

用户指南

AWS Cloud9



AWS Cloud9: 用户指南

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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、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拥有者的财产，这些拥有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 AWS Cloud9 ?	1
AWS Cloud9 是如何运行的?	1
AWS Cloud9 环境	1
环境和计算资源	2
AWS Cloud9 可以用来做什么?	2
怎样入门?	2
其他主题	3
它可以用来做什么?	3
附加信息	5
相关视频	5
AWS 网站上的相关主题	5
定价	5
我有其他疑问或需要帮助	6
设置 AWS Cloud9	7
单个用户设置	7
注册获取 AWS 账户	8
创建具有管理访问权限的用户	8
身份验证的其他方法	9
后续步骤	10
团队设置	11
注册获取 AWS 账户	8
创建具有管理访问权限的用户	8
步骤 2 : 创建一个 IAM 组 and 用户并在组中添加该用户	14
步骤 3 : 向群组添加 AWS Cloud9 访问权限	19
第 4 步 : 登录 AWS Cloud9 控制台	22
后续步骤	23
企业设置	23
步骤 1 : 为企业创建管理账户	26
步骤 2 : 为管理账户创建企业	26
步骤 3 : 将成员账户添加到企业	27
步骤 4 : 在企业中启用 IAM Identity Center	28
步骤 5 : 在企业中设置组和用户	28
步骤 6 : 在企业中允许组和用户使用 AWS Cloud9	29
步骤 7 : 开始使用 AWS Cloud9	31

后续步骤	32
其他设置选项 (团队和企业)	32
步骤 1 : 创建客户托管式策略	33
步骤 2 : 向组添加客户托管式策略	34
使用 AWS Cloud9 的团队的客户托管式策略示例	35
后续步骤	40
入门 : 基本教程	41
欢迎使用 AWS Cloud9 (控制台)	41
先决条件	41
步骤	41
步骤 1 : 创建环境	42
步骤 2 : 基本导览	46
第 3 步 : 清除	51
相关信息	53
Hello AWS Cloud9 (CLI)	54
先决条件	55
步骤	55
步骤 1 : 创建环境	55
步骤 2 : 基本导览	57
第 3 步 : 清除	62
相关信息	63
使用环境	66
创建环境	66
创建 EC2 环境	67
创建 SSH 环境	82
使用 Systems Manager 访问非入口 EC2 实例	87
在 EC2 环境中使用 Systems Manager 的益处	88
管理 Systems Manager 权限	90
授予用户访问由会话管理器管理的实例的权限	91
使用 AWS CloudFormation 创建非入口 EC2 环境	94
为 Amazon S3 配置 VPC 终端节点以下载依赖项	96
为私有连接配置 VPC 终端节点	100
打开环境	100
从环境中调用 AWS 服务	102
创建和使用实例配置文件管理临时凭证	103
在环境中创建和存储永久访问凭证	109

更改环境设置	112
更改环境首选项	113
使用控制台更改环境设置	113
使用代码更改环境设置	115
使用共享环境	116
共享环境使用案例	116
关于环境成员访问角色	117
邀请与环境位于相同账户中的用户	119
让与环境位于相同账户中的 AWS Cloud9 管理员邀请他们自己或其他人	121
打开共享环境	122
查看环境成员列表	123
打开环境成员的活动文件	124
打开环境成员的打开文件	124
转到环境成员的活动光标	125
与其他环境成员聊天	125
在共享的环境中查看聊天消息	125
从共享的环境中删除聊天消息	125
从共享的环境中删除所有聊天消息	126
更改环境成员的访问角色	126
从共享的环境中删除您的用户	127
删除另一个环境成员	129
环境共享最佳实践	130
移动环境并对 Amazon EBS 卷调整大小/加密	130
移动环境	131
AWS Cloud9 将 EC2 环境移至不同的亚马逊系统映像 (AMI)	133
调整环境使用的 Amazon EBS 卷大小	138
加密使用的 Amazon EBS 卷 AWS Cloud9	140
删除环境	143
使用控制台删除环境	143
使用代码删除环境	146
使用 IDE	148
了解 IDE	149
先决条件	150
步骤 1：菜单栏	150
第 2 步：控制面板	152
步骤 3：“Environment (环境)”窗口	152

步骤 4：编辑器、选项卡和窗格	153
第 5 步：控制台	155
步骤 6：打开文件部分	156
第 7 步：间距	156
步骤 8：状态栏	157
步骤 9：“Outline (大纲)”窗口	158
步骤 10：“Go (转到)”窗口	159
步骤 11：“Immediate (立即运行)”选项卡	161
步骤 12：进程列表	162
第 13 步：首选项	163
第 14 步：终端	164
步骤 15：“Debugger (调试程序)”窗口	165
最终思考	171
语言支持	171
AWS Cloud9 IDE 中支持的编程语言版本	173
增强语言支持	174
增强版 Java 支持	174
增强 TypeScript 支持	181
菜单命令参考	185
AWS Cloud9 菜单	185
文件菜单	186
Edit Menu (编辑菜单)	187
Find Menu (查找菜单)	191
View Menu (查看菜单)	192
Go Menu (转到菜单)	193
Run Menu (运行菜单)	194
工具菜单	195
Window Menu (窗口菜单)	196
Support Menu (支持菜单)	198
Preview Menu (预览菜单)	199
Other Menu Bar Commands (其他菜单栏命令)	199
查找和替换文本	200
查找单个文件中的文本	200
替换单个文件中的文本	200
查找多个文件中的文本	201
替换多个文件中的文本	202

查找和替换选项	203
预览文件	204
打开要预览的文件	204
重新加载文件预览	206
更改文件预览类型	206
在单独的 Web 浏览器标签页中打开文件预览	206
切换到其他文件预览	206
预览运行的应用程序	206
运行应用程序	207
预览运行的应用程序	209
重新加载应用程序预览	210
更改应用程序预览类型	210
在单独的 Web 浏览器标签页中打开应用程序预览	211
切换到不同的预览 URL	211
通过互联网共享运行的应用程序	211
使用文件修订	216
使用图像文件	218
查看或编辑图像	218
调整图像大小	219
裁剪图像	219
转动图像	220
翻转图像	220
缩放图像	220
平滑图像	220
使用生成器、运行程序和调试程序	220
内置的生成、运行和调试支持	221
生成您的项目文件	221
运行代码	222
调试代码	222
更改内置运行程序	223
创建运行配置	224
创建生成器或运行程序	224
定义生成器或运行程序	225
使用自定义环境变量	229
设置命令级别的自定义环境变量	229
在 <code>~/.bash_profile</code> 中设置自定义的用户环境变量	230

设置本地自定义环境变量	230
在 ~/.bashrc 中设置自定义的用户环境变量	230
在 ENV List 中设置自定义环境变量	230
使用项目设置	231
查看或更改项目设置	232
将一个环境的当前项目设置应用于另一个环境	232
您可以更改的项目设置	232
手动停止环境的 EC2 实例	239
使用用户设置	240
查看或更改用户设置	240
与其他用户共享您的用户设置	241
您可以执行的用户设置更改	241
使用 AWS 项目和用户设置	250
项目级别设置	250
用户级别设置	251
使用键绑定	251
查看或更改键绑定	251
与其他用户共享您的键绑定	252
更改您的键盘模式	252
更改您的操作系统键绑定	252
更改特定的键绑定	253
删除您的所有自定义键绑定	254
使用主题	254
查看或更改您的主题	255
您可以更改的整体主题设置	255
主题覆盖	255
管理初始化脚本	256
打开初始化脚本	256
MacOS 默认键绑定参考	256
常规	257
选项卡	260
面板	262
代码编辑器	263
emmet	269
终端	270
运行和调试	270

MacOS Vim 键绑定参考	271
常规	272
选项卡	274
面板	276
代码编辑器	277
emmet	284
终端	285
运行和调试	285
MacOS Emacs 键绑定参考	285
常规	286
选项卡	289
面板	291
代码编辑器	292
emmet	298
终端	299
运行和调试	299
MacOS Sublime 键绑定参考	300
常规	301
选项卡	305
面板	307
代码编辑器	307
emmet	314
终端	315
运行和调试	315
Windows/Linux 默认键绑定参考	316
常规	316
选项卡	319
面板	321
代码编辑器	322
emmet	329
终端	329
运行和调试	330
Windows/Linux Vim 键绑定参考	330
常规	331
选项卡	333
面板	336

代码编辑器	336
emmet	343
终端	343
运行和调试	344
Windows/Linux Emacs 键绑定参考	344
常规	345
选项卡	347
面板	350
代码编辑器	350
emmet	357
终端	357
运行和调试	358
Windows / Linux Sublime 键绑定参考	358
常规	359
选项卡	363
面板	365
代码编辑器	365
emmet	372
终端	372
运行和调试	373
命令参考	373
使用其他AWS服务	375
使用 Amazon Lightsail 实例	375
步骤 1：创建基于 Linux 的 Lightsail 实例	376
步骤 2：设置实例以将其与 AWS Cloud9 一起使用	379
步骤 3：创建并连接到 AWS Cloud9 SSH 开发环境	381
步骤 4：使用 AWS Cloud9 IDE 更改实例上的代码	383
使用 AWS CodeStar 项目	384
第 1 步：准备好使用 AWS CodeStar 项目	385
第 2 步：在 AWS CodeStar 中创建项目	385
步骤 3：创建 AWS Cloud9 开发环境并将其连接到项目	385
使用 Amazon Q	386
什么是 Amazon Q？	386
为 Amazon Q 启用 IAM 权限	386
使用 AWS CodePipeline	387
步骤 1：创建或确定源代码存储库	388

步骤 2：创建 AWS Cloud9 开发环境，将其连接到代码存储库并上传代码	388
步骤 3：准备使用 AWS CodePipeline	389
步骤 4：在 AWS CodePipeline 中创建管道	389
与 CodeCatalyst	390
入门 CodeCatalyst	391
在 Amazon 中复制 AWS Cloud9 代码资源 CodeCatalyst	391
使用复制工具	403
有关复制过程的常见问题	406
中的开发环境 CodeCatalyst	408
使用 AWS CDK	412
AWS CDK 应用程序	412
使用 Git 面板进行可视化源代码控制	415
使用 Git 面板管理源代码控制	416
初始化或克隆 Git 存储库	418
暂存和提交文件	421
查看不同的文件版本	423
使用分支	424
使用远程存储库	427
存储和检索文件	429
参考：Git 面板中可用的 Git 命令	429
Git 面板菜单中可用的 Git 命令的参考	431
Git 面板搜索字段中可用的 Git 命令	433
AWS Toolkit	436
为什么使用 AWS Toolkit？	436
启用 AWS Toolkit	438
管理 AWS Toolkit 的访问凭证	438
使用 IAM 角色向在 EC2 实例上运行的应用程序授予权限	440
识别 AWS Toolkit 组件	440
禁用 AWS Toolkit	441
AWS Toolkit 主题	441
导航和配置	442
使用 AWS Explorer 处理多个区域中的服务和资源	442
访问和使用 AWS Toolkit 菜单	443
使用“AWS Configuration (Amazon 配置)”窗格修改 AWS Toolkit 设置	445
API Gateway	447
调用 REST API	447

AWS App Runner	448
先决条件	449
定价	451
创建 App Runner 服务	452
管理 App Runner 服务	454
AWS CloudFormation 堆栈	456
删除 AWS CloudFormation 堆栈	457
Amazon CloudWatch Logs	457
查看 CloudWatch 日志组和日志流	457
使用 CloudWatch 日志事件	458
AWS Lambda 函数	460
调用远程 Lambda 函数	460
下载、上传和删除 Lambda 函数	461
资源	466
用于访问资源的 IAM 权限	466
与现有资源交互	467
Amazon S3	467
使用 Amazon S3 存储桶	468
使用 Amazon S3 对象	470
AWS 无服务器应用程序	472
创建无服务器应用程序	472
运行和调试无服务器应用程序	474
同步无服务器应用程序	481
启用 AWS Toolkit 代码镜头	482
删除无服务器应用程序	482
调试无服务器应用程序的配置选项	483
AWS Step Functions	486
先决条件	486
创建和发布状态机	486
在 AWS 工具包中运行状态机	488
下载状态机定义文件并可视化其 workflow	489
AWS Systems Manager	489
假设和先决条件	490
Systems Manager 自动化文档的 IAM 权限	490
创建新的 Systems Manager 自动化文档	491
发布 Systems Manager 自动化文档	491

编辑现有 Systems Manager 自动化文档	492
使用版本	493
删除 Systems Manager 自动化文档	493
运行 Systems Manager 自动化文档	494
故障排除	494
Amazon ECR	494
先决条件	495
将 Amazon ECR 与 IDE 配合使用 AWS Cloud9	495
AWS IoT	504
AWS IoT先决条件	504
AWS IoT 事物	504
AWS IoT 证书	506
AWS IoT 策略	509
Amazon ECS	511
Amazon ECS Exec	512
Amazon EventBridge	514
使用 Amazon EventBridge Schemas	514
AWS Cloud9 的教程	517
AWS CLI 和 aws-shell 教程	517
先决条件	518
步骤 1：在您的环境中安装 AWS CLI 和/或 aws-shell	518
步骤 2：在环境中设置凭证管理	520
步骤 3：在环境中使用 AWS CLI 或 aws-shell 运行基本命令	520
步骤 4：清除	521
AWS CodeCommit 教程	522
先决条件	522
步骤 1：设置具有所需访问权限的 IAM 组	522
步骤 2：在 CodeCommit 中创建存储库	524
步骤 3：将您的环境连接到远程存储库	525
步骤 4：将远程存储库克隆到您的环境	526
步骤 5：向存储库添加文件	527
步骤 6：清除	529
Amazon DynamoDB 教程	529
先决条件	530
步骤 1：在您的环境中安装和配置 AWS CLI 和/或 AWS CloudShell	530
步骤 2：创建表	531

步骤 3：向表添加项目	532
步骤 4：向表添加多个项目	533
步骤 5：创建全局二级索引	537
步骤 6：获取表中的项目	540
步骤 7：清除	544
AWS CDK 教程	544
先决条件	545
步骤 1：安装所需工具	545
步骤 2：添加代码	548
步骤 3：运行代码	551
步骤 4：清除	553
LAMP 教程	553
先决条件	554
步骤 1：安装工具	554
步骤 2：设置 MySQL	556
步骤 3：设置网站	558
步骤 4：清除	561
WordPress 教程	563
先决条件	563
安装概览	564
步骤 1：安装和配置 MariaDB Server	564
步骤 2：安装和配置 WordPress	565
步骤 3：配置 Apache HTTP Server	566
步骤 4：预览 WordPress web 内容	567
管理混合内容错误	567
Java 教程	568
先决条件	568
步骤 1：安装所需工具	569
步骤 2：添加代码	570
步骤 3：生成并运行代码	571
步骤 4：设置以使用 AWS SDK for Java	571
步骤 5：在环境中设置 AWS 凭证管理	578
步骤 6：添加 AWS 开发工具包代码	578
步骤 7：构建并运行 AWS 开发工具包代码	580
步骤 8：清除	581
C++ 教程	581

先决条件	581
步骤 1：安装 g++ 和所需的开发软件包	582
步骤 2：安装 CMake	583
步骤 3：获取并构建 SDK for C++	583
步骤 4：创建 C++ 和 CMakeLists 文件	584
步骤 5：构建并运行 C++ 代码	588
步骤 6：清除	589
Python 教程	589
先决条件	590
步骤 1：安装 Python	590
步骤 2：添加代码	591
步骤 3：运行代码	591
步骤 4：安装和配置 AWS SDK for Python (Boto3)	592
步骤 5：添加 AWS 开发工具包代码	593
步骤 6：运行 AWS 开发工具包代码	595
步骤 7：清除	596
.NET 教程	596
先决条件	596
步骤 1：安装所需工具	597
步骤 2（可选）：为 Lambda 函数安装 .NET CLI 扩展	599
步骤 3：创建 .NET 控制台应用程序项目	599
步骤 4：添加代码	599
步骤 5：生成并运行代码	600
步骤 6：创建并设置使用 AWS SDK for .NET 的 .NET 控制台应用程序项目	602
步骤 7：添加 AWS 开发工具包代码	603
步骤 8：构建并运行 AWS 开发工具包代码	605
步骤 9：清除	606
Node.js 教程	606
先决条件	607
步骤 1：安装所需工具	607
步骤 2：添加代码	609
步骤 3：运行代码	609
第 4 步：在 Node.js JavaScript 中安装和配置 AWS 开发工具包	610
第 5 步：添加 AWS SDK 代码	612
第 6 步：运行 S AWS DK 代码	615
步骤 7：清除	616

PHP 教程	616
先决条件	616
步骤 1：安装所需工具	617
步骤 2：添加代码	618
步骤 3：运行代码	619
步骤 4：安装和配置 AWS SDK for PHP	619
第 5 步：添加 AWS SDK 代码	621
第 6 步：运行 S AWS DK 代码	622
步骤 7：清除	623
Ruby	624
Go 教程	624
先决条件	624
步骤 1：安装所需工具	625
步骤 2：添加代码	626
步骤 3：运行代码	626
步骤 4：安装和配置 AWS SDK for Go	628
步骤 5：添加 AWS 开发工具包代码	629
步骤 6：运行 AWS 开发工具包代码	632
步骤 7：清除	633
TypeScript 教程	633
先决条件	633
步骤 1：安装所需工具	634
步骤 2：添加代码	635
步骤 3：运行代码	636
步骤 4：安装并配置 AWS SDK for JavaScript in Node.js	637
步骤 5：添加 AWS 开发工具包代码	638
步骤 6：运行 AWS 开发工具包代码	640
步骤 7：清除	641
Docker 教程	641
先决条件	642
步骤 1：安装并运行 Docker	642
步骤 2：构建镜像	643
步骤 3：运行容器	646
步骤 4：创建环境	648
步骤 5：运行代码	652
步骤 6：清理	653

相关教程	654
AWS Cloud9 的高级主题	655
EC2 环境与 SSH 环境的比较	655
Amazon VPC 设置	656
亚马逊 VPC 对以下内容的要求 AWS Cloud9	657
创建 VPC 以及其他 VPC 资源	669
仅创建 VPC	670
为创建子网 AWS Cloud9	671
将子网配置为公有子网或私有子网	673
SSH 环境主机要求	674
何时以及如何创建 SSH 环境	675
SSH 托管要求	676
AWS Cloud9 安装程序	678
下载并运行 AWS Cloud9 安装程序	678
AWS Cloud9 安装程序故障排除	679
入站 SSH IP 地址范围	681
未在 ip-ranges.json 中的 IP 地址	682
AMI 内容	682
Amazon Linux 2023/Amazon Linux 2	683
Ubuntu Server	684
服务相关角色	685
AWS Cloud9 的服务相关角色权限	686
为 AWS Cloud9 创建服务相关角色	690
为 AWS Cloud9 编辑服务相关角色	690
删除 AWS Cloud9 的服务相关角色	690
AWS Cloud9 服务相关角色的受支持区域	690
使用 CloudTrail 记录 API 调用	690
CloudTrail 中的 AWS Cloud9 信息	691
了解 AWS Cloud9 日志文件条目	692
Tags	708
将标签更新传播到底层资源	709
的安全性 AWS Cloud9	711
数据保护	711
数据加密	712
Identity and Access Management	714
受众	715

使用身份进行身份验证	715
使用策略管理访问	718
如何 AWS Cloud9 与 IAM 配合使用	720
基于身份的策略示例	726
故障排除	728
如何 AWS Cloud9 使用 IAM 资源和操作	730
AWS 托管策略	733
为以下各项创建客户托管策略 AWS Cloud9	743
AWS Cloud9 权限参考	757
AWS 托管的临时证书	763
日记账记录和监控	768
使用监控活动 CloudTrail	768
监控 EC2 环境性能	768
合规性验证	768
韧性	773
基础设施安全性	773
软件更新和修补	774
安全最佳实操	774
故障排除 AWS Cloud9	775
Installer (安装程序)	775
AWS Cloud9 安装程序挂起或失败	775
AWS Cloud9 显示 “Package Cloud9 IDE 1” 后，安装程序无法完成	775
无法安装依赖项	776
SSH 环境错误：“需要 Python 版本 3 才能安装 pty.js”	777
AWS Cloud9 环境	777
环境创建错误：“我们无法创建 EC2 实例.....”	777
环境创建错误：“未授权执行 sts:AssumeRole”	777
联合身份无法创建环境	778
控制台错误：“用户无权对资源执行操作”	778
无法连接到环境	779
无法打开环境	779
无法打开 AWS Cloud9 环境：“协作者目前无法访问此环境。Please wait until the removal of managed temporary credentials is complete, or contact the owner of this environment.” (协作者当前无法访问此环境。请等待托管式临时凭证删除完成，或联系此环境的拥有者。)	781
环境删除错误：“One or more environments failed to delete (一个或多个环境删除失败) ” ...	781
在 AWS Cloud9 IDE 中更改环境的超时时间	782

由于 AWS Cloud9 环境没有足够的磁盘空间，因此在 AWS Toolkit 中本地运行 SAM 应用程序时出错	783
无法使用早期版本的 Microsoft Edge 浏览器加载 IDE	783
无法在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment。	784
无法在 IDE 的文件资源管理器中为其创建子文件夹结构 /projects/ projects。 AWS Cloud9 CodeCatalyst	784
由于 tmux 会话错误，无法与 AWS Cloud9 中的终端窗口交互	784
Amazon EC2	786
Amazon EC2 实例未自动更新	786
AWS CLI 或者 AWS-shell 错误：在 EC2 环境中“请求中包含的安全令牌无效”	786
无法连接到 EC2 环境，因为 VPC 的 IP 地址被 Docker 使用	787
无法在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment。	784
当 AWS License Manager 许可证配置与 Amazon EC2 实例关联时，无法 AWS Cloud9 从控制台启动	788
无法在 EC2 环境中运行某些命令或脚本	788
使用创建 EC2 环境时报告错误消息“账户中 AWSCloud9SSMInstanceProfile 不存在实例配置文件” AWS CloudFormation	789
使用 AWS CloudFormation 创建 EC2 环境时，报告错误消息“未授权对资源执行 perform: ssm:StartSession”	789
使用 AWS CLI 创建 EC2 环境时，报告错误消息“no authorization to perform: on resource: instance profile”（未授权对资源执行 iam:GetInstanceProfile：实例配置文件 AWSCloud9SSMInstanceProfile）	789
将默认加密应用于 Amazon EBS 卷时，无法创建环境	790
EC2-Classic 账户的 VPC 错误“Unable to access your environment (无法访问您的环境)”	790
其他 AWS 服务	791
无法在 IDE 的文件资源管理器中为其创建子文件夹结构 /projects/ projects。 AWS Cloud9 CodeCatalyst	784
无法显示在 IDE 以外运行的应用程序	792
运行 AWS Toolkit 时出错：“您的环境已用完索引节点，请增加'fs.inotify.max_user_watches'限制。”	793
Lambda 本地函数运行错误：无法安装 SAM Local	794
AWS Control Tower 尝试使用以下方法创建 Amazon EC2 环境时出错 AWS Cloud9：“创建环境失败，错误为：以下挂钩失败：[:: GuardControlTower:: Hook]。”	794
将默认加密应用于 Amazon EBS 卷时，无法创建环境	790

当 AWS License Manager 许可证配置与 Amazon EC2 实例关联时，无法 AWS Cloud9 从控制台启动	788
应用程序预览	795
重新加载环境之后，必须刷新应用程序预览	795
应用程序预览或文件预览通知：“Third-party cookies disabled (第三方 cookie 已禁用)”	796
应用程序预览选项卡显示错误或空白	800
无法在 IDE 中预览 Web 内容，因为与站点的连接不安全	801
预览文件时返回 499 错误	801
Performance	802
AWS Cloud9 IDE 冻结了很长时间	802
控制台警告：“Switching to the minimal code completion engine... (切换到最小代码完成引擎.....)”	802
IDE 警告：“This environment is running low on memory (此环境内存不足)”或“This environment has high CPU load (此环境具有较高 CPU 负载)”	803
无法在 AWS Cloud9 IDE 中上传文件	804
AWS Cloud9 IDE 中的下载速度很慢	804
无法在 IDE 中预览 Web 内容，因为与站点的连接不安全	801
第三方应用程序和服务	805
由于 tmux 会话错误，无法与 AWS Cloud9 中的终端窗口交互	784
无法使用早期版本的 Microsoft Edge 浏览器加载 IDE	783
调试 C++ 项目时 gdb 出现错误	806
PHP 运行器出现问题 AWS Cloud9	807
与 Node.js 相关的 GLIBC 错误	808
支持的浏览器	809
限制	811
AWS Cloud9 限制	811
AWS Cloud9 IDE 下载限制	812
相关 AWS 服务限制	812
文档历史记录	813
.....	dcccxxiv

什么是 AWS Cloud9 ？

AWS Cloud9 是一个集成开发环境或 IDE。

AWS Cloud9 IDE 提供丰富的代码编辑体验，对多种编程语言和运行时调试程序的支持以及内置终端。它包含一套工具，可用于对软件进行编码、构建、运行、测试和调试，并帮助您将软件发布到云中。

您可以通过 Web 浏览器访问 AWS Cloud9 IDE。您可以根据自己的偏好配置 IDE。您可以切换颜色主题、绑定快捷键、启用编程语言特定的语法颜色和代码格式设置等。

(明白了！我已准备好试用 AWS Cloud9。 [怎样入门？](#))

AWS Cloud9 是如何运行的？

下图大致地概述了 AWS Cloud9 的工作原理。

从图中（从底部开始），您可以使用 AWS Cloud9 IDE，同时在本地电脑上的 Web 浏览器中运行，以与 AWS Cloud9 环境进行交互。计算资源（例如，Amazon EC2 实例或您自己的服务器）连接到该环境。最后，您的工作存储在 AWS CodeCommit 存储库或另一种类型的远程存储库中。



AWS Cloud9 环境

AWS Cloud9 环境是指您在其中存储自己的项目文件并运行工具来开发应用程序的地方。

使用 AWS Cloud9 IDE，您可以执行以下操作：

- 将您的项目文件存储在本地实例或服务器中。
- 将远程代码存储库（如 AWS CodeCommit 中的存储库）克隆到您的环境中。
- 在环境中使用本地文件和克隆文件的组合。

您可以多个环境之间创建和切换，并为每个特定开发项目设置一个环境。通过在云中存储环境，您不再是只能通过单个电脑或服务器设置来处理项目。这使您能够轻松地执行各种操作，如在计算机之间切换，而且你能更快地招募开发人员来组建团队。

环境和计算资源

在后台，有两种方法可以将您的环境连接到计算资源：

- 您可以指示 AWS Cloud9 创建一个 Amazon EC2 实例，然后将环境连接到这个新创建的 EC2 实例。这种类型的设置被称为 EC2 环境。
- 您可以指示 AWS Cloud9 将环境连接到现有云计算实例或您自己的服务器。这种类型的设置被称为 SSH 环境。

EC2 环境和 SSH 环境既有相似之处，也存在差异。如果您是 AWS Cloud9 新用户，我们建议您使用 EC2 环境，因为 AWS Cloud9 能为您完成许多配置。随着您了解有关 AWS Cloud9 的更多信息，您可能需要更好地了解这些相似和不同之处，请参阅 [AWS Cloud9 中 EC2 环境与 SSH 环境的比较](#)。

有关 AWS Cloud9 如何工作的更多信息，请参阅这些相关[视频](#)和[网页](#)。

AWS Cloud9 可以用来做什么？

借助 AWS Cloud9，您可以在许多激动人心的场景和变体中进行编码、构建、运行、测试、调试和发布软件。包括（但不限于）：

- 使用几种编程语言编写代码和 AWS Cloud Development Kit (AWS CDK)。
- 处理正在运行的 Docker 容器中的代码。
- 使用在线代码存储库。
- 实时与他人协作。
- 与各种数据库和网站技术交互。
- 定向 AWS Lambda、Amazon API Gateway 和 AWS 无服务器应用程序。
- 利用其他 AWS 产品，如 Amazon Lightsail、AWS CodeStar 和 AWS CodePipeline。

若要获取更详细的列表，请参阅 [AWS Cloud9 可以用来做什么？](#)

怎样入门？

若要开始使用 AWS Cloud9，请按照 [设置 AWS Cloud9](#) 中的步骤操作，然后浏览[基本教程](#)。

其他主题

- [AWS Cloud9 可以用来做什么？](#)
- [有关 AWS Cloud9 的其他信息](#)

AWS Cloud9 可以用来做什么？

浏览以下资源，了解使用 AWS Cloud9 的一些常见场景。

主要场景

场景	资源
使用 AWS Toolkit 在 AWS Lambda 函数和无服务器应用程序中创建、运行和调试代码。	通过 AWS Toolkit 使用 AWS Lambda 函数
使用已预配置 WordPress、LAMP (Linux、Apache、MySQL 和 PHP)、Node.js、Nginx、Drupal 和 Joomla 等常用应用程序和框架，以及 Amazon Linux、Ubuntu、Debian、FreeBSD 和 openSUSE 等 Linux 发行版的 Amazon Lightsail 实例。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例
使用 AWS CodeStar 中的 AWS 软件开发项目和工具链中的代码。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CodeStar 项目
使用 AWS CodePipeline 中持续交付解决方案中的代码。	在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline
使用 AWS CLI 和 AWS CloudShell 自动执行 AWS 服务。	适用于 AWS Cloud9 的 AWS Command Line Interface 和 aws-shell 教程
使用 AWS CodeCommit 中的源代码存储库。	适用于 AWS Cloud9 的 AWS CodeCommit 教程
使用带 Git 面板界面的 GitHub 中的源代码存储库。	使用 Git 面板进行可视化源代码控制

场景	资源
使用 Amazon DynamoDB 中的 NoSQL 数据库。	适用于 AWS Cloud9 的 Amazon DynamoDB 教程
使用 LAMP (Linux、Apache HTTP Server、MySQL 和 PHP) 堆栈。	适用于 AWS Cloud9 的 LAMP 教程
使用 WordPress 网站。	适用于 AWS Cloud9 的 WordPress 教程
使用 Java 代码以及 AWS SDK for Java。	适用于 AWS Cloud9 的 Java 教程
使用 C++ 代码以及 AWS SDK for C++。	适用于 AWS Cloud9 的 C++ 教程
使用 Python 代码以及 AWS SDK for Python (Boto)。	适用于 AWS Cloud9 的 Python 教程
使用 .NET Core 代码以及 AWS SDK for .NET。	适用于 AWS Cloud9 的 .NET 教程
使用 Node.js 代码以及 AWS SDK for JavaScript。	Node.js 的教程 AWS Cloud9
使用 PHP 代码以及 AWS SDK for PHP。	的 PHP 教程 AWS Cloud9
使用 Ruby 代码以及 AWS SDK for Ruby。	AWS Cloud9 中的 Ruby
使用 Go 代码以及 AWS SDK for Go。	适用于 AWS Cloud9 的 Go 教程
使用 TypeScript 代码以及 AWS SDK for JavaScript。	适用于 AWS Cloud9 的 TypeScript 教程
使用 AWS Cloud Development Kit (AWS CDK) 代码。	适用于 AWS Cloud9 的 AWS CDK 教程
处理运行的 Docker 容器中的代码。	适用于 Docker 的教程 AWS Cloud9
邀请其他用户与您一起实时使用环境并提供文本聊天支持。	在 AWS Cloud9 中使用共享环境
在 AWS RoboMaker 中使用用于智能机器人应用程序的代码。	AWS RoboMaker 开发人员指南中的 使用 AWS Cloud9 进行开发

有关 AWS Cloud9 的其他信息

本主题提供了详细信息，以帮助您了解 AWS Cloud9。

主题

- [相关视频](#)
- [AWS 网站上的相关主题](#)
- [定价](#)
- [我有其他疑问或需要帮助](#)

相关视频

- [AWS re:Invent 2017 - AWS Cloud9 简介：Werner Vogels 主题演讲](#) (9 分钟，YouTube 网站)
- [AWS re:Invent Launchpad 2017 - AWS Cloud9](#) (15 分钟，YouTube 网站)
- [AWS Cloud9 简介 - AWS 在线技术研讨会](#) (33 分钟，YouTube 网站)
- [AWS 2018 年悉尼峰会：AWS Cloud9 和 AWS CodeStar](#) (25 分钟，YouTube 网站)

AWS 网站上的相关主题

- [AWS Cloud9 简介](#)
- [AWS Cloud9 – 云开发人员环境](#)
- [AWS Cloud9 概述](#)
- [AWS Cloud9 功能](#)
- [AWS Cloud9 常见问题](#)

定价

AWS Cloud9 无需额外付费。如果您将 Amazon EC2 实例用于 AWS Cloud9 开发环境，则只需为用于运行和存储代码的计算和存储资源（例如，Amazon EC2 实例、Amazon EBS 卷）付费。您可以通过 SSH 将您的环境连接到现有 Linux 服务器（例如本地服务器）而无需额外付费。

您只需按您的实际用量付费；既没有最低费用，也无需预付费。您需要以正常的 AWS 费率为您在环境中创建或使用的任何 AWS 资源（例如 AWS Lambda 函数）付费。

有资格使用 AWS 免费套餐的 AWS 新客户可免费使用 AWS Cloud9。如果您的环境使用 AWS 免费套餐之外的资源，您需要以正常的 AWS 费率为这些资源付费。

有关更多信息，请参阅下列内容。

- AWS Cloud9 定价，请参阅 [AWS Cloud9 定价](#)。
- AWS 服务定价：请参阅 [Amazon EC2 定价](#)、[Amazon EBS 定价](#)、[AWS Lambda 定价](#) 和 [AWS 定价](#)。
- AWS 免费套餐：请参阅 AWS Billing and Cost Management 用户指南中的 [使用 AWS 免费套餐和跟踪您的免费套餐使用情况](#)。
- 教育定价：请参阅 [AWS 教育](#) 计划。

我有其他疑问或需要帮助

要在 AWS Cloud9 社区中提问或寻求帮助，请参阅 [AWS Cloud9 论坛](#)。（当您进入此论坛时，AWS 可能会要求您登录。）

另请参阅我们的 [常见问题 \(FAQ\)](#)，或者 [直接与我们联系](#)。

设置 AWS Cloud9

要开始使用 AWS Cloud9，请根据您的计划如何使用 AWS Cloud9，按照下列过程之一执行。

使用模式	执行以下过程
我是使用自己的 AWS 账户的唯一用户，并且不是学生。	单个用户设置
我属于在单个 AWS 账户中具有多个用户的团队。	团队设置
我属于在单个企业中拥有一个或多个 AWS 账户的企业。	企业设置

有关 AWS Cloud9 的一般信息，请参阅 [AWS Cloud9 是什么？](#)。

主题

- [个人用户设置 AWS Cloud9](#)
- [团队设置 AWS Cloud9](#)
- [适用于 AWS Cloud9 的企业设置](#)
- [适用于 AWS Cloud9 的其他设置选项（团队和企业）](#)

个人用户设置 AWS Cloud9

本主题介绍当你不是学生 AWS 账户时，如何设置和使用 AWS Cloud9 作为你的唯一用户。您可以 AWS Cloud9 为任何其他使用模式进行设置。有关说明，请参阅[设置 AWS Cloud9](#)。

要 AWS Cloud9 作为您的唯一用户使用 AWS 账户，请注册一个（AWS 账户如果您还没有）。接下来，登录 AWS Cloud9 控制台。

主题

- [注册获取 AWS 账户](#)
- [创建具有管理访问权限的用户](#)
- [身份验证的其他方法](#)

- [后续步骤](#)

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

要注册 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建 AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行 [需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就可以不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。 [AWS Management Console](#) 在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 [以根用户身份登录](#)。

2. 为您的根用户启用多重身份验证 (MFA)。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备 \(控制台\)](#)。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[启用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《[用户指南](#)》[IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限](#)。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[创建权限集](#)。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[添加组](#)。

身份验证的其他方法

Warning

为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证，而是使用与身份提供商的联合身份验证，例如 [AWS IAM Identity Center](#)。

管理跨区域的访问权限 AWS 账户

作为安全最佳实践，我们建议 AWS Organizations 与 IAM Identity Center 配合使用来管理所有人的访问权限 AWS 账户。有关更多信息，请参阅《[IAM 用户指南](#)》中的 IAM 安全最佳实践。

您可以在 IAM Identity Center 中创建用户，使用 Microsoft Active Directory，使用 SAML 2.0 身份提供商 (IdP)，或者将你的 IdP 单独联合到其中。AWS 账户您可以使用其中一种方法，为用户提供单点登录体验。您还可以强制执行多重身份验证 (MFA) 并使用临时证书 AWS 账户 进行访问。这与 IAM 用户不同，后者是一种可以共享的长期凭证，并且可能会增加 AWS 资源的安全风险。

仅为沙盒环境创建 IAM 用户

如果您不熟悉 AWS，可以创建一个测试 IAM 用户，然后使用它来运行教程并探索 AWS 所提供的內容。在学习时可以使用此类凭证，但我们建议您避免在沙盒环境之外使用。

对于以下用例，开始使用 IAM 用户可能是有意义的 AWS：

- 开始使用您的 AWS SDK 或工具，并在沙盒环境 AWS 服务 中进行探索。
- 在学习过程中，运行不支持人工参与登录流程的计划脚本、作业和其他自动化流程。

如果您在这些用例之外使用 IAM 用户，请尽快过渡到 IAM Identity Center 或将您的身份提供商联合到该 AWS 账户 中心。有关更多信息，请参阅 [AWS 中的身份联合验证](#)。

确保 IAM 用户访问密钥安全

您应该定期轮换 IAM 用户访问密钥。参阅《IAM 用户指南》，按照[轮换访问密钥](#)中的指导进行操作。如果您认为自己不小心共享了您的 IAM 用户访问密钥，请轮换您的访问密钥。

IAM 用户访问密钥应存储在本地计算机上的共享 AWS credentials 文件中。请勿将 IAM 用户访问密钥存储在您的代码中。请勿将包含 IAM 用户访问密钥的配置文件存储到任何源代码管理软件中。开源项目 [git-secrets](#) 等外部工具可以帮助您避免无意中将敏感信息提交到 Git 存储库。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 身份 \(用户、用户组和角色\)](#)。

后续步骤

学习任务	主题
学习如何使用 AWS Cloud9 IDE。	入门：基本教程 和 使用 IDE

更多高级任务	主题
创建 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 在新环境中处理代码。	创建环境
利用文本聊天支持实时邀请其他用户与您一起使用新环境。	使用共享环境

团队设置 AWS Cloud9

本主题介绍[AWS IAM Identity Center](#)如何使用在单个 AWS 账户 用户中允许多个用户使用 AWS Cloud9。要设置 AWS Cloud9 为用于任何其他使用模式，请参见[设置 AWS Cloud9](#)以获取正确的说明。

这些说明假定您具有或将具有对单个 AWS 账户的管理访问权限。有关更多信息，请参阅 IAM 用户指南中的[AWS 账户 根用户和创建您的第一个管理员和群组](#)。如果您已经拥有该账户 AWS 账户 但没有管理权限，请 AWS 账户 咨询您的管理员。

Warning

为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证，而是使用与身份提供商的联合身份验证，例如 [AWS IAM Identity Center](#)。

Note

您可以使用 [IAM Identity Center](#) 而不是 IAM，以允许单个 AWS 账户 身份中心中的多个用户使用 AWS Cloud9。在这种使用模式中，单一账户 AWS 账户 充当中组织的管理账户 AWS Organizations。此外，该组织没有成员账户。要使用 IAM Identity Center，请跳过此主题并改为按照[企业设置](#)中的说明进行操作。有关信息，请参阅以下资源：

- AWS Organizations 用户指南中的 [AWS Organizations 是什么](#) (需要使用 IAM 身份中心 AWS Organizations)
- AWS IAM Identity Center 用户指南中的 [什么是 AWS IAM Identity Center](#)
- 4 分钟视频 [AWS Knowledge Center 视频：如何开始使用 on AWS Organizations](#) YouTube
- 7 分钟的视频 [使用 IAM Identity Center 管理用户对多个 AWS 账户的访问权限](#) YouTube

- 9 分钟的视频 [如何为您的本地 Active Directory 用户设置 IAM 身份中心](#) YouTube

要使单个 AWS 账户用户可以开始使用 AWS Cloud9，请启动适用于您拥有的 AWS 资源的步骤。

你有 AWS 账号吗？	您是否在该账户中至少有一个 IAM 组 and 用户？	从这一步开始
否	—	第 1 步：注册 AWS 账户
是	不支持	步骤 2：创建一个 IAM 组 and 用户并在组中添加该用户
是	是	步骤 3：向群组添加 AWS Cloud9 访问权限

主题

- [注册获取 AWS 账户](#)
- [创建具有管理访问权限的用户](#)
- [步骤 2：创建一个 IAM 组 and 用户并在组中添加该用户](#)
- [步骤 3：向群组添加 AWS Cloud9 访问权限](#)
- [第 4 步：登录 AWS Cloud9 控制台](#)
- [后续步骤](#)

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

要注册 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行[需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就可以不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。[AWS Management Console](#)在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的[以根用户身份登录](#)。

2. 为您的根用户启用多重身份验证 (MFA)。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户 \(控制台\) 启用虚拟 MFA 设备](#)。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[启用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《[用户指南](#)》[IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限](#)。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[创建权限集](#)。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的[添加组](#)。

步骤 2：创建一个 IAM 组 and 用户并在组中添加该用户

在此步骤中，您将在 AWS Identity and Access Management (IAM) 中创建一个组和一个用户，将该用户添加到组，然后使用该用户进行访问 AWS Cloud9。这是一种 AWS 安全最佳实践。有关更多信息，请参阅《IAM 用户指南》中的[IAM 最佳实践](#)。

如果您已经拥有所需的所有 IAM 组和用户，请跳至[步骤 3：向组添加 AWS Cloud9 访问权限](#)。

Note

您的企业可能已为您设置一个 IAM 组 and 用户。如果您的组织有 AWS 账户 管理员，请在开始以下步骤之前咨询该管理员。

您可以使用 [AWS Management Console](#) 或 [AWS Command Line Interface \(AWS CLI\)](#) 完成这些任务。

要观看与以下控制台过程相关的 9 分钟视频，请参阅[如何设置 IAM 用户并 AWS Management Console 使用 IAM 证书登 YouTube 录](#)。

步骤 2.1：使用控制台创建 IAM 组

1. 如果你尚未登录 AWS Management Console，请[通过 https://console.aws.amazon.com/codecommit](https://console.aws.amazon.com/codecommit) 登录。

Note

您可以使用创建时提供的电子邮件地址和密码登录。AWS Management Console AWS 账户这称为以根用户身份登录。但是，这不是 AWS 安全最佳实践。将来，我们建议您使用 AWS 账户中管理员用户的凭据登录。管理员用户拥有与 AWS 账户 root 用户相似的 AWS 访问权限，可以避免一些相关的安全风险。如果您无法以管理员用户身份登录，请咨询您的 AWS 账户管理员。有关更多信息，请参阅《IAM 用户指南》中的[创建您的第一个 IAM 用户和组](#)。

2. 打开 IAM 控制台。为此，请在 AWS 导航栏中选择“服务”。然后，选择 IAM。
3. 在 IAM 控制台的导航窗格中，选择 Groups (组)。
4. 选择 Create New Group (创建新组)。
5. 在设置组名页的组名框中输入新组的名称。
6. 选择下一步。
7. 在附加策略页中，选择下一步而不附加任何策略。您将在[步骤 3：向群组添加 AWS Cloud9 访问权限](#)中附加策略。
8. 选择创建组。

Note

我们建议您重复此过程以创建至少两个组：一个 AWS Cloud9 用户组，另一个组用于 AWS Cloud9 管理员。此 AWS 安全最佳实践可以帮助您更好地控制、跟踪和解决 AWS 资源访问问题。


向前跳到[步骤 2.2：使用控制台创建 IAM 用户并在组中添加该用户](#)。

步骤 2.1：使用创建一个 IAM 群组 AWS CLI

Note

如果您使用的是[AWS 托管临时凭证](#)，则无法在 AWS Cloud9 IDE 中使用终端会话来运行本节中的部分或全部命令。为了解决 AWS 安全最佳实践，AWS 托管临时证书不允许运行某些命令。相反，您可以从单独安装的 AWS Command Line Interface (AWS CLI) 中运行这些命令。


1. 如果您尚未在计算机上安装和配置 AWS CLI，请执行此操作。为此，请参阅 [AWS Command Line Interface 用户指南](#) 中的以下内容：
 - [安装 AWS 命令行界面](#)
 - [快速配置](#)

 Note

您可以使用与 AWS CLI 创建时提供的电子邮件地址和密码关联的凭据进行配置。这称为以根用户身份登录。但是，这不是 AWS 安全最佳实践。相反，我们建议您为 AWS 账户中的 IAM 管理员用户配置 AWS CLI 使用证书。IAM 管理员用户拥有与 AWS 账户根用户相似的 AWS 访问权限，可以避免一些相关的安全风险。如果您无法将配置 AWS CLI 为 IAM 管理员用户，请咨询您的 AWS 账户管理员。有关更多信息，请参阅 IAM 用户指南中的 [创建您的第一个 IAM 管理员用户和组](#)。

2. 运行 `iam create-group` 命令，并指定新组的名称（如 `MyCloud9Group`）。

```
aws iam create-group --group-name MyCloud9Group
```

 Note

我们建议您重复此过程以创建至少两个组：一个 AWS Cloud9 用户组，另一个组用于 AWS Cloud9 管理员。此 AWS 安全最佳实践可以帮助您更好地控制、跟踪和解决 AWS 资源访问问题。

跳至 [步骤 2.2：创建 IAM 用户并使用 AWS CLI 将该用户添加到群组](#)。

步骤 2.2：使用控制台创建 IAM 用户并在组中添加该用户

1. 使用之前步骤打开 IAM 控制台，在导航窗格中选择 **Users**（用户）。
2. 选择添加用户。
3. 对于用户名，为新用户输入名称。

Note

您可以选择添加其他用户以同时创建多个用户。该过程中的其他设置适用于其中的每个新用户。

4. 选中 Programmatic access (编程访问) 和 AWS Management Console access (控制台访问) 复选框。这能让新用户使用各种 AWS 开发工具和服务控制台。
5. 保留默认选项自动生成密码。这会创建一个随机密码，供新用户登录控制台。或者，选择 Custom password (自定义密码)，然后为新用户输入特定的密码。
6. 保留默认选项需要重置密码。这会提示新用户首次登录到控制台后更改其密码。
7. 选择下一步：权限。
8. 保留将用户添加到组或向组添加多个用户 (对于多个用户) 的默认选择。
9. 在组列表中，选中要将用户添加到的组旁边的复选框 (不是名称)。
10. 选择 Next: Review (下一步: 审核)。
11. 选择 Create user。或者，对于多个用户，则 Create users (创建用户)。
12. 在向导的最后一页上，执行下列操作之一：
 - 在每个新用户旁边，选择发送电子邮件，并按照屏幕上的说明通过电子邮件向新用户发送其控制台登录 URL 和用户名。然后，分别向每位新用户传达他们的控制台登录密码、AWS 访问密钥 ID 和 AWS 私有访问密钥。
 - 选择下载 .csv。然后，向每位新用户传达其主机登录 URL、控制台登录密码、AWS 访问 AWS 密钥 ID 和已下载文件中的私有访问密钥。
 - 在每个新用户旁边，为 Secret access key (秘密访问密钥) 和 Password (密码) 选择 Show (显示)。然后向每位新用户传达他们的主机登录 URL、控制台登录密码、AWS 访问密钥 ID 和 AWS 私有访问密钥。

Note

如果您不选择 Download .csv，则这是您唯一一次可以查看新用户的 AWS 私有访问密钥和控制台登录密码。要为新用户生成新的 AWS 私有访问密钥或控制台登录密码，请参阅 IAM 用户指南中的以下内容。

- [创建、修改和查看访问密钥 \(控制台 \)](#)
- [创建、更改或删除 IAM 用户密码 \(控制台 \)](#)

13. 为希望创建的其他每个 IAM 用户重复此过程，然后向前跳至[步骤 3：向组添加 AWS Cloud9 访问权限](#)。

步骤 2.2：创建 IAM 用户并将该用户添加到群组中 AWS CLI

Note

如果您使用的是[AWS 托管临时凭证](#)，则无法在 AWS Cloud9 IDE 中使用终端会话来运行本节中的部分或全部命令。为了解决 AWS 安全最佳实践，AWS 托管临时证书不允许运行某些命令。相反，您可以从单独安装的 AWS Command Line Interface (AWS CLI) 中运行这些命令。

1. 运行 IAM `create-user` 命令以创建用户，并指定新用户的名称 (如 `MyCloud9User`)。

```
aws iam create-user --user-name MyCloud9User
```

2. 运行 IAM `create-login-profile` 命令，以便为用户创建新的控制台登录密码，并指定用户的名称和初始登录密码 (如 `MyC10ud9Us3r!`)。在用户登录后，AWS 会要求用户更改其登录密码。

```
aws iam create-login-profile --user-name MyCloud9User --password MyC10ud9Us3r! --password-reset-required
```

如果您需要稍后为用户生成替代控制台登录密码，请参阅 IAM 用户指南中的[创建、更改或删除 IAM 用户密码 \(API、CLI PowerShell\)](#)。

3. 运行 IAM `create-access-key` 命令为用户创建新的 AWS 访问 AWS 密钥和相应的私有访问密钥。

```
aws iam create-access-key --user-name MyCloud9User
```

记下显示的 `AccessKeyId` 和 `SecretAccessKey` 值。运行 IAM `create-access-key` 命令后，这是您唯一一次可以查看用户的 AWS 私有访问密钥。如果您需要稍后为用户生成新的 AWS 私有访问密钥，请参阅 IAM 用户指南中的[创建、修改和查看访问密钥 \(API、CLI PowerShell\)](#)。

4. 运行 IAM `add-user-to-group` 命令以在组中添加用户，并指定组和用户的名称。

```
aws iam add-user-to-group --group-name MyCloud9Group --user-name MyCloud9User
```

5. 向用户传达他们的主机登录 URL、初始控制台登录密码、AWS 访问密钥 ID 和 AWS 私有访问密钥。
6. 对您需要创建的每个其他 IAM 用户重复此过程。

步骤 3：向群组添加 AWS Cloud9 访问权限

默认情况下，大多数 IAM 群组和用户都无权访问任何群组和用户 AWS 服务 AWS Cloud9，包括（IAM 管理员群组和 IAM 管理员用户除外，它们在 AWS 账户默认情况下可以访问其 AWS 服务中的所有群组）。在此步骤中，您将使用 IAM 直接向一个或多个用户所属的 IAM 群组添加 AWS Cloud9 访问权限。这样，您可以确保这些用户可以访问 AWS Cloud9。

Note

您的组织可能已为您设置一个具有相应访问权限的组。如果您的组织有 AWS 账户管理员，请在开始以下步骤之前咨询该管理员。

您可以使用 [AWS Management Console](#) 或 [AWS CLI](#) 完成此任务。

使用控制台向群组添加 AWS Cloud9 访问权限

1. 如果你尚未登录 AWS Management Console，请[通过 https://console.aws.amazon.com/codecommit](https://console.aws.amazon.com/codecommit) 登录。

Note

您可以使用创建时提供的电子邮件地址和密码登录。AWS Management Console AWS 账户这称为以根用户身份登录。但是，这不是 AWS 安全最佳实践。以后，我们建议您使用 AWS 账户中的 IAM 管理员用户的凭证登录。管理员用户拥有与 AWS 账户 root 用户相似的 AWS 访问权限，可以避免一些相关的安全风险。如果您无法以管理员用户身份登录，请咨询您的 AWS 账户管理员。有关更多信息，请参阅 IAM 用户指南中的[创建您的第一个 IAM 管理员用户和组](#)。

2. 打开 IAM 控制台。为此，请在 AWS 导航栏中选择“服务”。然后，选择 IAM。
3. 选择 Groups（组）。
4. 选择组名。

5. 决定是要向群组添加 AWS Cloud9 用户访问权限还是 AWS Cloud9 管理员访问权限。这些权限应用于组中的每个用户。

AWS Cloud9 用户访问权限允许群组中的每位用户在其内部执行以下操作 AWS 账户：

- 创建自己的 AWS Cloud9 开发环境。
- 获取有关自己的环境的信息。
- 更改自己的环境的设置。

AWS Cloud9 管理员访问权限允许群组中的每位用户在其内部执行其他操作 AWS 账户：

- 为自己或其他人创建环境。
- 获取有关自己或其他人的环境的信息。
- 删除自己或其他人的环境。
- 更改自己或其他人的环境的设置。

Note

我们建议您仅将有限数量的用户添加到 AWS Cloud9 管理员组中。此 AWS 安全最佳实践可以帮助您更好地控制、跟踪和解决 AWS 资源访问问题。

6. 在 Permissions (权限) 选项卡中，在 Manged Policies (托管式策略) 处，选择 Attach Policy (附加策略)。
7. 在策略名称列表中，选中 AWS Cloud9 用户访问权限或 AWS Cloud9 管理员访问权限旁边AWSCloud9Administrator的复选框。AWSCloud9User如果在列表中看不到任一策略名称，请在 Filter (筛选条件) 框中输入策略名称以显示该策略。
8. 选择附加策略。

Note

如果您要向多个群组添加 AWS Cloud9 访问权限，请对每个群组重复此步骤。

要查看这些 AWS 托管策略授予群组的访问权限列表，请参阅[AWS 托管 \(预定义 \) 策略](#)。

要了解除了所需的 AWS 访问权限之外还可以添加到群组的访问权限 AWS Cloud9，请参阅 IAM 用户指南中的[托管策略和内联策略](#)以及[了解策略授予的权限](#)。

向前跳至 [步骤 4：登录到 AWS Cloud9 控制台](#)。

使用向群组添加 AWS Cloud9 访问权限 AWS CLI

Note

如果您使用的是[AWS 托管临时凭证](#)，则无法在 AWS Cloud9 IDE 中使用终端会话来运行本节中的部分或全部命令。为了解决 AWS 安全最佳实践，AWS 托管临时证书不允许运行某些命令。相反，您可以从单独安装的 AWS Command Line Interface (AWS CLI) 中运行这些命令。

1. 如果您尚未在计算机上安装和配置 AWS CLI，请执行此操作。为此，请参阅 AWS Command Line Interface 用户指南中的以下内容：

- [安装 AWS 命令行界面](#)
- [快速配置](#)

Note

您可以使用与 AWS CLI 创建时提供的电子邮件地址和密码关联的凭据进行配置。这称为以根用户身份登录。但是，这不是 AWS 安全最佳实践。相反，我们建议您在 IAM 管理员用户配置 AWS CLI 使用证书 AWS 账户。IAM 管理员用户拥有与 AWS 账户根用户相似的 AWS 访问权限，可以避免一些相关的安全风险。如果您无法将配置 AWS CLI 为管理员用户，请咨询您的 AWS 账户管理员。有关更多信息，请参阅[IAM 用户指南](#)中的创建您的第一个 IAM 管理员用户和组。

2. 决定是向群组添加 AWS Cloud9 用户访问权限还是 AWS Cloud9 管理员访问权限。这些权限应用于组中的每个用户。

AWS Cloud9 用户访问权限允许群组中的每位用户在其内部执行以下操作 AWS 账户：

- 创建自己的 AWS Cloud9 开发环境。
- 获取有关自己的环境的信息。
- 更改自己的环境的设置。

AWS Cloud9 管理员访问权限允许群组中的每位用户在其内部执行其他操作 AWS 账户：

- 为自己或其他人创建环境。
- 获取有关自己或其他人的环境的信息。
- 删除自己或其他人的环境。

- ~~更改自己或其他人的环境的设置。~~

Note

我们建议您仅将有限数量的用户添加到 AWS Cloud9 管理员组中。此 AWS 安全最佳实践可以帮助您更好地控制、跟踪和解决 AWS 资源访问问题。

3. 运行 IAM `attach-group-policy` 命令，指定群组名称和要添加的 AWS Cloud9 访问权限策略的 Amazon 资源名称 (ARN)。

对于 AWS Cloud9 用户访问权限，请指定以下 ARN。

```
aws iam attach-group-policy --group-name MyCloud9Group --policy-arn
arn:aws:iam::aws:policy/AWSCloud9User
```

要获得 AWS Cloud9 管理员访问权限，请指定以下 ARN。

```
aws iam attach-group-policy --group-name MyCloud9Group --policy-arn
arn:aws:iam::aws:policy/AWSCloud9Administrator
```

Note

如果您要向多个群组添加 AWS Cloud9 访问权限，请对每个群组重复此步骤。

要查看这些 AWS 托管策略授予群组的访问权限列表，请参阅[AWS 托管（预定义）策略](#)。

要了解除了所需的 AWS 访问权限之外还可以添加到群组的访问权限 AWS Cloud9，请参阅 IAM 用户指南中的[托管策略和内联策略](#)以及[了解策略授予的权限](#)。

第 4 步：登录 AWS Cloud9 控制台

完成本主题中前面的步骤后，您和您的用户就可以登录 AWS Cloud9 控制台了。

1. 如果您已 AWS 账户以 root 用户 AWS Management Console 身份登录，请退出控制台。
2. 打开 AWS Cloud9 控制台，[网址为 https://console.aws.amazon.com/cloud9/](https://console.aws.amazon.com/cloud9/)。
3. 输入您之前创建或识别的 IAM 用户的 AWS 账户号码，然后选择下一步。

Note

如果您看不到用于输入 AWS 账号的选项，请选择“登录其他帐户”。在下一页上输入 AWS 帐户号码，然后选择 Next (下一步)。

4. 输入您以前创建或确定的 IAM 用户的登录凭证，然后选择 Sign In (登录)。
5. 如果出现提示，请按照屏幕上的说明更改您用户的初始登录密码。将您的新登录密码保存在安全的位置。

AWS Cloud9 控制台随即显示出来，您可以开始使用 AWS Cloud9 了。

后续步骤

Task	请参阅本主题
限制您的其他人 AWS Cloud9 使用 AWS 帐户，以控制成本。	其他设置选项
创建 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 在新环境中处理代码。	创建环境
学习如何使用 AWS Cloud9 IDE。	入门：基本教程 和 使用 IDE
利用文本聊天支持实时邀请其他用户与您一起使用新环境。	使用共享环境

适用于 AWS Cloud9 的企业设置

本主题介绍如何使用 [AWS IAM Identity Center](#) 以允许一个或多个 AWS 帐户 在企业中使用 AWS Cloud9。要进行设置以将 AWS Cloud9 用于任何其他使用模式，请参阅 [设置 AWS Cloud9](#) 获取正确说明。

Warning

为了避免安全风险，在开发专用软件或处理真实数据时，请勿使用 IAM 用户进行身份验证。而是使用与身份提供商的联合身份验证，例如 [AWS IAM Identity Center](#)。

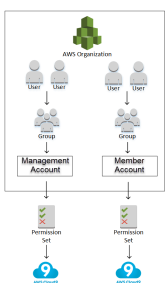
这些说明假定您具有或将有 AWS Organizations 中组织的管理访问权限。如果您在 AWS Organizations 中还没有对于组织的管理访问权限，请联系 AWS 账户 管理员。有关更多信息，请参阅以下资源：

- 《AWS Organizations 用户指南》中的[管理适用于 AWS Organizations 的访问权限](#) (IAM Identity Center 需要使用 AWS Organizations)
- 《AWS IAM Identity Center 用户指南》中的[对 IAM Identity Center 资源的访问权限管理的概览](#)
- [使用 AWS Control Tower](#)，这是一项可用于设置和管理 AWS 多账户环境的服务。AWS Control Tower 利用其他 AWS 服务 (包括 AWS Organizations、AWS Service Catalog 和 AWS IAM Identity Center) 的功能，在不到一小时的时间内建立登录区。

有关与此主题相关的介绍性信息，请参阅以下资源：

- 《AWS Organizations 用户指南》中的[什么是 AWS Organizations](#) (IAM Identity Center 需要使用 AWS Organizations)
- AWS IAM Identity Center 用户指南中的[什么是 AWS IAM Identity Center](#)
- 《AWS Control Tower 用户指南》中的[AWS Control Tower 入门](#)
- YouTube 上的 4 分钟视频 [AWS 知识中心视频：如何开始使用 AWS Organizations](#)
- YouTube 上的 7 分钟视频[使用 AWS IAM Identity Center 管理用户对多个 AWS 账户的访问权限](#)
- YouTube 上的 9 分钟视频[如何为本地 Active Directory 用户设置 AWS Single Sign On](#)

以下概念图说明了您将了解的内容。



要使一个或多个 AWS 账户 能够在企业中开始使用 AWS Cloud9，请按照您已经拥有的 AWS 资源执行步骤。

您是否已有可以在 AWS Organizations 中用作组织的管理账户的 AWS 账户？	您在 AWS Organizations 的管理账户中是否有一个企业？	所需的全部 AWS 账户是否均为该组织的成员？	该企业是否设置为使用 IAM Identity Center？	对于需要使用 AWS Cloud9 的所有用户和组，是否在该组织中进行了设置？	从这一步开始
否	—	—	—	—	步骤 1：为企业创建管理账户
是	否	—	—	—	步骤 2：为管理账户创建企业
是	是	否	—	—	步骤 3：将成员账户添加到企业
是	是	是	否	—	步骤 4：在企业中启用 IAM Identity Center
是	是	是	是	否	步骤 5. 在企业中设置组和用户
是	是	是	是	是	步骤 6：在企业中允许组和用户使用 AWS Cloud9

步骤 1：为企业创建管理账户

Note

您的企业可能已经设置了一个管理账户。如果您的企业有 AWS 账户 管理员，请在开始执行以下过程之前与该管理员联系。如果您已有一个管理账户，请向前跳至[步骤 2：为管理账户创建企业](#)。

要使用 AWS IAM Identity Center (IAM Identity Center)，您必须具有 AWS 账户。您的 AWS 账户用作 AWS Organizations 中某个组织的管理账户。有关更多信息，请参阅 AWS Organizations 用户指南中 [AWS Organizations 术语和概念](#) 中有关管理账户的讨论。

要观看与以下过程相关的 4 分钟视频，请观看 YouTube 上的 [Creating an Amazon Web Services account](#) (创建 Amazon Web Services 账户)。

要创建管理账户：

1. 访问 <https://aws.amazon.com/>。
2. 选择 Sign In to the Console (登录控制台)。
3. 选择创建新的 AWS 账户。
4. 按照屏幕上的说明完成该过程。其中包括向 AWS 提供您的电子邮件地址和信用卡信息。另外，您必须使用手机输入 AWS 提供的验证码。

在完成创建账户后，AWS 会向您发送确认电子邮件。不要转到下一步，直到您收到该确认。

步骤 2：为管理账户创建企业

Note

您的企业可能已经将 AWS Organizations 设置为使用管理账户。如果您的企业有 AWS 账户 管理员，请在开始执行以下过程之前与该管理员联系。如果您已将 AWS Organizations 设置为使用管理账户，请向前跳至[步骤 3：将成员账户添加到企业](#)。

要使用 IAM Identity Center，您在 AWS Organizations 中必须有使用管理账户的企业。有关更多信息，请参阅 AWS Organizations 用户指南中的 [AWS Organizations 术语和概念](#) 中有关企业的讨论。

要在 AWS Organizations 中为管理 AWS 账户 创建组织，请按照《AWS Organizations 用户指南》中的以下说明操作：

1. [创建企业](#)
2. [启用企业中的所有功能](#)

要观看与这些过程相关的 4 分钟视频，请查看 YouTube 上的 [AWS 知识中心视频：如何开始使用 AWS Organizations](#)。

步骤 3：将成员账户添加到企业

Note

您的企业可能已经为 AWS Organizations 设置了所需的成员账户。如果您的企业有 AWS 账户管理员，请在开始执行以下过程之前与该管理员联系。如果您已使用所需成员账户设置 AWS Organizations，请向前跳至 [步骤 4：在企业中启用 IAM Identity Center](#)。

在此步骤中，您在 AWS Organizations 中添加将用作组织的成员账户的 AWS 账户。有关更多信息，请参阅 AWS 用户指南中 [AWS Organizations Organizations 术语和概念](#) 中有关成员账户的讨论。

Note

您不需要将任何成员账户添加到组织。您可以在企业中只有一个管理账户时使用 IAM Identity Center。稍后，您可以添加成员到组织（如果需要）。如果您不希望现在添加成员账户，请向前跳至 [步骤 4：在企业中启用 IAM Identity Center](#)。

要将成员账户添加到 AWS Organizations 中的企业，请按照 AWS Organizations 用户指南 中的下列一组或两组说明操作。您可以根据需要多次按照这些说明重复操作，直至所需的全部 AWS 账户 均已成为组织的成员：

- [在组织中创建 AWS 账户](#)
- [邀请 AWS 账户 加入组织](#)

步骤 4：在企业中启用 IAM Identity Center

Note

您的企业可能已经将 AWS Organizations 设置为使用 IAM Identity Center。如果您的企业有 AWS 账户 管理员，请在开始执行以下过程之前与该管理员联系。如果您已将 AWS Organizations 设置为使用 IAM Identity Center，请向前跳至[步骤 5：在企业中设置组和用户](#)。

在本步骤中，您允许 AWS Organizations 中的企业使用 IAM Identity Center。为此，请按照《AWS IAM Identity Center 用户指南》中的这些说明集操作：

1. [IAM Identity Center 先决条件](#)
2. [启用 IAM Identity Center](#)

步骤 5：在企业中设置组和用户

Note

您的企业可能已经从 IAM Identity Center 目录、AWS Managed Microsoft AD 或在 AWS Directory Service 中托管的 AD Connector 目录为 AWS Organizations 设置了组和用户。如果您的企业有 AWS 账户 管理员，请在开始执行以下过程之前与该管理员联系。如果您已经从 IAM Identity Center 目录或 AWS Directory Service 为 AWS Organizations 设置了组和用户，请向前跳至[步骤 6：允许组和用户在企业中使用 AWS Cloud9](#)。

在此步骤中，您可以在 IAM Identity Center 目录中为组织创建组和用户。或者，您连接到 AWS Managed Microsoft AD 或在组织的 AWS Directory Service 中托管的 AD Connector 目录。在后面的步骤中，您向组授予使用 AWS Cloud9 所需的访问权限。

- 如果您使用的是适用于企业的 IAM Identity Center 目录，请按照《AWS IAM Identity Center 用户指南》中的这些说明集操作。根据需要多次重复这些步骤，直至您添加了所有需要的组和用户：
 1. [添加组](#)。我们建议为组织中的任何 AWS Cloud9 管理员创建至少一个组。然后，重复此步骤，为组织中的所有 AWS Cloud9 用户创建另一个组。（可选）您也可以重复此步骤，为组织中您要与其共享现有 AWS Cloud9 开发环境的所有用户创建第三个组。但不允许他们自行创建环境。为便于使用，我们建议分别将这些组命名为 AWSCloud9Administrators、AWSCloud9Users 和

AWSCloud9EnvironmentMembers。有关更多信息，请参阅[适用于 AWS Cloud9 的 AWS 托管式（预定义）策略](#)。

2. [添加用户](#)。

3. [将用户添加到组](#)。将任何 AWS Cloud9 管理员添加到 AWSCloud9Administrators 组，重复此步骤以将 AWS Cloud9 用户添加到 AWSCloud9Users 组。或者，也可以重复此步骤，以将所有剩余的用户添加到 AWSCloud9EnvironmentMembers 组中。将用户添加到组是 AWS 最佳安全实践，可帮助您更好地控制、跟踪 AWS 资源访问以及排除问题。


- 如果您为组织使用 AWS Managed Microsoft AD 或在您 AWS Directory Service 中管理的 AD Connector 目录，请参阅《AWS IAM Identity Center 用户指南》中的[连接至 Microsoft AD 目录](#)。

步骤 6：在企业中允许组和用户使用 AWS Cloud9

原定设置情况下，在 AWS Organizations 中，组织中的大多数用户和组没有 AWS 服务的访问权限，包括 AWS Cloud9。在此步骤中，您使用 IAM Identity Center 以允许 AWS Organizations 中企业的用户和组通过任意参与账户的组合使用 AWS Cloud9。

1. 在 [IAM Identity Center 控制台](#) 中，在服务导航窗格中选择 AWS 账户。
2. 选择 Permission sets (权限集) 选项卡。
3. 选择 Create permission set (创建权限集) 设置。
4. 选择 Create a custom permission set (创建自定义权限集)。
5. 为此权限集输入 Name (名称)。我们建议为组织中的任何 AWS Cloud9 管理员创建至少一个权限集。然后，重复此过程中的步骤 3 到 10，为组织中的所有 AWS Cloud9 用户创建另一个权限集。（可选）您也可以重复此过程中的步骤 3 到 10，为组织中您希望与其共享现有 AWS Cloud9 开发环境的所有用户创建第三个权限集。但不允许他们自行创建环境。为便于使用，我们建议分别将这些权限集命名为 AWSCloud9AdministratorsPerms、AWSCloud9UsersPerms 和 AWSCloud9EnvironmentMembersPerms。有关更多信息，请参阅[适用于 AWS Cloud9 的 AWS 托管式（预定义）策略](#)。
6. 为权限集输入可选的 Description (描述)。
7. 为权限集选择 Session duration (会话持续时间)，或者保留默认的会话持续时间 1 hour (1 小时)。
8. 选择附加 AWS 托管式策略。
9. 在策略列表中，选中正确的 Policy name (策略名称) 条目旁边的下列框之一。（请勿选择策略名称。如果在列表中没有看到任何策略名称，请在 Search (搜索) 框中输入策略名称以显示该名称。）
 - 对于 AWSCloud9AdministratorsPerms 权限集，选择 AWSCloud9Administrator。

- 对于 AWSCloud9UsersPerms 权限集，选择 AWSCloud9User。
- (可选) 对于 AWSCloud9EnvironmentMembersPerms 权限集，选择 AWSCloud9EnvironmentMember。

 Note

要了解除 AWS Cloud9 所需策略之外您还可以添加的策略，请参阅 IAM 用户指南中的[托管式策略和内联策略](#)以及[了解策略授予的权限](#)。


10 选择 Create (创建)。

11 完成创建所需的全部权限集之后，在 AWS Organizations 选项卡上，选择要为其分配 AWS Cloud9 访问权限的 AWS 账户。如果 AWS Organization 选项卡不可见，则在服务导航窗格中，选择 AWS 账户。这将显示 AWS Organization 选项卡。

12 选择 Assign users (分配用户)。

13 在 Groups (组) 选项卡上，选择要将 AWS Cloud9 访问权限分配到的组名称旁边的框。请勿选择组名称本身。

- 如果您为组织使用 IAM Identity Center 目录，您可能已经为 AWS Cloud9 管理员创建了一个名为 AWSCloud9Administrators 的组。
- 如果您为组织使用 AWS Managed Microsoft AD 或在您 AWS Directory Service 中管理的 AD Connector 目录，请选择目录的 ID。接下来，输入组的部分或全部名称，然后选择 Search connected directory (搜索连接的目录)。最后，选择要将 AWS Cloud9 访问权限分配到的组名称旁边的框。

 Note

我们建议您将 AWS Cloud9 访问权限分配到组而非单独的用户。此 AWS 安全最佳实践可帮助您更好地控制、跟踪 AWS 资源访问以及排除问题。

14 选择 Next: Permissions sets (下一步 : 权限集)。

15 选中要分配给该组的权限集名称旁边的框 (例如，一组 AWS Cloud9 管理员的 AWSCloud9AdministratorsPerms)。请勿选择权限集名称本身。

16 选择 Finish (结束)。

17 选择转至 AWS 账户。

18 对于要分配给组织中 AWS 账户 的其他 AWS Cloud9 访问权限，请重复此过程中的步骤 11 到 17。

步骤 7：开始使用 AWS Cloud9

在完成本主题中的上述步骤后，您和您的用户就可以登录到 IAM Identity Center 并开始使用 AWS Cloud9。

1. 如果您已登录到 AWS 账户或 IAM Identity Center，请注销。要执行此操作，请参阅 AWS Support 网站上的[如何注销我的 AWS 账户](#)或《AWS IAM Identity Center 用户指南》中的[如何注销用户门户](#)。
2. 要登录 IAM Identity Center，请按照《AWS IAM Identity Center 用户指南》中[如何接受邀请加入 IAM Identity Center](#)中的说明操作。这包括转至唯一登录 URL，并使用登录凭证进行登录。您的 AWS 账户 管理员将通过电子邮件或其他方式向您提供此信息。

Note

确保将您获得的唯一登录 URL 加入书签。这样，以后就可以轻松地返回到它了。此外，请确保将此 URL 的唯一登录凭证存储在安全的位置。

根据 AWS 账户 管理员向您提供的不同级别的 AWS Cloud9 访问权限，此 URL、用户名和密码的组合可能会有不同。例如，您可以使用一个 URL、用户名和密码来获得对一个账户的 AWS Cloud9 管理员访问权限。您可以使用不同的 URL、用户名和密码，仅允许 AWS Cloud9 用户访问不同的账户。

3. 登录 IAM Identity Center 之后，选择 AWS 账户磁贴。
4. 从显示的下拉列表中选择用户的显示名称。如果显示多个名称，请选择开始使用 AWS Cloud9 所需的名称。如果您不确定应该选择哪个名称，请联系 AWS 账户 管理员。
5. 选择您的用户显示名称旁的 Management console (管理控制台) 链接。如果显示多个 Management console (管理控制台) 链接，请选择正确权限集旁边的链接。如果您不确定应该选择哪个链接，请联系 AWS 账户 管理员。
6. 在 AWS Management Console 中，执行下列操作之一：
 - 如果已经显示 Cloud9，则选择此项。
 - 展开 All services (所有服务)，然后选择 Cloud9。
 - 在 Find services (查找服务) 框中，键入 Cloud9，然后按 Enter。
 - 在 AWS 导航栏上，选择 Service (服务)，然后选择 Cloud9。

此时将显示 AWS Cloud9 控制台，您可以开始使用 AWS Cloud9。

后续步骤

任务	请参阅本主题
创建 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 在新环境中处理代码。	创建环境
了解如何使用 AWS Cloud9 IDE。	入门：基本教程 和 使用 IDE
利用文本聊天支持实时邀请其他用户与您一起使用新环境。	使用共享环境

适用于 AWS Cloud9 的其他设置选项（团队和企业）

本主题假定您已完成[团队设置](#)或[企业设置](#)中的设置步骤。

在[团队设置](#)或[企业设置](#)中，您创建了组并直接向这些组添加了 AWS Cloud9 访问权限。这是为了确保这些组中的用户可以访问 AWS Cloud9。在本主题中，您添加更多的访问权限，以限制这些组中的用户可以创建的环境类型。这有助于控制您的 AWS 账户和企业中 AWS Cloud9 的相关成本。

要添加这些访问权限，您可以创建一组自己的策略，定义您要实施的 AWS 访问权限。我们将其中每个策略称为客户管理型策略。然后，您将这些客户管理型策略附加到用户所属的组。在某些情况下，您还必须分离已附加到这些组的现有 AWS 托管式策略。要进行此设置，请按照本主题中的步骤操作。

Note

以下过程仅介绍为 AWS Cloud9 用户附加和分离策略。这些过程假设您已经有单独的 AWS Cloud9 用户组和 AWS Cloud9 管理员组。这些过程还假设您在 AWS Cloud9 管理员组中只有有限数量的用户。此 AWS 安全最佳实践可帮助您更好地控制、跟踪 AWS 资源访问以及排除问题。

- [步骤 1：创建客户托管式策略](#)
- [步骤 2：向组添加客户托管式策略](#)
- [使用 AWS Cloud9 的团队的客户托管式策略示例](#)

步骤 1：创建客户托管式策略

您可以使用 [AWS Management Console](#) 或 [AWS Command Line Interface \(AWS CLI\)](#) 创建客户托管式策略。

Note

此步骤仅用于为 IAM 组创建托管式策略。要为 AWS IAM Identity Center 中的组创建自定义权限集，请跳过此步骤并按照《AWS IAM Identity Center 用户指南》中的[创建权限集](#)中的说明操作。在本主题中，按照说明创建自定义权限集。如需了解相关的自定义权限策略，请参阅本主题后面的[使用 AWS Cloud9 的团队的客户托管式策略示例](#)。

使用控制台创建客户托管式策略

1. 登录 AWS Management Console (如果尚未登录) 。

我们建议您使用您 AWS 账户 中的管理员用户的凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

2. 打开 IAM 控制台。为此，请在控制台的导航栏中选择 Service (服务)。然后，选择 IAM。

3. 在服务导航窗格中，选择策略。

4. 选择 Create policy (创建策略) 。

5. 在 JSON 选项卡中，粘贴我们建议的[客户托管式策略示例](#)之一。

Note

您还可以创建自己的客户托管式策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略参考](#)和 AWS 服务的[文档](#)。

6. 选择Review policy (查看策略) 。

7. 在查看策略页面上，为策略键入名称和可选的描述，然后选择创建策略。

对您要创建的每个其他客户管理型策略重复此步骤。然后，向前跳到[使用控制台向组添加客户管理型策略](#)。

使用 AWS CLI 创建客户托管式策略

1. 在运行 AWS CLI 的计算机上，创建文件来描述策略（例如，policy.json）。

如果您创建具有不同文件名的文件，请在整个过程中使用此名称进行替换。

2. 将我们建议的[客户托管式策略示例](#)之一粘贴到 policy.json 文件中。

Note

您还可以创建自己的客户托管式策略。有关更多信息，请参阅《IAM 用户指南》和 AWS 服务的[文档](#)中的 [IAM JSON 策略参考](#)。

3. 从终端或命令提示符处，切换到包含 policy.json 文件的目录。
4. 运行 IAM create-policy 命令，为策略和 policy.json 文件指定名称。

```
aws iam create-policy --policy-document file://policy.json --policy-name MyPolicy
```

在上述命令中，将 MyPolicy 替换为策略的名称。

向前跳到[使用 AWS CLI 向组添加客户托管式策略](#)。

步骤 2：向组添加客户托管式策略

您可以使用 [AWS Management Console](#) 或 [AWS 命令行界面 \(AWS CLI\)](#) 向组添加客户管理型策略。

Note

此步骤仅用于将客户托管式策略添加到 IAM 组。要将自定义权限集添加到 AWS IAM Identity Center 中的组，请跳过此步骤并按照《AWS IAM Identity Center 用户指南》中的[分配用户访问权限](#)中的说明操作。

使用控制台向组添加客户托管式策略

1. 使用之前步骤中打开的 IAM 控制台，在服务的导航窗格中，选择 Groups（组）。
2. 选择组名。
3. 在 Permissions（权限）选项卡中，在 Managed Policies（托管式策略）处，选择 Attach Policy（附加策略）。

4. 在策略名称列表中，选择要附加到组的各个客户管理型策略旁的复选框。如果在列表中没有看到特定策略名称，请在 Filter（筛选条件）框中输入策略名称以显示该名称。
5. 选择 Attach Policy（附加策略）。

使用 AWS CLI 向组添加客户托管式策略

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

运行 IAM `attach-group-policy` 命令，在命令中指定组的名称和策略的 Amazon Resource Name (ARN)。

```
aws iam attach-group-policy --group-name MyGroup --policy-arn
arn:aws:iam::123456789012:policy/MyPolicy
```

在上述命令中，将 `MyGroup` 替换为组的名称。将 `123456789012` 替换为 AWS 账户 ID。并将 `MyPolicy` 替换为客户管理型策略的名称。

使用 AWS Cloud9 的团队的客户托管式策略示例

以下是一些策略示例，您可以使用它们来限制组中的用户可以在 AWS 账户 中创建的环境。

- [防止组中的用户创建环境](#)
- [防止组中的用户创建 EC2 环境](#)
- [允许组中的用户仅使用特定的 Amazon EC2 实例类型创建 EC2 环境](#)
- [允许组中的用户在每个 AWS 区域仅创建一个 EC2 环境](#)

防止组中的用户创建环境

以下客户管理型策略在附加到 AWS Cloud9 用户组时，可防止这些用户在 AWS 账户 中创建环境。如果您希望 AWS 账户 中的管理员用户来管理创建环境事宜，此功能非常有用。否则，AWS Cloud9 用户组中的用户会执行此操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloud9:CreateEnvironmentEC2",
        "cloud9:CreateEnvironmentSSH"
      ],
      "Resource": "*"
    }
  ]
}
```

前面的客户管理型策略显式覆盖了已经附加到 AWS Cloud9 用户组的 AWSCloud9User 托管式策略中 "Resource": "*" 上适用于 "Action": "cloud9:CreateEnvironmentEC2" 和 "cloud9:CreateEnvironmentSSH" 的 "Effect": "Allow"。

防止组中的用户创建 EC2 环境

以下客户管理型策略在附加到 AWS Cloud9 用户组时，可以防止这些用户在 AWS 账户中创建 EC2 环境。如果您希望 AWS 账户中的管理员用户来管理创建 EC2 环境，此功能非常有用。否则，AWS Cloud9 用户组中的用户会执行此操作。这假定您尚未附加策略以阻止该组中的用户创建 SSH 环境。否则，这些用户无法创建环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*"
    }
  ]
}
```

