。

5. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

相关文档

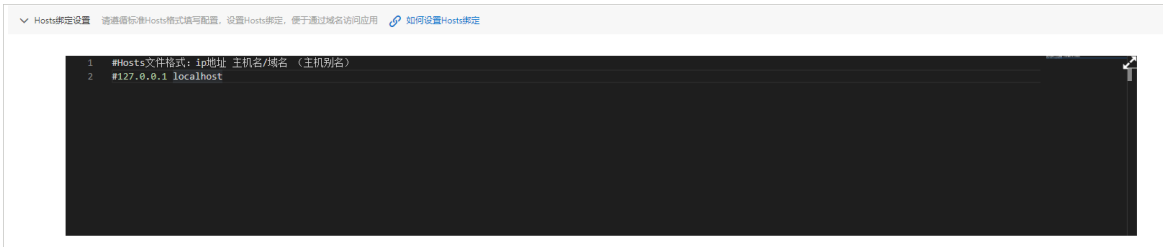
- 管理和使用配置项（K8s）

5.6. 设置Hosts绑定

SAE支持应用级别的实例，通过Hosts绑定对主机名进行解析，方便应用实例通过主机名进行访问。

在应用创建过程中设置Hosts绑定

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页面，展开**Hosts绑定设置**区域，输入Hosts配置项。



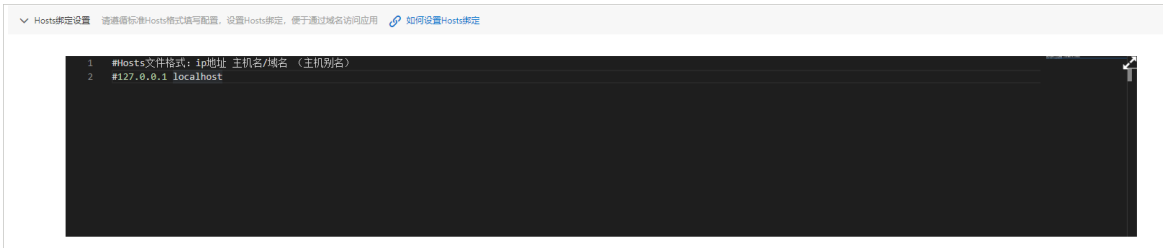
6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
8. 结果验证。

   应用部署完成后，所设置的Hosts配置生效，您可以通过域名访问应用。

应用部署完成后配置Hosts绑定

Hosts配置可以在创建应用过程中设置，也可以在应用部署完成后进行配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**Hosts绑定设置**区域，输入Hosts配置项。



5. 配置完成后单击**确认**。

   > **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

6. 结果验证。

   应用部署完成后，所设置的Hosts配置生效，您可以通过域名访问应用。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.7. 设置健康检查

将应用部署在SAE后，您可以使用健康检查功能查看应用实例与业务运行是否正常，以便运行异常时定位问题。本文介绍如何在创建应用过程中和应用部署完成后配置健康检查。

背景信息

健康检查是指由Liveness探针或者Readiness探针对应用实例与应用进行定时检查，并将结果反馈给SAE控制台的过程，帮助您了解集群环境下整个服务的运行状态，方便您定位问题。

> **说明** 此方式适用于以下业务：
> - 应用启动时需要加载磁盘数据。
> - 依赖外部模块而导致应用启动时间长。

SAE基于Kubernetes，提供了以下两种健康检查方式：

- **应用实例存活检查（Liveness配置）**：针对单个应用实例进行健康检查，检测应用实例是否已经启动。
  - 检查成功：表示应用实例健康，SAE不执行任何操作。
  - 检查失败：表示应用实例不健康，SAE会重启应用实例。
- **应用业务就绪检查（Readiness配置）**：针对应用业务进行健康检查，检测处理客户请求的应用是否已经就绪。
  - 检查成功：表示应用准备就绪，SAE会为应用分配业务流量进行处理。
  - 检查失败：表示应用未准备就绪，SAE会上报应用异常，不为该应用分配业务流量。

> **说明**
> - 健康检查成功一次即认为检查通过。
> - 如果健康检查失败，SAE会按照您设置的检查周期时间重新进行健康检查。如果检查连续3次失败，则认为健康检查失败，SAE会开始重启应用实例或不为该应用分配业务流量。

在创建应用过程中配置健康检查

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。

5. 在**应用部署配置**页签，展开**应用健康检查设置**区域，选择**检查方式**，并设置相关参数。

> **说明**
>
> - 应用实例存活检查和应用业务就绪检查的参数相同。
> - 在健康检查设置中**应用实例存活检查（Liveness配置）**与**应用业务就绪检查（Readiness配置）**二者可选配其一，也可二者都配（推荐）。如果二者都配，SAE会根据配置的检查延迟时间依次执行健康检查。

- **HTTP请求检查**：通过向实例发送HTTP请求检查健康情况。通过HTTP访问获得的HTTP状态码位于[200,399]之间，则认为访问成功，实例健康；否则认为失败，实例不健康。

参数说明如下所示。

参数名	说明
路径	设置访问HTTP Server的路径。
端口	设置访问HTTP Server的端口。
高级设置	展开**高级设置**后，您可以选择判断返回的字符串中是否包含或不包含设置的关键字。
协议	选择**HTTP**或**HTTPS**。
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动后5秒开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。

- **TCP端口检查**：通过向实例发送一个TCP Socket来检查实例的健康情况。如果可以建立连接，则认为实例健康；如果连接失败，则认为不健康。

参数说明如下所示。

参数名	说明
TCP端口	设置TCP检查访问的端口。
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动后5秒开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。

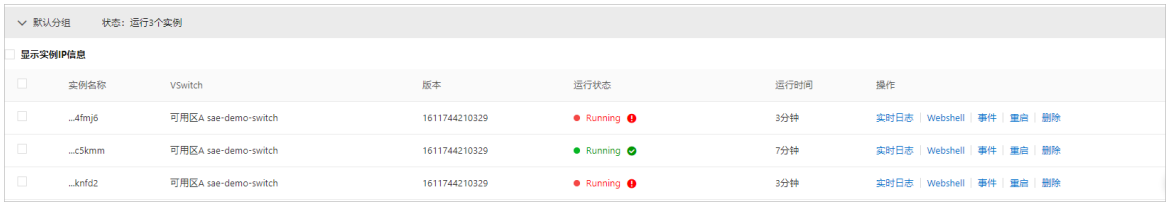
- **执行命令检查**：通过在实例中执行探针检测命令来检查实例的健康情况。如果执行命令后返回的状态码为0，则认为执行成功，实例健康；如果不为0，则认为执行失败，实例不健康。

参数说明如下所示。

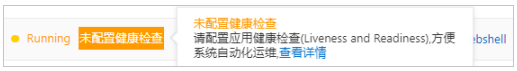
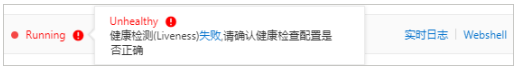
参数名	说明
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动5秒后开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。
执行命令	设置应用实例或者进程内部执行的健康检查命令。如果该命令返回码为0，则表示应用健康。 执行命令的具体格式，请参见右侧的示例区域或下文中的示例。健康检查相关命令，请参见Kubernetes官网Configure Probes。 **注意** 设置执行命令时，首行填写命令，次行之后再填写参数，多条参数时以换行分隔，请避免出现空行情况。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
8. 结果验证。

   在**应用列表**页面，单击**实例部署信息**页签，在**默认分组**区域查看实例的运行状态，将鼠标移至状态后的标志上，可查看详细信息。



   状态说明如下所示。

运行状态	说明
○ Liveness健康检查未配置 ○ Readiness健康检查未配置 ○ Liveness与Readiness健康检查未配置  	表示实例未配置健康检查。 **说明** ○ 推荐既配置Liveness也配置Readiness检查。 ○ 如果您需要了解健康检查的配置说明，您可以单击**查看详情**，查看配置文档。
 	表示实例正在或等待进行健康检查。 **说明** 应用启动后，需要等待健康检查所设置的延迟时间，才会开始健康检查。
○ Liveness健康检查失败  ○ Readiness健康检查失败 ○ Liveness与Readiness健康检查失败 	表示健康检查失败，实例不健康。 **说明** 如果您需要查看检查失败原因，您可以单击**失败**，页面会自动跳转至**应用事件**页面，展示健康检查的详细信息。
 	表示健康检查通过，实例健康。

应用部署完成后配置健康检查

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**应用健康检查设置**区域，配置相关参数。

> **说明**
> ○ 应用实例存活检查和应用业务就绪检查的参数相同。
> ○ 在健康检查设置中**应用实例存活检查（Liveness配置）**与**应用业务就绪检查（Readiness配置）**二者可选配其一，也可二者都配（推荐）。如果二者都配，SAE会根据配置的检查延迟时间依次执行健康检查。

