
Amazon Toolkit for Eclipse

用户指南

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅[中国的 Amazon Web Services 服务入门](#)。

Table of Contents

什么是AmazonToolkit for Eclipse ?	1
其他文档和资源	1
入门	2
设置 Toolkit	2
Prerequisites	2
安装 ToToolkit for Eclipse	2
升级 ToToolkit for Eclipse	3
设置Amazon凭证	3
获取您的 Amazon 访问密钥	3
添加您的Amazon访问 Toolkit for Eclipse	4
使用多个Amazon使用 Toolkit for Eclipse	4
更改Amazon凭证文件位置	5
将私有密钥与 Amazon EC2 密钥对关联	5
Toolkit for Eclipse 基础知识基础知识	6
构建AmazonJava 应用程序	6
构建和运行 Amazon Simple Queue Service 示例	6
无服务器项目	7
创建无服务器项目	7
无服务器项目蓝图	7
无服务器项目结构	7
部署无服务器项目	8
另请参阅	8
区分Amazon具有命名的资源	8
使用 Amazon Services	10
如何访问Amazon探险者	10
将 Lambda 与 Toolkit for Eclipse 配合使用	10
教程：如何创建、上传和调用Amazon Lambda函数	11
Amazon Lambda 界面参考	14
这些区域有：Amazon CloudFormation模板编辑器	18
添加和访问Amazon CloudFormation在 Eclipse 中	18
部署Amazon CloudFormation在 Eclipse 中	18
更新Amazon CloudFormation在 Eclipse 中	19
验证Amazon CloudFormation在 Eclipse 中	20
使用 DynamoDBAmazon探险者	20
创建 DynamoDB 表	20
以网格形式查看 DynamoDB 表	20
编辑属性和值	21
扫描 DynamoDB 表	21
启动 Amazon EC2 实例	21
从管理安全组Amazon探险者	22
创建新安全组	22
向安全组添加权限	23
查看和添加 Amazon SNS 通知	23
查看 Amazon SNS 通知	23
添加 Amazon SNS 通知	23
连接到 Amazon Relational Database Service (Amazon RDS)	24
Identity and Access Management	24
关于 Amazon Identity and Access Management	24
创建 IAM 用户	25
创建 IAM 组	25
将 IAM 用户添加到 IAM 组	25
管理 IAM 用户的凭证	26
创建 IAM 角色	27
将 IAM 策略附加到用户、组或角色	28

设置密码策略	28
调试无服务器应用程序AmazonSAM Local	28
Prerequisites	2
从 AWS CodeStar 应用程序导入 SAM 应用程序	29
本地调试 Lambda 函数	29
在本地测试 API Gateway	30
高级设置	30
更多信息	14
故障排除	32
AmazonCodeCommit 插件-Eclipse 无法写入到安全存储。	32
文档历史记录	33
.....	xxxiv

什么是AmazonToolkit for Eclipse ?

这些区域有：AmazonToolkit for Eclipse 是 Eclipse 集成开发环境 (IDE) 的开源插件，它让开发人员能够更轻松的开发、调试和部署使用 Amazon Web Services 的 Java 应用程序。它通过附加功能增强了 Eclipse IDE 的功能：

- 这些区域有：Amazon SDK for Java当您创建新建Amazon项目使用 Toolkit for Eclipse。
- AmazonExplorer 是 Amazon Web Services 的界面，您可以通过该界面管理Amazon来自 Eclipse 环境中的资源。
- Amazon Lambda Java 项目和无服务器应用程序模型 (SAM) 项目蓝图创建、部署和调试
- Amazon CodeCommit 存储库克隆
- 集成 AWS CodeStar
- 部署和调试 Elastic Beanstalk
- Amazon CloudFormation 模板编辑器
- Support 多个Amazon账户

Important

您可以免费使用 Toolkit for Eclipse，但您可能会产生费用Amazon创建或使用的费用Amazon [应计费的资源](#)，例如运行 Amazon EC2 实例或使用 Amazon S3 存储。您可以使用[Amazon定价计算器](#)估算使用各种Amazon资源的费用。

其他文档和资源

除了本指南外，还有许多其他资源可供 Eclipse 用户使用 Toolkit for Toolkit for Eclipse 用户使用：

- [AmazonSDK for Java 开发工具包](#)
- [AmazonSDK for Java API 参考](#)
- [Java 开发人员博客](#)
- [Java 开发人员论坛](#)
- GitHub:
 - [文档源](#)
 - [文档问题](#)
 - [工具包源](#)
 - [工具包问题](#)
- [@awsforjava \(Twitter\)](#)
- [工具包许可证](#)
- [工具包常见问题](#)
- [入门Amazon适用于 Java 的开发工具包](#)
- [使用Amazon带有 Elastic BeanstalkAmazonToolkit for Eclipse \(视频 \)](#)
- [AmazonToolkit for Eclipse : Amazon EC2 管理 \(视频 \)](#)

入门

本节提供有关 Toolkit for Eclipse 入门的信息，包括有关如何安装和配置 Toolkit for Eclipse 信息。

主题

- [设置 Toolkit \(p. 2\)](#)
- [设置 Amazon 凭证 \(p. 3\)](#)
- [将私有密钥与 Amazon EC2 密钥对关联 \(p. 5\)](#)

设置 Toolkit

本部分介绍如何安装或升级 Toolkit for Eclipse。

Prerequisites

Toolkit for Eclipse 具有以下先决条件：

- 亚马逊 Web Services 账户— 获取 Amazon 账户，转至 [Amazon 主页](#) 然后单击立即注册。通过注册，您将能够使用由 Amazon。
- 受支持的操作系统— Windows、Linux、macOS 或 Unix 上受支持 Toolkit for Eclipse。
- Java 1.8
- 适用于 Java 开发人员的 Eclipse IDE 4.2 或更高版本— 我们尽量使 Toolkit for Eclipse 保持同步，并在 [Eclipse 下载页面](#)。

Note

Eclipse 提供了多个不同的下载。我们建议安装面向企业 Java 开发人员的 Eclipse IDE，其中包括 [Eclipse Web 工具平台](#) Elastic Beanstalk 要求，[Eclipse 数据工具平台](#) 对于 Amazon SimpleDB 功能而言，[Eclipse eGit](#)，以及 [M2Eclipse](#)。如果您安装了其他版本的 Eclipse，请确保您拥有（或者已使用提供的链接安装）对这些功能的支持。

安装 ToToolkit for Eclipse

安装 Amazon Toolkit for Eclipse

1. 在 Eclipse 中，单击 Help (帮助)，然后单击 Install New Software (安装新软件)。
2. 在使用框，键入 <https://aws.amazon.com/eclipse>；键入按 Enter。
3. 选择您希望安装 Toolkit for Eclipse。单击 Select All (全选) 可一次安装所有组件。

Note

- Amazon Toolkit for Eclipse ore (在 Amazon 核心管理工具部分) 规定的；所有其他组件都是可选的。
 - 对 Amazon RDS 或 Amazon SimpleDB 管理器的 Support 要求 Eclipse 数据工具平台 (DTP) 已安装。默认情况下，DTP 与“Java EE 开发人员”版本的 Eclipse 一起安装，也可以 [单独安装](#)。
4. 完成选择后，单击 Next (下一步) (或 Finish (完成)) 以完成安装。

设置 Toolkit for Eclipse 之后，您应该[配置Amazon凭证](#) (p. 3).

Note

根据所选的选项以及诸如网络速度、服务器延迟和系统容量等因素，最长可能需要 30 分钟来完成安装。

升级 ToToolkit for Eclipse

要升级或重新安装 Toolkit for Eclipse，请使用与[安装工具包](#) (p. 2).

某些 Eclipse 版本（特别是 Mars 和 Neon），可能会由于旧版本 [Oomph 插件](#) 中的缺陷而无法提取最新的项目。要解决此问题：

1. 确保您在使用<https://aws.amazon.com/eclipse/site.xml>作为 Toolkit for Eclipse 更新站点。
2. 删除 `~/.eclipse/org.eclipse.oomph.p2/cache/` 目录以删除缓存的内容。
3. 安装 [Oomph \(Eclipse 安装程序\)](#) 的最新版本。

设置Amazon凭证

要使用 Toolkit for Eclipse 访问 Amazon Web Services，您必须将 Toolkit for Eclipse 配置为 Amazon 账户凭证。

获取您的 Amazon 访问密钥

访问密钥包含访问密钥 ID 和秘密访问密钥，用于签署对 Amazon 发出的编程请求。如果没有访问密钥，您可以使用进行创建。[入门Amazon Web Services Management Console](#). 我们建议您使用 IAM 访问密钥而不是 Amazon 根账户访问密钥。IAM 让您您可以安全地控制对您 Amazon 账户中的 Amazon 服务和资源的访问。

Note

要创建访问密钥，您必须拥有执行所需 IAM 操作的权限。有关更多信息，请参阅 IAM 用户指南中的[授予 IAM 用户管理密码策略和凭证的权限](#)。

要获取访问密钥 ID 和秘密访问密钥

1. 打开 [IAM 控制台](#)。
2. 在导航菜单上，选择用户。
3. 选择您的 IAM 用户名称 (而不是复选框)。
4. 打开安全凭证选项卡，然后选择创建访问密钥。
5. 要查看新的访问密钥，请选择显示。您的凭证与下面类似：
 - 访问密钥 ID : AKIAIOSFODNN7EXAMPLE
 - 秘密访问密钥 : wJalrXUtnFEMI/K7MDENG/bPxrFc1YEXAMPLEKEY
6. 要下载密钥对，请选择下载 .csv 文件。将这些密钥

存储在安全的位置。

Important

请对密钥保密以保护您的 Amazon 账户，切勿通过电子邮件发送密钥。请勿对企业外部共享密钥，即使有来自 Amazon 或 Amazon.com 的询问。合法代表 Amazon 的任何人永远都不会要求您提供秘密密钥。

相关主题

- [IAM 用户指南](#)中的什么是 IAM？。
- [Amazon安全凭证](#)中的Amazon一般参考。

添加您的Amazon访问 Toolkit for Eclipse

Toolkit for Eclipse 使用相同的系统进行定位和使用Amazon访问密钥与AmazonCLI 和AmazonJava 开发工具包。在 Eclipse IDE 中输入的访问密钥将保存到共享的Amazon凭证文件调用调用(credentials) 在 .aws 主目录中的子目录。

Note

可以修改凭证文件的位置。有关设置此文件位置的信息，请参阅[更改Amazon凭证文件位置 \(p. 5\)](#)。

如果你已经设置了Amazon使用凭证AmazonCLI，然后 Eclipse Toolkit for Eclipse 将自动检测并使用这些凭证。有关如何使用的更多信息AmazonCLI，请参阅[Amazon CLI命令参考](#)。

将您的访问密钥添加到AmazonToolkit for Eclipse

1. 打开 Eclipse 的Preferences (首选项) 对话框然后单击Amazon工具包在边栏里。
2. 键入或粘贴Amazon中的访问密钥 ID访问密钥 ID。
3. 键入或粘贴Amazon中的私有访问密钥秘密访问密钥。
4. 单击 Apply (应用) 或 OK (确定)，以存储访问密钥信息。