前面的客户管理型策略显式覆盖了已经附加到 AWS Cloud9 用户组的 AWSCloud9User 托管式策略中 "Resource": "*" 上适用于 "Action": "cloud9:CreateEnvironmentEC2" 的 "Effect": "Allow"。

允许组中的用户仅使用特定的 Amazon EC2 实例类型创建 EC2 环境

以下客户管理型策略附加到 AWS Cloud9 用户组时，允许用户组中的用户在 AWS 账户 中创建仅使用以 t2 开头的实例类型的 EC2 环境。此策略假定您尚未附加策略以阻止该组中的用户创建 EC2 环境。否则，这些用户无法创建 EC2 环境。

您可以将以下策略中的 "t2.*" 替换为其他实例类 (例如, "m4.*")。或者，您可以将其限制为多个实例类或实例类型 (例如, ["t2.*", "m4.*"] 或 ["t2.micro", "m4.large"])。

对于 AWS Cloud9 用户组，将 AWSCloud9User 托管式策略与该组分离。然后，将以下客户管理型策略添加到其位置。如果不分离 AWSCloud9User 托管式策略，以下客户管理型策略将没有效果。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentSSH",
        "cloud9:ValidateEnvironmentName",
        "cloud9:GetUserPublicKey",
        "cloud9:UpdateUserSettings",
        "cloud9:GetUserSettings",
        "iam:GetUser",
        "iam:ListUsers",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "cloud9:InstanceType": "t2.*"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
```

```

    "cloud9:DescribeEnvironmentMemberships"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "Null": {
      "cloud9:UserArn": "true",
      "cloud9:EnvironmentId": "true"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "cloud9.amazonaws.com"
    }
  }
}
]
}

```

前面的客户管理型策略还允许这些用户创建 SSH 环境。要同时防止这些用户创建 SSH 环境，请从前面的客户托管式策略中删除 "cloud9:CreateEnvironmentSSH",。

允许组中的用户在每个 AWS 区域中仅创建一个 EC2 环境

以下客户管理型策略在附加到 AWS Cloud9 用户组时，允许每个用户为 AWS Cloud9 在其中可用的每个 AWS 区域中创建最多一个 EC2 环境。这通过将环境的名称限制为该 AWS 区域中的一个特定名称来实现。在此示例中，环境仅限于 my-demo-environment。

Note

AWS Cloud9 不允许将环境限制为特定的 AWS 区域进行创建。AWS Cloud9 也不对可以创建的总环境数启用限制。唯一的例外是已发布的[服务限制](#)。

对于 AWS Cloud9 用户组，从组中分离 AWSCloud9User 托管式策略，然后将以下客户托管式策略添加到其位置。如果不分离 AWSCloud9User 托管式策略，以下客户管理型策略没有效果。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentSSH",
        "cloud9:ValidateEnvironmentName",
        "cloud9:GetUserPublicKey",
        "cloud9:UpdateUserSettings",
        "cloud9:GetUserSettings",
        "iam:GetUser",
        "iam:ListUsers",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentEC2"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloud9:EnvironmentName": "my-demo-environment"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:DescribeEnvironmentMemberships"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Null": {

```

```

        "cloud9:UserArn": "true",
        "cloud9:EnvironmentId": "true"
    }
}
},
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "cloud9.amazonaws.com"
        }
    }
}
]
}

```

前面的客户管理型策略允许这些用户创建 SSH 环境。要同时防止这些用户创建 SSH 环境，请从前面的客户托管式策略中删除 "cloud9:CreateEnvironmentSSH",。

有关更多示例，请参阅 [客户管理型策略示例](#)：

后续步骤

任务	请参阅本主题
创建 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 在新环境中处理代码。	创建环境
了解如何使用 AWS Cloud9 IDE。	入门：基本教程 和 使用 IDE
利用文本聊天支持实时邀请其他用户与您一起使用新环境。	使用共享环境

入门：AWS Cloud9 的基本教程

您是首次使用 AWS Cloud9 吗？如果您尚未执行此操作，请查看[什么是 AWS Cloud9](#) 中有关 AWS Cloud9 的一般信息。

在以下教程中，您将在 AWS Cloud9 创建一个环境，然后使用该环境创建一个简单的应用程序。两个教程具有相同的输入和结果，但一个使用 AWS Cloud9 控制台，另一个使用 [AWS Command Line Interface \(AWS CLI \)](#)。您可以选择执行其中一个或执行两者。

当您完成这些教程后，您可以在 [了解 AWS Cloud9 IDE](#) 中了解有关 AWS Cloud9 IDE 的更多信息。

主题

- [教程：欢迎使用 AWS Cloud9 \(控制台 \)](#)
- [教程：Hello AWS Cloud9 \(CLI\)](#)

教程：欢迎使用 AWS Cloud9 (控制台)

本教程将为您提供 AWS Cloud9 的一览。它涵盖了如何使用和浏览 AWS Cloud9 控制台。

在本教程中，您设置一个 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 进行第一个应用程序的代码编写、运行和调试。

完成本教程需要大约一个小时。

Warning

完成本教程可能会在您的 AWS 区域中产生费用。其中包括可能对 Amazon EC2 收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

先决条件

要成功完成本教程，您必须首先完成 [设置 AWS Cloud9](#) 中的步骤。

步骤

- [步骤 1：创建环境](#)

- [步骤 2：IDE 基本导览](#)
- [第 3 步：清除](#)
- [相关信息](#)

步骤 1：创建环境

([教程：欢迎使用 AWS Cloud9 \(控制台\)](#) 的第一步)

在此步骤中，使用 AWS Cloud9 控制台先创建、再打开 AWS Cloud9 开发环境。

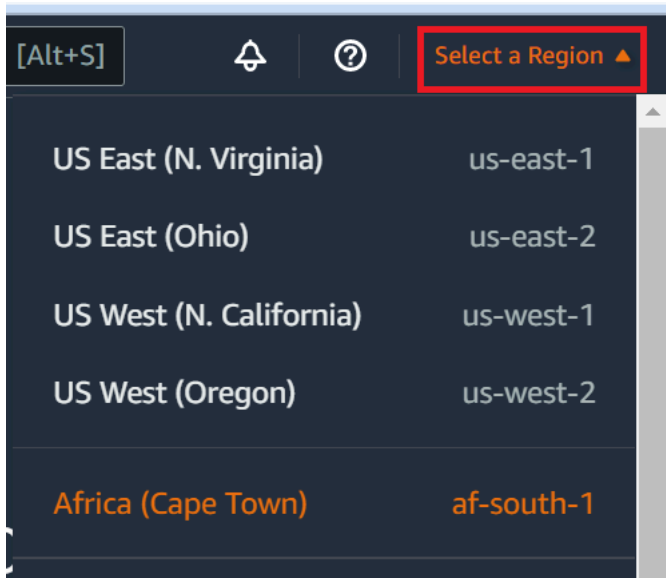
Note

如果您已创建要用于本教程的环境，请打开该环境，然后跳至 [步骤 2：IDE 基本导览](#)。

在 AWS Cloud9 中，开发环境（或简称环境）是指存储开发项目的文件以及运行工具以开发应用程序的位置。在本教程中，您会创建一个 EC2 环境的特殊环境，然后在该环境中使用文件和工具。

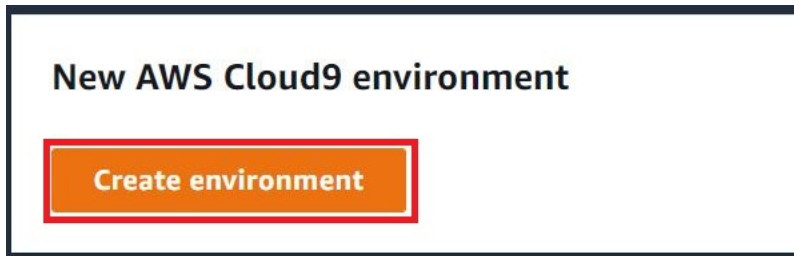
使用控制台创建 EC2 环境

1. 登录到 AWS Cloud9 控制台：
 - 如果您是使用 AWS 账户的唯一用户，或者您是单个 AWS 账户中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的企业使用 AWS IAM Identity Center，请联系 AWS 账户管理员以获取登录说明。
 - 如果您是课堂中的学生，请联系您的导师以获取登录说明。
2. 登录到 AWS Cloud9 控制台后，在顶部导航栏中选择要在其中创建环境的 AWS 区域。有关可用的 AWS 区域列表，请参阅《AWS 一般参考》中的 [AWS Cloud9](#)。



3. 在显示的位置之一选择大型 Create environment (创建环境) 按钮。

如果您还没有 AWS Cloud9 环境，按钮在欢迎页面上显示。



如果您已拥有 AWS Cloud9 环境，按钮会以下面的方式显示。



4. 在 Create environment (创建环境) 页面上，在 Name (名称) 中输入环境的名称。
5. 在 Description (描述) 中输入对环境的描述。在本教程中，请使用 This environment is for the AWS Cloud9 tutorial.
6. 在 Environment type (环境类型) 中，选择 Create EC2 instance (创建 EC2 实例) 以创建 Amazon EC2 环境。
 - New EC2 instance (新 EC2 实例) – 启动 AWS Cloud9 可以直接通过 SSH 连接的新 Amazon EC2 实例。您可以使用 Systems Manager 与新的 Amazon EC2 实例进行交互，有关更多信息，请参阅[使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

- Existing compute (现有计算) – 启动需要 SSH 登录详细信息的现有 Amazon EC2 实例，该 Amazon EC2 实例必须具有入站安全组规则。
- 如果您选择 Existing compute (现有计算) 选项，则会自动创建服务角色。您可以在设置屏幕底部的备注中查看服务角色的名称。

Note

对于使用现有计算的 Amazon EC2 实例创建的 AWS Cloud9 环境，自动关闭功能不可用。

Warning

为环境创建 Amazon EC2 实例可能会对您的 AWS 账户收取 Amazon EC2 费用。使用 Systems Manager 管理与 EC2 实例的连接不会产生额外费用。

7. 在新 EC2 实例面板的 Instance type (实例类型) 上，保留默认选择。此选项可能会具有较少的 RAM 和 vCPU。但是，对于本教程来说，这个内存量已经足够。

Warning

选择 RAM 和 vCPU 更高的实例类型可能会导致对您的 AWS 账户收取额外的 Amazon EC2 费用。

8. 对于平台，选择您所需的 Amazon EC2 实例的类型：Amazon Linux 2023、Amazon Linux 2 或 Ubuntu 22.04 LTS。AWS Cloud9 会创建实例，然后将环境连接到该实例。

Important

建议您为 EC2 环境选择 Amazon Linux 2023 选项。在提供安全、稳定和高性能的运行系统环境之外，Amazon Linux 2023 AMI 还包括到 2024 年的长期支持。

有关更多信息，请参阅 [AL2023 页](#)。

9. 为 Timeout (超时) 选择时间段。此选项决定 AWS Cloud9 在处于非活动状态多长时间之后进入休眠。在连接到环境的 IDE 的所有 Web 浏览器实例都关闭之后，AWS Cloud9 等待这样一段时间，然后关闭环境的 Amazon EC2 实例。

⚠ Warning

选择较长的时间段可能会导致在您的 AWS 账户 中产生更多费用。

10. 在 Network settings (网络设置) 面板上，从以下两个选项中选择如何访问您的环境：
 - AWS System Manager (SSM) – 此方法使用 SSM 访问环境，没有开放入站端口。
 - Secure Shell (SSH) – 此方法使用 SSH 访问环境，需要打开入站端口。
11. 选择 VPC Settings (VPC 设置) 以显示您的环境的 Amazon Virtual Private Cloud 和子网。AWS Cloud9 使用 Amazon Virtual Private Cloud (Amazon VPC) 与新创建的 Amazon EC2 实例通信。在本教程中，我们不建议您更改预先选择的默认设置。使用默认设置，AWS Cloud9 会尝试自动使用与新环境在同一 AWS 账户 和区域的默认 VPC 及其单个子网。

您可以在[使用控制台创建 EC2 环境](#)和[AWS Cloud9 开发环境的 VPC 设置](#)中找到有关 Amazon VPC 选择的更多信息。

12. 通过为每个标签提供 Key (键) 和 Value (值)，添加最多 50 个标签。要执行此操作，请选择 Add new tag (添加新标签)。这些标签将作为资源标签附加到 AWS Cloud9 环境，并传播到以下底层资源：AWS CloudFormation 堆栈、Amazon EC2 实例和 Amazon EC2 安全组。要了解有关标签的更多信息，请参阅[IAM 用户指南](#)中的[使用 AWS 资源标签控制访问](#)以及本指南中的[高级信息](#)。

⚠ Warning

如果您在创建标签后更新这些标签，则更改不会传播到底层资源。有关更多信息，请参阅[标签高级信息](#)中的[将标签更新传播到底层资源](#)。

13. 选择 Create (创建) 以创建您的环境，然后系统会将您重定向到主页。如果成功创建账户，则 AWS Cloud9 控制台顶部会出现一个绿色的闪光条。您可以选择新环境并选择 Open in Cloud9 (在 Cloud9 中打开) 以启动 IDE。

Delete

View details

Open in Cloud9 

Create environment

如果未能创建账户，则 AWS Cloud9 控制台顶部会出现一个红色的闪光条。您可能会因为 Web 浏览器、AWS 访问权限、实例或关联的网络的问题而未能创建账户。您可以在[AWS Cloud9 故障排除](#)部分找到有关可能的修复方法的信息。

Note

AWS Cloud9 同时支持 IMDSv1 和 IMDSv2。我们建议采用 IMDSv2，因为与 IMDSv1 相比，它提供了增强型安全级别。有关 IMDSv2 优势的更多信息，请参阅 [AWS 安全博客](#)。关于如何从 IMDSv1 过渡到 IMDSv2 的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 [过渡到使用实例元数据服务版本 2](#)。

Note

如果您的环境使用代理访问互联网，则必须将代理详细信息提供给 AWS Cloud9，以便它可以安装依赖关系。有关更多信息，请参阅 [无法安装依赖项](#)。

后续步骤

[步骤 2：IDE 基本导览](#)

步骤 2：IDE 基本导览

(上一步：[步骤 1：创建环境](#))

本部分教程介绍了几种方式，让您可以使用 AWS Cloud9 IDE 来创建和测试应用程序。

- 您可以使用 editor (编辑器) 窗口来创建和编辑代码。
- 您可以使用 terminal (终端) 窗口或 Run Configuration (运行配置) 窗口来运行您的代码，无需调试。
- 您可以使用 Debugger (调试程序) 窗口调试代码。

使用 JavaScript 和 Node.js 引擎执行以下三个任务。有关使用其他编程语言的说明，请参阅 [AWS Cloud9 的教程](#)。

主题

- [让您的环境就绪](#)
- [编写代码](#)
- [运行代码](#)
- [调试代码](#)

- [后续步骤](#)

让您的环境就绪

运行和调试 JavaScript 代码所需的大部分工具都已安装。但是，在学习本教程的过程中，您还需要获取 Node.js 软件包。请按照如下步骤完成安装。

1. 在菜单栏上 AWS Cloud9 IDE 的顶部，依次选择 Window (窗口)、New Terminal (新终端)，也可使用现有的终端窗口。
2. 在终端窗口 (IDE 底部的一个选项卡)，输入以下内容。

```
npm install readline-sync
```

验证结果是否类似于以下所示。如果还显示 npm WARN 消息，您可以忽略它们。

```
+ readline-sync@1.4.10
added 1 package from 1 contributor and audited 5 packages in 0.565s
found 0 vulnerabilities
```

编写代码

首先，编写一些代码。

1. 在菜单栏上，选择 File (文件)、New File (新建文件)。
2. 将以下 JavaScript 添加到新文件。

```
var readline = require('readline-sync');
var i = 10;
var input;

console.log("Hello Cloud9!");
console.log("i is " + i);

do {
  input = readline.question("Enter a number (or 'q' to quit): ");
  if (input === 'q') {
    console.log('OK, exiting.')
  }
  else{
```

```
        i += Number(input);
        console.log("i is now " + i);
    }
} while (input != 'q');

console.log("Goodbye!");
```

3. 选择 File (文件) 和 Save (保存), 然后将文件另存为 `hello-cloud9.js`。

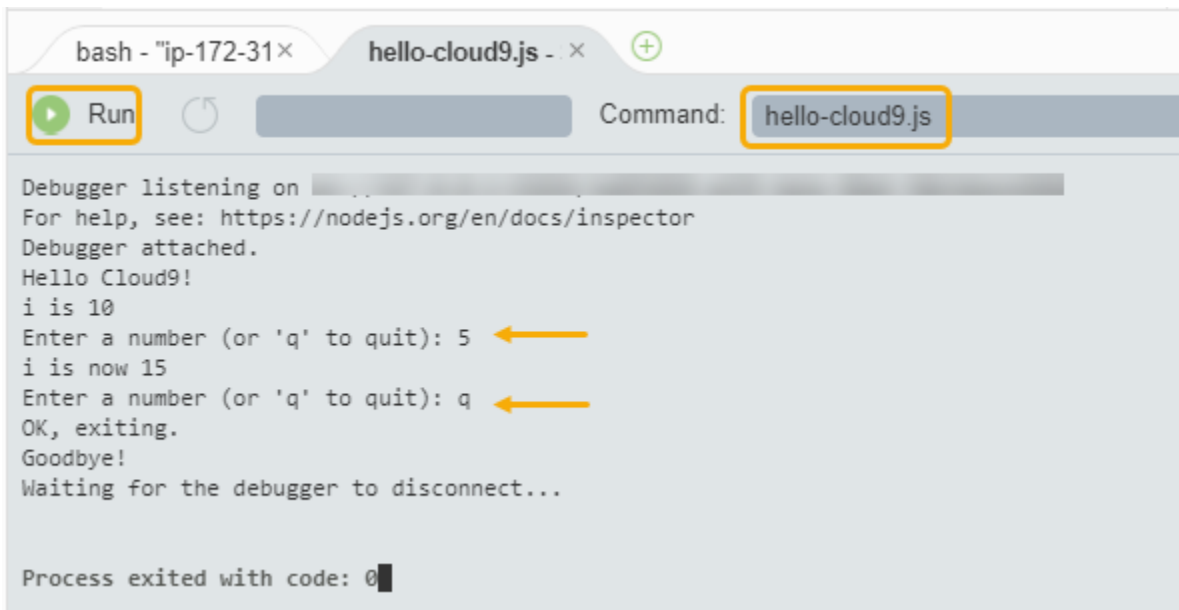
运行代码

接下来, 您可以运行代码了。

根据您正在使用的编程语言, 可能有多种方式可以运行代码。本教程使用的是 JavaScript, 您可以使用终端窗口或 Run Configuration (运行配置) 窗口运行该语言。

使用“运行配置”窗口运行代码

1. 在菜单栏上依次选择运行、运行配置和新建运行配置。
2. 在新的 Run Configuration (运行配置) 窗口 (IDE 底部的一个选项卡) 中, 在 Command (命令) 字段中输入 `hello-cloud9.js`, 然后选择 Run (运行)。
3. 请确保 Run Configuration (运行配置) 提示符处于活动状态, 然后在提示符处输入一个数字来与应用程序交互。
4. 在 Run Configuration (运行配置) 窗口查看代码的输出内容。其内容类似于以下所示。



```
bash - "ip-172-31x" hello-cloud9.js - x
Run Command: hello-cloud9.js
Debugger listening on [redacted]
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Waiting for the debugger to disconnect...

Process exited with code: 0
```

使用终端窗口以运行代码

1. 转到您早前使用的终端窗口（或打开一个新的窗口）。
2. 在终端窗口中，在终端提示符处输入 `ls`，并核实您的代码文件在文件列表中。
3. 在提示符处输入 `node hello-cloud9.js`，启动应用程序。
4. 在提示符处输入一个数字，与应用程序交互。
5. 在您的终端窗口查看代码的输出内容。其内容类似于以下所示。

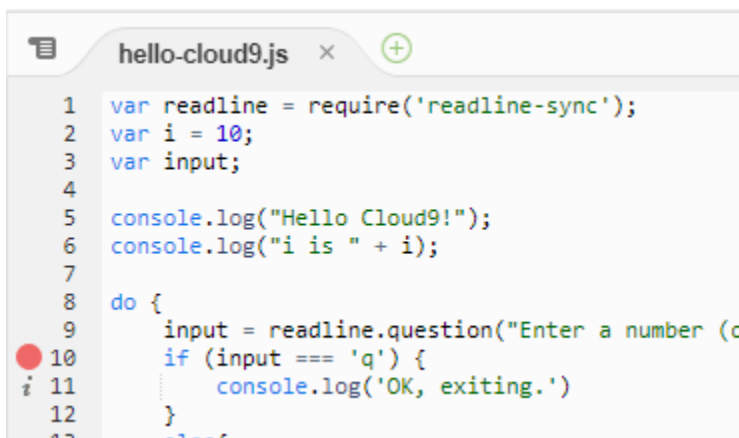


```
node - "ip-172-31" x hello-cloud9.js - ! x +
Admin:~/environment $ node hello-cloud9.js
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Admin:~/environment $
```

调试代码

最后，您可以通过使用 Debugger (调试程序) 窗口调试您的代码。

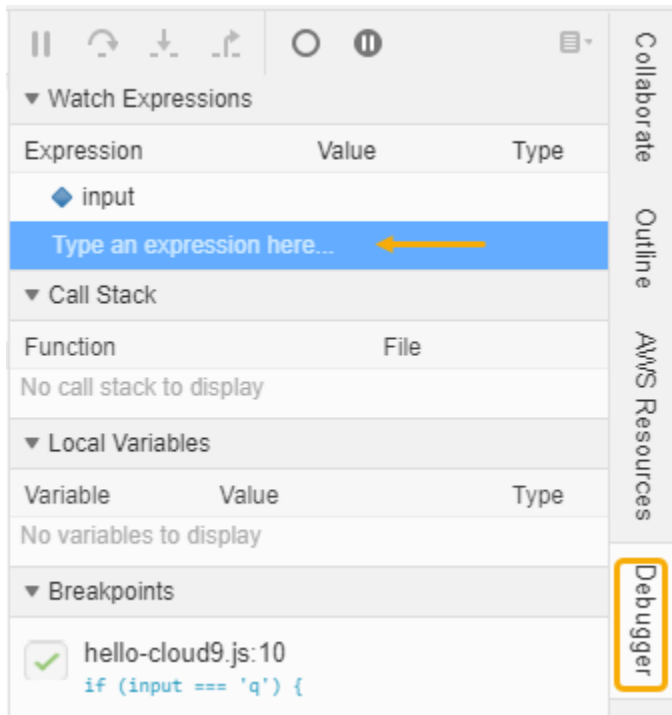
1. 通过选择第 10 行旁边的空白，将断点添加到代码第 10 行 (`if (input === 'q')`) 处。该行号旁边显示一个红色圆圈，如下所示。



```
hello-cloud9.js x +
1 var readline = require('readline-sync');
2 var i = 10;
3 var input;
4
5 console.log("Hello Cloud9!");
6 console.log("i is " + i);
7
8 do {
9   input = readline.question("Enter a number (o
10   if (input === 'q') {
11     console.log('OK, exiting.')
12   }
13 }
```

2. 通过选择 IDE 右侧的 Debugger (调试程序) 按钮，打开 Debugger (调试程序) 窗口。也可以在菜单栏上依次选择 Window 和 Debugger (调试程序)。

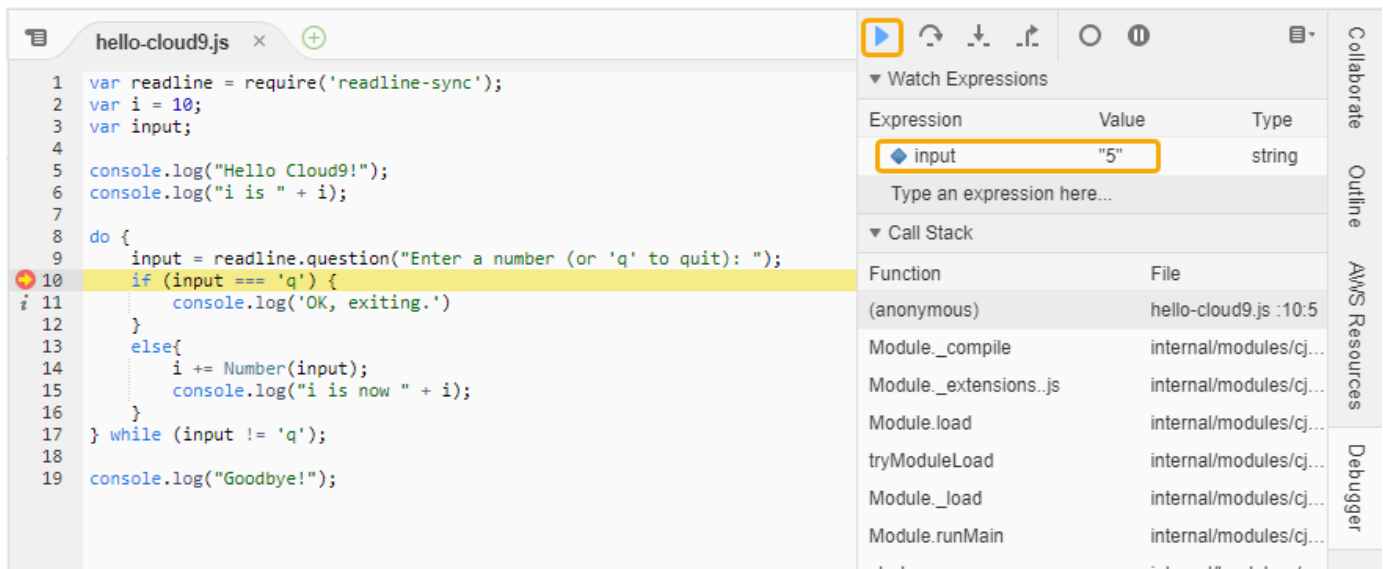
然后，通过在 Debugger (调试程序) 窗口的 Watch Expressions (监视表达式) 部分中选择 Type an expression here (在此处键入表达式) ，对 input 变量进行监视。



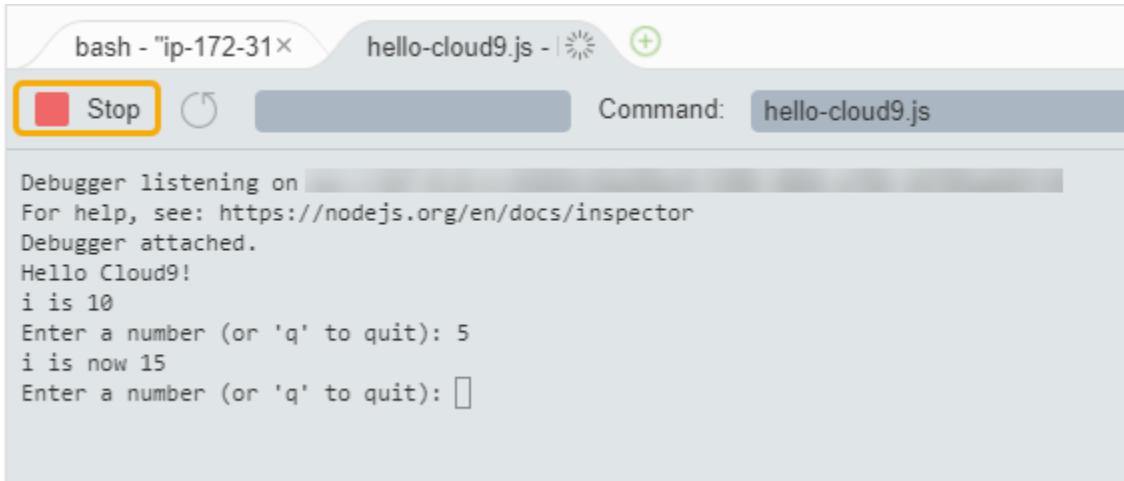
3. 转到您之前使用的 Run Configuration (运行配置) 窗口以运行代码。选择运行。

或者，您可以打开一个新的 Run Configuration (运行配置) 窗口并开始运行代码。要执行此操作，请从菜单栏中依次选择 Run (运行) 、 Run With (运行方式) 、 Node.js 。

4. 在 Run Configuration (运行配置) 提示符下输入一个数字，看到代码在第 10 行暂停。Debugger (调试程序) 窗口显示您在 Watch Expressions (监视表达式) 中输入的值。



5. 在 Debugger (调试程序) 窗口中，选择 Resume (恢复)。这是上一张屏幕截图中突出显示的蓝色箭头图标。
6. 在 Run Configuration (运行配置) 窗口中选择 Stop (停止) 以停止调试程序。



后续步骤

[第 3 步：清除](#)

第 3 步：清除

(上一步：[步骤 2：IDE 基本导览](#))

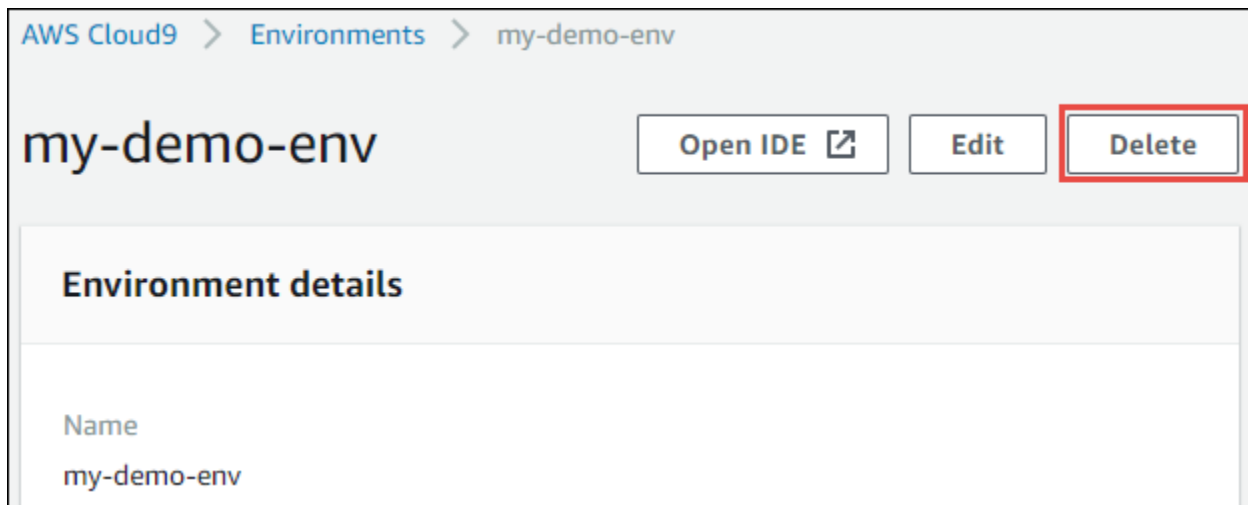
为防止一直对您的 AWS 账户 收取与本教程相关的费用，请删除环境。

Warning

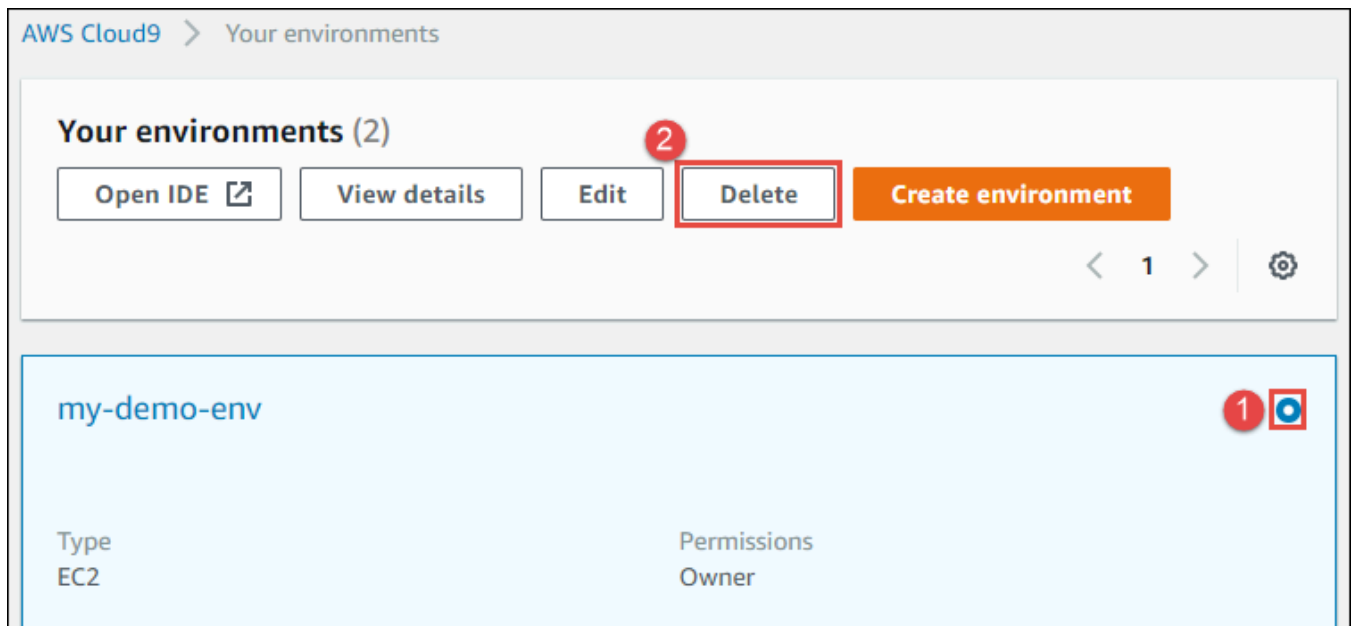
环境在删除后无法恢复。

使用 AWS Cloud9 控制台删除环境

1. 要打开控制面板，请在 IDE 中的菜单栏上依次选择 AWS Cloud9、Go To Your Dashboard (转到您的控制面板)。
2. 请执行以下操作之一：
 - 选择 my-demo-environment 卡内的标题，然后选择 Delete (删除)。



- 选择 my-demo-environment 卡，然后选择 Delete (删除)。



3. 在 Delete (删除) 对话框中，输入 Delete，然后选择 Delete (删除)。删除操作需要几分钟时间。

Note

如果您完全按照本教程进行操作，那么该环境就是一个 EC2 环境，并且 AWS Cloud9 还会终止连接到该环境的 Amazon EC2 实例。

但是，如果您使用 SSH 环境而不是按照教程进行操作，并且该环境已连接到 Amazon EC2 实例，则 AWS Cloud9 不会终止该实例。如果您稍后不终止该实例，可能会一直对您的 AWS 账户收取与该实例相关的 Amazon EC2 费用。

后续步骤

[相关信息](#)

相关信息

以下是 [教程：欢迎使用 AWS Cloud9 \(控制台\)](#) 的更多信息。

- 在创建 EC2 环境时，该环境在默认情况下不包含任何示例代码。要创建环境以及示例代码，请参阅以下任一主题：
 - [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 Amazon Lightsail 实例](#)
 - [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 AWS CodeStar 项目](#)
- 在创建 AWS Cloud9 开发环境时，您将被引导到 AWS Cloud9 来创建一个 Amazon EC2 实例。AWS Cloud9 已创建了实例，然后将环境与其连接。您也可以使用现有的云计算实例或自己的服务器，这称为 SSH 环境。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。

可选的后续步骤

探索下列任何一个或全部主题，进一步熟悉 AWS Cloud9。

Task	请参阅本主题
了解您使用环境可以完成的任务的更多信息。	在环境中工作 AWS Cloud9
尝试其他计算机语言。	AWS Cloud9 的教程
了解有关 AWS Cloud9 IDE 的更多信息。	了解 AWS Cloud9 IDE 中的使用 IDE
利用文本聊天支持实时邀请其他用户使用新环境。	在 AWS Cloud9 中使用共享环境
创建 SSH 环境。这些是使用您创建的云计算实例或服务器（而不是 AWS Cloud9 为您创建的 Amazon EC2 实例）的环境。	在中创建环境 AWS Cloud9 和 SSH 环境主机要求
使用 AWS Toolkit 在 AWS Lambda 函数和无服务器应用程序中创建、运行和调试代码。	通过 AWS Toolkit 使用 AWS Lambda 函数

Task	请参阅本主题
将 AWS Cloud9 与 Amazon Lightsail 结合使用。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例
将 AWS Cloud9 与 AWS CodeStar 结合使用。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CodeStar 项目
将 AWS Cloud9 与 AWS CodePipeline 结合使用。	在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline
将 AWS Cloud9 与 AWS CLI、AWS CloudShell、AWS CodeCommit、AWS Cloud Development Kit (AWS CDK)、GitHub 或者 Amazon DynamoDB，以及 Node.js、Python 或其他编程语言结合使用。	AWS Cloud9 的教程
在 AWS RoboMaker 中使用用于智能机器人应用程序的代码。	AWS RoboMaker 开发人员指南中的 使用 AWS Cloud9 进行开发

要从社区中获取 AWS Cloud9 帮助，请参阅 [AWS Cloud9 开发论坛](#)。（当您进入此论坛时，AWS 可能会要求您登录。）

要从 AWS 直接获得 AWS Cloud9 帮助，请参阅 [AWS Support](#) 页中的支持选项。

教程：Hello AWS Cloud9 (CLI)

本教程将为您提供 AWS Cloud9 的一览。它使用 [AWS Command Line Interface \(AWS CLI\)](#)，使您能够使用命令行设置和卸载所需的资源，而不必通过[图形用户界面](#)。

在本教程中，您设置一个 AWS Cloud9 开发环境，然后使用 AWS Cloud9 IDE 进行第一个应用程序的代码编写、运行和调试。

完成本教程大约需要 1 个小时。

Warning

完成本教程可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

先决条件

要成功完成本教程，您必须首先完成 [设置 AWS Cloud9](#) 中的步骤。

步骤

- [步骤 1：创建环境](#)
- [步骤 2：IDE 基本导览](#)
- [第 3 步：清除](#)
- [相关信息](#)

步骤 1：创建环境

([教程：Hello AWS Cloud9 \(CLI\)](#) 的第一步)

在此步骤中，您使用 AWS CLI 来创建 AWS Cloud9 开发环境。

在 AWS Cloud9 中，开发环境（或简称环境）是指存储开发项目的文件以及运行工具以开发应用程序的位置。在本教程中，您会创建一个 EC2 环境的特殊环境，然后在该环境中使用文件和工具。

使用 AWS CLI 创建 EC2 环境

1. 如果尚未安装并配置 AWS CLI，执行相应操作。为此，请参阅 AWS Command Line Interface 用户指南中的以下内容：
 - [安装 AWS Command Line Interface](#)
 - [快速配置](#)

您可以从以下用户中选择一个，然后使用凭证配置 AWS CLI：

- 已在 [团队设置 AWS Cloud9](#) 中创建的 IAM 用户。

- AWS 账户中的 IAM 管理员，前提是您为该账户中的多位用户定期处理 AWS Cloud9 资源。如果无法以 IAM 管理员身份配置 AWS CLI，请与您的 AWS 账户管理员联系。有关更多信息，请参阅 IAM 用户指南中的[创建您的第一个 IAM 管理员用户和组](#)。
 - AWS 账户根用户，但前提是您始终是使用您自己的 AWS 账户的唯一用户，并且不需要与任何其他用户共享您的环境。我们不建议使用此选项，因为它不是 AWS 安全最佳实践。有关更多信息，请参阅《Amazon Web Services 一般参考》中的[创建、禁用和删除 AWS 账户的访问密钥](#)。
 - 有关其他选项，请联系 AWS 账户管理员或课堂讲师。
2. 在以下 AWS Cloud9 命令中，为 `--region` 和 `--subnet-id` 提供一个值。然后，运行命令并记下 "environmentId" 值以供将来清除。

```
aws cloud9 create-environment-ec2 --name my-demo-environment --description "This environment is for the AWS Cloud9 tutorial." --instance-type t2.micro --image-id resolve:ssm:/aws/service/cloud9/amis/amazonlinux-2-x86_64 --region MY-REGION --connection-type CONNECT_SSM --subnet-id subnet-12a3456b
```

在上述命令中：

- `--name` 代表环境的名称。在本教程中，我们使用名称 `my-demo-environment`。
- `--description` 代表环境的可选描述。
- `--instance-type` 代表 AWS Cloud9 将启动并连接到新环境的 Amazon EC2 实例类型。本示例指定 `t2.micro`，它的 RAM 和 vCPU 相对较低，足以满足本教程的需要。如果指定 RAM 和 vCPU 更高的实例类型，可能会导致对您的 AWS 账户收取额外的 Amazon EC2 费用。有关可用实例类型的列表，请参阅 AWS Cloud9 控制台中的创建环境向导。
- `--image-id` 指定用于创建 EC2 实例的 Amazon Machine Image (AMI) 的标识符。要为实例选择 AMI，您必须指定有效的 AMI 别名或有效的 AWS Systems Manager (SSM) 路径。在上面的示例中，指定了 Amazon Linux 2 AMI 的 SSM 路径。

有关更多信息，请参阅《AWS CLI 命令参考》中的 [create-environment-ec2](#)。

- `--region` 代表 AWS Cloud9 在其中创建环境的 AWS 区域的 ID。有关可用的 AWS 区域列表，请参阅《Amazon Web Services 一般参考》中的 [AWS Cloud9](#)。
- `--connection-type CONNECT_SSM` 指定 AWS Cloud9 通过 Systems Manager 连接到其 Amazon EC2 实例。此选项可确保入站流量禁止进入实例。有关更多信息，请参阅[使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

Note

使用此选项时，您需要创建 `AWSCloud9SSMAccessRole` 服务角色和 `AWSCloud9SSMInstanceProfile`（如果它们还没有创建）。有关更多信息，请参阅 [使用 AWS CLI 管理 Systems Manager 的实例配置文件](#)。

- `--subnet-id` 表示您希望 AWS Cloud9 使用的子网。将 `subnet-12a3456b` 替换为必须与 AWS Cloud9 兼容的 Amazon Virtual Private Cloud (VPC) 的子网 ID。有关更多信息，请参阅 [AWS Cloud9 开发环境的 VPC 设置](#) 中的 [创建 VPC 以及其他 VPC 资源](#)。
 - 当为环境连接到 IDE 的所有 Web 浏览器实例都关闭后，AWS Cloud9 会关闭环境的 Amazon EC2 实例。要配置此时间段，请添加 `--automatic-stop-time-minutes` 和分钟数。较短的时间段可能会减少您的 AWS 账户费用。同样，较长的时间段可能会产生更多的费用。
 - 默认情况下，调用此命令的实体拥有环境。要更改此设置，请添加 `--owner-id` 以及拥有实体的 Amazon Resource Name (ARN)。
3. 成功运行此命令后，为新创建的环境打开 AWS Cloud9 IDE。要执行此操作，请参阅 [在 AWS Cloud9 中打开环境](#)。然后返回到本主题，继续完成 [步骤 2：IDE 基本导览](#)，了解如何使用 AWS Cloud9 IDE 处理您的新环境。

如果您尝试打开环境，但 AWS Cloud9 在至少 5 分钟后未显示 IDE，可能是您的 Web 浏览器、AWS 访问权限、实例或关联的 VPC 出现问题。有关可能的修复措施，请参阅 [无法打开环境](#)。

下一个步骤

[步骤 2：IDE 基本导览](#)

步骤 2：IDE 基本导览

(上一步：[步骤 1：创建环境](#))

本部分教程介绍了几种方式，让您可以使用 AWS Cloud9 IDE 来创建和测试应用程序。

- 您可以使用 editor (编辑器) 窗口来创建和编辑代码。
- 您可以使用 terminal (终端) 窗口或 Run Configuration (运行配置) 窗口来运行您的代码，无需调试。
- 您可以使用 Debugger (调试程序) 窗口调试代码。

使用 JavaScript 和 Node.js 引擎执行这三项任务。有关使用其他编程语言的说明，请参阅[AWS Cloud9 的教程](#)。

主题

- [让您的环境就绪](#)
- [编写代码](#)
- [运行代码](#)
- [调试代码](#)
- [下一个步骤](#)

让您的环境就绪

运行和调试 JavaScript 代码所需的大多数工具都已为您安装。但是，在学习本教程的过程中，您还需要获取 Node.js 软件包。请按照如下步骤完成安装。

1. 在菜单栏上 AWS Cloud9 IDE 的顶部，依次选择 Window (窗口)、New Terminal (新终端)，也可使用现有的终端窗口。
2. 在终端窗口 (IDE 底部的一个选项卡)，输入以下内容。

```
npm install readline-sync
```

验证结果是否类似于以下所示。如果还显示 npm WARN 消息，您可以忽略它们。

```
+ readline-sync@1.4.10
added 1 package from 1 contributor and audited 5 packages in 0.565s
found 0 vulnerabilities
```

编写代码

首先，编写一些代码。

1. 在菜单栏上，选择 File (文件)、New File (新建文件)。
2. 将以下内容 JavaScript 添加到新文件中。

```
var readline = require('readline-sync');
var i = 10;
```

```
var input;

console.log("Hello Cloud9!");
console.log("i is " + i);

do {
    input = readline.question("Enter a number (or 'q' to quit): ");
    if (input === 'q') {
        console.log('OK, exiting.')
    }
    else{
        i += Number(input);
        console.log("i is now " + i);
    }
} while (input !== 'q');

console.log("Goodbye!");
```

3. 选择 File (文件) 和 Save (保存), 然后将文件另存为 hello-cloud9.js。

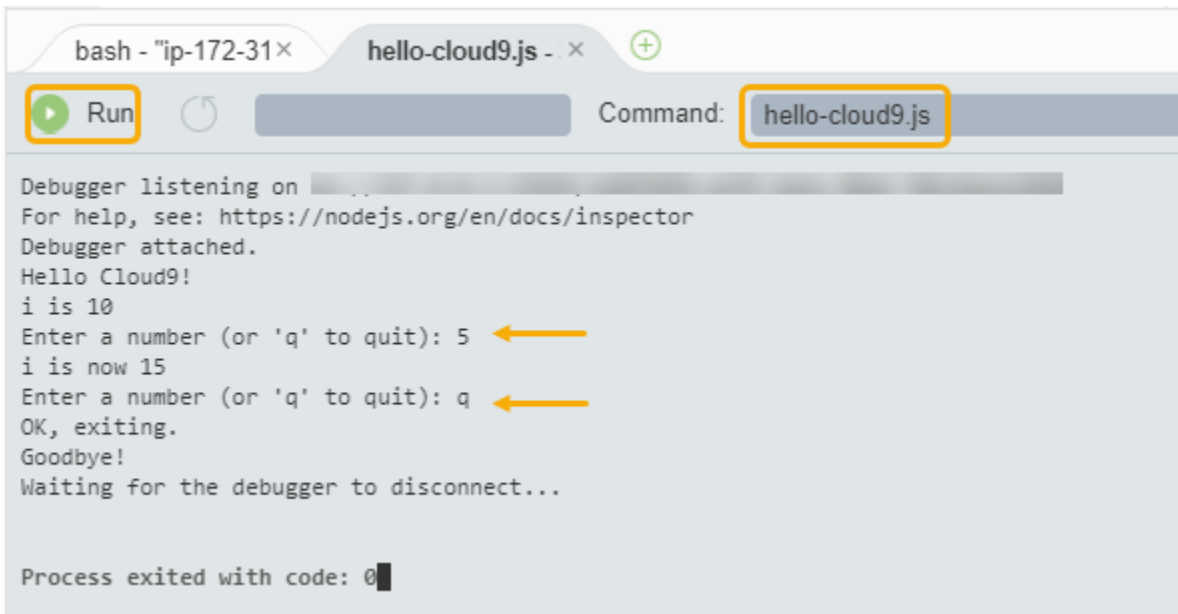
运行代码

接下来, 您可以运行代码了。

根据您正在使用的编程语言, 可能有多种方式可以运行代码。本教程使用 JavaScript, 您可以使用终端窗口或“运行配置”窗口来运行它。

使用“运行配置”窗口运行代码

1. 在菜单栏上依次选择运行、运行配置和新建运行配置。
2. 在新的 Run Configuration (运行配置) 窗口 (IDE 底部的一个选项卡) 中, 在 Command (命令) 字段中输入 hello-cloud9.js, 然后选择 Run (运行)。
3. 请确保 Run Configuration (运行配置) 提示符处于活动状态, 然后在提示符处输入一个数字来与应用程序交互。
4. 在 Run Configuration (运行配置) 窗口查看代码的输出内容。其内容类似于以下所示。



```
bash - "ip-172-31" x hello-cloud9.js - x +
Run Command: hello-cloud9.js
Debugger listening on [redacted]
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Waiting for the debugger to disconnect...

Process exited with code: 0
```

使用终端窗口以运行代码

1. 转到您早前使用的终端窗口（或打开一个新的窗口）。
2. 在终端窗口中，在终端提示符处输入 `ls`，并核实您的代码文件在文件列表中。
3. 在提示符处输入 `node hello-cloud9.js`，启动应用程序。
4. 在提示符处输入一个数字，与应用程序交互。
5. 在您的终端窗口查看代码的输出内容。其内容类似于以下所示。

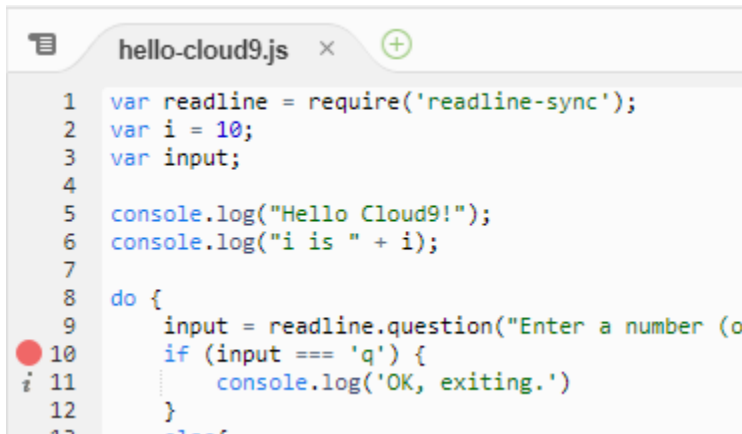


```
node - "ip-172-31" x hello-cloud9.js - ! x +
Admin:~/environment $ node hello-cloud9.js
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Admin:~/environment $
```

调试代码

最后，您可以通过使用 Debugger (调试程序) 窗口调试您的代码。

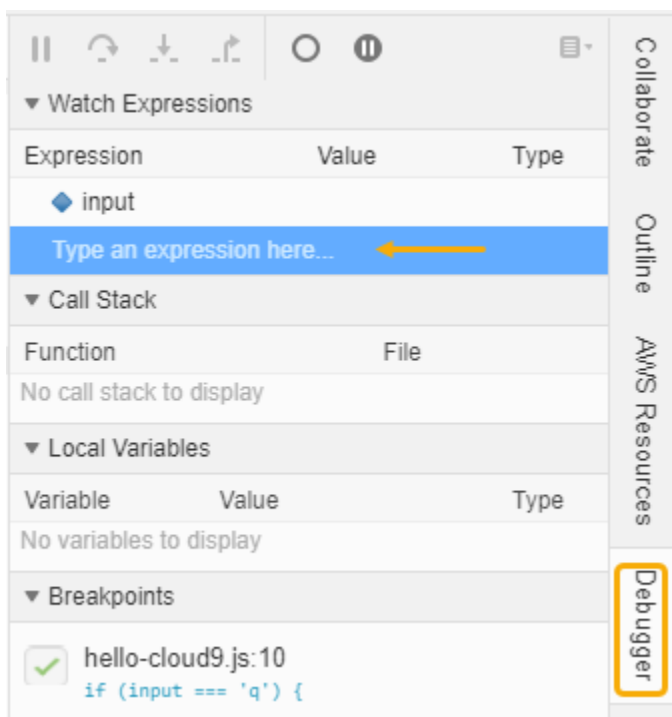
1. 通过选择第 10 行旁边的空白，将断点添加到代码第 10 行 (`if (input === 'q')`) 处。该行号旁边显示一个红色圆圈，如下所示。



```
1 var readline = require('readline-sync');
2 var i = 10;
3 var input;
4
5 console.log("Hello Cloud9!");
6 console.log("i is " + i);
7
8 do {
9     input = readline.question("Enter a number (o
10    if (input === 'q') {
11        console.log('OK, exiting.')
12    }
13
```

2. 通过选择 IDE 右侧的 Debugger (调试程序) 按钮，打开 Debugger (调试程序) 窗口。也可以在菜单栏上依次选择 Window 和 Debugger (调试程序)。

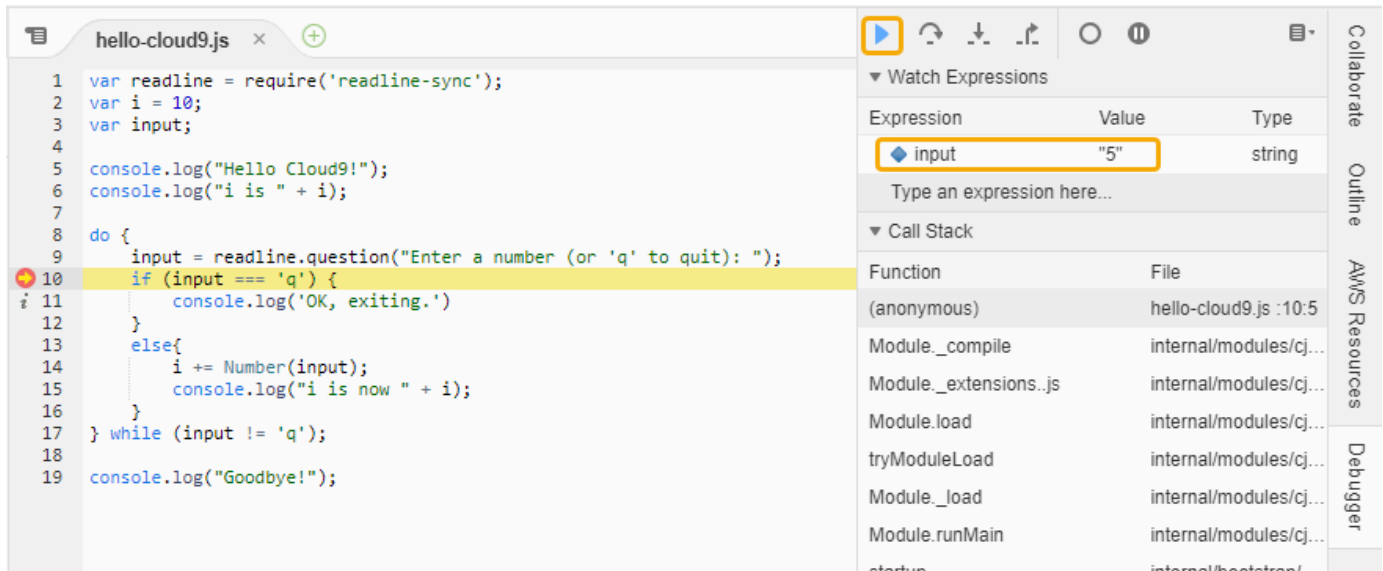
然后，通过在 Debugger (调试程序) 窗口的 Watch Expressions (监视表达式) 部分中选择 Type an expression here (在此处键入表达式)，对 `input` 变量进行监视。



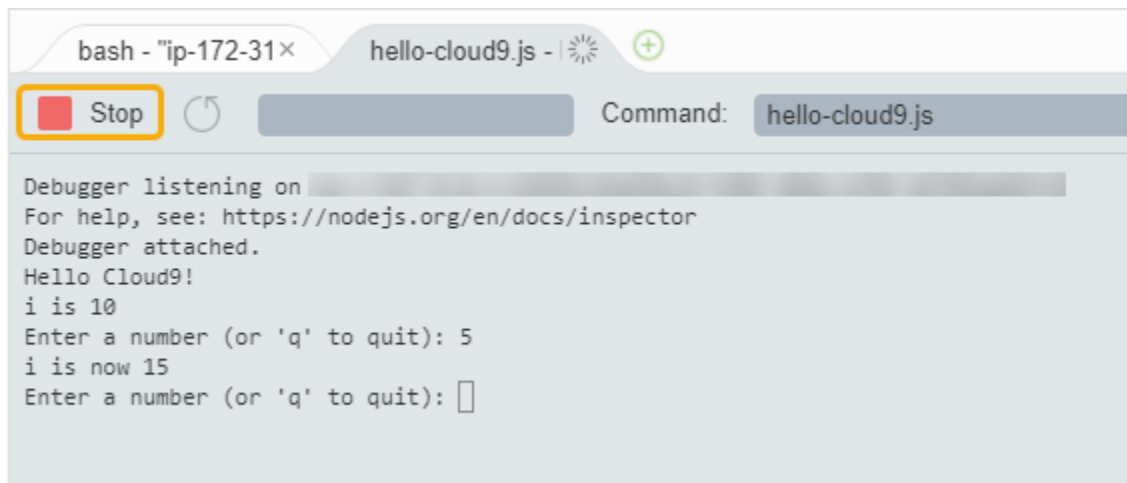
3. 转到您之前使用的 Run Configuration (运行配置) 窗口以运行代码。选择运行。

或者，您可以打开一个新的 Run Configuration (运行配置) 窗口并开始运行代码。要执行此操作，请从菜单栏中依次选择 Run (运行)、Run With (运行方式)、Node.js。

- 在 Run Configuration (运行配置) 提示符下输入一个数字，看到代码在第 10 行暂停。Debugger (调试程序) 窗口显示您在 Watch Expressions (监视表达式) 中输入的值。



- 在 Debugger (调试程序) 窗口中，选择 Resume (恢复)。这是上一张屏幕截图中突出显示的蓝色箭头图标。
- 在 Run Configuration (运行配置) 窗口中选择 Stop (停止) 以停止调试程序。



下一个步骤

[第 3 步：清除](#)

第 3 步：清除

(上一步：[步骤 2：IDE 基本导览](#))

为防止一直对您的 AWS 账户收取有关本教程的费用，您应删除环境。

Warning

无法撤消删除环境的操作。

使用 AWS CLI 删除环境

1. 运行 AWS Cloud9 `delete-environment` 命令，指定要删除的环境的 ID。

```
aws cloud9 delete-environment --region MY-REGION --environment-id
12a34567b8cd9012345ef67abcd890e1
```

在上述命令中，将 `MY-REGION` 替换为已经创建了环境的 AWS 区域，并且将 `12a34567b8cd9012345ef67abcd890e1` 替换为要删除的环境的 ID。

如果您在创建该环境时没有保存 ID，可以使用 AWS Cloud9 控制台找到 ID。选择控制台中环境的名称，然后查找环境 ARN 的最后一部分。

2. 如果您已为本教程创建 Amazon VPC，但不再需要它，请在 <https://console.aws.amazon.com/vpc> 网址上使用 Amazon VPC 控制台删除 VPC。

下一个步骤

[相关信息](#)

相关信息

以下是 [教程：Hello AWS Cloud9 \(CLI\)](#) 的更多信息。

- 在创建 EC2 环境时，该环境在默认情况下不包含任何示例代码。要创建环境以及示例代码，请参阅以下任一主题：
 - [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 Amazon Lightsail 实例](#)
 - [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 AWS CodeStar 项目](#)
- 在创建 AWS Cloud9 开发环境时，您将被引导到 AWS Cloud9 来创建一个 Amazon EC2 实例。AWS Cloud9 已创建了实例，然后将环境与其连接。您也可以使用现有的云计算实例或自己的服务器，这称为 SSH 环境。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。

可选的后续步骤

探索下列任何一个或全部主题，进一步熟悉 AWS Cloud9。

Task	请参阅本主题
了解您使用环境可以完成的的任务的更多信息。	在环境中工作 AWS Cloud9
尝试其他计算机语言。	AWS Cloud9 的教程
了解有关 AWS Cloud9 IDE 的更多信息。	使用 IDE 中的 了解 AWS Cloud9 IDE
利用文本聊天支持实时邀请其他用户使用新环境。	在 AWS Cloud9 中使用共享环境
创建 SSH 环境。这些是使用您创建的云计算实例或服务器（而不是 AWS Cloud9 为您创建的 Amazon EC2 实例）的环境。	在中创建环境 AWS Cloud9 和 SSH 环境主机要求
使用 AWS Toolkit 在 AWS Lambda 函数和无服务器应用程序中创建、运行和调试代码。	通过 AWS Toolkit 使用 AWS Lambda 函数
结合 Amazon Lightsail 使用 AWS Cloud9。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例
将 AWS Cloud9 与 AWS CodeStar 结合使用。	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CodeStar 项目
将 AWS Cloud9 与 AWS CodePipeline 结合使用。	在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline
AWS Cloud9与AWS CLI、 AWS云开发套件 (AWSCDK) AWS CodeCommit 、 亚马逊 DynamoDB 以及 Node.js GitHub 、 Python 或其他编程语言一起使用。 AWS CloudShell	AWS Cloud9 的教程
在中使用智能机器人应用程序的代码AWS RoboMaker。	AWS Cloud9在《 AWS RoboMaker 开发者指南 》中@@ 使用 开发工具

要从社区中获取 AWS Cloud9 帮助，请参阅 [AWS Cloud9 开发论坛](#)。（当您进入此论坛时，AWS 可能会要求您登录。）

要从 AWS 直接获得 AWS Cloud9 帮助，请参阅 [AWS Support](#) 页中的支持选项。

在环境中工作 AWS Cloud9

开发环境是存储项目文件和运行工具来开发应用程序的地方。AWS Cloud9

AWS Cloud9 提供两种类型的开发环境：EC2 环境和 SSH 环境。为了理解开发环境之间的主要相似和不同之处，请参阅 [AWS Cloud9 中 EC2 环境与 SSH 环境的比较](#)。

阅读以下一个或多个主题 AWS Cloud9，学习如何在环境中工作。

主题

- [在中创建环境 AWS Cloud9](#)
- [使用 AWS Systems Manager 访问非入口 EC2 实例](#)
- [在 AWS Cloud9 中打开环境](#)
- [从 AWS Cloud9 中的环境调用 AWS 服务](#)
- [在 AWS Cloud9 中更改环境设置](#)
- [在 AWS Cloud9 中使用共享环境](#)
- [移动环境并对 Amazon EBS 卷调整大小或加密](#)
- [在 AWS Cloud9 中删除环境](#)

在中创建环境 AWS Cloud9

要创建 AWS Cloud9 开发环境，请根据您的计划使用方式按照提供的步骤之一进行操作 AWS Cloud9。

如果您不确定选择什么，我们建议 [创建 EC2 环境](#)。

要进行快速设置，请创建 EC2 环境。AWS Cloud9 会在您的中自动创建和设置一个新的 Amazon EC2 实例 AWS 账户。AWS Cloud9 还会自动为您将新实例连接到环境。

为了理解开发环境之间的主要相似和不同之处，请参阅 [AWS Cloud9 中 EC2 环境与 SSH 环境的比较](#)。

源代码提供者	开发环境主机提供者	相关步骤
您	AWS Cloud9	创建 EC2 环境
您	您	创建 SSH 环境
Amazon Lightsail 或您	您，通过使用 Lightsail	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例
AWS CodeStar 或您	AWS Cloud9 (使用 AWS CodeStar)	在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CodeStar 项目
您 (使用 AWS CodePipeline)	AWS Cloud9 或者你	创建 EC2 或 SSH 环境，以及在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline
您，通过使用 AWS CodeCommit	AWS Cloud9 或者你	适用于 AWS Cloud9 的 AWS CodeCommit 教程
您 (使用 GitHub)	AWS Cloud9 或者你	创建 EC2 或 SSH 环境，然后使用 Git 面板界面

主题

- [创建 EC2 环境](#)
- [创建 SSH 环境](#)

创建 EC2 环境

在此过程中，AWS Cloud9 创建一个 EC2 环境和一个新的 Amazon EC2 实例，并将该环境连接到该实例。AWS Cloud9 管理此实例的生命周期，包括根据需要启动、停止和重启实例。如果您删除此环境，则 AWS Cloud9 会自动终止此实例。

您可以在 [AWS Cloud9 控制台](#) 中或使用 [代码](#) 创建 AWS Cloud9 EC2 开发环境。

Note

完成此过程可能会导致向您收费 AWS 账户。这包括可能对 Amazon EC2 收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

Warning

与 AWS Cloud9 AWS Control Tower 主动控制 [CT.EC2.PR.8](#) 存在兼容性问题。如果启用此控件，则无法在 AWS Cloud9 中创建 EC2 环境。有关此问题的更多信息，请参阅 [故障排除 AWS Cloud9](#)。

先决条件

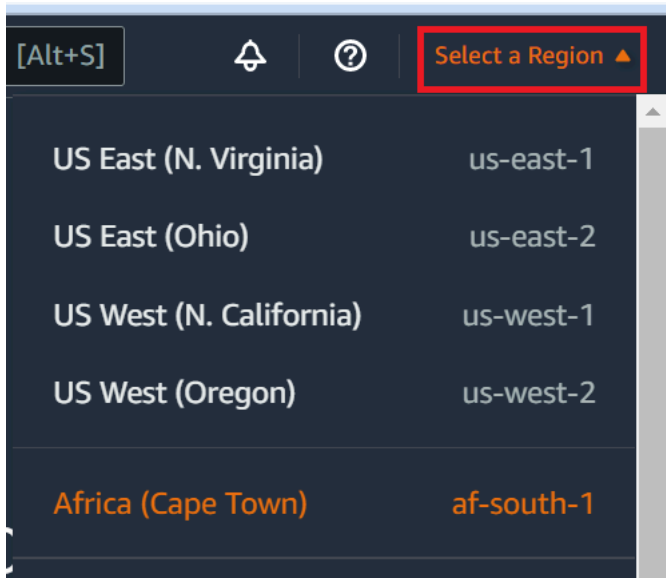
完成中的步骤，[设置 AWS Cloud9](#) 以便您可以登录 AWS Cloud9 控制台并创建环境。

使用控制台创建 EC2 环境

1. 登录 AWS Cloud9 控制台：

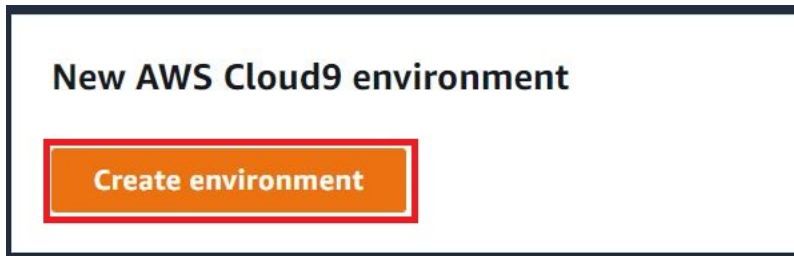
- 如果你是唯一一个使用你的用户，AWS 账户 或者你是 IAM 用户 AWS 账户，请访问 <https://console.aws.amazon.com/cloud9/>。
- 如果您的组织使用 AWS IAM Identity Center，请向 AWS 账户 管理员询问登录说明。
- 如果您是课堂中的学生，请联系您的导师以获取登录说明。

2. 登录 AWS Cloud9 控制台后，在顶部导航栏中选择一个 AWS 区域 来创建环境。有关可用列表 AWS 区域，请参阅 [AWS Cloud9](#) 中的 AWS 一般参考。



3. 在显示的位置之一选择大型 Create environment (创建环境) 按钮。

如果您还没有 AWS Cloud9 环境，则该按钮会显示在欢迎页面上。



如果您已经有 AWS Cloud9 环境，则按钮如下所示。




4. 在 Create environment (创建环境) 页面上，在 Name (名称) 中输入环境的名称。
5. 要为您的环境添加描述，请在 Description (描述) 字段中输入内容。
6. 在 Environment type (环境类型) 中，选择 Create EC2 instance (创建 EC2 实例) 以创建 Amazon EC2 环境。
 - New EC2 instance (新 EC2 实例) – 启动 AWS Cloud9 可以直接通过 SSH 连接的新 Amazon EC2 实例。您可以使用 Systems Manager 与新的 Amazon EC2 实例进行交互，有关更多信息，请参阅[使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

- Existing compute (现有计算) – 启动需要 SSH 登录详细信息的现有 Amazon EC2 实例，该 Amazon EC2 实例必须具有入站安全组规则。
- 如果您选择 Existing compute (现有计算) 选项，则会自动创建服务角色。您可以在设置屏幕底部的备注中查看服务角色的名称。


 Note

对于使用现有计算的 Amazon EC2 实例创建的 AWS Cloud9 环境，自动关闭不可用。

 Warning

为环境创建 Amazon EC2 实例可能会对您的 AWS 账户收取 Amazon EC2 费用。使用 Systems Manager 管理与 EC2 实例的连接不会产生额外费用。

7. 对于 Instance type (实例类型)，根据所要执行的任务类型，选择带有您认为所需的 RAM 和 vCPU 数量的实例类型。

 Warning

选择具有更多 RAM 和 vCPU 的实例类型可能会导致您收取 AWS 账户 Amazon EC2 的额外费用。有关适合您工作负载的实例类型的信息，请参阅 [Amazon EC2 实例类型](#) 页面。

8. 对于平台，选择你想要的亚马逊 EC2 实例类型：亚马逊 Linux 2023、亚马逊 Linux 2 或 Ubuntu 22.04 LTS。AWS Cloud9 创建实例，然后将环境连接到该实例。

 Important

建议您为 EC2 环境选择 Amazon Linux 2023 选项。在提供安全、稳定和高性能的运行系统环境之外，Amazon Linux 2023 AMI 还包括到 2024 年的长期支持。

有关更多信息，请参阅 [AL2023 页](#)。

9. 为 Timeout (超时) 选择时间段。此选项决定 AWS Cloud9 在处于非活动状态多长时间之后进入休眠。当所有连接到 IDE 的环境的 Web 浏览器实例都关闭后，将 AWS Cloud9 等待指定的时间，然后关闭该环境的 Amazon EC2 实例。

⚠ Warning

选择较长的时间段可能会导致在您的 AWS 账户中产生更多费用。

10. 在 Network settings (网络设置) 面板上，从以下两个选项中选择如何访问您的环境：
 - AWS Systems Manager (SSM)-此方法使用 SSM 访问环境，无需打开入站端口。
 - Secure Shell (SSH) – 此方法使用 SSH 访问环境，需要打开入站端口。
11. 选择 VPC 设置以显示您的环境的 Amazon Virtual Private Cloud 和子网。AWS Cloud9 使用亚马逊虚拟私有云 (亚马逊 VPC) 与新创建的亚马逊 EC2 实例通信。在本教程中，我们不建议您更改预先选择的默认设置。使用默认设置时，AWS Cloud9 尝试使用默认 VPC，其单个子网与新环境相同，区域 AWS 账户 与新环境相同。根据 Amazon VPC 的设置方式，请按照以下一系列说明之一进行操作。

如果您不确定选择什么，我们建议您向前跳到此过程中的下一步。

如果您跳过网络设置 (高级) 并保留预先选择的默认设置，则会 AWS Cloud9 尝试使用带有单个子网的默认 VPC。AWS Cloud9 根据您选择的实例类型选择子网。它们与新环境位于同一个 AWS 账户和 AWS 区域。

⚠ Important

如果您选择 Existing Compute (现有计算) 作为您的环境类型，您可以将实例启动到公有子网或私有子网中。

- 公有子网：将互联网网关附加到子网，以允许实例 SSM 代理与 Systems Manager 通信。
- 私有子网：创建 NAT 网关以使实例能够与互联网和其他 AWS 服务通信。

目前，您无法使用[AWS 托管临时凭证](#)允许 EC2 环境 AWS 服务代表 AWS 实体 (例如 IAM 用户) 进行访问。

有关配置子网的更多信息，请参阅 [AWS Cloud9 开发环境的 VPC 设置](#)。

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
否	—	—	—	<p>如果不存在 VPC，则创建一个。</p> <p>要在与新环境相同 AWS 账户和区域内创建 VPC，请选择创建新 VPC，然后按照屏幕上的说明进行操作。有关更多信息，请参阅 创建 VPC 以及其他 VPC 资源。</p> <p>要在与新环境 AWS 账户不同的环境中创建 VPC，请参阅 Amazon VPC 用户指南中的使用共享 VPC。</p>


AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	是	是	是	<p>跳至本过程中的下一步。</p> <p>当您跳过网络设置（高级）并且不更改预先选择的默认设置时，会 AWS Cloud9 尝试使用默认 VPC，其单个子网位于与新环境相同的账户和区域。</p>

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	是	是	不支持	<p>如果默认 VPC 有多个子网，请展开 Network settings (advanced) (网络设置 (高级))。对于 Subnet (子网)，选择希望 AWS Cloud9 在预先选择的默认 VPC 中使用的子网。</p> <p>如果默认 VPC 没有子网，请创建一个子网。为此，请选择 Create new subnet (新建子网)，然后按照屏幕上的说明进行操作。有关更多信息，请参阅 为创建子网 AWS Cloud9。</p>

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	是	否	是	展开 Network settings (网络设置)。对于 Network (VPC) (网络 (VPC))，选择希望 AWS Cloud9 使用的 VPC。

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	是	否	否	<p>展开 Network settings (网络设置)。</p> <p>对于 Network (VPC) (网络 (VPC))，选择希望 AWS Cloud9 使用的 VPC。</p> <p>如果所选 VPC 有多个子网，请展开 Network settings (advanced) (网络设置 (高级))。对于子网，选择 AWS Cloud9 要在所选 VPC 中使用的子网。</p> <p>如果所选 VPC 没有子网，请创建一个子网。为此，请选择 Create new subnet (新建子网)，然后按照屏幕上的说明进行操作。有关更多信息，请参</p>

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
				阅 为创建子网 AWS Cloud9。
是	否	是	—	AWS Cloud9 不能在与新环境的账户不同的中使用默认 VPC。 AWS 账户 在此列表中选择不同的选项。

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	否	否	是	<p>展开 Network settings (网络设置)。</p> <p>对于 Network (VPC) (网络 (VPC))，选择希望 AWS Cloud9 使用的 VPC。</p> <div data-bbox="1273 829 1507 1480"><p> Note</p><p>VPC 必须与新环境位于同一个区域，即使 VPC 在不同的账户中也是如此。</p></div>

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
是	否	否	否	<p>展开 Network settings (网络设置)。</p> <p>对于 Network (VPC) (网络 (VPC))，选择希望 AWS Cloud9 使用的 VPC。</p> <p>对于 Subnet (子网)，请选择您希望 AWS Cloud9 在所选 VPC 中使用的子网。</p> <p>如果所选的 VPC 没有子网，要在与新环境 AWS 账户不同的环境中为 VPC 创建子网，请参阅Amazon VPC 用户指南中的使用共享 VPC。</p> <div data-bbox="1273 1654 1507 1837" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>VPC 和子网必</p> </div>

AWS 账户 他们可以访问亚马逊 VPC 吗？	那个 VPC 是否 AWS 账户 与新环境位于同一区域中？	该 VPC 是否为其 AWS 账户默认的 VPC？	该 VPC 是否包含单个子网？	按照以下说明进行操作
				<p>须与新环境位于同一个区域，即使 VPC 和子网在不同的账户中也是如此。</p>

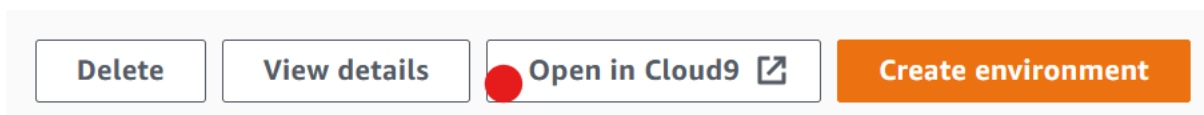
有关这些选择的更多信息，请参阅 [AWS Cloud9 开发环境的 VPC 设置](#)。

- 通过为每个标签提供 Key (键) 和 Value (值) ，添加最多 50 个标签。要执行此操作，请选择 Add new tag (添加新标签) 。这些标签作为资源标签附加到 AWS Cloud9 环境，并传播到以下基础资源：AWS CloudFormation 堆栈、Amazon EC2 实例和 Amazon EC2 安全组。要了解有关标签的更多信息，请参阅 [IAM 用户指南](#) 中的 [使用 AWS 资源标签控制访问权限](#) 以及本指南中的 [高级信息](#)。

Warning

如果您在创建标签后更新这些标签，则更改不会传播到底层资源。有关更多信息，请参阅 [标签高级信息](#) 中的 [将标签更新传播到底层资源](#)。

- 选择 Create (创建) 以创建您的环境，然后系统会将您重定向到主页。如果成功创建了帐户，则 AWS Cloud9 控制台顶部会出现一个绿色的闪光条。您可以选择新环境并选择 Open in Cloud9 (在 Cloud9 中打开) 以启动 IDE。



如果帐户创建失败，则 AWS Cloud9 控制台顶部会出现一个红色的闪光条。您可能会因为 Web 浏览器、AWS 访问权限、实例或关联的网络的问题而未能创建账户。您可以在 [AWS Cloud9 故障排除](#) 部分找到有关可能的修复方法的信息。

Note

AWS Cloud9 同时支持 imdsv1 和 imdsv2。我们建议采用 IMDSv2，因为与 IMDSv1 相比，它提供了增强型安全级别。有关 IMDSv2 优势的更多信息，请参阅 [AWS 安全博客](#)。关于如何从 IMDSv1 过渡到 IMDSv2 的信息，请参阅《适用于 Linux 实例的 Amazon EC2 用户指南》中的 [过渡到使用实例元数据服务版本 2](#)。

Note

如果您的环境使用代理访问互联网，则必须向其提供代理详细信息，AWS Cloud9 以便它可以安装依赖关系。有关更多信息，请参阅 [无法安装依赖项](#)。

使用代码创建环境

要使用代码在中创建 EC2 环境 AWS Cloud9，请调用 AWS Cloud9 创建 EC2 环境操作，如下所示。

AWS CLI	create-environment-ec2
AWS SDK for C++	CreateEnvironmentec2Request , ec2Result CreateEnvironment
AWS SDK for Go	CreateEnvironmentEC2 、 CreateEnvironmentec2Request 、 EC2 CreateEnvironment WithContext
AWS SDK for Java	CreateEnvironmentec2Request , ec2Result CreateEnvironment
AWS SDK for JavaScript	createEnvironmentEC2

AWS SDK for .NET	CreateEnvironmentec2Request、ec2Resps ons CreateEnvironment
AWS SDK for PHP	createEnvironmentEC2
AWS SDK for Python (Boto)	create_environment_ec2
AWS SDK for Ruby	create_environment_ec2
AWS Tools for Windows PowerShell	New-C9EnvironmentEC2
AWS Cloud9 API	CreateEnvironmentEC2

Note

如果您的环境使用代理访问互联网，则必须向其提供代理详细信息，AWS Cloud9 以便它可以安装依赖关系。有关更多信息，请参阅 [无法安装依赖项](#)。

创建 SSH 环境

您可以使用 AWS Cloud9 控制台创建 AWS Cloud9 SSH 开发环境。您无法使用 CLI 创建 SSH 环境。

先决条件

- 首先确保您已完成 [设置 AWS Cloud9](#) 中的步骤。这样，您便可以登录到 AWS Cloud9 控制台并创建环境。
- 确定要连接到环境的现有云计算实例（例如，您的中的 Amazon EC2 实例 AWS 账户）或您 AWS Cloud9 自己的服务器。
- 确保现有实例或您自己的服务器满足所有 [SSH 托管要求](#)。这包括安装特定版本的 Python、Node.js 和其他组件；对于您希望的登录后 AWS Cloud9 开始目录设置特定权限；以及设置任何关联的 Amazon Virtual Private Cloud。

创建 SSH 环境

1. 确保您已完成上述先决条件。

2. 使用 SSH 客户端连接到现有实例或您自己的服务器（如果您尚未连接）。这确保了您可以向实例或服务器添加必要的公有 SSH 密钥值。这些内容将在此过程的稍后介绍。

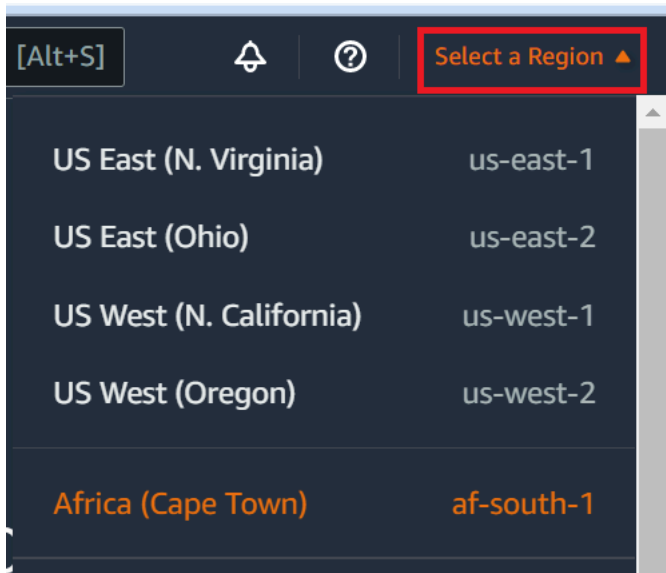
Note

要连接到现有的 AWS Cloud 计算实例，请参阅以下一项或多项资源：

- 对于亚马逊 EC2，请参阅亚马逊 EC2 用户指南中的[连接到您的 Linux 实例](#)。
- 有关 Amazon Lightsail，请参阅[Amazon Lightsail 文档](#)中的[连接到基于 Linux/Unix 的 Lightsail 实例](#)。
- 有关信息 AWS Elastic Beanstalk，请参阅《AWS Elastic Beanstalk 开发人员指南》中的[列出和连接到服务器实例](#)。
- 有关信息 AWS OpsWorks，请参阅《AWS OpsWorks 用户指南》中的[“使用 SSH 登录 Linux 实例”](#)。
- 有关其他 AWS 服务，请参阅该特定服务的文档。

要连接到您自己的服务器，请使用 SSH。已经在 macOS 和 Linux 操作系统上安装了 SSH。要在 Windows 上使用 SSH 连接到服务器，您必须安装 [PuTTY](#)。

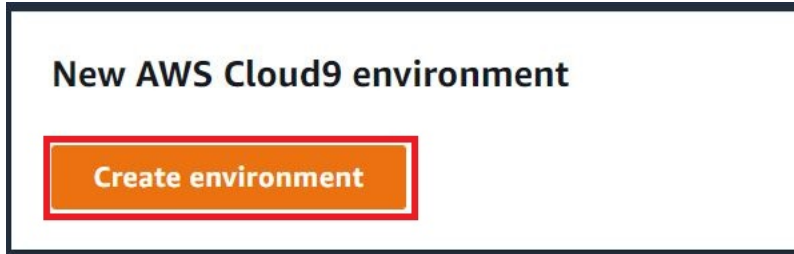
3. 登录 AWS Cloud9 主机，[网址为 https://console.aws.amazon.com/cloud9/](https://console.aws.amazon.com/cloud9/)。
4. 登录 AWS Cloud9 控制台后，在顶部导航栏中选择一个 AWS 区域 来创建环境。有关可用列表 AWS 区域，请参阅[AWS Cloud9](#)中的 AWS 一般参考。



5. 如果您是第一次创建开发环境，则将显示欢迎页面。在新 AWS Cloud9 环境面板中，选择创建环境。

如果您之前创建过开发环境，您也可以展开屏幕左侧的窗格。选择 Your environments (您的环境) ，然后选择 Create environment (创建环境) 。

在欢迎页面：



或在 Your environment (您的环境) 页面内：



6. 在 Create environment (创建环境) 页面上，输入环境的名称。
7. 在 Description (描述) 中输入对环境的描述。在本教程中，请使用 This environment is for the AWS Cloud9 tutorial.
8. 对于 Environment type (环境类型) ，请从以下选项中选择 Existing Compute (现有计算) ：
 - 新的 EC2 实例 — 启动 AWS Cloud9 可以直接通过 SSH 或 SSM 连接的 Amazon EC2 实例。
 - 现有计算-启动一个需要 SSH 登录详细信息以及端口 22 才能打开的现有 Amazon EC2 实例。AWS Cloud9 通过连接到实例[AWS Systems Manager](#)。
 - 如果您选择 Existing compute (现有计算) 选项，则会自动创建服务角色。您可以在界面下方的 Systems Manager 访问的服务角色和实例配置文件部分查看服务角色名称。有关更多信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

⚠ Warning

为您的环境创建 EC2 实例可能会导致您 AWS 账户收取 Amazon EC2 费用。使用 Systems Manager 管理与 EC2 实例的连接不会产生额外费用。

⚠ Warning

AWS Cloud9 使用 SSH 公钥安全地连接到您的服务器。要建立安全连接，请将我们的公有密钥添加到您的 `~/.ssh/authorized_keys` 文件中，并在以下步骤中提供您的登录凭证。选择 Copy key to clipboard (将密钥复制到剪贴板) 以复制 SSH 密钥，或选择 View public SSH key (查看 SSH 公有密钥) 以查看密钥。

9. 在 Existing compute (现有计算) 面板上，对于 User (用户)，请输入您在此过程中之前连接到实例或服务器时使用的登录名。例如，对于 AWS Cloud 计算实例，该登录名可能为 `ec2-user`、`ubuntu` 或 `root`。

ℹ Note

我们建议将登录名与实例或服务器上的管理权限或管理员用户关联。更具体地说，我们建议此登录名应在实例或服务器上拥有 Node.js 安装。要进行此检查，请在实例或服务器的终端上运行命令 `ls -l $(which node)` (如果您使用的是 `nvm`，请运行 `ls -l $(nvm which node)`)。此命令会显示 Node.js 安装的拥有者名称。它还会显示安装的权限、组名称和位置。

10. 对于 Host (主机)，请输入公有 IP 地址 (首选) 或者实例或服务器的主机名。
11. 对于端口，输入您 AWS Cloud9 要用来尝试连接到实例或服务器的端口。您也可保留默认端口。
12. 选择 Additional details - optional (其他详细信息 - 可选) 以显示环境路径、node.js 二进制文件路径和 SSH 跳转主机信息。
13. 在环境路径中，输入您要 AWS Cloud9 从中启动的实例或服务器上的目录路径。您之前在此步骤的先决条件中确定了这一点。如果将此项留空，AWS Cloud9 将使用实例或服务器在登录之后通常开始使用的目录。这通常为主目录或默认目录。
14. 对于 Path to Node.js binary path (Node.js 二进制文件路径)，请输入路径信息以指定实例或服务器上指向 Node.js 二进制文件的路径。要获取路径，您可以在实例或服务器上运行命令 `which node` (在使用 `nvm` 时为 `nvm which node`)。例如，路径可能为 `/usr/bin/node`。如果将此项留空，AWS Cloud9 在尝试连接时，将尝试猜测 Node.js 二进制文件所在的位置。
15. 对于 SSH jump host (SSH 跳转主机)，输入有关实例或服务器使用的跳转主机的信息。使用格式 `USER_NAME@HOSTNAME:PORT_NUMBER` (例如 `ec2-user@ip-192-0-2-0:22`)。

跳转主机必须满足以下要求：

- 它必须可使用 SSH 通过公有互联网访问。
 - 它必须允许任意 IP 地址通过指定端口的入站访问。
 - 复制到现在实例或服务器上 `~/.ssh/authorized_keys` 文件中的公有 SSH 密钥值还必须复制到跳转主机上的 `~/.ssh/authorized_keys` 文件中。
 - 必须安装有 Netcat。
16. 通过为每个标签提供键和值，添加最多 50 个标签。要执行此操作，请选择 Add new tag (添加新标签)。这些标签作为资源标签附加到 AWS Cloud9 环境，并传播到以下基础资源：AWS CloudFormation 堆栈、Amazon EC2 实例和 Amazon EC2 安全组。要了解有关标签的更多信息，请参阅 [IAM 用户指南中的使用 AWS 资源标签控制访问权限](#)以及本指南中有关标签的[高级信息](#)。

⚠ Warning

如果您在创建标签后更新这些标签，则更改不会传播到底层资源。有关更多信息，请参阅[标签高级信息中的将标签更新传播到底层资源](#)。

17. 选择 Create (创建) 以创建您的环境，然后系统会将您重定向到主页。成功创建帐户后，AWS Cloud9 控制台顶部会出现一个绿色的闪光条。您可以选择新环境并选择 Open in Cloud9 (在 Cloud9 中打开) 以启动 IDE。

Delete

View details

Open in Cloud9 

Create environment

如果未能创建账户，则 AWS Cloud9 控制台顶部会出现一个红色的闪光条。由于您的网络浏览器、AWS 访问权限、实例或关联网络存在问题，您的账户可能无法创建。您可以在 [AWS Cloud9 故障排除](#)部分找到有关可能修复导致账户失败问题的信息。

i Note

如果您的环境使用代理访问互联网，则必须向其提供代理详细信息，AWS Cloud9 以便它可以安装依赖关系。有关更多信息，请参阅 [无法安装依赖项](#)。

使用 AWS Systems Manager 访问非入口 EC2 实例

为 EC2 环境创建的“非入口 EC2 实例”使 AWS Cloud9 能够连接到其 Amazon EC2 实例，而无需打开该实例上的任何入站端口。在使用[控制台](#)、[命令行界面](#)或[AWS CloudFormation 堆栈](#)创建 EC2 环境时，您可以选择非入口选项。

Important

使用 Systems Manager 会话管理器管理与 EC2 实例的连接不会产生额外费用。

在控制台的 Create environment (创建环境) 部分选择环境类型时，您可以选择需要入站连接的新 EC2 实例，也可以选择不需要以下各项的新的非入口 EC2 实例：

- [New EC2 instance](#) (新 EC2 实例) – 选择此设置时，实例的安全组具有允许传入的网络流量的规则。将传入的网络流量限制为[已获批连接 AWS Cloud9 的 IP 地址](#)。打开的入站端口使 AWS Cloud9 能够通过 SSH 连接到其实例。如果您使用 AWS Systems Manager 会话管理器，则可以通过 SSM 访问您的 Amazon EC2 实例，而无需打开入站端口 (无入口)。此方法仅适用于新 Amazon EC2 实例。有关更多信息，请参阅[在 EC2 环境中使用 Systems Manager 的益处](#)。
- [Existing compute](#) (现有计算) – 通过此设置，可以访问需要 SSH 登录详细信息的现有 Amazon EC2 实例，该实例必须具有对应的入站安全组规则。如果您选择此选项，则会自动创建服务角色。您可以在设置屏幕底部的备注中查看服务角色的名称。

如果使用 [AWS CLI](#) 创建环境，您可以在调用 create-environment-ec2 命令时设置 --connection-type CONNECT_SSM 选项来配置非入口 EC2 实例。有关创建所需的服务角色和实例配置文件的更多信息，请参阅[使用 AWS CLI 管理 Systems Manager 的实例配置文件](#)。

使用非入口 EC2 实例完成创建环境后，请确认以下事项：

- Systems Manager 会话管理器具有代表您对 EC2 实例执行操作的权限。有关更多信息，请参阅[管理 Systems Manager 权限](#)。
- AWS Cloud9 用户可以访问由会话管理器管理的实例。有关更多信息，请参阅[授予用户访问由会话管理器管理的实例的权限](#)。

在 EC2 环境中使用 Systems Manager 的益处

允许[会话管理器](#)来处理 AWS Cloud9 及其 EC2 实例之间的安全连接有两个重大益处：

- 无需为实例打开入站端口
- 将实例启动到公有子网或私有子网内的选项

No open inbound ports

AWS Cloud9 与其 EC2 实例之间的安全连接由[会话管理器](#)处理。会话管理器是一种完全托管式的 Systems Manager 功能，可启用 AWS Cloud9 到其 EC2 实例的连接，无需打开入站端口。

Important

使用 Systems Manager 进行非入口连接的选项目前仅在创建新的 EC2 环境时可用。

随着会话管理器会话的启动，将建立与目标实例的连接。建立连接后，环境现在可以通过 Systems Manager 服务与实例进行交互。Systems Manager 服务通过 Systems Manager Agent ([SSM Agent](#)) 与实例交互。

原定设置情况下，SSM Agent 安装在 EC2 环境使用的所有实例上。

Private/public subnets

当在 Network settings (advanced) 网络设置 (高级) 部分中为实例选择子网时，如果环境实例通过 Systems Manager 访问，则可以选择私有子网或公有子网。

▼ **Network settings (advanced)**

Network (VPC)
Launch your EC2 instance into an existing Amazon Virtual Private Cloud (VPC) or create a new one.

vpc- [dropdown] [refresh] [Create new VPC]

Subnet
Select the subnet in which the EC2 instance is created. For a private subnet, ensure it has internet connectivity by adding a NAT gateway. Public or private IP depends on the subnet (public or private).