- **HTTP请求检查**：通过向实例发送HTTP请求检查健康情况。通过HTTP访问获得的HTTP状态码位于[200,399]之间，则认为访问成功，实例健康；否则认为失败，实例不健康。

参数说明如下所示。

参数名	说明
路径	设置访问HTTP Server的路径。
端口	设置访问HTTP Server的端口。
高级设置	展开**高级设置**后，您可以选择判断返回的字符串中是否包含或不包含设置的关键字。
协议	选择**HTTP**或**HTTPS**。
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动后5秒开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。

- **TCP端口检查**：通过向实例发送一个TCP Socket来检查实例的健康情况。如果可以建立连接，则认为实例健康；如果连接失败，则认为不健康。

参数说明如下所示。

参数名	说明
TCP端口	设置TCP检查访问的端口。
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动后5秒开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。

- **执行命令检查**：通过在实例中执行探针检测命令来检查实例的健康情况。如果执行命令后返回的状态码为0，则认为执行成功，实例健康；如果不为0，则认为执行失败，实例不健康。

参数说明如下所示。

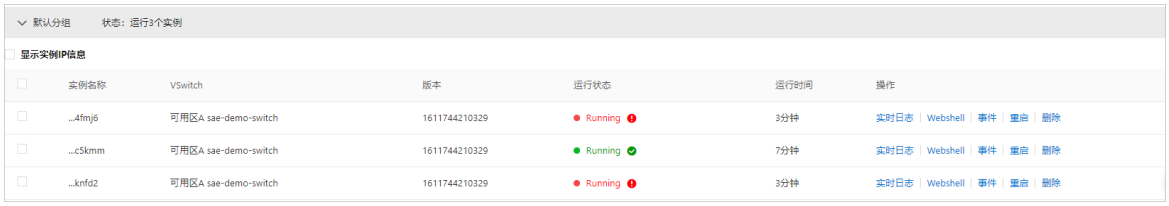
参数名	说明
延迟时间（秒）	设置健康检查延迟检测时间，单位为秒。默认为10。例如设置为5，表示从应用启动5秒后开始检测。
超时时间（秒）	设置健康检查超时等待时间，单位为秒。默认为1。例如设置为10，如果超时等待时间超过10秒，则本次健康检查失败，上报超时异常。若设置为0或不设置，默认超时等待时间为1秒。
检查周期（秒）	设置健康检查周期，单位为秒。默认为30。例如设置为5，表示每隔5秒检查一次。
执行命令	设置应用实例或者进程内部执行的健康检查命令。如果该命令返回码为0，则表示应用健康。 执行命令的具体格式，请参见右侧的示例区域或下文中的示例。健康检查相关命令，请参见Kubernetes官网Configure Probes。 **注意** 设置执行命令时，首行填写命令，次行之后再填写参数，多条参数时以换行分隔，请避免出现空行情况。

5. 配置完成后单击**确认**。

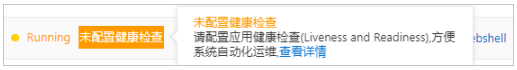
> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

6. 结果验证。

   在**应用列表**页面，单击**实例部署信息**页签，在**默认分组**区域查看实例的运行状态，将鼠标移至状态后的标志上，可查看详细信息。



   状态说明如下所示。

运行状态	说明
○ Liveness健康检查未配置 ○ Readiness健康检查未配置 ○ Liveness与Readiness健康检查未配置  	表示实例未配置健康检查。 **说明** ○ 推荐既配置Liveness也配置Readiness检查。 ○ 如果您需要了解健康检查的配置说明，您可以单击**查看详情**，查看配置文档。
 	表示实例正在或等待进行健康检查。 **说明** 应用启动后，需要等待健康检查所设置的延迟时间，才会开始健康检查。
○ Liveness健康检查失败 ○ Readiness健康检查失败 ○ Liveness与Readiness健康检查失败 	表示健康检查失败，实例不健康。 **说明** 如果您需要查看检查失败原因，您可以单击**失败**，页面会自动跳转至**应用事件**页面，展示健康检查的详细信息。
 	表示健康检查通过，实例健康。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.8. 设置日志收集

SAE实时日志功能支持查看500行日志信息，如果您有更高的查阅需求可以使用文件日志收集功能，SAE会将业务文件日志（不包含stdout和stderr日志）收集并输入SLS中，实现无限制行数查看日志、自行聚合分析日志，方便业务日志对接，并按日志使用量计费。

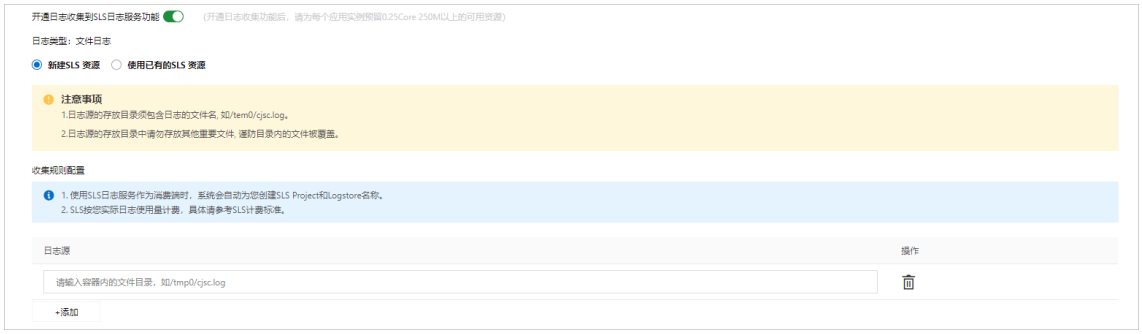
前提条件

- 开通SLS。
- 确保应用中每个实例至少预留0.25 Core CPU和250 MB内存的可用资源。

在创建应用过程中配置文件日志收集

> **注意**
>
> - SLS账号内提供200个Logstore资源、50个Project资源。
> - 应用开始创建后，系统自动检查SLS服务是否开启、SLS账号的内置资源是否足够。
> - 如果SLS服务未开启，请按提示开通SLS。
> - 如果内置资源不足，请创建工单以申请增额。
> - SLS的计费详情，请参见SLS计费说明。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页签，展开**日志收集服务**区域，打开**开通日志收集到SLS日志服务功能**开关，设置SLS资源。
   - **新建SLS资源**：选中**新建SLS资源**，并配置**日志源**，指定日志源存放的文件目录。



> **注意**
>
> - 日志源的存放目录须包含日志的文件名，如*/tmp0/cjsc.log*。
> - 请勿在日志源的存放目录中存放其他重要文件，谨防目录内的文件被覆盖。

- **使用已有的SLS资源**：选中**使用已有的SLS资源**，从**使用已有的SLS Project**下拉框，选择需要存放日志的Project；配置**日志源**，指定文件目录；从**logstore**下拉框，选择Logstore；在**Logtail**下拉框中选择已有的Logtail或新建Logtail。Logtail详细信息，请参见Logtail采集概述。

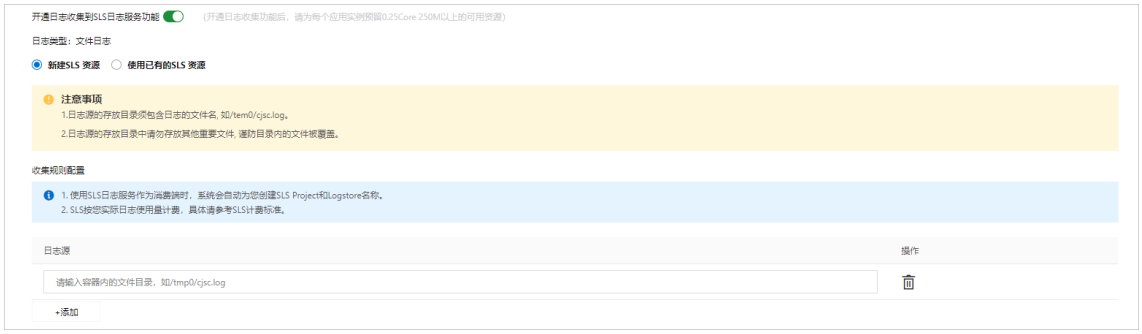
如果需要配置多条收集规则，请单击**添加**。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
8. 验证结果。

   应用部署完成后，SAE依据所配的日志收集规则收集日志并存放到指定的文件内。
   您可以在应用详情页面左侧导航栏中，选择**日志管理 > 文件日志**，在**文件日志**页面查看所收集的日志信息。
   如果存在日志数据，则表示日志收集配置成功，您可以依据日志信息进行相关业务分析。