以下是一组已配置的默认凭证示例：

使用多个Amazon使用 Toolkit for Eclipse

这些区域有：Preferences (首选项) 对话框允许您添加多个信息Amazonaccount. 多个账户可能会很有用，例如为开发人员和管理员提供分别用于开发和发行/发布的资源。

单独的套装Amazon凭证存储为配置文件在共享Amazon中所述的凭证文件[添加您的Amazon访问密钥 AmazonToolkit for Eclipse \(p. 4\)](#). 所有已配置的配置文件均可在顶部的下拉框中查看。AmazonToolkit 首选项“全局配置”屏幕默认配置式。

添加一组新的访问密钥

1. 在存储库的AmazonToolkit 的“在 Eclipse 的屏幕Preferences (首选项) 对话框中，单击添加个人资。
2. 将新账户信息添加到 Profile Details (配置文件详细信息) 部分。

为 Profile Name (配置文件名称) 选择一个描述性名称，然后在 Access Key ID (访问密钥 ID) 和 Secret Access Key (秘密访问密钥) 框中输入访问密钥信息。

3. 单击 Apply (应用) 或 OK (确定)，以存储访问密钥信息。

您可以重复此过程对任意组数量的组合Amazon你需要的账户信息。

输入所有信息后Amazon账户信息，选择默认账户，方法是从默认配置式下拉列表。Amazon Explorer 会显示与默认账户关联的资源，当您通过 Toolkit for Eclipse 创建新应用程序时，该应用程序将这些凭证用于已配置的默认账户。

Note

对于分离你的另一种方法Amazon资源，请参阅[区分Amazon具有命名的资源 \(p. 8\)](#)。

更改Amazon凭证文件位置

使用 Toolkit for Eclipse 的 “Preferences (首选项)” 屏幕上，您可以更改 Toolkit 用来存储和加载凭证的位置。

要设置Amazon凭证文件位置

- 在Amazon工具包首选项对话框中，找到凭证文件位置部分，然后输入要在其中的文件的路径名。Amazon 存储的凭证。

Important

这是强烈推荐你不会存储你的Amazon任何网络共享目录或任何源代码控制管理的项目中的凭证信息。请务必严控制您的Amazon访问密钥！

将私有密钥与 Amazon EC2 密钥对关联

Toolkit for Eclipse 可以从以下位置获取您的 Amazon EC2 密钥对Amazon. 但是，您需要关联私有密钥，才能将其与 Toolkit for Eclipse 配合使用。

Amazon EC2 在Amazon将私有密钥与其关联

1. 打开 EclipsePreferences (首选项) 对话框，然后单击旁边的三角形Amazon工具包在边栏中显示 Eclipse 设置的其他类别的 Toolkit。
2. 选择 Key Pairs (密钥对)。

Eclipse 将显示密钥对的滚动列表。如果密钥对旁边显示红色的 X，那么需要将私有密钥与此密钥对关联才能使用此密钥对。

3. 右键单击密钥对，然后从上下文菜单中选择 Select Private Key File... (选择私有密钥文件...)
4. 导航到私有密钥文件，然后将其选中以将其与您的密钥对关联。

Toolkit for Eclipse 基础知识基础知识

此部分提供有关如何使用 Toolkit for Eclipse 完成常见开发任务的信息。

主题

- [构建AmazonJava 应用程序 \(p. 6\)](#)
- [无服务器项目 \(p. 7\)](#)
- [区分Amazon具有命名的资源 \(p. 8\)](#)

构建AmazonJava 应用程序

在本部分中，我们将使用 Toolkit for Eclipse 构建和运行访问的本地 Java 应用程序。Amazon资源费用。

适用于 Toolkit for Eclipse 包括 SDK for Java 的开发工具包和多个 Java 示例应用程序。Toolkit for Eclipse 实现了轻松构建和运行任意这些示例。演示 Toolkit for Eclipse 如何帮助你构建和运行Amazon在 Java 中使用应用程序，我们将使用AmazonSimpleQueueService以示例为例。这些区域有：Amazon随 Eclipse Toolkit 提供的 Explorer 可用于查看正在运行的 Amazon SQS 队列。

Note

SDK for Java 包示例在samples在开发工具包下载中的目录，也可以在[GitHub](#)。有关适用于 Java 的开发工具包本身的更多信息，请查看[AmazonSDK for Java 开发工具包](#)。

构建和运行 Amazon Simple Queue Service 示例

构建和运行 Amazon Simple Queue Service 示例

1. 单击Amazon图标在 Eclipse 工具栏上，然后单击NewAmazonJava 项目。
2. 在显示的对话框中，在 Project name (项目名称) 框中键入项目的名称，然后选择 Amazon Simple Queue Service Sample。
3. 单击 Finish (完成)。
4. 示例应用程序显示在 Project Explorer (项目浏览器) 中。展开此项目的树视图。
5. 在 src 节点下，双击 SimpleQueueService.java 源文件以在编辑器窗格中打开该文件。找到以下行：

```
System.out.println("Receiving messages from MyQueue.\n");
```

6. 右键单击编辑器窗格的左侧边缘，然后选择 Toggle Breakpoint (切换断点)。
7. 右键单击 Project Explorer 中的项目节点 (在我们的示例中，这将是名为 myJavaSqsApp 的节点)，然后依次单击 Debug As > Java Application。
8. 在 Select Java Application (选择 Java 应用程序) 对话框中，选择 SQS 应用程序，然后单击 OK (确定)。
9. 应用程序在断点处停止时，Eclipse 将询问是否应切换到调试模式。单击否 (调试透视图不包括Amazon资源管理器)。
10. 转到Amazon探险者展开Amazon SQS节点。
11. 双击 MyQueue 并查看 Java 客户端应用程序创建的队列的内容。
12. 按 F8. Java 客户端应用程序将继续运行并正常终止。
13. 在中刷新视图Amazon探险者. 您将看到 MyQueue 队列不再出现，并且应用程序在退出之前将删除该队列。

Note

如果您重复运行此示例应用程序，在接下来的运行之前，您应至少等待 60 秒。Amazon SQS 要求在删除队列之后至少 60 秒才能创建同名的队列。

无服务器项目

Toolkit for Eclipse 包含项目创建向导，可用于快速配置和创建无服务器项目，这些项目在上部署。Amazon CloudFormation 并运行 Lambda 函数以响应 RESTful Web 请求。

创建无服务器项目

创建无服务器项目

1. 选择 Amazon 选择工具栏中的图标，然后选择 New Amazon 无服务器项目... 从出现的菜单中选择。
2. 输入 Project name (项目名称)。
3. 输入项目的 Package namespace (程序包命名空间)。这将用作您的项目创建的源命名空间的前缀。
4. 选取 Select a blueprint (选择一个蓝图) 或 Select a serverless template file (选择一个无服务器模板文件)：

选择一个蓝图

选择一个用于无服务器项目的 [预定义项目蓝图 \(p. 7\)](#)。

选择无服务器模板文件

在您的文件系统中选择 JSON 格式无服务器应用程序模型 (SAM) .template 文件，以完全自定义您的无服务器项目。

Note

有关 .template 文件结构和内容的信息，请查看 GitHub 上的 [当前版本的规范](#)。

5. 按 Finish (完成) 按钮可创建新的无服务器项目。

无服务器项目蓝图

以下无服务器项目蓝图可供使用：

article

此蓝图会创建用于存储文章内容的 S3 存储桶和用于文章元数据的 DynamoDB 表。它包含用于检索的 Lambda 函数 (GetArticle) 和存储 (PutArticle) 文章，这些文章由 API Gateway 事件触发。

hello-world

一个可创建包含单个字符串的 Lambda 函数的简单蓝图。它的输出为 Hello, *value*，其中 *value* 是传递的字符串。如果未将任何字符串传递到函数，则输出为 world。

无服务器项目结构

无服务器项目向导将为您创建一个新 Eclipse 项目，其中包含以下部分：

- `src` 目录包含两个子目录，每个子目录都使用您选择的 Package namespace (程序包命名空间) 作为前缀：

`mynamespace.function`

包含由无服务器模板定义的 Lambda 函数的类文件。

`mynamespace.model`

包含通用 `ServerlessInput` 和 `ServerlessOutput` 为您的 Lambda 函数定义输入和输出模型 的类。

Note

有关模型类中使用的输入和输出格式的更多信息，请参阅[为代理资源配置代理集成](#)页面中的 页面API Gateway 开发人员指南。

- 这些区域有：`serverless.template`文件定义了Amazon项目使用的资源和 Lambda 函数（类型为“Amazon::Serverless: Function”的资源）。

部署无服务器项目

部署无服务器项目

1. 在 Eclipse 的 Project Explorer (项目浏览器) 窗口中，选择项目，然后打开上下文菜单（右键单击或长按）。
2. 在上下文菜单中选择 Amazon Web Services > Deploy Serverless Project... (Amazon Web Services > 部署无服务器项目...)。这将显示将无服务器部署到AmazonCloudFormation对话框。
3. 选择Amazon区域要使用。此字段可确定您部署的 Amazon CloudFormation 堆栈所在的位置。
4. 选择S3Bucket用于存储 Lambda 函数代码，或者选择Create按钮以创建新的 S3 存储桶以存储代码。
5. 为 Amazon CloudFormation 堆栈选择一个名称。
6. 按Finish按钮，将 Lambda 函数上传到 Amazon S3 并将项目模板部署到Amazon CloudFormation。

部署项目后，将会显示 Amazon CloudFormation 堆栈详细信息窗口，其中提供了有关您的部署及其当前状态的信息。它最初将其状态显示为 `CREATE_IN_PROGRESS`。当状态为 `CREATE_COMPLETE` 时，部署已激活。

要随时返回此窗口，请打开Amazon探险者，选择AmazonCloudFormation节点，然后选择Amazon CloudFormation你指定的堆栈。

Note

如果在部署过程中出现错误，您的堆栈可能会回滚。请参阅[故障排除](#)中的Amazon CloudFormation 用户指南有关如何诊断堆栈部署错误的信息。

另请参阅

- [Amazon无服务器应用模型 \(GitHub\)](#)
- [这些区域有：AmazonCloudFormation 模板编辑器 \(p. 18\)](#)
- [结合使用 Lambda 与AmazonToolkit for Eclipse \(p. 10\)](#)

区分Amazon具有命名的资源

在开发新的产品或功能期间，保持Amazon用于开发的资源独立于用于生产的资源。维持这种分离的一种方法在[设置Amazon凭证 \(p. 3\)](#)，即将不同的账户用于开发资源和生产资源。这种方法在使用时效果尤其好

Amazon因为 ExplorerAmazon资源管理器根据账户凭据显示资源。此部分将讨论一种替代方法，其中使用命名约定来区分开发资源和生产资源，并且在代码中实施对命名约定的支持。