No preference (default subnet in any Availability Zone) [dropdown] [refresh] [Create new subnet]

Temporary managed credentials can't be used in private subnets.

No tags associated with the resource.

[Add new tag]

You can add 50 more tags.

私有子网

对于私有子网，请确保实例仍然可以连接到 SSM 服务。此操作可以通过[在公有子网中设置 NAT 网关](#)或者为 [Systems Manager 配置 VPC 终端节点](#)完成。

使用 NAT 网关的优点在于它可以防止互联网启动到私有子网中的实例的连接。为您的环境的实例分配了一个私有 IP 地址，而不是公有 IP 地址。因此，NAT 网关将流量从实例转发到互联网或其他 AWS 服务，然后将响应发送回实例。

对于 VPC 选项，请为 Systems Manager 创建至少三个所需的接口端点：`com.amazonaws.region.ssm`、`com.amazonaws.region.ec2messages` 和 `com.amazonaws.region.ssmmessages`。有关更多信息，请参阅 AWS Systems Manager 用户指南中的[创建 Systems Manager 的 VPC 终端节点](#)。

⚠ Important

目前，如果您的环境的 EC2 实例启动到私有子网，则无法使用 [AWS 托管式临时凭证](#)来允许 EC2 环境代表 AWS 实体（例如 IAM 用户）访问 AWS 服务。

公有子网

如果您的开发环境使用 SSM 访问 EC2 实例，请确保该实例被启动到的公有子网分配了公有 IP 地址。为此，您可以指定自己的 IP 地址或启用公有 IP 地址的自动分配。有关修改自动分配 IP 设置所涉及的步骤，请参阅 Amazon VPC 用户指南中的[您的 VPC 中的 IP 地址](#)。

有关为环境实例配置私有子网和公有子网的更多信息，请参阅 [为创建子网 AWS Cloud9](#)。

管理 Systems Manager 权限

默认情况下，Systems Manager 没有在您的 EC2 实例上执行操作的权限。访问权限通过 AWS Identity and Access Management (IAM) 实例配置文件提供。（实例配置文件是一个容器，可在启动时将 IAM 角色信息传递给 EC2 实例。）

当您使用 AWS Cloud9 控制台创建非入口 EC2 实例时，服务角色 (AWSCloud9SSMAccessRole) 和 IAM 实例配置文件 (AWSCloud9SSMInstanceProfile) 会自动为您创建。（您可以在 IAM 管理控制台中查看 AWSCloud9SSMAccessRole。实例配置文件不会显示在 IAM 控制台中。）

Important

如果您首次使用 AWS CLI 创建非入口 EC2 环境，您必须明确定义所需的服务角色和实例配置文件。有关更多信息，请参阅[使用 AWS CLI 管理 Systems Manager 的实例配置文件](#)。

Important

如果您正在创建 AWS Cloud9 环境并使用附加了 AWSCloud9Administrator 或 AWSCloud9User 策略的 Amazon EC2 Systems Manager，则还必须附加具有特定 IAM 权限的自定义策略，请参阅[用于创建 SSM 环境的自定义 IAM 策略](#)。这是由于 AWSCloud9Administrator 和 AWSCloud9User 策略存在权限问题。

为了获得额外的安全保护，AWS Cloud9 服务相关角色 AWSServiceRoleforAWSCloud9 在其 AWSCloud9ServiceRolePolicy 策略中带有 PassRole 限制。在您将 IAM 角色传递到服务时，则允许该服务代入该角色并代表您执行操作。在这种情况下，PassRole 权限可确保 AWS Cloud9 只能传递 AWSCloud9SSMAccessRole 角色（及其权限）到 EC2 实例。这将在 EC2 实例上执行的操作限制为仅 AWS Cloud9 要求的操作。

Note

如果您不再需要使用 Systems Manager 来访问实例，您可以删除 AWSCloud9SSMAccessRole 服务角色。有关更多信息，请参阅 IAM 用户指南中的[删除角色或实例配置文件](#)。

使用 AWS CLI 管理 Systems Manager 的实例配置文件

您还可以使用 AWS CLI 创建非入口 EC2 环境。当您调用 `create-environment-ec2` 时，将 `--connection-type` 选项设置为 `CONNECT_SSM`。


如果您使用此选项，则 AWSCloud9SSMAccessRole 服务角色和 AWSCloud9SSMInstanceProfile 不会自动创建。因此，要创建所需的服务配置文件和实例配置文件，请执行下列操作之一：

- 使用控制台创建 EC2 环境，然后 AWSCloud9SSMAccessRole 服务角色和 AWSCloud9SSMInstanceProfile 之后将自动创建。创建后，服务角色和实例配置文件可用于其他使用 AWS CLI 创建的 EC2 环境。
- 运行以下 AWS CLI 命令以创建服务角色和实例配置文件。

```
aws iam create-role --role-name AWSCloud9SSMAccessRole --path /service-role/ --assume-role-policy-document '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service": ["ec2.amazonaws.com", "cloud9.amazonaws.com"]}, "Action": "sts:AssumeRole"}]}'
aws iam attach-role-policy --role-name AWSCloud9SSMAccessRole --policy-arn arn:aws:iam::aws:policy/AWSCloud9SSMInstanceProfile
aws iam create-instance-profile --instance-profile-name AWSCloud9SSMInstanceProfile --path /cloud9/
aws iam add-role-to-instance-profile --instance-profile-name AWSCloud9SSMInstanceProfile --role-name AWSCloud9SSMAccessRole
```

授予用户访问由会话管理器管理的实例的权限

要打开中通过 Systems Manager 连接到 EC2 实例的 AWS Cloud9 环境，用户必须具有 `StartSession`，API 操作的权限。此操作为会话管理器会话启动与托管 EC2 实例的连接。您可以通过使用特定于 AWS Cloud9 的托管式策略（推荐）或编辑 IAM 策略并添加必要的权限来向用户授予访问权限。

方法	描述
使用特定于 AWS Cloud9 的托管式策略	<p>我们建议使用 AWS 托管式策略，以允许用户访问由 Systems Manager 管理的 EC2 实例。托管式策略为标准 AWS Cloud9 使用案例提供一系列权限，并且可以轻松地附到 IAM 实体上。</p> <p>所有托管式策略还包括运行 StartSession API 操作的权限。以下是特定于 AWS Cloud9 的托管式策略：</p> <ul style="list-style-type: none">• AWSCloud9Administrator (arn:aws:iam::aws:policy/AWSCloud9Administrator)• AWSCloud9User (arn:aws:iam::aws:policy/AWSCloud9User)• AWSCloud9EnvironmentMember (arn:aws:iam::aws:policy/AWSCloud9EnvironmentMember) <div data-bbox="829 1115 1507 1671" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important</p><p>如果您正在创建 AWS Cloud9 环境并使用附加了 AWSCloud9Administrator 或 AWSCloud9User 策略的 Amazon EC2 Systems Manager，则还必须附加具有特定 IAM 权限的自定义策略，请参阅用于创建 SSM 环境的自定义 IAM 策略。这是由于 AWSCloud9Administrator 和 AWSCloud9User 策略存在权限问题。</p></div> <p>有关更多信息，请参阅AWS 的托管策略 AWS Cloud9。</p>

方法	描述
编辑 IAM 策略并添加所需的策略语句	<p>要编辑现有策略，您可以添加对于 StartSession API 的权限。要使用 AWS Management Console 或 AWS CLI 编辑策略，请参阅《IAM 用户指南》中的编辑 IAM policy所提供的说明。</p> <p>编辑策略时，添加允许要运行的 ssm:startSession API 操作的 policy statement (请参阅下面的内容)。</p>

您可以使用以下权限运行 StartSession API 操作。ssm:resourceTag 条件键指定可以为任何实例启动会话管理器会话 (Resource: arn:aws:ec2:*:*:instance/*)，条件是该实例为 AWS Cloud9 EC2 开发环境 (aws:cloud9:environment)。

Note

以下托管式策略还包括这些策略语句：AWSCloud9Administrator、AWSCloud9User 和 AWSCloud9EnvironmentMember。

```
{
  "Effect": "Allow",
  "Action": "ssm:StartSession",
  "Resource": "arn:aws:ec2:*:*:instance/*",
  "Condition": {
    "StringLike": {
      "ssm:resourceTag/aws:cloud9:environment": "*"
    },
    "StringEquals": {
      "aws:CalledViaFirst": "cloud9.amazonaws.com"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
```

```
    "Resource": [  
      "arn:aws:ssm:*:*:document/*"  
    ]  
  }  
}
```

使用 AWS CloudFormation 创建非入口 EC2 环境

使用 [AWS CloudFormation 模板](#) 定义非入口 Amazon EC2 开发环境，请在创建堆栈之前执行以下操作：

1. 创建 `AWSCloud9SSMAccessRole` 服务角色和 `AWSCloud9SSMInstanceProfile` 实例配置文件。有关更多信息，请参阅 [使用 AWS CloudFormation 模板创建服务角色和实例配置文件](#)。
2. 更新 IAM 实体调用 AWS CloudFormation 的策略。这样，此实体可以启动连接到 EC2 实例的会话管理器会话。有关更多信息，请参阅 [将 Systems Manager 权限添加到 IAM 策略](#)。

使用 AWS CloudFormation 模板创建服务角色和实例配置文件

您需要创建服务角色 `AWSCloud9SSMAccessRole` 和实例配置文件 `AWSCloud9SSMInstanceProfile` 以使 Systems Manager 能够管理支持您的开发环境的 EC2 实例。

如果您之前已通过创建非入口 EC2 环境 [with the console](#) 或 [运行 AWS CLI 命令](#) 来创建 `AWSCloud9SSMAccessRole` 和 `AWSCloud9SSMInstanceProfile`，则服务角色和实例配置文件已可用。

Note

假设您尝试为非入口 EC2 环境创建 AWS CloudFormation 堆栈，但没有首先创建所需的服务角色和实例配置文件。那么，不会创建堆栈，并显示以下错误消息：
账户中不存在实例配置文件 `AWSCloud9SSMInstanceProfile`。

首次使用 AWS CloudFormation 创建非入口 EC2 环境时，您可以在模板中将 `AWSCloud9SSMAccessRole` 和 `AWSCloud9SSMInstanceProfile` 定义为 IAM 资源。

摘自示例模板的此内容显示了如何定义这些资源。AssumeRole 操作返回安全凭证，该凭证提供对 AWS Cloud9 环境及其 EC2 实例的访问权限。

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Resources:
  AWSCloud9SSMAccessRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - cloud9.amazonaws.com
                - ec2.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      Description: 'Service linked role for AWS Cloud9'
      Path: '/service-role/'
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AWSCloud9SSMInstanceProfile
      RoleName: 'AWSCloud9SSMAccessRole'

  AWSCloud9SSMInstanceProfile:
    Type: "AWS::IAM::InstanceProfile"
    Properties:
      InstanceProfileName: AWSCloud9SSMInstanceProfile
      Path: "/cloud9/"
      Roles:
        -
          Ref: AWSCloud9SSMAccessRole
```

将 Systems Manager 权限添加到 IAM 策略

在 [AWS CloudFormation 模板](#) 中 [定义服务角色和实例配置文件](#) 后，请确保创建堆栈的 IAM 实体具有启动会话管理器会话的权限。会话是使用会话管理器建立的与 EC2 实例的连接。

Note

如果您没有在为非入口 EC2 环境创建堆栈之前添加启动会话管理器会话的权限，则会返回 `AccessDeniedException` 错误。

通过调用 AWS CloudFormation，向 IAM 实体的策略添加以下权限。

```
{
  "Effect": "Allow",
  "Action": "ssm:StartSession",
  "Resource": "arn:aws:ec2:*:*:instance/*",
  "Condition": {
    "StringLike": {
      "ssm:resourceTag/aws:cloud9:environment": "*"
    },
    "StringEquals": {
      "aws:CalledViaFirst": "cloudformation.amazonaws.com"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": [
    "arn:aws:ssm:*:*:document/*"
  ]
}
```

为 Amazon S3 配置 VPC 终端节点以下载依赖项

如果您的 AWS Cloud9 环境的 EC2 实例无法访问互联网，您需要为指定的 Amazon S3 桶创建 VPC 端点。此存储桶包含使 IDE 保持最新状态所需的依赖项。

为 Amazon S3 设置 VPC 端点还涉及自定义访问策略。您希望访问策略仅允许访问包含要下载的依赖项的可信 S3 桶。

Note

您可以使用 AWS Management Console、AWS CLI 或 Amazon VPC API 来创建和配置 VPC 终端节点。以下过程说明如何使用控制台界面创建 VPC 端点。

为 Amazon S3 创建和配置 VPC 终端节点

1. 在 AWS Management Console，转至 Amazon VPC 的控制台页面。
2. 在导航窗格中，选择 Endpoints (端点)。

3. 在 Endpoints (端点) 页面中，选择 Create Endpoint (创建端点)。
4. 在 Create Endpoint (创建端点) 页面中，在搜索字段中输入“s3”，然后按 Return (返回) 以列出 Amazon S3 在当前 AWS 区域内可用的端点。
5. 从返回的 Amazon S3 端点列表中选择 Gateway (网关) 类型。
6. 接下来，选择包含您环境的 EC2 实例的 VPC。
7. 现在选择 VPC 的路由表。这样，关联的子网可以访问端点。您环境的 EC2 实例位于其中一个子网中。
8. 在 Policy (策略) 部分，选择 Custom (自定义) 选项，并将标准策略替换为以下策略。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "Access-to-C9-bucket-only",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::{bucket_name}/content/dependencies/*"
    }
  ]
}
```

对于 Resource 元素，将 {bucket_name} 替换为您的 AWS 区域内可用的桶的实际名称。例如，如果您正在欧洲 (爱尔兰) 区域使用 AWS Cloud9，您可指定以下命令："Resource": "arn:aws:s3:::static-eu-west-1-prod-static-hld3vzaf7c4h/content/dependencies/。

下表列出了 AWS Cloud9 可用的 AWS 区域的桶名称。

AWS Cloud9 区域的 Amazon S3 存储桶

AWS 区域	存储桶名称
美国东部 (俄亥俄州)	static-us-east-2-prod-static-1c3sfcvf9hy4m
美国东部 (弗吉尼亚北部)	static-us-east-1-prod-static-mft1klnkc4h1

AWS 区域	存储桶名称
美国西部 (俄勒冈州)	static-us-west-2-prod-static-p21mksqx9zlr
美国西部 (北加利福尼亚)	static-us-west-1-prod-static-16d59zrrp01z0
非洲 (开普敦)	static-af-south-1-prod-static-v6v7i5ypdppv
亚太地区 (香港)	static-ap-east-1-prod-static-171xhpfkrorh6
Asia Pacific (Mumbai)	static-ap-south-1-prod-static-ykocre202i9d
亚太地区 (大阪)	static-ap-northeast-3-prod-static-ivmxqzrx2ioi
亚太地区 (首尔)	static-ap-northeast-2-prod-static-1wxyctlhwiajm
亚太地区 (新加坡)	static-ap-southeast-1-prod-static-13ibpyrx4vk6d
亚太地区 (悉尼)	static-ap-southeast-2-prod-static-1cjsl8bx27rfu
亚太地区 (东京)	static-ap-northeast-1-prod-static-4fwvbdisquj8
加拿大 (中部)	static-ca-central-1-prod-static-g80lpejy486c
欧洲地区 (法兰克福)	static-eu-central-1-prod-static-14lbgls2vrkh
欧洲地区 (爱尔兰)	static-eu-west-1-prod-static-hld3vzaf7c4h

AWS 区域	存储桶名称
欧洲地区 (伦敦)	static-eu-west-2-prod-static-361bg202837x
欧洲地区 (米兰)	static-eu-south-1-prod-static-1379tzkd3ni7d
欧洲地区 (巴黎)	static-eu-west-3-prod-static-1rwpkf766ke58
欧洲地区 (斯德哥尔摩)	static-eu-north-1-prod-static-1qzw982y7yu7e
中东 (巴林)	static-me-south-1-prod-static-gmljex38qtqx
南美洲 (圣保罗)	static-sa-east-1-prod-static-1cl8k0y7opidt
以色列 (特拉维夫)	static-il-central-1-prod-static-k02vrnhcesue

9. 选择 Create Endpoint (创建端点)。

如果您提供了正确的配置信息，则会出现一条消息，显示已创建的端点的 ID。

10. 要检查您的 IDE 是否可以访问 Amazon S3 存储桶，请在菜单栏选择 Window (窗口) > New Terminal (新建终端) 以启动终端会话。然后，运行下面的命令，同时将 {bucket_name} 替换为您的区域的桶名称。

```
ping {bucket_name}.s3.{region}.amazonaws.com.
```

例如，如果您在美国东部 (弗吉尼亚州北部) 区域为 S3 桶创建了端点，则运行以下命令。

```
ping static-us-east-1-prod-static-mft1klnkc4hl.s3.us-east-1.amazonaws.com
```

如果 ping 得到响应，则确认 IDE 可以访问该存储桶及其依赖项。

有关此功能的更多信息，请参阅《AWS PrivateLink 指南》中的 [Amazon S3 的端点](#)。

为私有连接配置 VPC 终端节点

当您将实例启动到带有 access using Systems Manager (使用 Systems Manager 访问) 选项的子网内，则其安全组没有允许传入网络流量的入站规则。但是，安全组具有出站规则，以允许来自实例的出站流量。必须下载所需的软件包和库以保持 AWS Cloud9 IDE 是最新的。

要防止实例的出站和入站流量，请为 Systems Manager 创建和配置 Amazon VPC 端点。通过接口 VPC 端点 (接口端点)，您可以连接到 [AWS PrivateLink](#) 支持的服务。AWS PrivateLink 技术可用于通过使用私有 IP 地址私下访问 Amazon EC2 和 Systems Manager API。要将 VPC 终端节点配置为使用 Systems Manager，请按照此 [知识中心资源](#) 提供的说明进行操作。

Warning

假设您配置的安全组不允许入站或出站网络流量。然后，支持您的 AWS Cloud9 IDE 的 EC2 实例无法访问互联网。您需要创建一个 [适用于您的 VPC 的 Amazon S3 端点](#)，以允许访问包含在可信 S3 桶中的依赖项。此外，如果无法访问互联网，一些 AWS 服务 (如 AWS Lambda) 可能无法按预期工作。

选择 AWS PrivateLink 后，对于通过 VPC 端点处理的每个 GB 都会收取数据处理费用。这与流量的来源或目的地无关。有关更多信息，请参阅 [AWS PrivateLink 定价](#)。

在 AWS Cloud9 中打开环境

此过程介绍如何在 AWS Cloud9 中打开环境。

Note

此过程假定您已经创建了一个 AWS Cloud9 开发环境。要创建环境，请参阅 [创建环境](#)。

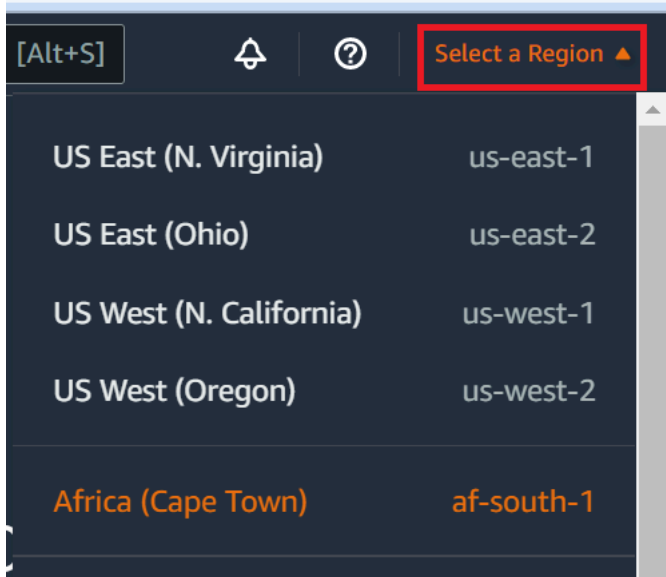
1. 如下所示登录 AWS Cloud9 控制台：

- 如果您是使用 AWS 账户的唯一用户，或者您是单个 AWS 账户中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。
- 如果您的企业使用 AWS IAM Identity Center，请联系 AWS 账户管理员以获取登录说明。

⚠ Important

如果您[退出 AWS 账户](#)，则之后最多仍可访问 AWS Cloud9 IDE 五分钟。然后，当所需权限到期时，将拒绝访问。

2. 在顶部导航栏中，选择环境所在的 AWS 区域。



3. 在环境列表中，对于要打开的环境，请执行以下操作之一：

- 在卡片的内部，选择 Open in Cloud9 (在 Cloud9 中打开) 链接。
- 选择卡片，然后选择 Open in Cloud9 (在 Cloud9 中打开) 按钮。



如果控制台中未显示您的环境，请尝试执行以下一个或多个操作来显示它。

- 在 Environments (环境) 页面的下拉菜单栏中，选择以下一项或多项。
 - 选择 My environments (我的环境) 以显示您的 AWS 实体在选定的 AWS 区域和 AWS 账户中拥有的所有环境。
 - 选择 Shared with me (已与我共享) 以显示您的 AWS 实体在所选 AWS 区域和 AWS 账户中受邀加入的所有环境。
 - 选择 All account environments (所有账户环境) 以显示您的 AWS 实体在选定的 AWS 区域和 AWS 账户中有权显示的所有环境。

- 如果认为您是某个环境的成员，但在 Shared with you (已与您共享) 列表中未显示该环境，请与该环境的拥有者联系。
- 在顶部导航栏中，选择不同的 AWS 区域。

从 AWS Cloud9 中的环境调用 AWS 服务

您可以从 AWS Cloud9 开发环境中调用 AWS 服务。例如，您可以执行以下操作：

- 在 Amazon Simple Storage Service (Amazon S3) 存储桶中上传和下载数据。
- 通过 Amazon Simple Notification Service (Amazon SNS) 主题发送广播通知。
- 在 Amazon DynamoDB (DynamoDB) 数据库中读取和写入数据。

您可以通过几种方法从您的环境中调用 AWS 服务。例如，您可以使用 AWS Command Line Interface (AWS CLI) 或 AWS CloudShell 从终端会话中运行命令。您也可以从您在环境中运行的代码调用 AWS 服务。您可以通过对编程语言 (例如 JavaScript、Python、Ruby、PHP、Go 和 C++) 使用 AWS SDK 来实现此目的。有关更多信息，请参阅 [AWS CLI 和 aws-shell 示例](#)、[AWS Command Line Interface 用户指南](#) 和 [AWS 开发工具包](#)。

每次 AWS CLI、AWS CloudShell 或您的代码调用 AWS 服务时，AWS CLI、AWS CloudShell 或您的代码必须随调用提供一组 AWS 访问凭证。这些凭证确定调用方是否具有相应的权限以进行调用。如果凭证没有包括合适的权限，则调用失败。

可以通过几种方法为您的环境提供凭证。下表介绍了一些方法。

环境类型	方法
EC2	<p>使用 AWS 托管式临时凭证。</p> <p>我们建议您在 EC2 环境中使用此方法。AWS 托管式临时凭证管理 AWS 在 EC2 环境中代表您访问凭证的权限，同时还符合 AWS 安全最佳实践。</p> <p>如果您使用的是 EC2 环境，可以跳过本主题的其余部分。这是因为已在环境中设置了 AWS 托管式临时凭证。</p>

环境类型	方法
EC2	<p>有关更多信息，请参阅 AWS 托管式临时凭证。</p> <p>将一个 IAM 实例配置文件附加到实例。</p> <p>只有在由于某种原因无法使用 AWS 托管式临时凭证时，才使用该方法。与 AWS 托管式临时凭证相似，实例配置文件管理 AWS 代表您访问凭证的权限。不过，您必须自行创建和管理实例配置文件，并将其附加到 Amazon EC2 实例。</p> <p>有关说明，请参阅 创建和使用实例配置文件管理临时凭证。</p>
EC2 或 SSH	<p>将您的 AWS 永久访问凭证存储在环境中。</p> <p>该方法不如使用 AWS 临时访问凭证安全。不过，这是 SSH 环境支持的唯一方法。</p> <p>有关说明，请参阅 在环境中创建和存储永久访问凭证。</p>
EC2 或 SSH	<p>将您的 AWS 永久访问凭证直接插入到您的代码中。</p> <p>我们不建议使用该方法，因为它不符合 AWS 安全最佳实践。</p> <p>由于我们不建议使用该方法，因此，在本主题中没有介绍该方法。</p>

创建和使用实例配置文件管理临时凭证

Note

您无法在 AWS Cloud9 SSH 开发环境中使用此步骤。请向前跳到 [在环境中创建和存储永久访问凭证](#)。

我们建议您使用 AWS 托管式临时凭证，而不是实例配置文件。只有在出于某种原因无法使用 AWS 托管式临时凭证时，才应按照这些说明进行操作。有关更多信息，请参阅 [AWS 托管式临时凭证](#)。

此过程使用 IAM 和 Amazon EC2 创建 IAM 实例配置文件，并将其附加到连接到您环境的 Amazon EC2 实例。该实例配置文件代表您管理临时凭证。此过程假定您已经在 AWS Cloud9 中创建了一个环境。要创建一个环境，请参阅 [创建环境](#)。

您可以使用 [IAM 和 Amazon EC2 控制台](#) 或 [AWS Command Line Interface \(AWS CLI\)](#) 完成这些任务。

使用 IAM 控制台创建实例配置文件

Note

如果您已具有一个包含实例配置文件的 IAM 角色，请向前跳到 [使用 Amazon EC2 控制台将实例配置文件附加到实例](#)。

1. 通过以下网址登录到 IAM 控制台：<https://console.aws.amazon.com/iam/>。

对于该步骤，我们建议您使用您的 AWS 账户 中的管理员级别凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

2. 在导航栏中，选择 Roles (角色)。

Note

您无法使用 IAM 控制台创建实例配置文件本身。您必须创建一个包含实例配置文件的 IAM 角色。

3. 选择 Create role (创建角色)。
4. 在 Select type of trusted entity (选择可信实体的类型) 页面中，在已选择 AWS 服务的情况下，在 Choose the service that will use this role (选择将使用此角色的服务) 中选择 EC2。
5. 在选择您的使用案例中选择 EC2。
6. 选择 Next: Permissions (下一步：权限)。
7. 在 Attach permissions policies (附加权限策略) 页面中，在策略列表中选中 AdministratorAccess 旁边的框，然后选择 Next: Review (下一步：审核)。

Note

AdministratorAccess 策略允许不受限制地访问您的 AWS 账户 中的所有 AWS 操作和资源。只应将它用于实验目的。有关更多信息，请参阅 IAM 用户指南中的 [IAM 策略](#)。

- 在 Review (查看) 页面上，为 Role Name (角色名称) 键入角色的名称 (例如 my-demo-cloud9-instance-profile)。
- 请选择 Create Role(创建角色)。

向前跳到[使用 Amazon EC2 控制台将实例配置文件附加到实例](#)。

使用 AWS CLI 创建实例配置文件

Note

如果您已具有一个包含实例配置文件的 IAM 角色，请向前跳到[使用 AWS CLI 将实例配置文件附加到实例](#)。

对于本主题，我们建议您使用您的 AWS 账户 中的管理员级别凭证配置 AWS CLI。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

- 在 AWS 中为实例配置文件所需的 IAM 角色定义信任关系。为此，请创建并保存一个包含以下内容的文件 (如 my-demo-cloud9-instance-profile-role-trust.json)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
```

```
"Effect": "Allow",
"Principal": {
  "Service": "ec2.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
}
```

2. 通过使用终端或命令提示符，切换到刚在其中保存该文件的目录。
3. 为实例配置文件创建一个 IAM 角色。为此，请运行 IAM `create-role` 命令。执行此操作时，请指定新 IAM 角色的名称（例如 `my-demo-cloud9-instance-profile-role`），以及刚刚保存的文件的名称。

```
aws iam create-role --role-name my-demo-cloud9-instance-profile-role --assume-role-policy-document file://my-demo-cloud9-instance-profile-role-trust.json
```

4. 将 AWS 访问权限附加到实例配置文件的 IAM 角色。为此，请运行 IAM `attach-role-policy` 命令。指定现有 IAM 角色的名称以及名为 `AdministratorAccess` 的 AWS 托管式策略的 Amazon 资源名称（ARN）。

```
aws iam attach-role-policy --role-name my-demo-cloud9-instance-profile-role --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

Note

`AdministratorAccess` 策略允许不受限制地访问您的 AWS 账户中的所有 AWS 操作和资源。只应将它用于实验目的。有关更多信息，请参阅 IAM 用户指南中的 [IAM 策略](#)。

5. 创建实例配置文件。为此，请运行 IAM `create-instance-profile` 命令，并指定新实例配置文件的名称（如 `my-demo-cloud9-instance-profile`）。

```
aws iam create-instance-profile --instance-profile-name my-demo-cloud9-instance-profile
```

6. 将 IAM 角色附加到实例配置文件。为此，请运行 IAM `add-role-to-instance-profile`，并指定现有 IAM 角色和实例配置文件的名称。

```
aws iam add-role-to-instance-profile --role-name my-demo-cloud9-instance-profile-role
--instance-profile-name my-demo-cloud9-instance-profile
```

向前跳到[使用 AWS CLI 创建实例配置文件](#)。

使用 Amazon EC2 控制台将实例配置文件附加到实例

1. 登录到 Amazon EC2 控制台，网址：<https://console.aws.amazon.com/ec2/>。

对于该步骤，我们建议您使用您的 AWS 账户 中的管理员级别凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

2. 在导航栏中，确保区域选择器显示的 AWS 区域 与您环境的区域匹配。例如，如果您在美国东部（俄亥俄州）区域中创建了环境，请在此处的区域选择器中选择 US East (Ohio) [美国东部（俄亥俄州）]。
3. 选择正在运行的实例链接，或者在导航窗格中展开实例，然后选择实例。
4. 在实例列表中，选择 Name（名称）包含您的环境名称的实例。例如，如果您的环境名称为 my-demo-environment，请选择 Name（名称）包含 my-demo-environment 的实例。
5. 依次选择 Actions（操作）、Security（安全）和 Modify IAM role（修改 IAM 角色）。

Note

虽然您将一个角色附加到实例，但该角色包含一个实例配置文件。

6. 在 Modify IAM role（修改 IAM 角色）页面中，为 IAM role（IAM 角色）选择您确定的角色名称或在上一过程中创建的角色名称，然后选择 Apply（应用）。
7. 返回到环境中，使用 AWS CLI 运行 `aws configure` 命令，或使用 AWS CloudShell 运行 `configure` 命令。不要为 AWS 访问密钥 ID 或 AWS 秘密访问密钥指定任何值（在出现每个提示后按 Enter）。对于 Default Region name（原定设置区域名称），请指定离您最近的 AWS 区域或您的 AWS 资源所在的区域。例如，对于美国东部（俄亥俄）区域，请指定 us-east-2。有关区域的列表，请参阅《Amazon Web Services 一般参考》中的 [AWS 区域 and Endpoints](#)。（可选）为 Default output format（默认输出格式）指定一个值（如 json）。

您现在可以开始从您的环境中调用 AWS 服务。要使用 AWS CLI 和/或 aws-shell 调用 AWS 服务，请参阅 [AWS CLI 和 aws-shell 示例](#)。要从您的代码中调用 AWS 服务，请参阅其他[教程和示例](#)。

使用 AWS CLI 将实例配置文件附加到实例

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

1. 运行 Amazon EC2 `associate-iam-instance-profile` 命令。指定实例配置文件的名称以及环境的 Amazon EC2 实例的 ID 和 AWS 区域 ID。

```
aws ec2 associate-iam-instance-profile --iam-instance-profile Name=my-demo-cloud9-instance-profile --region us-east-2 --instance-id i-12a3b45678cdef9a0
```

在上述命令中，将 `us-east-2` 替换为实例的 AWS 区域 ID，并将 `i-12a3b45678cdef9a0` 替换为实例的 ID。

例如，要获取实例 ID，您可以运行 Amazon EC2 `describe-instances` 命令，并指定环境的名称和 AWS 区域 ID。

```
aws ec2 describe-instances --region us-east-2 --filters Name=tag:Name,Values=*my-environment* --query "Reservations[*].Instances[*].InstanceId" --output text
```

在上述命令中，将 `us-east-2` 替换为实例的 AWS 区域 ID，并将 `my-environment` 替换为环境名称。

2. 返回到环境中，使用 AWS CLI 运行 `aws configure` 命令，或使用 `aws-shell` 运行 `configure` 命令。不要为 AWS 访问密钥 ID 或 AWS 秘密访问密钥指定任何值。在出现每个提示后按 Enter。对于 Default Region name（原定设置区域名称），请指定离您最近的 AWS 区域或您的 AWS 资源所在的区域。例如，对于美国东部（俄亥俄）区域，请指定 `us-east-2`。有关区域的列表，请参阅《Amazon Web Services 一般参考》中的 [AWS Regions and Endpoints](#)。（可选）为 Default output format（默认输出格式）指定一个值（如 `json`）。

您现在可以开始从您的环境中调用 AWS 服务。要使用 AWS CLI 和/或 `aws-shell` 调用 AWS 服务，请参阅 [AWS CLI 和 `aws-shell` 示例](#)。要从您的代码中调用 AWS 服务，请参阅其他 [教程和示例](#)。

在环境中创建和存储永久访问凭证

Note

如果您使用的是 AWS Cloud9 EC2 开发环境中，我们建议您使用 AWS 托管式临时凭证，而不是 AWS 永久访问凭证。要使用 AWS 托管式临时凭证，请参阅 [AWS 托管的临时证书](#)。

在本节中，您使用 AWS Identity and Access Management (IAM) 生成一组永久凭证。调用 AWS 服务时，AWS CLI、aws-shell 或您的代码可以使用这组凭证。该组凭证包括 AWS 访问密钥 ID 和 AWS 秘密访问密钥，这些凭证对于您的 AWS 账户中的用户是唯一的。如果您已具有 AWS 访问密钥 ID 和 AWS 秘密访问密钥，请记住这些凭证，然后向前跳到[在环境中存储永久访问凭证](#)。

您可以使用 [IAM 控制台](#) 或 [AWS CLI](#) 创建一组永久凭证。

授权以编程方式访问

如果用户需要在 AWS Management Console 之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。

要向用户授予编程式访问权限，请选择以下选项之一。

哪个用户需要编程式访问权限？	目的	方式
员工身份 (在 IAM Identity Center 中管理的用户)	使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 <ul style="list-style-type: none"> 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的IAM Identity Center 身份验证。

哪个用户需要编程式访问权限？	目的	方式
IAM	使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照《IAM 用户指南》中 将临时凭证用于 AWS 资源 中的说明进行操作。
IAM	(不推荐使用) 使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。	按照您希望使用的界面的说明进行操作。 <ul style="list-style-type: none"> 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。

使用 AWS CLI 创建永久访问凭证

Note

对于本节，我们建议使用您的 AWS 账户中的 IAM 管理员级别凭证配置 AWS CLI。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行

某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

运行 IAM `create-access-key` 命令，以便为用户创建新的 AWS 访问密钥和相应的 AWS 秘密访问密钥。

```
aws iam create-access-key --user-name MyUser
```

在上述命令中，将 `MyUser` 替换为用户的名称。

在一个安全位置中，保存显示的 `AccessKeyId` 和 `SecretAccessKey` 值。在运行 IAM `create-access-key` 命令后，这是您唯一一次可以使用 AWS CLI 查看用户的 AWS 秘密访问密钥。要在以后需要时为用户生成新的 AWS 秘密访问密钥，请参阅 IAM 用户指南中的[创建、修改和查看访问密钥 \(API、CLI、PowerShell\)](#)。

在环境中存储永久访问凭证

在此过程中，您使用 AWS Cloud9 IDE 将 AWS 永久访问凭证存储在您的环境中。此过程假定您已在 AWS Cloud9 中创建了一个环境，打开了该环境，并在 Web 浏览器中显示 AWS Cloud9 IDE。有关更多信息，请参阅[创建环境](#)和[打开环境](#)。

Note

以下过程介绍了如何使用环境变量存储您的永久访问凭证。如果您在环境中安装了 AWS CLI 或 `aws-shell`，您可以改为使用 `aws configure` 命令（对于 AWS CLI）或 `configure` 命令（对于 `aws-shell`）存储您的永久访问凭证。有关说明，请参阅 AWS Command Line Interface 用户指南中的[快速配置](#)。

1. 在 AWS Cloud9 IDE 中打开您的环境的情况下，启动新的终端会话（如果尚未启动）。要启动新的终端会话，请在菜单栏上选择 `Window (窗口) > New Terminal (新建终端)`。
2. 运行每个以下命令（每次运行一个命令），以设置表示您的永久访问凭证的本地环境变量。在这些命令中，在 `AWS_ACCESS_KEY_ID:` 后面输入您的 AWS 访问密钥 ID。在 `AWS_SECRET_ACCESS_KEY` 后面，输入您的 AWS 秘密访问密钥。在 `AWS_DEFAULT_REGION_ID` 后面，输入与离您最近的 AWS 区域（或您首选的 AWS 区域）关联的 AWS 区域标识符。有关可用的标识符列表，请参阅《Amazon Web Services 一般参考》中的[AWS 区域 and Endpoints](#)。例如，对于美国东部（俄亥俄州），请使用 `us-east-2`。

```
export AWS_ACCESS_KEY_ID=  
export AWS_SECRET_ACCESS_KEY=  
export AWS_DEFAULT_REGION=
```

3. 请注意，上述环境变量仅适用于当前终端会话。要使这些环境变量在终端会话之间可用，您必须将其作为用户环境变量添加到 shell 配置文件中，如下所示。
 - a. 在 IDE 的 Environment (环境) 窗口中，选择齿轮图标，然后选择 Show Home in Favorites (在收藏夹中显示主页)。重复该步骤，然后选择 Show Hidden Files (显示隐藏的文件)。
 - b. 打开 ~/.bashrc 文件。
 - c. 在文件末尾输入或粘贴以下代码。在这些命令中，在 AWS_ACCESS_KEY_ID: 后面输入您的 AWS 访问密钥 ID。在 AWS_SECRET_ACCESS_KEY 后面，输入您的 AWS 秘密访问密钥。在 AWS_DEFAULT_REGION_ID 后面，输入与离您最近的 AWS 区域 (或您首选的 AWS 区域) 关联的 AWS 区域 标识符。有关可用的标识符列表，请参阅《Amazon Web Services 一般参考》中的 [AWS 区域 and Endpoints](#)。例如，对于美国东部 (俄亥俄州) 区域，请使用 us-east-2。

```
export AWS_ACCESS_KEY_ID=  
export AWS_SECRET_ACCESS_KEY=  
export AWS_DEFAULT_REGION=
```

- d. 保存该文件。
- e. 获取 ~/.bashrc 文件以加载这些新环境变量。

```
. ~/.bashrc
```

您现在可以开始从您的环境中调用 AWS 服务。要使用 AWS CLI 或 aws-shell 调用 AWS 服务，请参阅 [AWS CLI 和 aws-shell 示例](#)。要从您的代码中调用 AWS 服务，请参阅其他[教程和示例](#)。

在 AWS Cloud9 中更改环境设置

您可以更改 AWS Cloud9 开发环境的首选项或设置。

- [更改环境首选项](#)
- [使用控制台更改环境设置](#)
- [使用代码更改环境设置](#)

更改环境首选项

1. 打开您要更改其设置的环境。要打开环境，请参阅[打开环境](#)。
2. 在打开AWS Cloud9的情况下，在 AWS Cloud9 菜单栏上依次选择 、 Preferences (首选项) 。
3. 在 Preferences (首选项) 窗口中，选择 Project Settings (项目设置) 。
4. 根据需要更改任意可用项目设置。这包括 Code Editor (Ace) (代码编辑器 (Ace)) 和 Find in Files (在文件中查找) 等设置。

Note

有关更多信息，请参阅[您可以进行的项目设置更改](#)。

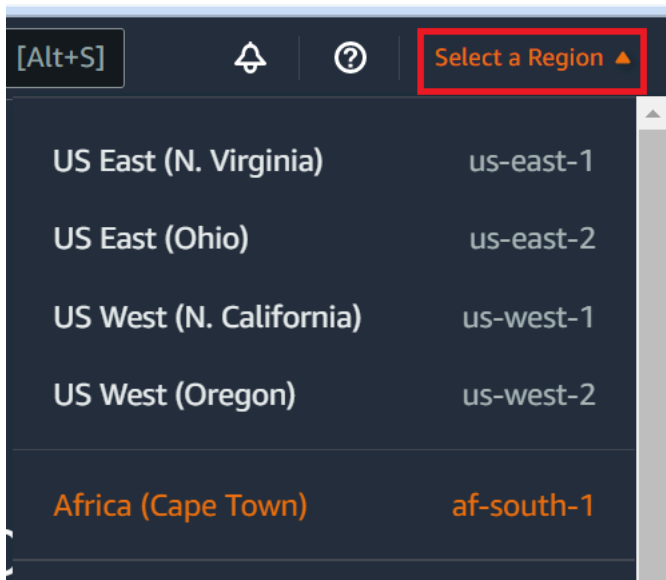
在 AWS Cloud9 IDE 中调整环境的超时

以下步骤概述了如何在 AWS Cloud9 IDE 中更新 Amazon EC2 环境的超时时间。这个时间将是环境停止之前所能持续的时间。

1. 打开要配置的环境。
2. 在 AWS Cloud9 IDE 中，在菜单栏上依次选择 AWS Cloud9、首选项。
3. 在首选项窗口中，滚动至 Amazon EC2 实例部分。
4. 从可用列表中选择超时值并更新。

使用控制台更改环境设置

1. 如下所示登录 AWS Cloud9 控制台：
 - 如果您是使用 AWS 账户 的唯一一个人用户，或者您是单个 AWS 账户 中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的组织使用 AWS IAM Identity Center，请联系 AWS 账户 管理员以获取登录说明。
2. 在顶部导航栏中，选择环境所在的 AWS 区域。



3. 在环境列表中，对于要更改其设置的环境，请执行以下操作之一。
 - 选择环境的选项卡标题。然后，在下一页中选择 Delete (删除)。
 - 选择环境的卡，然后选择 View details (查看详细信息) 按钮。
4. 进行更改，然后选择 Save changes (保存更改)。

您可以使用 AWS Cloud9 控制台更改以下设置。

- 对于 EC2 环境，可更改 Name (名称) 和 Description (描述)。
- 对于 SSH 环境：可更改 Name (名称)、Description (描述)、User (用户)、Host (主机)、Port (端口)、Environment path (环境路径)、Node.js binary path (Node.js 二进制文件路径) 和 SSH jump host (SSH 跳转主机)。

要更改其他设置，请执行以下操作。

- 对于 EC2 环境，请执行以下操作之一。
 - 您无法更改 Type (类型)、Security groups (安全组)、VPC、Subnet (子网)、Environment path (环境路径) 或 Environment ARN (环境 ARN)。
 - 对于 Permissions (权限) 或 Number of members (成员数)，请参阅[更改环境成员的访问角色](#)、[删除用户](#)、[邀请 IAM 用户](#)和[删除其他环境成员](#)。
 - 对于 EC2 instance type (EC2 实例类型)、Memory (内存) 或 vCPU，请参阅[移动环境或调整大小](#)。
- 对于 SSH 环境，请执行以下操作之一。
 - 您无法更改 Type (类型) 或 Environment ARN (环境 ARN)。

- 对于 Permissions (权限) 或 Number of members (成员数) ，请参阅[更改环境成员的访问角色](#)、[删除用户](#)、[邀请 IAM 用户](#)和[删除其他环境成员](#)。

如果控制台中未显示您的环境，请尝试执行以下一个或多个操作来显示它。

- 在 Environments (环境) 页面的下拉菜单栏中，选择以下一项或多项。
 - 选择 My environments (我的环境) 以显示您的 AWS 实体在选定的 AWS 区域 和 AWS 账户 中拥有的所有环境。
 - 选择 Shared with me (已与我共享) 以显示您的 AWS 实体在所选 AWS 区域 和 AWS 账户 中受邀加入的所有环境。
 - 选择 All account environments (所有账户环境) 以显示您的 AWS 实体在选定的 AWS 区域 和 AWS 账户 中有权显示的所有环境。
- 如果认为您是某个环境的成员，但在 Shared with you (已与您共享) 列表中未显示该环境，请与该环境的拥有者联系。
- 在顶部导航栏中，选择不同的 AWS 区域。

使用代码更改环境设置

要在 AWS Cloud9 中使用代码更改环境的设置，请调用 AWS Cloud9 更新环境操作，如下所示。

AWS CLI	update-environment
AWS SDK for C++	UpdateEnvironmentRequest 、 UpdateEnvironmentResult
AWS SDK for Go	UpdateEnvironment 、 UpdateEnvironmentRequest 、 UpdateEnvironmentWithContext
AWS SDK for Java	UpdateEnvironmentRequest 、 UpdateEnvironmentResult
AWS SDK for JavaScript	updateEnvironment
AWS SDK for .NET	UpdateEnvironmentRequest 、 UpdateEnvironmentResponse

AWS SDK for PHP	updateEnvironment
AWS SDK for Python (Boto)	update_environment
AWS SDK for Ruby	update_environment
AWS Tools for Windows PowerShell	Update-C9Environment
AWS Cloud9 API	UpdateEnvironment

在 AWS Cloud9 中使用共享环境

共享环境是指已邀请多个用户参加的 AWS Cloud9 开发环境。本主题提供了在 AWS Cloud9 中共享环境以及如何参与共享环境的说明。

要邀请用户参加您拥有的环境，请按照其中一组过程操作。根据要邀请的用户类型进行选择。

- 如果您是一名用户且位于与环境相同的 AWS 账户中，则应[邀请与环境位于相同账户中的用户](#)。
- 如果您是一名 AWS Cloud9 管理员且位于与环境相同的 AWS 账户中，特别是 AWS 账户根用户、管理员用户或附加了 AWS 托管式策略 `AWSCloud9Administrator` 的用户，则应自行邀请 AWS Cloud9 管理员（请参阅[邀请与环境位于相同账户中的用户](#)），或者让 AWS Cloud9 管理员邀请自己或同一个 AWS 账户中的其他人（请参阅[让与环境位于相同账户中的 AWS Cloud9 管理员邀请他们自己或其他人](#)）。

共享环境使用案例

共享环境非常适合以下使用案例。

- 配对编程（也称为对等编程）：这是指，两个用户在单个环境中一起处理相同的代码。在配对编程中，通常一个用户编写代码，同时另一个用户观察正在编写的代码。观察者为代码编写者提供即时输入和反馈。在完成项目期间，他们的位置经常发生变换。如果没有共享环境，结对编程人员团队通常坐在一台计算机前面。每次只有一个用户可以编写代码。在共享环境中，两个用户可以坐在自己的计算机前面。此外，他们可以同时编写代码，即使他们在不同的办公室工作也是如此。
- 计算机科学课程：当老师或助教要访问学生的环境时，这非常有用。这样做可以检查学生的家庭作业或实时解决其环境的问题。学生还可以与同学一起在单个环境中实时编写代码，以便共同完成共享家庭作业项目。即使他们可能位于不同的位置并使用不同的计算机操作系统和 Web 浏览器类型，他们也可以完成该任务。

- 任何其他情况 - 多个用户需要实时协作处理相同的代码。

关于环境成员访问角色

在 AWS Cloud9 中共享环境或参与共享的环境之前，您应该了解共享环境的访问权限级别。我们将这些权限级别称为环境成员访问角色。

AWS Cloud9 中的共享环境提供了三种环境成员访问角色：拥有者、读取/写入以及只读。

- 拥有者拥有环境的完全控制权。每个环境有且仅有一个拥有者，其是环境创建者。拥有者可以执行以下操作。
 - 添加、更改和删除环境的成员
 - 打开、查看和编辑文件
 - 运行代码
 - 更改环境设置
 - 与其他成员聊天
 - 删除现有的聊天消息

在 AWS Cloud9 IDE 中，环境拥有者显示为具有读取 + 写入访问权限。

- 读取/写入成员可以执行以下操作。
 - 打开、查看和编辑文件
 - 运行代码
 - 从 AWS Cloud9 IDE 中更改各种环境设置
 - 与其他成员聊天
 - 删除现有的聊天消息

在 AWS Cloud9 IDE 中，成员显示为具有读取 + 写入访问权限。

- 只读成员可以执行以下操作。
 - 打开和查看文件
 - 与其他成员聊天
 - 删除现有的聊天消息

在 AWS Cloud9 IDE 中，成员显示为具有只读访问权限。

用户必须满足以下任一条件，然后才能成为环境拥有者或成员。

- 用户是 AWS 账户 根用户。
- 用户是管理员用户。有关更多信息，请参阅 [IAM 用户指南](#) 中的创建您的第一个 IAM 管理员用户和组。
- 用户是属于 IAM 组的用户、代入角色的用户或代入角色的联合身份用户，并且该组或角色附加了 AWS 托管式策略 AWSCloud9Administrator 或 AWSCloud9User (或 AWSCloud9EnvironmentMember，仅适用于成员)。有关更多信息，请参阅 [AWS 托管式 \(预定义 \) 策略](#)。
- 要将上述任一托管式策略附加到 IAM 组，您可以使用 [AWS Management Console](#) 或 [AWS 命令行界面 \(AWS CLI\)](#)，如以下过程所述。
- 您可以在 IAM 中创建一个角色 (该角色具有前面的托管式策略之一)，供用户或联合用户代入。有关更多信息，请参阅《IAM 用户指南》中的 [创建角色](#)。要让用户或联合身份用户代入角色，请参阅 IAM 用户指南中的 [使用 IAM 角色](#) 中的带入角色相关内容。

使用控制台将 AWS Cloud9 的 AWS 托管式策略附加到组

以下过程概述了如何使用控制台将 AWS Cloud9 的 AWS 托管式策略附加到组。

1. 登录 AWS Management Console (如果尚未登录)。

对于该步骤，我们建议您使用您的 AWS 账户 中的 IAM 管理员级别凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户管理员。

2. 打开 IAM 控制台。为此，请在控制台导航栏中选择 Services (服务)。然后，选择 IAM。
3. 选择 Groups (组)。
4. 选择组的名称。
5. 在 Permissions (权限) 选项卡中，在 Manged Policies (托管式策略) 处，选择 Attach Policy (附加策略)。
6. 在策略名称列表中，选中以下框之一。
 - AWSCloud9User (首选) 或 AWSCloud9Administrator 以允许组中的每个用户成为环境拥有者
 - AWSCloud9EnvironmentMember 以允许组中的每个用户仅成为成员

(如果在列表中看不到某个策略名称，请在搜索框中键入策略名称以显示该策略。)

7. 选择 Attach policy (附上策略)。

使用 AWS CLI 将 AWS Cloud9 的 AWS 托管式策略附加到组

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

运行 IAM `attach-group-policy` 命令以将 AWS Cloud9 的 AWS 托管式策略附加到组。指定组名称和策略的 Amazon 资源名称 (ARN)。

```
aws iam attach-group-policy --group-name MyGroup --policy-arn arn:aws:iam::aws:policy/  
POLICY_NAME
```

在上述命令中，将 `MyGroup` 替换为组的名称。将 `POLICY_NAME` 替换为以下某个 AWS 托管式策略的名称。

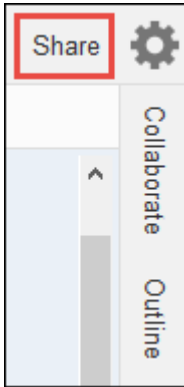
- `AWSCloud9User` (首选) 或 `AWSCloud9Administrator` 以允许组中的每个用户成为环境拥有者
- `AWSCloud9EnvironmentMember` 以允许组中的每个用户仅成为成员

邀请与环境位于相同账户中的用户

按照本节中的说明，将您在您的 AWS 账户中拥有的 AWS Cloud9 开发环境与该相同账户中的用户共享。

1. 假设您要邀请的用户不是以下类型的用户之一。确保您要邀请的用户已经具有相应的环境成员访问角色。有关说明，请参阅[关于环境成员访问角色](#)。
 - AWS 账户根用户。
 - 管理员用户。
 - 属于 IAM 组的用户、代入角色的用户或代入角色的联合身份用户，并且该组或角色附加了 AWS 托管式策略 `AWSCloud9Administrator`。
2. 打开您拥有并希望邀请用户加入的环境 (如果尚未打开该环境)。
3. 在 AWS Cloud9 IDE 的菜单栏中，执行以下任一操作。
 - 依次选择 Window (窗口)、Share (共享)。

- 选择 Share (共享) (位于 Preferences (首选项) 齿轮图标旁边)。



4. 在 Share this environment (共享此环境) 对话框中，为 Invite Members (邀请成员) 键入以下各项之一。
 - 要邀请 IAM 用户，请输入该用户的名称。
 - 要邀请 AWS 账户 根用户，请输入 `arn:aws:iam::123456789012:root`。将 123456789012 替换为您的 AWS 账户 ID。
 - 要邀请具有代入角色的用户或具有代入角色的联合用户，请输入 `arn:aws:sts::123456789012:assumed-role/MyAssumedRole/MyAssumedRoleSession`。将 123456789012 替换为您的 AWS 账户 ID，并将 MyAssumedRole 替换为代入角色的名称。将 MyAssumedRoleSession 替换为代入角色的会话名称。
5. 要使该用户成为只读成员，请选择 R。要使该用户成为读取/写入成员，请选择 RW。
6. 选择 Invite。

Note

如果使该用户成为读取/写入成员，将显示一个对话框，其中包含有关可能给您的 AWS 安全凭证带来风险的信息。以下信息提供了有关该问题的更多背景信息。

您只应与信任的人员共享环境。

读取/写入成员或许能够在您的环境中使用 AWS CLI、AWS CloudShell 或 AWS SDK，以代表您在 AWS 中执行操作。此外，如果将您的永久 AWS 访问凭证存储在环境中，该成员可能会复制这些凭证并在环境外部使用这些凭证。

从您的环境中删除永久 AWS 访问凭证并使用临时 AWS 访问凭证并不能完全解决该问题。这能降低成员复制这些临时凭证并在环境外部使用的可能性（因为这些临时凭证仅在有限的时间内有效）。不过，临时凭证仍允许读取/写入成员从环境中代表您在 AWS 中执行操作。

7. 与用户联系，并告诉他们可以打开并开始使用该环境。

让与环境位于相同账户中的 AWS Cloud9 管理员邀请他们自己或其他人

Note

如果您使用的是 [AWS 托管式临时凭证](#)，您将无法使用 AWS Cloud9 IDE 中的终端会话来运行本部分中的某些或所有命令。为了符合 AWS 安全最佳实践，AWS 托管式临时凭证不允许运行某些命令。不过，您可以从单独的 AWS Command Line Interface (AWS CLI) 安装来运行这些命令。

以下类型的用户可以邀请其自己（或相同 AWS 账户中的其他用户）加入相同账户中的任何环境。

- AWS 账户 根用户。
- 管理员用户。
- 属于 IAM 组的用户、代入角色的用户或代入角色的联合身份用户，并且该组或角色附加了 AWS 托管策略 `AWSCloud9Administrator`。

假设受邀用户不是上述类型的用户之一。确保该用户已经拥有相应的环境成员访问角色。有关说明，请参阅 [关于环境成员访问角色](#)。

要邀请用户，请使用 AWS CLI 或 AWS CloudShell 来运行 AWS Cloud9 `create-environment-membership` 命令。

```
aws cloud9 create-environment-membership --environment-id
12a34567b8cd9012345ef67abcd890e1 --user-arn USER_ARN --permissions PERMISSION_LEVEL
```

在上述命令中，将 `12a34567b8cd9012345ef67abcd890e1` 替换为环境的 ID。将 `PERMISSION_LEVEL` 替换为 `read-write` 或 `read-only`。并将 `USER_ARN` 替换为以下各项之一：

- 要邀请 IAM 用户，请输入 `arn:aws:iam::123456789012:user/MyUser`。将 `123456789012` 替换为您的 AWS 账户 ID，并将 `MyUser` 替换为用户的名称。
- 要邀请 AWS 账户 根用户，请输入 `arn:aws:iam::123456789012:root`。将 `123456789012` 替换为您的 AWS 账户 ID。
- 要邀请具有代入角色的用户或具有代入角色的联合用户，请输入 `arn:aws:sts::123456789012:assumed-role/MyAssumedRole/MyAssumedRoleSession`。将 `123456789012` 替换为您的 AWS 账户 ID。将 `MyAssumedRole` 替换为代入角色的名称。并且，将 `MyAssumedRoleSession` 替换为代入角色的会话名称。

例如，要邀请具有账户 ID 123456789012 的 AWS 账户 根用户加入 ID 为 12a34567b8cd9012345ef67abcd890e1 的环境以作为读取/写入成员，请运行以下命令。

```
aws cloud9 create-environment-membership --environment-id
12a34567b8cd9012345ef67abcd890e1 --user-arn arn:aws:iam::123456789012:root --
permissions read-write
```

Note

如果您使用 AWS CloudShell，请在上述命令中省略 `aws` 前缀。

打开共享环境

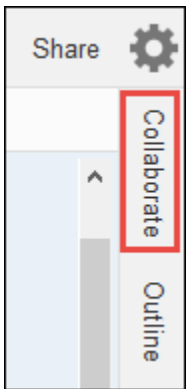
要打开共享环境，您可以使用 AWS Cloud9 控制面板。使用 AWS Cloud9 IDE 在共享环境中执行操作和完成工作。示例是处理文件 and 与其他团队成员聊天。

1. 确保将相应的访问策略附加到您的用户的组或角色。有关更多信息，请参阅[关于环境成员访问角色](#)。
2. 如下所示登录 AWS Cloud9 控制台：
 - 如果您是使用 AWS 账户 的唯一一个人用户，或者您是单个 AWS 账户 中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的组织使用 IAM Identity Center，请联系 AWS 账户 管理员以获取登录说明。
 - 如果您是课堂中的学生，请联系您的导师以获取登录说明。
3. 从 AWS Cloud9 控制面板中打开共享环境。有关更多信息，请参阅[打开 AWS Cloud9 中的环境](#)。

您可以使用 Collaborate (协作) 窗口与其他成员进行交互，如本主题的其余部分中所述。

Note

如果未显示 Collaborate (协作) 窗口，请选择 Collaborate (协作)。如果未显示 Collaborate (协作) 按钮，请在菜单栏上依次选择 Window (窗口)、Collaborate (协作)。

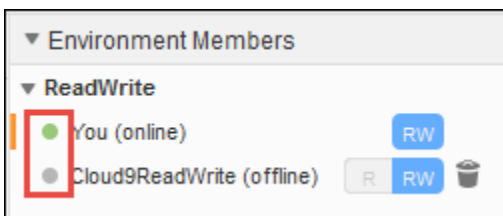


查看环境成员列表

在打开共享环境的情况下，如果未显示成员列表，请在 Collaborate（协作）窗口中展开 Environment Members（环境成员）。

每个成员旁边的圆圈指示其在线状态，如下所示：

- 活动成员具有绿色圆圈。
- 离线成员具有灰色圆圈。
- 空闲成员具有橙色圆圈。



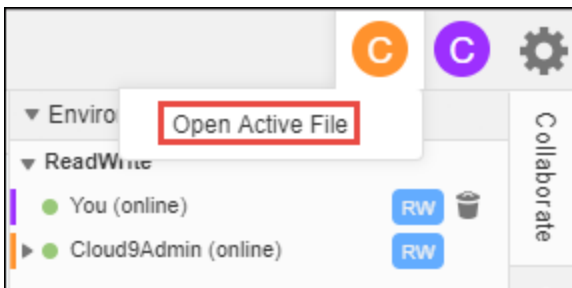
要使用代码获取环境成员列表，请调用 AWS Cloud9 描述环境成员资格操作，如下所示。

AWS CLI	describe-environment-memberships
AWS SDK for C++	DescribeEnvironmentMembershipsRequest 、 DescribeEnvironmentMembershipsResult
AWS SDK for Go	DescribeEnvironmentMemberships 、 DescribeEnvironmentMembershipsRequest 、 DescribeEnvironmentMembershipsWithContext

AWS SDK for Java	DescribeEnvironmentMembershipsRequest 、 DescribeEnvironmentMembershipsResult
AWS SDK for JavaScript	describeEnvironmentMemberships
AWS SDK for .NET	DescribeEnvironmentMembershipsRequest 、 DescribeEnvironmentMembershipsResponse
AWS SDK for PHP	describeEnvironmentMemberships
AWS SDK for Python (Boto)	describe_environment_memberships
AWS SDK for Ruby	describe_environment_memberships
AWS Tools for Windows PowerShell	Get-C9EnvironmentMembershipList
AWS Cloud9 API	DescribeEnvironmentMemberships

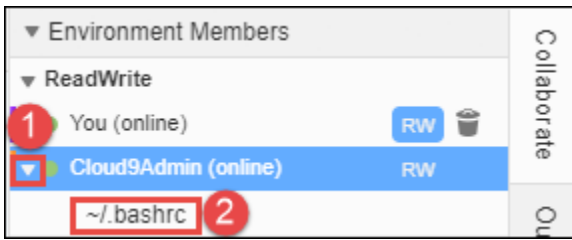
打开环境成员的活动文件

在打开共享环境的情况下，在菜单栏中选择成员名称。然后，选择 Open Active File (打开活动的文件)。



打开环境成员的打开文件

1. 在打开共享环境的情况下，如果未显示成员列表，请在 Collaborate (协作) 窗口中展开 Environment Members (环境成员)。
2. 展开要在您的环境中打开其文件的用户的名称。
3. 打开 (双击) 要打开的文件的名称。

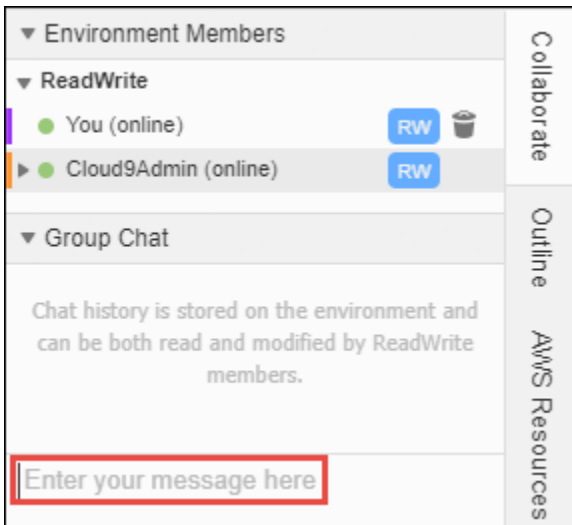


转到环境成员的活动光标

1. 在打开共享环境的情况下，如果未显示成员列表，请在 Collaborate (协作) 窗口中展开 Environment Members (环境成员)。
2. 打开成员名称的上下文 (右键单击) 菜单，然后选择 Show Location (显示位置)。

与其他环境成员聊天

在打开共享环境的情况下，在 Collaborate (协作) 窗口底部的 Enter your message here (在此处输入您的消息) 框中键入您的聊天消息，然后按 Enter。



在共享的环境中查看聊天消息

在打开共享环境的情况下，如果未显示聊天消息列表，请在 Collaborate (协作) 窗口中展开 Group Chat (群聊)。

从共享的环境中删除聊天消息

在打开共享环境的情况下，在 Collaborate (协作) 窗口中，对于 Group Chat (群聊) 中的聊天消息打开上下文 (右键单击) 菜单。然后，选择 Delete Message (删除消息)。

Note

在删除聊天消息时，将从环境中为所有成员删除该消息。

从共享的环境中删除所有聊天消息

在打开共享环境的情况下，在 Collaborate（协作）窗口中，在 Group Chat（群聊）中的任何位置打开上下文（右键单击）菜单。然后，选择 Clear history（清除历史记录）。

Note

在删除所有聊天消息时，将从环境中为所有成员删除这些消息。

更改环境成员的访问角色

1. 打开您拥有并包含要更改其访问角色的成员的环境（如果尚未打开该环境）。有关更多信息，请参阅[打开 AWS Cloud9 中的环境](#)。
2. 如果未显示成员列表，请在 Collaborate（协作）窗口中展开 Environment Members（环境成员）。
3. 执行以下任一操作：
 - 在要更改其访问角色的成员名称旁边，选择 R 或 RW 以使该成员分别成为拥有者或读取/写入成员。
 - 要将读取/写入成员更改为只读，请对于成员名称打开上下文（右键单击）菜单，然后选择 Revoke Write Access（撤消写入访问权限）。
 - 要将只读成员更改为读取/写入，请对于成员名称打开上下文（右键单击）菜单，然后选择 Grant Read+Write Access（授予读取+写入访问权限）。

Note

如果使用该用户成为读取/写入成员，将显示一个对话框，其中包含有关可能给您的 AWS 安全凭证带来风险的信息。除非您信任某个用户代表您在 AWS 中执行操作，否则，不要使用该用户成为读取/写入成员。有关更多信息，请参阅[邀请与环境位于相同账户中的用户](#)中的相关说明。

要使用代码更改环境成员的访问角色，请调用 AWS Cloud9 更新环境成员资格操作，如下所示。

AWS CLI	update-environment-membership
AWS SDK for C++	UpdateEnvironmentMembershipRequest 、 UpdateEnvironmentMembershipResult
AWS SDK for Go	UpdateEnvironmentMembership 、 UpdateEnvironmentMembershipRequest 、 UpdateEnvironmentMembershipWithContext
AWS SDK for Java	UpdateEnvironmentMembershipRequest 、 UpdateEnvironmentMembershipResult
AWS SDK for JavaScript	updateEnvironmentMembership
AWS SDK for .NET	UpdateEnvironmentMembershipRequest 、 UpdateEnvironmentMembershipResponse
AWS SDK for PHP	updateEnvironmentMembership
AWS SDK for Python (Boto)	update_environment_membership
AWS SDK for Ruby	update_environment_membership
AWS Tools for Windows PowerShell	Update-C9EnvironmentMembership
AWS Cloud9 API	UpdateEnvironmentMembership

从共享的环境中删除您的用户

Note

如果您是环境拥有者，则无法从环境中删除您的用户。
 如果从环境中删除您的用户，并不会从 IAM 中删除该用户。

1. 在打开共享环境的情况下，如果未显示成员列表，请在 Collaborate (协作) 窗口中展开 Environment Members (环境成员)。
2. 执行以下任一操作：
 - 在 You (您) 旁边，选择垃圾桶图标。
 - 打开 You (您) 的上下文 (右键单击) 菜单，然后选择 Leave environment (离开环境)。
3. 在出现提示时，选择 Leave (退出)。

要使用代码从共享环境中删除您的用户，请调用 AWS Cloud9 删除成员资格操作，如下所示。

AWS CLI	delete-environment-membership
AWS SDK for C++	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResult
AWS SDK for Go	DeleteEnvironmentMembership 、 DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipWithContext
AWS SDK for Java	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResult
AWS SDK for JavaScript	deleteEnvironmentMembership
AWS SDK for .NET	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResponse
AWS SDK for PHP	deleteEnvironmentMembership
AWS SDK for Python (Boto)	delete_environment_membership
AWS SDK for Ruby	delete_environment_membership
AWS Tools for Windows PowerShell	Remove-C9EnvironmentMembership
AWS Cloud9 API	DeleteEnvironmentMembership

删除另一个环境成员

Note

要从环境中删除您的用户以外的任何成员，您必须使用环境拥有者的凭证登录到 AWS Cloud9。
删除成员并不会从 IAM 中删除该用户。

1. 打开包含要删除的成员的环境（如果尚未打开该环境）。有关更多信息，请参阅[打开 AWS Cloud9 中的环境](#)。
2. 如果未显示成员列表，请在 Collaborate（协作）窗口中展开 Environment Members（环境成员）。
3. 请执行下列操作之一：
 - 在要删除的成员名称旁边，选择垃圾桶图标。
 - 打开要删除的成员名称的上下文（右键单击）菜单，然后选择 Revoke Access（撤销访问权限）。
4. 在出现提示时，选择 Remove Member（删除成员）。

要使用代码从环境中删除成员，请调用 AWS Cloud9 删除环境成员资格操作，如下所示。

AWS CLI	delete-environment-membership
AWS SDK for C++	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResult
AWS SDK for Go	DeleteEnvironmentMembership 、 DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipWithContext
AWS SDK for Java	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResult
AWS SDK for JavaScript	deleteEnvironmentMembership

AWS SDK for .NET	DeleteEnvironmentMembershipRequest 、 DeleteEnvironmentMembershipResponse
AWS SDK for PHP	deleteEnvironmentMembership
AWS SDK for Python (Boto)	delete_environment_membership
AWS SDK for Ruby	delete_environment_membership
AWS Tools for Windows PowerShell	Remove-C9EnvironmentMembership
AWS Cloud9 API	DeleteEnvironmentMembership

环境共享最佳实践

在共享环境时，我们建议使用以下做法：

- 仅邀请您信任的读取/写入成员加入您的环境。
- 对于 EC2 环境，读取/写入成员可以使用环境拥有者的 AWS 访问凭证从环境中对 AWS 服务 进行调用。而不使用他们自己的证书。为防止出现这种情况，环境拥有者可以禁用该环境的 AWS 托管式临时凭证。不过，这也会禁止环境拥有者进行调用。有关更多信息，请参阅 [AWS 托管式临时凭证](#)。
- 启用 AWS CloudTrail 以跟踪您的环境中的活动。有关更多信息，请参阅 [AWS CloudTrail 用户指南](#)。
- 不要使用您的 AWS 账户 根用户创建和共享环境。请改用该账户中的 IAM 用户。有关更多信息，请参阅《IAM 用户指南》中的 [仅限首次访问：您的根用户凭证](#) 以及 [IAM 用户](#)。

移动环境并对 Amazon EBS 卷调整大小或加密

您可以将 AWS Cloud9 开发环境从一个 Amazon EC2 实例移动到另一个 Amazon EC2 实例。例如，您可以执行以下操作之一：

- 从受损的或以意料之外（与正常运行的实例相比）方式运行的 Amazon EC2 实例中传输环境。
- 将环境从现有的实例传输到已安装最新系统更新的实例。
- 增加或减少实例的计算资源（由于该环境在当前实例上已过度使用或使用不充分）。