应用部署完成后配置文件日志收集

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面下方，展开**日志收集服务**区域，打开**开通日志收集到SLS日志服务功能**开关，设置SLS资源。
   - **新建SLS资源**：选中**新建SLS资源**，并配置**日志源**，指定日志源存放的文件目录。



> **注意**
>
> - 日志源的存放目录须包含日志的文件名，如*/tmp0/cjsc.log*。
> - 请勿在日志源的存放目录中存放其他重要文件，谨防目录内的文件被覆盖。

- **使用已有的SLS资源**：选中**使用已有的SLS资源**，从**使用已有的SLS Project**下拉框，选择需要存放日志的Project；配置**日志源**，指定文件目录；从**logstore**下拉框，选择Logstore；在**Logtail**下拉框中选择已有的Logtail或新建Logtail。Logtail详细信息，请参见Logtail采集概述。

如果需要配置多条收集规则，请单击**添加**。

如果需要配置多条收集规则，请单击**添加**。

5. 配置完成后，单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。
> 如果您在应用部署时采用分批发布或者灰度发布策略，同时您配置的日志源不变，只改变日志收集的Project或者Logstore，那么在所有批次的部署流程成功完成之前，您的日志仍被收集到老的Project与Logstore中。因为SLS不支持同一个文件被同时采集到不同的Project或Logstore中。

6. 验证结果。

   应用部署完成后，SAE依据所配的日志收集规则收集日志并存放到指定的文件内。
   您可以在应用详情页面左侧导航栏中，选择**日志管理 > 文件日志**，在**文件日志**页面查看所收集的日志信息。
   如果存在日志数据，则表示日志收集配置成功，您可以依据日志信息进行相关业务分析。

常见问题

- SAE日志功能是否支持滚动日志？
  支持。您可以通过给日志文件名称添加通配符的方式滚动日志。
  文件名称通配符支持星号（*）和问号（?）。
- 查看文件日志时，在SLS控制台无数据展示，如何处理？

通常SLS默认查询最近15分钟内的查询结果。如果SLS无数据显示，建议使用Webshell查看您的应用是否有文件日志信息。详细操作，请参见使用Webshell诊断应用：

- 如果有日志信息，请提交工单。
- 如果没有日志信息，请检查您的应用信息。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.9. 设置NAS存储

在通常情况下，存储在容器中数据是非持久化的，在容器销毁以后数据也将丢失。在当代高速发展的互联网环境下，数据丢失意味着线上生产环境的灾难性事件。SAE支持了NAS存储功能，解决了应用实例数据持久化和实例间数据分发的问题。

前提条件

- 开通NAS服务
- 通过控制台创建通用型NAS文件系统
- 添加挂载点

应用部署时配置NAS存储

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 展开**持久化存储**区域，单击**NAS文件系统**页签，打开**应用NAS存储**开关。
6. 在**使用已有的NAS文件系统**所在行的下拉列表中选择待挂载的NAS，并设置**挂载源**和**容器路径**。

7. 单击**下一步：确认规格**。
8. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后挂载NAS

您可以在创建应用过程中设置NAS挂载，也可以在应用部署完成后配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**持久化存储**区域，单击**NAS文件系统**页签，打开**应用NAS存储**开关，配置相关参数。

5. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

结果验证

本文介绍不同系统下验证NAS挂载是否成功的方式，您可以根据实际需要选择验证方式：

- 从变更详情判断。
  如果单次创建或部署的变更已经成功，变更生成的新的实例没有出现异常事件，则说明NAS挂载成功。

- 从容器角度判断。
  在Webshell执行以下命令查询应用中是否存在NAS挂载信息。

```
cat /proc/mounts | grep nfs
```

  当显示如下信息时，表示NAS挂载成功。

```
sh-4.2# cat /proc/mounts | grep nfs
[redacted] rw,relatime,vers=3,rsize
=1048576,wsize=1048576,namlen=255,hard,nolock,noresvport,proto=tcp,timeo=600,ret
rans=2,sec=sys,mountaddr=10.0.0.80,mountvers=3,mountport=4002,mountproto=tcp,loc
al_lock=all,addr=10.0.0.80 0 0
sh-4.2#
```

- 从业务角度判断。
  在Webshell中，对挂载NAS路径进行操作，如果可以从NAS文件系统中同步看到，则说明NAS挂载成功。

取消挂载NAS

挂载NAS后，如果您不再使用NAS存储，可以取消挂载NAS。具体操作，请参见应用部署时配置NAS存储，在步骤4中关闭**应用NAS存储**开关，并单击**确认**。

> **注意**
> - 取消挂载NAS后，SAE会重新部署应用，请在业务较少的时间段配置。
> - 在SAE控制台取消挂载NAS后，您在NAS中所存储的数据仍然存在，不会被删除。

常见问题

如何查看NAS中的内容？

如果您需要查看NAS中的内容，需要将NAS挂载到具体的ECS或容器上查看。具体操作，请参见以下文档：

- Windows系统挂载NFS文件系统
- Windows系统挂载SMB文件系统
- Linux系统挂载SMB文件系统
- Linux系统挂载NFS文件系统

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.10. 设置OSS存储

相比于NAS，OSS提供了便捷的工具以及控制台，支持可视化地管理Bucket，并在解决应用实例数据持久化和实例间数据分发问题的基础上，进一步降低成本。本文提供设置OSS存储的操作步骤。

前提条件

- 开通OSS服务
- 创建存储空间
- 获取AccessKey

应用部署时配置OSS存储

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 展开**持久化存储**区域，单击**OSS对象存储**页签。
6. 填写**AccessKey ID**和**AccessKey Secret**。

注意 强烈建议您遵循阿里云安全最佳实践，使用RAM用户的AccessKey调用OSS接口，并确保授予该RAM用户访问OSS资源的最小权限。例如，您需要该RAM用户只读test-sae Bucket的oss-test/目录，那么您可以授予该RAM用户以下最小权限：

```
{
  "Statement": [
    {
      "Action": "oss:GetBucket",
      "Effect": "Allow",
      "Resource": "test-sae"
    },
    {
      "Action": "oss:GetObject",
      "Effect": "Allow",
      "Resource": "oss-test/"
    }
  ],
  "Version": "1"
}
```

7. 单击**添加**，并配置以下参数。

参数	说明	示例值
Bucket	您在OSS创建的Bucket。	bucketname
挂载目录	您在OSS创建的目录或OSS对象，如果OSS挂载目录不存在，会触发异常。	示例如下： ○ / **说明** 代表挂载Bucket的根目录。 ○ tmp/oss-test/ ○ tmp/oss-demo.log
容器路径	您在SAE的容器路径。如果路径已存在，为覆盖关系；如果路径不存在，会新建。	k8s/demo/
权限	容器路径对挂载目录资源的权限，取值如下： ○ 只读 ○ 读写	只读

8. 单击**下一步：确认规格**。
9. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后配置OSS存储

您可以在创建应用过程中配置OSS存储，也可以在应用部署完成后配置。

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 展开**持久化存储**区域，单击**OSS对象存储**页签。
5. 填写**AccessKey ID**和**AccessKey Secret**。

> **注意** 强烈建议您遵循阿里云安全最佳实践，使用RAM用户的AccessKey调用OSS接口，并确保授予该RAM用户访问OSS资源的最小权限。例如，您需要该RAM用户只读test-sae Bucket的oss-test/目录，那么您可以授予该RAM用户以下最小权限：

```
{
  "Statement": [
    {
      "Action": "oss:GetBucket",
      "Effect": "Allow",
      "Resource": "test-sae"
    },
    {
      "Action": "oss:GetObject",
      "Effect": "Allow",
      "Resource": "oss-test/"
    }
  ],
  "Version": "1"
}
```

6. 单击**添加**，并配置以下参数。

参数	说明	示例值
Bucket	您在OSS创建的Bucket。	bucketname
挂载目录	您在OSS创建的目录或OSS对象，如果OSS挂载目录不存在，会触发异常。	示例如下： ○ / **说明** 代表挂载Bucket的根目录。 ○ tmp/oss-test/ ○ tmp/oss-demo.log
容器路径	您在SAE的容器路径。如果路径已存在，为覆盖关系；如果路径不存在，会新建。	k8s/demo/

参数	说明	示例值
权限	容器路径对挂载目录资源的权限，取值如下： ○ 只读 ○ 读写	只读

7. 单击**确认**。

结果验证

本文介绍不同系统下验证OSS挂载是否成功的方式，您可以根据实际需要选择验证方式：

- 从变更详情判断。
  如果单次创建或部署的变更已经成功，变更生成的新的实例没有出现异常事件，则说明OSS挂载成功。

- 从容器角度判断。
  在Webshell执行以下命令查询应用中是否存在OSS挂载信息。

```
cat /proc/mounts | grep ossfs
```