基本的想法是区分你的Amazon资源，例如，通过将指定的字符串值添加到资源名称来添加指定的字符串值，例如 Simple Storage Service (Amazon S3) 桶或 Amazon SimpleDB 域。例如，您可以将 Amazon SimpleDB 域命名为“customers-dev” (供开发使用) 或“Customer Prod” (供生产使用)，而不是将此域命名为“客户-dev”。不过，如果您需要将开发代码移至生产中，则会出现问题。此时，您将需要通过大量全局搜索和替换操作来更改所有这些字符串；这一过程可能繁琐或容易出错。一个更高效的方法是，在代码中添加对命名约定的支持。

StageUtils 类公开以下方法。

```
public static String getResourceSuffixForCurrentStage()
```

getResourceSuffixForCurrentStage 方法返回一个字符串，该字符串对应于软件生命周期中使用资源的“阶段”，例如“dev”、“beta”或“prod”。随后，可以将该字符串附加到代码中使用的资源标识符。您可以使用 getResourceSuffixForCurrentStage 构建资源名称。例如，以下 getTopicName 方法返回 Amazon SNS 主题的唯一名称。请注意它如何将 getResourceSuffixForCurrentStage 中的返回值嵌入此名称。

```
private String getTopicName (Entry entry) {  
    return "entry" + StageUtils.getResourceSuffixForCurrentStage() + "-" + entry.getId();  
}
```

从 Java 系统属性“application.stage”中检索由 getResourceSuffixForCurrentStage 返回的值。您可以通过在 Elastic Beanstalk 的容器配置中设置系统属性来指定此值。

Note

在 Toolkit for Eclipse Elastic Beanstalk 应用程序中，您必须启动并运行 Elastic Beanstalk 应用程序，才能访问容器配置。更改并保存配置将导致应用程序自动使用新配置重新启动。

访问您的“Container/JVM Options”面板AmazonElastic Beanstalk 应用程序

1. InAmazon探险者在下，展开AmazonElastic Beanstalk节点和您的应用程序节点。
2. 在应用程序节点下，双击您的 Elastic Beanstalk 环境。
3. 在 Overview (概述) 窗格底部，单击 Configuration (配置) 选项卡。
4. 在 Container (容器) 区域中，配置容器选项。
5. 在 Additional Tomcat JVM command line options (更多 Tomcat JVM 命令行选项) 框中，通过添加 -D 命令行选项指定 application.stage 系统属性的值。例如，您可以使用以下语法指定字符串值应为“-beta”。

```
-Dapplication.stage=beta
```

请注意，getResourceSuffixForCurrentStage 会自动在您指定的字符串值前加上连字符。

6. 添加系统属性值之后，单击 File (文件) 菜单，然后单击 Save (保存)。Eclipse 将保存新配置。该应用程序将自动重新启动。您可以在 Eclipse 编辑器底部的 Events 选项卡上查看是否有指示新配置已成功部署到环境的事件。
7. 在应用程序重新启动后，展开Amazon SimpleDB节点在Amazon探险者。现在，您将看到一组使用您指定的字符串值的新域。

Note

有关配置容器的更多信息，请参阅[在上创建和部署 Java 应用程序AmazonElastic Beanstalk](#)中的Amazon Elastic Beanstalk开发人员指南。

使用 Amazon Services

Amazon您可以同时查看并操作多个 Amazon Web Services。本节提供有关如何访问和使用AmazonEclipse中的资源管理器视图。

假设您已在[已安装 \(p. 2\)](#)您的系统上 Toolkit on Eclipse。

主题

- [如何访问Amazon探险者 \(p. 10\)](#)
- [将 Lambda 与 Toolkit for Eclipse 配合使用 \(p. 10\)](#)
- [这些区域有 : Amazon CloudFormation模板编辑器 \(p. 18\)](#)
- [使用 DynamoDBAmazon探险者 \(p. 20\)](#)
- [启动 Amazon EC2 实例 \(p. 21\)](#)
- [从管理安全组Amazon探险者 \(p. 22\)](#)
- [查看和添加 Amazon SNS 通知 \(p. 23\)](#)
- [连接到 Amazon Relational Database Service \(Amazon RDS\) \(p. 24\)](#)
- [Identity and Access Management \(p. 24\)](#)
- [调试无服务器应用程序AmazonSAM Local \(p. 28\)](#)

如何访问Amazon探险者

显示Amazon资源管理器，单击Amazon图标，然后选择Show (显式)AmazonExplorer View.

Note

如果Amazon图标在工具栏上不可见，单击窗口菜单，然后单击开放视角 | 其他. 单击Amazon管理从 Eclipse 透视图列表中。

您可以在中展开每个节点Amazon浏览器查看资源Amazon这些信息与您的账户关联。例如，如果单击左侧的白色三角形Amazon EC2节点，它将扩展并显示与您的关联的 Amazon EC2 资源Amazonaccount. ToToolkit for Eclipse 使用Amazon在中配置的账户[设置Amazon凭证 \(p. 3\)](#)以确定要显示哪些资源。

如果您选择 Amazon EC2 的任何子节点，Eclipse 将打开包含这些资源的详细信息的视图。例如，双击实例打开一个视图，其中列出了每个 Amazon EC2 实例的相关信息，例如其公有 DNS 名称、可用区和启动时间。

将 Lambda 与 Toolkit for Eclipse 配合使用

Toolkit for Eclipse 提供对编写代码的支持[AmazonLambda](#). Lambda 是完全托管的计算服务，该服务运行您的代码以响应由自定义代码生成的事件或来自各种代码Amazon诸如 Amazon S3、DynamoDB、Kinesis、Amazon SNS 和 Amazon Cognito 等服务。

本部分着重介绍了如何使用 Toolkit for Eclipse 来创建、部署和执行 Lambda 函数。

主题

- [教程：如何创建、上传和调用Amazon Lambda函数 \(p. 11\)](#)
- [Amazon Lambda 界面参考 \(p. 14\)](#)

教程：如何创建、上传和调用Amazon Lambda函数

本教程将引导您完成典型的过程Amazon Lambda并为您提供将 Lambda 与 Toolkit for Eclipse 结合使用的第一手经验。

Important

本教程假定您有Amazon账户，已经[安装AmazonToolkit for Eclipse \(p. 2\)](#)，并且您了解 Lambda 的基本概念和功能。如果您对 Lambda 不熟悉，可以在[Lambda主页](#)和[AmazonLambda 开发人员指南](#)。

创建 Amazon Lambda 项目

要开始 Lambda 项目，请先将代码作为处理程序类中的方法实施。Toolkit for Eclipse 提供了一个新的项目向导来帮助您创建新的处理程序类。Lambda 项目是使用 POM.xml 文件管理程序包依赖关系的 Maven 项目。您可以使用 Maven 命令行工具构建、测试和部署您的应用程序。有关 Maven 的更多信息，请参阅 [Maven 项目文档](#)。

创建 Amazon Lambda 项目

1. 在 Eclipse 工具栏上，打开 Amazon Web Services 菜单（由 Amazon 主页图标），然后选择 NewAmazonLambda Java 项目。或者，在 Eclipse 菜单栏上，选择 Eclipse 菜单栏 File (文件)、New、AmazonLambda Java 项目。
2. 在关联的输入框中添加 Project name (项目名称)、Group ID (组 ID)、Artifact ID (构件 ID) 和 class name (类名称)。组 ID 和项目 ID 是标识 Maven 构建项目的 ID。本教程将使用以下示例值：

- Project name (项目名称)：HelloLambda
- Group ID (组 ID)：com.example.lambda
- Artifact ID (构件 ID)：demo
- 类名称：你好

Package Name (程序包名称) 字段是 Amazon Lambda 处理程序类的程序包命名空间。此字段的默认值是后跟 Maven 项目约定的组 ID 和项目 ID 的串联。更新 Group ID (组 ID) 和 Artifact ID (构件 ID) 字段时，此字段会自动更新。

3. 对于 Input Type (输入类型)，选择 Custom (自定义)。有关每种可用的输入类型的信息，请参阅 [NewAmazonLambda Java 项目对话框 \(p. 14\)](#)。
4. 验证条目是否类似于以下屏幕截图（如果不是，请修改），然后选择 Finish (完成)。

键入时，Source preview (来源预览) 中的代码将发生变化，以体现您在对话框中所做的更改。

5. 选择 Finish (完成) 后，Eclipse 工作区中将生成您项目的目录和源文件。新的 Web 浏览器窗口将打开，其中显示 README.html (这是在您的项目根目录中为您创建的)。README.html 提供了说明以指导您完成实施、测试、上传和调用新 Lambda 函数的后续步骤。通读此说明以熟悉此处描述的步骤。

接下来，您将在已在 Eclipse 中为您创建的 HelloLambda Java 项目中实施该函数。

实施处理程序方法

可使用 Create New Project (创建新项目) 对话框创建骨架项目。现在填入将在调用 Lambda 函数时运行的代码。(在此示例中，按照您在设置方法的输入参数时所指定，通过将字符串发送到您的函数的自定义事件。)

实施 Lambda 处理程序方法

1. 在 Eclipse 中项目浏览器打开，打开 Hello.java 中的 HelloLambda 项目。它将包含与下类似的代码。

```
package com.example.lambda.demo;
```

```
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class Hello implements RequestHandler<Object, String> {

    @Override
    public String handleRequest(Object input, Context context) {
        context.getLogger().log("Input: " + input);

        // TODO: implement your handler
        return "Hello from Lambda";
    }
}
```

2. 将 `handleRequest` 函数的内容替换为以下代码。

```
@Override
public String handleRequest(String input, Context context) {
    context.getLogger().log("Input: " + input);
    String output = "Hello, " + input + "!";
    return output;
}
```

允许 Lambda 代入 IAM 角色

要使 Lambda 能够访问 Lambda 函数，您必须创建一个 IAM 角色以向其提供对您的访问权限的 IAM 角色。Amazon 资源费用。您可以通过两种方式创建角色：Amazon Web Services Management Console 或使用 Toolkit for Eclipse。本节描述如何在控制台中创建 IAM 角色。请参阅[上传代码 \(p. 13\)](#)使用 Toolkit for Eclipse 创建一个。

为 Lambda 创建 IAM 角色

1. 登录到[Amazon 管理控制台](#)。
2. 从 Services (服务) 菜单中，打开 **IAM 控制台**。
3. 在导航窗格中，选择 Roles (角色)，然后选择 Create role (创建角色)。
4. 适用于选择受信任实体的类型，选择 Amazon 服务，然后选择 Lambda 对于将使用此角色的服务。接下来，选择 Next (下一步)：Permissions (下一步：权限)。
5. 对于 Attach permissions policy (附加权限策略)，选择 AWSLambdaBasicExecutionRole。这允许 Lambda 写入您的 CloudWatch Logs 资源。接下来，选择 Next (下一步)：审核。
6. 为您的角色添加一个名称 (如 hello-lambda-role) 和角色描述。然后选择 **创建角色** 完成 IAM 角色的创建。