您可以通过迁移到新的 AWS Cloud9 EC2 环境从一个 AWS Cloud9 支持的 AMI 升级到另一个受支持的 AMI，同时保留项目文件。您可能需要升级到另一个版本的 AMI，因为：

- 当前环境的 AMI 已达到 end-of-life，不再受支持。
- 您需要的软件包在当前 AMI 中已过时。

您还可以调整与用于环境的与 Amazon EC2 实例关联的 Amazon Elastic Block Store (Amazon EBS) 卷的大小。例如，您可以执行以下操作之一或两个操作均执行：

- 增加卷的大小，因为实例上的存储空间不足。
- 减小卷的大小，因为您不希望为不使用的额外存储空间付费。

在移动环境或调整其大小之前，您可用尝试停止环境中一些正在运行的进程或向环境中添加交换文件。有关处理内存不足或 CPU 使用率高的更多信息，请参阅[故障排除](#)。

Note

本主题仅介绍将从环境一个 Amazon EC2 实例移动到另一个实例或调整 Amazon EBS 卷的大小。要调整您自己的其中一个服务器中的环境，或者要更改您自己的其中一个服务器的存储空间，请参考服务器的文档。

最后，您可以加密 Amazon EBS 资源，以确保两者的安全，data-at-rest 以及实例 data-in-transit 与其连接的 EBS 存储之间的安全。

主题

- [移动环境](#)
- [AWS Cloud9 将 EC2 环境移至不同的亚马逊系统映像 \(AMI\)](#)
- [调整环境使用的 Amazon EBS 卷大小](#)
- [加密使用的 Amazon EBS 卷 AWS Cloud9](#)

移动环境

在开始移动过程之前，请注意以下条件：

- 您不能将环境移动到相同类型的 Amazon EC2 实例。在您进行移动时，必须为新实例选择不同的 Amazon EC2 实例类型。

Important

如果您将环境移至其他 Amazon EC2 实例类型，则当前版本 AWS Cloud9 中还必须支持该实例类型 AWS 区域。要查看每个区域中可用的实例类型，请转到 Configure settings (配置设置) 页面，该页面在[使用控制台创建 EC2 环境](#)时显示。您在实例类型部分中的选择由控制台右上角的选择决定。AWS 区域

- 您必须先停止与环境关联的 Amazon EC2 实例，然后才能更改实例类型。当实例停止时，您和任何成员都不能使用与已停止实例关联的环境。
- AWS 将实例移动到新硬件，但是，实例的 ID 不会改变。
- 如果实例在 Amazon VPC 中运行且具有公有 IPv4 地址，则 AWS 释放该地址并为该实例提供新的公有 IPv4 地址。实例会保留其私有 IPv4 地址以及任何弹性 IP 地址或 IPv6 地址。
- 当实例停止时，请计划停机时间。此过程可能需要几分钟时间。

移动环境

1. (可选) 如果新实例类型需要现有实例上未安装的驱动程序，则连接到您的实例并安装这些驱动程序。有关更多信息，请参阅 Amazon EC2 用户指南中的[调整实例大小的兼容性](#)。
2. 关闭当前显示环境的所有 Web 浏览器选项卡。

Important


如果您不关闭当前显示环境的所有 Web 浏览器选项卡，则 AWS Cloud9 可能会干扰此过程的完成。具体而言，AWS Cloud9 可能会在此过程中尝试在错误的时间重启与环境关联的 Amazon EC2 实例。在此过程的最后一步完成之前，该实例必须保持停止状态。

3. 如果你尚未登录 AWS Management Console，请[通过 https://console.aws.amazon.com](https://console.aws.amazon.com) 登录。

我们建议您使用自己的管理员级证书登录。AWS 账户如果您无法执行此操作，请咨询您的 AWS 账户 管理员。


4. 打开 Amazon EC2 控制台。为此，请在 Services (服务) 列表中，选择 EC2。
5. 在 AWS 导航栏中，选择 AWS 区域 包含要移动的环境 (例如，美国东部 (俄亥俄州))。
6. 在服务导航窗格中，展开 Instances (实例)，然后选择 Instances (实例)。

- 在实例列表中，选择与您要移动的环境关联的实例。对于 EC2 环境，实例名称以 `aws-cloud9-` 开头，后跟环境名称。例如，如果环境名为 `my-demo-environment`，实例名称以 `aws-cloud9-my-demo-environment` 开头。
- 如果实例状态未为“已停止”，请选择操作、实例状态、停止。当屏幕上出现提示时，请选择 Yes, Stop (是, 停止)。停止实例可能需要几分钟时间。
- 在 Instance State (实例状态) 为 stopped (已停止) 且实例仍处于选中状态的情况下，选择 Actions (操作) > Instance Settings (实例设置) > Change Instance Type (更改实例类型)。
- 在 Change Instance Type (更改实例类型) 对话框中，在 Instance Type (实例类型) 处选择环境使用的新实例类型。

 Note


如果列表中未显示您所需的实例类型，则说明它与实例的配置不兼容。例如，由于虚拟化类型，该实例可能不兼容。

- (可选) 如果您选择的实例类型支持 EBS 优化，则选择 EBS-optimized (EBS 优化) 以启用 EBS 优化，或取消选择 EBS-optimized (EBS 优化) 以禁用 EBS 优化。

 Note

如果您选择的实例类型默认情况下已经过 EBS 优化，则 EBS-optimized (EBS 优化) 已选中，您无法清除它。

- 选择 Apply (应用) 以接受新设置。

 Note

如果您在本过程的前面部分没有对 Instance Type (实例类型) 选择其他实例类型，则在您选择 Apply (应用) 后不会有任何效果。

- 重新打开环境。有关更多信息，请参阅 [在 AWS Cloud9 中打开环境](#)。

有关上述过程的更多信息，请参阅 Amazon EC2 用户指南中的 [更改实例类型](#)。

AWS Cloud9 将 EC2 环境移至不同的亚马逊系统映像 (AMI)

本主题介绍如何将 AWS Cloud9 EC2 环境从一个亚马逊 Linux AMI 迁移到另一个 AWS Cloud9 受支持的 AMI。

Note

如果您想在更新操作系统版本的情况下将环境迁移到新实例，请参阅[the section called “移动环境”](#)。

您可以使用以下过程之一在环境之间迁移数据：

通过将存档文件下载到本地计算机来移动环境

1. 在同一个可用区中使用不同的基础映像创建新环境：
 - a. 完成[the section called “创建 EC2 环境”](#)部分中的步骤以创建新环境。

Note

在选择平台时，请选择要将环境迁移到的平台。

- b. 默认情况下，创建的环境的容量为 10 GiB。如果您没有足够的空间将档案上传或解压缩到新环境，请完成[the section called “调整环境使用的 Amazon EBS 卷大小”](#)程序中的步骤来调整 Amazon EBS 卷的大小。
2. 在 AWS Cloud9 IDE 中打开要迁移的环境。
3. AWS Cloud9 IDE 加载后，从菜单中选择“文件”>“下载项目”，下载包含环境项目目录内容的存档。
4. 在新环境中打开 AWS Cloud9 IDE。
5. 选择“文件”>“上传本地文件...”上传档案。
6. (可选) 要将旧 .c9 目录备份到 .c9.backup，请在环境终端中运行以下命令：

```
cp .c9 .c9.backup
```

如果您想稍后恢复配置文件，则可能需要这些备份文件。

7. 要解压缩档案，请运行以下命令：

```
tar xzvf <old_environment_name>.tar.gz -C ~/
```

8. 要从项目目录中删除档案，请运行以下命令：

```
rm <old_environment_name>.tar.gz
```

确保新环境按预期运行。

9. 现在，您可以删除旧环境。

使用 Amazon EBS 卷移动环境

如果您无法下载档案，或者生成的档案太大，则可以使用 Amazon EBS 卷进行迁移。此外，此方法还允许您复制位于~/environment目录之外的文件。

1. 关闭在现有环境中打开的所有 AWS Cloud9 IDE 选项卡。
2. 完成以下步骤以停止现有实例：
 - a. 在 AWS Cloud9 控制台中，选择要导航的环境以查看其详细信息。
 - b. 在环境详细信息页面的 EC2 实例选项卡下，选择管理 EC2 实例。
 - c. 在 EC2 控制台中，选择要导航到实例详细信息的实例。
 - d. 确保实例状态设置为“已停止”。如果不是，请从“实例状态”下拉列表中选择“停止实例”。出现提示时，选择“停止”。停止实例可能需要几分钟时间。
3. 在同一个可用区中使用不同的基础映像创建新环境：
 - a. 完成[the section called “创建 EC2 环境”](#)部分中的步骤以创建新环境。

Note

在选择平台时，请选择要将环境迁移到的平台。

- b. 默认情况下，创建的环境的容量为 10 GiB。如果您没有足够的空间将文件从源卷移动到新环境，请完成[the section called “调整环境使用的 Amazon EBS 卷大小”](#)过程中的步骤来调整 Amazon EBS 卷的大小。
4. 完成以下步骤，将该卷与现有实例分离：
 - a. 在实例摘要页面上，选择存储选项卡并选择卷。所选卷的设备名称必须与根设备详细信息部分的根设备名称中指定的名称相同。
 - b. 在卷详细信息页面上，选择操作 > 分离卷。

- c. 成功分离卷后，选择操作 > 附加卷，然后从下拉列表中查找并选择新环境的实例。您选择的 Amazon EC2 实例的名称必须包含前缀aws-c1oud9为的 AWS Cloud9 环境名称。
5. 在新环境中打开 AWS Cloud9 IDE。
 6. 环境加载后，要识别新连接的卷的设备，请在终端中运行以下命令：

```
lsblk
```

在以下示例输出中，根nvme0n1设备的分区nvme0n1p1已挂载，因此还必须装载该nvme1n1p1分区。其设备的完整路径是/dev/nvme1n1p1：

```
Admin:~/environment $ lsblk
NAME                MAJ:MIN RM  SIZE RO  TYPE MOUNTPOINTS
nvme0n1             259:0   0  10G  0  disk
##nvme0n1p1        259:2   0  10G  0  part /
##nvme0n1p127     259:3   0   1M  0  part
##nvme0n1p128     259:4   0  10M  0  part /boot/efi
nvme1n1             259:1   0  10G  0  disk
##nvme1n1p1        259:5   0  10G  0  part
##nvme1n1p128     259:6   0   1M  0  part
```

Note

当您在终端中运行此命令时，输出会有所不同。

7. 在环境终端中完成以下步骤以挂载现有卷：
 - a. 要创建用于挂载卷分区的临时目录，请运行以下命令：

```
MOUNT_POINT=$(mktemp -d)
```

- b. 根据lsblk命令的示例输出，指定要安装的设备的路径：

```
MOUNT_DEVICE=/dev/nvme1n1p1
```

Note

当您在终端中运行此命令时，输出会有所不同。

- c. 要挂载现有卷，请运行以下命令：

```
sudo mount $MOUNT_DEVICE $MOUNT_POINT
```

- d. 完成以下步骤以验证现有卷是否已正确装入：

- i. 要确保输出中包含该卷，请运行以下命令：

```
df -h
```

- ii. 要验证卷中的内容，请运行以下命令：

```
ls $MOUNT_POINT/home/ec2-user/environment/
```

8. (可选) 要将旧 .c9 目录备份到 .c9.backup，请在环境终端中运行以下命令：

```
cp .c9 .c9.backup
```

如果您想稍后恢复配置文件，则可能需要这些备份文件。

9. 要从现有卷中复制旧环境，请运行以下命令：

```
cp -R $MOUNT_POINT/home/ec2-user/environment ~
```

Note

如果需要，您还可以使用上述命令复制环境目录之外的文件或目录。

确保新环境按预期运行。

10. 要卸载以前的设备，请运行以下两个命令之一：

```
sudo umount $MOUNT_DEVICE
```

```
sudo umount $MOUNT_POINT
```

11. 从“操作”下拉列表中选择“分离卷”，以分离您在步骤 3 中连接的卷。

12. 现在，您可以删除旧环境及其音量。

Note

由于该卷不再连接到环境的 Amazon EC2 实例，因此您需要手动将其删除。您可以通过在卷详细信息页面上选择删除来执行此操作。

调整环境使用的 Amazon EBS 卷大小

1. 针对您要调整其大小的 Amazon EBS 卷，打开与 Amazon EC2 实例关联的环境。
2. 在环境的 AWS Cloud9 IDE 中，创建一个包含以下内容的文件，然后使用扩展名保存该文件.sh (例如，resize.sh)。

备注

此脚本适用于连接到运行 AL2023、Amazon Linux 2、Amazon Linux 或 Ubuntu Server 并配置为使用 IMDSv2 的 EC2 实例的 Amazon EBS 卷。
该脚本还调整在基于 Nitro 的实例上显示为 NVMe 块设备的 Amazon EBS 卷的大小。有关基于 Nitro 系统的实例列表，请参阅 Amazon EC2 用户指南中的[Nitro 基于实例](#)的内容。

```
#!/bin/bash

# Specify the desired volume size in GiB as a command line argument. If not
# specified, default to 20 GiB.
SIZE=${1:-20}

# Get the ID of the environment host Amazon EC2 instance.
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 60")
INSTANCEID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" -v
http://169.254.169.254/latest/meta-data/instance-id 2> /dev/null)
REGION=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/placement/region 2> /dev/null)

# Get the ID of the Amazon EBS volume associated with the instance.
VOLUMEID=$(aws ec2 describe-instances \
--instance-id $INSTANCEID \
--query "Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.VolumeId" \
```

```
--output text \  
--region $REGION)  
  
# Resize the EBS volume.  
aws ec2 modify-volume --volume-id $VOLUMEID --size $SIZE  
  
# Wait for the resize to finish.  
while [ \  
  "$(aws ec2 describe-volumes-modifications \  
    --volume-id $VOLUMEID \  
    --filters Name=modification-state,Values="optimizing","completed" \  
    --query "length(VolumesModifications)"\  
    --output text)" != "1" ]; do  
  sleep 1  
done  
  
# Check if we're on an NVMe filesystem  
if [[ -e "/dev/xvda" && $(readlink -f /dev/xvda) = "/dev/xvda" ]]  
then  
# Rewrite the partition table so that the partition takes up all the space that it  
  can.  
  sudo growpart /dev/xvda 1  
# Expand the size of the file system.  
# Check if we're on AL2 or AL2023  
STR=$(cat /etc/os-release)  
SUBAL2="VERSION_ID=\"2\""  
SUBAL2023="VERSION_ID=\"2023\""  
if [[ "$STR" == *"$SUBAL2"* || "$STR" == *"$SUBAL2023"* ]]  
then  
  sudo xfs_growfs -d /  
else  
  sudo resize2fs /dev/xvda1  
fi  
  
else  
# Rewrite the partition table so that the partition takes up all the space that it  
  can.  
  sudo growpart /dev/nvme0n1 1  
  
# Expand the size of the file system.  
# Check if we're on AL2 or AL2023  
STR=$(cat /etc/os-release)  
SUBAL2="VERSION_ID=\"2\""  
SUBAL2023="VERSION_ID=\"2023\""
```

```
if [[ "$STR" == *"$SUBAL2"* || "$STR" == *"$SUBAL2023"* ]]
then
  sudo xfs_growfs -d /
else
  sudo resize2fs /dev/nvme0n1p1
fi
fi
```

3. 从 IDE 中的终端会话中，切换到包含 `resize.sh` 文件的目录。然后，运行以下任一命令，同时将 `20` 替换为要将 Amazon EBS 卷调整至的期望大小（以 GiB 为单位）：

- ```
bash resize.sh 20
```

- ```
chmod +x resize.sh
./resize.sh 20
```

加密使用的 Amazon EBS 卷 AWS Cloud9

Amazon EBS 加密可对以下数据进行加密：

- 卷中的静态数据
- 在卷和实例之间移动的所有数据
- 从卷创建的所有快照
- 从这些快照创建的所有卷

对于 AWS Cloud9 EC2 开发环境使用的 Amazon EBS 卷，您有两个加密选项：

- 原定设置加密 – 您可以配置 AWS 账户 对您创建的新 EBS 卷和快照副本进行加密。原定设置情况下，在 AWS 区域级别启用加密。因此，您不能为该区域中的单个卷或快照启用加密。此外，Amazon EBS 对您启动实例时创建的卷进行加密。因此，在创建 EC2 环境之前，必须启用此设置。有关更多信息，请参阅 Amazon EC2 用户指南中的[默认加密](#)。
- 加密 EC2 环境使用的现有 Amazon EBS 卷 – 您可以加密已为 EC2 实例创建的指定 Amazon EBS 卷。此选项涉及使用 AWS Key Management Service (AWS KMS) 来管理对加密卷的访问权限。有关相关步骤，请参阅 [对 AWS Cloud9 使用的现有 Amazon EBS 卷进行加密](#)。

⚠ Important

如果您的 AWS Cloud9 IDE 使用默认加密的 Amazon EBS 卷，则 AWS Identity and Access Management 服务相关角色 AWS Cloud9 需要访问这些 EBS 卷 AWS KMS key 的。如果未提供访问权限，AWS Cloud9 IDE 可能无法启动，调试可能会很困难。

要提供访问权限，请将服务相关角色添加到 Amazon EBS 卷使用的 KMS 密钥中。AWS Cloud9AWSServiceRoleForAWSCloud9 有关此任务的更多信息，请参阅 AWS 规范指导模式中的 [创建使用 Amazon EBS 卷和默认加密的 AWS Cloud9 IDE](#)。

对 AWS Cloud9 使用的现有 Amazon EBS 卷进行加密

加密现有 Amazon EBS 卷 AWS KMS 需要使用创建 KMS 密钥。创建要替换的卷的快照后，您可以使用 KMS 密钥对快照的副本进行加密。

接下来，使用该快照创建加密卷。然后，您可以通过将未加密卷从 EC2 实例中分离并附加已加密卷来替换未加密卷。

最后，您必须更新客户托管式密钥的密钥策略，以启用对 AWS Cloud9 服务角色的访问权限。

📘 Note

以下过程重点介绍使用客户托管式密钥对卷进行加密。您也可以在您的账户 AWS 服务中 AWS 托管式密钥使用 for a。Amazon EBS 的别名为 aws/ebs。如果为加密选择此默认选项，请在创建客户托管式密钥时跳过步骤 1。此外，跳过在其中更新密钥策略的步骤 8。这是因为您无法更改的密钥策略 AWS 托管式密钥。

对现有 Amazon EBS 卷进行加密

1. 在 AWS KMS 控制台中，创建对称 KMS 密钥。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的 [创建对称 KMS 密钥](#)。
2. 在 Amazon EC2 控制台中，停止由环境使用的 Amazon EBS 支持的实例。您可以 [使用控制台或命令行停止实例](#)。
3. 在 Amazon EC2 控制台的导航窗格中，选择 Snapshots (快照) 来 [为您想要加密的现有卷创建快照](#)。
4. 在 Amazon EC2 控制台的导航窗格中，选择 Snapshots (快照) 来 [复制快照](#)。在 Copy snapshot (复制快照) 对话框中，执行以下操作以启用加密：

- 选择 Encrypt this snapshot (加密此快照) 。
- 在 Master Key (主密钥) 中 , 选择您之前创建的 KMS 密钥。 (如果您使用的是 AWS 托管式密钥 , 请保留 (默认) aws/ebs 设置。)

5. [从已加密快照创建新卷。](#)

Note

从加密快照创建的新 Amazon EBS 卷会自动加密。

6. [将旧的 Amazon EBS 卷](#)与 Amazon EC2 实例分离。
7. [将新的加密卷](#)附到 Amazon EC2 实例上。
8. [使用 AWS Management Console 默认视图、策略视图或 AWS KMS API 更新 KMS 密钥的密钥 AWS Management Console 策略。](#)添加以下密钥策略声明以允许 AWS Cloud9 服务访问 KMS 密钥。AWSServiceRoleForAWSCloud9

Note

如果您使用的是 AWS 托管式密钥 , 请跳过此步骤。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:{Partition}:iam::{AccountId}:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
```

```
"Principal": {
  "AWS": "arn:{Partition}:iam::{AccountId}:role/aws-service-role/
cloud9.amazonaws.com/AWSServiceRoleForAWSCloud9"
},
"Action": [
  "kms:CreateGrant",
  "kms:ListGrants",
  "kms:RevokeGrant"
],
"Resource": "*",
"Condition": {
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
}
```

9. 重新启动 Amazon EC2 实例。有关重启 Amazon EC2 实例的更多信息，请参阅[停止并启动您的实例](#)。

在 AWS Cloud9 中删除环境

为了防止在您的 AWS 账户中持续产生与不再使用的 AWS Cloud9 开发环境相关的任何费用，请删除该环境。

- [使用控制台删除环境](#)
- [使用代码删除环境](#)

使用控制台删除环境

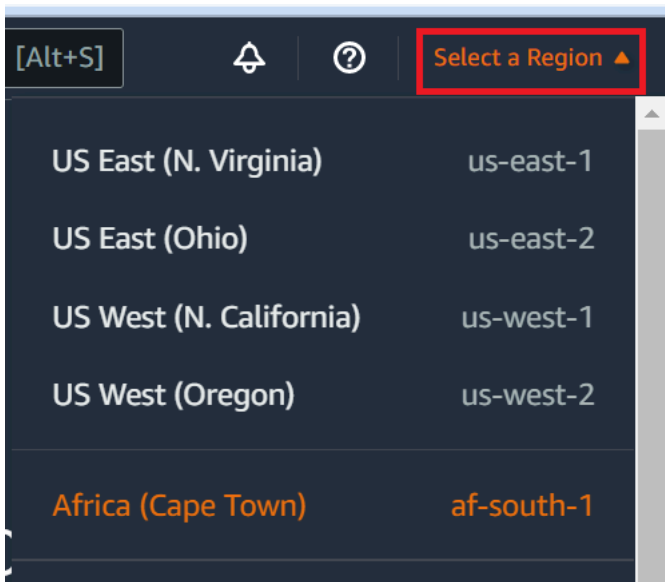
Warning

在删除环境时，AWS Cloud9 将永久删除环境。这包括永久删除所有相关设置、用户数据和未提交的代码。删除的环境无法恢复。

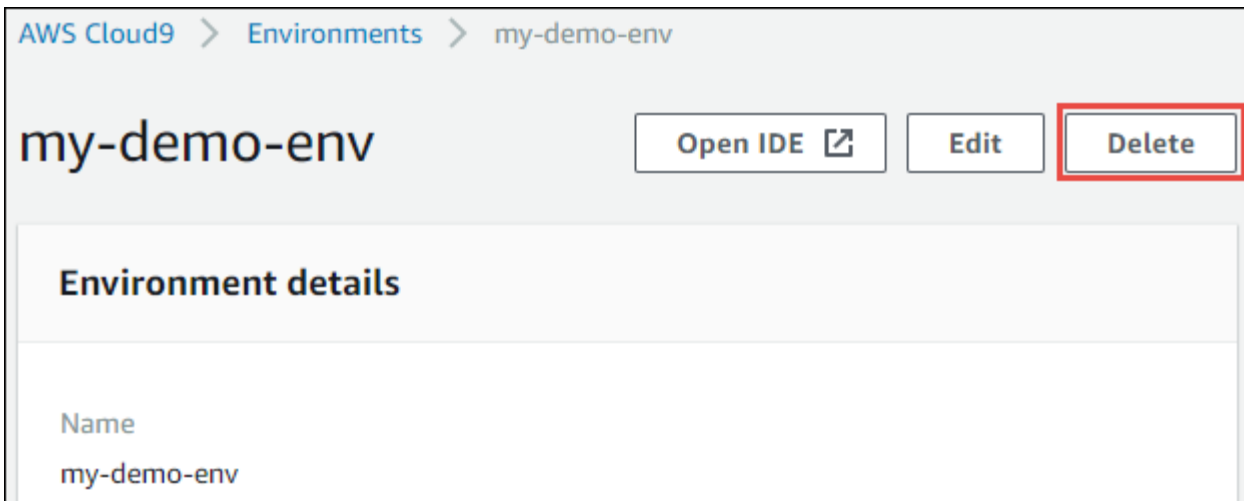
1. 登录到 AWS Cloud9 控制台：

- 如果您是使用 AWS 账户的唯一用户，或者您是单个 AWS 账户中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。

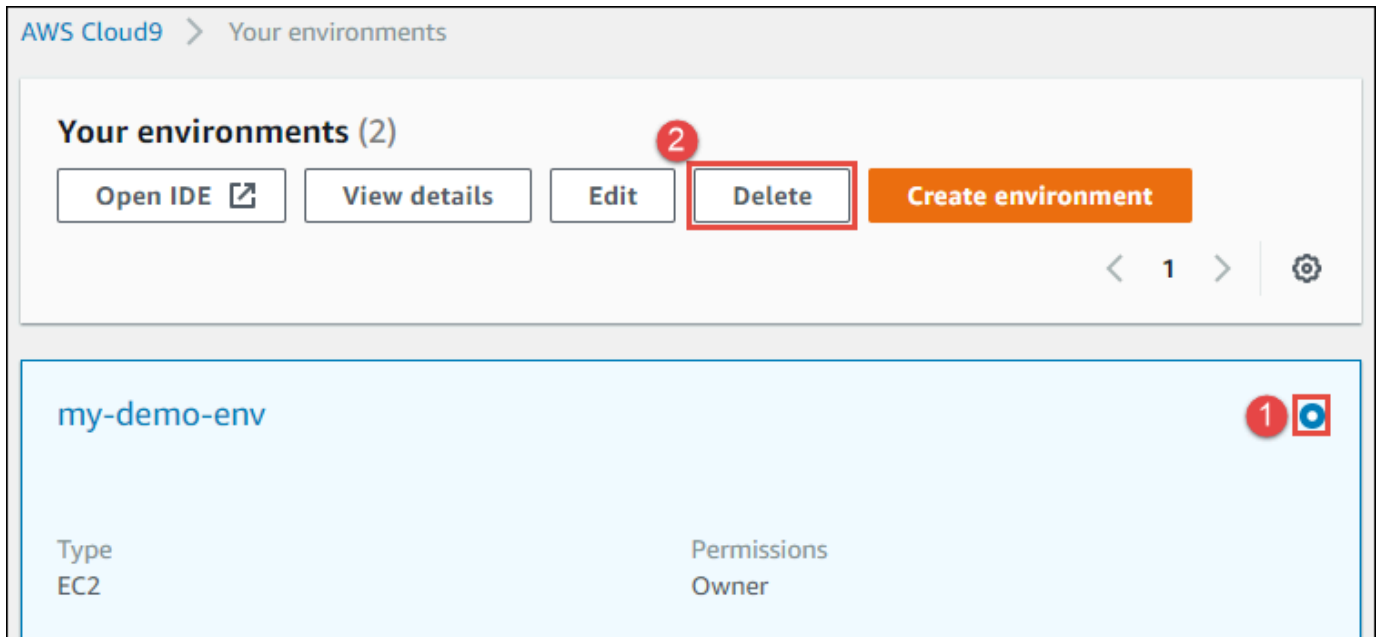
- 如果您的企业使用 AWS IAM Identity Center，请联系 AWS 账户 管理员以获取登录说明。
2. 在顶部导航栏中，选择环境所在的 AWS 区域。



3. 在环境列表中，对于要删除的环境，请执行以下操作之一。
 - 选择环境的选项卡标题。然后，在下一页选择 Delete (删除)。



- 选择环境的卡，然后选择 Delete (删除) 按钮。



4. 在 Delete (删除) 对话框中, 键入 Delete, 然后选择 Delete (删除)。

- EC2 环境

AWS Cloud9 还会终止连接到该环境的 Amazon EC2 实例。

Note

如果账户删除失败, 控制台网页的顶部将显示横幅。此外, 环境的卡片 (如果存在) 表示环境删除失败。

- SSH 环境

如果环境连接到一个 Amazon EC2 实例, AWS Cloud9 不会终止该实例。如果您稍后不终止该实例, 可能会一直对您的 AWS 账户收取与该实例相关的 Amazon EC2 费用。

5. 如果环境是一个 SSH 环境, AWS Cloud9 将在云计算实例或您自己 (连接到该环境) 的服务器上遗留一个隐藏的子目录。如果您要删除它, 您现在可以安全地删除该子目录。该子目录的名称为 .c9。此子目录位于在创建环境时指定的 Environment path (环境路径) 目录中。

如果控制台中未显示您的环境, 请尝试执行以下一个或多个操作来显示它。

- 在 Environments (环境) 页面的下拉菜单栏中, 选择以下一项或多项。
 - 选择 My environments (我的环境) 以显示您的 AWS 实体在选定的 AWS 区域和 AWS 账户中拥有的所有环境。

- 选择 Shared with me (已与我共享) 以显示您的 AWS 实体在所选 AWS 区域 和 AWS 账户 中受邀加入的所有环境。
- 选择 All account environments (所有账户环境) 以显示您的 AWS 实体在选定的 AWS 区域 和 AWS 账户 中有权显示的所有环境。
- 如果认为您是某个环境的成员，但在 Shared with you (已与您共享) 列表中未显示该环境，请与该环境的拥有者联系。
- 在顶部导航栏中，选择不同的 AWS 区域。

使用代码删除环境

Warning

在删除环境时，AWS Cloud9 将永久删除环境。这包括永久删除所有相关设置、用户数据和未提交的代码。删除的环境无法恢复。

要在 AWS Cloud9 中使用代码删除环境，请调用 AWS Cloud9 删除操作，如下所示。

AWS CLI	delete-environment
AWS SDK for C++	DeleteEnvironmentRequest 、 DeleteEnvironmentResult
AWS SDK for Go	DeleteEnvironment 、 DeleteEnvironmentRequest 、 DeleteEnvironmentWithContext
AWS SDK for Java	DeleteEnvironmentRequest 、 DeleteEnvironmentResult
AWS SDK for JavaScript	deleteEnvironment
AWS SDK for .NET	DeleteEnvironmentRequest 、 DeleteEnvironmentResponse
AWS SDK for PHP	deleteEnvironment
AWS SDK for Python (Boto)	delete_environment

AWS SDK for Ruby	delete_environment
AWS Tools for Windows PowerShell	Remove-C9Environment
AWS Cloud9 API	DeleteEnvironment

使用 AWS Cloud9 集成开发环境 (IDE)

集成开发环境 (IDE) 提供一组代码编写生产力工具，如源代码编辑器、调试器和生成工具。

Important

我们建议采用以下最佳实践来使用 AWS Cloud9：

- 使用 source control and backup (源代码控制和备份) 以经常备份您的环境。AWS Cloud9 不执行自动备份。
- 对环境执行定期的 updates of software (软件更新)。AWS Cloud9 不执行自动软件更新。
- 在您的 AWS 账户中使用 Turn on AWS CloudTrail (开启 AWS CloudTrail) 以跟踪您环境中的活动。有关更多信息，请参阅[使用 AWS Cloud9 记录 AWS CloudTrail API 调用](#)。
- 仅与 trusted users (可信用户) 共享您的环境。共享您的环境可能会让您的 AWS 访问凭证面临危险。有关更多信息，请参阅[在 AWS Cloud9 中使用共享环境](#)。

阅读其中一个或多个主题，了解如何使用 AWS Cloud9 IDE。

主题

- [了解 AWS Cloud9 IDE](#)
- [AWS Cloud9 集成开发环境 \(IDE\) 中的语言支持](#)
- [AWS Cloud9 集成开发环境 \(IDE\) 中的增强型语言支持](#)
- [AWS Cloud9 集成开发环境 \(IDE\) 的菜单栏命令参考](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中查找和替换文本](#)
- [预览 AWS Cloud9 集成开发环境 \(IDE\) 中的文件](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中预览正在运行的应用程序](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用文件修订](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用图像文件](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用生成器、运行程序和调试程序](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用自定义环境变量](#)
- [在 AWS Cloud9 集成式开发环境 \(IDE \) 中使用项目设置](#)
- [使用 AWS Cloud9 IDE 中的用户设置](#)

- [在 AWS Cloud9 集成式开发环境 \(IDE \) 中使用 AWS 项目和用户设置](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用键绑定](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用主题](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中管理初始化脚本](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 MacOS 默认键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 MacOS Vim 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 MacOS Emacs 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 MacOS Sublime 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 Windows/Linux 默认键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 Windows/Linux Vim 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 Windows/Linux Emacs 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的 Windows/Linux Sublime 键绑定参考](#)
- [用于 AWS Cloud9 集成开发环境 \(IDE\) 的命令参考](#)

了解 AWS Cloud9 IDE

本主题提供了对 AWS Cloud9 集成开发环境 (IDE) 的基本导览。为了充分利用此导览，请按照以下顺序逐步操作。

主题

- [先决条件](#)
- [步骤 1：菜单栏](#)
- [第 2 步：控制面板](#)
- [步骤 3：“Environment \(环境 \)”窗口](#)
- [步骤 4：编辑器、选项卡和窗格](#)
- [第 5 步：控制台](#)
- [步骤 6：打开文件部分](#)
- [第 7 步：间距](#)
- [步骤 8：状态栏](#)
- [步骤 9：“Outline \(大纲 \)”窗口](#)

- [步骤 10：“Go \(转到 \)”窗口](#)
- [步骤 11：“Immediate \(立即运行 \)”选项卡](#)
- [步骤 12：进程列表](#)
- [第 13 步：首选项](#)
- [第 14 步：终端](#)
- [步骤 15：“Debugger \(调试程序 \)”窗口](#)
- [最终思考](#)

先决条件

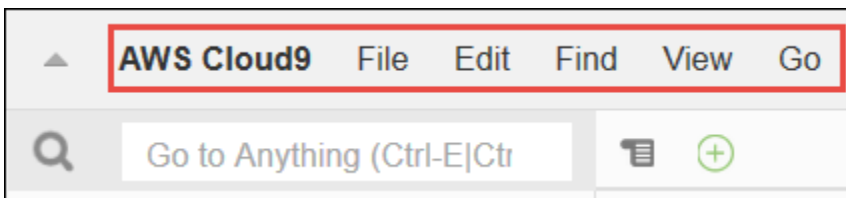
要继续本导览，您必须有一个 AWS 账户和一个开放的 AWS Cloud9 开发环境。要了解如何执行这些操作，您可以执行 [入门：AWS Cloud9 的基本教程](#) 中的步骤。您也可以探索单独的相关主题，例如 [设置 AWS Cloud9](#) 和 [在环境中工作 AWS Cloud9](#)。

Warning

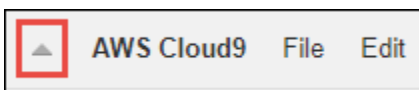
拥有 AWS Cloud9 开发环境可能会导致向您的 AWS 账户收费。如果您使用 EC2 环境，其中包括可能对 Amazon EC2 收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

步骤 1：菜单栏

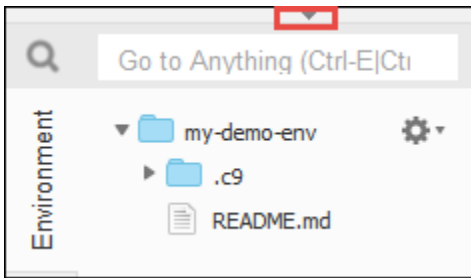
IDE 顶部边缘的菜单栏 中包含用于处理文件和编码以及用于更改 IDE 设置的常用命令。还可以通过菜单栏预览和运行代码。



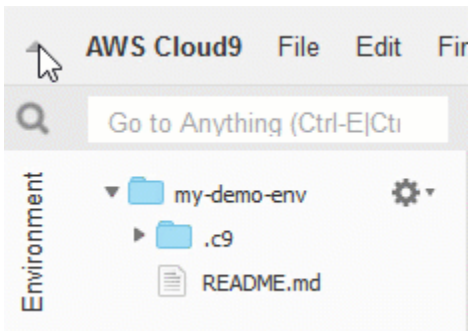
通过选择菜单栏边缘的箭头可以隐藏它，如下所示。



通过选择菜单栏之前位置中间的箭头，可以重新显示菜单栏，如下所示。



将结果与以下内容进行比较。



在本教程接下来的几节中，您可以使用 IDE 处理一组文件。要设置这些文件，请依次选择 File (文件)、New File (新建文件)。

接下来，将以下文本复制到 Untitled1 编辑器选项卡中。

```
fish.txt
-----
A fish is any member of a group of organisms that consist of
all gill-bearing aquatic craniate animals that lack limbs with
digits. They form a sister group to the tunicates, together
forming the olfactores. Included in this definition are
lampreys and cartilaginous and bony fish as well as various
extinct related groups.
```

要保存文件，请依次选择 File (文件)、Save (保存)。将该文件命名为 `fish.txt`，然后选择 Save (保存)。

按以上说明重复操作，使用以下内容将第二个文件保存为 `cat.txt`。

```
cat.txt
-----
The domestic cat is a small, typically furry, carnivorous mammal.
They are often called house cats when kept as indoor pets or
simply cats when there is no need to distinguish them from
```

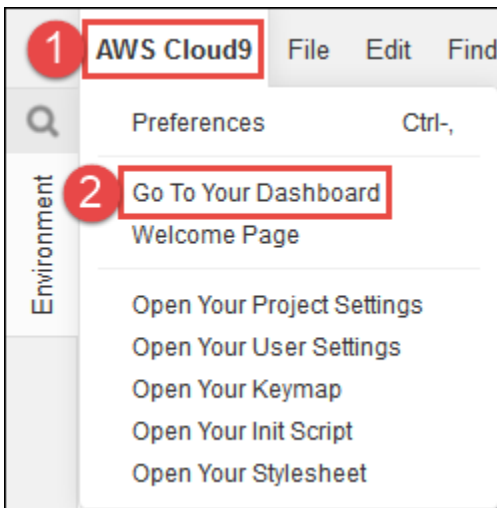
other felids and felines. Cats are often valued by humans for companionship and for their ability to hunt.

在 IDE 中，通常可以通过多种方法执行操作。例如，如果要隐藏菜单栏，除了选择其边缘的箭头外，还可以选择 View (视图)、Menu Bar (菜单栏)。如果要创建新文件，除了选择 File (文件) > New File (新建文件) 外，还可以按 Alt-N (在 Windows/Linux 中) 或 Control-N (在 MacOS 中)。为减少本教程的长度，我们只介绍一种操作方法。随着您逐渐熟悉 IDE，您可以随意探索并找出最适合您的方法。

第 2 步：控制面板

控制面板 让您可以快速访问您的各个环境。在控制面板中，您可以创建、打开和更改环境的设置。

要打开控制面板，在菜单栏上选择 AWS Cloud9 > Go To Your Dashboard (转到您的控制面板)。



要查看环境的设置，在 my-demo-environment 卡中选择标题。要返回控制面板，请使用 Web 浏览器的后退按钮或名为 Environments (环境) 的面包屑导航。

为您的环境打开 IDE，请选择 Open IDE (打开 IDE) 中的 my-demo-environment 卡。

Note

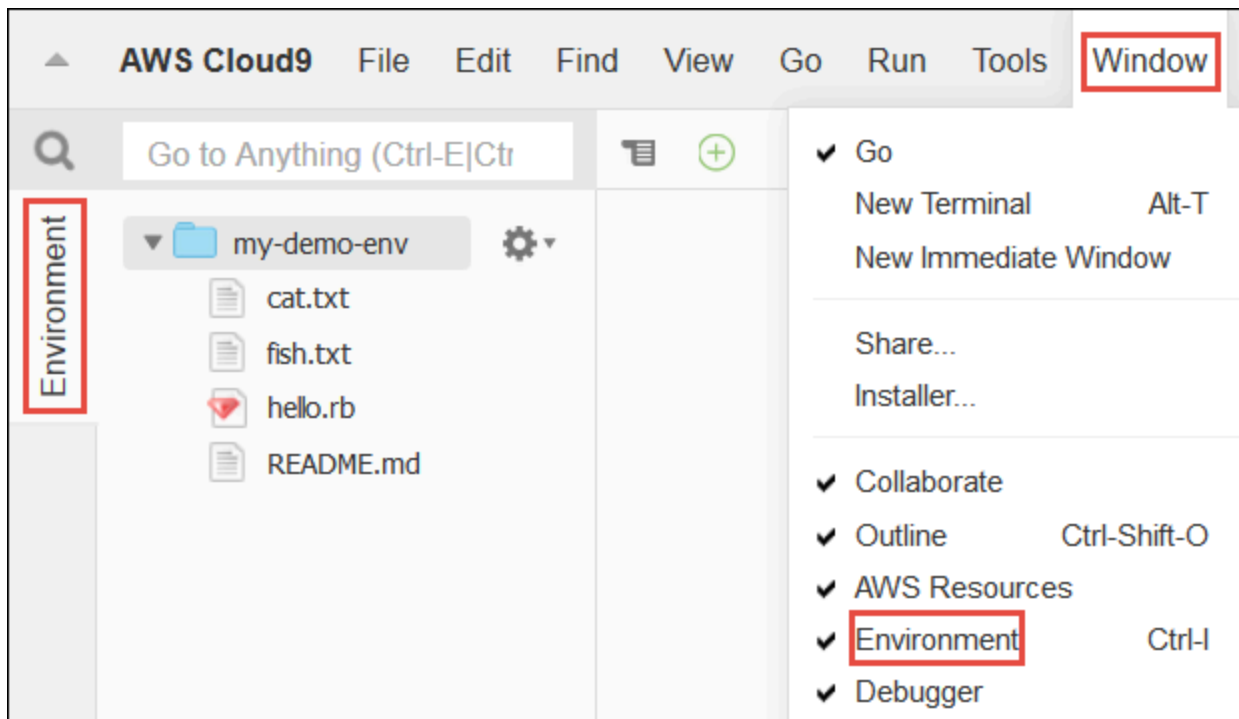
需要一些时间，才能再次显示 IDE。

步骤 3：“Environment (环境)”窗口

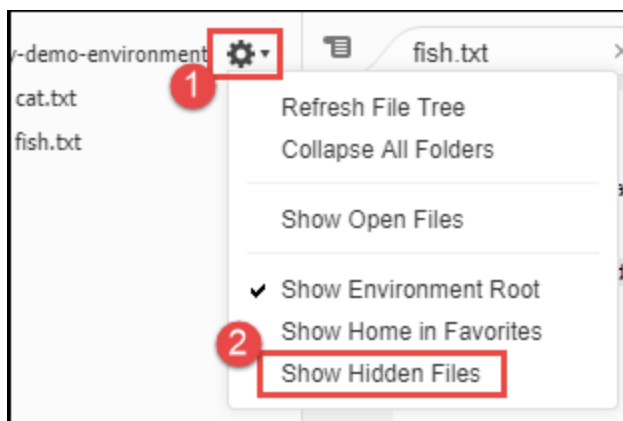
Environment (环境) 窗口中显示环境中的文件夹和文件的列表。还可以显示不同类型的文件，如隐藏文件。

要显示或隐藏 Environment (环境) 窗口的内容, 请选择 Environment (环境) 按钮。

要显示或隐藏 Environment (环境) 窗口和 Environment (环境) 按钮, 请在菜单栏上依次选择 Window (窗口)、Environment (环境)。



要显示或隐藏隐藏的文件, 请在 Environment (环境) 窗口中, 选择齿轮图标, 然后选择 Show Hidden Files (显示隐藏的文件)。



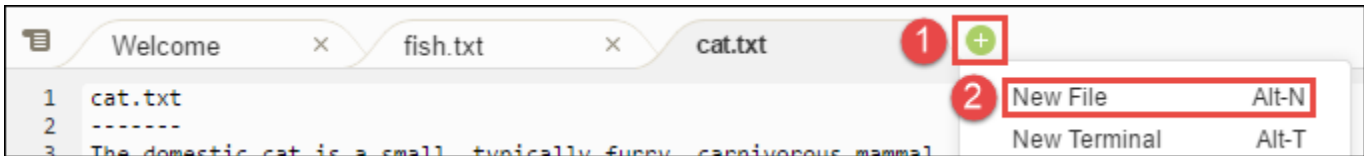
步骤 4：编辑器、选项卡和窗格

在编辑器中可以执行编写代码、运行终端会话和更改 IDE 设置等操作。打开文件的每个实例、终端会话等均由选项卡表示。选项卡可以分组到窗格中。选项卡显示在其窗格边缘。

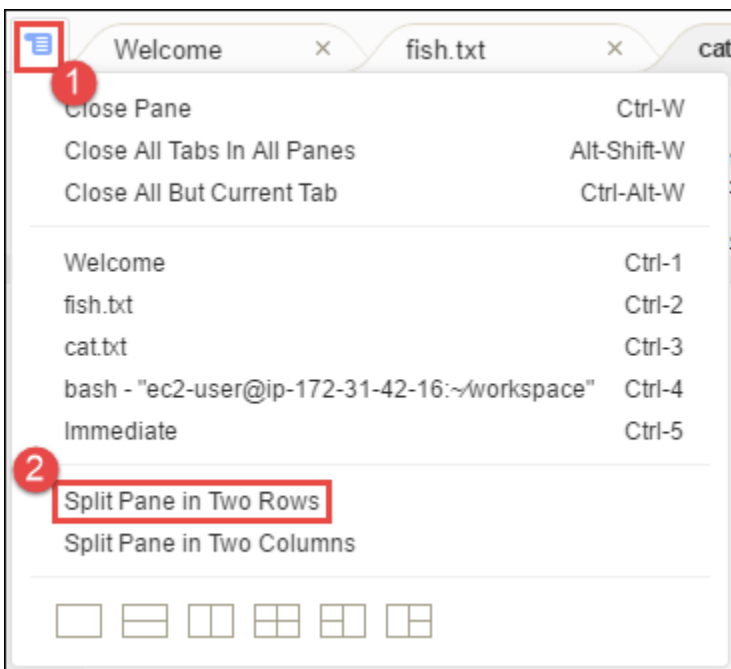


要显示或隐藏选项卡，请在菜单栏上选择 View (视图) > Tab Buttons (选项卡按钮)。

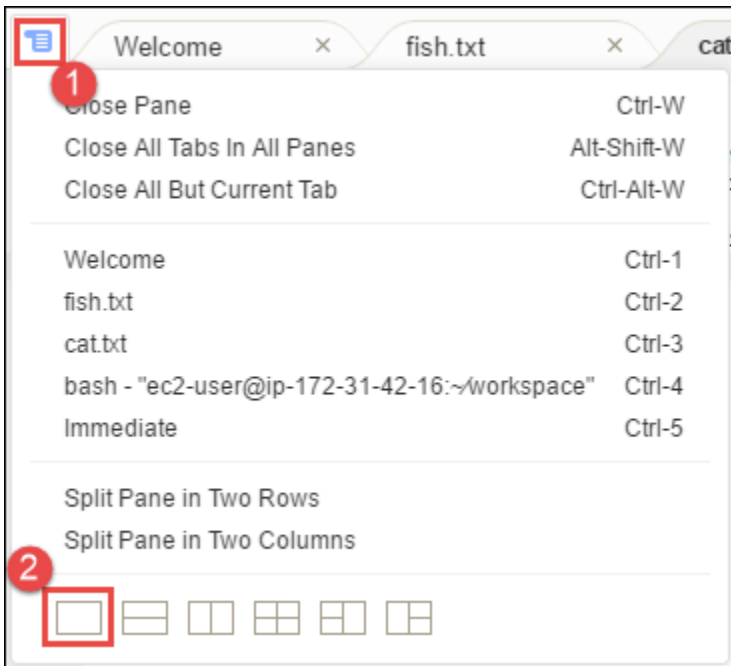
要打开新的选项卡，请选择选项卡所在行边缘的 + 图标。然后选择一个可用的命令，例如 New File (新建文件)，如下所示。



要显示两个窗格，请选择位于选项卡所在行边缘、看上去像下拉菜单的图标。然后选择 Split Pane in Two Rows (将窗格拆分为两排)，如下所示。

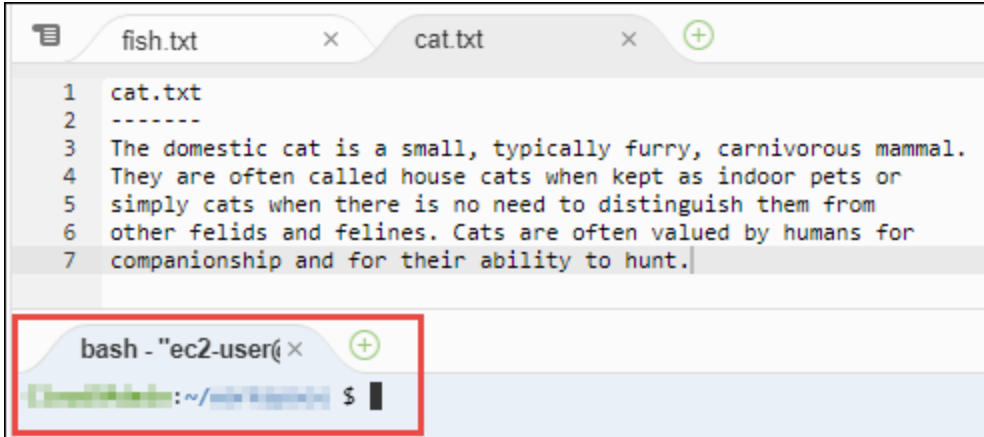


要返回到单个窗格中，请重新选择下拉菜单图标，然后选择单个方形图标，如下所示。



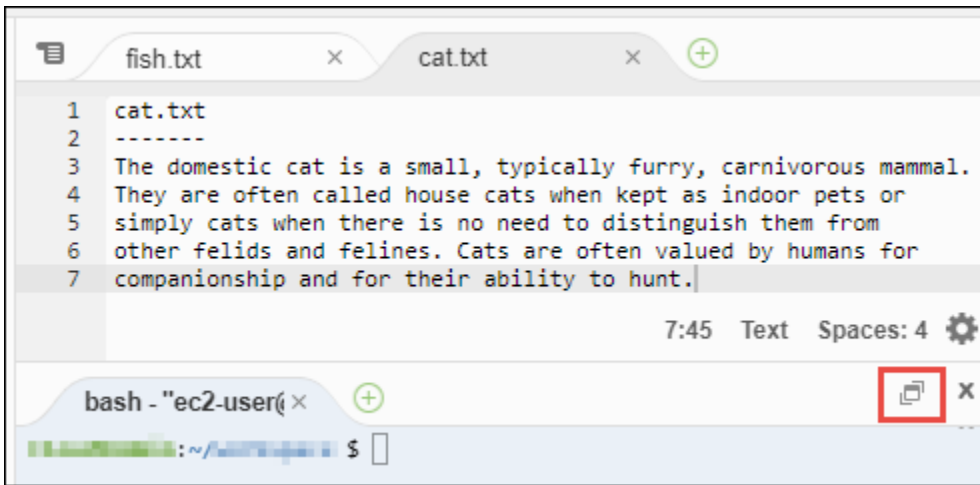
第 5 步：控制台

控制台是可创建和管理选项卡的另一个位置。默认情况下，它包含一个“Terminal (终端)”选项卡，但也可以包含其他类型的选项卡。



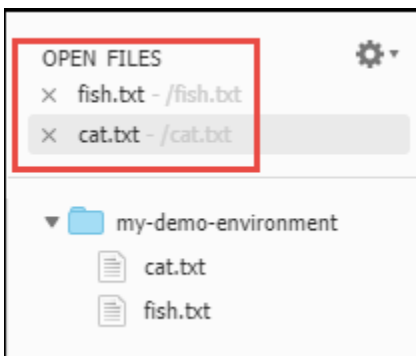
要显示或隐藏控制台，请在菜单栏上选择 View (视图) > Console (控制台)。

要展开或收缩控制台，请选择控制台边缘的调整大小图标，如下所示。



步骤 6：打开文件部分

Open Files (打开文件) 部分显示当前在编辑器中打开的所有文件的列表。Open Files (打开文件) 是 Environment (环境) 窗口的一部分。

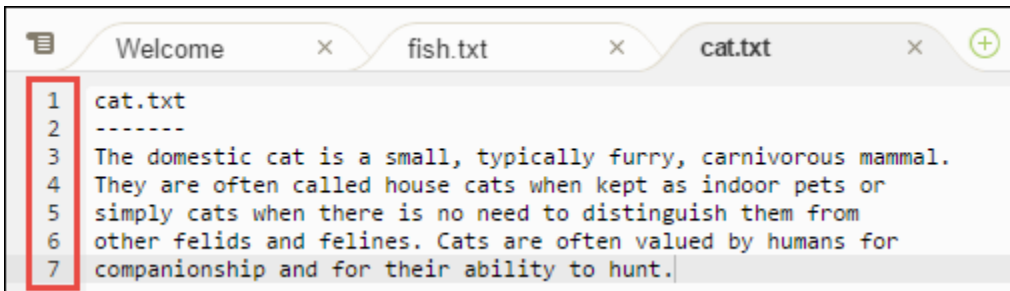


要显示或隐藏 Open Files (打开文件) 部分，请在菜单栏上选择 View (视图) > Open Files (打开文件)。

要在打开文件部分之间切换，请从列表中选择感兴趣的文件。

第 7 步：间距

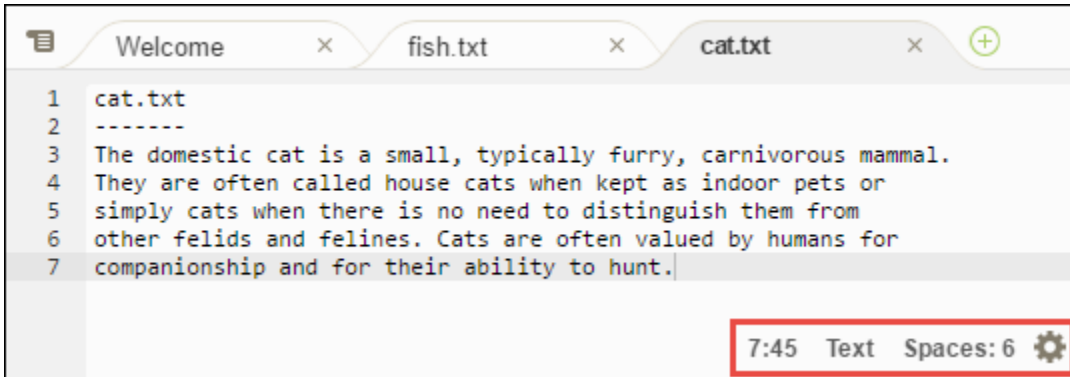
间距位于编辑器中各个文件的边缘，当您处理文件时，其中显示行号和上下文符号等内容。



要显示或隐藏间距，请在菜单栏上选择 View (视图) > Gutter (间距)。

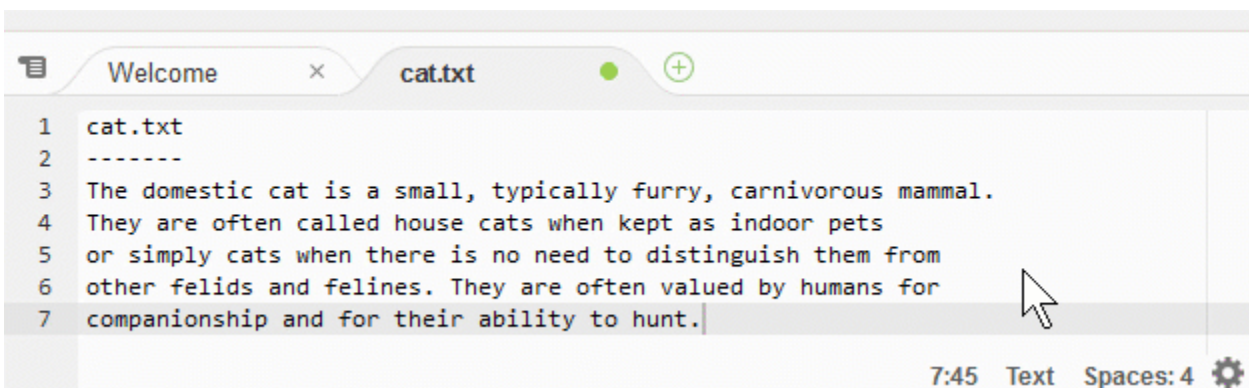
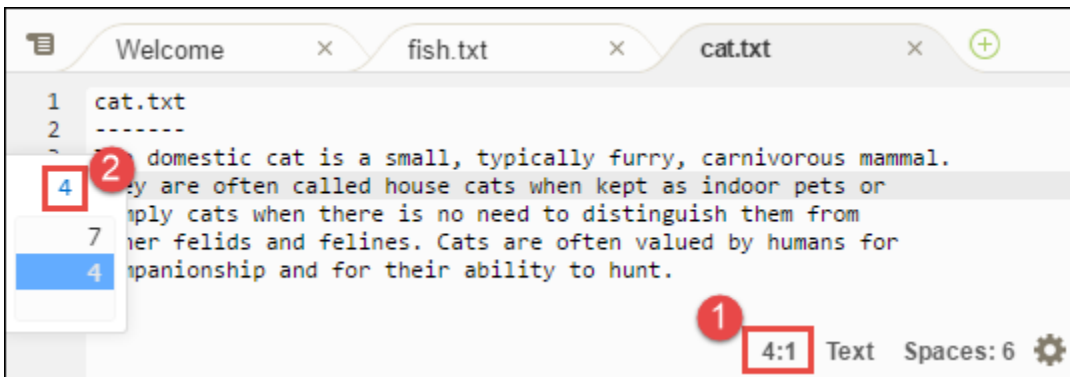
步骤 8：状态栏

状态栏位于编辑器中各个文件的边缘，其中显示行号、字号、文件类型首选项、空格和选项卡设置以及相关的编辑器设置等内容。

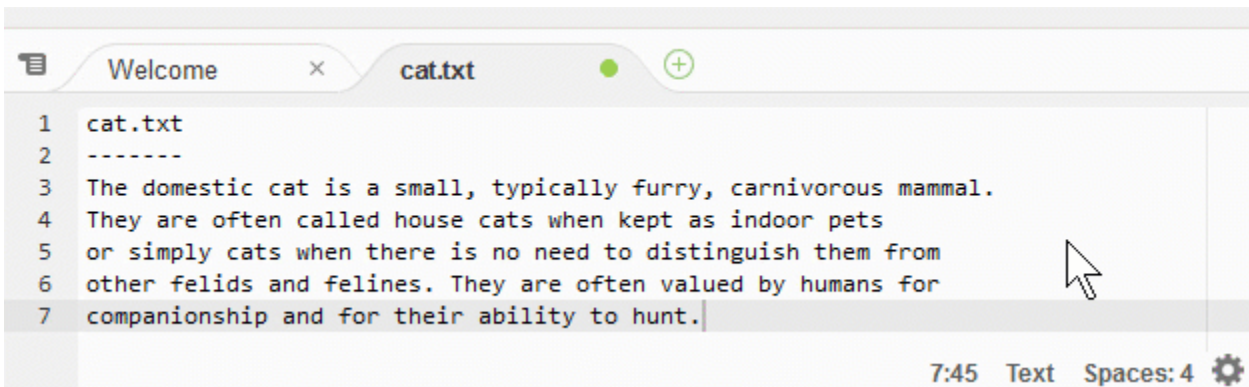
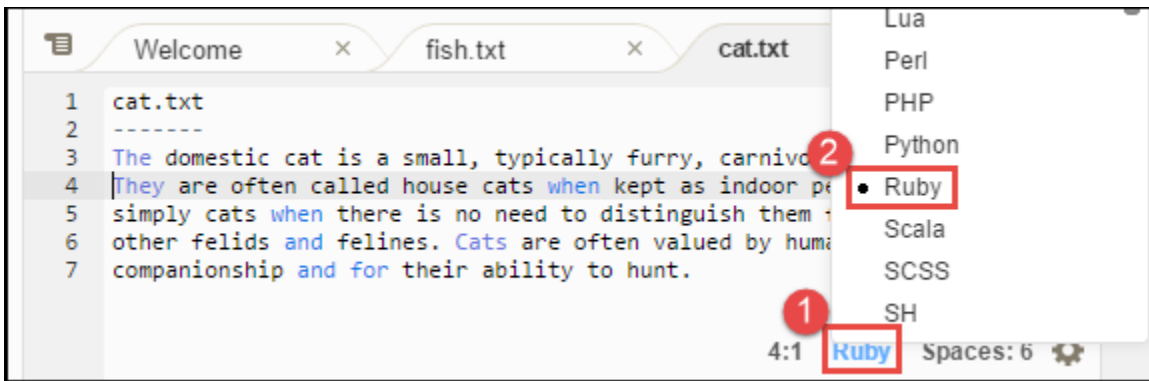


要显示或隐藏状态栏，请在菜单栏上选择 View (视图) > Status Bar (状态栏)。

要转到特定行号，请选择包含感兴趣文件的选项卡。然后在状态栏中选择行号和字号（应类似于 7:45）。键入行号（如 4），然后按 Enter，如下所示。



要更改文件类型首选项，请在状态栏中选择不同的文件类型。例如，为 `cat.txt` 选择 Ruby 可查看语法色彩的更改。要恢复纯文本颜色，请选择 Plain Text (纯文本)，如下所示。



步骤 9：“Outline (大纲)”窗口

您可以使用 Outline (大纲) 窗口快速转到特定的文件位置。

要显示或隐藏 Outline (大纲) 窗口和 Outline (大纲) 按钮，请在菜单栏上依次选择 Window (窗口)、Outline (大纲)。

要了解 Outline (大纲) 窗口的工作方式，请创建一个名为 `hello.rb` 的文件。将以下代码复制到文件中并保存它。

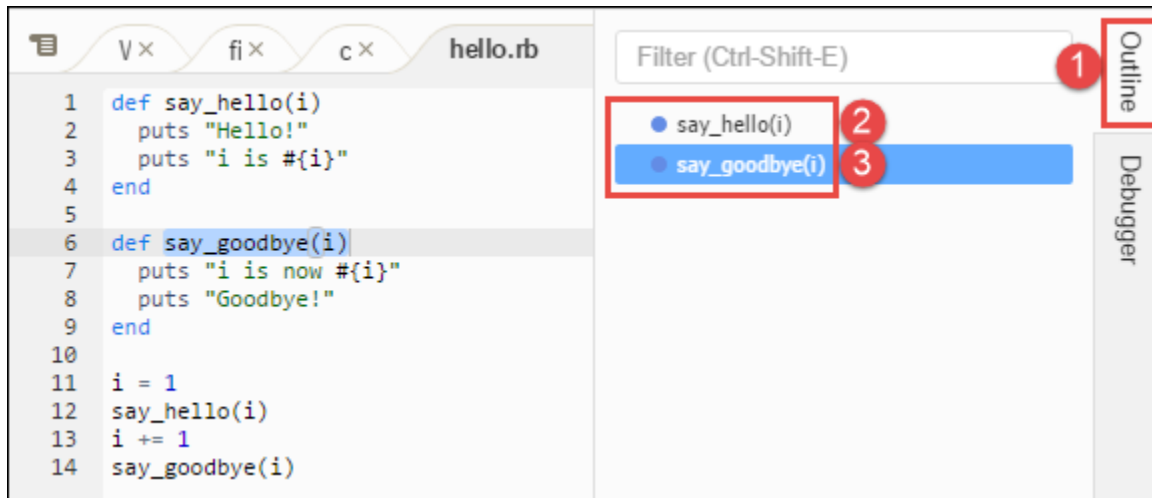
```
def say_hello(i)
  puts "Hello!"
  puts "i is #{i}"
end

def say_goodbye(i)
  puts "i is now #{i}"
  puts "Goodbye!"
end
```

```
i = 1
say_hello(i)
i += 1
say_goodbye(i)
```

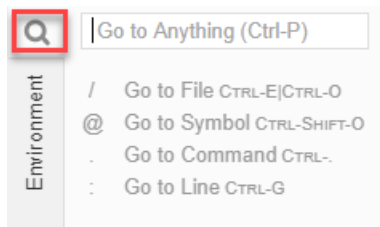
要显示或隐藏 Outline (大纲) 窗口的内容，请选择 Outline (大纲) 按钮。

在 Outline (大纲) 窗口中，选择 say_hello(i)，然后选择 say_goodbye(i)。



步骤 10：“Go (转到)”窗口

您可以使用 Go (转到) 窗口以在编辑器中打开文件，转到符号的定义，运行命令，或转到编辑器中活动文件的某一行。



要显示 Go (转到) 窗口的内容，请选择 Go (转到) 按钮 (放大镜图标)。

要显示或隐藏 Go (转到) 窗口和 Go (转到) 按钮，请在菜单栏上选择 Window (窗口)、Go (转到)。

在 Go (转到) 窗口打开的情况下，您可以：

- 键入一个正斜杠 (/)，后跟部分或全部文件名。在显示的匹配文件列表中，选择要在编辑器中打开的文件。例如，键入 /fish 会列出 fish.txt，而键入 /.txt 会列出 fish.txt 和 cat.txt。

Note

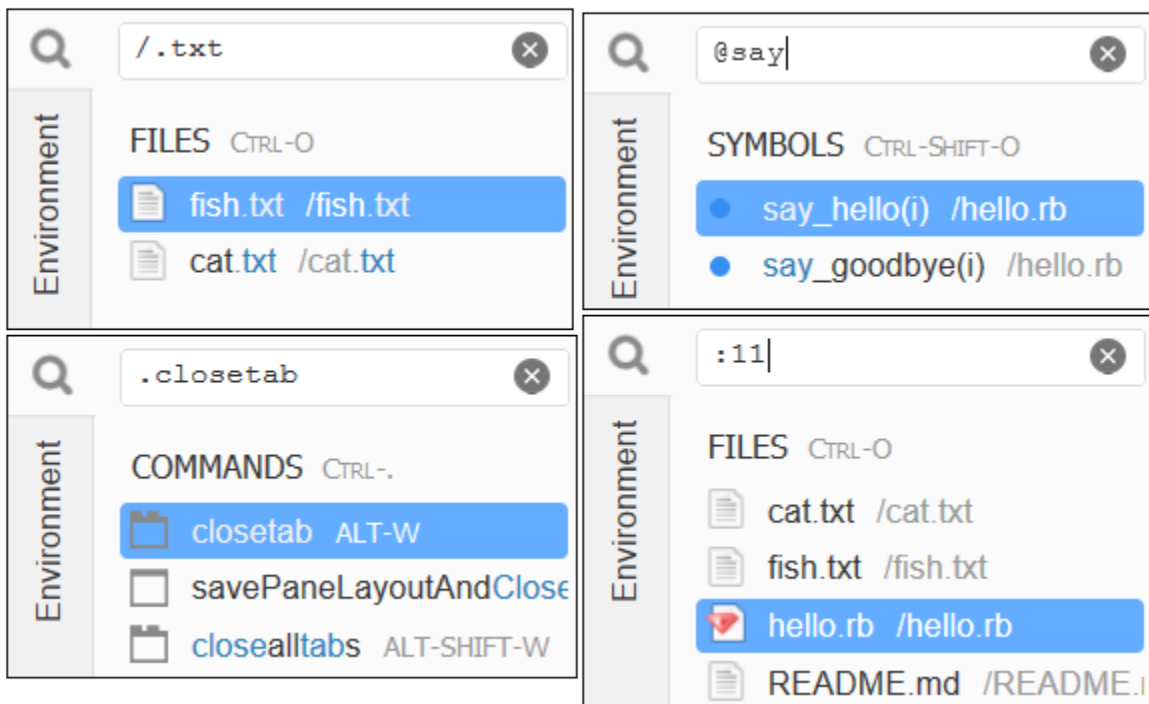
文件搜索范围仅限 Environment (环境) 窗口中的非隐藏文件和非隐藏文件夹。

- 键入一个 @ 符号 (@)，后跟符号名称。在显示的匹配符号列表中，选择要在编辑器中转到的符号。例如，在编辑器中打开 hello.rb 文件并处于活动状态时，键入 @hello 可列出 say_hello(i)，或键入 @say 可列出 say_hello(i) 和 say_goodbye(i)。

Note

如果编辑器中的活动文件是支持的语言项目的一部分，则符号搜索范围限于当前项目。否则，符号搜索范围仅限编辑器中的活动文件。有关更多信息，请参阅[增强 TypeScript 支持和特征](#)。

- 键入一个点号 (.)，后跟命令名称。在显示的命令列表中，选择一条命令以运行该命令。例如，键入 .closetab 然后按 Enter 键关闭编辑器中的当前选项卡。有关可用命令的列表，请参阅[用于 AWS Cloud9 集成开发环境 \(IDE\) 的命令参考](#)。
- 键入一个冒号 (:)，后跟一个数字，以转到编辑器中活动文件的该行号。例如，在编辑器中打开 hello.rb 文件并处于活动状态时，键入 :11 以转到该文件的第 11 行。



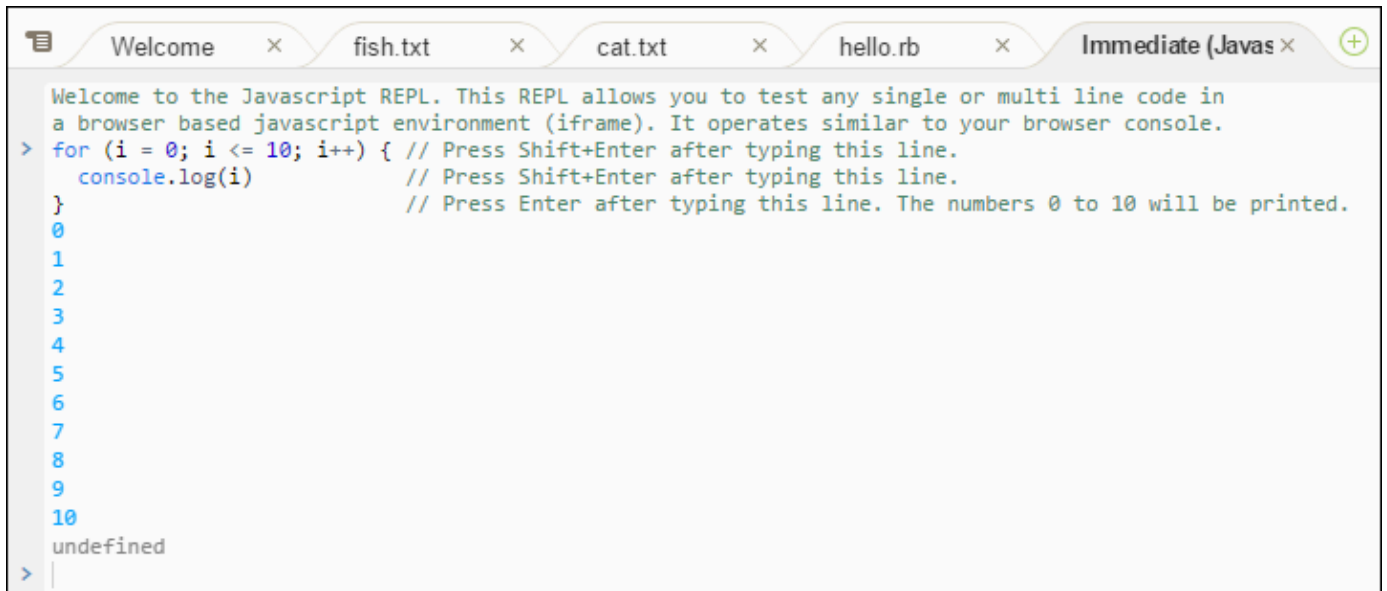
要根据当前键盘模式和操作系统查看每个操作的键盘绑定，请参阅菜单栏中 Go (转到) 菜单上的每个可用 Go To (转到) 命令。

步骤 11：“Immediate (立即运行)”选项卡

使用 Immediate (立即运行) 选项卡可以测试小段的 JavaScript 代码。要了解 Immediate (立即运行) 选项卡的工作方式，请执行以下操作。

1. 在菜单栏上选择 Window (窗口)、New Immediate Window (新建即时窗口)，打开 Immediate (立即运行) 选项卡。
2. 在 Immediate (立即运行) 选项卡中运行某些代码。要尝试该操作，请在窗口中键入以下代码，并在键入第一行和第二行以后分别按 Shift-Enter。在键入第三行以后按 Enter。(如果您在键入第一行或第二行以后按 Enter 而不是 Shift-Enter，代码将早于您的预期时间运行。)

```
for (i = 0; i <= 10; i++) { // Press Shift-Enter after typing this line.
  console.log(i)           // Press Shift-Enter after typing this line.
}                           // Press Enter after typing this line. The numbers 0 to
 10 will be printed.
```

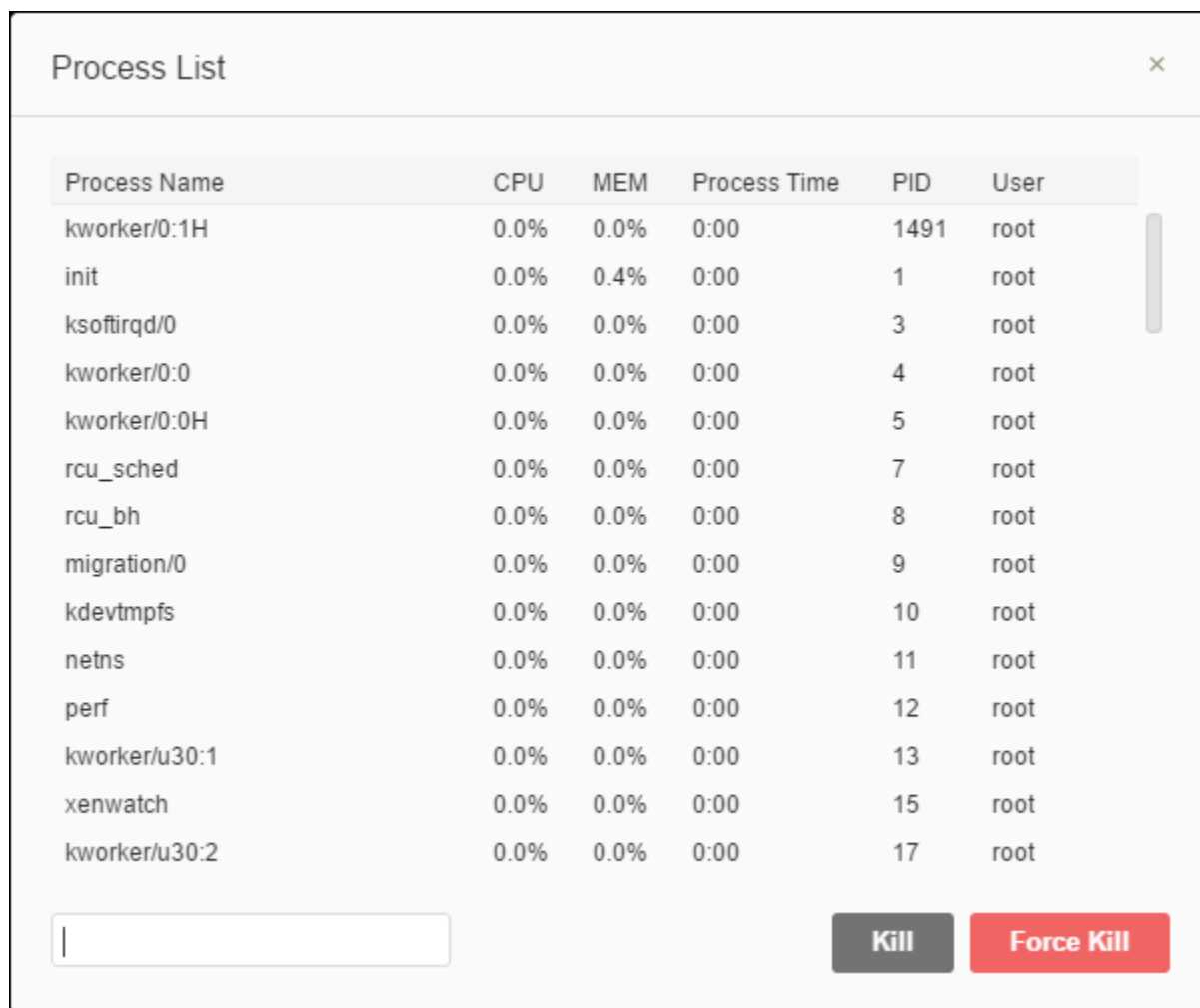


```
Welcome to the Javascript REPL. This REPL allows you to test any single or multi line code in a browser based javascript environment (iframe). It operates similar to your browser console.
> for (i = 0; i <= 10; i++) { // Press Shift+Enter after typing this line.
  console.log(i)           // Press Shift+Enter after typing this line.
}                           // Press Enter after typing this line. The numbers 0 to 10 will be printed.
0
1
2
3
4
5
6
7
8
9
10
undefined
>
```

步骤 12：进程列表

Process List (进程列表) 中显示所有正在运行的进程。您可以停止、甚至强制停止您不想继续运行的进程。要了解 Process List (进程列表) 窗口的工作方式，请执行以下操作。

1. 在菜单栏上选择 Tools (工具)、Process List (进程列表)，以显示 Process List (进程列表)。
2. 查找进程。在 Process List (进程列表) 中，键入进程的名称。
3. 停止或强制停止进程。在进程列表中选择进程，然后选择 Kill (终止) 或 Force Kill (强制终止)。

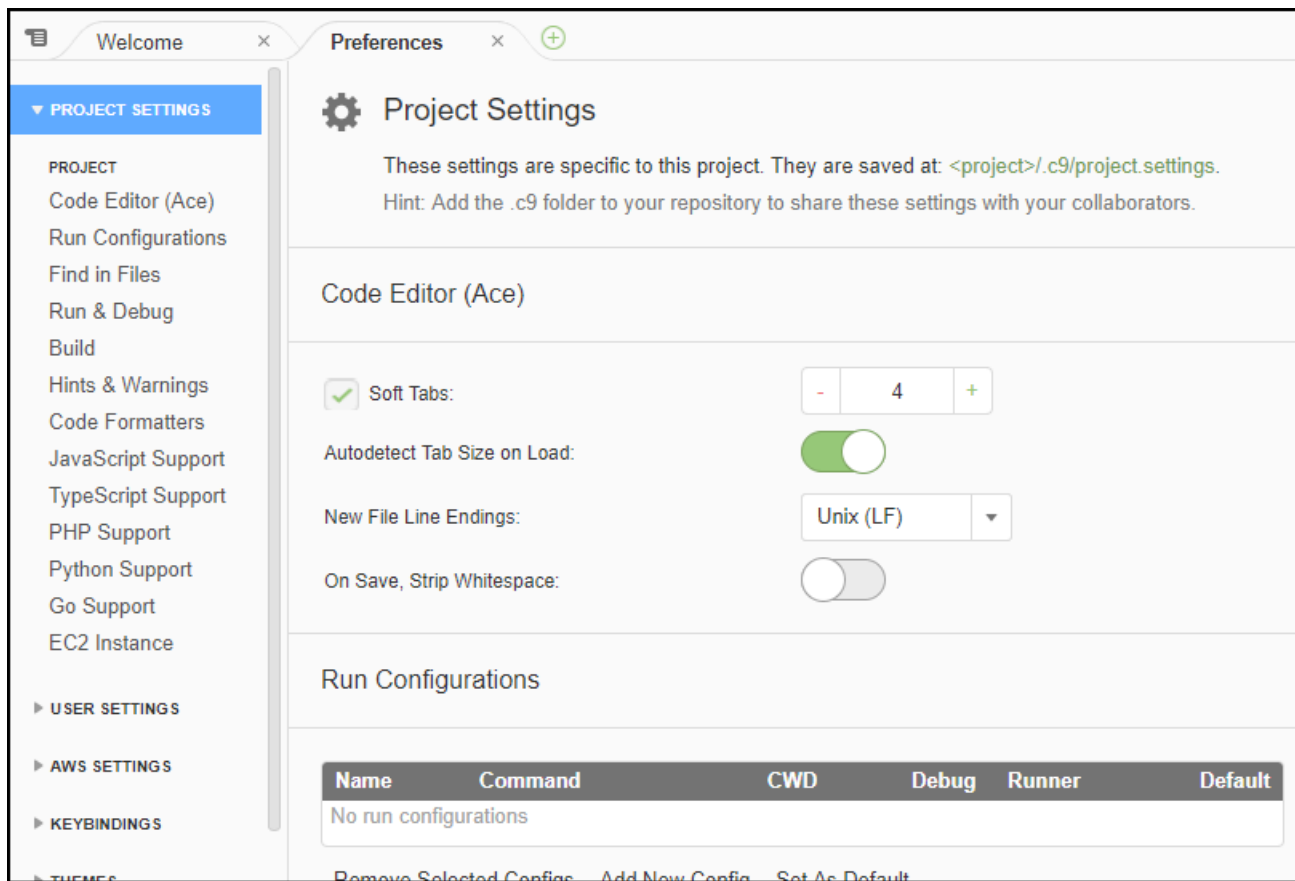


第 13 步：首选项

首选项 中包含以下设置。

- 仅适用于当前环境的设置，例如是否在编辑器中使用软制表符、要忽略的文件类型以及 PHP 和 Python 等语言的代码完成行为。
- 适用于每个环境的用户设置，如颜色、字体和编辑器行为。
- 您的键绑定，例如，您首选用于处理文件和编辑器的快捷键组合。
- IDE 的整体主题。

要显示首选项，请在菜单栏上选择 **AWS Cloud9 > Preferences (首选项)**。此时会显示类似于下面的内容：



第 14 步：终端

您可以在 IDE 中运行一个或多个终端会话。要开始一个终端会话，请在菜单栏上依次选择 Window (窗口)、New Terminal (新建终端)。或者，选择“Console (控制台)”选项卡旁边的“加号”图标，然后选择 New Terminal (新建终端)。

您可以尝试在终端中运行命令。例如，在终端中键入 `echo $PATH`，然后按 Enter 以显示 PATH 环境变量的值。

您也可以尝试运行其他命令。例如，尝试使用以下命令。

- **pwd** 可输出当前目录的路径。
- **aws --version** 可输出有关 AWS CLI 的版本信息。
- **ls -l** 可输出有关当前目录的信息。