  当显示如下信息时，表示OSS挂载成功。

```
# cat /proc/mounts | grep ossfs
ossfs /[redacted] fuse.ossfs ro,relatime,user_id=0,group_id=0,allow_other 0 0
#
```

- 从业务角度判断。
  在Webshell中，对挂载OSS文件系统路径进行操作，如果可以从OSS控制台同步看到，则说明OSS挂载成功。

取消挂载OSS

挂载OSS后，如果您不再使用OSS存储，可以取消挂载OSS。您可以执行应用部署时配置OSS存储的步骤1~步骤4，找到需要取消挂载的OSS配置，在其**操作**列，单击🗑图标。

> **注意**
> - 取消挂载OSS后，SAE会重新部署应用，请在业务较少的时间段配置。
> - 在SAE控制台取消挂载OSS后，您在OSS中所存储的数据仍然存在，不会被删除。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.11. 设置应用生命周期管理

如果您精通K8s，且需要在应用容器启动前或者关闭前执行一些操作，例如运行前部署资源或者停止前优雅下线应用，您可以设置应用生命周期管理。本文介绍设置应用生命周期管理的操作步骤。

背景信息

您可以在SAE上对应用进行以下配置：

- **启动后处理（PostStart设置）**：应用容器创建成功后，运行前的任务，例如资源部署、环境准备等。
- **停止前处理（PreStop设置）**：应用容器被终止前的任务，例如优雅关闭应用程序、通知其他系统等。
- **优雅下线超时设置（TerminationGracePeriodSeconds）**：应用实例停止前，等待的时间，超时后SAE将强制停止实例。

在创建应用过程中配置应用生命周期管理

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页面，展开**应用生命周期管理设置**面板，根据需要配置信息。

   > **注意** 命令行脚本：首行为命令，次行为参数，多条参数以换行分割，避免出现空行。

   - 配置**启动后处理（PostStart设置）**：
     在**命令行脚本**区域输入自定义的PostStart命令。

- 配置**停止前处理（PreStop设置）**：
  在**命令行脚本**区域输入自定义的PreStop命令。

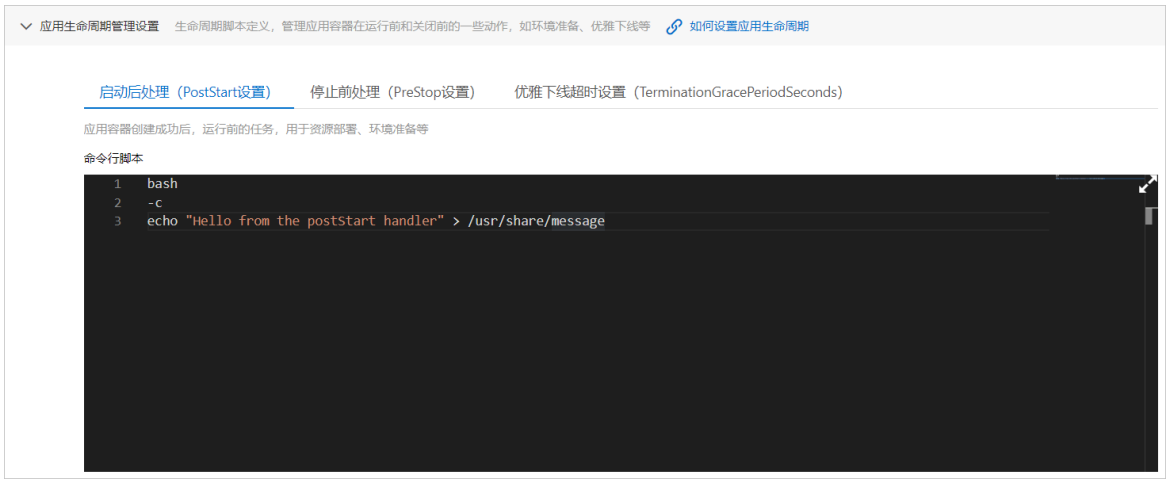
- 配置**优雅下线超时设置（TerminationGracePeriodSeconds）**：
  在**优雅下线超时时间**区域输入超时时间，最大支持60秒。

说明

- Poststart设置：一种容器钩子。该钩子在容器被创建后立刻触发，通知容器它已经被创建。该钩子不需要向其所对应的hook handler传入任何参数。如果该钩子对应的hook handler执行失败，则该容器会被杀死，并根据该容器的重启策略决定是否重启该容器。详情请参见Container Lifecycle Hooks。
- PreStop设置：一种容器钩子。该钩子在容器被删除前触发，其所对应的hook handler必须在删除该容器的请求发送给Docker daemon之前完成。在该钩子对应的hook handler完成后不论执行的结果如何，Docker daemon会发送SGTERN信号量给Docker daemon删除该容器。详情请参见Container Lifecycle Hooks。

6. 单击**下一步：确认规格**。
7. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。
8. 结果验证。

   例如应用启动前，需要将 `Hello from the postStart handler` 内容写入文件。
   PostStart配置如下：



   应用部署后，使用Webshell查看文件中是否存在 `Hello from the postStart handler` ，如果存在，那么表示配置成功。

应用部署后配置应用生命周期管理

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面下方，根据您的需求配置应用信息，并在配置完成后单击**确认**。

   > **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

5. 在**部署应用**页面，展开**应用生命周期管理设置**面板，根据需要配置信息。

   > **注意** 命令行脚本：首行为命令，次行为参数，多条参数以换行分割，避免出现空行。

   - 配置**启动后处理（PostStart设置）**：
     在**命令行脚本**区域输入自定义的PostStart命令。

- 配置**停止前处理（PreStop设置）**：
  在**命令行脚本**区域输入自定义的PreStop命令。

- 配置**优雅下线超时设置（TerminationGracePeriodSeconds）**：
  在**优雅下线超时时间**区域输入超时时间，最大支持60秒。

说明

- Poststart设置：一种容器钩子。该钩子在容器被创建后立刻触发，通知容器它已经被创建。该钩子不需要向其所对应的hook handler传入任何参数。如果该钩子对应的hook handler执行失败，则该容器会被杀死，并根据该容器的重启策略决定是否重启该容器。详情请参见Container Lifecycle Hooks。
- PreStop设置：一种容器钩子。该钩子在容器被删除前触发，其所对应的hook handler必须在删除该容器的请求发送给Docker daemon之前完成。在该钩子对应的hook handler完成后不论执行的结果如何，Docker daemon会发送SGTERN信号量给Docker daemon删除该容器。详情请参见Container Lifecycle Hooks。

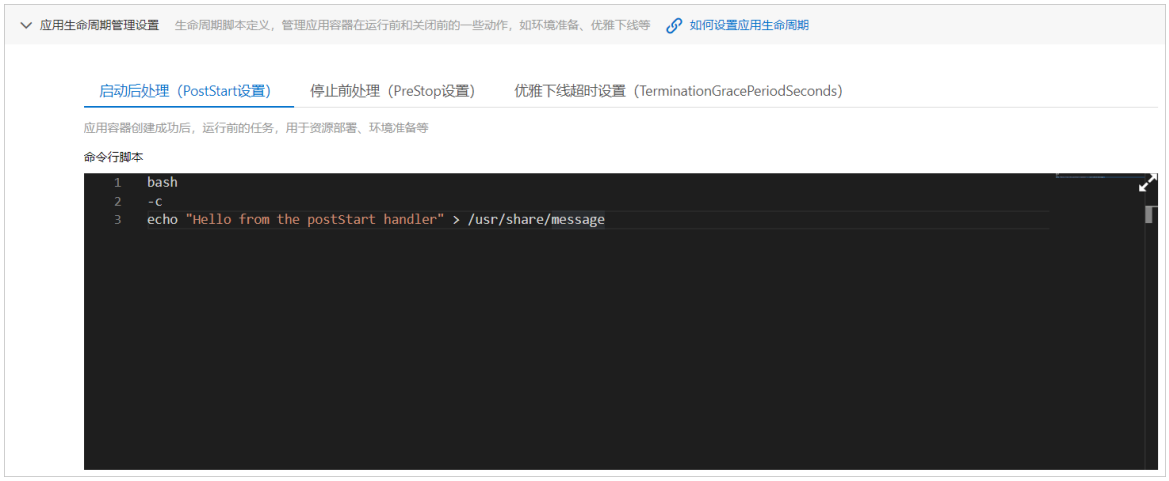
> **说明**
>
> - Poststart设置：一种容器钩子。该钩子在容器被创建后立刻触发，通知容器它已经被创建。该钩子不需要向其所对应的hook handler传入任何参数。如果该钩子对应的hook handler执行失败，则该容器会被杀死，并根据该容器的重启策略决定是否重启该容器。详情请参见Container Lifecycle Hooks。
> - PreStop设置：一种容器钩子。该钩子在容器被删除前触发，其所对应的hook handler必须在删除该容器的请求发送给Docker daemon之前完成。在该钩子对应的hook handler完成后不论执行的结果如何，Docker daemon会发送SGTERN信号量给Docker daemon删除该容器。详情请参见Container Lifecycle Hooks。

6. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

7. 结果验证。

   例如应用启动前，需要将 `Hello from the postStart handler` 内容写入文件。PostStart配置如下：



   应用部署后，使用Webshell查看文件中是否存在 `Hello from the postStart handler` ，如果存在，那么表示配置成功。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.12. 制作应用容器Docker镜像

Spring Cloud或Dubbo框架下开发并编译的应用WAR包或JAR包，如果需要在SAE上以镜像方式部署，需要将WAR包或JAR包制作为应用镜像，以便部署时使用。本文介绍制作应用镜像的使用规范、制作过程以及将镜像上传至镜像仓库的操作步骤。

使用规范

通过Dockerfile制作自定义镜像时，SAE在运行时会自动注入以下环境变量信息。

> **注意**
> - 为保证应用正常运行，请勿覆盖以下配置。
> - 请勿使用*/home/admin/.spas_key/default*文件，该文件在POD运行时将会被覆盖。

环境变量Key	描述
POD_IP	POD的IP地址。
EDAS_APP_ID	部署在SAE上应用的ID。
EDAS_ECC_ID	导入集群的ECS在应用中部署后的ID。
EDAS_PROJECT_NAME	与EDAS_ECC_ID类似，用于调用链解析。
EDAS_JM_CONTAINER_ID	与EDAS_ECC_ID类似，用于调用链解析。
EDAS_CATALINA_OPTS	中间件运行时所需的CATALINA_OPTS参数。
CATALINA_OPTS	与EDAS_CATALINA_OPTS类似，默认Tomcat启动参数。
CATALINA_HOME	安装的Tomcat路径。
PANDORA_LOCATION	安装的Pandora路径，仅HSF应用中可见。

创建标准Dockerfile

Dockerfile是以文本格式来快速创建自定义镜像的配置文件，Dockerfile的详细说明请参见Dockerfile。

标准的Dockerfile描述了SAE创建应用运行环境所需的所有指令，包括下载、安装和启动OpenJDK、Tomcat、WAR包和JAR包等。您还可以通过修改Dockerfile内容，完成如OpenJDK版本替换、Tomcat配置修改、改变运行时环境等操作，详情请参见Dockerfile。

您可以通过Dockerfile分别制作HSF、Spring Cloud或Dubbo应用镜像，创建不同框架应用的Dockerfile示例如下：

- HSF应用的Dockerfile示例（基于WAR包）
- HSF应用的Dockerfile示例（基于JAR包）
- Spring Cloud或Dubbo应用的Dockerfile示例（基于WAR包）
- Spring Cloud或Dubbo应用的Dockerfile示例（基于JAR包）

注意 制作应用镜像时请勿使用VOLUMES或USER root字段，否则应用运行时可能会导致文件丢失。

HSF应用的Dockerfile示例（基于WAR包）

```
FROM centos:7
MAINTAINER：SAE研发团队
#安装打包必备软件。
RUN yum install -y wget unzip telnet lsof net-tools bind-utils
#准备JDK以及Tomcat系统变量与路径。
ENV JAVA_HOME /usr/java/latest
ENV CATALINA_HOME /home/admin/taobao-tomcat
ENV PATH ${JAVA_HOME}/bin:${CATALINA_HOME}/bin:${PATH}
#设置EDAS-Container以及Pandora应用容器版本。
ENV EDAS_CONTAINER_VERSION V3.5.4
LABEL pandora V3.5.4
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#创建JAVA_HOME软链接。
RUN if [ ! -L "${JAVA_HOME}" ]; then mkdir -p `dirname ${JAVA_HOME}` && ln -s `readlink -f /usr/lib/jvm/java` ${JAVA_HOME}; fi
#下载安装Ali-Tomcat 7.0.92到/home/admin/taobao-tomcat目录。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-container/7.0.92/taobao-tomcat-production-7.0.92.tar.gz -O /tmp/taobao-tomcat.tar.gz && \
  mkdir -p ${CATALINA_HOME} && \
  tar -xvf /tmp/taobao-tomcat.tar.gz -C ${CATALINA_HOME} && \
  mv ${CATALINA_HOME}/taobao-tomcat-production-7.0.59.3/* ${CATALINA_HOME}/ && \
  rm -rf /tmp/taobao-tomcat.tar.gz ${CATALINA_HOME}/taobao-tomcat-production-7.0.59.3 && \
  chmod +x ${CATALINA_HOME}/bin/*sh
#根据环境变量，下载安装EDAS-Container以及Pandora应用容器版本。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/edas.sar.${EDAS_CONTAINER_VERSION}/taobao-hsf.tgz -O /tmp/taobao-hsf.tgz && \
  tar -xvf /tmp/taobao-hsf.tgz -C ${CATALINA_HOME}/deploy && \
  rm -rf /tmp/taobao-hsf.tgz
#下载部署SAE演示WAR包。
RUN wget http://edas.oss-cn-hangzhou.aliyuncs.com/demo/hello-edas.war -O /tmp/ROOT.war && \
  unzip /tmp/ROOT.war -d ${CATALINA_HOME}/deploy/ROOT && \
  rm -rf /tmp/ROOT.war
#设定Tomcat安装目录为容器启动目录，并采用run方式启动Tomcat，在标准命令行输出catalina日志。
WORKDIR ${CATALINA_HOME}
CMD ["catalina.sh", "run"]
```

HSF应用的Dockerfile示例（基于JAR包）

```
FROM centos:7
MAINTAINER：SAE研发团队
#安装打包必备软件。
RUN yum install -y wget unzip telnet lsof net-tools bind-utils
#准备JDK及Tomcat系统变量与路径。
ENV JAVA_HOME /usr/java/latest
ENV CATALINA_HOME /home/admin/taobao-tomcat
ENV PATH ${JAVA_HOME}/bin:${PATH}
ENV ADMIN_HOME /home/admin
#设置EDAS-Container以及Pandora应用容器版本。
ENV EDAS_CONTAINER_VERSION V3.5.4
LABEL pandora V3.5.4
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#创建JAVA_HOME软链接。
RUN if [ ! -L "${JAVA_HOME}" ]; then mkdir -p `dirname ${JAVA_HOME}` && ln -s `readlink -f /usr/lib/jvm/java` ${JAVA_HOME}; fi
#根据环境变量，下载安装EDAS-Container以及Pandora应用容器版本。
RUN mkdir -p ${CATALINA_HOME}/deploy
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/edas.sar.${EDAS_CONTAINER_VERSION}/taobao-hsf.tgz -O /tmp/taobao-hsf.tgz && \
  tar -xvf /tmp/taobao-hsf.tgz -C ${CATALINA_HOME}/deploy/ && \
  rm -rf /tmp/taobao-hsf.tgz
#下载部署SAE演示JAR包。
RUN mkdir -p ${ADMIN_HOME}/app && wget http://edas.oss-cn-hangzhou.aliyuncs.com/demoapp/fatjar-test-case-provider-0.0.1-SNAPSHOT.jar -O ${ADMIN_HOME}/app/provider.jar
#将启动命令写入启动脚本start.sh。
RUN echo 'eval exec java -jar $CATALINA_OPTS /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar'> /home/admin/start.sh && chmod +x /home/admin/start.sh
WORKDIR $CATALINA_HOME
CMD ["/bin/bash", "/home/admin/start.sh"]
```