为 Lambda 代码创建 Amazon S3 存储桶

Amazon Lambda 在您上传 Java 项目时，需要 Amazon S3 存储桶来存储此项目。您可以使用中已存在的存储桶 Amazon 您将在其中运行代码的区域，也可以创建专供 Lambda 使用的新代码 (推荐)。

您可以通过两种方式创建 Amazon S3 存储桶：Amazon Web Services Management Console 或使用 Toolkit for Eclipse。本节描述如何在控制台中创建 Amazon S3 存储桶。请参阅[上传代码 \(p. 13\)](#)使用 Toolkit for Eclipse 创建一个。

创建与 Lambda 配合使用的 Amazon S3 存储桶

1. 登录到[Amazon 管理控制台](#)。

2. 从 Services (服务) 菜单中，打开 [S3 控制台](#)。
3. 选择创建存储桶。
4. 输入存储桶名称，然后为存储桶选择区域。此区域应为您打算运行 Lambda 函数的区域。有关 Lambda 支持的区域的列表，请参阅 [区域和终端节点](#) 中的 Amazon 一般参考。
5. 选择 Create (创建) 完成存储桶的创建。

上传代码

接下来，将代码上传到 Amazon Lambda，以准备使用 Amazon Web Services Management Console 进行调用。

将函数上传到 Lambda

1. 在 Eclipse 代码窗口中右键单击，选择 AmazonLambda，然后选择将函数上传到 AmazonLambda。
2. 在存储库的 Select Target Lambda Function (选择目标 Lambda 函数) 页面上，选择 Amazon 要使用的区域。此区域应当是您为 [Amazon S3 存储桶 \(p. 12\)](#) 选择的同一区域。
3. 选择 Create a new Lambda function (新建 Lambda 函数)，然后为函数键入名称（例如，HelloFunction）。
4. 选择 Next (下一步)。
5. 在存储库的函数配置() 在页面上，输入目标 Lambda 函数的描述，然后选择您的函数将使用的 IAM 角色和 Amazon S3 存储桶。

有关可用选项的更多信息，请参阅 [将函数上传到 AmazonLambda 对话框 \(p. 15\)](#)。

6. 在存储库的函数配置页面上，选择 Create 在 Function Role 如果要为 Lambda 函数创建新的 IAM 角色。在 Create Role (创建角色) 对话框中，输入角色名称。
7. 在存储库的函数配置页面上，选择发布新版本如果您希望上传创建 Lambda 函数的新版本。要了解有关 Lambda 中的版本控制和别名的更多信息，请参阅 [AmazonLambda 函数版本控制和别名](#) 中的 Amazon Lambda 开发人员指南。
8. 如果选择发布新版本，将启用 Provide an alias to this new version (为此新版本提供一个别名) 选项。如果您要将别名与此版本的 Lambda 函数关联，请选择此选项。
9. 在存储库的函数配置页面上，选择 Create 中的用于函数代码的 S3 存储桶部分如果要为 Lambda 函数创建新的 Amazon S3 存储桶。在 Create Bucket (创建存储桶) 对话框中，输入存储桶名称。
10. 在 S3 Bucket for Function Code (用于函数代码的 S3 存储桶) 部分中，您还可选择加密上传的代码。对于本示例，保留选中的 None (无)。要了解有关 Amazon S3 加密的更多信息，请参阅 [使用服务器端加密保护数据](#) 中的 Amazon Simple Storage Service 用户指南。
11. 保留 Advanced Settings (高级设置) 选项不变。Toolkit for Eclipse 将为您选择默认值。选择 Finish 将 Lambda 函数上传到 Amazon。

如果上传成功，您将在项目浏览器查看。

如果您没看到这种情况发生，请打开 Eclipse 错误日志查看。Lambda 将有关函数上传或运行失败的信息写入此错误日志，以便您可调试它们。

调用 Lambda 函数

现在，您可以在 Amazon Lambda 上调用该函数。

调用 Lambda 函数

1. 在 Eclipse 代码窗口中右键单击，选择 AmazonLambda，然后选择打开函数 AmazonLambda。
2. 选择您要调用的处理程序类。

3. 在输入框中，键入有效的 JSON 字符串，例如"AmazonLambda"。

Note

您可以将新的 JSON 输入文件添加到您的项目，如果文件名以 .json 结束，文件就将显示在此对话框中。您可以使用此功能为 Lambda 函数提供标准输入文件。

4. 默认情况下，Show Live Log (显示实时日志) 框为选中状态。这将在 Eclipse 中显示来自 Lambda 函数输出的日志。控制台。
5. 选择 Invoke 将输入数据发送到 Lambda 函数。如果已正确设置所有项，则应在 Eclipse Console (控制台) 视图 (如果该视图尚未显示，将自动显示) 中看到函数已输出的返回值。

恭喜，您已从 Eclipse IDE 直接运行第一个 Lambda 函数！

后续步骤

现在，您已上传和部署您的函数，请尝试更改代码并重新运行函数。Lambda 将为您自动重新上传和调用函数，并将输出发送到 Eclipse。控制台。

更多信息

有关本教程中涵盖的每个页面的更多信息以及每个选项的完整说明，请参阅 [Amazon Lambda 接口参考 \(p. 14\)](#)。

有关 Lambda 以及编写 Lambda Java 代码的更多信息，请参阅在 [Java 中创作 Lambda 函数](#) 中的 Amazon Lambda 开发人员指南。

Amazon Lambda 界面参考

本部分提供适用于 Toolkit for Eclipse 添加到 Eclipse 的各个用户界面元素的详细信息。Amazon Lambda。

主题

- [NewAmazon LambdaJava Project 对话框 \(p. 14\)](#)
- [将函数上传到Amazon Lambda对话框 \(p. 15\)](#)
- [运行Amazon Lambda对话函数 \(p. 17\)](#)

NewAmazon LambdaJava Project 对话框

这些区域有：New Lambda Java 项目(对话框) 可帮助您创建和配置可用于编写 Lambda 函数的新 Java 项目。

启动对话框

New Lambda Java Project (新建 Lambda Java 项目) 对话框可通过以下方式启动：

- 打开 AmazonEclipse 工具栏中的菜单，然后选择 NewAmazonLambda Java 项目....
- 通过选择文件 > 新 > 其他... 在 Eclipse 菜单中，然后选择 Amazon > AmazonLambda Java 项目在生成的对话框中。

“Create Project Dialog”用户界面

项目名称

必填项。您必须提供项目名称。

程序包名称

Java 程序包的可选名称。它必须是有效的 Java 程序包名称，例如“com.mycompany.myproject”。在文本输入字段中输入程序包名称之后，该名称将添加到 Source Preview (来源预览) 窗口的内容中。

默认值：无，此参数是可选项。

类名称

必填项。用于标识包含 Lambda 代码的 Java 类的名称。它必须是有效的 Java 类名称。默认值为通用值。您可以在此处指定自己的名称，或更改 Package name (程序包名称) 以免与具有类似名称的类发生冲突。

默认值：LambdaFunctionHandler

输入类型

必填项。输入类型将用于调用 Lambda 函数。您可从下拉列表中选择一种类别：

- S3 事件 - 接收来自 [Amazon S3](#) 的事件。
- SNS 事件 - 接收来自 [Amazon SNS](#) 的事件。
- Kinesis 事件 - 接收来自 [Amazon Kinesis 流](#) 的事件。
- Cognito 事件 - 接收来自 [Amazon Cognito](#) 的事件。
- Custom (自定义)-接收自定义代码的事件。如果输入类型设置为 Custom (自定义)，那么还可以在类型选择旁边的框中，设置自定义输入类型的名称。默认情况下，使用通用 Object (对象) 类型。

Important

自定义输入类型必须是有效的 Java 类名称，而不是基元类型，例如 int、float 等。在这些情况下，您可以使用 Java 的标准盒装类型 (Integer、Float 等)。

可使用 Custom (自定义) 输入类型设置类似下面的事件源：

- [用户应用程序](#)
- [移动应用程序](#)
- 这些区域有：[Amazon 管理控制台](#)。
- 这些区域有：[Amazon CLI 调用命令](#)。

默认值：S3 事件

输出类型

输出类型。这必须是有效的 Java 对象。

默认值：Object

将函数上传到 Amazon Lambda 对话框

您将函数上传到 Amazon Lambda (用于创建 Lambda 函数并上传在调用 Lambda 函数时要运行的代码)。

启动此对话框

您可以启动将函数上传到 Amazon Lambda 对话框有两种方式：

- 打开您的上下文菜单 Amazon Eclipse 中的 Lambda Java 项目项目浏览器查看，然后选择 Amazon Web Services、将函数上传到 Amazon Lambda。
- 在 Java 类所在的代码窗口中打开上下文菜单，然后选择 Amazon Lambda、将函数上传到 Amazon Lambda。

这些区域有：将函数上传到 Amazon Lambda 对话框有两个页面：

- [Select Target Lambda Function \(选择目标 Lambda 函数\)](#) (p. 16)
- [函数配置](#) (p. 16)

“Select Target Lambda Function”选项

Select the Handler

(必需) 包含要上传的 Lambda 函数代码的处理程序类。

(默认值) 最近上传的处理程序或找到的第一个处理程序 (如果之前未上传任何处理程序)。

选择Amazon区域

(必需) 要在其中创建 Lambda 函数的区域。

(默认值) 默认值Amazon Web Services Management Console您的区域Amazonaccount.

Select or Create a Lambda Function

(必需) 您必须选择是使用下拉列表中的现有 Lambda 函数，还是通过输入函数名称创建一个新函数。

(默认值) Create a new Lambda function (创建新的 Lambda 函数)

当选择 Next (下一步) 时，Function Configuration (函数配置) 页面将打开。

“Function Configuration”选项

此页面分为五个部分，每个部分都具有自己的设置。

基本设置

此部分显示函数名称并允许您添加文本说明。

名称

(不可变) 此名称取决于您在 Select Target Lambda Function (选择目标 Lambda 函数) 页面上选择的名称。在此处不能修改该名称，但可以选择 Back (上一步)，在上一页中重新输入。

描述

(可选) 函数的文本描述。

(默认值) 描述为空。

Function Role

在此部分中，您可以选择要应用于函数的 IAM 角色。您还可以使用创建新的 IAM 角色Create按钮。您通过 Toolkit for Eclipse 创建的 IAM 角色是提供对 Amazon S3 的访问权限的基本角色。如果您需要更多访问权限 Amazon资源，您必须提供对中使用的每项服务的访问权限。Amazon Web Services Management Console.

IAM 角色

(必需) Lambda 用于访问您的角色Amazon函数执行期间的资源。

(默认值) 您的第一个 IAM 角色Amazonaccount.