```
hello.rb
1 def say_hello(i)
2   puts "Hello!"
3   puts "i is #{i}"
4 end
5
6 def say_goodbye(i)
7   puts "i is now #{i}"
8   puts "Goodbye!"
9 end
10
```

```
bash - "ip-172-31"
Cloud9Admin:~/environment $
```

步骤 15：“Debugger (调试程序)”窗口

您可以使用 Debugger (调试程序) 窗口调试代码。例如，您可以单步运行代码（即一次运行一部分）、观察一段时间内的变量值以及研究调用堆栈。

Note

此过程类似于任一[基本 IDE 教程](#)中的[步骤 2：IDE 基本导览](#)。

要显示或隐藏 Debugger (调试程序) 窗口和 Debugger (调试程序) 按钮，请在菜单栏上依次选择 Window (窗口)、Debugger (调试程序)。

在本教程中，您可以通过执行以下操作来试用 Debugger (调试程序) 窗口和一些 JavaScript 代码。

1. 在终端会话中运行以下命令：**node --version**，来检查环境中 Node.js 的安装情况。如果已安装 Node.js，则输出中会显示 Node.js 版本号，您可以跳至本过程的第 3 步（“编写一段 JavaScript 代码...”）。

2. 如果您需要安装 Node.js，请执行以下操作。

- a. 运行以下两条命令，一次一条，以确保环境拥有最新的更新，然后下载 Node Version Manager (nvm)。(nvm 是一个简单的 Bash shell 脚本，对安装和管理 Node.js 版本非常有用。有关更多信息，请参阅 GitHub 上的 [Node Version Manager](#)。)

对于 Amazon Linux：

```
sudo yum -y update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh |
  bash
```

对于 Ubuntu Server：

```
sudo apt update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh |
  bash
```

- b. 使用文本编辑器更新您的 Shell 配置文件（例如，`~/.bashrc`）以允许加载 nvm。例如，在 IDE 的 Environment（环境）窗口中选择齿轮图标，然后选择 Show Home in Favorites（在收藏夹中显示主页）。重复该步骤，然后选择 Show Hidden Files（显示隐藏的文件）。
- c. 打开 `~/.bashrc` 文件。
- d. 在文件末尾键入或粘贴以下代码，以启用 nvm 加载。

对于 Amazon Linux：

```
export NVM_DIR="/home/ec2-user/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm.
```

对于 Ubuntu Server：

```
export NVM_DIR="/home/ubuntu/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm.
```

- e. 保存该文件。
- f. 关闭该终端会话，并启动一个新会话。然后，运行以下命令来安装最新版本的 Node.js。

```
nvm install node
```

- 编写一段 JavaScript 代码进行调试。例如，创建一个文件，将以下代码添加到该文件中，并将其保存为 `hello.js`。

```
var i;

i = 10;

console.log("Hello!");
console.log("i is " + i);

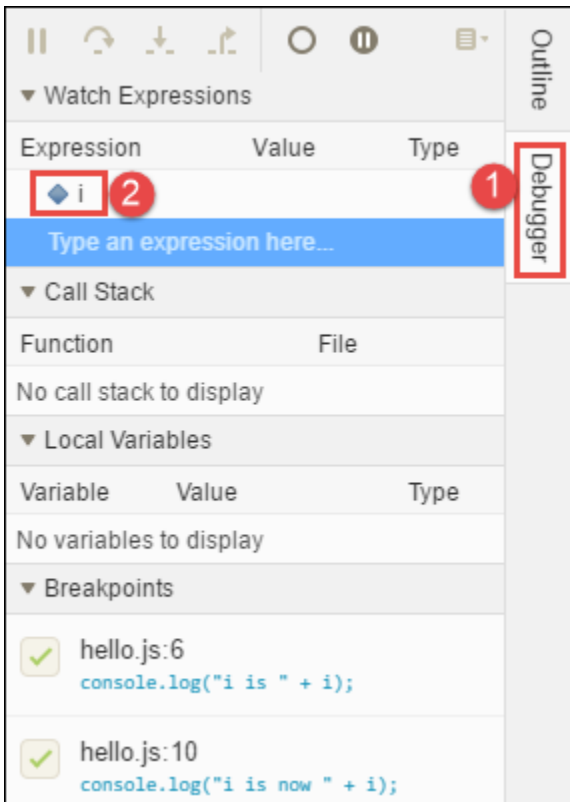
i += 1;

console.log("i is now " + i);
console.log("Goodbye!");
```

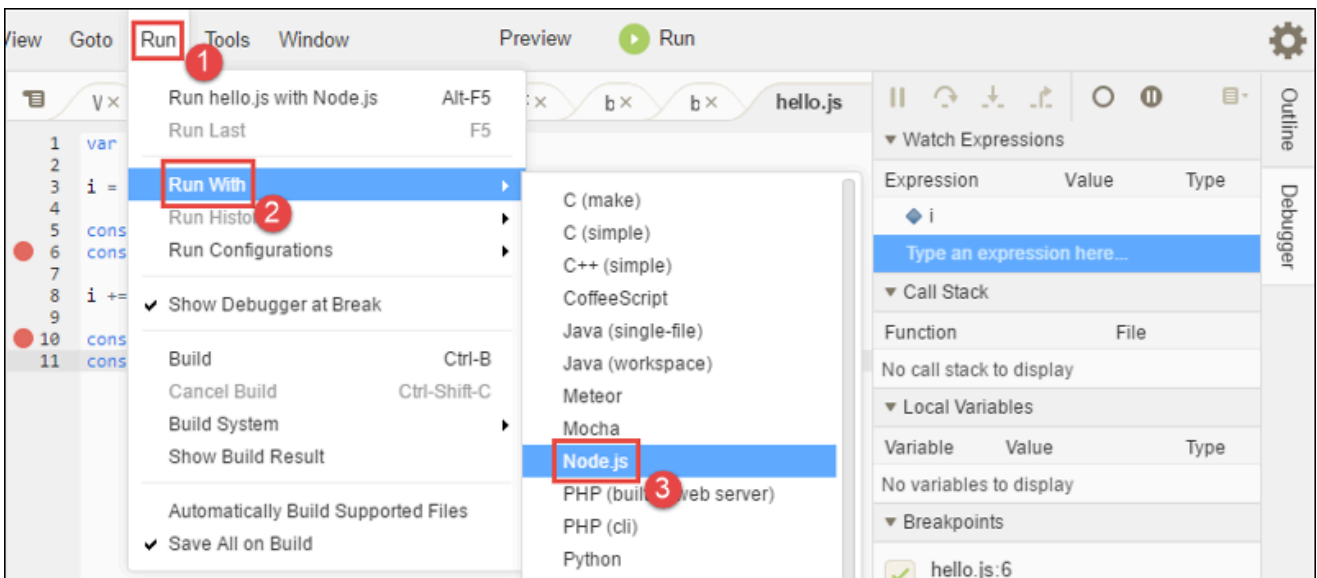
- 将一些断点添加到代码中。例如，在间距中选择第 6 行和第 10 行的页边距。以上两行的行号旁会显示一个红色圆圈，如下所示。



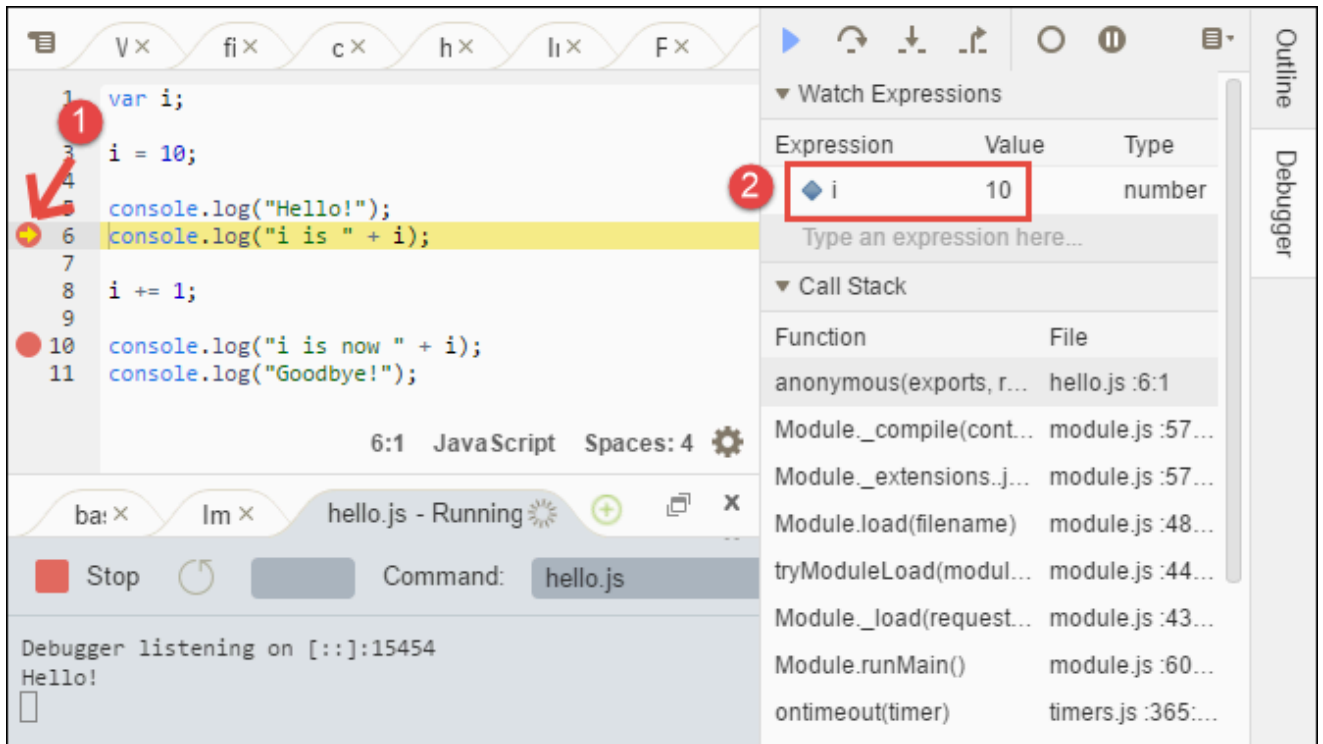
- 现在，已经做好调试 JavaScript 代码的准备了。为此，请执行以下操作。
 - 要显示或隐藏 Debugger (调试程序) 窗口的内容，请选择 Debugger (调试程序) 按钮，如下一步中所示。
 - 观察名为 `i` 的变量在代码运行过程中的值。在 Debugger (调试程序) 窗口中，为 Watch Expressions (监视表达式) 选择 Type an expression here (在此处键入表达式)。键入字母 `i`，然后按 Enter，如下所示。



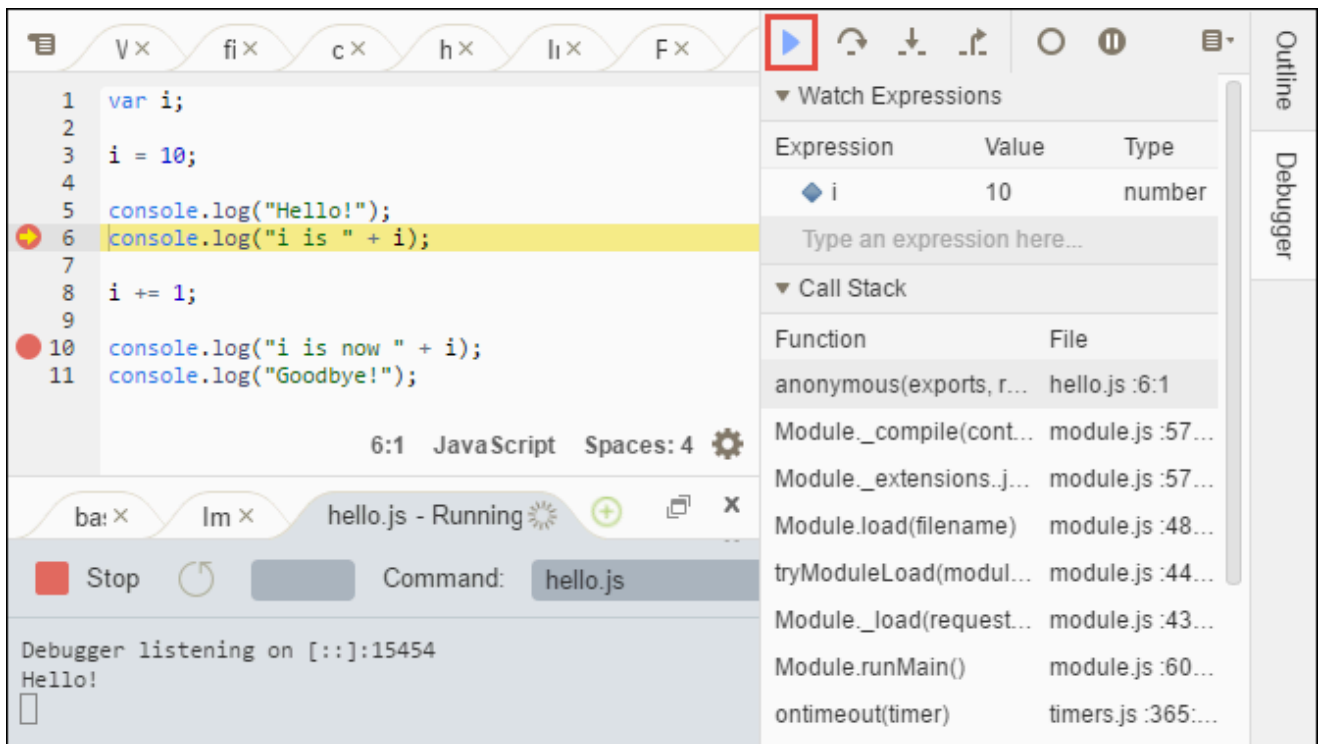
- c. 开始运行代码。依次选择 Run (运行)、Run With (运行方式)、Node.js，如下所示。



- d. 代码在第 6 行暂停运行。Debugger (调试程序) 窗口显示 Watch Expressions (监视表达式) 中 `i` 的值，当前值是 10。



- e. 在 Debugger (调试程序) 窗口中，选择 Resume (恢复)，即如下所示的蓝色箭头图标。



- f. 代码在第 10 行暂停运行。Debugger (调试程序) 窗口现在显示 i 的新值，当前值是 11。
- g. 再次选择 Resume (恢复)。代码会运行到结束。输出会转到控制台的 hello.js 选项卡，如下所示。

The screenshot displays the AWS Cloud9 IDE interface. The code editor on the left shows the following JavaScript code:

```
1 var i;  
2  
3 i = 10;  
4  
5 console.log("Hello!");  
6 console.log("i is " + i);  
7  
8 i += 1;  
9  
10 console.log("i is now " + i);  
11 console.log("Goodbye!");
```

The debugger on the right is open, showing the following sections:

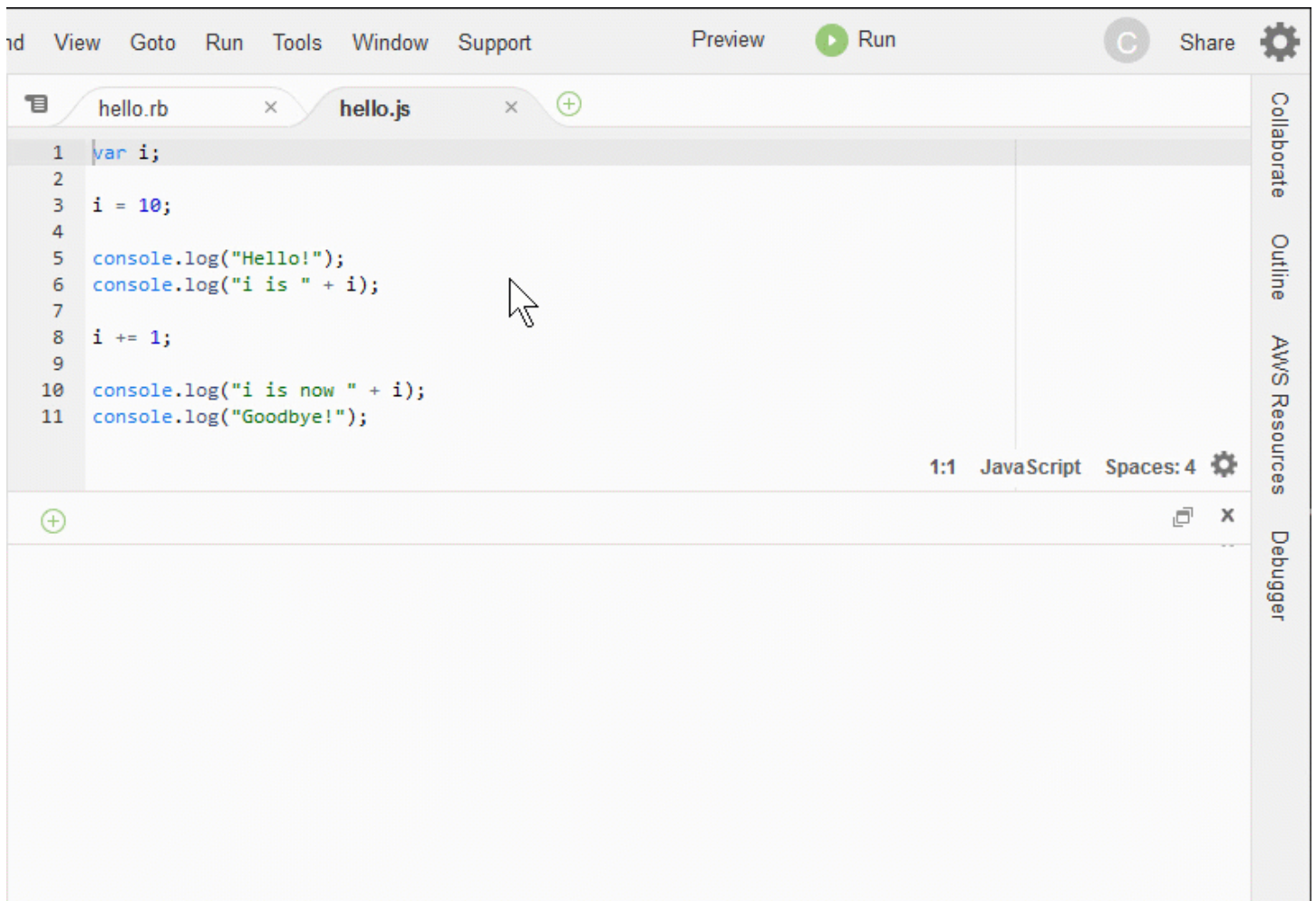
- Watch Expressions:** A table with columns "Expression", "Value", and "Type". It shows a single entry for the variable `i` with a value of `11` and a type of `number`.
- Call Stack:** Shows "No call stack to display".
- Local Variables:** Shows "No variables to display".
- Breakpoints:** Two breakpoints are listed: one at `hello.js:6` for `console.log("i is " + i);` and another at `hello.js:10` for `console.log("i is now " + i);`.

The console at the bottom shows the output of the script:

```
Debugger listening on [::]:15454  
Hello!  
i is 10  
i is now 11  
Goodbye!
```

The console output is highlighted with a red box. Below the console, it says "Process exited with code: 0".

将结果与以下内容进行比较。



最终思考

⚠ Warning

请记住，拥有 AWS Cloud9 开发环境可能导致向您的 AWS 账户收取费用。如果您使用 EC2 环境，其中包括可能对 Amazon EC2 收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

上一级章节 ([使用 IDE](#)) 中还有其他主题，可供您用于进一步探索。但是，当您完成 AWS Cloud9 IDE 导览，并且不再需要环境时，请务必将其及相关资源删除，如 [删除环境](#) 中所述。

AWS Cloud9 集成开发环境 (IDE) 中的语言支持

AWS Cloud9 IDE 支持多种编程语言。下表列出了支持的语言以及支持级别。

Language	语法突出显示 ¹	运行 UI ²	大纲视图	代码提示和检查	代码完成	调试 ³
C++	✓	✓	✓		✓ ⁵	✓ ⁴
C#	✓		✓		✓ ⁵	
CoffeeScript	✓	✓				
CSS	✓				✓	
Dart	✓					
Go	✓	✓	✓	✓	✓ ⁴	✓ ⁴
Haskell	✓					
HTML	✓	✓	✓		✓	
Java ⁶	✓	✓	✓	✓	✓	✓
JavaScript	✓	✓	✓	✓	✓	
Node.js	✓	✓	✓	✓	✓	✓
PHP	✓	✓	✓	✓	✓ ⁷	✓
Python	✓	✓	✓	✓	✓ ⁸	✓
Ruby	✓	✓	✓	✓	✓ ⁵	
Shell 脚本	✓	✓	✓	✓	✓ ⁵	
TypeScript ⁹	✓	✓	✓	✓	✓	

备注

¹ AWS Cloud9 IDE 为更多语言提供了语法高亮显示。有关完整列表，请在 IDE 菜单栏中选择 View > Syntax (视图 > 语法)。

- ² 您可以单击按钮，为标有 ✓ 的语言运行程序或脚本，而无需使用命令行。对于未标有 ✓ 或未显示在 IDE 中的 Run > Run With (运行 > 运行方式) 菜单栏上的语言，您可以为该语言创建一个运行程序。有关说明，请参阅[创建生成器或运行程序](#)。
- ³ 您可以使用 IDE 的内置工具调试标有 ✓ 的的语言的程序或脚本。有关说明，请参阅[调试您的代码](#)。
- ⁴ 对于该语言，该功能处于实验状态。未完全实现该功能，并且未介绍或不支持该功能。
- ⁵ 对于该语言，该功能仅支持本地函数。
- ⁶ 可以在具有 2 GiB 或更大内存的 AWS Cloud9 EC2 开发环境中激活对 Java SE 11 功能的增强支持。有关更多信息，请参阅[增强了对 Java 开发的支持](#)。
- ⁷ 要指定用于 AWS Cloud9 完成自定义 PHP 代码的路径，请在 AWS Cloud9 IDE 中打开“首选项”中的“项目”、“PHP 支持”、“启用 PHP 代码完成”设置，然后将自定义代码的路径添加到“项目”、“PHP 支持”、“PHP 完成包含路径”设置中。
- ⁸ 要指定用于 AWS Cloud9 完成自定义 Python 代码的路径，请在 AWS Cloud9 IDE 中打开“首选项”中的“项目”、“Python 支持”、“启用 Python 代码完成”设置，然后将自定义代码的路径添加到项目、Python 支持、PythonP ATH 设置中。
- ⁹ AWS Cloud9 IDE 在语言项目环境中为某些编程语言提供了额外支持，例如 TypeScript (AWS Cloud9 IDE 支持 3.7.5 版)。有关更多信息，请参阅[使用语言项目](#)。

AWS Cloud9 集成开发环境 (IDE) 中支持的编程语言版本

下表概述了 AWS Cloud9 IDE 中特定 AMI 支持哪些版本的编程语言。Ubuntu 18 于 2023 年下线，因此无法在 AWS Cloud9 中更新编程语言版本。

语言	Amazon Linux 2023	Amazon Linux 2	Ubuntu 18	Ubuntu 22
Python3	3.9	3.8	3.6	3.10
TypeScript	3.7.5	3.7.5	3.7.5	3.7.5
PHP	8.2	8.2	7.2	8.1
Ruby	3.2	3.0	3.0	3.2

语言	Amazon Linux 2023	Amazon Linux 2	Ubuntu 18	Ubuntu 22
Java	11、17	11	11	11、17
Python2	不适用	2.7	不适用	不适用
C++*	23	17	17	23
Go	1.20	1.20	1.9	1.21
CoffeeScript	2.7	2.7	2.7	2.7

*您可以运行以下命令，通过您要使用的编程语言版本编译 C++ 文件：

```
g++ -std=c++[version-number] "$file" -o "$file.o"
```

AWS Cloud9 集成开发环境 (IDE) 中的增强型语言支持

AWS Cloud9 在使用以下语言编写代码时，提供增强的支持以改善您的开发体验：

- Java：扩展允许提供诸如代码完成、错误检查、特定于上下文的操作和调试选项等功能。
- TypeScript：语言项目 提供对 TypeScript 的增强生产力功能的访问权限。

主题

- [增强了对 Java 开发的支持](#)
- [增强 TypeScript 支持和特征](#)

增强了对 Java 开发的支持

AWS Cloud9 提供额外的语言支持，以改善使用 Java 时的开发体验。主要的生产力特性包括代码完成、错误筛选、代码镜头，以及调试选项，如断点和步进。

Important

增强的生产力功能仅适用于连接到 Amazon EC2 实例的 AWS Cloud9 开发环境。

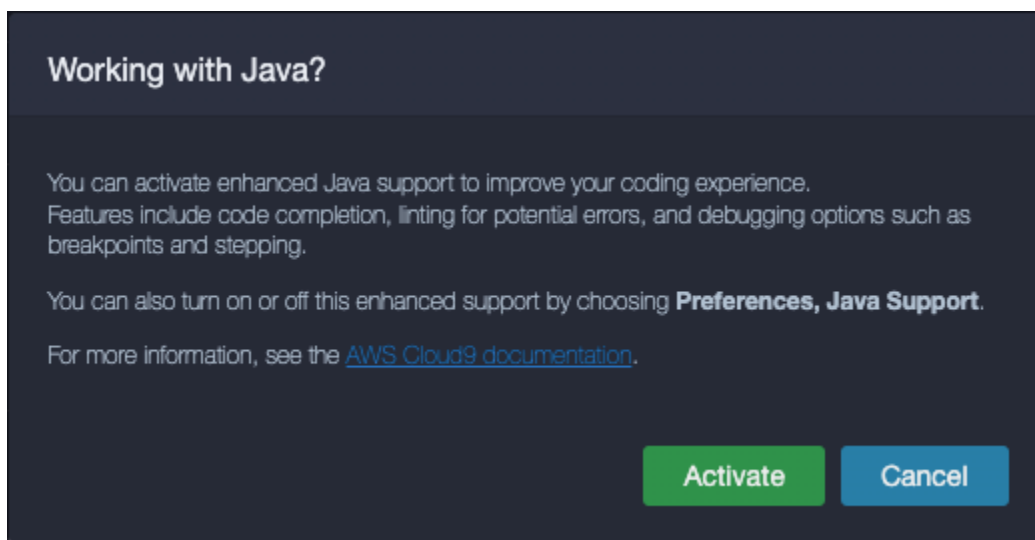
此外，为了确保在使用针对 Java 的增强语言支持时获得最佳的 IDE 体验，Amazon EC2 计算实例支持 AWS Cloud9 环境要求 2 GiB 或者更多的内存。如果 AWS Cloud9 检测到您的 EC2 计算实例没有足够的 RAM，那么您将无法激活 Java 的增强功能。

激活和自定义增强的 Java 支持

如果满足以下条件，将自动显示激活 Java 增强支持的选项：

- 您的 AWS Cloud9 环境连接到具有 2 GiB 或更多内存的 Amazon EC2 实例。
- 您正在处理与 Java 开发相关的文件。AWS Cloud9 检查以下文件名和扩展名：`*.java`、`*.gradle`（与 Gradle 构建工具关联），以及 `pom.xml`（与 Apache Maven 构建工具关联）。
- 您正工作所处的 AWS Cloud9 环境是在 2020 年 11 月 11 日后创建的。目前，不可能在此日期之前创建的开发环境中使用 Java 生产力功能。

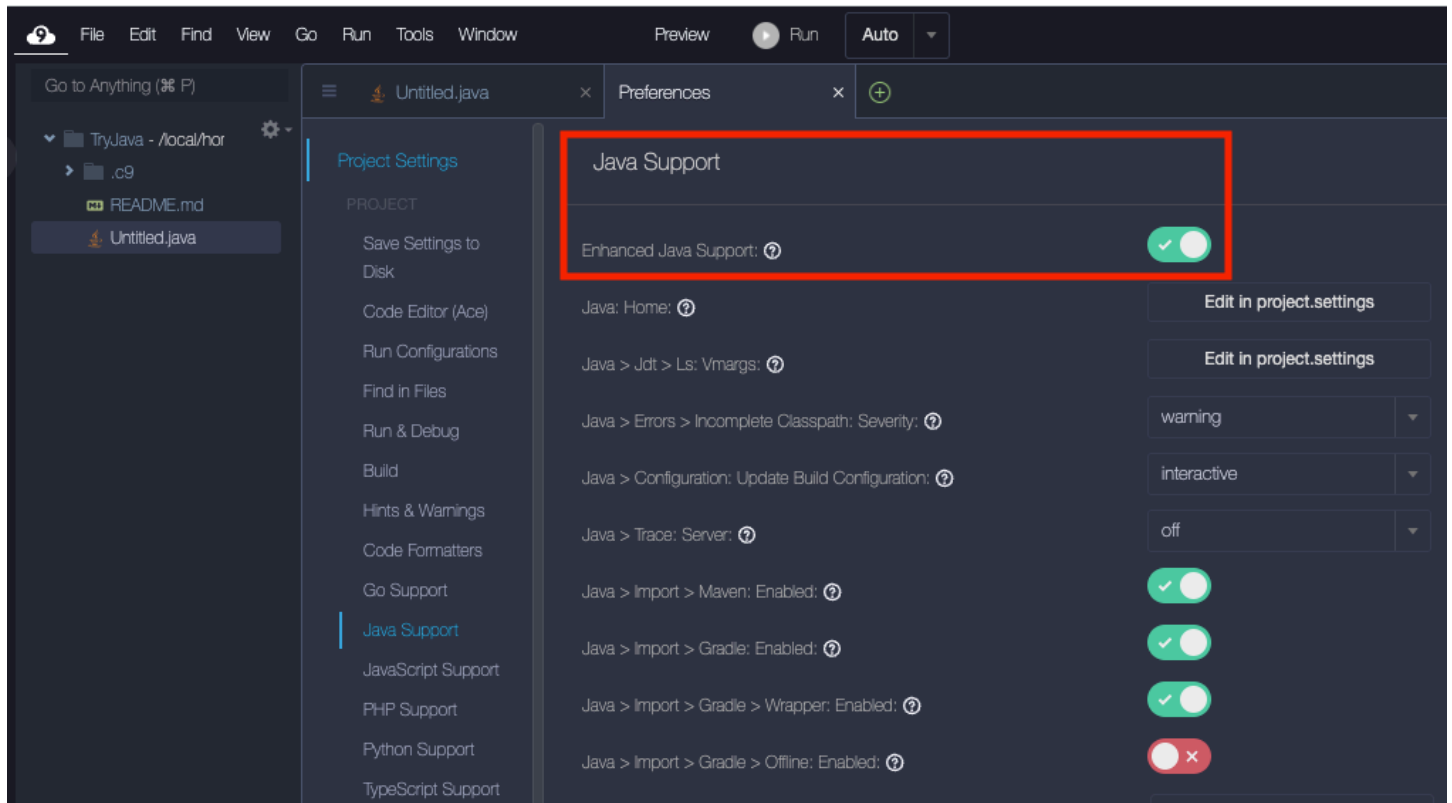
如果满足这些条件，将显示一个对话框，询问您是否要激活用于编码和调试 Java 的额外生产力功能。如果选择 **Activate**（激活），则可以开始使用 IDE 中的功能。



i Note

在创建 AWS Cloud9 环境时启动的 Amazon EC2 实例已安装 Amazon Corretto 11。Amazon Corretto 是开放 Java 开发工具包 (OpenJDK) 的免费、多平台、生产就绪型分发版。这意味着您可以在 AWS Cloud9 开箱即用的环境中开始开发和运行 Java 应用程序。

您还可以使用 AWS Cloud9 接口手动激活和停用增强的语言和调试支持。依次选择 Preferences (首选项)、Java Support (Java 支持) 和 Enhanced Java Support (增强版 Java 支持)。



IDE 的两个扩展提供了 AWS Cloud9 中对 Java 开发的增强支持：

- Red Hat 对 Java(TM) 的语言支持
- 适用于 Java 的调试程序

AWS Cloud9 界面允许您访问各种设置，这些设置可以自定义这些扩展的性能。要更改扩展设置，请依次选择 Preferences (首选项)、Java Support (Java 支持)。

有关这些设置的详细信息，请参阅扩展程序的 GitHub 存储库中的已安装版本自述文件页面：

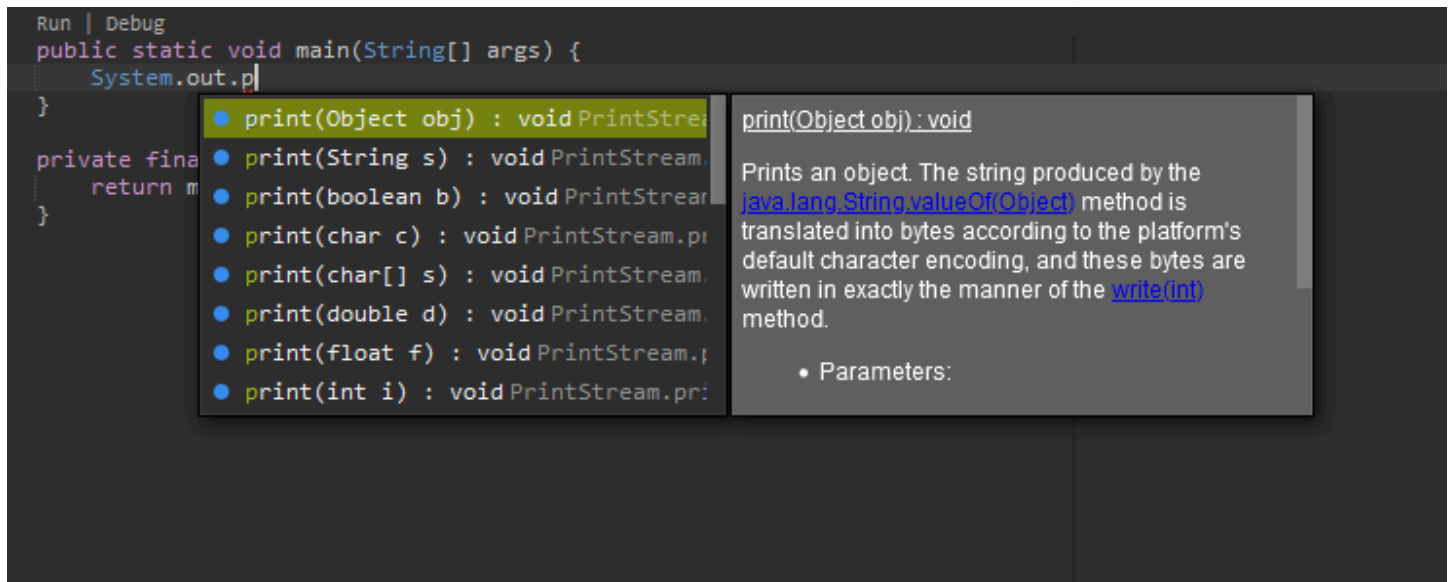
- [Red Hat 对 Java\(TM\) 的语言支持](#)
- [适用于 Java 的调试程序](#)

功能亮点

激活增强型 Java 支持后，您可以使用一系列提高生产力的功能。

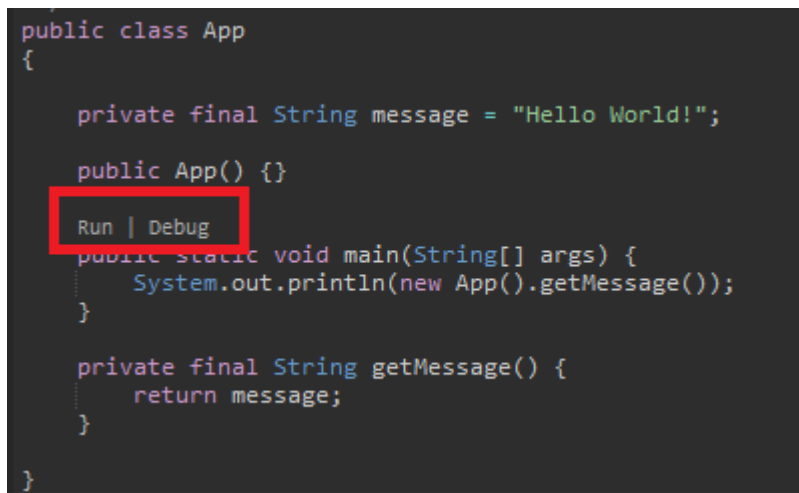
代码完成

代码完成后，编辑器会根据您输入的代码提供上下文感知建议。例如，如果在对象名称后键入点 (".") 运算符，则编辑器将显示该对象可用的方法或属性。



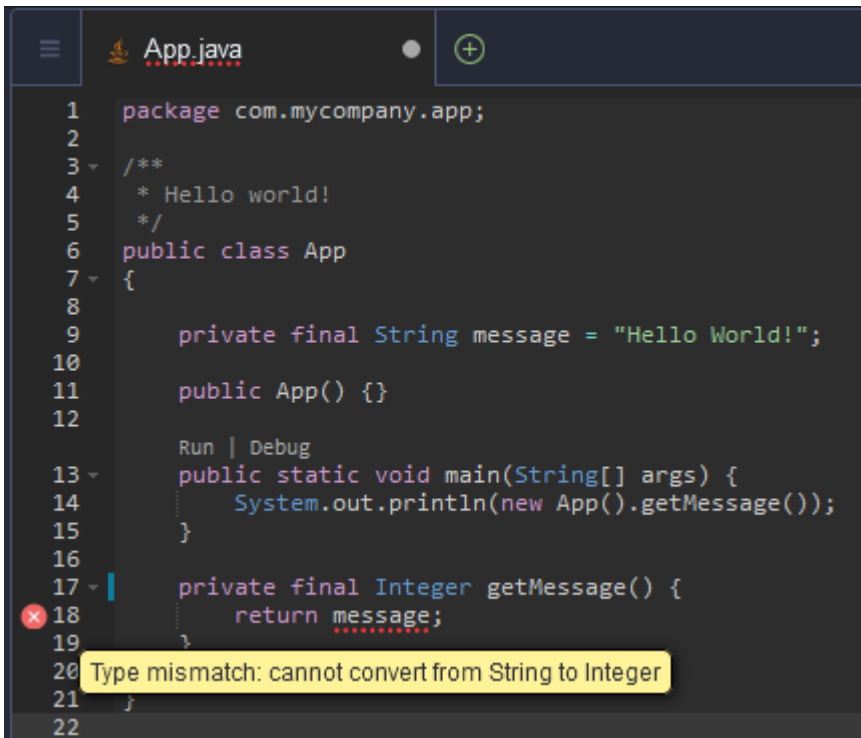
代码镜头

代码镜头允许您直接在源代码中访问上下文特定的操作。对于 Java 开发，代码透镜通过允许您运行和调试特定方法来促进单元测试。



代码检查

代码检查描述了编辑器在构建代码之前如何突出显示代码中的潜在错误。例如，如果您尝试使用未初始化的变量或试图为期望使用不同类型的变量分配值，检查工具就会调用。



```

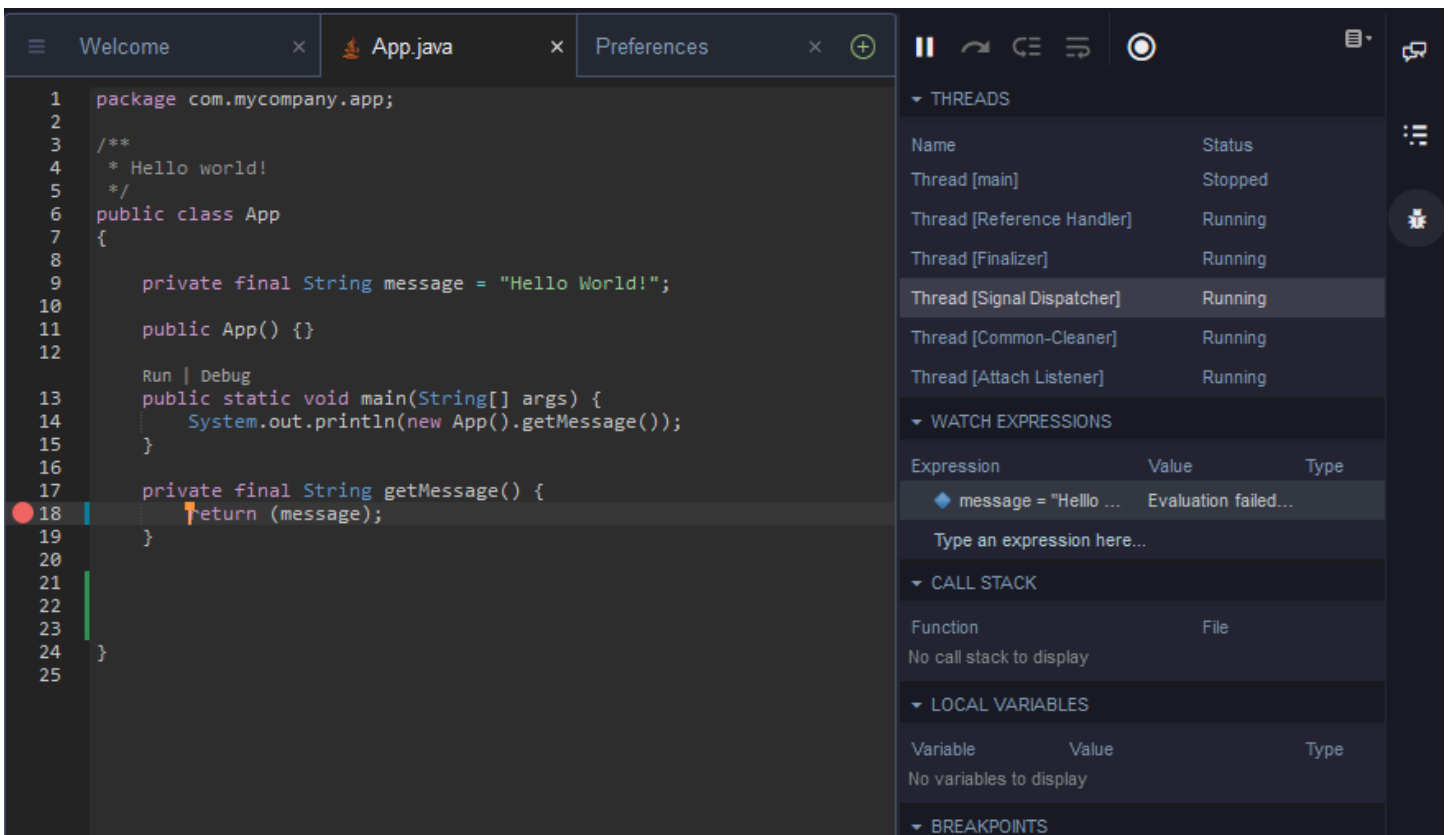
1 package com.mycompany.app;
2
3 /**
4  * Hello world!
5  */
6 public class App
7 {
8
9     private final String message = "Hello World!";
10
11     public App() {}
12
13     public static void main(String[] args) {
14         System.out.println(new App().getMessage());
15     }
16
17     private final Integer getMessage() {
18         return message;
19     }
20 }
21
22

```

Type mismatch: cannot convert from String to Integer

调试选项

可以实现断点和监视表达式。在源代码中设置断点并显示调试器窗格以定义相关条件。



```

1 package com.mycompany.app;
2
3 /**
4  * Hello world!
5  */
6 public class App
7 {
8
9     private final String message = "Hello World!";
10
11     public App() {}
12
13     public static void main(String[] args) {
14         System.out.println(new App().getMessage());
15     }
16
17     private final String getMessage() {
18         return (message);
19     }
20
21
22
23
24 }
25

```

Threads:

Name	Status
Thread [main]	Stopped
Thread [Reference Handler]	Running
Thread [Finalizer]	Running
Thread [Signal Dispatcher]	Running
Thread [Common-Cleaner]	Running
Thread [Attach Listener]	Running

Watch Expressions:

Expression	Value	Type
message = "Hello ...	Evaluation failed...	

Call Stack:

Function	File
No call stack to display	

Local Variables:

Variable	Value	Type
No variables to display		

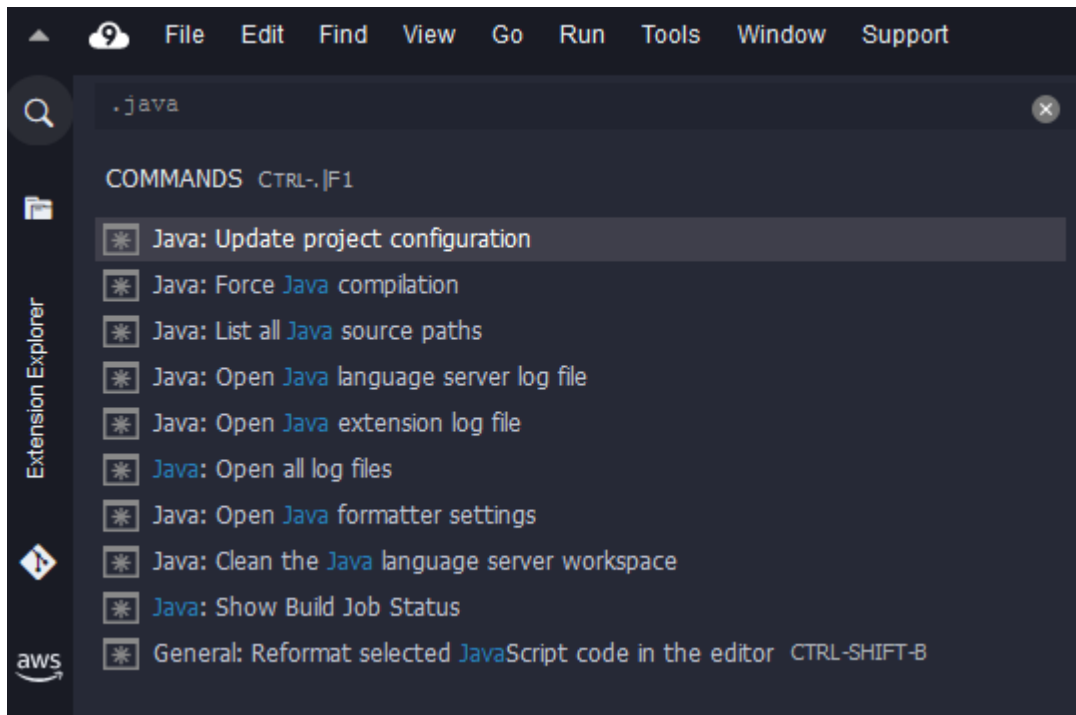
Breakpoints:

使用配置文件进行调试

您还可以使用 AWS Cloud9 通过 `launch.json` 和 `tasks.json` 配置文件支持的启动配置和任务来控制调试配置。有关启动配置及其使用方法的示例，请参阅 [Java 调试配置](#)。

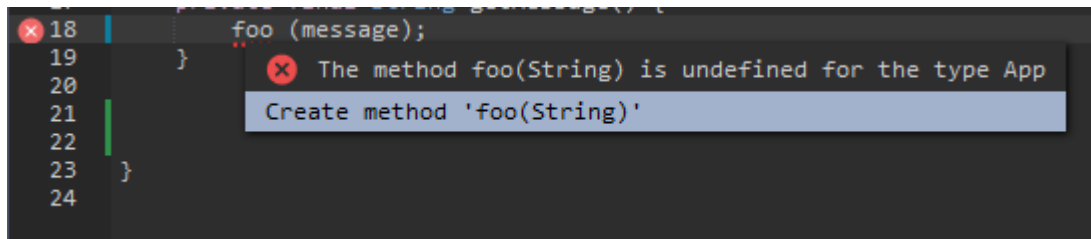
Java 命令

可以通过按下 `Ctrl+.` 或 `F1` 从 AWS Cloud9 命令面板中运行命令。然后通过输入“java”来过滤相关命令。



快速修复

通过快速修复，您可以通过为缺少的元素创建存根来解决由于使用未声明的变量或未定义的方法而导致的错误。



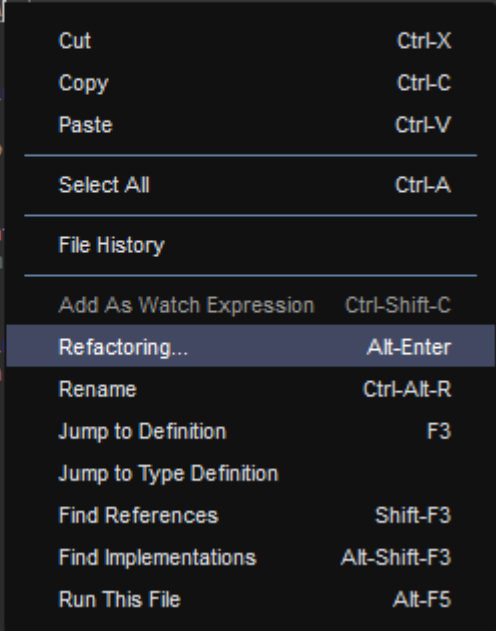
重构

重构允许您在不改变其行为的情况下重组代码。要访问诸如组织导入或创建构造函数等选项，请打开该项的上下文（右键单击）菜单，然后选择 Refactoring（重构）。

```

1  package com.mycompany.app;
2
3  /**
4   * Hello world!
5   */
6  public class App {
7  {
8
9   private final App() {}
10
11  public App() {}
12
13  Run | Debug
14  public static void main(String[] args) {
15  System.out.println("Hello World!");
16  }
17  private final App() {}
18  return new App();
19  }
20
21
22
23
24 }
25

```



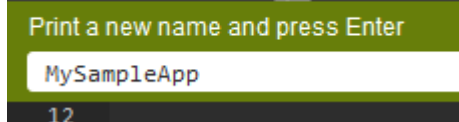
重命名

重命名是一项重构功能，允许您通过单个操作轻松修改代码中出现的所选变量、函数和类的名称。要更改名称，请打开该项目的上下文（右键单击）菜单，然后选择 Rename（重命名）。重命名会影响代码中名称的每个实例。

```

10
11  public App() {}

```



```

12

```

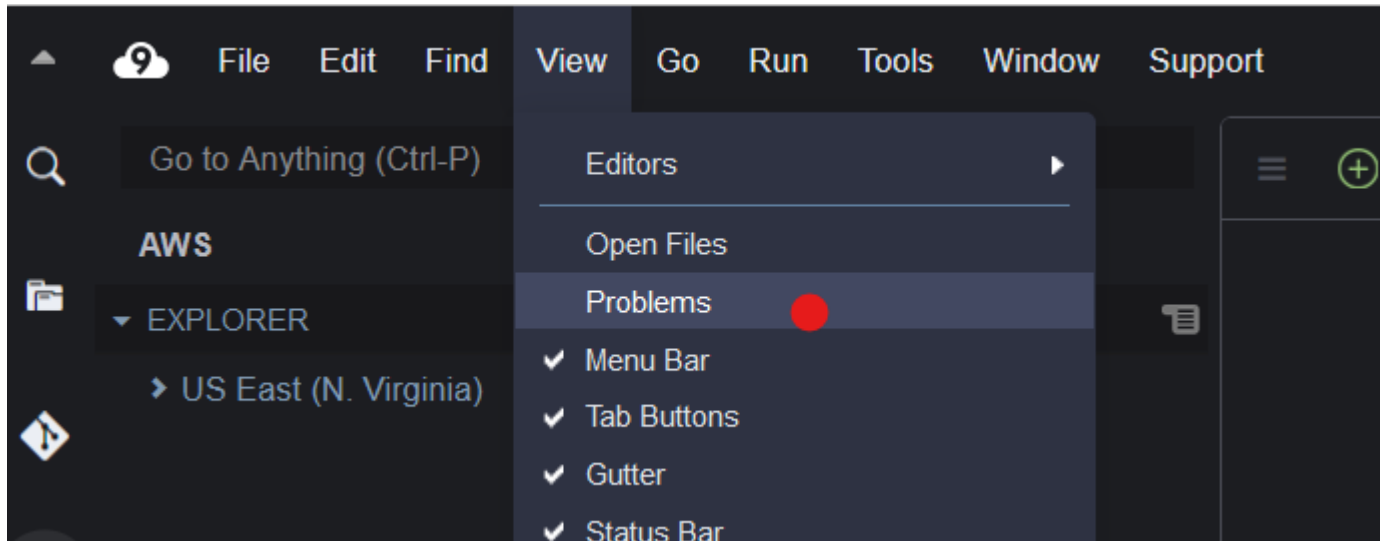
用于 Java 开发的可选工具

提供增强 Java 支持的扩展包括允许您将 Gradle 和 Maven 自动化工具集成到项目开发中的功能。这些工具没有预先安装在您的 AWS Cloud9 开发环境中。有关安装和使用这些可选构建工具的更多信息，请参阅以下资源：

- Gradle：[入门指南](#)
- Maven：[时长 5 分钟的 Maven](#)

Java 扩展的 Problems (问题) 选项卡

您可以在 AWS Cloud9 IDE 的 Problems (问题) 选项卡中，查看 Java 项目中 AWS Cloud9 环境的问题并进行故障排除。要显示 Problems (问题) 选项卡，请从 AWS Cloud9 IDE 中选择 View (查看)，然后从菜单栏中选择 Problem (问题)。



您也可以在控制台中选择 + 图标，然后选择 Open Problems (打开问题)，以此来打开 Problems (问题) 选项卡。当您在选项卡中选择问题时，它会打开受影响的文件并显示问题详细信息。

增强 TypeScript 支持和特征

AWS Cloud9 IDE 允许您使用 Typeject 以访问 TypeScript 的增强生产力功能。语言项目是 IDE 中用于 AWS Cloud9 开发环境的相关文件、文件夹和设置的集合。

要使用 IDE 在您的环境中创建语言项目，请参阅 [创建语言项目](#)。

可用的项目工作效率功能

AWS Cloud9 IDE 提供以下 TypeScript 项目工作效率功能。

自动完成

在编辑器中键入文件时，如果有符号可用，则在该上下文的插入点处显示符号列表。

若要在插入点从列表中插入符号，如果尚未选择符号，请使用上箭头或下箭头键选择该符号，然后按 Tab。

在您按 Tab 之前，您可能会看到一个屏幕提示，其中包含有关所选的符号 (如果信息可用)。

若不插入符号，要关闭列表，请按 `Esc`。

间距图标

图标可能会出现在活动文件的间距中。在您运行代码前，这些图标会突出显示可能出现的问题，如警告和代码中的错误。

有关问题的更多信息，请将指针暂停在问题的图标上。

快速修复

在编辑器内的活动文件中，您可以显示有关编码错误和警告的信息，以及可以自动应用于该代码的可能修复。要显示错误或警告信息以及可能的修复，请选择代码中带有红色虚线下划线（用于错误）或灰色虚线（用于警告）的任何部分。或者，如果光标停留在带有红色或灰色虚线下划线的代码上，请按 `Option-Enter`（适用于 macOS），或 `Alt-Enter`（适用于 Linux 或 Windows）。要应用建议的修复，请在列表中选择修复，或使用箭头键选择修复，然后按 `Enter`。要打开或关闭鼠标点击的选择快速修复，请选择 `AWS Cloud9 > Preferences（首选项）> User Settings（用户设置）> Language（语言）> Hints & Warning（提示和警告）> Show Available Quick Fixes on Click（点击时显示可用的快速修复）`。

查找引用

在编辑器内的活动文件中，如果 IDE 具有这些引用的访问权限，您可以在插入点处显示对该符号的所有引用。

要进行此操作，请在符号内任意位置的插入点处运行 **Find References** 命令。例如：

- 右键单击插入点处，然后选择 `Find References（查找引用）`。
- 在菜单栏上，选择 `Go > Find Replace（查找 > 替换）`。
- 对于 macOS、Windows 或 Linux，默认按 `Shift-F3`。

如果引用可用，则会在活动文件顶部的该符号旁边打开一个窗格。该窗格包含引用符号的文件的列表。该窗格将显示列表中的第一个引用。要显示其他参照，请在列表中选择该参照。

要关闭窗口格，请选择关闭 (X) 图标，或按 `Esc`。

在以下情况下，**Find References** 命令可能被禁用，或者可能无法按预期工作：

- 在活动文件的项目中没有对该符号的引用。
- IDE 无法在活动文件的项目中找到该符号的部分或全部引用。

- IDE 无权访问活动文件的项目中引用该符号的一个或多个位置。

转到定义

在编辑器内的活动文件中，您可以从符号转到定义该符号的位置（如果 IDE 可以访问该定义）。

要进行此操作，请在符号内任意位置的插入点处运行 **Jump to Definition** 命令。例如：

- 右键单击插入点处，然后选择 Jump to Definition（跳到定义）。
- 在菜单栏上，选择 Go > Jump to Definition（转到 > 转到定义）。
- 对于 macOS、Windows 或 Linux，默认按 F3。

如果定义可用，即使该定义位于单独的文件中，插入点也将切换到该定义。

在以下情况下，**Jump to Definition** 命令可能被禁用，或者可能无法按预期工作：

- 符号是该语言的原语符号。
- IDE 无法在活动文件的项目中找到定义的位置。
- IDE 无法在活动文件项目中访问定义的位置。

Go to Symbol（转到符号）

您可以转到项目中的指定符号，如下所示。

1. 通过在编辑器中打开项目中的其中一个文件，使其处于活动状态。如果已打开该文件，请选择编辑器中的选项卡以确认该文件为活动文件。
2. 运行 **Go to Symbol** 命令。例如：
 - 选择 Go（前往）窗口按钮（放大镜图标）。在 Go to Anything（转到任意内容）框中，键入 @，然后开始键入符号。
 - 在菜单栏上，选择 Go > Go To Symbol（转到 > 转到符号）。在 Go（转到）窗口中，开始在 @ 后键入符号。
 - 对于 macOS，默认按 Command-2 或 Command-Shift-0；对于 Windows 或 Linux，默认按 Ctrl-Shift-0。在 Go（转到）窗口中，开始在 @ 后键入符号。

例如，若要在名为 toString 的项目中查找所有符号，开始键入 @toString（或如果已显示 @，在 @ 后开始键入 toString）。

3. 如果您在 Symbols (符号) 列表中看到了想要转到的符号，单击选择它。或者使用上箭头或下箭头键选择它，然后按 Enter。然后插入点将切换到该符号。

如果要转到的符号不在活动文件的项目中，则此过程可能无法取得预期效果。

创建语言项目

使用以下过程创建一个语言项目，该项目将使用 AWS Cloud9 IDE 中支持的项目工作效率功能。

Note

我们建议您对属于语言项目一部分的文件使用受支持的项目生产力功能。尽管您可以在不属于项目的文件上使用某些受支持的项目生产力功能，但这些功能可能会产生意外结果。例如，您可以使用 IDE 从不属于项目的环境的根级别文件中搜索引用和定义。然后，IDE 可能只搜索同一根级别的文件。这可能导致找不到引用或定义，即使这些引用或定义实际位于同一环境中其他位置的语言项目中。

创建 TypeScript 语言项目

1. 确保您已在环境中安装了 TypeScript。有关更多信息，请参阅 [适用于 AWS Cloud9 的 TypeScript 教程](#) 中的 [步骤 1：安装所需工具](#)。
2. 从环境的 IDE 中的终端会话中，切换到您想要创建项目的目录。如果目录不存在，请创建并切换到该目录。例如，以下命令在环境的根目录下 (~/environment 中) 创建名为 my-demo-project 的目录，然后切换到该目录。

```
mkdir ~/environment/my-demo-project
cd ~/environment/my-demo-project
```

3. 在要创建项目的目录的根目录中，运行 TypeScript 编译器，并使用 **--init** 选项。

```
tsc --init
```

如果此命令成功，TypeScript 编译器将在项目目录的根目录中创建 tsconfig.json 文件。您可以使用此文件定义各种项目设置，例如 TypeScript 编译器选项和项目中包含或排除的特定文件。

有关 tsconfig.json 文件的更多信息，请参阅以下网站：

- TypeScript 网站上的 [tsconfig.json 概览](#)。

- json.schemastore.org 网站上的 [tsconfig.json 架构](#)。

AWS Cloud9 集成开发环境 (IDE) 的菜单栏命令参考

以下列表介绍了 AWS Cloud9 IDE 中的默认菜单栏命令。如果未显示菜单栏，请选择 IDE 上边缘的细条以显示菜单栏。

- [AWS Cloud9 菜单](#)
- [文件菜单](#)
- [Edit Menu](#) (编辑菜单)
- [Find Menu](#) (查找菜单)
- [查看菜单](#)
- [Go Menu](#) (转到菜单)
- [Run Menu](#) (运行菜单)
- [工具菜单](#)
- [Window Menu](#) (窗口菜单)
- [Support Menu](#) (支持菜单)
- [Preview Menu](#) (预览菜单)
- [Other Menu Bar Commands](#) (其他菜单栏命令)

AWS Cloud9 菜单

命令	描述
Preferences	<p>请执行下列操作之一：</p> <ul style="list-style-type: none"> • 如果 Preferences (首选项) 选项卡未打开，则打开它。 • 如果 Preferences (首选项) 选项卡虽然打开但处于非活动状态，则使其处于活动状态。 • 如果 Preferences (首选项) 选项卡处于活动状态，则隐藏它。

命令	描述
	请参阅 使用项目设置 、 使用用户设置 、 使用键绑定 、 使用主题 以及 使用初始化脚本 。
Go To Your Dashboard (转到您的控制面板)	在单独的 Web 浏览器标签页中打开 AWS Cloud9 控制台。请参阅 创建环境 、 打开环境 、 更改环境设置 以及 删除环境 。
欢迎页面	打开 Welcome (欢迎) 选项卡。
Open Your Project Settings (打开您的项目设置)	为当前环境打开 <code>project.settings</code> 文件。请参阅 使用项目设置 。
Open Your User Settings (打开您的用户设置)	为当前用户打开 <code>user.settings</code> 文件。请参阅 使用用户设置 。
Open Your Keymap (打开您的键映射)	为当前用户打开 <code>keybindings.settings</code> 文件。请参阅 使用键绑定 。
Open Your Init Script (打开您的初始化脚本)	为当前用户打开 <code>init.js</code> 文件。请参阅 使用初始化脚本 。
Open Your Stylesheet (打开您的样式表)	为当前用户打开 <code>styles.css</code> 文件。请参阅 使用主题 。

文件菜单

命令	描述
New File (新建文件)	创建新的文件。
New From Template (根据模板新建)	根据选定文件模板创建新的文件。
Open	显示并转到 Navigate (导航) 窗口。
Open Recent (打开最近的)	打开选定的文件。
Save	保存当前文件。

命令	描述
Save As	使用不同的文件名和/或位置保存当前文件。
Save All (全部保存)	保存所有未保存的文件。
Revert to Saved (恢复为保存的)	放弃自上次保存以来对当前文件所做的更改。
Revert All to Saved (全部恢复为保存的)	放弃自上次保存以来对所有未保存的文件所做的更改。
Show File Revision History (显示文件修订历史记录)	在编辑器中查看和管理对当前文件的更改。请参阅 使用文件修订 。
Upload Local Files (上传本地文件)	显示 Upload Files (上传文件) 对话框，您可以在其中将本地电脑中的文件拖到环境中。
Download Project (下载项目)	将环境中的文件合并为一个 .zip 格式文件，您可以将该文件下载到本地电脑中。
Line Endings (行结尾)	使用 Windows (回车符加换行符) 或 Unix (仅换行符) 行结尾。
Close File (关闭文件)	关闭当前文件。
Close All Files (关闭所有文件)	关闭所有打开的文件。

Edit Menu (编辑菜单)

命令	描述
Undo	撤消上次操作。
撤消	重做上次撤消的操作。
剪切	将所选内容移动到剪贴板中。
复制	将选定内容复制到剪贴板中。

命令	描述
粘贴	将剪贴板的内容复制到选择点。
Keyboard Mode (键盘模式)	要使用的一组键绑定，如 Default、Vim、Emacs 或 Sublime。请参阅 使用键绑定 。
Selection > Select All (选择 > 全选)	选择所有可选的内容。
Selection > Split Into Lines (选择 > 拆分为几行)	在当前行末尾添加光标。
Selection > Single Selection (选择 > 单一选择)	清除以前选择的所有内容。
Selection > Multiple Selections > Add Cursor Up (选择 > 多重选择 > 在上面添加光标)	在活动光标的上一行添加一个光标。如果已添加光标，则在该光标上面再添加一个光标。
Selection > Multiple Selections > Add Cursor Down (选择 > 多重选择 > 在下面添加光标)	在活动光标的下一行添加一个光标。如果已添加一个光标，则在该光标下面再添加一个光标。
Selection > Multiple Selections > Move Active Cursor Up (选择 > 多重选择 > 向上移动活动光标)	在活动光标的上一行添加第二个光标。如果已添加第二个光标，则将第二个光标向上移动一行。
Selection > Multiple Selections > Move Active Cursor Down (选择 > 多重选择 > 向下移动活动光标)	在活动光标的下一行添加第二个光标。如果已添加第二个光标，则将第二个光标向下移动一行。
Selection > Multiple Selections > Add Next Selection Match (选择 > 多重选择 > 添加下一个选择匹配项)	包含选定内容之后的更多匹配选择。
Selection > Multiple Selections > Add Previous Selection Match (选择 > 多重选择 > 添加上一个选择匹配项)	包含所选内容之前的更多选择匹配项。
Selection > Multiple Selections > Merge Selection Range (选择 > 多重选择 > 合并选择范围)	在当前行末尾添加光标。

命令	描述
Selection > Select Word Right (选择 > 选择右侧的单词)	在所选内容中包含光标右侧的下一个单词。
Selection > Select Word Left (选择 > 选择左侧的单词)	在所选内容中包含光标左侧的下一个单词。
Selection > Select to Line End (选择 > 选择到行末尾)	将从光标到当前行末尾的部分包含到选定内容中
Selection > Select to Line Start (选择 > 选择到行开头)	在所选内容中包含从当前行开头到光标的内容。
Selection > Select to Document End (选择 > 选择到文档末尾)	在所选内容中包含从光标到当前文件末尾的内容。
Selection > Select to Document Start (选择 > 选择到文档开头)	在所选内容中包含从光标到当前文件开头的内容。
Line > Indent (行 > 缩进)	将选定内容缩进一个制表符位置。
Line > Outdent (行 > 减少缩进)	将选定内容减少缩进一个制表符。
Line > Move Line Up (行 > 向上移动一行)	将所选内容向上移动一行。
Line > Move Line Down (行 > 向下移动一行)	将所选内容向下移动一行。
Line > Copy Lines Up (行 > 向上复制行)	复制一行的内容，并将复制的内容粘贴到上一行。
Line > Copy Lines Down (行 > 向下复制行)	复制一行的内容，并将复制的内容粘贴到下一行。
Line > Remove Line (行 > 删除行)	删除当前的内容。
Line > Remove to Line End (行 > 删除到行末尾)	删除从光标到当前行末尾的内容。
Line > Remove to Line Start (行 > 删除到行开头)	删除从当前行开头到光标的内容。

命令	描述
Line > Split Line (行 > 拆分行)	将光标到行末尾的内容移到自己的一行中。
Text > Remove Word Right (文本 > 删除右侧的单词)	删除光标右侧的单词。
Text > Remove Word Left (文本 > 删除左侧的单词)	删除光标左侧的单词。
Text > Align (文本 > 对齐)	如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置。
Text > Transpose Letters (文本 > 置换字母)	置换所选内容。
Text > To Upper Case (文本 > 更改为大写)	将所选内容更改为全部大写。
Text > To Lower Case (文本 > 更改为小写)	将所选内容更改为全部小写。
Comment > Toggle Comment (注释 > 切换注释)	在所选的每一行开头添加行注释字符，或删除现有的注释字符。
Code Folding > Toggle Fold (代码折叠 > 切换折叠)	折叠代码或展开折叠的代码。
Code Folding > Unfold (代码折叠 > 展开)	展开选定的代码。
Code Folding > Fold Other (代码折叠 > 折叠其他)	折叠所有可折叠的元素，当前选择范围除外。
Code Folding > Fold All (代码折叠 > 全部折叠)	折叠所有可折叠的元素。
Code Folding > Unfold All (代码折叠 > 全部展开)	展开整个文件的代码折叠。
Code Formatting > Apply Code Formatting (代码格式 > 应用代码格式)	重新设置选定 JavaScript 代码的格式。

命令	描述
Code Formatting > Open Language & Formatting Preferences (代码格式 > 打开语言和格式首选项)	打开 Preferences (首选项) 选项卡的 Project Settings (项目设置) 部分以指定语言设置。

Find Menu (查找菜单)

有关更多信息，请参阅[查找和替换文本](#)。

命令	描述
查找	显示当前文档的查找和替换栏，光标位于 Find (查找) 表达式上。
Find Next (查找下一个)	转至当前文档中下一处与您最后输入的查询匹配的内容。
Find Previous (查找上一个)	转至当前文档中前一处与您最后输入的查询匹配的内容。
Replace	显示当前文档的查找和替换栏，光标位于 Replace With (替换为) 表达式上。
Replace Next (替换下一个)	将当前文档的查找和替换栏中的下一个 Find (查找) 匹配项替换为 Replace With (替换内容)。
Replace Previous (替换上一个)	将当前文档的查找和替换栏中的上一个 Find (查找) 匹配项替换为 Replace With (替换内容)。
Replace All (全部替换)	将当前文档的查找和替换栏中的所有 Find (查找) 匹配项替换为 Replace With (替换内容)。
在文件中查找	显示多个文件的查找和替换栏。

View Menu (查看菜单)

命令	描述
编辑器	显示选定的编辑器。
Open Files (打开的文件)	在 Environment (环境) 窗口中显示 Open Files (打开的文件) 列表，或隐藏显示的该列表。
Problems (问题)	在终端的 Problems (问题) 面板中，显示 Java 项目中环境的任何问题。您可以选择问题来打开目标文件。
Menu Bar (菜单栏)	显示菜单栏，或隐藏显示的菜单栏。
Tab Buttons (选项卡按钮)	显示选项卡，或隐藏显示的选项卡。
Gutter (间距)	显示间距，或隐藏显示的间距。
Status Bar (状态栏)	显示状态栏，或隐藏显示的状态栏。
控制台	显示 Console (控制台) 窗口，或隐藏显示的该窗口。
Layout > Single (布局 > 单个)	显示单个窗格。
Layout > Vertical Split (布局 > 垂直拆分)	显示两个窗格：顶部和底部。
Layout > Horizontal Split (布局 > 水平拆分)	显示两个并排的窗格。
Layout > Cross Split (布局 > 交叉拆分)	显示四个相同大小的窗格。
Layout > Split 1:2 (布局 > 1:2 拆分)	在左侧显示一个窗格，在右侧显示两个窗格。
Layout > Split 2:1 (布局 > 2:1 拆分)	在左侧显示两个窗格，在右侧显示一个窗格。
Font Size > Increase Font Size (字体大小 > 增大字体大小)	增大字体大小。

命令	描述
Font Size > Decrease Font Size (字体大小 > 减小字体大小)	减小字体大小。
语法	显示当前文档的语法类型。
Themes (主题)	显示 IDE 主题类型。
Wrap Lines (换行)	换行到当前窗格边缘，或停止换行（如果已换行）。
Wrap To Print Margin (换行到打印边距)	换行到当前打印边距边缘，或停止换行（如果已换行）。

Go Menu (转到菜单)

命令	描述
Go To Anything (转到任意内容)	显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口。
Go To Symbol (转到符号)	显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。
Go To File (转到文件)	显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。
Go To Command (转到命令)	显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口。
Go To Line (转到行)	显示 Go to Line (转到行) 模式中的 Go (转到) 窗口。
Next Error (下一个错误)	转至下一错误。
Previous Error (上一个错误)	转至前一错误。
Word Right (右侧的单词)	向右移一个单词。

命令	描述
Word Left (左侧的单词)	向左移一个单词。
Line End (行末尾)	转至当前行末尾。
Line Start (行开头)	转至当前行开头。
Jump to Definition (跳到定义)	转到光标所在位置变量或函数的定义。
Jump to Matching Brace (跳到匹配大括号)	转到当前范围中的匹配符号。
Scroll to Selection (滚动到所选内容)	滚动所选内容以更好地进行查看。

Run Menu (运行菜单)

命令	描述
Run	运行或调试当前应用程序。
Run Last (运行上次)	运行或调试上次运行的文件。
Run With (运行方式)	使用选定的运行程序运行或调试。请参阅 使用生成器、运行程序和调试程序 。
Run History (运行历史记录)	查看运行历史记录。
Run Configurations (运行配置)	选择要运行或调试的运行配置，或者创建或管理运行配置。请参阅 使用生成器、运行程序和调试程序 。
Show Debugger at Break (在断点处显示调试程序)	在运行代码到达断点时，显示 Debugger (调试程序) 窗口。
构建	生成当前文件。
Cancel Build (取消生成)	停止生成当前文件。
Build System (生成系统)	使用选定的生成系统生成。

命令	描述
Show Build Result (显示生成结果)	显示相关的生成结果。
Automatically Build Supported Files (自动生成支持的文件)	自动生成支持的文件。
Save All on Build (在生成时保存所有内容)	在生成时，保存所有相关的未保存文件。

工具菜单

命令	描述
Strip Trailing Space (删除尾随空格)	删除行末尾的空格。
Preview, Preview File (预览 > 预览文件)	在预览标签页中预览当前文档。
Preview (预览)、Preview Running Application (预览运行的应用程序)	在单独的 Web 浏览器标签页中预览当前应用程序。
Preview > Configure Preview URL (预览 > 配置预览 URL)	打开 Preferences (首选项) 选项卡的 Project Settings (项目设置) 部分以显示 Run & Debug > Preview URL (运行和调试 > 预览 URL) 框。
Preview > Show Active Servers (预览 > 显示活动服务器)	在 Process List (进程列表) 对话框中显示可用的活动服务器地址列表。
Process List (进程列表)	显示 Process List (进程列表) 对话框。
Show Autocomplete (显示自动完成)	显示代码完成上下文菜单。
Rename Variable (重命名变量)	开始重命名/重构所选内容。
Toggle Macro Recording (切换宏录制)	开始按键录制或停止录制 (如果已在录制)。
Play Macro (播放宏)	播放以前录制的按键。

Window Menu (窗口菜单)

命令	描述
Go	显示 Go (转到) 窗口，或隐藏显示的该窗口。
New Terminal (新建终端)	打开新的 Terminal (终端) 选项卡。
New Immediate Window (新建即时窗口)	打开新的 Immediate (即时) 选项卡。
共享	显示 Share this environment (共享此环境) 对话框。
Installer (安装程序)	显示 AWS Cloud9 Installer (Amazon Cloud9 安装程序) 对话框。
协作	显示 Collaborate (协作) 窗口，或隐藏显示的该窗口。
Outline (大纲)	显示 Outline (大纲) 窗口，或隐藏显示的该窗口。
AWS 资源	显示 AWS Resources (Amazon 资源)窗口，或隐藏显示的该窗口。
Environment	显示 Environment (环境) 窗口，或隐藏显示的该窗口。
Debugger (调试程序)	显示 Debugger (调试程序) 窗口，或隐藏显示的该窗口。
Navigation > Tab to the Right (导航 > 右侧的选项卡)	向右移一个选项卡。
Navigation > Tab to the Left (导航 > 左侧的选项卡)	向左移一个选项卡。
Navigation > Next Tab in History (导航 > 历史记录中的下一个选项卡)	转到下一个选项卡。

命令	描述
Navigation > Previous Tab in History (导航 > 历史记录中的上一个选项卡)	转至前一选项卡。
Navigation > Move Tab to Right (导航 > 向右移动选项卡)	向右移动当前选项卡。如果该选项卡已在最右侧，则在该位置创建一个拆分选项卡。
Navigation > Move Tab to Left (导航 > 向左移动选项卡)	向左移动当前选项卡。如果该选项卡已在最左侧，则在该位置创建一个拆分选项卡。
Navigation > Move Tab to Up (导航 > 向上移动选项卡)	将当前选项卡向上移动一个窗格。如果该选项卡已在最顶端，则在该位置创建一个拆分选项卡。
Navigation > Move Tab to Down (导航 > 向下移动选项卡)	将当前选项卡向下移动一个窗格。如果该选项卡已在最底端，则在该位置创建一个拆分选项卡。
Navigation > Go to Pane to Right (导航 > 转到右侧的窗格)	向右移一个窗格。
Navigation > Go to Pane to Left (导航 > 转到左侧的窗格)	向左移一个窗格。
Navigation > Go to Pane to Up (导航 > 转到上面的窗格)	向上移一个窗格。
Navigation > Go to Pane to Down (导航 > 转到下面的窗格)	向下移一个窗格。
Navigation > Switch Between Editor and Terminal (导航 > 在编辑器和终端之间切换)	在编辑器和 Terminal (终端) 选项卡之间切换。
Navigation > Next Pane in History (导航 > 历史记录中的下一个窗格)	转到下一个窗格。
Navigation > Previous Pane in History (导航 > 历史记录中的上一个窗格)	转到上一个窗格。

命令	描述
Saved Layouts > Save (保存的布局 > 保存)	保存当前布局。要以后切换到该布局，请选择 Saved Layouts > LAYOUT-ID (保存的布局 > 布局 ID)。
Saved Layouts > Save and Close All (保存的布局 > 保存并全部关闭)	保存当前布局，然后关闭所有选项卡和窗格。
Saved Layouts > Show Saved Layouts in File Tree (保存的布局 > 在文件树中显示保存的布局)	在 Environment (环境) 窗口中显示所有保存的布局。
Tabs > Close Pane (选项卡 > 关闭窗格)	关闭当前窗格。
Tabs > Close All Tabs In All Panes (选项卡 > 关闭所有窗格中的所有选项卡)	关闭所有窗格中所有打开的选项卡。
Tabs > Close All But Current Tab (选项卡 > 关闭当前选项卡以外的所有选项卡)	关闭当前窗格中所有打开的选项卡 (当前选项卡除外)。
Tabs > Split Pane in Two Rows (选项卡 > 将窗格拆分为两排)	将当前窗格拆分为两个窗格：顶部和底部。
Tabs > Split Pane in Two Columns (选项卡 > 将窗格拆分为两列)	将当前窗格拆分为两个窗格：左窗格和右窗格。
Presets > Full IDE (预设 > 完整 IDE)	切换到完整 IDE 模式。
Presets > Minimal Editor (预设 > 最小编辑器)	切换到最小编辑器模式。
Presets > Sublime Mode (预设 > Sublime 模式)	切换到 Sublime 模式。

Support Menu (支持菜单)

命令	描述
欢迎页面	打开 Welcome (欢迎) 选项卡。

命令	描述
Get Help (Community) (获得帮助 (社区))	在单独的 Web 浏览器标签页中打开 AWS Cloud9 在线社区网站。
Read Documentation (阅读文档)	在单独的 Web 浏览器标签页中打开 AWS Cloud9 用户指南。

Preview Menu (预览菜单)

命令	描述
Preview File (预览文件)	在预览标签页中预览当前文档。
Preview Running Application (预览运行的应用程序)	在单独的 Web 浏览器标签页中预览当前应用程序。
Configure Preview URL (配置预览 URL)	打开 Preferences (首选项) 选项卡的 Project Settings (项目设置) 部分以显示 Run & Debug > Preview URL (运行和调试 > 预览 URL) 框。
Show Active Servers (显示活动服务器)	在 Process List (进程列表) 对话框中显示可用的活动服务器地址列表。

Other Menu Bar Commands (其他菜单栏命令)

命令	描述
Run	运行或调试当前应用程序。
共享	打开 Share this environment (共享此环境) 对话框。
Preferences (首选项) (齿轮图标)	打开 Preferences (首选项) 选项卡。

在 AWS Cloud9 集成开发环境 (IDE) 中查找和替换文本

您可以在 AWS Cloud9 IDE 中使用查找和替换栏查找和替换一个或多个文件中的文本。

- [查找单个文件中的文本](#)
- [替换单个文件中的文本](#)
- [查找多个文件中的文本](#)
- [替换多个文件中的文本](#)
- [查找和替换选项](#)

查找单个文件中的文本

1. 打开要在其中查找文本的文件。如果已打开该文件，请选择该文件的选项卡以使其处于活动状态。
2. 在菜单栏上，选择 Find > Find (查找 > 查找)。
3. 在查找和替换栏中，为 Find (查找) 键入要查找的文本。
4. 要指定额外的查找选项，请参阅[查找和替换选项](#)。
5. 如果具有任何匹配项，Find (查找) 框中的 0 of 0 (第 0 个 (共 0 个)) 将变为非零数字。如果具有任何匹配项，编辑器将转到第一个匹配项。如果具有多个匹配项，要转到下一个匹配项，请选择 Find (查找) 框中的右箭头，或者在菜单栏上选择 Find > Find Next (查找 > 查找下一个)。要转到上一个匹配项，请选择 Find (查找) 框中的左箭头，或者在菜单栏上选择 Find > Find Previous (查找 > 查找上一个)。

替换单个文件中的文本

1. 打开要在其中替换文本的文件。如果已打开该文件，请选择该文件的选项卡以使其处于活动状态。
2. 在菜单栏上，选择 Find > Replace (查找 > 替换)。
3. 在查找和替换栏中，为 Find (查找) 键入要查找的文本。
4. 对于 Replace With (替换为)，请键入要将 Find (查找) 中的文本替换为的文本。
5. 要指定额外的查找和替换选项，请参阅[查找和替换选项](#)。
6. 如果具有任何匹配项，Find (查找) 框中的 0 of 0 (第 0 个 (共 0 个)) 将变为非零数字。如果具有任何匹配项，编辑器将转到第一个匹配项。如果具有多个匹配项，要转到下一个匹配项，请选择 Find (查找) 框中的右箭头，或者在菜单栏上选择 Find > Find Next (查找 > 查找下一个)。要转到上一个匹配项，请选择 Find (查找) 框中的左箭头，或者在菜单栏上选择 Find > Find Previous (查找 > 查找上一个)。

7. 要将当前匹配项替换为 Replace With (替换为) 中的文本，然后转到下一个匹配项，请选择 Replace (替换)。要将所有匹配项替换为 Replace With (替换为) 中的文本，请选择 Replace All (全部替换)。

查找多个文件中的文本

1. 在菜单栏上，选择 Find > Find in Files (查找 > 在文件中查找)。
2. 在查找和替换栏中，为 Find (查找) 键入要查找的文本。
3. 要指定额外的查找选项，请参阅[查找和替换选项](#)。
4. 在 Find (查找) 按钮右侧的框 (带有 *.* , -* 的框) 中，键入要在查找中包括或排除的任何一组文件。例如：
 - 空白、* 或 *.* : 查找所有文件。
 - my-file.txt : 仅查找名为 my-file.txt 的文件。
 - my* : 仅查找文件名以 my 开头的文件。
 - my*.txt : 仅查找文件名以 my 开头并具有 .txt 文件扩展名的文件。
 - my*.htm* : 查找文件名以 my 开头并且文件扩展名以 .htm 开头的文件。
 - my*.htm, my*.html : 查找文件名以 my 开头并具有 .htm 或 .html 文件扩展名的所有文件。
 - -my-file.txt : 不搜索名为 my-file.txt 的文件。
 - -my* : 不搜索以 my 开头的任何文件。
 - -my*.htm* : 不搜索文件名以 my 开头并且文件扩展名以 .htm 开头的任何文件。
 - my*.htm*, -my*.html : 搜索文件名以 my 开头并且文件扩展名以 .htm 开头的文件。不过，不会搜索文件名以 my 开头并具有 .html 文件扩展名的任何文件。
5. 在上述框旁边的下拉列表中，选择以下选项之一以进一步将查找限制为仅特定的位置：
 - Environment (环境) : 仅查找 Environment (环境) 窗口中的文件。
 - Project (excludes .gitignore'd) (项目 (不包括 .gitignore)) : 查找环境中的任何文件，但环境的 .gitignore 文件中列出的文件或文件类型除外 (如果 .gitignore 文件存在)。
 - Selection: (选择 :) : 仅查找当前在 Environment (环境) 窗口中选择的文件。

Note

要进一步将查找限制为仅单个文件夹，请在 Environment (环境) 窗口中选择一个文件夹，然后选择 Selection (选择内容)。或者，您也可以直接在 Environment (环境) 窗口中

右键单击该文件夹，然后在上下文菜单中选择 Search In This Folder (在该文件夹中搜索)。

- Favorites (收藏夹) : 仅查找 Environment (环境) 窗口的 Favorites (收藏夹) 列表中的文件。
 - Active File (活动文件) : 仅查找活动文件。
 - Open Files (打开的文件) : 仅查找 Environment (环境) 窗口的 Open Files (打开的文件) 列表中的文件。
6. 选择 Find (查找) 。
 7. 要转到包含匹配项的文件，请在 Search Results (搜索结果) 选项卡上双击该文件名。要转到特定的匹配项，请在 Search Results (搜索结果) 选项卡中双击该匹配项。

替换多个文件中的文本

1. 在菜单栏上，选择 Find > Find in Files (查找 > 在文件中查找) 。
2. 在查找和替换栏中，为 Find (查找) 键入要查找的文本。
3. 要指定额外的查找选项，请参阅[查找和替换选项](#)。
4. 在 Find (查找) 按钮右侧的框 (带有 *.* , -* 的框) 中，键入要在查找中包括或排除的任何一组文件。例如：
 - 空白、* 或 *.* : 所有文件。
 - my-file.txt : 仅名为 my-file.txt 的文件。
 - my* : 仅文件名以 my 开头的文件。
 - my*.txt : 仅文件名以 my 开头并具有 .txt 文件扩展名的文件。
 - my*.htm* : 文件名以 my 开头并且文件扩展名以 .htm 开头的所有文件。
 - my*.htm, my*.html : 文件名以 my 开头并具有 .htm 或 .html 文件扩展名的所有文件。
 - -my-file.txt : 不搜索名为 my-file.txt 的文件。
 - -my* : 不搜索以 my 开头的任何文件。
 - -my*.htm* : 不搜索文件名以 my 开头并且文件扩展名以 .htm 开头的任何文件。
 - my*.htm*, -my*.html : 搜索文件名以 my 开头并且文件扩展名以 .htm 开头的所有文件。不过，不会搜索文件名以 my 开头并具有 .html 文件扩展名的任何文件。
5. 在上述框旁边的下拉列表中，选择以下选项之一以进一步将查找限制为仅特定的位置：
 - Environment (环境) : 仅 Environment (环境) 窗口中的文件。

- Project (excludes .gitignore'd) (项目 (不包括 .gitignore)) : 查找环境中的任何文件, 但环境的 .gitignore 文件中列出的文件或文件类型除外 (如果 .gitignore 文件存在)。
 - Selection (选择内容) : 仅当前选择的文件。
 - Favorites (收藏夹) : 仅 Environment (环境) 窗口的 Favorites (收藏夹) 列表中的文件。
 - Active File (活动文件) : 仅活动文件。
 - Open Files (打开的文件) : 仅 Environment (环境) 窗口的 Open Files (打开的文件) 列表中的文件。
- 对于 Replace With (替换为), 请键入要将 Find (查找) 内容替换为的文本。
 - 选择 Replace (替换)。

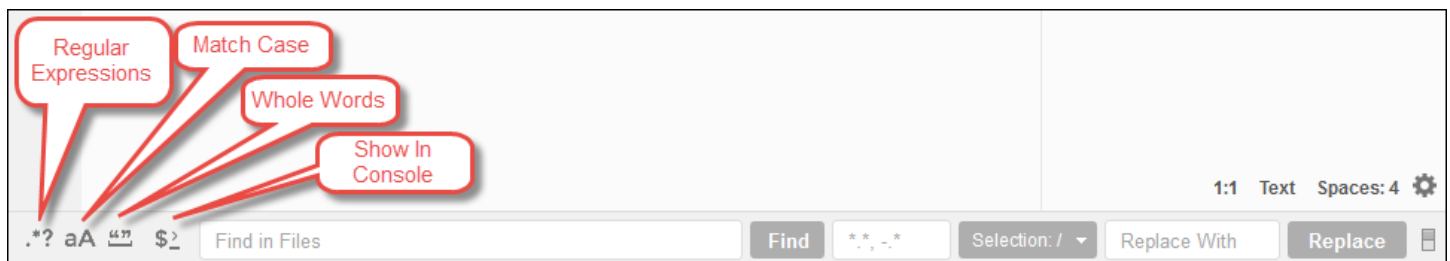
Note

将在范围内的所有文件中立即执行替换操作。无法方便地撤消该操作。如果要在开始执行替换操作之前了解将更改的内容, 请选择 Find (查找)。

- 要转到包含替换内容的文件, 请在 Search Results (搜索结果) 选项卡中双击该文件名。要转到特定的替换内容, 请在 Search Results (搜索结果) 窗格中双击该替换内容。

查找和替换选项

可以在查找和替换栏上选择任何以下按钮以修改查找和替换操作。



- **Regular Expressions (正则表达式)** : 查找与 Find (查找) 或 Find in Files (在文件中查找) 中指定的正则表达式匹配的文本。请参阅 Mozilla Developer Network 上的 [JavaScript 正则表达式](#) 主题中的编写正则表达式模式。
- **Match Case (匹配大小写)** : 查找与 Find (查找) 或 Find in Files (在文件中查找) 中指定的大小写匹配的文本。
- **Whole Words (整个单词)** : 使用标准单词字符规则查找 Find (查找) 或 Find in Files (在文件中查找) 中的文本。
- **Wrap Around (折回)** : 仅限单个文件，在转到下一个或上一个匹配项时，不要在文件末尾或开头处停止。
- **Search Selection (搜索选择内容)** : 仅限单个文件，仅在选择内容中查找。
- **Show in Console (在控制台中显示)** : 对于多个文件，在控制台中显示 Search Results (搜索结果) 选项卡，而不是在活动窗格中显示。
- **Preserve Case (保留大小写)** : 仅限单个文件，在替换文本时，保留大小写 (如果适用)。

预览 AWS Cloud9 集成开发环境 (IDE) 中的文件

您可以使用 AWS Cloud9 IDE 预览在 IDE 中的 AWS Cloud9 开发环境中的文件。

- [打开要预览的文件](#)
- [重新加载文件预览](#)
- [更改文件预览类型](#)
- [在单独的 Web 浏览器标签页中打开文件预览](#)
- [切换到其他文件预览](#)

打开要预览的文件

在 AWS Cloud9 IDE 中选择以下选项之一，以便在环境中打开文件预览标签页：

- 在环境窗口中，打开要预览文件的上下文 (右键单击) 菜单，然后选择预览。

Note

虽然可以使用此方法来预览任何文件，但带有以下文件扩展名的文件最适合预览：

- .htm

- .html
- .pdf
- .svg
- .xhtml
- 包含 Markdown 格式的内容的任何文件。

- 打开带有下列文件扩展名之一的文件：
 - .pdf
 - .svg
- 在打开并已激活您要预览的文件后，在菜单栏上依次选择 Preview (预览)、Preview File FILE_NAME (预览文件 FILE_NAME)。或者依次选择 Tools (工具)、Preview (预览)、Preview File FILE_NAME (预览文件 FILE_NAME)，其中 FILE_NAME 是要预览的文件的名称。

Note

这些命令仅适用于以下文件类型：

- .htm
- .html
- .markdown
- .md
- .pdf
- .svg
- .txt：如果文件内容是 Markdown 格式，则预览效果最好。
- .xhtml：如果文件包含或引用内容展示信息，则预览效果最好。

Note

文件预览标签页的 Preview Settings (预览设置) 菜单当前不起作用，选择其中的任何菜单命令都无效。

重新加载文件预览

在文件预览标签页上，选择 Refresh (刷新) 按钮 (圆形箭头)。

更改文件预览类型

在文件预览标签页上，从预览类型列表中选择以下选项之一：

- Browser (浏览器)：以 Web 浏览器格式预览文件，仅适用于以下文件类型：
 - .htm
 - .html
 - .pdf
 - .svg
 - .xhtml：如果文件包含或引用内容展示信息，则预览效果最好。
- Raw Content (UTF-8) (原始内容 (UTF-8))：使用 Unicode 转换格式 8 位 (UTF-8) 格式预览文件的原始内容。对于某些文件类型，显示的内容可能不正常。
- Markdown：预览包含 Markdown 格式的任何文件。尝试预览任何其他文件类型，但可能显示的内容不正常。

在单独的 Web 浏览器标签页中打开文件预览

在文件预览标签页上，选择 Pop Out Into New Window (在新的弹出窗口中显示)。

切换到其他文件预览

在文件预览标签页上，在地址栏中键入指向不同文件的路径。地址栏位于 Refresh (刷新) 按钮和预览类型列表之间。

在 AWS Cloud9 集成开发环境 (IDE) 中预览正在运行的应用程序

您可以使用 AWS Cloud9 IDE 在 IDE 中预览正在运行的应用程序。

主题

- [运行应用程序](#)
- [预览运行的应用程序](#)
- [重新加载应用程序预览](#)

- [更改应用程序预览类型](#)
- [在单独的 Web 浏览器标签页中打开应用程序预览](#)
- [切换到不同的预览 URL](#)
- [通过互联网共享运行的应用程序](#)

运行应用程序

必须先要在 AWS Cloud9 开发环境中运行应用程序，然后才能在 IDE 中预览应用程序。它必须通过以下端口使用 HTTP：

- 8080
- 8081
- 8082

以上所有端口都必须使用 IP 地址 127.0.0.1、localhost 或 0.0.0.0。

Note

您无需使用 HTTP 通过端口 8080、8081 或 8082，IP 地址 127.0.0.1、localhost 或 0.0.0.0 来运行应用程序。不过，如果您不这样做，就将无法从 IDE 中预览运行的应用程序。

Note

预览应用程序在 IDE 中运行并加载在 iframe 元素中。默认情况下，某些应用程序服务器可能会阻止来自 iframe 元素（例如 X-Frame-Options 标头）的请求。如果您的预览应用程序未显示在预览标签页中，请确保您的应用程序服务器不禁止在 iframe 中显示内容。

要编写代码以在特定端口和 IP 地址上运行您的应用程序，请参阅您的应用程序文档。

要运行您的应用程序，请参阅[运行代码](#)。

要测试此行为，请将以下 JavaScript 代码添加到环境根目录 `server.js` 中命名的文件中。该代码使用名为 Node.js 的文件运行服务器。

Note

在以下示例中，text/html 是所返回内容的 Content-Type。要以不同的格式返回内容，请指定不同的 Content-Type。例如，您可以对 CSS 文件格式指定 text/css。