Spring Cloud或Dubbo应用的Dockerfile示例（基于WAR包）

```
FROM centos:7
MAINTAINER：SAE研发团队
#安装打包必备软件。
RUN yum -y install wget unzip telnet
#准备JDK以及Tomcat系统变量。
ENV JAVA_HOME /usr/java/latest
ENV CATALINA_HOME /home/admin/apache-tomcat-7.0.91
ENV ADMIN_HOME /home/admin
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
RUN mkdir -p /home/admin
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#下载安装Tomcat。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/apache-tomcat-7.0.91.tar.gz -O /tmp/apache-tomca
t-7.0.91.tar.gz && \
  tar -xvf /tmp/apache-tomcat-7.0.91.tar.gz -C /home/admin && \
  rm /tmp/apache-tomcat-7.0.91.tar.gz && \
  chmod +x ${CATALINA_HOME}/bin/*sh
RUN mkdir -p ${CATALINA_HOME}/deploy/
#增加容器内中文支持。
ENV LANG="en_US.UTF-8"
#增强Webshell使用体验。
ENV TERM=xterm
#下载部署SAE演示WAR包。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/demo/1.0/hello-edas.war -O /tmp/ROOT.war && \
  rm -rf ${CATALINA_HOME}/webapps/ROOT &&\
  unzip /tmp/ROOT.war -d ${CATALINA_HOME}/deploy/ROOT/ && \
  rm -rf /tmp/ROOT.war
# 设定Tomcat安装目录为容器启动目录，并采用run方式启动Tomcat，在标准命令行输出catalina日志。
WORKDIR $ADMIN_HOME
CMD ["catalina.sh", "run"]
```

Spring Cloud或Dubbo应用的Dockerfile示例（基于JAR包）

```
FROM centos:7
MAINTAINER：SAE研发团队
#安装打包必备软件。
RUN yum -y install wget unzip telnet
#准备JDK以及Tomcat系统变量。
ENV JAVA_HOME /usr/java/latest
ENV PATH $PATH:$JAVA_HOME/bin
ENV ADMIN_HOME /home/admin
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#下载部署SAE演示JAR包。
RUN mkdir -p /home/admin/app/ && \
    wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/demo/1.0/hello-edas-0.0.1-SNAPSHOT.jar -O /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar
#增加容器内中文支持。
ENV LANG="en_US.UTF-8"
#增强Webshell使用体验。
ENV TERM=xterm
#将启动命令写入启动脚本start.sh。
RUN mkdir -p /home/admin
RUN echo 'eval exec java -jar $CATALINA_OPTS /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar'> /home/admin/start.sh && chmod +x /home/admin/start.sh
WORKDIR $ADMIN_HOME
CMD ["/bin/bash", "/home/admin/start.sh"]
```

自定义设置Dockerfile

通过编辑Dockerfile文件修改运行环境配置，如替换JDK版本、Tomcat修改配置、更改运行时环境等操作。

- 更换JDK版本
  在标准Dockerfile中，请参见以下示例更换其他版本的JDK。

```
# 下载安装JDK 8。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/agent/prod/files/jdk-8u65-linux-x64.rpm -O /tmp/jdk-8u65-linux-x64.rpm && \
  yum -y install /tmp/jdk-8u65-linux-x64.rpm && \
  rm -rf /tmp/jdk-8u65-linux-x64.rpm
```

- 在Tomcat启动参数中添加SAE运行环境
  SAE提供了JVM环境变量EDAS_CATALINA_OPTS，包含了运行所需的基本参数。此外Ali-Tomcat提供了自定义JVM参数配置选项JAVA_OPTS，可以设置Xmx、Xms等参数。

```
#设置SAE应用JVM参数。
ENV CATALINA_OPTS ${EDAS_CATALINA_OPTS}
#设置JVM参数。
ENV JAVA_OPTS="\
   -Xmx3550m \
   -Xms3550m \
   -Xmn2g \
   -Xss128k"
```

在本地制作镜像

从本地命令行进入Dockerfile文件所在的目录，执行 `docker build` 命令制作镜像。

```
docker build -t [标签名称，最好取应用名]:[版本号] .
或
docker build -t [标签名称，最好取应用名]:[版本号] -f /path/to/custom_dockerfile_name.#适用于您创建好的Dockerfile在其他位置或名称不为Dockerfile的场景。
```

例如：

```
docker build -t hsf-provider:1.0.0 .
```

制作完成后，执行 `docker images | grep <镜像标签名称>` 命令查看本地编译完成的镜像。

上传镜像到镜像仓库

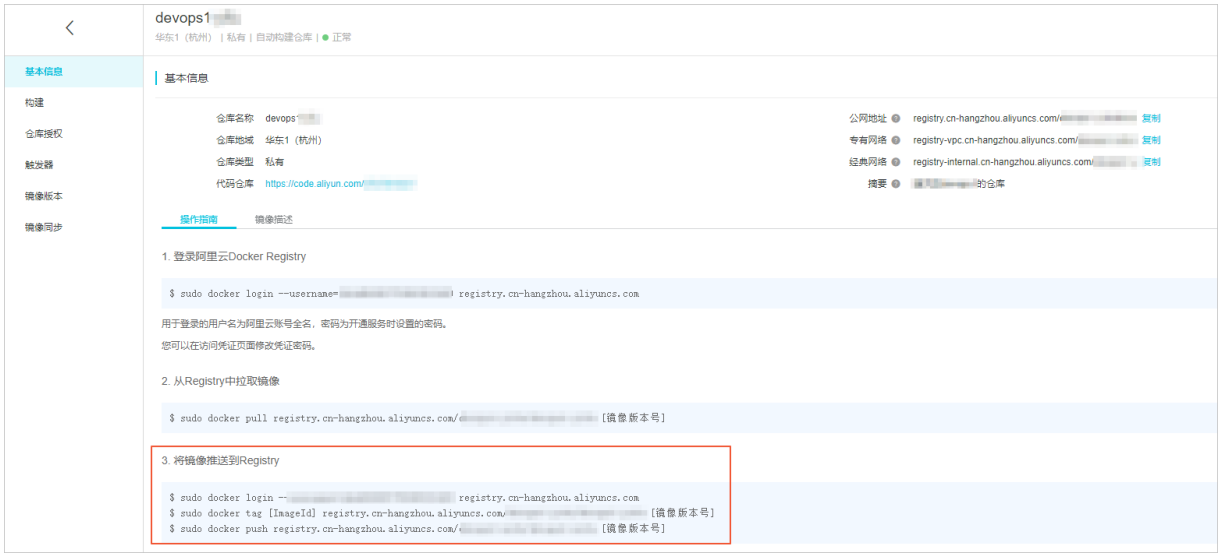
您可以将本地生成制作好的应用镜像上传到阿里云提供的容器镜像仓库，详细步骤请参见管理镜像。

执行指定镜像仓库**基本信息**页面提供的**将镜像推送到Registry**相关命令，将本地镜像上传到该镜像仓库。

```
docker login --username=[当前登录的账号名] registry.[region_id].aliyuncs.com  #输入的密码为阿里云容器镜像服务默认实例访问凭证页面设置的固定或者临时密码，并非阿里云账号密码。
docker tag [本地应用镜像ID] registry.[region_id].aliyuncs.com/[命令空间名称]/[镜像仓库名称]:[镜像版本号]
docker push registry.[region_id].aliyuncs.com/[命令空间名称]/[镜像仓库名称]:[镜像版本号]
```

例如：

```
docker login --username=aaa111@gmail.com registry.cn-hangzhou.aliyuncs.com
docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/webapps/hsf-provider:1.0.0
docker push registry.cn-hangzhou.aliyuncs.com/webapps/hsf-provider:1.0.0
```



更多信息

镜像制作完成后，您可以将其以镜像的方式部署在SAE，具体请参见在SAE控制台使用镜像部署PHP应用。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

5.13. 注入配置信息

配置项能够将环境配置信息和容器镜像解耦，方便您修改应用配置。本文介绍如何利用在SAE命名空间中创建的配置项向容器注入配置信息。

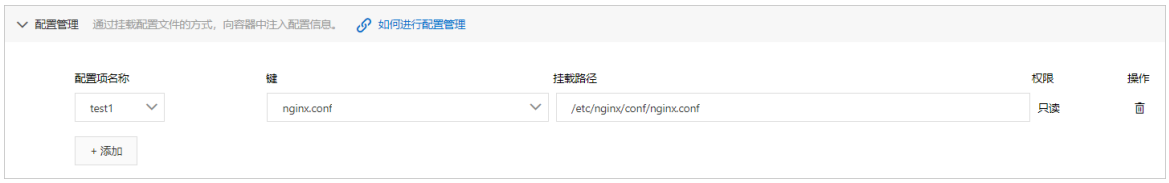
前提条件

您已在命名空间中创建配置项。具体信息，请参见管理和使用配置项（K8s）。

利用配置项向容器中注入配置信息

您可以使用在命名空间**配置项**页面创建的配置项来向容器中注入配置信息。

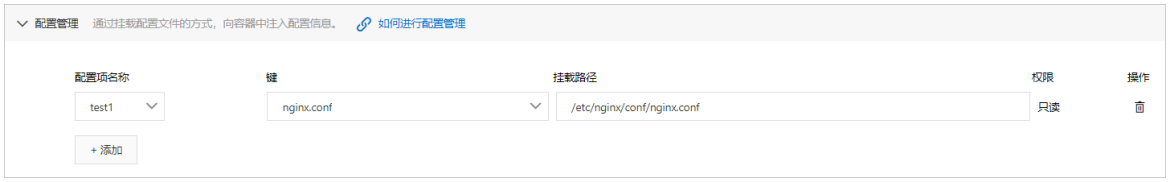
1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页签，展开**配置管理**区域，选择已创建的**配置项名称**和该配置项中的**键**，并输入挂载配置项的**挂载路径**。