Function Versioning and Alias

在此部分中，您可以发布新版本的 Lambda 函数并为该版本指定别名。要了解有关 Lambda 版本控制和别名的更多信息，请参阅[AmazonLambda 函数版本控制和别名](#)中的Amazon Lambda开发人员指南。

发布新版本

(默认值) 未选中。如果您选择此选项，上传将创建新版本的 Lambda 函数而不会替换它。
为此新版本提供一个别名

(默认值) 未选中。如果您选择此选项，可键入新的别名或使用现有名称。

用于函数代码的 S3 存储桶

在此部分中，您可以为 Lambda 函数设置要使用的 Amazon S3 存储桶。您还可以使用 Create 按钮并选择设置以在 Lambda 函数上传到 Amazon S3 时对其进行加密。

S3Bucket

(必需) 您的函数代码可使用的 Amazon S3 存储桶。此处只会显示位于您将运行函数的区域内的存储桶。

(默认值) 列表中的第一个存储桶或您将 Lambda 函数上传到的最后一个存储桶。

Encryption setting

(默认值) 选中“None”。要了解有关 Amazon S3 加密的更多信息，请参阅[使用服务器端加密保护数据](#)中的 Amazon Simple Storage Service 用户指南。

高级设置

此部分包含您可能不常使用的设置。与 Function Execution (函数执行) 部分中的设置相比，这些设置能为您提供更多的函数执行环境控制。

内存 (MB)

(必需) 可用于 Lambda 函数的内存的兆字节数。

(默认值) 512 MB。

超时 (秒)

(必需) 函数在完成执行的情况下被视为已失败之后的超时 (以秒为单位)。

(默认值) 15 秒。

运行 Amazon Lambda 对话函数

这些区域有：运行 Lambda 函数(对话框) 为您提供了从 Eclipse 用户界面直接调用 Lambda 函数的方法。

启动对话框

Run Lambda Function (运行 Lambda 函数) 对话框可通过以下方式启动：

- 通过打开你的上下文菜单 AmazonEclipse 中的 Lambda Java 项目项目浏览器查看，然后选择 Amazon Web Services > 运行函数 AmazonLambda...
- 在 Java 类所在的代码窗口中打开上下文菜单，然后选择 AmazonLambda > 运行函数 AmazonLambda...

“Invoke Function”对话框将显示如下：

Options

可通过两种方式将数据提供给您的函数。必须采用这两种方式之一。

- 选择其中一个 JSON 文件作为输入— 如果您有 .json 附加到项目的文件，可从提供的列表中选择其中一个文件。否则，此选项将灰显。
- 或者输入您的函数的 JSON 输入— 您可以在此为函数直接输入有效的 JSON 输入。您输入的数据类型必须与您的处理程序类中的 Java 方法的输入参数保持一致。

做出选择并提供输入数据之后，可以单击 Finish 调用 Lambda 函数，或单击 Cancel 退出对话框而不运行任何东西。

这些区域有：Amazon CloudFormation模板编辑器

Toolkit for Eclipse 包含一个内置的 Amazon CloudFormation 模板编辑器。在支持的功能中：

- 能够在当前编辑的模板中从 Eclipse IDE 直接创建和更新堆栈。
- 一个 JSON 验证程序，可帮助确保您的模板符合 JSON 格式和内容规则。

主题

- [添加和访问 Amazon CloudFormation 在 Eclipse 中 \(p. 18\)](#)
- [部署 Amazon CloudFormation 在 Eclipse 中 \(p. 18\)](#)
- [更新 Amazon CloudFormation 在 Eclipse 中 \(p. 19\)](#)
- [验证 Amazon CloudFormation 在 Eclipse 中 \(p. 20\)](#)

添加和访问 Amazon CloudFormation 在 Eclipse 中

将 CloudFormation 模板添加到 Eclipse 项目

1. 在系统的文件管理器中找到要添加到项目中的模板，将该文件拖放到项目的 Package Explorer (程序包浏览器) 窗口中。
2. 选择要将文件添加到项目的方式，然后单击 OK (确定)。

访问 Eclipse 项目中的 CloudFormation 模板

- 在 Package Explorer (程序包浏览器) 中双击模板名称，开始编辑该文件。

Note

以 .template 或 .json 结尾的文件将自动使用 Amazon CloudFormation 模板编辑器。如果文件未自动识别为 Amazon CloudFormation 模板，可以通过以下方式选择编辑器：在 Package Explorer (程序包浏览器) 中右键单击文件名，或右键单击已加载文件的编辑器窗口并依次选择 Open With (打开方式) 和 CloudFormation Template Editor (CloudFormation 模板编辑器)。

部署 Amazon CloudFormation 在 Eclipse 中

Note

只有以 .template 结尾的文件才能从 Eclipse IDE 启动。如果您的文件以其他扩展名结尾，例如 .json，您需要先使用 .template 扩展名将其重命名才能使用此功能。

从 Eclipse 部署 CloudFormation 模板

1. 有了你的Amazon CloudFormation.template在中打开文件Amazon CloudFormation模板编辑器 (请参阅[添加和访问AmazonCloudFormation 模板在 Eclipse 中 \(p. 18\)](#)有关详细信息) , 右键单击打开的模板并选择运行Amazon , 那么创建堆栈在上下文菜单中。
2. 在 Create New CloudFormation Stack (创建新 CloudFormation 堆栈) 对话框中 , 在 Stack Name (堆栈名称) 字段中输入堆栈名称。模板文件应在 Template File (模板文件) 字段中自动选择。
3. 选择以下任一选项 (或不选择任何选项) :

SNS 主题— 从列表中选择现有 SNS 主题以接收有关堆栈进度的通知 , 或者在框中键入电子邮件地址并单击以创建一个新的主题。创建新主题。

Creation Timeout - 指定在堆栈被声明失败 (并回滚 , 除非已取消选中 Amazon CloudFormationRollback on failure 选项) 之前 应留出的创建堆栈的时间量。

Rollback on failure (失败时回滚)— 如果您希望堆栈在失败时回滚 (自行删除) , 请选中此选项。如果您出于调试目的希望堆栈保持活动状态 , 请取消选中此项 , 即使堆栈未能完成启动也是如此。

4. 单击 Next (下一步) 以继续输入参数值。
5. 如果您的堆栈有参数 , 接下来您将为其输入值。对于带有预定义的可能响应列表的参数 , 您可从提供的列表中选择值。
6. 单击 Finish (完成) 以开始启动堆栈。

堆栈启动期间 , 您可以在堆栈启动期间 , 双击堆栈名称来查看其状态CloudFormation在中的节点Amazon探险者查看 , 或者右键单击堆栈名称并选择在堆栈编辑器中打开在上下文菜单中。

Note

如果你看不到你在中启动的堆栈Amazon探险者 , 您可能需要通过单击Refresh (刷新) Amazon探险者在顶部的图标Amazon探险者查看。

更新Amazon CloudFormation在 Eclipse 中

从 Eclipse 更新 CloudFormation 模板

1. 有了你的Amazon CloudFormation.template在中打开文件Amazon CloudFormation模板编辑器 (请参阅[添加和访问AmazonCloudFormation 模板在 Eclipse 中 \(p. 18\)](#)有关详细信息) , 右键单击打开的模板并选择运行Amazon , 那么更新堆栈在上下文菜单中。
2. 如果未自动选择 , 可在 Update CloudFormation Stack (更新 CloudFormation 堆栈) 对话框中 , 在 Stack Name (堆栈名称) 字段中选择堆栈名称。模板文件也应在 Template File (模板文件) 字段中自动选择。
3. 选择以下任一选项 (或不选择任何选项) :

SNS 主题— 从列表中选择现有 SNS 主题以接收有关堆栈进度的通知 , 或者在框中键入电子邮件地址并单击以创建一个新的主题。创建新主题。

Creation Timeout - 指定在堆栈被声明失败 (并回滚 , 除非已取消选中 Amazon CloudFormationRollback on failure 选项) 之前 应留出的创建堆栈的时间量。

Rollback on failure (失败时回滚)— 如果您希望堆栈在失败时回滚 (自行删除) , 请选中此选项。如果您出于调试目的希望堆栈保持活动状态 , 请取消选中此项 , 即使堆栈未能完成启动也是如此。

4. 单击 Next (下一步) 以继续输入参数值。
5. 如果您的堆栈有参数 , 接下来您将为其输入值。对于带有预定义的可能响应列表的参数 , 您可从提供的列表中选择值。
6. 单击 Finish (完成) 以开始更新堆栈。

在堆栈更新期间 , 您可以在堆栈更新期间 , 双击堆栈名称来查看其状态CloudFormation在中的节点Amazon探险者查看 , 或者右键单击堆栈名称并选择在堆栈编辑器中打开在上下文菜单中。

验证Amazon CloudFormation在 Eclipse 中

在 Eclipse 中验证 CloudFormation 模板

- 执行以下任一操作：
 - 在 Package Explorer (程序包浏览器) 视图中右键单击模板名称，并在上下文菜单中单击 Validate (验证)。
 - 在编辑器窗格中右键单击正在编辑的模板，然后在上下文菜单中单击 Validate (验证)。

Important

仅验证模板的 JSON 正确性，将不会验证模板的 CloudFormation 正确性。通过此方式验证堆栈模板后，仍可能会在启动或更新时失败。

使用 DynamoDB Amazon 探险者

Amazon DynamoDB 是一项快速、高度可扩展、高度可用且经济实惠的非关系数据库服务。DynamoDB 消除了传统上对数据存储可扩展性的限制，同时保留了低延迟性和可预测的性能。Toolkit for Eclipse 提供了用于在开发上下文中使用 DynamoDB 的功能。

在 Toolkit for Eclipse 中，Amazon Explorer 显示了与活动表关联的所有 DynamoDB 表。Amazon account.