```
var http = require('http');
var fs = require('fs');
var url = require('url');

http.createServer( function (request, response) {
  var pathname = url.parse(request.url).pathname;
  console.log("Trying to find '" + pathname.substr(1) + "...");

  fs.readFile(pathname.substr(1), function (err, data) {
    if (err) {
      response.writeHead(404, {'Content-Type': 'text/html'});
      response.write("ERROR: Cannot find '" + pathname.substr(1) + "'.");
      console.log("ERROR: Cannot find '" + pathname.substr(1) + "'.");
    } else {
      console.log("Found '" + pathname.substr(1) + "'.");
      response.writeHead(200, {'Content-Type': 'text/html'});
      response.write(data.toString());
    }
    response.end();
  });
}).listen(8080, 'localhost'); // Or 8081 or 8082 instead of 8080. Or '127.0.0.1'
instead of 'localhost'.
```

在环境根目录下，您可以将以下 Python 代码添加到名为 server.py 的文件中。在以下示例中，服务器使用 Python 运行。

```
import os
import http.server
import socketserver

ip = 'localhost' # Or '127.0.0.1' instead of 'localhost'.
port = '8080' # Or '8081' or '8082' instead of '8080'.
Handler = http.server.SimpleHTTPRequestHandler
httpd = socketserver.TCPServer((ip, int(port)), Handler)
httpd.serve_forever()
```

在环境根目录下，将以下 HTML 代码添加到名为 `index.html` 的文件中。

```
<html>
  <head>
    <title>Hello Home Page</title>
  </head>
  <body>
    <p style="font-family:Arial;color:blue">Hello, World!</p>
  </body>
</html>
```

要在应用程序预览标签页上查看该文件的 HTML 输出，请使用 Node.js 运行 `server.js`，或使用 Python 运行 `server.py` 文件。然后请按照下一节中的步骤预览它。在应用程序预览标签页上，将 `/index.html` 添加到 URL 末尾，然后按 Enter。

预览运行的应用程序

在预览您的应用程序之前，请确认以下事项：

- 您的应用程序使用 HTTP 协议通过端口 8080、8081 或 8082 运行。
- 您的应用程序在环境中的 IP 地址是 127.0.0.1、localhost 或 0.0.0.0。
- 您的应用程序代码文件已在 AWS Cloud9 IDE 中打开并处于活动状态。

确认所有这些详细信息后，从菜单栏中选择以下选项之一：

- Preview (预览)、Preview Running Application (预览运行的应用程序)
- Tools (工具)、Preview (预览)、Preview Running Application (预览运行的应用程序)

这些选项之一在环境中打开应用程序预览标签页，然后在标签页上显示应用程序的输出。

Note

如果应用程序预览标签页显示错误或空白，请执行 [应用程序预览选项卡显示错误或空白](#) 中的故障排除步骤。如果您在尝试预览应用程序或文件时收到以下通知：“预览功能已禁用，因为您的浏览器禁用了第三方 cookie”，请按照[应用程序预览或文件预览通知：“Third-party cookies disabled \(第三方 cookie 已禁用\)”](#)中的故障排除步骤进行操作。

Note

如果该应用程序尚未运行，将会在应用程序预览标签页上显示错误。要解决此问题，请运行或重新启动该应用程序，然后再次选择菜单栏命令。

例如，假设您的应用程序无法在上述任何端口或 IP 上运行。或者，您的应用程序必须同时在多个端口上运行。例如，您的应用程序必须同时在端口 8080 和 3000 上运行。如果是这样，则应用程序预览标签页可能会显示错误或为空。这是因为环境中的应用程序预览标签页仅适用于上述端口和 IP。此外，该应用程序一次只能使用一个端口。

我们建议不要与其他人共享应用程序预览标签页中的 URL。(网址采用以下格式：`https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-`

`east-2.amazonaws.com/`。在这种格式中，12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 分配给环境的 ID。us-east-2 是环境 AWS 区域的 ID。) 仅当环境的 IDE 处于打开状态并且应用程序正在同一个 Web 浏览器中运行时，此 URL 才有用。

如果您尝试使用 IDE 中的应用程序预览选项卡或 0.0.0.0 在 IDE 之外的单独的 Web 浏览器选项卡中访问、或的 IP，则默认情况下，AWS Cloud9 IDE 会尝试访问您的本地计算机，而不是连接到环境的实例或您自己的服务器。127.0.0.1 localhost

有关如何在 IDE 之外向其他人提供正在运行的应用程序预览的说明，请参阅[通过互联网共享运行的应用程序](#)。

重新加载应用程序预览

在应用程序预览选项卡上，选择 Refresh (刷新) 按钮 (圆形箭头)。

Note

该命令不会重新启动服务器。它仅刷新应用程序预览标签页的内容。

更改应用程序预览类型

在应用程序预览标签页上，在预览类型列表中选择以下选项之一：

- Browser (浏览器)：以 Web 浏览器格式预览输出。
- Raw Content (UTF-8) (原始内容 (UTF-8))：尝试以 Unicode 转换格式 8 位 (UTF-8) 格式预览输出 (如果适用)。
- Markdown：尝试以 Markdown 格式 (如果适用) 预览输出。

在单独的 Web 浏览器标签页中打开应用程序预览

在应用程序预览选项卡上，选择 Pop Out Into New Window (在新的弹出窗口中显示)。

Note

AWS Cloud9 IDE 还必须在同一 Web 浏览器的至少一个其他选项卡中运行。否则，应用程序预览不会在单独的 Web 浏览器标签页中显示。

AWS Cloud9 IDE 还必须在同一 Web 浏览器的至少一个其他选项卡中运行。否则，应用程序预览不会在单独的 Web 浏览器标签页中显示。如果应用程序预览标签页显示错误或空白，请执行 [应用程序预览或文件预览通知：“Third-party cookies disabled \(第三方 cookie 已禁用 \)”](#) 中的故障排除步骤。

切换到不同的预览 URL

在应用程序预览标签页上，在地址栏中键入不同 URL 的路径。地址栏位于 Refresh (刷新) 按钮和预览类型列表之间。

通过互联网共享运行的应用程序

在预览运行的应用程序后，您可以通过 Internet 为其他人提供该应用程序。

如果 Amazon EC2 实例已连接到您的环境，请执行以下步骤。否则，请查阅您的服务器的文档。

主题

- [步骤 1：获取实例的 ID 和 IP 地址](#)
- [步骤 2：为实例设置安全组](#)
- [步骤 3：为实例设置子网](#)
- [步骤 4：共享运行的应用程序 URL](#)

步骤 1：获取实例的 ID 和 IP 地址

在该步骤中，请记住连接到环境的 Amazon EC2 实例的实例 ID 和公有 IP 地址。您需要在后面的步骤中使用实例 ID 以允许传入应用程序请求。然后，与其他人分享该公有 IP 地址，以便他们可以访问运行的应用程序。

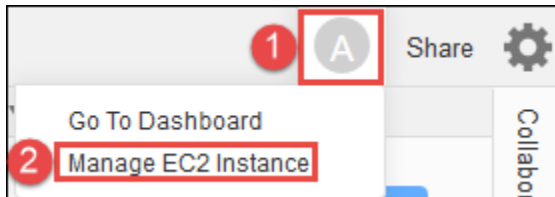
1. 获取 Amazon EC2 实例的 ID。要获取该 ID，请执行下列操作之一：

- 在 AWS Cloud9 IDE 环境的终端会话中，运行以下命令以获取 Amazon EC2 实例的 ID。

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

实例 ID 采用以下格式：i-12a3b456c789d0123。记下该实例 ID。

- 在环境的 IDE 中，在菜单栏上选择您的用户图标，然后选择 Manage EC2 Instance（管理 EC2 实例）。



在显示的 Amazon EC2 控制台中，记下在 Instance ID（实例 ID）列中显示的实例 ID。实例 ID 采用以下格式：i-12a3b456c789d0123。

2. 获取 Amazon EC2 实例的公有 IP 地址。要获取该 ID，请执行下列操作之一：

- 在环境的 IDE 中，在菜单栏上选择 Share（共享）。在 Share this environment（共享此环境）对话框中，记下 Application（应用程序）框中的公有 IP 地址。公有 IP 地址采用以下格式：192.0.2.0。
- 在环境 IDE 的终端会话中，运行以下命令以获取 Amazon EC2 实例的公有 IP 地址。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

公有 IP 地址采用以下格式：192.0.2.0。记下该公有 IP 地址。

- 在环境的 IDE 中，在菜单栏上选择您的用户图标，然后选择 Manage EC2 Instance（管理 EC2 实例）。在显示的 Amazon EC2 控制台中，在 Description（描述）选项卡上，记下 IPv4 Public IP（IPv4 公有 IP）字段中的公有 IP 地址。公有 IP 地址采用以下格式：192.0.2.0。

Note

您的应用程序的公有 IP 地址可能会在应用程序的实例重启时发生变化。要防止您的 IP 地址发生变化，请分配弹性 IP 地址。然后，将该地址分配给正在运行的实例。有关说明，请参阅 Amazon EC2 用户指南中的[分配弹性 IP 地址和将弹性 IP 地址与正在运行的实例](#)

[关联起来](#)。分配弹性 IP 地址可能会 AWS 账户 导致您产生费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

步骤 2：为实例设置安全组

在此步骤中，在 Amazon EC2 控制台上，为连接到环境的实例设置 Amazon EC2 安全组。将其设置为允许通过端口 8080、8081 或 8082 传入 HTTP 请求。

Note

您无需通过端口 8080、8081 或 8082 使用 HTTP 运行。如果您不这样做，将无法从 IDE 中预览运行的应用程序。有关更多信息，请参阅 [预览运行的应用程序](#)。如果在不同的协议或端口上运行，请在该步骤中替换为该协议或端口。

要提供额外的安全层，请为实例可使用的 VPC 中的子网设置网络访问控制列表 (ACL)。有关安全组和网络 ACL 的更多信息，请参阅以下内容：

- [步骤 3：为实例设置子网](#)
- Amazon VPC 用户指南中的 [安全](#)
- Amazon VPC 用户指南中的 [您的 VPC 安全组](#)
- Amazon VPC 用户指南中的 [网络 ACL](#)

1. 在环境的 IDE 中，在菜单栏上选择您的用户图标，然后选择 Manage EC2 Instance (管理 EC2 实例)。然后，跳到本过程的步骤 3。
2. 如果选择 Manage EC2 Instance (管理 EC2 实例) 或者该过程中的其他步骤返回错误，请使用您的 AWS 账户中管理员的凭证登录到 Amazon EC2 控制台。然后按照以下说明进行操作。如果您无法执行此操作，请咨询您的 AWS 账户 管理员。
 - a. 如果你尚未登录，请登录 <https://console.aws.amazon.com/>。AWS Management Console
 - b. 打开 Amazon EC2 控制台。为此，请在导航栏中选择 Services (服务)。然后选择 EC2。
 - c. 在导航栏中，选择您的环境所在 AWS 区域 的位置。
 - d. 如果显示 EC2 控制面板，请选择 Running Instances (正在运行的实例)。否则，在服务导航窗格中展开 Instances (实例) (如果尚未展开)，然后选择 Instances (实例)。
 - e. 在实例列表中，选择 Instance ID (实例 ID) 与您之前记下的实例 ID 相匹配的实例。
3. 在实例的 Description (描述) 选项卡中，选择 Security groups (安全组) 旁边的安全组链接。

- 在显示安全组的情况下，查看 Inbound (入站) 选项卡。如果有一个规则，其中 Type (类型) 设置为 Custom Tcp Rule (自定义 TCP 规则) ， Port Range (端口范围) 设置为 8080、8081 或 8082，请选择 Cancel (取消) ，然后向前跳到[步骤 3：为实例设置子网](#)。否则，请选择 Edit (编辑) 。
- 在 Edit inbound rules (编辑入站规则) 对话框中，选择 Add Rule (添加规则) 。
- 对于 Type (类型) ，选择 Custom TCP Rule (自定义 TCP 规则) 。
- 对于 Port Range (端口范围) ，输入 8080、8081 或 8082。
- 对于 Source (源) ，请选择 Anywhere (任何位置) 。

Note

通过对 Source (源) 选择 Anywhere (任何位置) ，则允许来自任何 IP 地址的传入请求。要将其限定为特定 IP 地址，请选择 Custom (自定义) ，然后输入 IP 地址范围。或者，请选择 My IP (我的 IP) 将请求限制为仅来自您的 IP 地址。

- 选择 Save (保存) 。

步骤 3：为实例设置子网

使用 Amazon EC2 和 Amazon VPC 控制台为连接到环境的 Amazon EC2 实例设置子网。然后，允许通过端口 8080、8081 或 8082 传入 HTTP 请求。

Note

您无需通过端口 8080、8081 或 8082 使用 HTTP 运行。不过，如果您不这样做，将无法从 IDE 中预览运行的应用程序。有关更多信息，请参阅[预览运行的应用程序](#)。如果在不同的协议或端口上运行，请在该步骤中替换为该协议或端口。

该步骤介绍如何为实例可使用的 Amazon VPC 中的子网设置网络 ACL。这不是必须执行的，但建议执行。设置网络 ACL 可增加额外的安全层。有关网络 ACL 的更多信息，请参阅以下内容：

- Amazon VPC 用户指南中的[安全](#)
- Amazon VPC 用户指南中的[网络 ACL](#)

- 在 Amazon EC2 控制台的服务导航窗格中，展开 Instances (实例) (如果尚未展开) ，然后选择 Instances (实例) 。

2. 在实例列表中，选择 Instance ID (实例 ID) 与您之前记下的实例 ID 相匹配的实例。
3. 在实例的 Description (描述) 选项卡中，记下 Subnet ID (子网 ID) 值。子网 ID 采用以下格式：subnet-1fab8aEX。
4. 打开 Amazon VPC 控制台。为此，请在 AWS 导航栏中选择“服务”，然后选择 VPC。

对于此步骤，建议使用您的 AWS 账户中管理员的凭证登录到 Amazon VPC 控制台。如果您无法执行此操作，请咨询您的 AWS 账户 管理员。

5. 如果显示 VPC Dashboard (VPC 控制面板)，请选择 Subnet (子网)。否则，在服务导航窗格中，选择 Subnets (子网)。
6. 在子网列表中，选择 Subnet ID (子网 ID) 值与您之前记下的子网 ID 相匹配的子网。
7. 在 Summary (摘要) 选项卡上，选择 Network ACL (网络 ACL) 旁边的网络 ACL 链接。
8. 在网络 ACL 列表中，选择网络 ACL。(只有一个网络 ACL。)
9. 查看网络 ACL 的 Inbound Rules (入站规则) 选项卡。如果已有一个规则，其中 Type (类型) 设置为 HTTP* (8080)、HTTP* (8081) 或 HTTP* (8082)，请向前跳到 [步骤 4：共享运行的应用程序 URL](#)。否则，请选择 Edit (编辑)。
10. 选择 Add another rule (添加其他规则)。
11. 对于 Rule # (规则号)，请输入该规则的编号 (例如 200)。
12. 对于 Type (类型)，选择 Custom TCP Rule (自定义 TCP 规则)。
13. 对于 Port Range (端口范围)，请键入 8080、8081 或 8082。
14. 对于 Source (源)，请键入允许发送传入请求的 IP 地址范围。例如，要允许来自任何 IP 地址的传入请求，请输入 0.0.0.0/0。
15. 在将 Allow/Deny (允许/拒绝) 设置为 ALLOW (允许) 的情况下，选择 Save (保存)。

步骤 4：共享运行的应用程序 URL

应用程序运行后，您可以通过提供应用程序的 URL 与其他人共享您的应用程序。为此，您需要之前提到的公有 IP 地址。要编写应用程序的完整 URL，请确保使用正确的协议启动应用程序的公有 IP 地址。接下来，如果您的应用程序端口不是它所用协议的默认端口，请添加端口号信息。以下是一个示例应用程序 URL：`http://192.0.2.0:8080/index.html` 通过端口 8080 使用 HTTP。

如果出现的 Web 浏览器标签页显示错误或空白，请执行[无法显示在 IDE 以外运行的应用程序](#)中的故障排除步骤。

Note

您的应用程序的公有 IP 地址可能会在应用程序的实例重启时发生变化。要禁止更改该 IP 地址，请分配弹性 IP 地址，然后将该地址分配给正在运行的实例。有关说明，请参阅 Amazon EC2 用户指南中的[分配弹性 IP 地址和将弹性 IP 地址与正在运行的实例关联起来](#)。分配弹性 IP 地址可能会 AWS 账户 导致您产生费用。有关更多信息，请参阅[Amazon EC2 定价](#)。

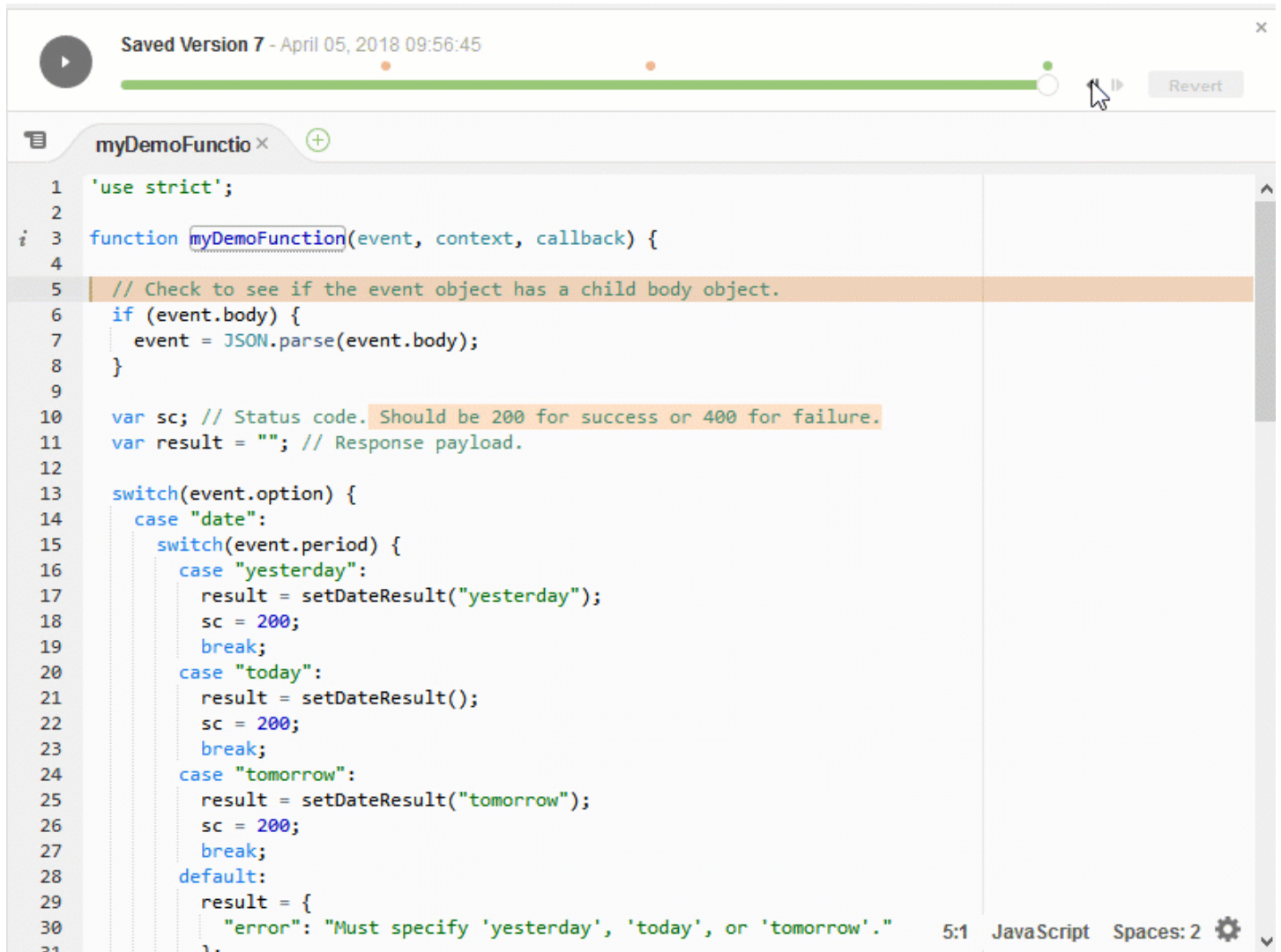
您无需使用 HTTP 通过端口 8080、8081 或 8082 运行应用程序。不过，如果您不这样做，将无法从 IDE 中预览运行的应用程序。有关更多信息，请参阅[预览运行的应用程序](#)。

例如，假设来自 VPN 的请求会阻止通过请求的协议或端口的流量。然后，那些访问您的应用程序 URL 的请求可能会失败。必须从允许通过请求的协议和端口传输流量的不同网络发出请求。有关更多信息，请与您的网络管理员联系。

建议不要与其他人共享 IDE 的应用程序预览标签页中的 URL。(此网址采用以下格式:https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/. 在这种格式中，12a34567b8cd9012345ef67abcd890e1是 AWS Cloud9 分配给环境的 ID。us-east-2是环境 AWS 区域的 ID。) 仅当环境的 IDE 处于打开状态并且应用程序正在同一个 Web 浏览器中运行时，此 URL 才有用。

在 AWS Cloud9 集成开发环境 (IDE) 中使用文件修订

您可以在 AWS Cloud9 IDE 中使用 File Revision History (文件修订历史记录) 查看和管理在 AWS Cloud9 EC2 开发环境中对文件的更改。File Revision History (文件修订历史记录) 窗格对于 AWS Cloud9 SSH 开发环境中的文件不可用。



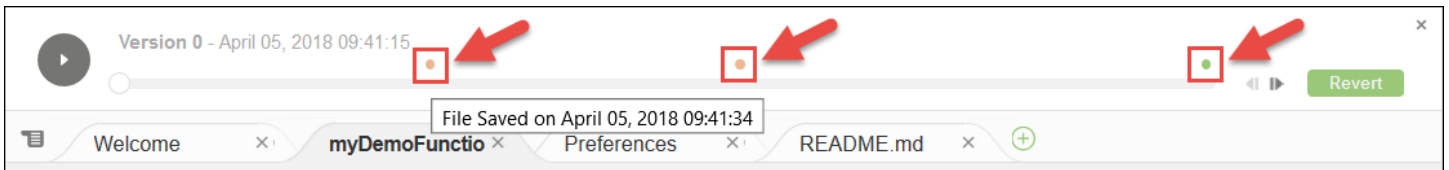
```
1 'use strict';
2
3 function myDemoFunction(event, context, callback) {
4
5 // Check to see if the event object has a child body object.
6 if (event.body) {
7     event = JSON.parse(event.body);
8 }
9
10 var sc; // Status code. Should be 200 for success or 400 for failure.
11 var result = ""; // Response payload.
12
13 switch(event.option) {
14     case "date":
15         switch(event.period) {
16             case "yesterday":
17                 result = setDateResult("yesterday");
18                 sc = 200;
19                 break;
20             case "today":
21                 result = setDateResult();
22                 sc = 200;
23                 break;
24             case "tomorrow":
25                 result = setDateResult("tomorrow");
26                 sc = 200;
27                 break;
28             default:
29                 result = {
30                     "error": "Must specify 'yesterday', 'today', or 'tomorrow'."
31                 }
32         }
33     }
34 }
```

要显示文件的 File Revision History (文件修订历史记录) 窗格，请在编辑器中打开文件。然后，在菜单栏上，选择 File, Show File Revision History (文件、显示文件修订历史记录)。

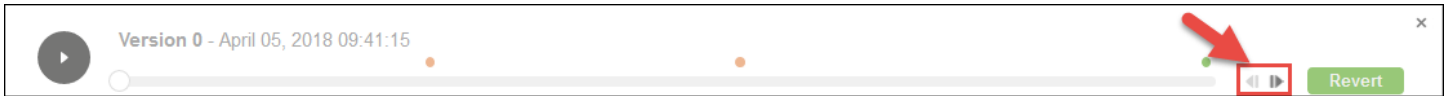
当您在环境中首次打开文件时，并且仅对于该环境，File Revision History (文件修订历史记录) 窗格将在 IDE 中开始跟踪文件的修订历史记录。File Revision History (文件修订历史记录) 窗格仅从编辑器本身跟进文件的修订。它不跟踪以任何其他方式对文件进行的修订 (例如，通过终端、Git 或其他文件修订工具)。

在显示 File Revision History (文件修订历史记录) 窗格时，您不能编辑文件。要隐藏窗格，请再次选择 File, Show Revision History (文件、显示修订历史记录)，或者选择窗格角上的 X (Close timeslider (关闭时间滑块))。

要跳转到与文件保存操作关联的文件的版本，请选择修订滑块上方的 File Saved on (文件保存于) 点。



要在修订滑块上，从当前选定的文件版本前进或后退一个版本，请选择一个步进箭头（ Step revision forward（向前步进一个版本）或 Step revision backward（向后步进一个版本））。



要从修订历史记录的开始一次自动前进一个版本到结尾，请选择播放按钮（ Playback file history（播放文件历史记录））。

要使当前选定的文件版本成为修订历史记录中的最新版本，请选择 Revert（还原）。

在 AWS Cloud9 集成开发环境 (IDE) 中使用图像文件

您可以使用 AWS Cloud9 IDE 查看和编辑图像文件。

- [查看或编辑图像](#)
- [调整图像大小](#)
- [裁剪图像](#)
- [旋转图像](#)
- [翻转图像](#)
- [缩放图像](#)
- [平滑图像](#)

查看或编辑图像

在 AWS Cloud9 IDE 中，打开您要查看或编辑的图像文件。支持的图像文件类型包括：

- .bmp
- .gif（仅查看）
- .ico（仅查看）
- .jpeg

- .jpg
- .png
- .tiff

调整图像大小

1. 在 IDE 中打开图像。
2. 在图像编辑栏中，选择 **Resize** (调整大小)。
3. 要更改图像宽度，请以像素为单位键入新的 **Width** (宽度)。或者选择 **Width** (宽度) 旁的 - 或 + 来更改当前宽度，一次一个像素。
4. 要更改图像高度，请以像素为单位键入新的 **Height** (高度)。或者选择 **Height** (高度) 旁的 - 或 + 来更改当前高度，一次一个像素。
5. 要保持图像的纵横比，请保留选中 **Maintain Aspect Ratio** (保持纵横比)。
6. 要确认图像的新大小，请在图像的编辑栏上查看宽度 (W) 和高度 (H)，以像素为单位。
7. 选择 **Resize** (调整大小)。
8. 要放弃调整大小，请在菜单栏上，依次选择 **Edit** (编辑)、**Undo** (撤消)。要保留新大小，请依次选择 **File** (文件)、**Save** (保存)。

裁剪图像

1. 在 IDE 中打开图像。
2. 将指针滚动到您要保留的图像部分。
3. 要确认选定内容的尺寸，请在图像编辑栏上查看 **Selection** (选定内容) 尺寸，如下所示：
 - 距离以像素为单位，从图像的左边缘到选定内容的左边缘 (L)
 - 距离以像素为单位，从图像的上边缘到选定内容的上边缘 (T)
 - 选定内容的宽度以像素为单位 (W)
 - 选定内容的高度以像素为单位 (H)
4. 在图像编辑栏中，选择 **Crop** (裁剪)。
5. 要放弃裁剪，请在菜单栏上依次选择 **Edit** (编辑)、**Undo** (撤消)。要保留新的裁剪图像，请依次选择 **File** (文件)、**Save** (保存)。

转动图像

1. 在 IDE 中打开图像。
2. 要逆时针转动图像，请在图像编辑栏上，选择 Rotate 90 Degrees Left (向左转动 90 度)。
3. 要顺时针转动图像，请在图像编辑栏上，选择 Rotate 90 Degrees Right (向右转动 90 度)。
4. 要放弃旋转，请在菜单栏上，依次选择 Edit (编辑)、Undo (撤消)。要保留新转动的图像，请依次选择 File (文件)、Save (保存)。

翻转图像

1. 在 IDE 中打开图像。
2. 要水平翻转图像，请在图像编辑栏上，选择 FlipH。
3. 要垂直翻转图像，请在图像编辑栏上，选择 FlipV。
4. 要放弃翻转，请在菜单栏上，依次选择 Edit (编辑)、Undo (撤消)。要保留新翻转的图像，请依次选择 File (文件)、Save (保存)。

缩放图像

1. 在 IDE 中打开图像。
2. 在图像编辑栏中，选择可用的缩放因素之一 (例如，75%、100% 或 200%)。

平滑图像

1. 在 IDE 中打开图像。
2. 在图像编辑栏中，选择 Smooth (平滑) 以减少图像的像素化。要放弃平滑，请取消选择 Smooth (平滑)。
3. 在菜单栏上，依次选择 File (文件)、Save (保存)。

在 AWS Cloud9 集成开发环境 (IDE) 中使用生成器、运行程序和调试程序

生成器 指示 AWS Cloud9 IDE 如何生成项目文件。运行程序 指示 AWS Cloud9 IDE 如何运行特定类型的文件。运行程序可使用调试程序来帮助找到文件源代码中的任何问题。

您可以通过下列方法，使用 AWS Cloud9 IDE 生成、运行和调试您的代码：

- 使用生成器来生成您的项目文件。请参阅[生成项目文件](#)。
- 使用运行程序可运行（也可以调试）您的代码。请参阅[内置的生成、运行和调试支持](#)以及[运行代码](#)。
- 更改一个内置运行程序的初始定义，以便通过不同方法运行（也可以调试）您的代码。请参阅[更改内置的运行程序](#)。
- 使用具有自定义的文件名、命令行选项，调试模式、当前工作目录和环境变量组合的运行程序来运行（也可以调试）您的代码。请参阅[创建运行配置](#)。
- 创建您自己的生成器或运行程序。请参阅[创建生成器或运行程序](#)。

内置的生成、运行和调试支持

AWS Cloud9 IDE 提供内置的、支持多种语言的代码生成、运行和调试支持。有关完整列表，请参阅[语言支持](#)。

内置的生成支持通过菜单栏上的 Run（运行）、Build System（生成系统）以及 Run（运行）、Build（生成）菜单命令提供。要添加对未列出的编程语言或工具的支持，请参阅[创建生成器或运行程序](#)。

内置的运行支持通过 Run（运行）按钮，菜单栏上的 Run（运行）、Run With（运行方式）以及 Run（运行）、Run Configurations（运行配置）菜单命令提供。要添加对未列出的编程语言或工具的支持，请参阅[创建生成器或运行程序](#)和[创建运行配置](#)。

内置的调试支持通过 Debugger（调试程序）窗口提供。要显示 Debugger（调试程序）窗口，请选择 Debugger（调试程序）按钮。如果 Debugger（调试程序）按钮不可见，请在菜单栏上依次选择 Window（窗口）、Debugger（调试程序）。

生成您的项目文件

1. 打开与要生成的代码对应的文件。
2. 在菜单栏上，依次选择 Run（运行）、Build System（生成系统），然后选择要使用的生成器的名称（如果尚未选择）。如果未列出您要使用的生成器，请停止本过程，去完成[创建生成器或运行程序](#)中的步骤，然后再返回本过程。
3. 依次选择 Run（运行）、Build（生成）。

运行代码

1. 如果尚未选择并打开与要运行的代码对应的文件，将该文件打开。
2. 在菜单栏上，选择下列项之一：
 - 要使用最匹配的内置运行程序来运行代码，请依次选择 Run (运行)、Run (运行)。如果 AWS Cloud9 找不到运行程序，则该命令会被禁用。
 - 要使用 AWS Cloud9 上次使用的运行配置运行代码，请依次选择 Run (运行)、(Run Last) 上次运行)。
 - 要使用特定的运行程序运行代码，请依次选择 Run (运行)、Run With (运行方式)，然后选择运行程序的名称。如果未列出您要使用的运行程序，请停止本过程，去完成[创建生成器或运行程序](#)中的步骤，然后再返回到本过程。
 - 要使用具有文件名、命令行选项、调试模式、当前工作目录和环境变量的自定义组合的特定运行程序来运行代码，请依次选择 Run (运行)、Run Configurations (运行配置)，然后选择运行配置的名称。在显示的运行配置选项卡中，选择 Runner: Auto (运行程序：自动)，再选择要使用的运行程序，然后选择 Run (运行)。如果未列出您要使用的运行程序，请停止本过程，去完成[创建生成器或运行程序](#)中的步骤，然后再返回到本过程。

调试代码

1. 在代码对应的运行配置选项卡上，选择 Run in Debug Mode (在调试模式下运行)。白色背景上的错误图标会变成绿色。有关更多信息，请参阅[运行代码](#)和[创建运行配置](#)。
2. 在代码中设置在运行期间暂停的任何断点，如下所示：
 - a. 打开要在其中设置断点的每个文件。
 - b. 在文件中要设置断点的每个点，选择其行号左侧间距中的空白区域。会显示红色圆圈。

要删除断点，请选择间距中的现有断点。

要禁用断点而不是删除它，请在 Debugger (调试程序) 窗口的 Breakpoints (断点) 中，清除要禁用的断点对应的框。要重新启用断点，请选中清除的框。

要一次性禁用所有断点，请在 Debugger (调试程序) 窗口中选择 Deactivate All Breakpoints (停用所有断点)。要重新启用所有断点，请选择 Activate All Breakpoints (激活所有断点)。

如果 Debugger (调试程序) 窗口不可见，则选择 Debugger (调试程序) 按钮。如果 Debugger (调试程序) 按钮不可见，则在菜单栏上选择 Window (窗口)、Debugger (调试程序)。

3. 在暂停运行的点设置要为其获取值的任何监视表达式，如下所示：
 - a. 在 Debugger (调试程序) 窗口中，在 Watch Expressions (监视表达式) 中选择 Type an expression here (在此处键入表达式)。
 - b. 键入要监视的表达式，然后按 Enter。

要更改现有的监视表达式，请右键单击对应监视表达式，然后选择 Edit Watch Expression (编辑监视表达式)。键入更改，然后按 Enter。

要删除现有的监视表达式，请右键单击对应监视表达式，然后选择 Remove Watch Expression (删除监视表达式)。

4. 按照[运行代码](#)中所述，运行您的代码。

每当运行暂停时，也可以将指针暂停在显示的任意一段代码 (例如，变量) 上，以便在工具提示中显示有关该端代码的所有可用信息。

更改内置运行程序

1. 在菜单栏上，依次选择 Run (运行)、Run With (运行方式)，然后选择您要更改的内置运行程序。
2. 在显示的运行配置选项卡上，选择 Stop (停止)，将试图运行代码的运行程序停止。
3. 选择 Runner: My Runner (运行程序：我的运行程序)，其中 My Runner (我的运行程序) 是要更改的运行程序的名称，然后选择 Edit Runner (编辑运行程序)。
4. 在显示的 My Runner.run 选项卡上，更改运行程序的当前定义。请参阅[定义生成器或运行程序](#)。
5. 依次选择 File (文件)、Save As (另存为)。使用相同名称 (My Runner.run) 将文件保存在 my-environment/.c9/runners 目录中，其中 my-environment 是您的 AWS Cloud9 开发环境的名称。

Note

您对内置运行程序所做的任何更改都仅应用于您在其中进行这些更改的环境。要将更改应用于单独的环境，请打开其他环境，然后按照以上步骤打开、编辑和保存对相关内置运行程序进行的那些相同更改。

创建运行配置

在菜单栏上依次选择 Run (运行)、Run Configurations (运行配置) 和 New Run Configuration (新建运行配置)。在显示的运行配置选项卡上，执行以下操作：

1. 在 Run (运行) 和 Restart (重启) 旁的框中，键入在 Run (运行)、Run Configurations (运行配置) 菜单上将为该运行配置显示的名称。
2. 在 Command (命令) 框中，键入要使用的任何自定义命令行选项。
3. 要让此运行配置使用运行程序的预定义调试设置，请选择 Run in Debug Mode (在调试模式下运行)。白色背景上的错误图标会变成绿色。
4. 要让此运行配置使用特定的工作目录，请选择 CWD 并选择要使用的目录，然后选择 Select (选择)。
5. 要让此运行配置使用特定的环境变量，请选择 ENV，然后键入每个环境变量的名称和值。

要使用该运行配置，请打开与要运行的代码对应的文件。在菜单栏上依次选择 Run (运行)、Run Configurations (运行配置)，然后选择此运行配置的名称。在显示的运行配置选项卡中，选择 Runner: Auto (运行程序：自动)，再选择要使用的运行程序，然后选择 Run (运行)。

Note

您创建的所有运行配置仅应用于您在其中创建运行配置的环境。要为单独的环境添加该运行配置，请打开其他环境，然后按照以上步骤在该环境中创建相同的运行配置。

创建生成器或运行程序

1. 要创建生成器，请在菜单栏上依次选择 Run (运行)、Build System (生成系统)、New Build System (新建生成系统)。要创建运行程序，请在菜单栏上依次选择 Run (运行)、Run With (运行方式)、New Runner (新建运行程序)。

2. 在显示的生成器选项卡（标记为 My Builder.build）或运行程序选项卡（标记为 My Runner.run）上，定义生成器或运行程序。请参阅[定义生成器或运行程序](#)。
3. 定义生成器或运行程序后，依次选择 File（文件）、Save As（另存为）。对于生成器，使用 .build 扩展名将文件保存在 my-environment/.c9/builders 目录中，其中 my-environment 是环境的名称。对于运行程序，使用 .run 文件扩展名将文件保存在 my-environment/.c9/runners 目录中，其中 my-environment 是环境的名称。您指定的文件名将是在 Run（运行）、Build System（生成系统）菜单（对于生成器）或者 Run（运行）、Run With（运行方式）菜单（对于运行程序）上显示的名称。因此，除非您指定不同的文件名，否则默认显示的名称将是 My Builder（我的生成器）（对于生成器）或 My Runner（我的运行程序）（对于运行程序）。

要使用此生成器或运行程序，请参阅[生成项目文件](#)或[运行代码](#)。

Note

您创建的任何生成器或运行程序仅适用于您在其中创建该生成器或运行程序的环境。要为单独的环境添加该运行生成器或运行程序，请打开其他环境，然后按照以上步骤在该环境中创建相同的生成器或运行程序。

定义生成器或运行程序

本过程假定您已经选择 Run（运行）、Build System（生成系统）、New Build System（新建生成系统）（对于生成器）或者 Run（运行）、Run With（运行方式）、New Runner（新建运行程序）（对于运行程序），以开始创建生成器或运行程序。

在显示的生成器或运行程序选项卡上，使用 JSON 定义运行程序或生成器。使用以下代码开头的模板。

对于生成器，使用以下代码开头。

```
{
  "cmd": [],
  "info": "",
  "env": {},
  "selector": ""
}
```

对于运行程序，使用以下代码开头。

```
{
  "cmd": [],
  "script": "",
  "working_dir": "",
  "info": "",
  "env": {},
  "selector": "",
  "debugger": "",
  "debugport": ""
}
```

在以上代码中：

- `cmd`：代表逗号分隔的字符串列表，AWS Cloud9 可将其作为单个命令运行。

当 AWS Cloud9 运行该命令时，列表中的各个字符串之间使用单个空格分隔。例如，AWS Cloud9 将 `"cmd": ["ls", "$file", "$args"]` 作为 `ls $file $args` 运行，此时 AWS Cloud9 会将 `$file` 替换为指向当前文件的完整路径，将 `$args` 替换为在文件名之后输入的任意参数。有关更多信息，请参阅本节后面所支持变量的列表。

- `script`：代表运行程序在终端执行的 `bash` 脚本（也可以根据需要，将其指定为行阵列以保证可读性）。
- `working_dir`：代表运行程序将从中运行的目录。
- `info`：代表在开始运行时，要向用户显示的任何文本字符串。此字符串可以包含变量，例如 `Running $project_path$file_name...`；此时 AWS Cloud9 会将 `$project_path` 替换为当前文件的目录路径，并将 `$file_name` 替换为当前文件的名称部分。请参阅本节后面所支持变量的列表。
- `env`：代表 AWS Cloud9 要使用的任何命令行参数数组，例如：

```
"env": {
  "LANG": "en_US.UTF-8",
  "SHLVL": "1"
}
```

- `selector`：代表您希望 AWS Cloud9 用于标识文件名的任何正则表达式，这些文件名将应用于此运行程序。例如，您可以为 Python 文件指定 `source.py`。
- `debugger`：代表您希望 AWS Cloud9 使用的、与此运行程序兼容的任何可用调试程序的名称。例如，您可以为 V8 调试程序指定 `v8`。

- `debugport` : 代表您希望 AWS Cloud9 在调试过程中使用的端口号。例如，您可以指定 15454 作为要使用的端口号。

下表给出您可以使用的变量。

变量	描述
<code>\$file_path</code>	当前文件的目录，例如， <code>/home/ec2-user/environment</code> 或 <code>/home/ubuntu/environment</code> 。
<code>\$file</code>	当前文件的完整路径，例如， <code>/home/ec2-user/environment/hello.py</code> 或 <code>/home/ubuntu/environment/hello.py</code> 。
<code>\$args</code>	在文件名之后输入的任何参数，例如， <code>"5"</code> 或 <code>"9"</code> 。
<code>\$file_name</code>	当前文件的名称部分，例如， <code>hello.py</code> 。
<code>\$file_extension</code>	当前文件的扩展名，例如， <code>py</code> 。
<code>\$file_base_name</code>	当前文件的名称（不带扩展名），例如， <code>hello</code> 。
<code>\$packages</code>	程序包文件夹的完整路径。
<code>\$project</code>	当前项目文件夹的完整路径。
<code>\$project_path</code>	当前项目文件的目录，例如， <code>/home/ec2-user/environment/</code> 或 <code>/home/ubuntu/environment/</code> 。
<code>\$project_name</code>	当前项目文件的名称（不带扩展名），例如， <code>my-demo-environment</code> 。
<code>\$project_extension</code>	当前项目文件的扩展名。

变量	描述
<code>\$project_base_name</code>	当前项目文件的名称（不带扩展名）。
<code>\$hostname</code>	环境的主机名，例如， <code>192.0.2.0</code> 。
<code>\$hostname_path</code>	环境的主机名，包含指向项目文件的相对路径，例如， <code>https://192.0.2.0/hello.js</code> 。
<code>\$url</code>	用于访问环境的完整 URL，例如， <code>https://192.0.2.0</code> 。
<code>\$port</code>	分配给环境的端口，例如， <code>8080</code> 。
<code>\$ip</code>	为环境运行进程的 IP 地址，例如， <code>0.0.0.0</code> 。

举例来说，名为 `G++.build` 的以下生成器文件为 GCC 定义一个生成器，它运行 `g++` 命令以及 `-o` 选项以将当前文件（例如 `hello.cpp`）编译到对象模块中。然后，它将对对象模块链接到与当前文件使用相同名称（例如，`hello`）的程序中。以下命令具有等效作用：`g++ -o hello hello.cpp`。

```
{
  "cmd": [ "g++", "-o", "$file_base_name", "$file_name" ],
  "info": "Compiling $file_name and linking to $file_base_name...",
  "selector": "source.cpp"
}
```

再举一个例子，名为 `Python.run` 的以下运行程序文件定义一个运行程序，它使用 Python 以及提供的所有参数运行当前文件。例如，如果当前文件名为 `hello.py` 且提供了 5 和 9 参数，则等效命令是 `python hello.py 5 9`。

```
{
  "cmd": [ "python", "$file_name", "$args" ],
  "info": "Running $file_name...",
  "selector": "source.py"
}
```

最后，名为 `Print Run Variables.run` 的以下运行程序定义一个运行程序，它只是输出每个可用变量的值，然后就停止。


```
{
  "info": "file_path = $file_path, file = $file, args = $args, file_name = $file_name,
file_extension = $file_extension, file_base_name = $file_base_name, packages
= $packages, project = $project, project_path = $project_path, project_name
= $project_name, project_extension = $project_extension, project_base_name =
$project_base_name, hostname = $hostname, hostname_path = $hostname_path, url = $url,
port = $port, ip = $ip"
}
```

在 AWS Cloud9 集成开发环境 (IDE) 中使用自定义环境变量

AWS Cloud9 IDE 支持设置自定义环境变量。您可以通过以下方法在 AWS Cloud9 IDE 中设置自定义环境变量。

- [设置命令级别的自定义环境变量](#)
- [在 ~/.bash_profile 中设置自定义的用户环境变量](#)
- [设置本地自定义环境变量](#)
- [在 ~/.bashrc 中设置自定义的用户环境变量](#)
- [在 ENV List 中设置自定义环境变量](#)

设置命令级别的自定义环境变量

可以在 AWS Cloud9 开发环境中运行命令的过程中设置命令级别的自定义环境变量。要测试该行为，请使用以下代码创建一个名为 `script.sh` 的文件：

```
#!/bin/bash

echo $MY_ENV_VAR
```

如果运行以下命令，终端会显示 Terminal session：

```
MY_ENV_VAR='Terminal session' sh ./script.sh
```

如果您使用本主题中所述的多种方法设置自定义环境变量，当您尝试获取自定义环境变量的值时，该设置的优先级高于所有其他设置。

在 ~/.bash_profile 中设置自定义的用户环境变量

您可以在环境中的 ~/.bash_profile 文件中设置自定义的用户环境变量。要测试该行为，请在环境中将以下代码添加到 ~/.bash_profile 文件中：

```
export MY_ENV_VAR='.bash_profile file'
```

如果之后从命令行运行 `sh ./script.sh`，则终端会显示 `.bash_profile file`。（前提是您已按上文所述创建了 `script.sh` 文件。）

设置本地自定义环境变量

您可以在终端会话中运行 **export** 命令，设置本地自定义环境变量。要测试该行为，请在终端会话中运行以下命令：

```
export MY_ENV_VAR='Command line export'
```

如果之后从命令行运行 `sh ./script.sh`，则终端会显示 `Command line export`。（前提是您已按上文所述创建了 `script.sh` 文件。）

如果您使用 **export** 命令以及在 ~/.bash_profile 文件中设置了相同的自定义环境变量，则当您尝试获取自定义环境变量的值时，**export** 命令设置的优先级更高。

在 ~/.bashrc 中设置自定义的用户环境变量

您可以在环境中的 ~/.bashrc 文件中设置自定义的用户环境变量。要测试该行为，请在环境中将以下代码添加到 ~/.bashrc 文件中：

```
export MY_ENV_VAR='.bashrc file'
```

如果之后从命令行运行 `sh ./script.sh`，则终端会显示 `.bashrc file`。（前提是您已按上文所述创建了 `script.sh` 文件。）

如果您使用 **export** 命令以及在 ~/.bashrc 文件中设置了相同的自定义环境变量，则当您尝试获取自定义环境变量的值时，**export** 命令设置的优先级更高。

在 ENV List 中设置自定义环境变量

您可以在 Run（运行）选项卡上的 ENV 列表中设置自定义环境变量。

要测试该行为，请执行以下操作：

1. 在菜单栏上依次选择 Run (运行) > Run Configurations (运行配置) > New Run Configuration (新建运行配置)。
2. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序：自动)，然后选择 Shell script (Shell 脚本)。
3. 选择 ENV，然后键入 MY_ENV_VAR 作为 Name (名称)，键入 ENV list 作为 Value (值)。
4. 在 Command (命令) 中，键入 ./script.sh。
5. 选择 Run (运行) 按钮，运行程序选项卡会显示 ENV list。(前提是您已按上文所述创建了 script.sh 文件。)

如果您在 ~/.bash_profile 文件中、使用 **export** 命令以及在 ~/.bashrc 文件和 ENV 列表中都设置了相同的自定义环境变量，则当您尝试获取自定义环境变量的值时，~/.bash_profile 文件设置的优先级最高，然后依次是 **export** 命令设置、~/.bashrc 文件设置和 ENV 列表设置。

Note

ENV 列表是使用独立于 Shell 脚本的代码获取和设置自定义环境变量的唯一方法。

在 AWS Cloud9 集成式开发环境 (IDE) 中使用项目设置

项目设置 (仅适用于当前 AWS Cloud9 开发环境) 包括以下类型的设置：

- 代码编辑器设置，例如，是否使用软制表符以及新文件行结尾
- 要忽略的文件类型
- 要显示或隐藏的提示和警告类型
- 编程语言 (例如 JavaScript、PHP、Python 和 Go) 的代码和格式设置
- 在运行和生成代码时使用的配置类型

虽然项目设置仅适用于单个环境，但您可以将某个环境的项目设置应用于其他环境。

- [查看或更改项目设置](#)
- [将一个环境的当前项目设置应用于另一个环境](#)
- [您可以执行的项目设置更改](#)

查看或更改项目设置

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 要查看当前环境的项目设置，请在 Preferences (首选项) 选项卡上的侧面导航窗格中选择 Project Settings (项目设置)。
3. 要更改环境的当前项目设置，请在 Project Settings (项目设置) 窗格中更改所需的设置。

请参阅[您可以进行的项目设置更改](#)。

将一个环境的当前项目设置应用于另一个环境

1. 在源和目标环境中，在 AWS Cloud9 IDE 的菜单栏上选择 AWS Cloud9, Open Your Project Settings (AC9，打开您的项目设置)。
2. 在源环境中，复制显示的 project.settings 选项卡的内容。
3. 在目标环境中，使用从源环境复制的内容覆盖 project.settings 选项卡的内容。
4. 在目标环境中，保存 project.settings 选项卡。

您可以更改的项目设置

以下几节介绍了可以在 Preferences (首选项) 选项卡的 Project Settings (项目设置) 窗格中更改的项目设置类型。

- [EC2 实例](#)
- [代码编辑器 \(Ace \)](#)
- [在文件中查找](#)
- [提示和警告](#)
- [JavaScript 支持](#)
- [构建](#)
- [运行和调试](#)
- [运行配置](#)
- [代码格式设置程序](#)
- [TypeScript 支持](#)
- [PHP 支持](#)
- [Python 支持](#)

- [Go](#) 支持

EC2 实例

Stop my environment (停止我的环境)

在关闭连接到该环境的 IDE 的所有 Web 浏览器实例后，选择何时自动停止环境的 Amazon EC2 实例（如果使用）。您可以选择从一周到 30 分钟的时间段范围。您还可以选择退出 AWS Cloud9 IDE 后永不自动停止 Amazon EC2 实例。

如果您希望在完成 IDE 后的 30 分钟内停止实例，则可以[使用控制台界面手动将其停止](#)。

代码编辑器 (Ace)

Soft Tabs (软制表符)

如果选定，每次按 Tab 时，将插入指定数量的空格而不是制表符。

Autodetect tab size on load (加载时自动检测制表符大小)

如果选定，AWS Cloud9 将尝试猜测制表符大小。

New file line endings (新文件行结尾)

用于新文件的行结尾类型。

有效选项包含以下内容：

- Windows (CRLF) 在行结尾包含回车符和换行符。
- Unix (LF) 在行结尾仅包含换行符。

On save, strip whitespace (保存时删除空格)

如果选定，每次保存文件时，AWS Cloud9 尝试从该文件中删除它认为是多余的空格和制表符。

在文件中查找

Ignore these Files (忽略这些文件)

在文件中查找时，AWS Cloud9 会忽略的文件类型。

Maximum number of files to search (in 1000) (要搜索的最大文件数 (以 1000 为单位))

在文件中查找时，AWS Cloud9 在当前范围内查找的最大文件数（1000 的倍数）。

提示和警告

警告级别

要启用的最低消息级别。

有效值包括：

- Info (信息) 启用信息性、警告和错误消息。
- Warning (警告) 仅启用警告和错误消息。
- Error (错误) 仅启用错误消息。

Mark Missing Optional Semicolons (标记缺少的可选分号)

如果启用，每次注意到可以在代码中使用分号但未使用时，AWS Cloud9 将在文件中进行标记。

Mark Undeclared Variables (标记未声明的变量)

如果启用，每次在代码中注意到未声明的变量时，AWS Cloud9 将在文件中进行标记。

Mark Unused Function Arguments (标记未使用的函数参数)

如果启用，每次在函数中注意到未使用的参数时，AWS Cloud9 将在文件中进行标记。

Ignore Messages Matching Regex (忽略消息匹配正则表达式)

AWS Cloud9 不显示与指定正则表达式匹配的任何消息。有关更多信息，请参阅 Mozilla Developer Network 上的 JavaScript 正则表达式主题中的[编写正则表达式模式](#)。

JavaScript 支持

使用 .eslintrc 自定义 JavaScript 警告

如果启用，AWS Cloud9 使用 .eslintrc 文件确定要启用或禁用哪些 JavaScript 警告。有关更多信息，请参阅 ESLint 网站上的[配置文件格式](#)。

JavaScript 库代码完成

AWS Cloud9 在尝试建议或执行自动代码完成时使用的 JavaScript 库。

Format Code on Save (保存时设置代码格式)

如果启用，每次保存 JavaScript 文件时，AWS Cloud9 尝试设置该文件中的代码格式。

将内置 JSBeautify 作为代码格式设置程序

如果启用，AWS Cloud9 将使用其内部 JSBeautify 实现尝试提高文件中的代码可读性。

Custom code formatter (自定义代码格式设置程序)

在设置 JavaScript 文件中的代码格式时尝试运行的 AWS Cloud9 命令。

构建

Builder path in environment (环境中的生成器路径)

任何自定义生成配置的路径。

运行和调试

Runner path in environment (环境中的运行程序路径)

任何自定义运行配置的路径。

Preview URL (预览 URL)

用于预览环境的应用程序的 URL。

运行配置

该环境的自定义运行配置。

Remove selected configs (删除选定的配置)

删除选定的运行配置。

Add new config (添加新配置)

创建新的运行配置。

Set as default (设置为原定设置)

将选定的运行配置设置为默认运行配置。

代码格式设置程序

JSBeautify 设置

用于提高文件中的代码可读性的设置。

Format Code on Save (保存时设置代码格式)

如果启用，每次保存代码文件时，AWS Cloud9 将尝试应用 JSBeautify 设置。

将 JSBeautify 用于 JavaScript

如果启用，每次保存 JavaScript 文件时，AWS Cloud9 将尝试应用 JSBeautify 设置。

Preserve empty lines (保留空行)

如果启用，AWS Cloud9 不会删除代码文件中的空行。

Keep array indentation (保留数组缩进)

如果启用，AWS Cloud9 将在代码文件中保留数组中的元素声明的缩进。

JSLint strict whitespace (JSLint 严格空格)

如果启用，AWS Cloud9 将尝试在代码文件中应用 JSLint 空格规则。有关更多信息，请参阅 [JSLint 帮助](#) 中的“空格”。

Braces (大括号)

指定代码中的大括号的对齐方式。

有效值包括：

- Braces with control statement (将大括号与控制语句对齐) - 根据需要，移动开头和结尾大括号以与相关的控制语句对齐。

例如，此代码的格式如下：

```
for (var i = 0; i < 10; i++) { if (i == 5) { console.log("Halfway done.") }}
```

在保存文件时变为以下代码：

```
for (var i = 0; i < 10; i++) {  
  if (i == 5) {  
    console.log("Halfway done.")  
  }  
}
```

- Braces on own line (大括号单独成行) - 根据需要，将每个大括号单独成行。

例如，此代码的格式如下：


```
for (var i = 0; i < 10; i++) { if (i == 5) { console.log("Halfway done.") }}
```

在保存文件时变为以下代码：

```
for (var i = 0; i < 10; i++) {if (i == 5)
{
    console.log("Halfway done.")
}
}
```

- End braces on own line (结尾大括号单独成行) - 根据需要，将每个结尾大括号单独成行。

例如，此代码的格式如下：

```
for (var i = 0; i < 10; i++) {
    if (i == 5) { console.log("Halfway done.") }
}
```

在保存文件时变为以下代码：