6. （可选）如需引用其他配置项，单击**添加**，并重复上一步。
7. 单击**下一步：确认规格**。
8. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后注入配置信息

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**配置管理**区域，选择已创建的**配置项名称**和该配置项中的**键**，并输入挂载配置项的**挂载路径**。



5. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

6.设置限流降级

SAE支持使用AHAS实现Java应用的限流降级，全面保障您的应用的可用性。本文介绍如何在SAE设置限流降级。

前提条件

- 您已成功部署Java应用，详情请参见在SAE控制台使用WAR包部署Java Web应用。
- 您已开通AHAS，详情请参见开通AHAS。

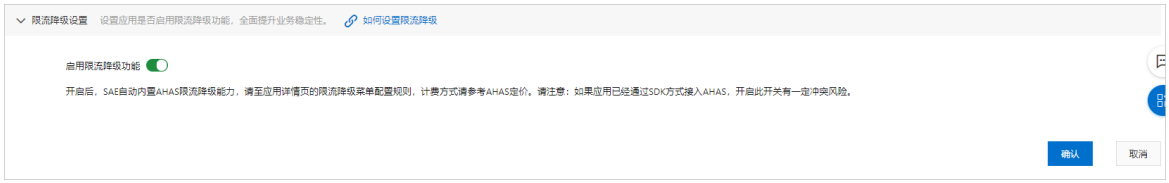
> **说明** 使用AHAS时会产生单独费用，AHAS的计费说明请参见价格及欠费说明。

背景信息

- 限流：通过调节流量阈值来控制通过系统的最大流量值，保证系统安全可靠运行。开启限流降级功能并设置限流策略后，如果流量在一定时间内超过了预设阈值，则会触发限流策略，进行限流处理。
- 降级：降级通常用于对下游出现超时的非核心应用进行低优先级调用，确保上游核心应用不被影响的场景。开启限流降级功能并设置降级策略后，如果下游应用响应时间超过了预设阈值，则会触发降级策略，进行降级处理。

操作步骤

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**限流降级设置**区域，打开**启用限流降级功能**开关，单击**确认**。



> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

开启限流降级功能后，您可以在**限流降级（仅支持Java）**菜单查看应用概览、机器监控、配置规则等，详情请参见以下文档。

功能	参考文档
应用概览	查看Java应用概览信息
机器监控	查看节点监控指标
配置流控规则	配置流控规则
配置隔离规则	配置隔离规则
配置降级规则	配置降级规则
配置系统规则	配置系统规则

功能	参考文档
配置热点规则	配置热点规则
查看操作日志	查看操作日志

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。

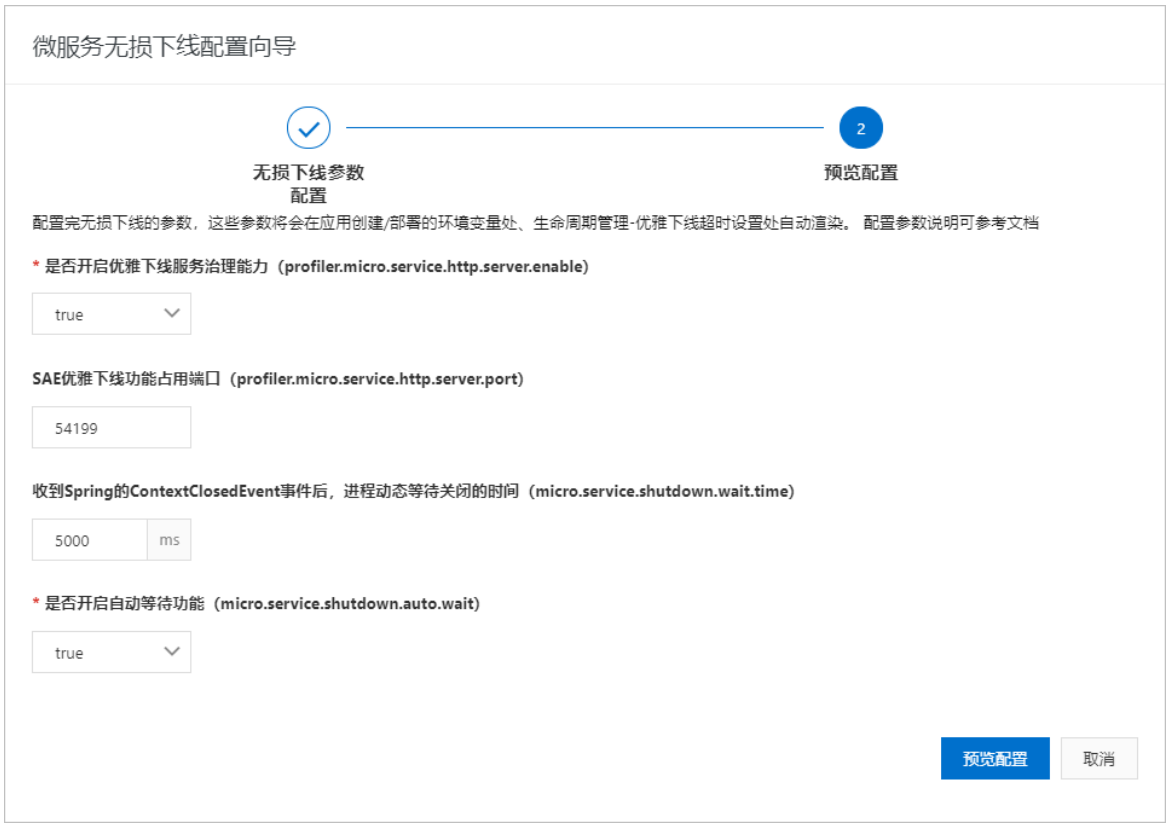
7.设置微服务无损下线

如果您期望托管至SAE的微服务应用处理完请求后再停止应用，您可以使用SAE无损下线功能。SAE支持Spring Cloud和Dubbo应用无损下线。本文介绍如何在SAE控制台上配置无损下线。

应用部署时设置微服务无损下线

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击**创建应用**。
3. 在**应用基本信息**页签设置应用相关信息，并单击**下一步：应用部署配置**。
4. 在**应用部署配置**页签，选择**技术栈语言**和**应用部署方式**，设置部署参数。
5. 在**应用部署配置**页面，展开**微服务无损下线配置**区域，单击**微服务无损下线配置向导**。