创建 DynamoDB 表

使用 Toolkit for Eclipse，您可以创建新 DynamoDB 表。

要在中创建新表 Amazon 探险者

1. 在 Amazon 探险者中，右键单击 Amazon DynamoDB，然后单击创建表。此时将打开 Create New DynamoDB Table (创建新 DynamoDB 表) 向导。
2. 在 Table name (表名称) 框中输入表名。
3. 在哈希键属性框中，然后从哈希键类型下拉列表中)。DynamoDB 使用主键属性构建无序哈希索引，并使用范围主键属性构建可选的有序范围索引。有关主哈希键属性的更多信息，请参阅 [分区和数据分配](#) 中的 Amazon DynamoDB 开发人员指南。
4. (可选) 通过选择 Use a range key (使用范围键) 指定范围主键。在 Range key attribute (范围键属性) 框中输入范围键属性，然后从 Range key type (范围键类型) 下拉列表中选择范围键类型。
5. 在 Read capacity units (读取容量单位) 框中指定读取容量单位的数量，然后在 Write capacity units (写入容量单位) 框中指定写入容量单位的数量。您必须至少指定 3 个读取容量单位和 5 个写入容量单位。有关读取和写入容量单位的更多信息，请参阅 [预配置吞吐量](#) 中的 Amazon DynamoDB 开发人员指南。
6. 单击 Finish (完成) 以创建表。点击中的刷新按钮 Amazon 探险者以在表列表中查看新表。

以网格形式查看 DynamoDB 表

要打开 DynamoDB 表的网格视图，请在中双击子节点。Amazon 探险者这对应于表格。从网格视图中，您可以查看存储在表中的项目、属性和值。每个行对应于表中的一个项目。表列与属性对应。表的每个单元格保存与该项目的该属性关联的值。

属性可以包含字符串或数字形式的值。某些属性包含由一系列字符串或数字组成的值。系列值显示为用方括号括起的逗号分隔列表。

编辑属性和值

表格视图是可编辑的；通过双击单元格，您可以编辑项目对应属性的值。对于系列值属性，您还可以在该系列中添加或删除单个值。

使用编辑 UI 不仅可以更改属性的值，而且还可更改属性的值的格式，但有一些限制。例如，任何数字值均可转换为字符串值。如果您有一个字符串值，内容为数字（例如“125”），那么编辑 UI 可让您将值的格式从字符串转换为数字。此外，编辑 UI 可让您将单一值转换为系列值。但是，您通常无法将系列值转换为单一值；有一个例外情况，即当系列值实际上只包含一个元素时。

Edit Values (编辑值) 对话框在您编辑值集时打开。编辑属性值之后，单击 Save set (保存集) 以确认更改。如果要放弃更改，可单击 Cancel (取消)。

确认更改后，属性值将以红色显示。这表示属性已更新，但新值尚未写回到 Amazon DynamoDB 数据库。要更改写回到 DynamoDB，请单击 File (文件)，然后单击 Save (保存)，或者从键盘上按下。要放弃更改，可单击 Scan Table (扫描表)，当 Toolkit 询问是否要在扫描之前提交更改时，单击 No (否)。

扫描 DynamoDB 表

从 Toolkit，您可以在 DynamoDB 表上执行扫描。在一次扫描中，您将定义一组条件，扫描将返回表中符合您的条件的的所有项目。扫描是代价高昂的操作，应谨慎使用以避免干扰表中优先级更高的生产流量。有关安全使用扫描操作的更多建议，请转到 Amazon DynamoDB 开发人员指南。

从以下方式对 Amazon DynamoDB 表执行扫描 Amazon 探险者

1. 在网格视图中，单击 Add scan condition (添加扫描条件)。此时将显示可供您编辑新的扫描子句的 UI。
2. 在扫描子句编辑器中，指定要与之匹配的属性、应如何匹配属性（开头为、包含等）、应匹配什么文字值以及该值应为字符串还是数字。
3. 根据需要为您的搜索添加更多扫描子句。扫描只会返回符合所有扫描子句中的条件的的项目。请注意，当与字符串值匹配时，扫描将执行区分大小写的比较。
4. 在网格视图顶部的按钮栏中，单击绿色播放按钮以运行扫描。

要删除扫描子句，请单击每个子句左侧的红色“X”。

要返回到包含所有项目的表的视图，请双击 Amazon DynamoDB 在 Amazon 探险者。

为扫描结果分页

视图顶部有三个按钮。

第二个按钮为扫描结果提供分页。单击最右侧按钮可将当前扫描的结果导出到 CSV 文件。

启动 Amazon EC2 实例

启动 EC2 实例之前，您应该创建一个安全组，以允许适用于您的应用程序的网络流量连接到该实例。安全组应该至少允许访问端口 22，这样才能使用 SSH 连接到 EC2 实例。您可能还想创建密钥对，但您也可以在完成启动向导时创建密钥对。最后，您应该考虑适用于您的应用程序的实例类型；EC2 实例类型的功能越强，价格通常越高。有关更多信息，请参阅<http://www.amazonaws.cn/ec2/pricing/>。

启动 Amazon EC2 实例

1. 在 Amazon 探险者中，展开 Amazon EC2 节点。右键单击 Amazon Machine Images (AMIs) (Amazon 系统映像 (AMI)) 子节点，然后选择 Open EC2 AMIs View (打开 EC2 AMI 视图)。
2. 配置 AMI 视图以显示我们将在此示例中使用的 AMI。在筛选器框中，键入 **start ebs**。这会筛选 AMI 列表，使其仅显示名称中同时包含“start”和“ebs”的 AMI。

右键单击 amazon/getting-started-with-ecs AMI，然后从上下文菜单中选择 Launch (启动)。

3. 在 Launch EC2 Instance (启动 EC2 实例) 对话框中，为应用程序配置 AMI。

主机数

将此值设置为要启动的 EC2 实例的数量。

实例类型

选择要启动的 EC2 实例的类型。

可用区

选择启动实例的可用区 (AZ)。请注意，并非所有 AZ 在所有区域中都可用。如果您选择的 AZ 不可用，Toolkit 将生成一条消息，提示您需要选择其他 AZ。有关 AZ 的更多信息，请转到[区域和可用区常见问题](#)中的适用于 Linux 实例的 Amazon EC2 用户指南。

密钥对

密钥对是一组用于在您使用 SSH 连接到 EC2 实例时进行验证的公有/私有加密密钥。选择您有权访问其私钥的密钥对。

安全组

安全组控制 EC2 实例将接受的网络流量的类型。应该选择将允许端口 22 (SSH 使用的端口) 上的传入流量的安全组，以便能连接到 EC2 实例。有关如何使用 Toolkit 创建安全组的信息，请参阅[从管理安全组Amazon探险者 \(p. 22\)](#)

实例配置文件

实例配置文件是 IAM 角色的逻辑容器。当您选择实例配置文件时，会将对应的 IAM 角色与 EC2 实例关联。IAM 角色是使用指定对特定访问权限的策略配置的。Amazon 服务和帐户资源。当 EC2 实例与 IAM 角色关联时，在实例上运行的应用程序软件将使用 IAM 角色指定的权限运行。这使得应用程序软件无需指定任何 Amazon 它自己的凭证，从而使软件更安全。有关 IAM 角色的更多信息，请转到[使用角色](#)中的 IAM 用户指南。

用户数据

用户数据是您为在 EC2 实例上运行的应用程序软件提供的的数据。应用程序软件可以通过 [Instance Meta Data Service \(IMDS\)](#) 访问此数据。

4. 单击 Finish (完成)。
5. 在 Amazon 资源管理器，在 Amazon EC2 节点中，右键单击实例子节点然后选择打开 EC2 实例视图。

您的 EC2 实例应该显示在 EC2 Instances (EC2 实例) 视图中。实例转换为 running (正在运行) 状态可能需要几分钟的时间。实例开始运行后，您可以右键单击该实例以弹出上下文菜单，其中显示了可对实例执行的操作。例如，您可以从该菜单中终止实例。您还可以复制实例的公有 DNS 地址。您需要使用此 IP 地址通过 SSH 连接到该实例。

从管理安全组Amazon探险者

通过 Toolkit for Eclipse，您可以创建和配置安全组以与 Amazon Elastic Compute Cloud (Amazon EC2) 实例配合使用。启动 Amazon EC2 实例时，您需要指定一个关联的安全组。

安全组的行为类似于传入网络流量的防火墙。该安全组指定 Amazon EC2 实例将允许接受的网络流量的类型。它还可指定仅接受来自特定 IP 地址或其他指定安全组的传入流量。

创建新安全组

在此部分中，我们将创建一个新安全组。创建安全组后，安全组不配置有任何权限。权限配置是通过其他操作来处理的。

创建新安全组

1. 在Amazon探险者中，单击右侧的 Amazon EC2 节点，然后单击打开 EC2 安全组视图。
2. 右键单击 EC2 Security Groups (EC2 安全组) 选项卡的左侧窗格，然后单击 New Group (新建组)。
3. 在此对话框中，为新安全组输入名称和说明。单击 OK (确定)。

向安全组添加权限

在此部分中，我们将为新安全组添加权限，以允许其他计算机使用安全 Shell (SSH) 协议连接到我们的 Amazon EC2 实例。

向安全组添加权限

1. 右键单击 EC2 Security Groups (EC2 安全组) 选项卡的右侧窗格，然后单击 Add Permissions (添加权限)。
2. 在此对话框中，选择 Protocol, port and network (协议、端口和网络)。单击 TCP 来自的协议下拉菜单。Enter 22 为端口范围。端口 22 是 SSH 的标准端口。Network Mask (网络掩码) 框使用 CIDR 格式指定允许的源 IP 地址，默认值为 0.0.0.0/0，该值指定安全组允许从任何外部 IP 地址建立到端口 22 (SSH) 的 TCP 连接。

例如，您还可以指定该连接应该仅允许在您本地计算机的子网中的计算机建立连接。在这种情况下，您应该指定后跟“/10”的本地计算机的 IP 地址。例如，“xxx.xxx.xxx.xxx/10”，其中“xxx”对应于组成本地计算机的 IP 地址的不同八位字节值。

单击 OK (确定)。

您还可通过指定用户 ID 和安全组名称，在安全组中设置权限。在此情况下，此安全组中的 Amazon EC2 实例将接受来自指定安全组中的 Amazon EC2 实例的所有传入网络流量。您还必须指定 UserID 作为区分安全组名称的方法；安全组名称在所有 Amazon 中。有关安全组的更多信息，请参见 [Amazon EC2 中的安全性](#) 中的适用于 Linux 实例的 Amazon EC2 用户指南。

查看和添加 Amazon SNS 通知

您可以使 Toolkit for Eclipse 查看与该应用程序相关联的 Amazon Simple Notification Service (Amazon SNS) 主题。Amazon SNS 是一项服务，可让您的应用程序在发生指定的事件时，使用电子邮件等协议发送通知。如需了解有关 Amazon SNS 的更多信息，请参见 [Amazon SNS 开发人员指南](#)。

查看 Amazon SNS 通知

以下过程说明了如何查看 Amazon SNS 通知。

查看通知

1. 在 Amazon Explorer 中，单击左侧的三角形 Amazon SNS 节点以展开并查看其中包含的 Amazon SNS 主题。
2. 双击此 SNS 主题以在 Eclipse 编辑器窗格中打开详细信息视图。在此示例中，订阅 ARN 列显示主题正在等待确认。Amazon SNS 需要由通过电子邮件地址指定的个人进行确认，然后 SNS 才能向该个人发送电子邮件通知。

添加 Amazon SNS 通知

您可以通过以下方式添加新 Amazon SNS 通知 Amazon Explorer。

添加新通知

1. 在Amazon探险者中，右键单击Amazon SNS，然后单击创建新主题。输入新主题的名称，然后单击 OK (确定)。
2. 双击新主题以显示主题的详细信息视图。在 Subscriptions (订阅) 区域中右键单击，然后单击 Create Subscription (创建订阅)。保留 Subscription Protocol (订阅协议) 框中的 Email (plain text) (电子邮件 (纯文本))，然后输入终端节点的电子邮件地址。单击 OK (确定)。此订阅中现在包括通知的详细信息视图。
3. 要删除订阅，可在订阅的 Protocol (协议) 列中右键单击条目，然后单击 Delete Subscription (删除订阅)。

Note

创建订阅将导致向在订阅的“终端节点”电子邮件地址中指定的个人发送验证电子邮件。此电子邮件地址将由Amazon只是为了发送通知。它不会被用于任何其他目的Amazon或 Amazon.com 的询问。