```
for (var i = 0; i < 10; i++) {
    if (i == 5) {
        console.log("Halfway done.")
    }
}
```

Preserve inline blocks (保留内联块)

如果启用，则当内联块的起始和终止括号在同一行中时，AWS Cloud9 不会尝试将其移动到单独的行中。

Space before conditionals (在条件前面添加空格)

如果启用，AWS Cloud9 将根据需要在每个条件声明前面添加空格。

Unescape strings (取消转义字符串)

如果启用，AWS Cloud9 将转义的字符串转换为未转义的等效字符串。例如，将 `\n` 转换为换行符，并将 `\r` 转换为回车符。

Indent inner HTML (缩进内部 HTML)

如果启用，AWS Cloud9 将在 HTML 代码中缩进 `<head>` 和 `<body>` 部分。

TypeScript 支持

Format Code on Save (保存时设置代码格式)

如果启用，每次保存 TypeScript 文件时，AWS Cloud9 将尝试设置 TypeScript 代码格式。

Custom code formatter (自定义代码格式设置程序)

TypeScript 代码的任何自定义代码格式配置的路径。

PHP 支持

启用 PHP 代码完成

如果启用，AWS Cloud9 将尝试完成 PHP 代码。

PHP 完成包含路径

AWS Cloud9 在尝试帮助完成 PHP 代码时使用的位置。例如，如果您具有希望 AWS Cloud9 在完成代码时使用的自定义 PHP 文件，并且这些文件位于 `~/environment` 目录中的某个位置，请将 `~/environment` 添加到该路径中。

Format Code on Save (保存时设置代码格式)

如果启用，每次保存 PHP 文件时，AWS Cloud9 将尝试设置 PHP 代码格式。

Custom code formatter (自定义代码格式设置程序)

PHP 代码的任何自定义代码格式配置的路径。

Python 支持

启用 Python 代码完成

如果启用，AWS Cloud9 将尝试完成 Python 代码。要设置 AWS Cloud9 在完成 Python 代码时使用的路径，请使用 `PYTHONPATH` 设置。

Python 版本

指定要使用的 Python 版本。

Pylint 命令行选项

AWS Cloud9 要在 Python 代码中用于 Pylint 的选项。有关更多信息，请参阅 Pylint 网站上的 [Pylint 用户手册](#)。

PYTHONPATH

供 AWS Cloud9 使用的 Python 库和程序包的路径。例如，如果在 `~/environment` 目录中具有自定义 Python 库和程序包，请将 `~/environment` 添加到该路径中。

Format Code on Save (保存时设置代码格式)

如果启用，每次保存 Python 文件时，AWS Cloud9 将尝试设置 Python 代码格式。

Custom code formatter (自定义代码格式设置程序)

Python 代码的任何自定义代码格式配置的路径。

Go 支持

启用 Go 代码完成

如果启用，AWS Cloud9 将尝试完成 Go 代码。

Format Code on Save (保存时设置代码格式)

如果启用，每次保存 Go 文件时，AWS Cloud9 将尝试设置 Go 代码格式。

Custom code formatter (自定义代码格式设置程序)

Go 代码的任何自定义代码格式配置的路径。

手动停止环境的 EC2 实例

[EC2 实例](#) 设置允许您在关闭所有连接到 IDE 的 Web 浏览器实例后 30 分钟内自动停止环境的 Amazon EC2 实例。

您也可以使用控制台立即手动停止实例。

手动停止环境的 EC2 实例

1. 在关闭连接到 IDE 的所有 Web 浏览器实例后，在 AWS Cloud9 控制台中选择 **Your environments** (您的环境)。
2. 选择窗格右上角显示所使用环境详细信息的按钮，然后选择 **View details** (查看详细信息)。
3. 在 **Environment details** (环境详细信息) 中，在 **EC2 Instance** (EC2 实例) 下，选择 **Go To Instance** (转到实例)。

4. 在 Amazon EC2 控制台中，在 Instance state (实例状态) 下，选中该复选框以选择环境的实例。Instance state (实例状态) 可能表明该实例仍在运行。
5. 依次选择 Instance state (实例状态)、Stop instance (停止实例)。
6. 当系统提示您确认时，选择 Stop。停止实例可能需要几分钟时间。

使用 AWS Cloud9 IDE 中的用户设置

用户设置是在与 AWS Identity and Access Management (IAM 用户) 关联的每个 AWS Cloud9 开发环境中应用的设置。它们包括以下设置：

- 一般用户界面设置，例如启用动画和标记已更改的标签页
- 文件系统导航设置
- 文件查找和搜索设置
- 终端会话和输出的颜色方案
- 其他代码编辑器设置，如字体大小、代码折叠、整行选择、滚动动画和字体大小

当您更改用户设置时，AWS Cloud9 将这些更改推送到云端，并将其与您的 IAM 用户相关联。AWS Cloud9 还会持续扫描云以查找与您的 IAM 用户关联的用户设置更改，并将这些设置应用于当前环境。您可以使用这种行为体验到相同的外观和感觉，无论您在什么 AWS Cloud9 环境中工作。

Note

为了存储和检索 IDE 设置，AWS Cloud9 会使用内部 API GetUserSettings 和 UpdateUserSettings。

您可以与其他用户共享您的用户设置，如下所示：

- [查看或更改用户设置](#)
- [与其他用户共享您的用户设置](#)
- [您可以执行的用户设置更改](#)

查看或更改用户设置

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。

2. 要查看每个环境的用户设置，请在 Preferences (首选项) 选项卡的侧导航窗格中选择 User Settings (用户设置)。
3. 在 User Settings (用户设置) 窗格中，更改每个环境中的用户设置。
4. 要对您的其他环境应用更改，只需打开相应环境。如果该环境已经打开，则刷新该环境的 Web 浏览器选项卡。

有关更多信息，请参阅[您可以执行的用户设置更改](#)。

与其他用户共享您的用户设置

1. 在源和目标环境中，在 AWS Cloud9 IDE 的菜单栏上选择 AWS Cloud9, Open Your User Settings (Amazon Cloud9，打开您的用户设置)。
2. 在源环境中，复制显示的 user.settings 选项卡的内容。
3. 在目标环境中，使用从源环境复制的内容覆盖 user.settings 选项卡的内容。
4. 在目标环境中，保存 user.settings 选项卡。

您可以执行的用户设置更改

以下各节说明了在 Preferences (首选项) 选项卡的 User Settings (用户设置) 窗格中，您可以更改的用户设置类型：

- [一般性问题](#)
- [用户界面](#)
- [协作](#)
- [Tree 和 Go 面板](#)
- [在文件中查找](#)
- [元数据](#)
- [观察程序](#)
- [终端](#)
- [输出](#)
- [代码编辑器 \(Ace \)](#)
- [输入](#)
- [提示和警告](#)

- [运行和调试](#)
- [预览](#)
- [构建](#)

常规

Reset to Factory Settings (重置为出厂设置)

如果选择 Reset to Default (重置为默认值) 按钮，AWS Cloud9 会将所有用户设置重置为 AWS Cloud9 的默认用户设置。如要确认，请选择 Reset settings (重置设置)。

Warning

您不能撤消此操作。

Warn Before Exiting (退出前警告)

每当您尝试关闭 IDE 时，AWS Cloud9 都会要求您确认要退出。

用户界面

Enable UI Animations (启用 UI 动画)

AWS Cloud9 在 IDE 中使用动画。

Use an Asterisk (*) to Mark Changed Tabs (使用星号 (*) 标记已更改选项卡)

AWS Cloud9 会为已更改但尚未保存其内容的标签页添加一个星号 (*)。

Display Title of Active Tab as Browser Title (显示活动标签页的标题作为浏览器标题)

AWS Cloud9 将关联 Web 浏览器标签页的标题更改为活动标签页的标题 [例如，Untitled1、hello.js、Terminal (终端)、Preferences (首选项)]。

Automatically Close Empty Panes (自动关闭空窗格)

每当您重新加载一个环境时，AWS Cloud9 都会自动关闭它认为是空窗格的所有窗格。

Environment Files Icon and Selection Style (环境文件图标和选择样式)

图标 AWS Cloud9 用于环境文件，以及 AWS Cloud9 使用的文件选择行为。

有效值包括：

- Default (默认值) – AWS Cloud9 使用默认图标和默认的文件选择行为。
- Alternative (替代值) – AWS Cloud9 使用替代图标和替代的文件选择行为。

协作

Disable collaboration security warning (禁用协作安全警告)

如果将读取/写入成员添加到环境，AWS Cloud9 不会显示安全警告对话框。

Show Authorship Info (显示作者信息)

AWS Cloud9 对其他环境成员输入的文本加下划线，并在间隔部分加上相关突出显示效果。

Tree 和 Go 面板

Scope Go to Anything to Favorites (将转到任何内容的范围限定为收藏夹)

Go (转到) 窗口中的 Go to File (转到文件) 显示结果仅作用于 Environment (环境) 窗口中的 Favorites (收藏夹)。

Enable Preview on Tree Selection (启用在树中选择时预览)

只需单击鼠标而不是双击鼠标，AWS Cloud9 就会显示选定文件。

Hidden File Pattern (隐藏文件模式)

希望 AWS Cloud9 将其视为隐藏文件的文件类型。

Reveal Active File in Project Tree (在项目树中显示活动文件)

AWS Cloud9 在 Environment (环境) 窗口中突出显示活动的文件。

Download Files As (下载文件方式)

AWS Cloud9 下载文件时使用的行为。

有效值包括：

- auto (自动) – AWS Cloud9 下载文件而不进行修改。
- tar.gz – AWS Cloud9 将文件下载为压缩的 TAR 文件。
- zip – AWS Cloud9 将文件下载为 .zip 文件。

在文件中查找

Search In This Path When 'Project' Is Selected (选择“项目”时在该路径中搜索)

在“Find in files (在文件中查找)”栏中，当选择 Project (项目) 作为搜索范围时，要在其中进行搜索的路径。

Show Full Path in Results (在结果中显示完整路径)

在 Search Results (搜索结果) 选项卡中显示指向每个匹配文件的完整路径。

Clear Results Before Each Search (每次搜索前清除结果)

在开始当前搜索之前，清除 Search Results (搜索结果) 选项卡中的之前所有搜索结果。

Scroll Down as Search Results Come In (显示搜索结果时向下滚动)

随着多条搜索结果得到确定，Search Results (搜索结果) 选项卡中的结果列表会滚动到底部。

Open Files when Navigating Results with (Up and Down) (导航结果时使用上下箭头打开文件)

在 Search Results (搜索结果) 选项卡的结果列表中按上下箭头键时，会打开每个匹配的文件。

元数据

Maximum of Undo Stack Items in Meta Data (元数据中的最大撤消堆栈项目数)

AWS Cloud9 在其可撤消操作的列表中可包含的最大项目数。

观察程序

Auto-Merge Files When a Conflict Occurs (发生冲突时自动合并文件)

每当发生合并冲突时，AWS Cloud9 都会尝试自动合并文件。

终端

Text Color (文本颜色)

Terminal (终端) 选项卡中文本的颜色。

Background Color (背景色)

Terminal (终端) 选项卡中的背景颜色。

Selection Color (选定内容颜色)

Terminal (终端) 选项卡中选定文本的颜色。

Font Family (字体系列)

Terminal (终端) 选项卡中文本的字体样式。

字体大小

Terminal (终端) 选项卡中文本的大小。

Antialiased Fonts (抗锯齿字体)

AWS Cloud9 尝试使 Terminal (终端) 选项卡中显示的文本更清楚。

Blinking Cursor (闪烁光标)

AWS Cloud9 在 Terminal (终端) 选项卡中持续闪烁光标。

Scrollback (回滚)

可以在 Terminal (终端) 选项卡中上下滚动的行数。

将 AWS Cloud9 用作默认编辑器

将 AWS Cloud9 用作默认文本编辑器。

输出

Text Color (文本颜色)

显示输出的选项卡中文本的颜色。

Background Color (背景色)

显示输出的选项卡中文本的背景色。

Selection Color (选定内容颜色)

显示输出的选项卡中选定文本的颜色。

Warn Before Closing Unnamed Configuration (关闭未命名的配置前发出警告)

AWS Cloud9 在关闭任何未保存的配置选项卡之前提示您进行保存。

Preserve log between runs (保留运行期间日志)

AWS Cloud9 保留每一次尝试运行时的日志。

代码编辑器 (Ace)

Auto-pair Brackets, Quotes, etc. (自动配对括号、引号等)

AWS Cloud9 尝试为在编辑器选项卡中键入的每个相关起始字符添加匹配的结束字符，如圆括号、引号和大括号。

Wrap Selection with Brackets, Quote, etc. (将选定内容放在括号、引号等内)

在编辑器选项卡中选中文本且键入相关起始字符后，AWS Cloud9 尝试在文本末尾插入匹配的结束字符，如圆括号、引号和大括号。

Code Folding (代码折叠)

AWS Cloud9 尝试根据相关代码语法规则，在编辑器选项卡中显示、展开、隐藏或折叠部分代码。

Fade Fold Widgets (淡入折叠小部件)

只要将鼠标暂停在编辑器选项卡中的代码折叠控件上，AWS Cloud9 就会在间隔部分显示这些控件。

复制空白所选内容

AWS Cloud9 允许您复制和/或剪切文本，此选项将决定是否可以将空白文本复制到剪贴板。

Full Line Selection (整行选择)

在编辑器选项卡中单击三次，AWS Cloud9 就会选择整行。

Highlight Active Line (突出显示活动行)

AWS Cloud9 在编辑器选项卡中突出显示整个活动行。

Highlight Gutter Line (突出显示行的间隔)

AWS Cloud9 在编辑器选项卡中当前行旁边的间隔部分突出显示位置。

Show Invisible Characters (显示不可见字符)

AWS Cloud9 在编辑器选项卡中显示它认为不可见的字符，例如，回车和换行符、空格和制表符。

Show Gutter (显示间隔)

AWS Cloud9 显示间隔部分。

Show Line Numbers (显示行号)

在间隔中显示行号的行为。

有效值包括：

- Normal (正常) – 显示行号。
- Relative (相对) – 显示相对于活动行的行号。
- None (无) – 隐藏行号。

Show Indent Guides (显示缩进对齐线)

AWS Cloud9 显示对齐线，从而更便于在编辑器选项卡中进行文本缩进。

Highlight Selected Word (突出显示选定单词)

AWS Cloud9 选择在编辑器选项卡中双击的整个单词。

Scroll Past the End of the Document (滚动到文档末尾以后)

允许用户在编辑器选项卡中滚动到当前文件末尾以后的行为。

有效值包括：

- Off (关闭) – 不允许滚动到当前文件末尾以后。
- Half Editor Height (编辑器半高) – 允许最多将当前文件末尾滚动到编辑器屏幕高度的一半。
- Full Editor Height (编辑器全高) – 允许最多将当前文件末尾滚动到编辑器屏幕的整个高度。

Animate Scrolling (动画方式滚动)

AWS Cloud9 在编辑器选项卡中执行滚动操作时应用动画行为。

Font Family (字体系列)

在编辑器选项卡中使用的字体样式。

字体大小

在编辑器选项卡中使用的字体大小。

Antialiased Fonts (抗锯齿字体)

AWS Cloud9 尝试使编辑器选项卡中显示的文本更清楚。

Show Print Margin (显示打印边距)

在编辑器选项卡中指定字符位置后显示一条垂直线。

Mouse Scroll Speed (鼠标滚动速度)

鼠标在编辑器选项卡中的相对滚动速度。值越大，滚动得越快。

Cursor Style (光标样式)

指针在编辑器选项卡中的样式和行为。

有效值包括：

- Ace (粗) – 将指针显示为相对于 Slim (细) 更宽的垂直条。
- Slim (细) – 将指针显示为相对较细的垂直条。
- Smooth (平滑) – 将指针显示为相对于 Slim (细) 更宽的垂直条，且与 Slim (细) 相比闪烁得更加连续。
- Smooth and Slim (平滑且细) – 将指针显示为相对较细的垂直条，且与 Slim (细) 相比闪烁得更加连续。
- Wide (宽) – 将指针显示为相对较宽的垂直条。

Merge Undo Deltas (合并撤消增量)

- Always (始终) – 允许恢复合并冲突。
- Never (从不) – 从不允许恢复合并冲突。
- Timed (定时) – 允许在特定的时间段后恢复合并冲突。

Enable Wrapping For New Documents (为新文档启用换行)

AWS Cloud9 在新文件中将代码换行。

输入

Complete As You Type (键入时完成)

AWS Cloud9 在您键入时尝试显示可能的最终文本。

Complete On Enter (按 Enter 后完成)

AWS Cloud9 在您按 Enter 后尝试显示可能的最终文本。

Highlight Variable Under Cursor (突出显示光标下的变量)

AWS Cloud9 在代码中突出显示引用的所有选定变量。

Use Cmd-Click for Jump to Definition (按 Cmd 键时单击以跳转到定义)

AWS Cloud9 转至在按住 Command (Mac) 或 Ctrl (Windows) 的同时选择的代码的原始定义。

提示和警告

Enable Hints and Warnings (启用提示和警告)

AWS Cloud9 显示适用的提示和警告消息。

点击后显示可用的快速修复

当您点击代码中的关键字时，AWS Cloud9 会显示一个包含重构建议的工具提示。

Ignore Messages Matching Regex (忽略消息匹配正则表达式)

AWS Cloud9 不显示与指定的正则表达式匹配的任何消息。有关更多信息，请参阅 Mozilla Developer Network 上的 [JavaScript 正则表达式](#) 主题中的编写正则表达式模式。

运行和调试

Save All Unsaved Tabs Before Running (运行之前保存所有未保存选项卡)

AWS Cloud9 在运行关联的代码之前，尝试保存所有未保存的、带有打开的选项卡的文件。

预览版

Preview Running Apps (预览运行应用程序)

每次按下 Preview (预览) 按钮时，AWS Cloud9 都会尝试在活动选项卡中显示代码输出的预览。

Default Previewer (默认预览格式)

AWS Cloud9 用于预览代码输出的格式。

有效值包括：

- Raw (原始) – 尝试使用纯文本格式显示代码输出。
- Browser (浏览器) – 尝试使用 Web 浏览器的首选格式显示代码输出。

When Saving Reload Previewer (保存时重新加载预览器)

每次保存代码文件时，AWS Cloud9 用于预览代码输出的行为。

有效值包括：

- Only on Ctrl-Enter (仅在按下 Ctrl-Enter 时) – 只要为当前代码选项卡按下 Ctrl+Enter，即尝试预览代码输出。

- Always (始终) – 尝试在保存代码文件时预览代码输出。

构建

Automatically Build Supported Files (自动生成支持的文件)

在启动构建操作且代码使用支持的格式的情况下，AWS Cloud9 尝试自动构建当前代码。

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS 项目和用户设置

AWS 服务 设置，位于首选项选项卡的 AWS 设置窗格中，包括以下设置：

- 为 AWS 资源窗口使用哪个 AWS 区域
- 是否使用 AWS 托管式临时凭证
- 以纯文本模式还是可视化模式显示 AWS Serverless Application Model (AWS SAM) 模板编辑器

要查看或更改这些设置，请在环境的 IDE 菜单栏中选择 AWS Cloud9, Preferences (Amazon Cloud9, 首选项)。

在以下列表中，项目级别设置仅适用于当前 AWS Cloud9 开发环境。相比之下，用户级别设置适用于与 IAM 用户关联的每个环境。有关更多信息，请参阅[将一个环境的当前项目设置应用于另一个环境](#)以及[与其他用户共享您的用户设置](#)。

- [项目级别设置](#)
- [用户级别设置](#)

项目级别设置

AWS 区域

为 AWS 资源窗口的 Lambda 部分使用哪个 AWS 区域。

AWS 托管式临时凭证

如果开启，则从环境中的 AWS CLI、AWS CloudShell 或 AWS SDK 代码调用 AWS 服务时，将使用 AWS 托管式临时凭证。有关更多信息，请参阅[AWS 托管式临时凭证](#)。

用户级别设置

使用 AWS SAM 可视化编辑器

如果开启，则使用 AWS 资源窗口的 Lambda 部分时，将以可视化模式显示 AWS Serverless Application Model (AWS SAM) 模板编辑器。如果关闭，则以文本模式显示编辑器。

在 AWS Cloud9 集成开发环境 (IDE) 中使用键绑定

键绑定 定义您的快捷键组合。键绑定适用于与您的 IAM 用户关联的每个 AWS Cloud9 开发环境。在对您的键绑定进行更改时，AWS Cloud9 将这些更改推送到云端，并将其与您的 IAM 用户相关联。另外，AWS Cloud9 会持续扫描云端以查找与您的 IAM 用户关联的键绑定更改，并将这些更改应用于当前环境。

您可以与其他用户共享您的键绑定。

- [查看或更改您的键绑定](#)
- [与其他用户共享您的键绑定](#)
- [更改您的键盘模式](#)
- [更改您的操作系统键绑定](#)
- [更改特定的键绑定](#)
- [删除您的所有自定义键绑定](#)

查看或更改键绑定

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 要查看每个环境中的键绑定，请在 Preferences (首选项) 选项卡的侧面导航窗格中选择 Keybindings (键绑定)。
3. 要更改每个环境中的键绑定，请在 Keybindings (键绑定) 窗格中更改所需的设置。
4. 要将更改应用于环境，请直接打开该环境。如果该环境已经打开，则刷新该环境的 Web 浏览器选项卡。

有关更多信息，请参阅下列内容：

- [MacOS 默认键绑定参考](#)

- [MacOS Vim 键绑定参考](#)
- [MacOS Emacs 键绑定参考](#)
- [MacOS Sublime 键绑定参考](#)
- [Windows/Linux 默认键绑定参考](#)
- [Windows/Linux Vim 键绑定参考](#)
- [Windows/Linux Emacs 键绑定参考](#)
- [Windows / Linux Sublime 键绑定参考](#)

与其他用户共享您的键绑定

1. 在源和目标环境中，在 AWS Cloud9 IDE 菜单栏上选择 AWS Cloud9, Open Your Keymap (Amazon Cloud9 > 打开键映射)。
2. 在源环境中，复制所显示的 keybindings.settings 选项卡的内容。
3. 在目标环境中，使用从源环境复制的内容覆盖 keybindings.settings 选项卡内容。
4. 在目标环境中，保存 keybindings.settings 选项卡。

更改您的键盘模式

在与您的 IAM 用户关联的每个环境中，您可以更改 AWS Cloud9 IDE 与编辑器中的文本交互时使用的键盘模式。

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，请选择以下键盘模式之一：
 - 选择 Default (默认) 以使用一组默认键绑定。
 - 选择 Vim 以使用 Vim 模式。有关更多信息，请参阅 [Vim 帮助文件](#) 网站。
 - 选择 Emacs 以使用 Emacs 模式。有关更多信息，请参阅 GNU 操作系统网站上的 [Emacs 编辑器](#)。
 - 选择 Sublime 以使用 Sublime 模式。有关更多信息，请参阅 [Sublime Text 文档](#) 网站。

更改您的操作系统键绑定

在与您的 IAM 用户关联的每个环境中，您可以更改 AWS Cloud9 IDE 可识别的一组操作系统键绑定。

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 Keybindings (键绑定)。
3. 对于 Operating System (操作系统)，请选择以下操作系统之一：
 - 选择 Auto (自动) 以使 AWS Cloud9 IDE 尝试检测要使用的一组操作系统键绑定。
 - 选择 MacOS 以使 AWS Cloud9 IDE 使用以 macOS 格式列出的键绑定。
 - 选择 Windows/Linux 以使 AWS Cloud9 IDE 使用以 Windows 和 Linux 格式列出的键绑定。

更改特定的键绑定

在与您的 IAM 用户关联的每个环境中，您可以更改各个键绑定。

同时更改一个键绑定

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 Keybindings (键绑定)。
3. 在键绑定列表中，在 Keystroke (按键) 列中打开 (双击) 要更改的键绑定。
4. 使用键盘指定替换组合键，然后按 Enter。

Note

要完全删除当前组合键，请按 Backspace (Windows 或 Linux) 或 Delete (macOS)。

同时更改多个键绑定

1. 在菜单栏上，依次选择 AWS Cloud9、Open Your Keymap (打开键映射)。
2. 在 `keybindings.settings` 文件中，定义要更改的每个键绑定。以下是示例语法。

```
[
  {
    "command": "addfavorite",
    "keys": {
      "win": ["Ctrl-Alt-F"],
      "mac": ["Ctrl-Option-F"]
    }
  },
]
```

```
{
  "command": "copyFilePath",
  "keys": {
    "win": ["Ctrl-Shift-F"],
    "mac": ["Alt-Shift-F"]
  }
}
```

在该示例中，addFavorite 和 copyFilePath 是 Preferences (首选项) 选项卡上 Keybindings (键绑定) 窗格的 Keystroke (按键) 列中的键绑定名称。对于 Windows 或 Linux 和 macOS，所需的键绑定分别为 win 和 mac。

要应用您的更改，请保存 keybindings.settings 文件。在短暂延迟后，将在 Keybindings (键绑定) 窗格中显示您的更改。

删除您的所有自定义键绑定

在与您的 IAM 用户关联的每个环境中，您可以删除所有自定义键绑定并将所有键绑定恢复为原定设置值。

Warning

不能撤消此操作。

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 Keybindings (键绑定)。
3. 选择 Reset to Defaults (重置为默认值)。

在 AWS Cloud9 集成开发环境 (IDE) 中使用主题

主题 定义您的整体 IDE 颜色。这适用于每个与您的 IAM 用户关联的 AWS Cloud9 开发环境。在对您的主题进行更改时，AWS Cloud9 将这些更改推送到云端，并将其与您的 IAM 用户相关联。AWS Cloud9 还会持续扫描云端以查找与您的 IAM 用户关联的主题更改。AWS Cloud9 将这些更改应用于当前环境。

- [查看或更改您的主题](#)

- [您可以更改的整体主题设置](#)
- [主题覆盖](#)

查看或更改您的主题

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 要查看每个环境中的主题，请在 Preferences (首选项) 选项卡上的侧面导航窗格中选择 Themes (主题)。
3. 要更改每个环境中的主题，请在 Themes (主题) 窗格中更改所需的设置。要使用代码更改主题的某些部分，请选择 `your stylesheet` (您的样式表) 链接。
4. 要将更改应用于您的环境，请打开该环境。如果该环境已经打开，则刷新该环境的 Web 浏览器选项卡。

您可以更改的整体主题设置

您可以在 Preferences (首选项) 选项卡上的 Themes (主题) 窗格中更改以下类型的整体主题设置。

Flat Theme (平面主题)

在 AWS Cloud9 IDE 中应用内置的平面主题。

Classic Theme (经典主题)

在 AWS Cloud9 IDE 中应用选定的内置经典主题。

Syntax Theme (语法主题)

在 AWS Cloud9 IDE 中将选定的主题应用于代码文件。

主题覆盖

Important

AWS Cloud9 不再支持允许用户通过更新 `styles.css` 文件覆盖 IDE 的功能。用户可以继续使用编辑器查看、编辑和保存 `styles.css` 文件。但是，加载 AWS Cloud9 IDE 时不会应用任何主题覆盖。

如果 AWS Cloud9 检测到 `styles.css` 文件已被修改，则 IDE 中将显示以下消息：
已停止支持主题覆盖。此 `styles.css` 文件的内容将不再应用于加载 AWS Cloud9 IDE。

如果您需要使用样式表来定义 IDE 的主题，请直接[联系我们](#)。

在 AWS Cloud9 集成开发环境 (IDE) 中管理初始化脚本

Important

AWS Cloud9 不再支持允许用户自定义初始化脚本的实验性功能。此脚本在 IDE 中自动运行。用户可以继续使用编辑器查看、编辑和保存 `init.js` 文件。但不再允许自定义初始化脚本运行，也无法修改 IDE 的行为。

如果 AWS Cloud9 检测到 `init.js` 文件已被修改，则 IDE 中将显示以下消息：

已停止支持初始化脚本。加载 AWS Cloud9 IDE 时将不再执行此 `init.js` 文件的内容。

如果您需要为 IDE 运行自定义初始化脚本，请[联系我们](#)。

初始化脚本 定义在加载所有插件后在 IDE 中运行的初始化代码。这适用于每个与您的 IAM 用户关联的 AWS Cloud9 开发环境。AWS Cloud9 还会持续扫描初始化脚本的更改，并在发生修改时提示用户。

打开初始化脚本

要打开初始化脚本，请在菜单栏上依次选择 AWS Cloud9、Open Your Init Script (打开初始化脚本)。

Important

您可以使用编辑器编辑和保存 `init.js` 文件，但不允许自定义脚本在 IDE 中运行。

用于 AWS Cloud9 集成开发环境 (IDE) 的 MacOS 默认键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 MacOS 操作系统的默认键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Default (默认)。

4. 对于 Operating System (操作系统) , 选择 MacOS。

另请参阅[使用键绑定](#)。

- [一般性问题](#)
- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Command-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Control-Space Option-Space	complete
代码完成, 然后覆盖	Control-Shift-Space Option-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Command-C	copy
将选定内容剪切到剪贴板中	Command-X	cut
展开代码 (如果适用)	Tab	expandSnippet
显示当前文档的查找和替换栏	Command-F	find
选择当前文档中的所有查找匹配项	Control-Option-G	findAll

描述	键绑定	命令
转至当前文档中下一处与您最后输入的查询匹配的内容	Command-G	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Command-Shift-G	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Command-Shift-B	formatcode
显示 go to line (转到行) 框	Command-L	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Command-Shift-U	lambdaUploadFunction
创建新文件	Control-N	newfile
显示 Preferences (首选项) 选项卡	Command-,	openpreferences

描述	键绑定	命令
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Command-Option-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Command-V	paste
显示对修复错误的建议	Command-F3	quickfix
重做上次操作	Command-Shift-Z Command-Y	redo
刷新预览窗格	Command-Enter	reloadpreview
开始重命名/重构选定内容	Option-Command-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为 表达式上	Option-Command-F	replace
重新运行您的初始化脚本	Command-Enter	rerunInitScript
重启环境	Command-R	restartc9
将当前文件重置为上次保存的版本	Control-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Option-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Command-S	save
使用不同的文件名将当前文件保存到磁盘	Command-Shift-S	saveas
显示多个文件的查找和替换栏	Shift-Command-F	searchinfiles
显示 Process List (进程列表) 对话框	Command-Option-P	showprocesslist

描述	键绑定	命令
撤消上次操作	Command-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡（当前选项卡除外）	Option-Control-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Option-Shift-W	closealltabs
关闭当前窗格	Command-Control-W	closepane
关闭当前选项卡	Option-W	closetab
向下移一个窗格	Control-Command-Down	gotopanedown
向左移一个窗格	Control-Command-Left	gotopaneleft
向右移一个窗格	Control-Command-Right	gotopaneright
向上移一个窗格	Control-Command-Up	gottopaneup
向左移一个选项卡	Command-[gototableft
向右移一个选项卡	Command-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Left	movetableft

描述	键绑定	命令
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Up	movetabup
转至下一窗格	Option-Esc	nextpane
转至下一选项卡	Option-Tab	nexttab
转至前一窗格	Option-Shift-Esc	previouspane
转至前一选项卡	Option-Shift-Tab	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Option-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Command-Shift-L	revealtab
转至第十个选项卡	Command-0	tab0
转至第一个选项卡	Command-1	tab1
转至第二个选项卡	Command-2	tab2
转至第三个选项卡	Command-3	tab3
转至第四个选项卡	Command-4	tab4
转至第五个选项卡	Command-5	tab5
转至第六个选项卡	Command-6	tab6
转至第七个选项卡	Command-7	tab7
转至第八个选项卡	Command-8	tab8

描述	键绑定	命令
转至第九个选项卡	Command	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Command-E Command-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Command-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Command-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Command-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Command-Shift-E	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	Control-Esc	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Command-U	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Control-Option-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光标，将第二个光标向上移动一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Control-Option-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Control-Option-A	alignCursors
回退一个字符	Control-Backspace Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Control-]	blockindent
将选定内容减少缩进一个制表符位置	Control-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce

描述	键绑定	命令
将选定内容居中	Control-L	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Command-Option-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Command-Option-Up	copylinesup
删除一个字符	Delete Control-D elete Shift-Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Command-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Command-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Control-Shift-M	expandToMatching
折叠选定代码，如果选中的是折叠的单元，则将其展开	Command-Option-L Command-F1	fold
折叠所有可折叠的元素	Control-Command-Op tion-0	foldall
折叠所有可折叠的元素（当前选定内容范围除外）	Command-Option-0	fold0ther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
转至文件末尾	Command-End Command-Down	gotoend
向左移一个字符	Left Control-B	gotoleft

描述	键绑定	命令
转至当前行末尾	Command-Right End Control-E	gotolineend
转至当前行开头	Command-Left Home Control-A	gotolinestart
转至下一错误	F4	goToNextError
向下移一页	Page Down Control-V	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Shift-F4	goToPreviousError
向右移一个字符	Right Control-F	gotoright
转至文件开头	Command-Home Command-Up	gotostart
向左移一个单词	Option-Left	gotowordleft
向右移一个单词	Option-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Control-P	jumptomatching
增大字体大小	Command-+ Command-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Option-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Option-Shift-Up	modifyNumberUp
将选定内容向下移一行	Option-Down	movelinesdown
将选定内容向上移一行	Option-Up	movelinesup

描述	键绑定	命令
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
向下移一页	Option-Page Down	pagedown
向上移一页	Option-Page Up	pageup
删除当前行	Command-D	removeline
删除从光标到当前行末尾的内容	Control-K	removetolineend
删除从当前行开头直至光标所在位置的内容	Command-Backspace	removetolinestart
删除光标左侧的单词	Option-Backspace Control-Option-Backspace	removewordleft
删除光标右侧的单词	Option-Delete	removewordright
回放以前记录的击键	Command-Shift-E	replaymacro
选择所有可选的内容	Command-A	selectall
将下一行包含到选定内容中	Shift-Down Control-Shift-N	selectdown
将左侧下一字符包含到选定内容中	Shift-Left Control-Shift-B	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart

描述	键绑定	命令
包含选定内容之后的更多匹配选择	Control-Option-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Control-Option-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Control-Option-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Control-Option-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Control-G	selectOrFindNext
选择或查找前一个匹配选择	Control-Shift-G	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Command-Shift-End Command-Shift-Down	selectttoend
将从光标到当前行末尾的部分包含到选定内容中	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Command-Shift-Left Control-Shift-A	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Control-Shift-P	selecttomatching

描述	键绑定	命令
将从光标向上到当前文件开头的部分包含到选定内容中	Command-Shift-Home Command-Shift-Up	selecttostart
将上一行包含到选定内容中	Shift-Up Control-Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Option-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Option-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Command-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Command--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Command-Option-S	sortlines
在当前行末尾添加光标	Control-Option-L	splitIntoLines
将光标所在位置内容移至行末尾，单独成行	Control-0	splitline
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Command-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Command-/	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget

描述	键绑定	命令
折叠父代码，如果已折叠，则取消折叠	Option-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Command-Option-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Control-W	toggleWordWrap
将选定内容更改为全部小写	Control-Shift-U	tolowercase
将选定内容更改为全部大写	Control-U	touppercase
转置选定内容	Control-T	transposeletters
展开选定代码	Command-Option-Shift-L Command-Shift-F1	unfold
展开整个文件的代码折叠	Command-Option-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式（如 $2*4$ 或 $10/2$ ）并输出结果	Shift-Command-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码，具体取决于当前文件的语法	Control-Option-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab

描述	键绑定	命令
转至下一段可编辑代码	Shift-Command-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Command-,	emmet_select_previous_item
展开一个缩写，然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Control-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Option-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Option-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Command-B	build
恢复当前暂停的流程	F8 Command-\	resume
运行或调试当前应用程序	Option-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11 Command-;	stepinto

描述	键绑定	命令
单步退出当前函数的范围	Shift-F11 Command-Shift-'	stepout
跳过堆栈中的当前表达式	F10 Command-'	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Control-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 MacOS Vim 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 MacOS 操作系统的 Vim 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Vim。
4. 对于 Operating System (操作系统)，选择 MacOS。

另请参阅[使用键绑定](#)。

- [一般性问题](#)
- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Command-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Control-Space Option-Space	complete
代码完成，然后覆盖	Control-Shift-Space Option-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Command-C	copy
将选定内容剪切到剪贴板中	Command-X	cut
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Command-F	find
选择当前文档中的所有查找匹配项	Control-Option-G	findAll
转至当前文档中下一处与您最后输入的查询匹配的内容	Command-G	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Command-Shift-G	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment（环境）窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree

描述	键绑定	命令
重新设置选定 JavaScript 代码的格式	Command-Shift-B	formatcode
显示 go to line (转到行) 框	Command-L	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Command-Shift-U	lambdaUploadFunction
创建新文件	Control-N	newfile
显示 Preferences (首选项) 选项卡	Command-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Command-Option-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Command-V	paste
显示对修复错误的建议	Command-F3	quickfix
重做上次操作	Command-Shift-Z Command-Y	redo
刷新预览窗格	Command-Enter	reloadpreview
开始重命名/重构选定内容	Option-Command-R	renameVar

描述	键绑定	命令
显示当前文档的查找和替换栏，光标位于替换为 表达式上	Option-Command-F	replace
重新运行您的初始化脚本	Command-Enter	rerunInitScript
重启环境	Command-R	restartc9
将当前文件重置为上次保存的版本	Control-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Option-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Command-S	save
使用不同的文件名将当前文件保存到磁盘	Command-Shift-S	saveas
显示多个文件的查找和替换栏	Shift-Command-F	searchinfiles
显示 Process List (进程列表) 对话框	Command-Option-P	showprocesslist
撤消上次操作	Command-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡 (当前选项卡除外)	Option-Control-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Option-Shift-W	closealltabs
关闭当前窗格	Command-Control-W	closepane

描述	键绑定	命令
关闭当前选项卡	Option-W	closetab
向下移一个窗格	Control-Command-Down	gotopanedown
向左移一个窗格	Control-Command-Left	gotopaneleft
向右移一个窗格	Control-Command-Right	gotopaneright
向上移一个窗格	Control-Command-Up	gottopaneup
向左移一个选项卡	Command-[gototableft
向右移一个选项卡	Command-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Up	movetabup
转至下一窗格	Option-Esc	nextpane
转至下一选项卡	Option-Tab	nexttab
转至前一窗格	Option-Shift-Esc	previouspane
转至前一选项卡	Option-Shift-Tab	previoustab

描述	键绑定	命令
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Option-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Command-Shift-L	revealtab
转至第十个选项卡	Command-0	tab0
转至第一个选项卡	Command-1	tab1
转至第二个选项卡	Command-2	tab2
转至第三个选项卡	Command-3	tab3
转至第四个选项卡	Command-4	tab4
转至第五个选项卡	Command-5	tab5
转至第六个选项卡	Command-6	tab6
转至第七个选项卡	Command-7	tab7
转至第八个选项卡	Command-8	tab8
转至第九个选项卡	Command	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Command-E Command-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Command-. F1	gotocommand

描述	键绑定	命令
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Command-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Command-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Command-Shift-E	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	Control-Esc	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Command-U	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Control-Option-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光标，将第二个光标向上移动一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Control-Option-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光	Control-Option-Shift-Down	addCursorBelowSkipCurrent

描述	键绑定	命令
标，将第二个光标向下移动一行		
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Control-Option-A	alignCursors
回退一个字符	Control-Backspace Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Control-]	blockindent
将选定内容伸出一个制表符位置	Control-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Command-Z Command-S hift-Z Command-Y	cancelBrowserUndoInAce
将选定内容居中	Control-L	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Command-Option-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Command-Option-Up	copylinesup
删除一个字符	Delete Control-Delete Shift-Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Command-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Command-Shift-L	expandtoline

描述	键绑定	命令
将下一次出现的匹配符号之前的内容都选中	Control-Shift-M	expandToMatching
折叠选定代码，如果选中的是折叠的单元，则将其展开	Command-Option-L Command-F1	fold
折叠所有可折叠的元素	Control-Command-Option-0	foldall
折叠所有可折叠的元素（当前选定内容范围除外）	Command-Option-0	foldOther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
转至文件末尾	Command-End Command-Down	gotoend
向左移一个字符	Left Control-B	gotoleft
转至当前行末尾	Command-Right End Control-E	gotolineend
转至当前行开头	Command-Left Home Control-A	gotolinestart
转至下一错误	F4	goToNextError
向下移一页	Page Down Control-V	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Shift-F4	goToPreviousError
向右移一个字符	Right Control-F	gotoright
转至文件开头	Command-Home Command-Up	gotostart

描述	键绑定	命令
向左移一个单词	Option-Left	gotowordleft
向右移一个单词	Option-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Control-P	jumptomatching
增大字体大小	Command-+ Command-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Option-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Option-Shift-Up	modifyNumberUp
将选定内容向下移一行	Option-Down	movelinesdown
将选定内容向上移一行	Option-Up	movelinesup
将选定内容伸出一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
向下移一页	Option-Page Down	pagedown
向上移一页	Option-Page Up	pageup
删除当前行	Command-D	removeline
删除从光标到当前行末尾的内容	Control-K	removetolineend
删除从当前行开头直至光标所在位置的内容	Command-Backspace	removetolinestart

描述	键绑定	命令
删除光标左侧的单词	Option-Backspace Control-Option-Backspace	removewordleft
删除光标右侧的单词	Option-Delete	removewordright
回放以前记录的击键	Command-Shift-E	replaymacro
选择所有可选的内容	Command-A	selectall
将下一行包含到选定内容中	Shift-Down Control-Shift-N	selectdown
将左侧下一字符包含到选定内容中	Shift-Left Control-Shift-B	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Control-Option-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Control-Option-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Control-Option-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Control-Option-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Control-G	selectOrFindNext
选择或查找前一个匹配选择	Control-Shift-G	selectOrFindPrevious

描述	键绑定	命令
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Command-Shift-End Command-Shift-Down	selecttoend
将从光标到当前行末尾的部分包含到选定内容中	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Command-Shift-Left Control-Shift-A	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Control-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Command-Shift-Home Command-Shift-Up	selecttostart
将上一行包含到选定内容中	Shift-Up Control-Shift-P	selectup
将光标左侧下一单词包含到选定内容中	Option-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Option-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Command-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection

描述	键绑定	命令
减小字体大小	Command--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Command-Option-S	sortlines
在当前行末尾添加光标	Control-Option-L	splitIntoLines
将光标所在位置内容移至行末尾，单独成行	Control-0	splitline
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Command-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Command-/	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Option-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Command-Option-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Control-W	toggleWordWrap
将选定内容更改为全部小写	Control-Shift-U	tolowercase
将选定内容更改为全部大写	Control-U	touppercase
转置选定内容	Control-T	transposeletters

描述	键绑定	命令
展开选定代码	Command-Option-Shift-L Command-Shift-F1	unfold
展开整个文件的代码折叠	Command-Option-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式 (如 $2*4$ 或 $10/2$) 并输出结果	Shift-Command-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码, 具体取决于当前文件的语法	Control-Option-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Command-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Command-,	emmet_select_previous_item
展开一个缩写, 然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Control-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Option-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Option-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Command-B	build
恢复当前暂停的流程	F8 Command-\	resume
运行或调试当前应用程序	Option-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11 Command-;	stepinto
单步退出当前函数的范围	Shift-F11 Command-Shift-'	stepout
跳过堆栈中的当前表达式	F10 Command-'	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Control-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 MacOS Emacs 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 MacOS 操作系统的 Emacs 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Emacs。
4. 对于 Operating System (操作系统)，选择 MacOS。

另请参阅[使用键绑定](#)。

- [一般性问题](#)
- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Command-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Control-Space Option-Space	complete
完成代码，然后覆盖	Control-Shift-Space Option-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Command-C	copy
将选定内容剪切到剪贴板中	Command-X	cut

描述	键绑定	命令
展开代码 (如果适用)	Tab	expandSnippet
显示当前文档的查找和替换栏	Command-F	find
选择当前文档中的所有查找匹配项	Control-Option-G	findAll
转至当前文档中下一处与您最后输入的查询匹配的内容	Command-G	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Command-Shift-G	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Command-Shift-B	formatcode
显示 go to line (转到行) 框	Command-L	gotoline
隐藏查找和替换栏 (如果显示)	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Command-Shift-U	lambdaUploadFunction
创建新文件	Control-N	newfile

描述	键绑定	命令
显示 Preferences (首选项) 选项卡	Command-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Command-Option-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Command-V	paste
显示对修复错误的建议	Command-F3	quickfix
重做上次操作	Command-Shift-Z Command-Y	redo
刷新预览窗格	Command-Enter	reloadpreview
开始重命名/重构选定内容	Option-Command-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为 表达式上	Option-Command-F	replace
重新运行您的初始化脚本	Command-Enter	rerunInitScript
重启环境	Command-R	restartc9
将当前文件重置为上次保存的版本	Control-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Option-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Command-S	save
使用不同的文件名将当前文件保存到磁盘	Command-Shift-S	saveas
显示多个文件的查找和替换栏	Shift-Command-F	searchinfiles

描述	键绑定	命令
显示 Process List (进程列表) 对话框	Command-Option-P	showprocesslist
撤消上次操作	Command-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡 (当前选项卡除外)	Option-Control-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Option-Shift-W	closealltabs
关闭当前窗格	Command-Control-W	closepane
关闭当前选项卡	Option-W	closetab
向下移一个窗格	Control-Command-Down	gotopanedown
向左移一个窗格	Control-Command-Left	gotopaneleft
向右移一个窗格	Control-Command-Right	gotopaneright
向上移一个窗格	Control-Command-Up	gottopaneup
向左移一个选项卡	Command-[gototableft
向右移一个选项卡	Command-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Down	movetabdown

描述	键绑定	命令
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Left	movetableleft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Up	movetabup
转至下一窗格	Option-Esc	nextpane
转至下一选项卡	Option-Tab	nexttab
转至前一窗格	Option-Shift-Esc	previouspane
转至前一选项卡	Option-Shift-Tab	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Option-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Command-Shift-L	revealtab
转至第十个选项卡	Command-0	tab0
转至第一个选项卡	Command-1	tab1
转至第二个选项卡	Command-2	tab2
转至第三个选项卡	Command-3	tab3
转至第四个选项卡	Command-4	tab4
转至第五个选项卡	Command-5	tab5
转至第六个选项卡	Command-6	tab6

描述	键绑定	命令
转至第七个选项卡	Command-7	tab7
转至第八个选项卡	Command-8	tab8
转至第九个选项卡	Command	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Command-E Command-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Command-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Command-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Command-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Command-Shift-E	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	Control-Esc	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Command-U	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Control-Option-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光标，将第二个光标向上移动一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Control-Option-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Control-Option-A	alignCursors
回退一个字符	Control-Backspace Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Control-]	blockindent
将选定内容伸出一个制表符位置	Control-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce

描述	键绑定	命令
将选定内容居中	Control-L	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Command-Option-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Command-Option-Up	copylinesup
删除一个字符	Delete Control-D elete Shift-Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Command-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Command-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Control-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Command-Option-L Command-F1	fold
折叠所有可折叠的元素	Control-Command-Op tion-0	foldall
折叠所有可折叠的元素（当前选定内容范围除外）	Command-Option-0	fold0ther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
转至文件末尾	Command-End Command-Down	gotoend
向左移一个字符	Left Control-B	gotoleft

描述	键绑定	命令
转至当前行末尾	Command-Right End Control-E	gotolineend
转至当前行开头	Command-Left Home Control-A	gotolinestart
转至下一错误	F4	goToNextError
向下移一页	Page Down Control-V	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Shift-F4	goToPreviousError
向右移一个字符	Right Control-F	gotoright
转至文件开头	Command-Home Command-Up	gotostart
向左移一个单词	Option-Left	gotowordleft
向右移一个单词	Option-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Control-P	jumptomatching
增大字体大小	Command-+ Command-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Option-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Option-Shift-Up	modifyNumberUp
将选定内容向下移一行	Option-Down	movelinesdown
将选定内容向上移一行	Option-Up	movelinesup

描述	键绑定	命令
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
向下移一页	Option-Page Down	pagedown
向上移一页	Option-Page Up	pageup
删除当前行	Command-D	removeline
删除从光标到当前行末尾的内容	Control-K	removetolineend
删除从当前行开头直至光标所在位置的内容	Command-Backspace	removetolinestart
删除光标左侧的单词	Option-Backspace Control-Option-Backspace	removewordleft
删除光标右侧的单词	Option-Delete	removewordright
回放以前记录的击键	Command-Shift-E	replaymacro
选择所有可选的内容	Command-A	selectall
将下一行包含到选定内容中	Shift-Down Control-Shift-N	selectdown
将左侧下一字符包含到选定内容中	Shift-Left Control-Shift-B	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart

描述	键绑定	命令
包含选定内容之后的更多匹配选择	Control-Option-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Control-Option-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Control-Option-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Control-Option-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Control-G	selectOrFindNext
选择或查找前一个匹配选择	Control-Shift-G	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Command-Shift-End Command-Shift-Down	selectttoend
将从光标到当前行末尾的部分包含到选定内容中	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Command-Shift-Left Control-Shift-A	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Control-Shift-P	selecttomatching

描述	键绑定	命令
将从光标向上到当前文件开头的部分包含到选定内容中	Command-Shift-Home Command-Shift-Up	selecttostart
将上一行包含到选定内容中	Shift-Up Control-Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Option-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Option-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Command-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Command--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Command-Option-S	sortlines
在当前行末尾添加光标	Control-Option-L	splitIntoLines
将光标所在位置内容移至行末尾，单独成行	Control-0	splitline
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Command-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Command-/	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget

描述	键绑定	命令
折叠父代码，如果已折叠，则取消折叠	Option-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Command-Option-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Control-W	toggleWordWrap
将所选内容更改为全部小写	Control-Shift-U	tolowercase
将所选内容更改为全部大写	Control-U	touppercase
置换所选内容	Control-T	transposeletters
展开选定代码	Command-Option-Shift-L Command-Shift-F1	unfold
展开整个文件的代码折叠	Command-Option-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式（如 $2*4$ 或 $10/2$ ）并输出结果	Shift-Command-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码，具体取决于当前文件的语法	Control-Option-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab

描述	键绑定	命令
转至下一段可编辑代码	Shift-Command-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Command-,	emmet_select_previous_item
展开一个缩写，然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Control-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Option-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Option-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Command-B	build
恢复当前暂停的流程	F8 Command-\	resume
运行或调试当前应用程序	Option-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11 Command-;	stepinto

描述	键绑定	命令
单步退出当前函数的范围	Shift-F11 Command-Shift-'	stepout
跳过堆栈中的当前表达式	F10 Command-'	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Control-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 MacOS Sublime 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 MacOS 操作系统的 Sublime 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Sublime。
4. 对于 Operating System (操作系统)，选择 MacOS。

另请参阅[使用键绑定](#)。

- [一般性问题](#)
- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Command-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Control-Space Option-Space	complete
代码完成，然后覆盖	Control-Shift-Space Option-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Command-C	copy
将选定内容剪切到剪贴板中	Command-X	cut
删除从光标到行开头的内容	Command-K Command-Backspace Command-Backspace	delete_to_hard_bol
删除从光标到行末尾的内容	Command-K Command-K Command-Delete Control-K	delete_to_hard_eol
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Command-F	find
突出显示选定内容的所有匹配项	Control-Command-G	find_all_under
突出显示选定内容的下一匹配项	Option-Command-G	find_under
突出显示光标周围区域和突出显示的所有匹配项	Command-D	find_under_expand

描述	键绑定	命令
突出显示光标周围区域并为突出显示的所有匹配项添加轮廓线	Command-K Command-D	find_under_expand_skip
突出显示所选内容的上一匹配项	Shift-Option-Command-G	find_under_previous
选择当前文档中的所有查找匹配项	Control-Option-G	findAll
转至当前文档中下一处与您最后输入的查询匹配的内容	Command-G	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Shift-Command-G	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Control-Option-F	formatcode
显示“go to line (转到行)”框	Control-G	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F12 Command-Option-Down	jumptodef

描述	键绑定	命令
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Command-Shift-U	lambdaUploadFunction
转到当前单词的末尾	Option-Right	moveToWordEndRight
转到当前单词的开头	Option-Left	moveToWordStartLeft
创建新文件	Control-N	newfile
显示 Preferences (首选项) 选项卡	Command-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Command-Option-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Command-V	paste
显示对修复错误的建议	Command-F3	quickfix
重做上次操作	Command-Shift-Z Command-Y	redo
刷新预览窗格	Command-Enter	reloadpreview
开始重命名/重构选定内容	Option-Command-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为表达式上	Command-Option-F	replace
在查找和替换栏中，使用替换为表达式替换查找表达式的所有匹配项	Control-Option-Enter	replaceall

描述	键绑定	命令
在查找和替换栏中，使用替换为表达式替换查找表达式的下一个匹配项	Command-Option-E	replacenext
重新运行您的初始化脚本	Command-Enter	rerunInitScript
重启环境	Command-R	restartc9
将当前文件重置为上次保存的版本	Control-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Option-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Command-S	save
使用不同的文件名将当前文件保存到磁盘	Command-Shift-S	saveas
显示多个文件的查找和替换栏	Command-Shift-F	searchinfiles
将从光标到单词末尾的部分包含到选定内容中	Option-Shift-Right	selectToWordEndRight
将从光标到单词开头的部分包含到选定内容中	Option-Shift-Left	selectToWordStartLeft
显示 Process List (进程列表) 对话框	Command-Option-P	showprocesslist
撤消上次操作	Command-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡（当前选项卡除外）	Option-Control-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Option-Shift-W	closealltabs
关闭当前窗格	Command-Control-W	closepane
关闭当前选项卡	Option-W	closetab
向下移一个窗格	Control-Command-Down	gotopanedown
向左移一个窗格	Control-Command-Left	gotopaneleft
向右移一个窗格	Control-Command-Right	gotopaneright
向上移一个窗格	Control-Command-Up	gottopaneup
向左移一个选项卡	Command-Shift-[Command-Option-Left	gototableft
向右移一个选项卡	Command-Shift-] Command-Option-Right	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Right	movetabright

描述	键绑定	命令
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Command-Option-Shift-Up	movetabup
转至下一选项卡	Control-Tab	nexttab
转至前一窗格	Option-Shift-Esc	previouspane
转至前一选项卡	Control-Shift-Tab	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Command-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Command-E	revealtab
转至第十个选项卡	Command-0	tab0
转至第一个选项卡	Command-1	tab1
转至第二个选项卡	Command-2	tab2
转至第三个选项卡	Command-3	tab3
转至第四个选项卡	Command-4	tab4
转至第五个选项卡	Command-5	tab5
转至第六个选项卡	Command-6	tab6
转至第七个选项卡	Command-7	tab7
转至第八个选项卡	Command-8	tab8
转至第九个选项卡	Command	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Command-E Command-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Command-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Command-O	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Command-Shift-O	gotosymbol
显示 Outline (大纲) 窗口	Command-Shift-R	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	Control-`	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Command-K Command-B	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Control-Shift-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光	Control-Option-Shift-Up	addCursorAboveSkipCurrent

描述	键绑定	命令
标，将第二个光标向上移动一行		
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Control-Shift-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Control-Option-A	alignCursors
回退一个字符	Control-Backspace Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Control-]	blockindent
将选定内容减少缩进一个制表符位置	Control-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce
将选定内容居中	Command-K Command-C Control-L	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Command-Option-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Command-Option-Up	copylinesup

描述	键绑定	命令
删除一个字符	Delete Control-D delete Shift-Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Command-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Command-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Control-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Command-Option-L Command-F1	fold
折叠所有可折叠的元素	Control-Command-Op tion-0	foldall
折叠所有可折叠的元素（当前选定内容范围除外）	Command-K Command-1	foldOther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
转至文件末尾	Command-End Command-Down	gotoend
向左移一个字符	Left Control-B	gotoleft
转至当前行末尾	Command-Right End Control-E	gotolineend
转至当前行开头	Command-Left Home Control-A	gotolinestart
转至下一错误	Control-F6	goToNextError

描述	键绑定	命令
向下移一页	Page Down Control-V	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Control-Shift-F6	goToPreviousError
向右移一个字符	Right Control-F	gotoright
转至文件开头	Command-Home Command-Up	gotostart
向左移一个单词	Option-Left	gotowordleft
向右移一个单词	Option-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
将选定的行合并为一行	Command-J	joinlines
转至当前范围中的匹配符号	Control-M	jumptomatching
增大字体大小	Command-= Command-+	largerfont
如果光标左侧是一个数字，则将其将减少 1	Option-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Option-Up	modifyNumberUp
将选定内容向下移一行	Control-Command-Down	movelinesdown
将选定内容向上移一行	Control-Command-Up	movelinesup
将选定内容伸出一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite

描述	键绑定	命令
向下移一页	Option-Page Down	pagedown
向上移一页	Option-Page Up	pageup
删除当前行的内容	Control-Shift-K	removeline
删除从光标到当前行末尾的内容	Control-K	removetolineend
删除从当前行开头直至光标所在位置的内容	Command-Backspace	removetolinestart
删除光标左侧的单词	Option-Backspace Control-Option-Backspace	removewordleft
删除光标右侧的单词	Option-Delete	removewordright
回放以前记录的击键	Control-Shift-Q	replaymacro
选择所有可选的内容	Command-A	selectall
将下一行包含到选定内容中	Shift-Down Control-Shift-N	selectdown
将左侧下一字符包含到选定内容中	Shift-Left Control-Shift-B	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
在所选内容中包含从当前行开头到光标的内容	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Control-Option-Right	selectMoreAfter

描述	键绑定	命令
包含选定内容之前的更多匹配选择	Control-Option-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Control-Option-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Control-Option-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Control-G	selectOrFindNext
选择或查找前一个匹配选择	Control-Shift-G	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Command-Shift-End Command-Shift-Down	selectttoend
将从光标到当前行末尾的部分包含到选定内容中	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Command-Shift-Left Control-Shift-A	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Control-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Command-Shift-Home Command-Shift-Up	selecttostart

描述	键绑定	命令
将上一行包含到选定内容中	Shift-Up Control-Shift-P	selectup
将光标左侧下一单词包含到选定内容中	Option-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Option-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Command-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Command--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	F5	sortlines
在当前行末尾添加光标	Command-Shift-L	splitIntoLines
将光标所在位置内容移至行末尾，单独成行	Control-0	splitline
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Command-Option-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Command-/	togglecomment
折叠代码，如果已折叠，则取消折叠	Command-Option-[toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Option-F2	toggleParentFoldWidget

描述	键绑定	命令
开始击键记录，如果已在记录，则停止	Control-Q	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Control-W	toggleWordWrap
将选定内容更改为全部小写	Command-K Command-L	toLowerCase
将选定内容更改为全部大写	Command-K Command-U	toUpperCase
转置选定内容	Control-T	transposeLetters
展开选定代码	Command-Option-]	unfold
展开整个文件的代码折叠	Command-K Command-O Command-K Command-J	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式（如 $2*4$ 或 $10/2$ ）并输出结果	Shift-Command-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码，具体取决于当前文件的语法	Control-Option-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Command-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Command-,	emmet_select_previous_item

描述	键绑定	命令
展开一个缩写，然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Control-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Option-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Option-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	F7 Command-B	build
恢复当前暂停的流程	F8 Command-\	resume
运行或调试当前应用程序	Command-Shift-B	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11 Command-;	stepinto
单步退出当前函数的范围	Shift-F11 Command-Shift-'	stepout
跳过堆栈中的当前表达式	F10 Command-'	stepover
停止运行或调试当前应用程序	Shift-F5	stop

描述	键绑定	命令
停止生成当前文件	Control-Break	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 Windows/Linux 默认键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 Windows/Linux 操作系统的默认键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Default (默认)。
4. 对于 Operating System (操作系统)，选择 Windows/Linux。

另请参阅[使用键绑定](#)。

- [一般性问题](#)
- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Ctrl-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut

描述	键绑定	命令
显示代码完成上下文菜单	Ctrl-Space Alt-Space	complete
代码完成，然后覆盖	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Ctrl-C	copy
将选定内容剪切到剪贴板中	Ctrl-X	cut
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Ctrl-F	find
选择当前文档中的所有查找匹配项	Ctrl-Alt-K	findall
转至当前文档中下一处与您最后输入的查询匹配的内容	Ctrl-K	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Ctrl-Shift-K	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment（环境）窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Ctrl-Shift-B	formatcode
显示“go to line（转到行）”框	Ctrl-G	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace

描述	键绑定	命令
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Ctrl-Shift-U	lambdaUploadFunction
创建新文件	Alt-N	newfile
显示 Preferences (首选项) 选项卡	Ctrl-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Alt-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Ctrl-V	paste
显示对修复错误的建议	Ctrl-F3	quickfix
重做上次操作	Ctrl-Shift-Z Ctrl-Y	redo
刷新预览窗格	Ctrl-Enter	reloadpreview
开始重命名/重构选定内容	Ctrl-Alt-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为表达式上	Alt-Shift-F Ctrl-H	replace
重新运行您的初始化脚本	Ctrl-Enter	rerunInitScript
重启环境	Ctrl-R	restartc9
将当前文件重置为上次保存的版本	Ctrl-Shift-Q	reverttosaved

描述	键绑定	命令
将每个打开的文件重置为上次保存的版本	Alt-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Ctrl-S	save
使用不同的文件名将当前文件保存到磁盘	Ctrl-Shift-S	saveas
显示多个文件的查找和替换栏	Ctrl-Shift-F	searchinfiles
显示 Process List (进程列表) 对话框	Ctrl-Alt-P	showprocesslist
撤消上次操作	Ctrl-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡 (当前选项卡除外)	Ctrl-Alt-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Alt-Shift-W	closealltabs
关闭当前窗格	Ctrl-W	closepane
关闭当前选项卡	Alt-W	closetab
向下移一个窗格	Ctrl-Meta-Down	gotopanedown
向左移一个窗格	Ctrl-Meta-Left	gotopaneleft
向右移一个窗格	Ctrl-Meta-Right	gotopaneright
向上移一个窗格	Ctrl-Meta-Up	gottopaneup

描述	键绑定	命令
向左移一个选项卡	Ctrl-[gototableft
向右移一个选项卡	Ctrl-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Up	movetabup
转至下一窗格	Ctrl-`	nextpane
转至下一选项卡	Ctrl-Tab Alt-`	nexttab
转至前一窗格	Ctrl-Shift-`	previouspane
转至前一选项卡	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Alt-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Ctrl-Shift-L	revealtab
转至第十个选项卡	Ctrl-0	tab0
转至第一个选项卡	Ctrl-1	tab1

描述	键绑定	命令
转至第二个选项卡	Ctrl-2	tab2
转至第三个选项卡	Ctrl-3	tab3
转至第四个选项卡	Ctrl-4	tab4
转至第五个选项卡	Ctrl-5	tab5
转至第六个选项卡	Ctrl-6	tab6
转至第七个选项卡	Ctrl-7	tab7
转至第八个选项卡	Ctrl-8	tab8
转至第九个选项卡	Ctrl-9	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Ctrl-E Ctrl-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Ctrl-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Ctrl-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Ctrl-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Ctrl-Shift-E	outline

描述	键绑定	命令
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	F6	<code>toggleconsole</code>
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Ctrl-I	<code>toggletree</code>

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Ctrl-Alt-Up	<code>addCursorAbove</code>
在当前光标上一行添加第二个光标，如果已添加了第二个光标，将第二个光标向上移动一行	Ctrl-Alt-Shift-Up	<code>addCursorAboveSkipCurrent</code>
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Ctrl-Alt-Down	<code>addCursorBelow</code>
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Ctrl-Alt-Shift-Down	<code>addCursorBelowSkipCurrent</code>
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Ctrl-Alt-A	<code>alignCursors</code>

描述	键绑定	命令
回退一个字符	Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Ctrl-]	blockindent
将选定内容减少缩进一个制表符位置	Ctrl-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
将选定内容居中	Ctrl-L	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Alt-Shift-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Alt-Shift-Up	copylinesup
剪切选定内容，如果没有选定内容则删除一个空格	Shift-Delete	cut_or_delete
删除一个字符	Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Ctrl-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Ctrl-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Ctrl-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Alt-L Ctrl-F1	fold

描述	键绑定	命令
折叠所有可折叠的元素	Ctrl-Command-Option-0	foldall
折叠所有可折叠的元素 (当前选定内容范围除外)	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
转至文件末尾	Ctrl-End	gotoend
向左移一个字符	Left	gotoleft
转至当前行末尾	Alt-Right End	gotolineend
转至当前行开头	Alt-Left Home	gotolinestart
转至下一错误	Alt-E	goToNextError
向下移一页	Page Down	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Alt-Shift-E	goToPreviousError
向右移一个字符	Right	gotoright
转至文件开头	Ctrl-Home	gotostart
向左移一个单词	Ctrl-Left	gotowordleft
向右移一个单词	Ctrl-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Ctrl-P	jumptomatching

描述	键绑定	命令
增大字体大小	Ctrl-+ Ctrl-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Ctrl-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Ctrl-Shift-Up	modifyNumberUp
将选定内容向下移一行	Alt-Down	movelinesdown
将选定内容向上移一行	Alt-Up	movelinesup
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
向下移一页	Option-Page Down	pagedown
向上移一页	Option-Page Up	pageup
删除当前行的内容	Ctrl-D	removeline
删除从光标到当前行末尾的内容	Alt-Delete	removetolineend
删除从当前行开头直至光标所在位置的内容	Alt-Backspace	removetolinestart
删除光标左侧的单词	Ctrl-Backspace	removewordleft
删除光标右侧的单词	Ctrl-Delete	removewordright
回放以前记录的击键	Ctrl-Shift-E	replaymacro
将当前文件向下滚动一行	Ctrl-Down	scrolldown
将当前文件向上滚动一行	Ctrl-Up	scrollup

描述	键绑定	命令
选择所有可选的内容	Ctrl-A	selectall
将下一行包含到选定内容中	Shift-Down	selectdown
将左侧下一字符包含到选定内容中	Shift-Left	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Ctrl-Alt-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Ctrl-Alt-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Ctrl-Alt-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Ctrl-Alt-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Alt-K	selectOrFindNext
选择或查找前一个匹配选择	Alt-Shift-K	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright

描述	键绑定	命令
将从光标向下到当前文件末尾的部分包含到选定内容中	Ctrl-Shift-End	selecttoend
将从光标到当前行末尾的部分包含到选定内容中	Alt-Shift-Right	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Alt-Shift-Left	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Ctrl-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Ctrl-Shift-Home	selecttostart
将上一行包含到选定内容中	Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Ctrl-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Ctrl-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Ctrl-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Ctrl--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Ctrl-Alt-S	sortlines
在当前行末尾添加光标	Ctrl-Alt-L	splitIntoLines
将光标所在位置内容移至行末尾，单独成行	Ctrl-0	splitline

描述	键绑定	命令
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Ctrl-Shift-/ /	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Ctrl-/ /	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Alt-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Ctrl-Alt-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Ctrl-Q	toggleWordWrap
将选定内容更改为全部小写	Ctrl-Shift-U	toLowerCase
将选定内容更改为全部大写	Ctrl-U	toUpperCase
转置选定内容	Alt-X	transposeLetters
展开选定代码	Alt-Shift-L Ctrl-Shift-F1	unfold
展开整个文件的代码折叠	Alt-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式 (如 $2*4$ 或 $10/2$) 并输出结果	Shift-Ctrl-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码, 具体取决于当前文件的语法	Ctrl-Alt-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Ctrl-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Ctrl-,	emmet_select_previous_item
展开一个缩写, 然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Ctrl-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Alt-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Alt-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Ctrl-B	build
恢复当前暂停的流程	F8	resume
运行或调试当前应用程序	Alt-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11	stepinto
单步退出当前函数的范围	Shift-F11	stepout
跳过堆栈中的当前表达式	F10	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Ctrl-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 Windows/Linux Vim 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 Windows/Linux 操作系统的 Vim 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Vim。
4. 对于 Operating System (操作系统)，选择 Windows/Linux。

另请参阅[使用键绑定](#)。

- [一般性问题](#)

- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Ctrl-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Ctrl-Space Alt-Space	complete
代码完成，然后覆盖	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Ctrl-C	copy
将选定内容剪切到剪贴板中	Ctrl-X	cut
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Ctrl-F	find
选择当前文档中的所有查找匹配项	Ctrl-Alt-K	findall
转至当前文档中下一处与您最后输入的查询匹配的内容	Ctrl-K	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Ctrl-Shift-K	findprevious

描述	键绑定	命令
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Ctrl-Shift-B	formatcode
显示“go to line (转到行)”框	Ctrl-G	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Ctrl-Shift-U	lambdaUploadFunction
创建新文件	Alt-N	newfile
显示 Preferences (首选项) 选项卡	Ctrl-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Alt-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Ctrl-V	paste
显示对修复错误的建议	Ctrl-F3	quickfix

描述	键绑定	命令
重做上次操作	Ctrl-Shift-Z Ctrl-Y	redo
刷新预览窗格	Ctrl-Enter	reloadpreview
开始重命名/重构选定内容	Ctrl-Alt-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为表达式上	Alt-Shift-F Ctrl-H	replace
重新运行您的初始化脚本	Ctrl-Enter	rerunInitScript
重启环境	Ctrl-R	restartc9
将当前文件重置为上次保存的版本	Ctrl-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Alt-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Ctrl-S	save
使用不同的文件名将当前文件保存到磁盘	Ctrl-Shift-S	saveas
显示多个文件的查找和替换栏	Ctrl-Shift-F	searchinfiles
显示 Process List (进程列表) 对话框	Ctrl-Alt-P	showprocesslist
撤消上次操作	Ctrl-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡 (当前选项卡除外)	Ctrl-Alt-W	closeallbutme

描述	键绑定	命令
关闭所有窗格中所有打开的选项卡	Alt-Shift-W	closealltabs
关闭当前窗格	Ctrl-W	closepane
关闭当前选项卡	Alt-W	closetab
向下移一个窗格	Ctrl-Meta-Down	gotopanedown
向左移一个窗格	Ctrl-Meta-Left	gotopaneleft
向右移一个窗格	Ctrl-Meta-Right	gotopaneright
向上移一个窗格	Ctrl-Meta-Up	gottopaneup
向左移一个选项卡	Ctrl-[gototableft
向右移一个选项卡	Ctrl-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Up	movetabup
转至下一窗格	Ctrl-`	nextpane
转至下一选项卡	Ctrl-Tab Alt-`	nexttab

描述	键绑定	命令
转至前一窗格	Ctrl-Shift-`	previouspane
转至前一选项卡	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Alt-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Ctrl-Shift-L	revealtab
转至第十个选项卡	Ctrl-0	tab0
转至第一个选项卡	Ctrl-1	tab1
转至第二个选项卡	Ctrl-2	tab2
转至第三个选项卡	Ctrl-3	tab3
转至第四个选项卡	Ctrl-4	tab4
转至第五个选项卡	Ctrl-5	tab5
转至第六个选项卡	Ctrl-6	tab6
转至第七个选项卡	Ctrl-7	tab7
转至第八个选项卡	Ctrl-8	tab8
转至第九个选项卡	Ctrl-9	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Ctrl-E Ctrl-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Ctrl-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Ctrl-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Ctrl-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Ctrl-Shift-E	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	F6	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Ctrl-I	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Ctrl-Alt-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	键绑定	命令
标，将第二个光标向上移动一行		
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Ctrl-Alt-Down	addCursorBelow
在当前光标下一行添加第二个光标。如果已添加了第二个光标，将第二个光标向下移动一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Ctrl-Alt-A	alignCursors
回退一个字符	Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Ctrl-]	blockindent
将选定内容减少缩进一个制表符位置	Ctrl-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
复制一行的内容，并将复制的内容粘贴到下一行	Alt-Shift-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Alt-Shift-Up	copylinesup
剪切选定内容。如果未选择任何内容，删除一个字符	Shift-Delete	cut_or_delete
删除一个字符	Delete	del

描述	键绑定	命令
复制选定的内容，并在选择后立即粘贴复制的内容	Ctrl-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Ctrl-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Ctrl-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Alt-L Ctrl-F1	fold
折叠所有可折叠的元素（当前选定内容范围除外）	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
转至文件末尾	Ctrl-End	gotoend
向左移一个字符	Left	gotoleft
转至当前行末尾	Alt-Right End	gotolineend
转至当前行开头	Alt-Left Home	gotolinestart
转至下一错误	Alt-E	goToNextError
向下移一页	Page Down	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Alt-Shift-E	goToPreviousError
向右移一个字符	Right	gotoright
转至文件开头	Ctrl-Home	gotostart

描述	键绑定	命令
向左移一个单词	Ctrl-Left	gotowordleft
向右移一个单词	Ctrl-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Ctrl-P	jumptomatching
增大字体大小	Ctrl-+ Ctrl-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Ctrl-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Ctrl-Shift-Up	modifyNumberUp
将选定内容向下移一行	Alt-Down	movelinesdown
将选定内容向上移一行	Alt-Up	movelinesup
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
删除当前行的内容	Ctrl-D	removeline
删除从光标到当前行末尾的内容	Alt-Delete	removetolineend
删除从当前行开头直至光标所在位置的内容	Alt-Backspace	removetolinestart
删除光标左侧的单词	Ctrl-Backspace	removewordleft
删除光标右侧的单词	Ctrl-Delete	removewordright
回放以前记录的击键	Ctrl-Shift-E	replaymacro

描述	键绑定	命令
将当前文件向下滚动一行	Ctrl-Down	scrolldown
将当前文件向上滚动一行	Ctrl-Up	scrollup
选择所有可选的内容	Ctrl-A	selectall
将下一行包含到选定内容中	Shift-Down	selectdown
将左侧下一字符包含到选定内容中	Shift-Left	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Ctrl-Alt-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Ctrl-Alt-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Ctrl-Alt-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Ctrl-Alt-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Alt-K	selectOrFindNext
选择或查找前一个匹配选择	Alt-Shift-K	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup

描述	键绑定	命令
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Ctrl-Shift-End	selecttoend
将从光标到当前行末尾的部分包含到选定内容中	Alt-Shift-Right	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Alt-Shift-Left	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Ctrl-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Ctrl-Shift-Home	selecttostart
将上一行包含到选定内容中	Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Ctrl-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Ctrl-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Ctrl-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Ctrl--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Ctrl-Alt-S	sortlines
在当前行末尾添加光标	Ctrl-Alt-L	splitIntoLines

描述	键绑定	命令
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Ctrl-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Ctrl-/	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Alt-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Ctrl-Alt-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Ctrl-Q	toggleWordWrap
将选定内容更改为全部小写	Ctrl-Shift-U	tolowercase
将选定内容更改为全部大写	Ctrl-U	touppercase
转置选定内容	Alt-X	transposeletters
展开选定代码	Alt-Shift-L Ctrl-Shift-F1	unfold
展开整个文件的代码折叠	Alt-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式 (如 $2*4$ 或 $10/2$) 并输出结果	Shift-Ctrl-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码, 具体取决于当前文件的语法	Ctrl-Alt-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Ctrl-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Ctrl-,	emmet_select_previous_item
展开一个缩写, 然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Ctrl-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Alt-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Alt-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Ctrl-B	build
恢复当前暂停的流程	F8	resume
运行或调试当前应用程序	Alt-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11	stepinto
单步退出当前函数的范围	Shift-F11	stepout
跳过堆栈中的当前表达式	F10	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Ctrl-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 Windows/Linux Emacs 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 Windows/Linux 操作系统的 Emacs 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Emacs。
4. 对于 Operating System (操作系统)，选择 Windows/Linux。

另请参阅[使用键绑定](#)。

- [一般性问题](#)

- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Ctrl-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Ctrl-Space Alt-Space	complete
代码完成，然后覆盖	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Ctrl-C	copy
将选定内容剪切到剪贴板中	Ctrl-X	cut
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Ctrl-F	find
选择当前文档中的所有查找匹配项	Ctrl-Alt-K	findall
转至当前文档中下一处与您最后输入的查询匹配的内容	Ctrl-K	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Ctrl-Shift-K	findprevious

描述	键绑定	命令
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Ctrl-Shift-B	formatcode
显示“go to line (转到行)”框	Ctrl-G	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace
转到光标所在位置变量或函数的定义	F3	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Ctrl-Shift-U	lambdaUploadFunction
创建新文件	Alt-N	newfile
显示 Preferences (首选项) 选项卡	Ctrl-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Alt-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Ctrl-V	paste
显示对修复错误的建议	Ctrl-F3	quickfix

描述	键绑定	命令
重做上次操作	Ctrl-Shift-Z Ctrl-Y	redo
刷新预览窗格	Ctrl-Enter	reloadpreview
开始重命名/重构选定内容	Ctrl-Alt-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为表达式上	Alt-Shift-F Ctrl-H	replace
重新运行您的初始化脚本	Ctrl-Enter	rerunInitScript
重启环境	Ctrl-R	restartc9
将当前文件重置为上次保存的版本	Ctrl-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Alt-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Ctrl-S	save
使用不同的文件名将当前文件保存到磁盘	Ctrl-Shift-S	saveas
显示多个文件的查找和替换栏	Ctrl-Shift-F	searchinfiles
显示 Process List (进程列表) 对话框	Ctrl-Alt-P	showprocesslist
撤消上次操作	Ctrl-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡 (当前选项卡除外)	Ctrl-Alt-W	closeallbutme

描述	键绑定	命令
关闭所有窗格中所有打开的选项卡	Alt-Shift-W	closealltabs
关闭当前窗格	Ctrl-W	closepane
关闭当前选项卡	Alt-W	closetab
向下移一个窗格	Ctrl-Meta-Down	gotopanedown
向左移一个窗格	Ctrl-Meta-Left	gotopaneleft
向右移一个窗格	Ctrl-Meta-Right	gotopaneright
向上移一个窗格	Ctrl-Meta-Up	gottopaneup
向左移一个选项卡	Ctrl-[gototableft
向右移一个选项卡	Ctrl-]	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Right	movetabright
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Up	movetabup
转至下一窗格	Ctrl-`	nextpane
转至下一选项卡	Ctrl-Tab Alt-`	nexttab

描述	键绑定	命令
转至前一窗格	Ctrl-Shift-`	previouspane
转至前一选项卡	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Alt-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Ctrl-Shift-L	revealtab
转至第十个选项卡	Ctrl-0	tab0
转至第一个选项卡	Ctrl-1	tab1
转至第二个选项卡	Ctrl-2	tab2
转至第三个选项卡	Ctrl-3	tab3
转至第四个选项卡	Ctrl-4	tab4
转至第五个选项卡	Ctrl-5	tab5
转至第六个选项卡	Ctrl-6	tab6
转至第七个选项卡	Ctrl-7	tab7
转至第八个选项卡	Ctrl-8	tab8
转至第九个选项卡	Ctrl-9	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Ctrl-E Ctrl-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Ctrl-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Ctrl-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Ctrl-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Ctrl-Shift-E	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	F6	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Ctrl-I	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Ctrl-Alt-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	键绑定	命令
标，将第二个光标向上移动一行		
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Ctrl-Alt-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Ctrl-Alt-A	alignCursors
回退一个字符	Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Ctrl-]	blockindent
将选定内容减少缩进一个制表符位置	Ctrl-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
复制一行的内容，并将复制的内容粘贴到下一行	Alt-Shift-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Alt-Shift-Up	copylinesup
剪切选定内容，如果没有选定内容则删除一个空格	Shift-Delete	cut_or_delete
删除一个字符	Delete	del

描述	键绑定	命令
复制选定的内容，并在选择后立即粘贴复制的内容	Ctrl-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Ctrl-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Ctrl-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Alt-L Ctrl-F1	fold
折叠所有可折叠的元素（当前选定内容范围除外）	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
转至文件末尾	Ctrl-End	gotoend
向左移一个字符	Left	gotoleft
转至当前行末尾	Alt-Right End	gotolineend
转至当前行开头	Alt-Left Home	gotolinestart
转至下一错误	Alt-E	goToNextError
向下移一页	Page Down	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Alt-Shift-E	goToPreviousError
向右移一个字符	Right	gotoright
转至文件开头	Ctrl-Home	gotostart

描述	键绑定	命令
向左移一个单词	Ctrl-Left	gotowordleft
向右移一个单词	Ctrl-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
转至当前范围中的匹配符号	Ctrl-P	jumptomatching
增大字体大小	Ctrl-+ Ctrl-=	largerfont
如果光标左侧是一个数字，则将其将减少 1	Ctrl-Shift-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Ctrl-Shift-Up	modifyNumberUp
将选定内容向下移一行	Alt-Down	movelinesdown
将选定内容向上移一行	Alt-Up	movelinesup
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
删除当前行的内容	Ctrl-D	removeline
删除从光标到当前行末尾的内容	Alt-Delete	removetolineend
删除从当前行开头直至光标所在位置的内容	Alt-Backspace	removetolinestart
删除光标左侧的单词	Ctrl-Backspace	removewordleft
删除光标右侧的单词	Ctrl-Delete	removewordright
回放以前记录的击键	Ctrl-Shift-E	replaymacro

描述	键绑定	命令
将当前文件向下滚动一行	Ctrl-Down	scrolldown
将当前文件向上滚动一行	Ctrl-Up	scrollup
选择所有可选的内容	Ctrl-A	selectall
将下一行包含到选定内容中	Shift-Down	selectdown
将左侧下一空格包含在选定内容中	Shift-Left	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Ctrl-Alt-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Ctrl-Alt-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Ctrl-Alt-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Ctrl-Alt-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Alt-K	selectOrFindNext
选择或查找前一个匹配选择	Alt-Shift-K	selectOrFindPrevious
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup

描述	键绑定	命令
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Ctrl-Shift-End	selecttoend
将从光标到当前行末尾的部分包含到选定内容中	Alt-Shift-Right	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Alt-Shift-Left	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Ctrl-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Ctrl-Shift-Home	selecttostart
将上一行包含到选定内容中	Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Ctrl-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Ctrl-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Ctrl-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection
减小字体大小	Ctrl--	smallerfont
如果已选择多行，则按排序顺序将其重新排列	Ctrl-Alt-S	sortlines
在当前行末尾添加光标	Ctrl-Alt-L	splitIntoLines

描述	键绑定	命令
将光标所在位置内容移至行末尾，单独成行	Ctrl-0	splitline
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Ctrl-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Ctrl-/	togglecomment
折叠代码，如果已折叠，则取消折叠	F2	toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Alt-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Ctrl-Alt-E	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Ctrl-Q	toggleWordWrap
将选定内容更改为全部小写	Ctrl-Shift-U	tolowercase
将选定内容更改为全部大写	Ctrl-U	touppercase
转置选定内容	Alt-X	transposeletters
展开选定代码	Alt-Shift-L Ctrl-Shift-F1	unfold
展开整个文件的代码折叠	Alt-Shift-0	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式 (如 $2*4$ 或 $10/2$) 并输出结果	Shift-Ctrl-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码, 具体取决于当前文件的语法	Ctrl-Alt-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Ctrl-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Ctrl-,	emmet_select_previous_item
展开一个缩写, 然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Ctrl-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Alt-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Alt-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	Ctrl-B	build
恢复当前暂停的流程	F8	resume
运行或调试当前应用程序	Alt-F5	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11	stepinto
单步退出当前函数的范围	Shift-F11	stepout
跳过堆栈中的当前表达式	F10	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Ctrl-Shift-C	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的 Windows/Linux Sublime 键绑定参考

下面是一组在 AWS Cloud9 IDE 中用于 Windows/Linux 操作系统的 Sublime 键盘模式键绑定。

有关更多信息，请在 AWS Cloud9 IDE 中：

1. 在菜单栏上，依次选择 AWS Cloud9 > Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，选择 Keybindings (键绑定)。
3. 对于 Keyboard Mode (键盘模式)，选择 Sublime。
4. 对于 Operating System (操作系统)，选择 Windows/Linux。

另请参阅[使用键绑定](#)。

- [一般性问题](#)

- [选项卡](#)
- [面板](#)
- [代码编辑器](#)
- [emmet](#)
- [终端](#)
- [运行和调试](#)

常规

描述	键绑定	命令
添加选定内容作为监视表达式	Ctrl-Shift-C	addwatchfromselection
删除剪贴板中的剪切选择	Esc	clearcut
显示代码完成上下文菜单	Ctrl-Space	complete
代码完成，然后覆盖	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
将选定内容复制到剪贴板中	Ctrl-C	copy
将选定内容剪切到剪贴板中	Ctrl-X	cut
删除从光标到行开头的内容	Ctrl-Shift-Backspace Ctrl-K Ctrl-Backspace	delete_to_hard_bol
删除从光标到行末尾的内容	Ctrl-Shift-Delete Ctrl-K Ctrl-K	delete_to_hard_eol
展开代码（如果适用）	Tab	expandSnippet
显示当前文档的查找和替换栏	Ctrl-F	find
突出显示选定内容的所有匹配项	Alt-F3	find_all_under

描述	键绑定	命令
突出显示选定内容的下一匹配项	Ctrl-F3	find_under
突出显示光标周围区域和要突出显示的所有匹配项	Ctrl-D	find_under_expand
突出显示光标周围区域并为要突出显示的所有匹配项添加轮廓线	Ctrl-K Ctrl-D	find_under_expand_skip
突出显示选定内容的上一匹配项	Ctrl-Shift-F3	find_under_prev
选择当前文档中的所有查找匹配项	Ctrl-Alt-K	findAll
转至当前文档中下一处与您最后输入的查询匹配的内容	F3	findnext
转至当前文档中前一处与您最后输入的查询匹配的内容	Shift-F3	findprevious
在编辑器内活动文件中显示对插入点符号的所有已知引用	Shift-F3	findReferences
打开 Environment (环境) 窗口，然后使文件列表处于活动状态	Shift-Esc	focusTree
重新设置选定 JavaScript 代码的格式	Ctrl-Alt-F	formatcode
显示“go to line (转到行)”框	Ctrl-G	gotoline
如果显示查找和替换栏，则将其隐藏	Esc	hidesearchreplace

描述	键绑定	命令
转到光标所在位置变量或函数的定义	F12	jumptodef
如果在 AWS Resources (Amazon 资源) 窗口的 Lambda 的部分中选择了本地 Lambda 函数，则尝试将函数作为远程函数上传到 Lambda	Ctrl-Shift-U	lambdaUploadFunction
转到当前单词的末尾	Ctrl-Right	moveToWordEndRight
转到当前单词的开头	Ctrl-Left	moveToWordStartLeft
创建新文件	Alt-N	newfile
显示 Preferences (首选项) 选项卡	Ctrl-,	openpreferences
打开 Terminal (终端) 选项卡，然后切换到文件列表中的选定文件的父文件夹	Alt-L	opentermhere
在光标所在位置粘贴当前剪贴板中的内容	Ctrl-V	paste
显示对修复错误的建议	Ctrl-F3	quickfix
重做上次操作	Ctrl-Shift-Z Ctrl-Y	redo
刷新预览窗格	Ctrl-Enter	reloadpreview
开始重命名/重构选定内容	Ctrl-Alt-R	renameVar
显示当前文档的查找和替换栏，光标位于替换为表达式上	Ctrl-H	replace

描述	键绑定	命令
在查找和替换栏中，使用替换为表达式替换查找表达式的所有匹配项	Ctrl-Alt-Enter	replaceall
在查找和替换栏中，使用替换为表达式替换查找表达式的下一个匹配项	Ctrl-Shift-H	replacenext
重新运行您的初始化脚本	Ctrl-Enter	rerunInitScript
重启环境	Ctrl-R	restartc9
将当前文件重置为上次保存的版本	Ctrl-Shift-Q	reverttosaved
将每个打开的文件重置为上次保存的版本	Alt-Shift-Q	reverttosavedall
将当前文件保存到磁盘	Ctrl-S	save
使用不同的文件名将当前文件保存到磁盘	Ctrl-Shift-S	saveas
显示多个文件的查找和替换栏	Ctrl-Shift-F	searchinfiles
将从光标到单词末尾的部分包含到选定内容中	Ctrl-Shift-Right	selectToWordEndRight
将从光标到单词开头的部分包含到选定内容中	Ctrl-Shift-Left	selectToWordStartLeft
显示 Process List (进程列表) 对话框	Ctrl-Alt-P	showprocesslist
撤消上次操作	Ctrl-Z	undo

选项卡

描述	键绑定	命令
关闭当前窗格中所有打开的选项卡（当前选项卡除外）	Ctrl-Alt-W	closeallbutme
关闭所有窗格中所有打开的选项卡	Alt-Shift-W	closealltabs
关闭当前窗格	Ctrl-W	closepane
关闭当前选项卡	Alt-W	closetab
向下移一个窗格	Ctrl-Meta-Down	gotopanedown
向左移一个窗格	Ctrl-Meta-Left	gotopaneleft
向右移一个窗格	Ctrl-Meta-Right	gotopaneright
向上移一个窗格	Ctrl-Meta-Up	gottopaneup
向左移一个选项卡	Ctrl-Page Up	gototableft
向右移一个选项卡	Ctrl-Page Down	gototabright
将当前选项卡下移一个窗格，如果选项卡已在最底部，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Down	movetabdown
将当前选项卡左移一个窗格，如果选项卡已在最左侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Left	movetableft
将当前选项卡右移一个窗格，如果选项卡已在最右侧，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Right	movetabright

描述	键绑定	命令
将当前选项卡上移一个窗格，如果选项卡已在最顶端，则在对应位置创建一个拆分选项卡	Ctrl-Meta-Up	movetabup
转至下一选项卡	Ctrl-Tab	nexttab
转至前一窗格	Ctrl-Shift-`	previouspane
转至前一选项卡	Ctrl-Shift-Tab	previoustab
返回上一选项卡	Esc	refocusTab
再次打开上一选项卡	Ctrl-Shift-T	reopenLastTab
在文件树中显示当前选项卡	Ctrl-E	revealtab
转至第十个选项卡	Ctrl-0	tab0
转至第一个选项卡	Ctrl-1	tab1
转至第二个选项卡	Ctrl-2	tab2
转至第三个选项卡	Ctrl-3	tab3
转至第四个选项卡	Ctrl-4	tab4
转至第五个选项卡	Ctrl-5	tab5
转至第六个选项卡	Ctrl-6	tab6
转至第七个选项卡	Ctrl-7	tab7
转至第八个选项卡	Ctrl-8	tab8
转至第九个选项卡	Ctrl-9	tab9

面板

描述	键绑定	命令
显示 Go to Anything (转到任意内容) 模式中的 Go (转到) 窗口	Ctrl-E Ctrl-P	gotoanything
显示 Go to Command (转到命令) 模式中的 Go (转到) 窗口	Ctrl-. F1	gotocommand
显示 Go to File (转到文件) 模式中的 Go (转到) 窗口。	Ctrl-0	gotofile
显示 Go to Symbol (转到符号) 模式中的 Go (转到) 窗口。	Ctrl-Shift-0	gotosymbol
显示 Outline (大纲) 窗口	Ctrl-R Ctrl-Shift-R	outline
显示隐藏的 Console (控制台) 窗口，或隐藏显示的控制台窗口	Ctrl-`	toggleconsole
显示隐藏的 Environment (环境) 窗口，或隐藏显示的环境窗口	Ctrl-K Ctrl-B	toggletree

代码编辑器

描述	键绑定	命令
在当前光标上一行添加一个光标，如果已添加了光标，则在其上方再添加一个光标	Ctrl-Alt-Up	addCursorAbove
在当前光标上一行添加第二个光标，如果已添加了第二个光	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	键绑定	命令
标，将第二个光标向上移动一行		
在当前光标下一行添加一个光标，如果已添加了光标，则在其下方再添加一个光标	Ctrl-Alt-Down	addCursorBelow
在当前光标下一行添加第二个光标，如果已添加了第二个光标，将第二个光标向下移动一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
如果各行光标未对齐，则将各行上的所有光标移至与当前光标对齐的位置	Ctrl-Alt-A	alignCursors
回退一个字符	Shift-Backspace Backspace	backspace
将选定内容缩进一个制表符位置	Ctrl-]	blockindent
将选定内容减少缩进一个制表符位置	Ctrl-[blockoutdent
控制是否可以将光标从编辑器切换到 IDE 中其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
将选定内容居中	Ctrl-K Ctrl-C	centerselection
复制一行的内容，并将复制的内容粘贴到下一行	Alt-Shift-Down	copylinesdown
复制一行的内容，并将复制的内容粘贴到上一行	Alt-Shift-Up	copylinesup
剪切选定内容，如果没有选定内容则删除一个空格	Shift-Delete	cut_or_delete

描述	键绑定	命令
删除一个字符	Delete	del
复制选定的内容，并在选择后立即粘贴复制的内容	Ctrl-Shift-D	duplicateSelection
将当前行的内容包含在选定内容中	Ctrl-Shift-L	expandtoline
将下一次出现的匹配符号之前的内容都选中	Ctrl-Shift-M	expandToMatching
折叠选定代码；如果选中的是折叠的单元，则将其展开	Alt-L Ctrl-F1	fold
折叠所有可折叠的元素（当前选定内容范围除外）	Ctrl-K Ctrl-1	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
转至文件末尾	Ctrl-End	gotoend
向左移一个字符	Left	gotoleft
转至当前行末尾	Alt-Right End	gotolineend
转至当前行开头	Alt-Left Home	gotolinestart
转至下一错误	Ctrl-F6	goToNextError
向下移一页	Page Down	gotopagedown
向上移一页	Page Up	gotopageup
转至前一错误	Ctrl-Shift-F6	goToPreviousError
向右移一个字符	Right	gotoright

描述	键绑定	命令
转至文件开头	Ctrl-Home	gotostart
向左移一个单词	Ctrl-Left	gotowordleft
向右移一个单词	Ctrl-Right	gotowordright
将选定内容缩进一个制表符位置	Tab	indent
将从光标到单词开头的部分包含到选定内容中	Ctrl-J	joinlines
转至当前范围中的匹配符号	Ctrl-M	jumptomatching
增大字体大小	Ctrl-- Ctrl-= Ctrl-+	largerfont
如果光标左侧是一个数字，则将其将减少 1	Alt-Down	modifyNumberDown
如果光标左侧是一个数字，则将其将增加 1	Alt-Up	modifyNumberUp
将选定内容向下移一行	Ctrl-Shift-Down	movelinesdown
将选定内容向上移一行	Ctrl-Shift-Up	movelinesup
将选定内容减少缩进一个制表符位置	Shift-Tab	outdent
打开/关闭覆盖模式	Insert	overwrite
删除当前行的内容	Ctrl-Shift-K	removeline
删除从光标到当前行末尾的内容	Alt-Delete	removetolineend
删除从当前行开头直至光标所在位置的内容	Alt-Backspace	removetolinestart

描述	键绑定	命令
删除光标左侧的单词	Ctrl-Backspace	removewordleft
删除光标右侧的单词	Ctrl-Delete	removewordright
回放以前记录的击键	Ctrl-Shift-Q	replaymacro
将当前文件向下滚动一行	Ctrl-Down	scrolldown
将当前文件向上滚动一行	Ctrl-Up	scrollup
选择所有可选的内容	Ctrl-A	selectall
将下一行包含到选定内容中	Shift-Down	selectdown
将左侧下一空格包含在选定内容中	Shift-Left	selectleft
将当前行上光标之后的其余部分包含到选定内容中	Shift-End	selectlineend
将当前行上从开头到光标的部分包含到选定内容中	Shift-Home	selectlinestart
包含选定内容之后的更多匹配选择	Ctrl-Alt-Right	selectMoreAfter
包含选定内容之前的更多匹配选择	Ctrl-Alt-Left	selectMoreBefore
包含选定内容之后的下一个匹配选择	Ctrl-Alt-Shift-Right	selectNextAfter
包含选定内容之前的下一个匹配选择	Ctrl-Alt-Shift-Left	selectNextBefore
选择或查找下一个匹配选择	Alt-K	selectOrFindNext
选择或查找前一个匹配选择	Alt-Shift-K	selectOrFindPrevious

描述	键绑定	命令
将从光标向下到当前页末尾的部分包含到选定内容中	Shift-Page Down	selectpagedown
将从光标向上到当前页开头的部分包含到选定内容中	Shift-Page Up	selectpageup
将光标右侧的下一字符包含到选定内容中	Shift-Right	selectright
将从光标向下到当前文件末尾的部分包含到选定内容中	Ctrl-Shift-End	selecttoend
将从光标到当前行末尾的部分包含到选定内容中	Alt-Shift-Right	selecttolineend
将从当前行开头到光标部分包含到选定内容中	Alt-Shift-Left	selecttolinestart
包含从光标到当前范围中下一匹配符号的内容	Ctrl-Shift-P	selecttomatching
将从光标向上到当前文件开头的部分包含到选定内容中	Ctrl-Shift-Home	selecttostart
将上一行包含到选定内容中	Shift-Up	selectup
将光标左侧下一单词包含到选定内容中	Ctrl-Shift-Left	selectwordleft
将光标右侧下一单词包含到选定内容中	Ctrl-Shift-Right	selectwordright
显示 Preferences (首选项) 选项卡	Ctrl-,	showSettingsMenu
清除所有之前的选择	Esc	singleSelection

描述	键绑定	命令
减小字体大小	Ctrl-- Ctrl-Shift-= Ctrl-Shift-+	smallerfont
如果已选择多行，则按排序顺序将其重新排列	F9	sortlines
在当前行末尾添加光标	Ctrl-Shift-L	splitIntoLines
在选定内容周围环绕块注释字符，如果已有块注释字符，则将其删除	Ctrl-Shift-/	toggleBlockComment
在选定的每一行开头添加行注释字符，如果已有行注释字符，则将其删除	Ctrl-/	togglecomment
折叠代码，如果已折叠，则取消折叠	Ctrl-Shift-[toggleFoldWidget
折叠父代码，如果已折叠，则取消折叠	Alt-F2	toggleParentFoldWidget
开始击键记录，如果已在记录，则停止	Ctrl-Q	toggleRecording
设置单词换行，如果已设置，则停止单词换行	Ctrl-Q	toggleWordWrap
将选定内容更改为全部小写	Ctrl-K Ctrl-L	tolowercase
将选定内容更改为全部大写	Ctrl-K Ctrl-U	touppercase
转置选定内容	Alt-X	transposeletters
展开选定代码	Ctrl-Shift-]	unfold
展开整个文件的代码折叠	Ctrl-K Ctrl-O Ctrl-K Ctrl-J	unfoldall

emmet

描述	键绑定	命令
计算简单数学表达式 (如 $2*4$ 或 $10/2$) 并输出结果	Shift-Ctrl-Y	emmet_evaluate_math_expression
将 CSS 等缩写扩展为 HTML、XML 或 CSS 代码, 具体取决于当前文件的语法	Ctrl-Alt-E	emmet_expand_abbreviation
使用制表键来遍历展开的 CSS 等缩写	Tab	emmet_expand_abbreviation_with_tab
转至下一段可编辑代码	Shift-Ctrl-.	emmet_select_next_item
转至前一段可编辑代码	Shift-Ctrl-,	emmet_select_previous_item
展开一个缩写, 然后将当前选定内容放入所生成片段的最后一个元素内	Shift-Ctrl-A	emmet_wrap_with_abbreviation

终端

描述	键绑定	命令
打开新的 Terminal (终端) 选项卡	Alt-T	openterminal
在编辑器和 Terminal (终端) 选项卡之间切换	Alt-S	switchterminal

运行和调试

描述	键绑定	命令
生成当前文件	F7 Ctrl-B	build
恢复当前暂停的流程	F8	resume
运行或调试当前应用程序	Ctrl-Shift-B	run
运行或调试上次运行的文件	F5	runlast
单步进入堆栈中的下一个函数	F11	stepinto
单步退出当前函数的范围	Shift-F11	stepout
跳过堆栈中的当前表达式	F10	stepover
停止运行或调试当前应用程序	Shift-F5	stop
停止生成当前文件	Ctrl-Break	stopbuild

用于 AWS Cloud9 集成开发环境 (IDE) 的命令参考

要在 AWS Cloud9 IDE 中运行命令，请执行以下操作：

1. 选择 Go (前往) 按钮 (放大镜) 以显示 Go (前往) 窗口。如果未显示 Go (前往) 按钮，请在菜单栏上选择 Window (窗口)、Go (前往)。
2. 在 Go to Anything (转到任意内容) 框中，开始键入命令组 (如代码编辑器) 的名称。一个组包含围绕公共主题或 IDE 功能组织的多个命令。
3. 在 Command (命令) 标题中，从组中选择要运行的特定命令。

可用命令组

命令组	描述
AWS	用于 AWS Toolkit 的命令

命令组	描述
Clipboard	用于复制和粘贴内容的命令
Code Editor	用于导航代码编辑器界面并与编辑器内容进行交互的命令
Emmet	用于使用用于 HTML 和 CSS 内容的 Emmet 工具包的命令
General	用于管理 IDE 配置和项目文件的其他命令
Panels	用于管理 IDE 界面中面板显示的命令
Run & Debug	用于在 AWS Cloud9 中运行和调试项目的命令
Tabs	用于管理 IDE 界面中面板显示和标签页导航的命令
Terminal	用于管理命令行终端的命令
Window	用于管理 IDE 窗口中窗格布局的命令

使用其他AWS服务

当使用 AWS Cloud9 时，您可以与 Amazon Lightsail、AWS CodeStar 和 AWS CodePipeline 紧密配合。本节中的主题如何执行此操作。

Important

AWS Toolkit 功能提供了一个方便的可视化界面，用于使用关键 AWS 服务，例如 AWS Lambda、AWS Serverless Application Model 和 Amazon S3。有关更多信息，请参阅 [AWS Toolkit](#)。

主题

- [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 Amazon Lightsail 实例](#)
- [在 AWS Cloud9 集成式开发环境 \(IDE\) 中使用 AWS CodeStar 项目](#)
- [使用与 Amazon Q 开发者合作 AWS Cloud9](#)
- [在 AWS Cloud9 集成开发环境 \(IDE\) 中使用 AWS CodePipeline](#)
- [与亚马逊合作 CodeCatalyst](#)
- [在 AWS Cloud9 集成式开发环境 \(IDE \) 中使用 AWS CDK](#)

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例

您可以使用 AWS Cloud9 IDE 在预配置了常用应用程序和框架的 Amazon Lightsail 实例上处理代码。它们包括 WordPress、LAMP (Linux、Apache、MySQL 和 PHP)、Node.js、NGINX、Drupal 和 Joomla。包括 Linux 发行版 (如 Amazon Linux、Ubuntu、Debian、FreeBSD 和 openSUSE)。

Lightsail 提供了一种方便且可快速设置的虚拟专用服务器解决方案。Lightsail 提供了计算、存储以及网络功能以及在云端部署和管理网站和 Web 应用程序的功能。您可以使用 Lightsail 以低廉、可预测的每月价格快速启动项目。有关更多信息，请参阅 [Amazon Lightsail Features](#)。

在本主题中，您创建并设置一个基于 Linux 且与 AWS Cloud9 兼容的 Lightsail 实例。然后，您创建一个 AWS Cloud9 SSH 开发环境并将其连接到该 Lightsail 实例。

Note

完成这些过程可能会对您的 AWS 账户 收费。其中包括可能对 Lightsail 等服务收取的费用。有关更多信息，请参阅 [Amazon Lightsail 定价](#)。

要创建并设置更高级的解决方案（包括具有 AWS Cloud9 IDE、源代码控制、构建、部署、虚拟服务器或无服务器资源等的工具链），请参阅 [Working with AWS CodeStar Projects](#)。

要使用 AWS Cloud9 IDE 处理运行 Amazon Linux 或 Ubuntu Server 且不包含示例代码的 Amazon EC2 实例，请参阅 [入门：基本教程](#)。

- [步骤 1：创建基于 Linux 的 Lightsail 实例](#)
- [步骤 2：设置实例以将其与 AWS Cloud9 一起使用](#)
- [步骤 3：创建并连接到 AWS Cloud9 SSH 开发环境](#)
- [步骤 4：使用 AWS Cloud9 IDE 更改实例上的代码](#)

步骤 1：创建基于 Linux 的 Lightsail 实例

在该步骤中，您使用 Lightsail 控制台创建一个在基于 Linux 的发行版中运行应用程序的 Amazon EC2 实例。该实例自动包含以下各项：

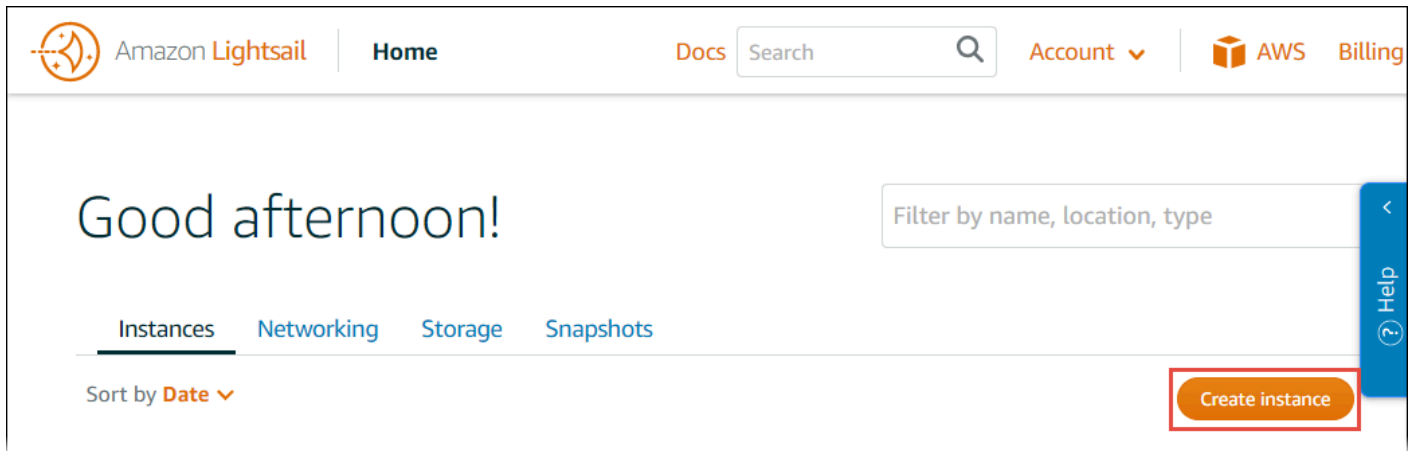
- 公有和私有 IP 地址。（您以后可以创建静态公有 IP。）
- 通过端口 22 使用 SSH、通过端口 80 使用 HTTP 以及通过端口 443 使用 HTTPS 访问该实例的权限。（您可以更改这些设置。）
- 数据块存储磁盘。（您以后可以附加额外的磁盘。）
- 内置系统报告。

在 Lightsail 控制台上，您可以稍后备份、重启、停止或删除该实例。

1. 打开并随后登录到 Lightsail 控制台 (<https://lightsail.aws.amazon.com>)。

我们建议您使用您 AWS 账户 中的 IAM 管理员用户的凭证登录。如果无法以 IAM 管理员用户身份登录，请与 AWS 账户 管理员联系。


2. 如果屏幕上出现提示，请在控制台中选择要使用的语言，然后选择 Save（保存）。
3. 如果屏幕上出现提示，请选择 Let's get started（我们开始吧）。
4. 在主页中，在已选择 Instances（实例）选项卡的情况下，选择 Create instance（创建实例）。




5. 对于 Instance location (实例位置)，确保该位置是要在其中创建实例且 AWS Cloud9 可用的 AWS 区域。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 [AWS Cloud9](#)。要更改 AWS 区域 和/或可用区，请选择 更改 AWS 区域和可用区，然后按照屏幕上的说明进行操作。
6. 对于 Pick a instance image (选择实例镜像)，Select a platform (选择平台) 中已选择了 Linux/Unix，并且 Select a blueprint (选择蓝图) 中已选择了 Apps + OS (应用 + 操作系统)，选择一个蓝图。

Pick your instance image ?

Select a platform



Linux/Unix
16 blueprints




Microsoft Windows
3 blueprints


Select a blueprint

Apps + OS


OS Only




WordPress
4.8.1




LAMP Stack
5.6.31




Node.js
8.4.0




Joomla
3.7.5




Magento
2.1.8-1




MEAN
3.4.7




Drupal
8.3.7-1




GitLab CE
9.5.0



Redmine
3.4.2-2



Nginx
1.12.1



Plesk Hosting Stack on Ubuntu
17.5.3

i Note

如果要创建没有应用程序的实例，请选择 OS Only（仅限操作系统）而不是 Apps + OS（应用 + 操作系统），然后选择一个发行版。

要了解可用的选项，请参阅 Lightsail 网站上的选择[选择 Amazon Lightsail 实例镜像](#)。

7. 对于 Choose your instance plan（选择实例计划），请选择一个计划，或保留选定的默认计划。
8. 对于 Name your instance（为您的实例命名），输入实例的名称，或保留建议的原定设置名称。
9. 对于实例数，请输入要创建的实例数，或保留原定设置值，即单个实例（x 1）。
10. 选择创建。

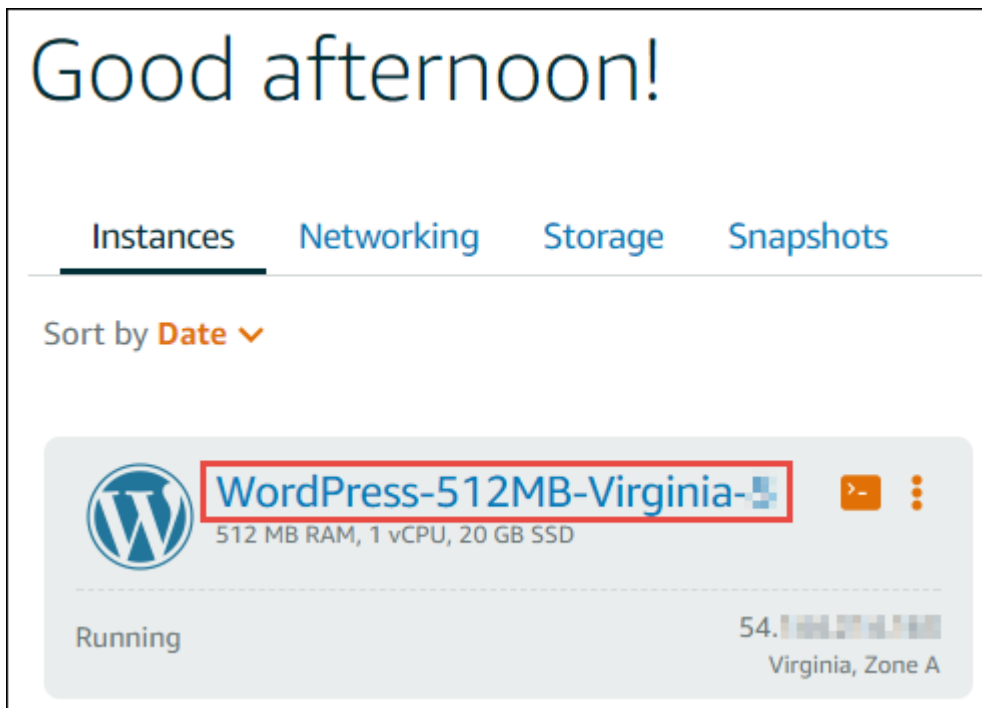
步骤 2：设置实例以将其与 AWS Cloud9 一起使用

在该步骤中，您连接到运行的实例，然后对其进行设置，以便 AWS Cloud9 以后可以使用该实例。

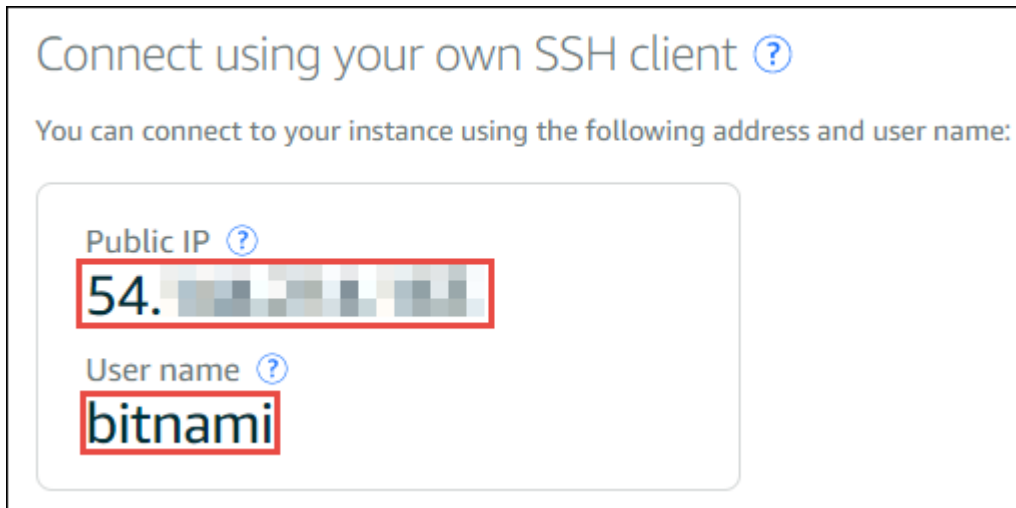
Note

以下说明假定您在上一步中选择了 Apps + OS (应用 + 操作系统)。如果选择了 OS Only (仅限操作系统) 和 Ubuntu 以外的发行版，您可能需要相应地修改以下说明。

1. 在上一步的 Lightsail 控制台仍处于打开的情况下，在 Instances (实例) 选项卡上该实例的卡中，选择该实例的名称。



2. 在 Connect (连接) 选项卡上，对于 Connect using your own SSH client (使用您自己的 SSH 客户端连接)，记下 Public IP (公有 IP) 和 User name (用户名) 值，因为您以后要用到它们。



3. 选择 **Connect using SSH** (使用 SSH 连接)。
4. 确保该实例具有最新的系统更新。为此，请在显示的终端会话中运行 **sudo apt update** 命令。
5. 检查是否安装了 Python，如果已安装，则检查版本是否为 2.7。要检查版本，请运行 **python --version** 命令，并记下显示的版本号。如果未显示版本号或版本不是 2.7，请运行 **sudo apt install -y python-minimal** 命令以在该实例上安装 Python 2.7。
6. 检查是否安装了 Node.js，如果已安装，请检查版本是否为 0.6.16 或更高版本。要检查版本，请运行 **node --version** 命令，并记下显示的版本号。如果未显示版本号或版本不是 0.6.16 或更高版本，我们建议您使用 Node Version Manager (nvm) 在该实例上安装 Node.js。

为此，请按以下顺序，运行以下命令（每次运行一个）以更新该实例，在实例上安装 Node Version Manager (nvm)，在实例上激活 nvm，然后在实例上安装最新版本的 Node.js。