6. 在**微服务无损下线配置向导**面板，设置相关参数，单击**预览配置**。

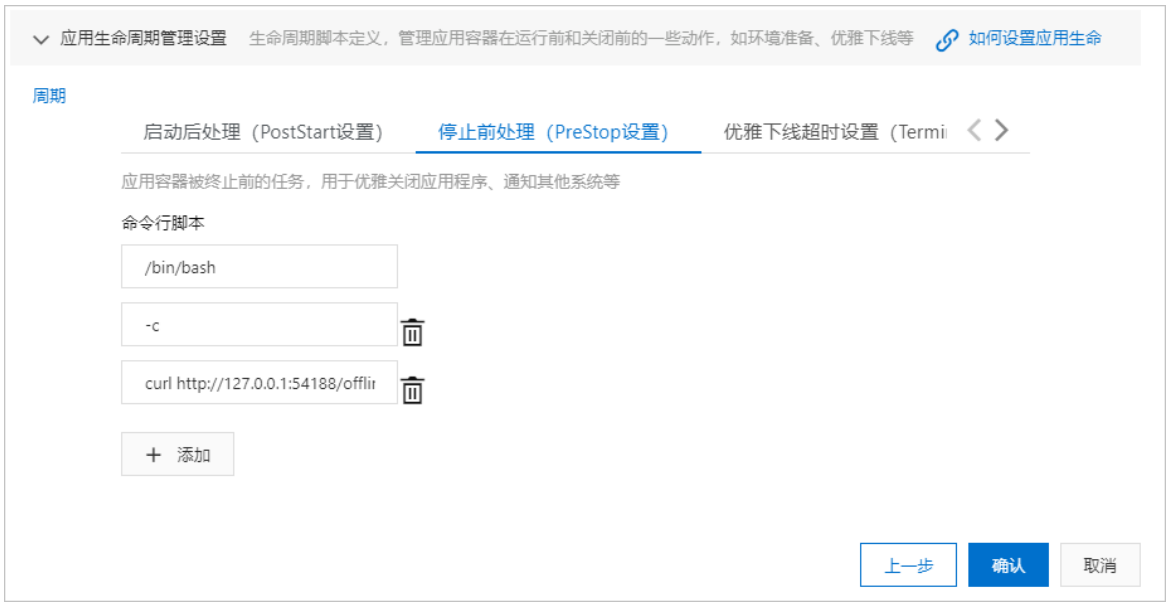


参数说明如下。

变量名称	参数说明

变量名称	参数说明
是否开启优雅下线服务治理能力	默认为true。取值说明如下： ○ true：开启。 ○ false：关闭。
SAE优雅下线功能占用端口	默认端口为54199。如果此默认端口和您应用的端口产生冲突，请配置新的端口。
收到Spring的ContextClosedEvent事件后，进程动态等待关闭的时间	默认取值为5000，即进程收到kill信号后，会等待5000 ms再关闭。 配置该参数后，需要设置是否开启自动等待功能参数为false。
是否开启自动等待功能	默认为true。取值说明如下： ○ true：开启。进程可能会随着流量停止而提前关闭。 ○ false：关闭。进程直接等待收到Spring的ContextClosedEvent事件后，进程动态等待关闭的时间所配置的值的对应时长后再关闭。

7. 在**应用生命周期管理设置**区域，确认相关配置，单击**确认**。

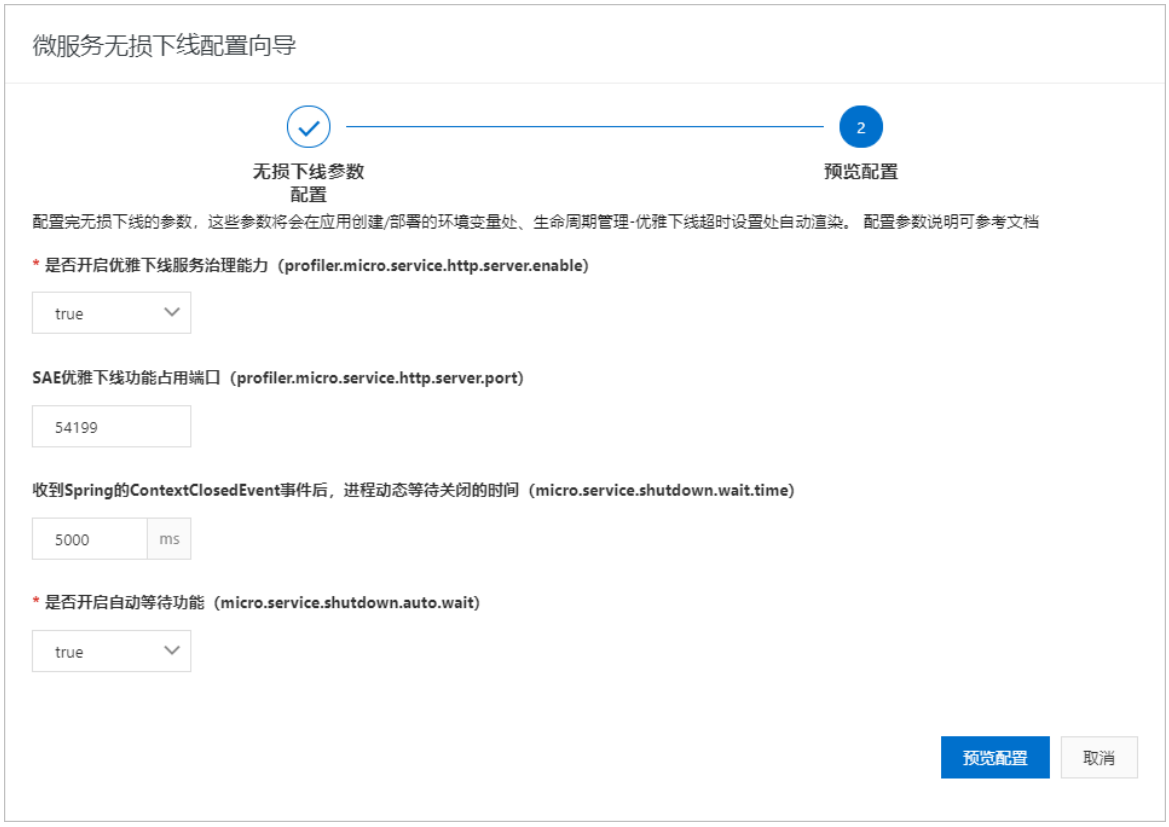


8. 单击**下一步：确认规格**。
9. 在**确认规格**页签，查看您所创建应用的详细信息以及配置费用情况，并单击**确认创建**。

应用部署完成后设置微服务无损下线

1. 登录SAE控制台。
2. 在左侧导航栏单击**应用列表**，在**应用列表**页面上方选择**地域**，单击具体应用名称。
3. 在应用详情页面的右上角，单击**部署应用**。
4. 在**部署应用**页面，展开**微服务无损下线配置**区域，单击**微服务无损下线配置向导**。

5. 在**微服务无损下线配置向导**面板，设置相关参数，单击**预览配置**。

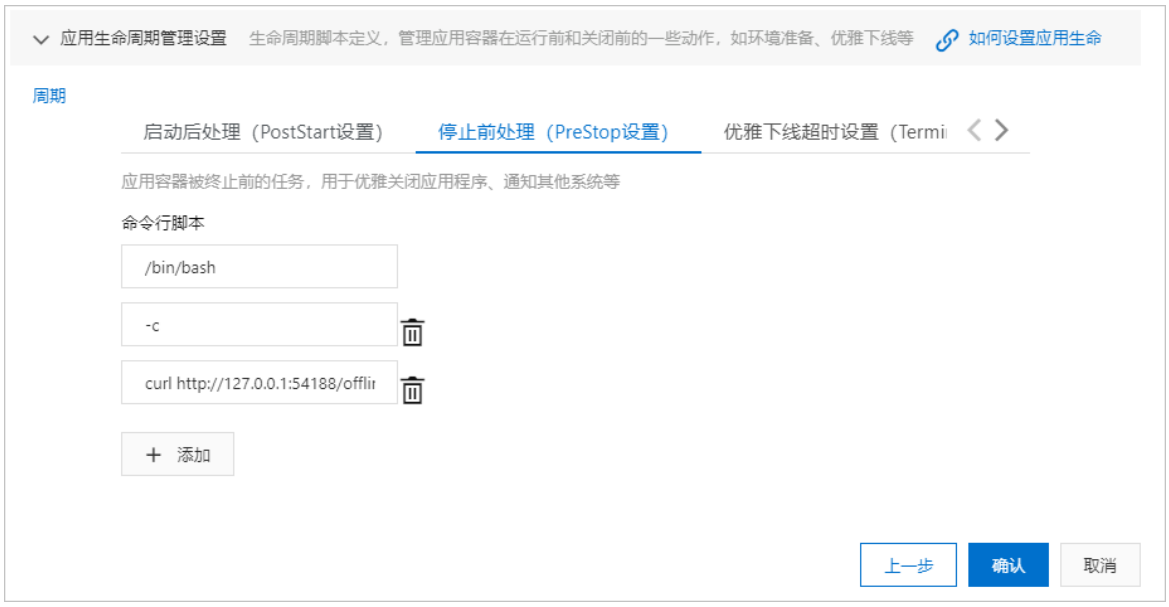


参数说明如下。

变量名称	参数说明
是否开启优雅下线服务治理能力	默认为true。取值说明如下： ○ true：开启。 ○ false：关闭。
SAE优雅下线功能占用端口	默认端口为54199。如果此默认端口和您应用的端口产生冲突，请配置新的端口。
收到Spring的ContextClosedEvent事件后，进程动态等待关闭的时间	默认取值为5000，即进程收到kill信号后，会等待5000 ms再关闭。 配置该参数后，需要设置是否开启自动等待功能参数为false。

变量名称	参数说明
是否开启自动等待功能	默认为true。取值说明如下： ○ true：开启。进程可能会随着流量停止而提前关闭。 ○ false：关闭。进程直接等待收到Spring的ContextClosedEvent事件后，进程动态等待关闭的时间所配置的值的对应时长后再关闭。

6. 在**应用生命周期管理设置**区域，确认相关配置，单击**确认**。



7. 配置完成后单击**确认**。

> **注意** 单击**确认**后，该应用将会被重启，请在业务较少的时间段进行。

问题反馈

如果您在使用SAE过程中有任何疑问，欢迎您扫描下面的二维码或搜索钉钉群号23198618，加入钉钉群与我们交流。