连接到 Amazon Relational Database Service (Amazon RDS)

在本部分中，我们将使用 Toolkit for Eclipse 连接到 Amazon Relational Database Service (Amazon RDS) 上的数据库实例。逐步执行下述过程之前，您需要一个与您的Amazonaccount。您可以使用Amazon Web Services Management Console。创建数据库实例时，请将数据库用来接收连接的 TCP 端口设置为可从您所在位置访问的值。例如，如果您启用了防火墙，请选择防火墙允许的 TCP 端口来建立连接。有关更多信息，请参阅 [Amazon RDS 用户指南](#)。

1. 在Amazon探险者中，在下拉菜单中，展开Amazon RDS节点。您应该可以看到与您的关联的数据库实例的列表。Amazonaccount。右键单击其中一个实例，然后单击 Connect (连接)。
2. Toolkit for Eclipse 会显示身份验证对话框。输入您在创建数据库实例时指定的主密码。单击 Finish (完成)。
3. Eclipse Toolkit 会在 Eclipse Data Source Explorer 中打开与数据库实例的连接。在这里，您可以检查数据库的结构和数据。

Identity and Access Management

Amazon Identity and Access Management (IAM)，您可以控制哪些人能访问Amazon资源以及他们能使用它们执行哪些操作。这些区域有：Amazon Explorer，您可以创建和管理 IAM 用户、组和角色。您也可以为用户设置一个密码策略，该策略可让您指定密码要求 (如最小长度)，并可让您指定是否允许用户更改自己的密码。

Note

这是最佳实践所有用户，甚至是账户所有者访问Amazon资源作为 IAM 用户。这可确保在某个 IAM 用户的凭证泄露的情况下，无需更改账户的根凭证即可撤销受影响的凭证。

关于 Amazon Identity and Access Management

您可以创建 IAM 用户 (每个用户均可有自己的密码和安全凭证)，而不是共享账户的密码和安全凭证 (访问密钥 ID 和秘密访问密钥)。随后，您可以向用户附加策略，在策略中指定权限以确定用户能执行哪些操作以及用户能访问哪些资源。

为方便起见，您可以创建 IAM 组 (例如 Admins 和 Developers)，向其附加策略，然后将用户添加到这些组，而不是向单个用户附加策略。您还可以创建具有带权限的策略的角色。角色可由其他账户中的用户、服务以及没有 IAM 身份的用户代入。有关 IAM 的更多信息，请参阅 [IAM 用户指南](#)。

创建 IAM 用户

您可以创建 IAM 用户，以便组织中的其他人可以有自己的 IAM 用户 Amazon 身份。您可以将 IAM 策略附加到用户或将用户分配到组，由此向 IAM 用户分配权限。分配到某个组的 IAM 用户将从附加到该组的策略派生其权限。有关更多信息，请参阅[创建 IAM 组 \(p. 25\)](#)和[将 IAM 用户添加到 IAM 组 \(p. 25\)](#)。

使用 Toolkit，您还可以生成 Amazon IAM 用户的凭证 (访问密钥 ID 和秘密访问密钥)。有关更多信息，请参阅[管理 IAM 用户的凭证 \(p. 26\)](#)。

创建 IAM 用户

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击用户节点，然后选择创建新用户。
2. 在 Create New Users (创建新用户) 对话框中，为新 IAM 用户输入最多 5 个名称，然后单击 Finish (完成)。有关 IAM 用户名称的约束信息，请参阅[IAM 实体的限制](#)中的 IAM 用户指南。

有关向组添加用户的信息，请参阅[将 IAM 用户添加到 IAM 组 \(p. 25\)](#)。有关如何创建策略并将其附加到用户的信息，请参阅[将 IAM 策略附加到用户、组或角色 \(p. 28\)](#)。

创建 IAM 组

您可以将 IAM 用户添加到组，以便更轻松地管理权限。附加到组的任何权限均适用于该组中的任何用户。有关 IAM 组的更多信息，请参阅[使用用户和组](#)中的 IAM 用户指南。

当您创建一个组时，您可以创建一个策略，其中包含该组的成员将具有的权限。

创建 IAM 组

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击 Groups 节点，然后选择创建新组。
2. 输入新 IAM 组的名称，然后单击 Next (下一步)。
3. 输入策略的名称，该策略确立组成员可执行的操作。以 JSON 文档的形式输入策略，然后单击 OK (确定)。

该策略名称在您的账户内必须是唯一的。您为策略输入的 JSON 必须确认有效，否则您将无法保存策略。有关如何创建策略的信息，请参阅[策略概述](#)中的 IAM 用户指南。

4. 单击 Finish (完成)。

有关将其他策略附加到 IAM 组的信息，请参阅[将 IAM 策略附加到用户、组或角色 \(p. 28\)](#)。

将 IAM 用户添加到 IAM 组

如果将一个 IAM 用户添加到组，则已附加到组的任何策略也对该用户生效。有关 IAM 用户的更多信息，请参阅[用户和群组](#)中的 IAM 用户指南。

将 IAM 用户添加到 IAM 组

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击 Groups 节点，然后选择打开组编辑器。请注意，您将从中将 IAM 用户添加到 IAM 组。Groups 在中的节点 Amazon 探险者而不是来自用户节点。
2. 在 Groups (组) 编辑器中，选择要将用户添加到的组，然后单击 Users (用户) 选项卡。
3. 在底部窗格的右侧，单击 Add Users (添加用户) 按钮。

4. 在 Add Users to Group (将用户添加到组) 对话框中，选择要添加的用户，然后单击 OK (确定)。

管理 IAM 用户的凭证

对于每个用户，您可以添加一个密码。IAM 用户使用密码 Amazon 中的资源 Amazon Web Services Management Console。

为 IAM 用户创建密码

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击用户节点，然后选择打开用户编辑器。
2. 在用户列表中，选择要为其创建密码的用户，然后单击 Summary (摘要) 选项卡。
3. 在底部窗格的右侧，单击 Update Password (更新密码) 按钮。
4. 在 Update User Password (更新用户密码) 对话框中，输入密码，然后单击 OK (确定)。

Note

新密码将覆盖任何现有密码。

您还可以为每个用户生成一组访问密钥 (访问密钥 ID 和秘密访问密钥)。这些密钥可以用来表示用户以编程方式访问 Amazon— 例如，要使用 Amazon 使用 SDK 签署编程请求，或访问命令行界面 (CLI)，或访问 Amazon 通过工具包提供服务。(有关如何指定与 Toolkit 配合使用的凭证的信息，请参阅 [设置 Amazon 凭证 \(p. 3\)](#)。)

为 IAM 用户生成访问密钥

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击用户节点，然后选择打开用户编辑器。
2. 在用户列表中，选择要为其生成密钥的用户，然后单击 Summary (摘要) 选项卡。
3. 单击 Manage Access Keys (管理访问密钥) 按钮。

将显示一个窗口，您可以在其中管理用户的访问密钥。

4. 单击 Create Access Key (创建访问密钥) 按钮。

将显示 Manage Access Key (管理访问密钥) 对话框。

5. 单击 Download (下载) 按钮，下载包含已生成的凭证的逗号分隔值 (CSV) 文件。

Note

这将是您查看和下载这些访问密钥的唯一机会。如果您丢失了这些密钥，则必须将其删除并创建一组新的访问密钥。

您只能为每个 IAM 用户生成两组凭证。如果您已经有两组凭证并需要再创建一组凭证，则必须先删除现有的凭证组之一。

您还可以停用凭证。在这种情况下，凭证仍然存在，但任何请求 Amazon 使用这些凭证制作的将失败。如果您希望暂时禁用访问权限，这样做会很有用。Amazon 对于那组凭证。您可以重新激活之前已停用的凭证。

删除、停用或重新激活 IAM 用户的访问密钥

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击用户节点，然后选择打开用户编辑器。
2. 在用户列表中，选择要为其管理访问密钥的用户，单击 Summary (摘要) 选项卡，然后单击 Manage Access Keys (管理访问密钥) 按钮。
3. 在列出了该用户的访问密钥的窗口中，右键单击要管理的凭证，然后选择下列选项之一：

- Delete Access Key (删除访问密钥)
- Make Inactive (转为非活动)
- Make Active (转为活动)

创建 IAM 角色

使用 Amazon Toolkit，您可以创建 IAM 角色。然后该角色可以是假设按您希望允许其访问的实体 Amazon 资源的费用。附加到角色的策略将确定谁能够担任角色（受信任实体 或 委托人），以及这些实体可执行哪些操作。

在 Toolkit 中，您可以指定以下可信实体：

- 网络 ACL 和安全组都允许（因此可到达您的实例）的发起 ping 的 Amazon 服务。例如，您可以指定 Amazon EC2 可调用其他 EC2。Amazon 服务或 Amazon Data Pipeline 允许管理 Amazon EC2 实例。这称作服务角色。
- 您拥有的其他账户。如果您有多个 Amazon 账户，您可能需要允许某个账户中的用户使用角色来获取对您的另一个账户中的资源的访问权限。
- 一个第三方账户。您可以让第三方供应商管理 Amazon 资源的费用。在这种情况下，您可以创建一个角色，其中可信实体为第三方的 Amazon account。

在您指定谁是可信实体之后，可以指定一个策略来确定允许角色执行的操作。

例如，您可以创建一个角色并将只允许访问您的其中一个 Amazon S3 存储桶的策略附加到该角色。随后，您可以将角色与 Amazon EC2 实例关联。当应用程序在 Amazon EC2 实例上运行时，该应用程序只能访问角色策略允许您访问的 Amazon S3 存储桶。

有关 IAM 角色的更多信息，请参阅《IAM 用户指南》中的 [IAM 角色](#)。

创建 IAM 角色

1. 在 Amazon 探险者中，展开 Amazon Identity and Access Management 节点中，右键单击角色节点，然后选择创建新角色。
2. 输入 IAM 角色的名称，然后单击 Next (下一步)。
3. 选择角色的可信实体。要创建服务角色，请选择 Amazon 服务角色随后，从下拉列表中选择服务角色。

为在其他用户中定义的用户提供访问权限 Amazon 选择您拥有的账户 ID 然后输入 Amazon 另一个账户的账号。

要为第三方账户提供访问权限，请选择账户 ID 然后输入第三方的 Amazon 账号。如果第三方还提供了外部 ID，请同时输入此 ID。

4. 单击 Next (下一步)。
5. 输入策略的名称，该策略确立角色可执行的操作。随后，以 JSON 文档的形式输入策略，然后单击 OK (确定)。

该策略名称在您的账户内必须是唯一的。您为策略输入的 JSON 必须确认有效，否则您将无法保存策略。有关如何创建策略的信息，请参阅 [使用 IAM 指南](#) 中的策略概述。

6. 单击 Finish (完成)。

新的 IAM 角色将显示在 Roles (角色) 编辑器中。

有关展示如何访问的示例 Amazon 使用与 Amazon EC2 实例关联的 IAM 角色，请参阅 [使用 IAM 角色授予对资源的访问权](#) Amazon Amazon EC2 上的资源 中的 Amazon SDK for Java 开发人员指南。

将 IAM 策略附加到用户、组或角色

策略是定义权限的文档。例如，附加到用户的策略可指定用户的策略Amazon允许用户调用的操作以及用户可对其执行操作的资源。如果将策略附加到一个组，则权限适用于该组中的用户。如果将策略附加到一个角色，则权限应用于代入该角色的人员。

将策略附加到用户或组的过程类似。对于角色，您可以附加一个指定角色可执行的操作的策略。您使用单独的过程来附加或编辑用于确定可代入角色(即，管理信任关系)的人员的策略。

Note

如果您之前已将一个策略附加到用户、组或角色，则可以使用此过程来附加额外的策略。要编辑用户、组或角色的现有策略，请使用 IAM 控制台、命令行工具或 API 调用。

为用户、组或角色创建 IAM 策略

1. InAmazon探险者在中，展开AmazonIdentity and Access Management节点，然后双击Groups节点，用户节点，或者角色节点。
2. 选择要将策略附加到的组、用户或角色，然后单击 Permissions (权限) 选项卡。
3. 在底部窗格的右侧，单击 Attach Policy (附加策略) 按钮。
4. 在 Manage Group Policy (管理组策略)、Manage User Policy (管理用户策略) 或 Manage Role Permissions (管理角色权限) 对话框中，输入策略的名称。随后，以 JSON 文档的形式输入策略，然后单击 OK (确定)。

该策略名称在您的账户内必须是唯一的。您为策略输入的 JSON 必须确认有效，否则您将无法保存策略。有关如何创建策略的信息，请参阅[IAM 策略概述](#)中的IAM 用户指南。

为角色创建或管理信任关系

1. InAmazon探险者在中，展开AmazonIdentity and Access Management节点，然后双击角色节点。
2. 在 Roles (角色) 编辑器中，选择要管理的角色，然后单击 Trust Relationships (信任关系) 选项卡。
3. 在底部窗格的右侧，单击 Edit Trust Relationship (编辑信任关系) 按钮。
4. 在 Edit Trust Relationship (编辑信任关系) 对话框中，编辑 JSON 策略文档，然后单击 OK (确定)。

设置密码策略

在 Toolkit for Eclipse 中，您可以为您的账户设置密码策略。这使您能够确保为 IAM 用户创建的密码遵循特定的长度和复杂性准则。这还可让您指定是否允许用户更改其密码。有关更多信息，请参阅 [管理 IAM 密码策略](#) 中的IAM 用户指南。

为用户或组创建 IAM 策略

1. InAmazon探险者在下，Identity and Access Management在中，双击密码策略节点。
2. 在密码策略窗格中，为您的所需策略选项指定所需的策略选项Amazon账号，然后单击应用密码策略。

调试无服务器应用程序AmazonSAM Local

本教程将指导您完成通过 Eclipse 工具包调试无服务器应用程序项目的过程中的过程。AmazonSAM Local。SAM LocalAmazonCLI 工具，用于管理使用Amazon无服务器应用程序模型 (AmazonSAM)。有关更多信息，请参阅 SAM Local [自述文件](#)。

Prerequisites

要使用本教程，您必须拥有 Eclipse、Docker 和 Toolkit for Eclipse。Amazon 已安装 SAM Local。请参阅 Amazon SAM Local [自述文件](#) 有关 Docker 和 SAM Local 安装说明。请参阅 [开始使用 \(p. 2\)](#) 有关安装和设置 Toolkit for Eclipse 说明的主题。

Note

使用 Amazon Toolkit for Eclipse 的 SAM Local 功能，您的项目必须是有效 Maven 项目并包含有效的 pom.xml 文件。

安装所需工具后，请打开 Eclipse Preferences (首选项) 中的对话框日食菜单。配置 SAM Local Executable (SAM Local 可执行文件) 路径，如下所示。这样一来，Toolkit for Eclipse 将知道在何处查找您的 SAM Local 安装。

从 AWS CodeStar 应用程序导入 SAM 应用程序

在本教程中，您在 AWS CodeStar 中需要一个示例项目。请参阅 [在中创建无服务器项目 Amazon CodeStar](#) 中的教程 AWS CodeStar 用户指南创建示例项目。

从 AWS CodeStar 导入 SAM 应用程序

1. 在 Eclipse 工具栏上，打开 Amazon Web Services 菜单 (由 Amazon 主页图标)，然后选择导入 Amazon CodeStar 项目。或者，在 Eclipse 菜单栏上，选择 File (文件)、导入、Amazon、Amazon CodeStar 项目。
2. 选择示例应用程序的创建区域。
3. 从 Project Name (项目名称) 列表中选择示例项目。
4. 添加您的 Git 凭证。请参阅 [Amazon CodeCommit 用户指南](#) 以了解如何为 CodeCommit 获取 Git 凭证。
5. 选择 Next (下一步)。
6. 选择 Branch Selection 页面上的 Next。
7. 选择 Local Destination 页面上的 Finish。

接下来，您可以在 Eclipse 内使用 SAM Local 本地调试此无服务器应用程序。

本地调试 Lambda 函数

为您的无服务器应用程序创建调试配置并使用 SAM Local 本地运行此应用程序。

本地调试 Lambda 函数

1. 在 Eclipse 的 Project Explorer (项目浏览器) 中，打开 HelloWorldHandler.java。
2. 在 Eclipse 代码窗口中右键单击，选择调试方式，然后选择 Amazon SAM Local。
3. 在此示例中，保留 Project (项目) 和 Template (模板) 不变。
4. 选择 Run as 字段中的 Lambda Function。
5. 选择 Function identifier 字段中的 GetHelloWorld。
6. 在此示例中，我们将提供 Amazon S3 事件。选择 Event 输入框旁的 Generate。
7. 选择您的 Amazon S3 存储桶所在的区域。
8. 输入有效的 Amazon S3 存储桶名称。
9. 输入有效的 Amazon S3 对象键，然后选择确定。
10. 在 Save As (另存为) 页面上，选择当前项目并输入事件文件名称。在此示例中，我们使用的是 s3-event.json。

11. 选择 OK (确定), 保存事件文件并返回主对话框。
12. 保持高级设置不变。要详细了解这些字段, 请参阅[高级设置 \(p. 30\)](#)。
13. 选择 Apply, 然后选择 Debug。

这将在本地运行 Lambda 函数。您可以像对其他应用程序一样设置断点来调试代码。

在本地测试 API Gateway

您还可以使用 SAM Local 测试 HTTP 请求/响应功能。

本地测试 API 网关

1. 在 Eclipse 代码窗口中右键单击, 选择 Debug As (调试方式)、Debug Configuration (调试配置)。
2. 为此运行创建新的调试配置并将其命名为其他名称。
3. 选择 Run as 字段中的 API Gateway。
4. 保持所有其他字段不变, 您的配置看上去应类似于下面这样。
5. 选择 Apply, 然后选择 Debug。

这将生成一个可用于测试您的应用程序的本地 API 网关。调试输出将包含可用于验证代码的请求/响应功能的 HTTP 链接。

高级设置

此节描述了 SAM Local 调试配置页面上可用的高级选项。

Amazon 配置

选择配置文件

(必需) 要用于的配置文件 Amazon 凭证。

(默认值) 默认配置文件

选择区域

(必需) 应用程序的部署区域。

(默认值) 美国东部 (弗吉尼亚)

SAM Local 配置

Maven 目标

(必需) 在构建应用程序时要执行的 Maven 目标。如果默认情况下不会生成包含所有依赖项的 Jar 文件 (fat Jar), 则您必须自定义这些目标。要了解如何使用插件创建 fat Jar, 请参阅 Maven 项目中的 [Maven Shade 插件](#)。

(默认值) 清理程序包

SAM 运行时

(必需) 指向 SAM 可执行文件的路径。

(默认值) /usr/local/bin/sam

调试端口

(必需) Eclipse 调试程序用于连接到 SAM Local 的端口。

(默认值) 5858

Env vars

(可选) JSON 文件的路径，该文件包含 Lambda 函数所使用环境变量的值。要了解此文件必需的语法，请参阅 SAM Local 用户指南中的[环境变量文件](#)。

(默认值) 空

Lambda 函数配置

代码 URI

(可选) 指向代码存档文件的路径。对于此页面上的示例，这将是指向 .jar 文件的路径。

(默认值) template.yml 文件中的路径

Timeout

(必需) Lambda 函数运行时超时值。

(默认值) 300

更多信息

有关 的更多信息 AmazonSAM 本地，请参阅[AmazonSAM Local GitHub](#) 中的用户指南。有关的更多信息 Amazon 无服务器应用程序模型 (SAM)，请参阅[AmazonSAM GitHub](#) 中的项目。

故障排除

AmazonCodeCommit 插件-Eclipse 无法写入到安全存储。

问题：签出或签入时AmazonCodeCommit 存储库，我有一个错误说写入安全存储失败，未提供密码。

解决方案：打开Preferences (首选项) ->普通的->安全->安全存储->目录->GIT->Delete.

文档历史记录

下表介绍了自上次发布本指南以来的重要更改。

API 版本：2010-12-01

文档上次更新时间：2020 年 4 月 28 日

2016 年 12 月 1 日

新增一节，提供有关新的[无服务器项目向导 \(p. 7\)](#)的详细信息。

2015 年 12 月 22 日

其他资源 主题已删除 - 此页面的信息现在从指南的第一个页面上提供，位于标题[其他文档和资源 \(p. 1\)](#)下。

2015 年 10 月 22 日

- 本指南已从“入门指南”重命名为“用户指南”，以更好地呈现其功能。
- 安装说明已更新，用于补充您选择工具包的组件进行安装的方式改变。

2014 年 6 月 16 日

Toolkit for Eclipse 现在支持创作Amazon Lambda使用 Java 函数。有关更多信息，请参阅 [结合使用 Lambda 与AmazonToolkit for Eclipse \(p. 10\)](#)。

2013 年 9 月 27 日

- Toolkit for Eclipse 现在使用相同的系统来存储和访问Amazon将凭证作为AmazonCLI 和Amazon开发工具包，其中包括使用多个配置文件来存储多个凭证集的功能。有关信息，请参阅新更新的主题：[设置Amazon凭证 \(p. 3\)](#)。
- Eclipse 入门指南的工具包已根据其他AmazonSDK 文档（最值得注意的是，AmazonToolkit for Eclipse 依赖的 Java SDK）。大部分结构重组应该是符合逻辑且不证自明的，不过，中提供了指南各个重要部分的说明。[什么是AmazonToolkit for Eclipse？ \(p. 1\)](#)。
- 对 Eclipse 4.3 (“Kepler”) 的[入门 \(p. 2\)](#)进行了更新。

2013 年 9 月 9 日

本主题跟踪近期对指南的更改。旨在与[发行说明历史记录](#)一起提供。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。