```
sudo apt update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
. ~/.bashrc
nvm install node
```

7. 运行 **which node** 命令，并记下显示的值。您将在以后需要它。

Note

如果命令 **which node** 的输出类似于 `/usr/sbin/node`，则 AWS Cloud9 在该路径中找不到 Node.js。应转而使用 nvm 安装 Node.js，如本过程的上一步中所述。然后，再次运行 **which node** 命令，并记下显示的新值。

8. 在实例中 [下载并运行 AWS Cloud9 安装程序](#)。

步骤 3：创建并连接到 AWS Cloud9 SSH 开发环境

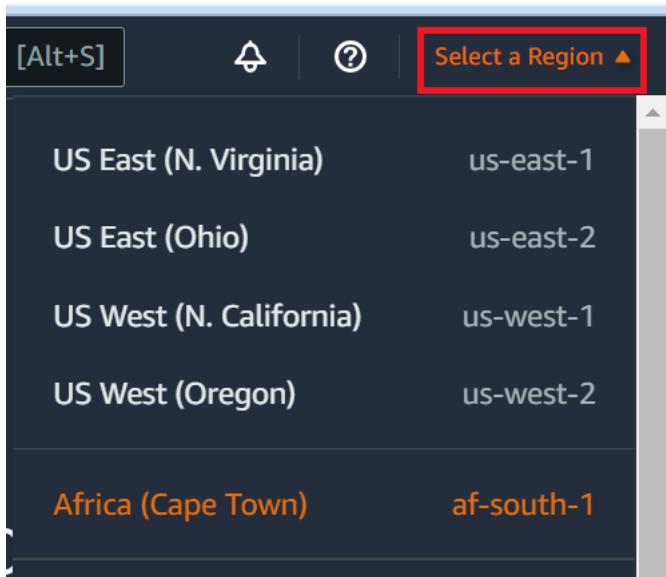
在该步骤中，您使用 AWS Cloud9 控制台和实例的终端创建一个 SSH 环境，然后将该环境连接到运行的实例。

1. 在上一步的终端会话仍处于打开的状态下，登录到 AWS Cloud9 控制台，如下所示：
 - 如果您是使用 AWS 账户 的唯一一个人用户，或者您是单个 AWS 账户 中的 IAM 用户，请转到 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的组织使用 AWS IAM Identity Center，请联系 AWS 账户 管理员以获取登录说明。

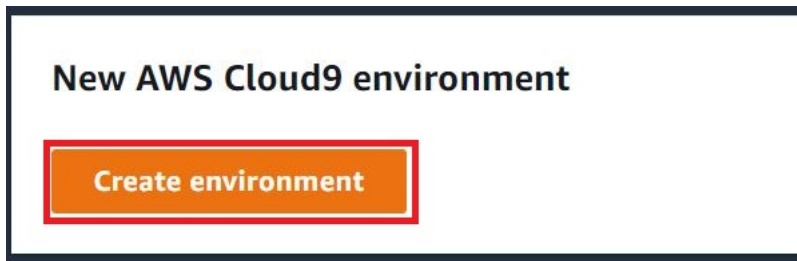
Note

对于该步骤，您同时使用两个不同的 AWS 服务。现在，假设您以 IAM 管理员用户身份登录到 Lightsail 控制台，但希望用另一个实体拥有新的 SSH 环境。在这种情况下，我们建议您打开不同的 Web 浏览器，然后以该实体身份登录到 AWS Cloud9 控制台。

2. 在 AWS Cloud9 控制台中，选择与您在其中创建该实例的区域匹配的 AWS 区域。



3. 如果显示欢迎页面，则为 New AWS Cloud9 environment (新建 Amazon Cloud9 环境) 选择 Create environment (创建环境)。如未显示，则选择 Create environment (创建环境)。



或者：



4. 在 Name environment (命名环境) 页面上，在 Name (名称) 中输入环境的名称。
5. 在描述字段中向您的环境添加描述。
6. 对于环境类型，选择现有计算。这很重要，因为您需要选择此选项才能显示用户和主机选项。
7. 对于 User (用户)，请输入您之前记下的 User name (用户名) 值。
8. 对于 Host (主机)，请输入您之前记下的 Public IP (公有 IP) 值。
9. 对于 Port (端口)，请保留默认值 22。
10. 展开其他详细信息。
11. 对于 Environment path (环境路径)，输入从登录后 AWS Cloud9 开始的路径，即 ~/.。这是用户的主目录的根目录。
12. 对于 Node.js binary path (Node.js 二进制文件路径)，请输入您之前记下的 **which node** 命令值。
13. 将 SSH jump host (SSH 跳转主机) 保留空白。
14. 将 AWS Cloud9 为该环境创建的公有 SSH 密钥存储在系统剪贴板中。为此，请选择 Copy key to clipboard (将密钥复制到剪贴板)。

Note

要查看复制的公有 SSH 密钥值，请展开 View public SSH key (查看公有 SSH 密钥)。

15. 将刚复制的公有 SSH 密钥值保存到实例中。为此，请使用 vi (常用文本编辑器)，该编辑器已安装在实例上：
 - a. 在实例的终端会话中，运行 **vi ~/.ssh/authorized_keys** 命令。
 - b. 在显示的 vi 编辑器中，转到文件末尾，然后切换为插入模式。为此，请按 I，然后按 A。[将在 vi 编辑器底部显示 -- INSERT -- (-- 插入 --)。]

- c. 按两次 Enter 以在文件末尾添加两个回车符。
- d. 将系统剪贴板内容 (包含刚复制的公有 SSH 密钥值) 粘贴到终端会话剪贴板中。为此, 请在终端会话窗口底角选择剪贴板按钮, 然后将系统剪贴板内容粘贴到框中。



- e. 将终端会话剪贴板内容粘贴到 vi 编辑器中。为此, 请在 vi 编辑器中的插入点处按 Ctrl + Shift + V。
- f. 保存该文件。为此, 请按 Esc 以进入命令模式。(-- INSERT -- (-- 插入 --) 将从 vi 编辑器底部消失。) 键入 :wq (以 write 文件, 然后 quit vi 编辑器), 然后按 Enter。

16 返回到 AWS Cloud9 控制台, 选择 Next step (下一步)。

17 在 Review choices (审核选项) 页面上, 选择 Create environment (创建环境)。等待 AWS Cloud9 创建环境, 随后会显示该环境的 AWS Cloud9 IDE。这个过程可能需要几分钟。

AWS Cloud9 创建环境后, 会显示该环境的 AWS Cloud9 IDE。

如果 AWS Cloud9 在至少 5 分钟后未显示 IDE, 可能是您的 Web 浏览器、您的 AWS 访问权限、实例或关联的 Virtual Private Cloud (VPC) 出现问题。有关可能的解决方案, 请参阅[故障排除](#)中的无法打开环境。

步骤 4 : 使用 AWS Cloud9 IDE 更改实例上的代码

在为新的环境显示 IDE 后, 您可以使用 IDE 中的终端会话, 而不是 Lightsail 终端会话。IDE 提供丰富的代码编辑体验以及对多种编程语言和运行时调试程序的支持。IDE 还包括颜色主题、快捷键绑定、编程语言特定的语法颜色以及代码格式。

要了解如何使用 IDE, 请参阅 [了解 AWS Cloud9 IDE](#)。

要了解如何更改您的实例上的代码, 我们建议使用以下资源 :

- All : Lightsail 网站上的[获取“由 Bitnami 支持的”Lightsail 映像的应用程序密码](#)
- Drupal : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiDrupal](#) , 以及 Drupal 网站上的[教程和网站使用技巧](#)
- GitLab CE : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiGitLab CE](#) , 以及 GitLab 网站上的[GitLab 文档](#)
- Joomla : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiJoomla!](#) , 以及 Joomla! 网站上的[Joomla! 入门](#)
- LAMP 堆栈 : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiLAMP](#)
- Magento : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiMagento](#) , 以及 Magento 网站上的[Magento 用户指南](#)
- MEAN : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiMEAN](#)
- NGINX : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiNGINX](#) , 以及 NGINX 网站上的[NGINX Wiki](#)
- Node.js : Bitnami 网站上的[适用于 AWS Cloud 的 BitnamiNode.js](#) , 以及 Node.js 网站上的[入门指南](#)
- Ubuntu 上的 Plesk 托管堆栈 : [在 Amazon Lightsail 上设置和配置 Plesk](#)。
- Redmine : Bitnami 网站上的[适用于 AWS Cloud 的 Bitnami Redmine](#) , 以及 Redmine 网站上的[入门](#)
- WordPress : Lightsail 网站上的[Amazon Lightsail 实例中的 WordPress 使用入门](#)以及 Bitnami 网站上的[适用于 AWS Cloud 的 Bitnami WordPress](#)

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CodeStar 项目

您可以使用 AWS Cloud9 IDE 处理 AWS CodeStar 项目中的代码。

AWS CodeStar 是一项基于云的服务，用于在 AWS 上创建、管理和使用软件开发项目。您可以使用 AWS CodeStar 项目在 AWS 上快速开发、构建和部署应用程序。AWS CodeStar 项目会为您的项目开发工具链创建和集成 AWS 服务。根据您的选择的 AWS CodeStar 项目模板，该工具链可能包括源代码控制、生成、部署、虚拟服务器或无服务器资源等。有关更多信息，请参阅[AWS CodeStar 用户指南](#)。

Note

完成这些过程可能会对您的 AWS 账户 收费。其中可能包括对各项服务收费，例如 Amazon EC2、AWS CodeStar 以及 AWS CodeStar 支持的 AWS 服务。有关更多信息，请参阅[Amazon EC2 定价](#)、[AWS CodeStar 定价](#)和[云服务定价](#)。

要使用 AWS Cloud9 IDE 处理新启动的 Amazon EC2 实例，该实例已预配置了常用应用程序或框架（例如 WordPress、MySQL、PHP、Node.js、NGINX、Drupal 或 Joomla）或 Linux 发行版（如 Ubuntu、Debian、FreeBSD 或 openSUSE），您可以将 Amazon Lightsail 与 AWS Cloud9 结合使用。为此，请跳过本主题的其余部分，直接参阅[使用 Amazon Lightsail 实例](#)。

要使用 AWS Cloud9 IDE 处理新启动且运行不包含示例代码的 Amazon Linux 的 Amazon EC2 实例，请跳过本主题的其余部分，而参阅[入门：基本教程](#)。

- [第 1 步：准备好使用 AWS CodeStar 项目](#)
- [第 2 步：在 AWS CodeStar 中创建项目](#)
- [第 3 步：创建 AWS Cloud9 开发环境并将其连接到项目](#)

第 1 步：准备好使用 AWS CodeStar 项目

在此步骤中，创建一个 AWS CodeStar 服务角色和 Amazon EC2 密钥对，以便可以开始创建和使用 AWS CodeStar 项目。

如果您之前使用过 AWS CodeStar，请向前跳至[第 2 步：在 AWS CodeStar 中创建项目](#)。

在此步骤中，按照 AWS CodeStar 用户指南中的[设置 AWS CodeStar](#)的说明进行操作。请不要按照上述说明创建新的 AWS 账户、IAM 用户或 IAM 组。请使用您在 [AWS Cloud9 团队设置](#)中创建或标识的对应账户、用户或组。完成按说明操作后，返回到本主题。

第 2 步：在 AWS CodeStar 中创建项目

在此步骤中，在 AWS CodeStar 中创建项目。

如果您在 AWS CodeStar 中已经有要使用的项目，请向前跳至[第 3 步：创建 AWS Cloud9 开发环境并将其连接到项目](#)。

在此步骤中，请按照 AWS CodeStar 用户指南中的[在 AWS CodeStar 中创建项目](#)的说明操作。在 AWS CodeStar 创建项目向导中，当您进入 Set up tools (设置工具) 页面或 Connect to your source repository (连接到源存储库) 页面时，选择 Skip (跳过)，然后返回到本主题。

步骤 3：创建 AWS Cloud9 开发环境并将其连接到项目

在此步骤中，您在 AWS CodeStar 或 AWS Cloud9 控制台中创建 AWS Cloud9 开发环境。然后，将新的环境连接到 AWS CodeStar 项目。

在此步骤中，请按照以下系列说明之一进行操作。具体取决于您要使用的 AWS Cloud9 开发环境类型以及 AWS CodeStar 项目存储其代码的存储库类型。

环境类型	存储库类型	说明
EC2 环境	CodeCommit	AWS CodeStar 用户指南中的 为项目创建 AWS Cloud9 环境
SSH 环境	CodeCommit	AWS CodeCommit 示例
EC2 或 SSH 环境	GitHub	AWS CodeStar 用户指南中的 将 GitHub 与 AWS Cloud9 结合使用

使用与 Amazon Q 开发者合作 AWS Cloud9

什么是 Amazon Q ?

Amazon Q Developer 是一款生成式人工智能 (AI) 驱动的对话助手，可以帮助您理解、构建、扩展和操作 AWS 应用程序。在集成 AWS 编码环境中，Amazon Q 可以根据开发者的代码以及他们的自然语言注释生成代码推荐。Amazon Q 对 Java、JavaScript、TypeScript、Python、C#、Go、PHP、Rust、Kotlin、以及基础设施即代码 (IaC) 语言 JSON (CloudFormation)、SQL、YAML (CloudFormation)、HCL (Terraform)、AWS CloudFormation) 和 CDK (AWS CloudFormation Typescript、Python) 的支持最多。它还支持 Ruby、C++、Shell、和生成代码 Scala。有关 Amazon Q 如何与 IDE 集成 AWS Cloud9 并在 AWS Cloud9 IDE 中显示代码建议的[示例](#)，请参阅[Amazon Q 开发者用户指南中的代码示例](#)。

有关将 Amazon Q 与配合使用的更多信息 AWS Cloud9，请参阅[Amazon Q 开发者用户指南](#)。

AWS Identity and Access Management 的权限 AWS Cloud9

要让 Amazon Q 在 AWS Cloud9 控制台中提供建议，您必须为您的 IAM 用户或角色启用正确的 IAM 权限。您必须添加 `codewhisperer:GenerateRecommendations` 权限，如以下 IAM policy 示例中所概述的那样：

Note

前codewhisperer缀是与 Amazon Q Developer 合并的服务的旧名称。有关更多信息，请参阅 [Amazon Q 开发者重命名-变更摘要](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonQDeveloperPermissions",
      "Effect": "Allow",
      "Action": ["codewhisperer:GenerateRecommendations"],
      "Resource": "*"
    }
  ]
}
```

最佳实践是使用 IAM policy 向 IAM 主体授予限制性权限。

在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline

您可以使用 AWS Cloud9 IDE 来处理与 AWS CodePipeline 兼容的存储库中的源代码。

CodePipeline 是一种持续交付服务，可用于建模、可视化和自动执行发布软件以及进行持续更改所需的步骤。您可以使用 CodePipeline 快速对软件发布过程的不同阶段进行建模和配置。有关更多信息，请参阅 [AWS CodePipeline 用户指南](#)。

Note

完成这些步骤可能会对您的 AWS 账户收费。其中包括可能的服务费用，例如 Amazon EC2、CodePipeline、Amazon S3 和 CodePipeline 支持的 AWS 服务。有关更多信息，请参阅 [Amazon EC2 定价](#)、[AWS CodePipeline 定价](#)、[Amazon S3 定价](#)和[云服务定价](#)。除了管道之外，AWS CodeStar 还提供了额外功能，例如项目模板、控制面板和团队。要使用 AWS CodeStar 而不是 CodePipeline，请跳过本主题的剩余部分，直接参阅[使用 AWS CodeCommit 项目](#)。

- [步骤 1：创建或确定源代码存储库](#)

- [步骤 2：创建 AWS Cloud9 开发环境，将其连接到代码存储库并上传代码](#)
- [步骤 3：准备使用 AWS CodePipeline](#)
- [步骤 4：在 AWS CodePipeline 中创建管道](#)

步骤 1：创建或确定源代码存储库

在此步骤中，您将创建或确定与 CodePipeline 兼容的源代码存储库。

在本主题的后文中，您会将软件的源代码上传到该存储库。CodePipeline 将使用您另外构建的相关管道，生成、测试和部署上传到该存储库中的源代码。

您的源代码存储库必须是 CodePipeline 支持的以下存储库类型之一：

- AWS CodeCommit，如果您已有要使用的 CodeCommit 存储库，请向前跳到[步骤 2：创建 AWS Cloud9 开发环境，将其连接到代码存储库并上传代码](#)。否则，要使用 CodeCommit，请按顺序按照 AWS CodeCommit 示例中的以下说明操作，然后返回本主题：
 - [步骤 1：设置具有所需访问权限的 IAM 组](#)
 - [步骤 2：在 AWS CodeCommit 中创建存储库](#)
- Amazon S3。如果您已有要使用的 Amazon S3 存储桶，请向前跳到[步骤 2：创建 AWS Cloud9 开发环境，将其连接到代码存储库并上传代码](#)。否则，要使用 Amazon S3，请按此顺序根据 Amazon Simple Storage Service 用户指南中的这些说明进行操作，然后返回本主题：
 - [注册 Amazon S3](#)
 - [创建存储桶](#)
- GitHub。如果您已经在 GitHub 中有一个存储库，则可以使用 [Git 面板](#) 界面在开发环境上克隆一个本地副本。如果您还没有在 GitHub 上设置账户或存储库，请参阅[相关文档](#)了解说明。

步骤 2：创建 AWS Cloud9 开发环境，将其连接到代码存储库并上传代码

在这一步中，您在 AWS Cloud9 控制台中创建 AWS Cloud9 开发环境。然后，将环境连接到 CodePipeline 将要使用的存储库。最后，为环境使用 AWS Cloud9 IDE 以便将源代码上传到存储库。

要创建环境，请按照[创建环境](#)中的说明操作，然后返回本主题。（如果您已有一个环境，可以使用该环境，无需创建新的环境。）

要将环境连接到存储库（并在存储库中没有源代码时上传源代码），请使用以下几组说明之一。您选择的说明取决于存储源代码的存储库类型。

存储库类型	说明
CodeCommit	<p>按照 AWS CodeCommit GitHub 示例 中的以下说明操作：</p> <ul style="list-style-type: none"> • 步骤 3：将您的环境连接到远程存储库 • 步骤 4：将远程存储库克隆到您的环境 • 步骤 5：将文件添加到存储库，在此步骤中替换为您自己的源代码
Amazon S3	<ul style="list-style-type: none"> • 在环境中安装和配置 AWS CLI 或 AWS CloudShell，如 AWS CLI 和 AWS CloudShell 示例 中所述。 • 要将源代码上传到存储桶，请在环境中使用 AWS CLI 或 AWS CloudShell 运行 aws s3 cp 命令。（对于 AWS CloudShell，您可以从命令中删除 aws。）
GitHub	<p>您可以克隆托管在 GitHub 上的存储库，并通过使用 Git 面板 界面与其互动。</p>

将环境连接到存储库之后，每当您将源代码更改从 AWS Cloud9 IDE 推送到存储库时，CodePipeline 就会自动通过相关管道发送这些更改以执行生成、测试和部署操作。您将在本主题的后文中创建相关管道。

步骤 3：准备使用 AWS CodePipeline

在此步骤中，您将特定 AWS 托管式策略附加到在[团队设置](#)中创建或确定的 IAM 组。这使得组用户可以在 CodePipeline 中开始创建和使用管道。

如果您之前使用过 CodePipeline，请向前跳到[步骤 4：在 AWS CodePipeline 中创建管道](#)。

对于此步骤，请按照 AWS CodePipeline 用户指南中的[步骤 3：使用 IAM 托管式策略将 AWS CodePipeline 权限分配给 IAM 用户](#)的说明操作，然后返回此主题。

步骤 4：在 AWS CodePipeline 中创建管道

在此步骤中，您使用在本主题上文中创建或确定的存储库，在 CodePipeline 中创建管道。

对于此步骤，请按照 AWS CodePipeline 用户指南中的[在 AWS CodePipeline 中创建管道](#)的说明操作。

创建管道后，CodePipeline 会通过管道发送存储库中的当前源代码版本以执行生成、测试和部署操作。然后，每当您将源代码更改从 AWS Cloud9 IDE 推送到存储库时，CodePipeline 就会自动通过管道发送这些更改以执行生成、测试和部署操作。

要查看管道，请按照 AWS CodePipeline 用户指南中的[查看 AWS CodePipeline 中的管道详细信息和历史记录](#)的说明操作。

与亚马逊合作 CodeCatalyst

Amazon CodeCatalyst 是一个面向软件开发团队的基于云的协作空间。CodeCatalyst 是一个使用持续集成/交付 (CI/CD) 工具进行工作、协作编写代码以及构建、测试和部署应用程序的统一场所。您可以通过将您的空间连接到您的 CodeCatalyst 空间来将 AWS 资源与您的 AWS 账户项目联系起来。您还可以 CodeCatalyst 使用快速、自信地交付软件。有关的更多信息 CodeCatalyst，请参阅[Amazon 是什么 CodeCatalyst？](#)在《亚马逊 CodeCatalyst 指南》中。

开发环境是基于云的开发环境，你可以在中使用它 CodeCatalyst 来处理存储在项目源存储库中的代码。你可以在中创建开发环境 CodeCatalyst。然后，在那里你可以使用支持的集成开发环境 (IDE) 在特定项目中 CodeCatalyst 处理代码。您也可以创建一个空的开发环境，从第三方存储库中克隆代码，以便在支持的 IDE 中工作。

用于在 CodeCatalyst 控制台中访问开发环境的 AWS Cloud9 IDE 与在其上运行的 AWS Cloud9 IDE 不同 AWS。在 CodeCatalyst AWS Cloud9 IDE 中，您将自动登录 CodeCatalyst 并可以使用 IDE 中的 aws-explorer 选项访问该服务。有关该 AWS 工具包的更多信息，请参阅 AWS Cloud9 指南 AWS Cloud9 中的[AWS 工具包](#)。

主题

- [开始使用](#)
- [在 Amazon 中复制 AWS Cloud9 代码资源 CodeCatalyst](#)
- [使用复制工具](#)
- [有关复制过程的常见问题](#)
- [Amazon 中的开发环境 CodeCatalyst](#)

开始使用

本节概述了如何开始使用 CodeCatalyst。本节中的主题涵盖如何 AWS Cloud9 在 Amazon 中使用 CodeCatalyst 以及如何在中复制您的 AWS Cloud9 环境 CodeCatalyst。后面的主题还将详细介绍如何创建 CodeCatalyst 开发环境以及如何使用 AWS Cloud9 IDE 访问您的开发环境。

AWS 工具包是特定于 IDE 的软件开发套件 (SDK)，可快速访问 AWS Cloud 帐户、服务和资源。通过您在 AWS Toolkit 中的 CodeCatalyst 帐户，您可以在便捷的界面中查看、编辑和管理您的 CodeCatalyst 开发环境、空间和项目。要详细了解通过 AWS Toolkits 提供的 AWS Cloud 服务和功能，请参阅[什么是 AWS Toolkit for Visual Studio Code ?](#)，[AWS 工具包的用途](#) AWS Cloud9，以及[什么是 AWS Toolkit for JetBrains ?](#)。AWS Toolkit for JetBrains指南@@ [是什么](#)。

要 AWS Cloud9 在 IDE 中使用 CodeCatalyst，您必须拥有在 CodeCatalyst 控制台中创建的现有空间、项目和开发环境。

Note

不要在 AWS Cloud9 IDE 的文件系统中为的同名文件夹中创建名为 `projects` 的子文件夹。CodeCatalyst 如果这样做，您将无法访问此目录中的任何文件。此问题会影响文件路径 `/projects/projects`。诸如 `/test/projects` 和 `/projects/test/projects` 之类的文件路径不受此问题的影响。这是一个已知问题，仅影响 AWS Cloud9 IDE 文件资源管理器。

Note

当前无法使用 AWS Cloud9 IDE 的文件系统在同名文件夹中创建名为 `projects` 的子文件夹。CodeCatalyst 您将无法从 AWS Cloud9 IDE 文件资源管理器访问此目录中的任何文件，但可以使用命令行访问它们。请使用其他文件夹名称。此问题仅影响文件路径 `/projects/projects`，类似于 `/test/projects` 和 `/projects/test/projects` 的文件路径应能正常使用。这是一个已知问题，仅影响 AWS Cloud9 IDE 文件资源管理器。

在 Amazon 中复制 AWS Cloud9 代码资源 CodeCatalyst

AWS Cloud9 in CodeCatalyst 为与之交互提供了完全托管的体验 AWS Cloud9。您可以在 Amazon 中手动复制您当前的 AWS Cloud9 代码资源 CodeCatalyst。该操作过程将在以下各节详细介绍。要移动代码资源并复制它们，请在其中创建一个空间 CodeCatalyst。空间代表的是您的公司、部门或团体。

您需要创建空间来添加项目、成员以及您在中创建的关联云资源 CodeCatalyst。当用户接受项目邀请时，CodeCatalyst 会自动将其添加到空间。具有空间管理员角色的用户可以管理空间。

在此空间中，您可以创建项目并添加源存储库。项目是支持开发团队和任务 CodeCatalyst 的协作空间。创建项目后，您可以添加、更新或删除资源。您还可以自定义项目控制面板，并监控团队的工作进度。一个空间内可以有多个项目。源存储库的可添加数量取决于您已在 AWS Cloud9 环境中使用的存储库数量。创建此项目并添加了适用的源存储库后，您可能需要返回到您的 AWS Cloud9 环境并将环境数据复制到中的这些新存储库中 CodeCatalyst。您需要进行的操作取决于您在 AWS Cloud9 中所拥有的源存储库类型。

创建空间、项目和源存储库后，可以在开发环境中 CodeCatalyst 启动您的环境。AWS Cloud9 开发环境是云端开发环境。您可以在中使用开发环境 CodeCatalyst 来处理存储在项目源存储库中的代码。您还可以在中创建开发环境 CodeCatalyst，以便在具有受支持的集成开发环境 (IDE) 的特定项目开发环境中处理代码。

您也可以 CodeCatalyst 使用复制工具将当前的 AWS Cloud9 代码资源复制到。这是您下载并在您的 AWS Cloud9 环境中运行的工具。如果您已经注册并创建了空间 CodeCatalyst，则该工具会自动在此空间中创建一个项目，并将您的代码资源复制到中的新存储库中 CodeCatalyst。这一过程与手动复制过程类似。这取决于您在 AWS Cloud9 中所拥有的源存储库类型。例如，如果您有 GitHub 存储库，则仍需要使用 CodeCatalyst 控制台中的 GitHub 扩展程序来复制这些存储库。

- [第 1 步。注册 Amazon CodeCatalyst 并创建空间](#)
- [第 2 步 在空间中创建项目](#)
- [第 3 步 在项目中创建一个源存储库](#)
- [第 4 步 将您的 AWS Cloud9 代码资源复制到中的源存储库 CodeCatalyst](#)
- [第 5 步。在 CodeCatalyst 使用中创建开发环境 AWS Cloud9](#)

第 1 步。注册 Amazon CodeCatalyst 并创建空间

CodeCatalyst 无需邀请您加入现有空间或项目，即可注册 Amazon。注册后便会自动创建一个空间和项目。您可以输入您使用的现有 AWS 账户 ID AWS Cloud9。这同样 AWS 账户 可以用于计费。有关如何查找您的 AWS 账户 身份的信息，请参阅[您的 AWS 账户 身份证及其别名](#)。按照此步骤注册您的 Amazon CodeCatalyst 个人资料、创建空间并为您的空间添加账户。

以新用户身份注册

1. 打开[CodeCatalyst 控制台](#)。

2. 在欢迎页中，选择注册。

此时将显示“创建您的 AWS 建造者 ID”页面。AWS Builder ID 是您创建用于登录的身份。此 ID 与 AWS 账户 ID 不同。要了解有关 AWS 生成器 ID 的更多信息，请参阅《AWS 登录用户指南》中的[AWS 生成器 ID 和其他 AWS 凭证](#)。

3. 在您的电子邮件地址中，输入您要关联的电子邮件地址 CodeCatalyst。然后选择下一步。
4. 在“您的姓名”中，输入要在使用 AWS 建筑商 ID 的应用程序中显示的名字和姓氏。

这个名字是你的 AWS Builder ID 个人资料名称。您之后可以在需要时更改这个名称。

选择下一步。此时将显示电子邮件验证页面。验证码将发送到您指定的电子邮件中。

5. 在验证码中，输入您收到的验证码，然后选择验证。

如果您在 5 分钟后仍未收到验证码，也没能在垃圾邮件文件夹中找到该验证码，请选择重新发送验证码。

6. 验证完成后，输入密码并选择确认密码。

选中确认您已阅读并确认 AWS 客户协议和 AWS 服务条款的复选框，然后选择创建我的个人资料。

7. 在创建您的别名页面上，输入要使用的别名 CodeCatalyst。其他 CodeCatalyst 用户将在评论和拉取请求中使用这个别名给 @mention you。您的 CodeCatalyst 个人资料将包含您在 AWS 建造者 ID 中的全名和您的 CodeCatalyst 别名。您无法更改您的 CodeCatalyst 别名。

您的全名和别名将显示在中的不同区域 CodeCatalyst。例如，您的资料名称会出现在您的活动源中，但项目成员会使用您的别名来 @mention 提及您。

选择创建别名。页面将更新，并显示创建您的空间部分。

8. 在空间名称中，输入空间名称，然后选择下一步。

您无法更改空间名称。

9. 在 AWS 账户 ID 中，关联您要连接到空间的账户的十二位 ID。

在 AWS 账户 验证令牌中，复制生成的令牌 ID。令牌会自动复制。但是，您可能需要在批准 AWS 连接请求时将其存储。

10. 选择“验证”AWS。
11. “验证 Amazon CodeCatalyst 空间”页面将在中打开 AWS Management Console。

这是 Amazon CodeCatalyst Spaces 页面。您可能需要登录才能访问该页面。

要访问该页面，请登录中的 Amazon CodeCatalyst Spaces [AWS Management Console](#)。

中的验证令牌字段会自动填充中生成的令牌 CodeCatalyst。AWS Management Console

12. 选择验证空间。

系统将显示账户验证成功消息，表示该账户已添加到空间。

默认情况下，您将使用 CodeCatalyst 免费套餐。如果要更改，请选择要启用标准套餐或为此空间添加 IAM 角色，请查看空间详细信息。

有关 CodeCatalyst 定价套餐的更多信息，请参阅 [Amazon CodeCatalyst -定价](#)。

CodeCatalyst 空间详情页面将在中打开 AWS Management Console。这是 Amazon CodeCatalyst Spaces 页面。您可能需要登录才能访问该页面。

13. 选择“前往[亚马逊](#)” CodeCatalyst。

14. 在的创建页面上 CodeCatalyst，选择创建空间。

创建空间时会显示一条状态消息。创建空间后，将 CodeCatalyst 打开您的空间页面。该视图默认为项目选项卡。

Note

如果显示权限错误或横幅，请刷新页面并尝试再次查看该页面。

注册 CodeCatalyst 并创建空间后，复制过程的下一步是在该空间中创建一个项目。

第 2 步 在空间中创建项目

以下步骤概括了如何在上一步中创建的空间中创建一个空项目。通过此项目，您可以在以后手动添加所需的资源。在创建项目之前，您必须拥有空间管理员 角色，并且必须加入要创建项目的空间。创建空间时，CodeCatalyst 会自动为您分配空间管理员角色。空间管理员角色是最强大的角色 CodeCatalyst。有关此角色及其权限的更多信息，请参阅 [Space administrator role](#)。

创建空项目

1. 导航到要在其中创建项目的空间。
2. 在空间控制面板上，选择创建项目。
3. 选择从头开始。

4. 在为项目命名下，输入要分配给项目的名称。该名称在空间内必须是唯一的。
5. 选择创建项目。

创建项目后，复制过程的下一步是创建一个或多个源存储库。

第 3 步 在项目中创建一个源存储库

在您刚刚创建的项目中，您需要创建一个源存储库。此存储库包含一个文件，即 README.md 文件，您可以随时编辑或删除该文件。根据您在创建源存储库时所做的选择，它可能还包含一个 .gitignore 文件。

创建源存储库

1. 打开[CodeCatalyst控制台](#)。
2. 导航到您的项目。
3. 在导航窗格中，选择代码，然后选择源存储库。
4. 选择添加存储库，然后选择创建存储库。
5. 在存储库名称中，输入存储库的名称。

一个项目中的存储库名称必须唯一。有关存储库名称要求的更多信息，请参阅[中的源存储库配额 CodeCatalyst](#)。

6. (可选) 在描述中，添加对存储库的描述，这有助于项目中的其他用户了解存储库的用途。
7. (可选) 为您计划推送的代码类型添加一个 .gitignore 文件。
8. 选择创建。

Note

CodeCatalyst 在创建存储库时将 README.md 文件添加到存储库中。CodeCatalyst 还会在名为 main 的默认分支中为存储库创建初始提交。您可以编辑或删除 README.md 文件，但无法更改或删除默认分支。

9. 要获取源存储库克隆 URL 和 PAT，请选择克隆存储库。
10. 要复制每个 HTTPS 克隆 URL 和 PAT，请选择复制。然后，将克隆 URL 和 PAT 存储在您可以检索到的位置。

克隆 URL 和 PAT 将在第 4 步中使用，并被引用为 CODECATALYST_SOURCE_REPO_CLONE_URL 和 CODECATALYST_PAT。

在项目中创建源存储库后，将您的 AWS Cloud9 数据复制到该源存储库。

第 4 步 将您的 AWS Cloud9 代码资源复制到中的源存储库 CodeCatalyst

您的 AWS Cloud9 环境中源存储库的类型决定了将代码资源存入您创建的 CodeCatalyst 源存储库所遵循的复制方法。这些方法如下所示：

- [在中使用GitHub存储库 AWS Cloud9](#)
- [使用非（例如 GitLab 或 Bitbucket）中的存储库 GitHubAWS Cloud9](#)
- [在 AWS Cloud9中使用空存储库](#)。使用此方法，您在 AWS Cloud9中不会使用任何源存储库。

在中使用GitHub存储库 CodeCatalyst

借助GitHub存储库扩展，您可以 AWS Cloud9 在 Amazon CodeCatalyst 项目中使用链接GitHub存储库。以下步骤概述了如何从 CodeCatalyst 目录中安装GitHub扩展。这些步骤还展示了如何将您的现有GitHub账户连接到您的 CodeCatalyst 空间，以及如何将GitHub存储库链接到您的 CodeCatalyst 项目。

此方法的第一步是从 CodeCatalyst 目录中安装GitHub存储库扩展。要安装该扩展，请执行以下步骤。

Important

作为安装和配置 Github 存储库扩展的一部分，您必须在您的 Github 账户中安装扩展。为此，您必须是Github账户管理员和 CodeCatalyst空间管理员。

第 1 步。从 CodeCatalyst 目录中安装扩展

1. 打开[CodeCatalyst控制台](#)。
2. 导航到您的空间。

Tip

如果您属于多个空间，则可以在顶部导航栏中选择要查看的空间。

3. 在搜索栏旁边的顶部菜单栏中选择目录图标，即可导航到目录。CodeCatalyst 您可以搜索 GitHub 存储库或根据类别筛选扩展。
4. （可选）要查看有关扩展的更多详细信息，例如与之关联的权限，请选择 GitHub 存储库扩展名。
5. 选择安装。查看扩展所需的权限，如果要继续，请再次选择安装。

安装 GitHub 存储库扩展后，您将进入 GitHub 存储库扩展详细信息页面，您可以在其中查看和管理连接的 GitHub 账户和关联的 GitHub 存储库。

安装 GitHub 存储库扩展程序后，将您的 GitHub 账户关联到您的 CodeCatalyst 空间。要连接您的 GitHub 账户，请执行以下步骤。

第 2 步。将您的 GitHub 账户关联到 CodeCatalyst

1. 在已连接的 Github 账户选项卡中，选择连接 GitHub 账户以转到外部的 GitHub 站点。
2. 使用您的 GitHub 凭证登录您的 GitHub 账户，然后选择要安装 Amazon 的账户 CodeCatalyst。
3. 选择是否允许 CodeCatalyst 访问所有当前和将来的存储库。或者，也可以选择要在中使用的特定 GitHub 存储库 CodeCatalyst。默认选项是 GitHub 空间中的所有 GitHub 存储库。
4. 查看授予的权限 CodeCatalyst，然后选择“安装”。

将您的 GitHub 账户关联到后 CodeCatalyst，您可以在 GitHub 存储库扩展详情页面的 GitHub 账户选项卡中查看关联的账户。

在中使用存储 GitHub 库的最后一步 CodeCatalyst 是将存储库链接到要使用它的 CodeCatalyst 项目。要将存储 GitHub 库链接到 CodeCatalyst 项目，请执行整个过程的步骤 3 中概述的以下步骤：

第 3 步。从 GitHub 存储库扩展详情页面将 GitHub 仓库链接到 CodeCatalyst 项目

1. 在已关联的 GitHub 存储库选项卡中，选择关联 GitHub 存储库。
2. 在 GitHub 账户中，选择包含要关联的存储库的 GitHub 账户。
3. 对于 GitHub 存储库，选择要链接到 CodeCatalyst 项目的存储库。
4. 对于 CodeCatalyst 项目，选择要将 GitHub 存储库链接到的 CodeCatalyst 项目。
5. 选择关联。

你的 CodeCatalyst 仓库现在应该有你刚才推送的更新文件和提交。现在，您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。有关开发环境的详细信息，请参阅 [中的开发环境 CodeCatalyst](#)。

现在，您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。 [步骤 5：使用中的创建开发环境中概述了执行此操作 AWS Cloud9 的步骤](#) CodeCatalyst

在中使用非 GitHub 存储库 CodeCatalyst

在 AWS Cloud9 使用非 GitHub 存储库复制环境 CodeCatalyst 之前，您需要在 Amazon 中创建个人访问令牌 (PAT)。下一节概述了如何创建此令牌。

在 Amazon 中创建个人访问令牌 CodeCatalyst

您可以在本地计算机上使用 Git 客户端或在集成式开发环境 (IDE) 中访问您在项目中创建的源存储库。要执行此操作，您必须输入应用程序特定的密码。您可以创建用于此目的的个人访问令牌 (PAT)。您创建的个人访问令牌 (PAT) 与您在所有空间和项目中的 CodeCatalyst 用户身份相关联。您可以查看已创建的 PAT 的名称和到期日期，也可以删除不再需要的 PAT。您只能在创建 PAT 密钥时对其进行复制。

创建个人访问令牌 (PAT)

1. 打开 CodeCatalyst 控制台，[网址为 https://codecatalyst.aws/](https://codecatalyst.aws/)。
2. 在顶部菜单栏中，选择您的个人资料徽章，然后选择我的设置。

Tip

您还可以找到您的用户个人资料。要执行此操作，请在项目或空间的成员页面上，从成员列表中选择您的姓名。

3. 在个人访问令牌下，选择创建。
4. 在 PAT 名称中，为您的个人访问令牌 (PAT) 输入一个描述性名称。
5. 在到期日期中，保留默认日期，或者选择日历图标以自定义日期。到期日期默认为自当前日期起之后的 1 年时间。
6. 选择创建。

Tip

当为源存储库选择克隆存储库时，也可以创建此令牌。

7. 要复制 PAT 密钥，请选择复制。将 PAT 密钥储存到您可以检索到的位置。

Important

PAT 密钥仅显示一次。关闭窗口后将无法再查看。如果您未将 PAT 密钥保存在安全位置，则可以创建新的密钥。

为源存储库创建 PAT 后，CodeCatalyst 通过在 AWS Cloud9 环境中添加远程存储库并将数据推送到该存储库，将您的数据从您的 AWS Cloud9 环境复制到，如以下部分所述。

在您的 AWS Cloud9 环境中添加远程存储库

假设你运行的存储库不是 GitHub 存储库。您可以在 AWS Cloud9 环境中添加远程存储库，并将数据推送到中的源存储库 CodeCatalyst。要完成此过程，请运行以下命令。

在 AWS Cloud9 IDE 中，添加一个远程存储库，该存储库指向您在中复制过程的步骤 3 中创建的源存储库 CodeCatalyst。将命令中的 CODECATALYST_SOURCE_REPO_CLONE_URL 替换为您在 [第 3 步的第 10 步中保存的克隆 URL](#)（[第 3 步 在项目中创建一个源存储库](#)）。

```
git remote add codecatalyst CODECATALYST_SOURCE_REPO_CLONE_URL
```

使用以下命令将新分支推送到源存储库。当系统提示您输入密码时，请使用您在[第 3 步的第 10 步中保存的 CODECATALYST_PAT](#)（[第 3 步 在项目中创建一个源存储库](#)）：

```
git checkout -b replication && git push codecatalyst replication
```

以下是预期的命令运行输出的示例：

```
Switched to a new branch 'replication'
Password for 'https://[aws-account-id@[aws-region]].codecatalyst.aws/v1/MySpace222581768915/Replication/Repository':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 982 bytes | 122.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://[aws-account-id].codecatalyst.aws/v1/MySpace222581768915/Replication/Repository
* [new branch] replication # replication
```

此分支可在您在中创建的源存储库中找到 CodeCatalyst。您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。有关开发环境的更多信息，请参阅[中的开发环境 CodeCatalyst](#)。

现在，您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。[步骤 5：使用中的创建开发环境中概述了执行此操作 AWS Cloud9 的步骤 CodeCatalyst](#)

在中使用空存储库 AWS Cloud9

首先在 Amazon 中创建个人访问令牌 (PAT)，CodeCatalyst 然后才能 AWS Cloud9 使用空存储库复制您的环境。下一节概述了如何创建此令牌。

在 Amazon 中创建个人访问令牌 CodeCatalyst

您可以在本地计算机上使用 Git 客户端或在集成式开发环境 (IDE) 中访问您在项目中创建的源存储库。要执行此操作，您必须输入应用程序特定的密码。您可以创建用于此目的的个人访问令牌 (PAT)。您创建的个人访问令牌 (PAT) 与您在所有空间和项目中的 CodeCatalyst 用户身份相关联。您可以查看已创建的 PAT 的名称和到期日期，也可以删除不再需要的 PAT。您只能在创建 PAT 密钥时对其进行复制。

创建个人访问令牌 (PAT)

1. 打开 CodeCatalyst 控制台，[网址为 https://codecatalyst.aws/](https://codecatalyst.aws/)。
2. 在顶部菜单栏中，选择您的个人资料徽章，然后选择我的设置。

Tip

您还可以找到您的用户个人资料。要执行此操作，请在项目或空间的成员页面上，从成员列表中选择您的姓名。

3. 在个人访问令牌下，选择创建。
4. 在 PAT 名称中，为您的个人访问令牌 (PAT) 输入一个描述性名称。
5. 在到期日期中，保留默认日期，或者选择日历图标以自定义日期。到期日期默认为自当前日期起之后的 1 年时间。
6. 选择创建。

Tip

当为源存储库选择克隆存储库时，也可以创建此令牌。

7. 要复制 PAT 密钥，请选择复制。将 PAT 密钥储存到您可以检索到的位置。

⚠ Important

PAT 密钥仅显示一次。关闭窗口后将无法再查看。如果您未将 PAT 密钥保存在安全位置，则可以创建新的密钥。

为源存储库创建 PAT 后，CodeCatalyst 通过在 AWS Cloud9 环境中启动一个空存储库并指向您在中创建的源存储库，将数据从您的 AWS Cloud9 环境复制到 CodeCatalyst，如以下部分所述。

在中启动一个空存储库 AWS Cloud9

如果您没有在中设置任何源存储库 AWS Cloud9，请在中启动一个空存储库 AWS Cloud9。此外，请指向您在中创建的源存储库 CodeCatalyst，然后添加和推送要复制的文件 Git。执行以下步骤并运行以下命令将 AWS Cloud9 文件复制到其中 CodeCatalyst。

1. 在您的 AWS Cloud9 环境中，通过运行以下命令启动一个空存储库：

```
git init -b main
```

然后，您会看到如下所示的类似的输出：

```
Initialized empty Git repository in /home/ec2-user/environment/.git/
```

2. 从中克隆源存储库 URL CodeCatalyst。导航到您 CodeCatalyst 控制台中创建的 CodeCatalyst 项目，然后在导航窗格中选择“代码”，然后选择“源存储库”。
3. 从源存储库列表中选择所需的存储库，然后选择克隆存储库，复制克隆 URL。
4. 使用您克隆的 URL 添加 CodeCatalyst 存储库，并将已存在于空存储库中的 CodeCatalyst 内容推送到：

```
git remote add origin [...]  
git push origin --force
```

5. 添加您想要复制的文件。如果要复制环境目录中的所有文件，请运行 `git add -A`：

```
git add -A .  
git commit -m "replicate"
```

6. 合并两个不相关的历史记录。如果合并发生冲突，请解决问题：

```
git merge origin/main --allow-unrelated-histories
```

7. CodeCatalyst 通过运行以下命令将更改推送回源存储库。当系统提示您输入密码时，输入您在[第 3 步的第 10 步中生成的个人访问令牌 \(CODECATALYST_PAT\)](#)（[第 3 步 在项目中创建一个源存储库](#)）：

```
Admin:~/environment (main) $ git push origin main
Password for 'https://222581768915@git.us-west-2.codecatalyst.aws/v1/
MySpace222581768915/Replication/Replication':
```

完成此过程后，您的 CodeCatalyst 存储库将包含您刚刚推送的更新文件和提交。现在，您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。下一节中概述了执行此操作的步骤。

步骤 5：使用 AWS Cloud9 中的创建开发环境 CodeCatalyst

以下过程概述了如何在 using 中 CodeCatalyst 创建开发环境 AWS Cloud9 以及您刚刚复制的数据。

使用创建开发环境 AWS Cloud9

1. 打开 CodeCatalyst 控制台，[网址为 https://codecatalyst.aws/](https://codecatalyst.aws/)。
2. 导航到要在其中创建开发环境的项目。
3. 在导航窗格中，选择概览，然后导航到我的开发环境部分。
4. 选择创建开发环境。
5. AWS Cloud9 从下拉菜单中选择。
6. 选择克隆存储库。

Note

目前，CodeCatalyst 不支持克隆第三方存储库，但您可以创建开发环境并从所选的 IDE 中将第三方存储库克隆到该环境中。

7. 请执行以下操作之一：
 - a. 选择要克隆的存储库，选择在现有分支中工作，然后从现有分支下拉菜单中选择一个分支。
 - b. 选择要克隆的存储库，选择在新分支中工作，在分支名称字段中输入分支名称，然后从创建分支自下拉菜单中选择要从中创建新分支的分支。

8. 可选操作，为开发环境添加别名。
9. 可选操作，选择开发环境配置编辑按钮，编辑开发环境的计算、存储或超时配置。
10. 选择创建。在创建开发环境时，开发环境状态列将显示正在启动，开发环境创建完成后，状态列将显示正在运行。

使用复制工具

AWS Cloud9 in CodeCatalyst 为与之交互提供了完全托管的体验 AWS Cloud9。为了让客户能够尝试 AWS Cloud9 在中使用 CodeCatalyst，我们创建了一个复制工具。在 AWS Cloud9 环境中复制并运行脚本后，按照提示运行脚本并将您的代码资源从复制 AWS Cloud9 到 CodeCatalyst。有关复制工具和过程的更多信息，请参阅下面概述的 [FAQ's on the replication process](#)。

Note

此复制过程不会对您现有的 AWS Cloud9 环境产生任何影响。复制过程完成后，您可以删除开发环境、源存储库、项目和空间，这不会对您的 AWS Cloud9 环境产生任何影响。此工具只会将您的代码资源复制到 AWS Cloud9 中 CodeCatalyst，不会删除或配置您的现有 AWS Cloud9 环境。此复制工具已发布给初始选定的 AWS 账户组。因此，它可能不会出现在某些 AWS 账户中。

Note

建议您在下载该工具之前先注册 Amazon CodeCatalyst 并创建空间。有关注册的信息 CodeCatalyst，请参阅[注册 Amazon CodeCatalyst 并创建空间](#)。

在亚马逊 AWS Cloud9 上使用的好处 CodeCatalyst

以下部分概述了在 AWS Cloud9 上使用时将体验到的一些性能优势和增强功能 CodeCatalyst：

- CodeCatalyst 提供集成体验，使您能够使用完全托管的开发环境从一个位置管理整个软件开发生命周期。
- 改进了发布时的 Amazon EBS 卷大小选项。
- 支持临时环境，并能够按需扩展开发环境的计算。
- 自定义 AMI 支持，可通过自定义映像的规范获得。

- Devfile 支持，让您能够将配置描述为代码。

CodeCatalyst 使用复制工具复制您的 AWS Cloud9 代码资源

以下过程详细介绍了如何复制和运行复制工具，完成复制过程。

1. 复制下面的脚本并确保在 AWS Cloud9 环境中运行它：

```
curl https://dx5z5embsyrja.cloudfront.net -o /tmp/replicate-tool.tar.gz && tar
--no-same-owner --no-same-permissions -xvf /tmp/replicate-tool.tar.gz -C /tmp &&
node /tmp/cloud9-replication-tools
```

2. [可选] 复制工具使用您的 AWS 账户 ID 进行遥测。该操作的目的是帮助我们更好地识别您在使用该工具时可能遇到的任何问题。我们会发出 `tool starts`、`tool fails`、`tool is cancelled by user`、`tool completes successfully` 和 `tool creates a Dev Environment for the user` 遥测事件。如果要禁用复制工具遥测，请参阅下面的[禁用复制工具的遥测](#)。
3. 在您的 AWS Cloud9 环境中复制并运行复制工具后，您需要在浏览器中导航到访问网址，然后在 10 分钟内单击“允许”，从而将您的 AWS 账户与 AWS 构建器 ID 关联起来。请确保您只打开该链接一次，如果您多次打开该链接，则会导致错误，并且需要重新开始。有关 AWS 生成器 ID 的更多信息，请参阅《[登录用户指南](#)》中的“[使用 AWS 生成器 ID AWS 登录](#)”。这将授予复制工具访问您的代码资源的权限，以便在中 CodeCatalyst 进行复制。
4. 选择要使用的空间。如果您只有一个空间，则会选择该空间。有关空间的更多信息，请参阅 Amazon CodeCatalyst 用户指南 [CodeCatalyst 中的空间](#)。
5. 选择是要在中复制代码，CodeCatalyst 还是要在新的开发环境中试一试。我们建议直接在中 CodeCatalyst 复制您的代码。有关开发环境的更多信息，请参阅 Amazon CodeCatalyst 用户指南 [CodeCatalyst 中的开发环境](#)。
6. 输入项目的名称或按 Enter 键使用提供的默认名称。
7. 出现提示时，选择要如何将文件复制到中的新源存储库 CodeCatalyst。您可以选择将根文件夹推送到单个 CodeCatalyst 存储库，也可以选择将子文件夹推送到不同的 CodeCatalyst 存储库。
8. 工具完成后，通过终端消息中提供的 URL 在 CodeCatalyst 控制台中导航到该项目，以访问您的代码资源 CodeCatalyst。

完成此过程后，您的 CodeCatalyst 存储库将包含您刚刚推送的更新文件和提交。现在，您可以从该分支创建开发环境并使用 AWS Cloud9 将其打开。

禁用复制工具的遥测

以下步骤概述了如何设置环境变量来禁用复制工具的遥测。

1. 在您的 AWS Cloud9 环境中打开终端
2. 运行以下命令之一：

```
export CLOUD9_REPLICATION_TOOL_TELEMETRY=off
```

或者

```
export CLOUD9_REPLICATION_TOOL_TELEMETRY=0
```

3. 运行上述命令之一后，将设置环境变量并禁用复制工具的遥测。禁用遥测后，必须复制并重新运行复制工具脚本才能开始该过程。

复制工具反馈

如果您遇到任何问题，或者想就使用复制工具的体验提供反馈，请创建支持案例并提交。有关创建支持案例的更多信息，请参阅 [Creating support cases and case management](#)。

AWS Cloud9 和 Amazon 之间的区别 CodeCatalyst

下表概述了两者 AWS Cloud9 之间 AWS Cloud9 的一些区别 CodeCatalyst。

AWS Cloud9	AWS Cloud9 在亚马逊上 CodeCatalyst
私有 VPC 非常适合使用 AWS Cloud9。	上目前不支持使用私有 VP AWS Cloud9 C CodeCatalyst。
AWS Cloud9 支持预配置的 AWS 托管凭证。	需要为开手动配置凭证 CodeCatalyst。AWS Cloud9
间隔可以在 30 分钟到 7 天之间，也可以使用 AWS Cloud9禁用关机。	开启间隔可能为 15 分钟到 20 小时，CodeCatalyst 并且您无法禁用关机。AWS Cloud9
AWS Cloud9 支持 Ubuntu 和 AL2 操作系统平台。	AWS Cloud9 on CodeCatalyst 支持 MDE 通用镜像和可以包含 Ubuntu 和 AL2 的自定义镜

AWS Cloud9	AWS Cloud9 在亚马逊上 CodeCatalyst
	像。有关这方面的更多信息，请参阅 Amazon CodeCatalyst 用户指南中的 通用开发文件镜像 。
支持上传和下载 AWS Cloud9	on 目前不支持上传和 AWS Cloud9 下载 CodeCatalyst。用户需要使用 Amazon S3 存储桶进行上传和下载。
协作可用于 AWS Cloud9	协作功能目前不可用 AWS Cloud9 CodeCatalyst。

有关复制过程的常见问题

以下部分旨在回答与复制工具和复制过程有关的一些常见问题解答。

问题：如果我在上复制我的 AWS Cloud9 环境 CodeCatalyst，我的 AWS Cloud9 环境会受到影响吗？

答：不是，您的环境的复制只会将您的代码资源复制到，AWS Cloud9 以便您 CodeCatalyst 能够继续工作。您的代码资源和开启的环境 AWS Cloud9 不会受到任何影响。

问题：如果我想回滚，我的 AWS Cloud9 环境会受到影响吗？

答：不，您可以删除 CodeCatalyst 开发环境、源存储库、项目和空间，它不会对您的 AWS Cloud9 环境产生任何影响。

问：新位置是否符合 HIPAA、SOC 等标准？

答案：开启的开发环境 CodeCatalyst 目前不符合这些标准。这些标准的合规性已列入规划路线图。

问：我的代码资源会去哪里？

答：您的代码资源将被复制到项目中的源存储库中 CodeCatalyst。

问：使用会受到限制吗？

答：在复制过程中，您在免费套餐内可以创建 16 GB 的开发环境。这意味着您最多可以拥有 4 个开发环境。有关定价、存储空间和不同可用套餐的更多信息，请参阅[Amazon CodeCatalyst -定价](#)。

问：我的计算将去哪里？

答：您的现有计算不会发生任何变化。它将保持原样。

问题：我能否在中 CodeCatalyst 使用我现有的 AWS 账户凭证？这些凭证会被自动转移吗？

答：您可以在中手动配置您的 AWS 账户凭证 CodeCatalyst。它们不会自动转移。

问：这项服务要花多少钱？

答：您可以免费开始 CodeCatalyst 使用。有关定价和不同可用套餐的更多信息，请参阅 [Amazon CodeCatalyst -定价](#)。

问题：数据复制过程和数据存储是否 CodeCatalyst 安全？

答：是的，我们将使用带有 https 的 git push 来复制代码资源并将数据 CodeCatalyst 安全地存储在服务中。静态和传输中的所有数据均加密。有关数据保护的更多信息 CodeCatalyst，请参阅亚马逊 CodeCatalyst 用户指南 CodeCatalyst 中的亚马逊 [数据保护](#)。

问：我应该选择哪种复制方法？

答：复制工具提供了两种方法；您可以 CodeCatalyst 通过将代码资源推送 AWS Cloud9 到单个 CodeCatalyst 源存储库来将其从复制到，或者将每个子文件夹转换为不同的 CodeCatalyst 源存储库。我们建议使用第一种方法，因为它不需要事先了解源存储库等 CodeCatalyst 概念。这种方法是探索 AWS Cloud9 体验的好起点 CodeCatalyst，同时使用你习惯的类似设置 AWS Cloud9。

当您单独使用位于根 AWS Cloud9 环境文件夹下的子文件夹时，最好选择第二个选项。使用这种方法，根文件夹下的所有文件都不会被复制。有关源存储库的更多信息 CodeCatalyst，请参阅 Amazon CodeCatalyst 用户指南 CodeCatalyst [中的源存储库](#)。

问：复制过程中生成的个人访问令牌是什么？我为什么需要它？如果丢失了还能重新生成吗？

答案：个人访问令牌与您的用户身份相关联 CodeCatalyst。使用 git 将本地更改推送到 CodeCatalyst 源存储库时，需要将其作为密码。有关令牌及其生成方法的更多信息，请参阅《亚马逊 CodeCatalyst 用户指南》[中的在亚马逊 CodeCatalyst 管理个人访问令牌](#)。

问：如果在复制过程中出现错误会怎样？

答：如果在使用复制工具时出现错误，请先重试该工具。如果错误与源存储库有关，则可以在复制代码资源后将其手动推送到 CodeCatalyst 源存储库。这应该可以正常工作，因为本地存储库已经配置为与 CodeCatalyst 上游配合使用。如果问题仍然存在，请创建支持案例并提交。有关创建支持案例的更多信息，请参阅 [Creating support cases and case management](#)。

问题：为什么我需要使用我的 B AWS uilderID 对复制工具进行身份验证和授予权限？

答：在复制过程中，复制工具需要代表用户读取和写入多个资源（项目、开发环境、源存储库）CodeCatalyst 并复制本地内容，因此需要您的许可才能执行此操作。

问题：如果我移至，延迟会有变化 CodeCatalyst 吗？

答：根据您正在执行的操作，延迟可能会减少。这是因为 CodeCatalyst 服务器托管在 PDX 区域。

问：我安装的所有软件都会转移吗？

回答：不会，只会转移您的代码资源。二进制文件、配置和已安装的软件不会被传输。

Amazon 中的开发环境 CodeCatalyst

以下各节概述了如何 CodeCatalyst 使用 AWS Cloud9 IDE 创建和管理您的开发环境。

- [创建开发环境](#)
- [打开开发环境设置](#)
- [恢复开发环境](#)
- [删除开发环境](#)
- [编辑开发环境的存储库 devfile](#)
- [克隆存储库](#)
- [对开发环境进行故障排除](#)

创建开发环境

您可以通过多种方式创建开发环境：

- 使用“摘要”、“开发环境”或“CodeCatalyst 源存储库”页面中的 CodeCatalyst 源存储库创建开发环境。
- 创建一个未连接到开发环境中的 CodeCatalyst 源存储库的空开发环境。
- 在您选择的 IDE 中创建开发环境，然后将 CodeCatalyst 源存储库克隆到开发环境中。

您可以为每个分支和存储库创建一个开发环境。一个项目可以有多个存储库。您的开发环境仅与您的 CodeCatalyst 账户关联，并且只能由您的 CodeCatalyst 账户管理。您可以打开开发环境并将该环境与任何支持的 IDE 结合使用。选择特定 IDE 后，只能使用所选 IDE 打开该开发环境。如果要使用其他 IDE，则可以通过在导航栏中选择开发环境并选择“编辑”来更改 IDE，也可以通过创建新的开发环境来更改 IDE。默认情况下，系统使用双核处理器、4 GB RAM 和 16 GB 持久性存储创建开发环境。

有关如何在云中创建开发环境的更多信息 CodeCatalyst，请参阅 Amazon CodeCatalyst 指南中的[创建开发环境](#)。

有关在云中创建开发环境的信息和步骤 CodeCatalyst，请参阅 Amazon CodeCatalyst 用户指南中的[创建开发环境](#)。

Note

您现在可以使用第三方源存储库创建开发环境。有关将第三方源存储库链接到其中的项目的信息 CodeCatalyst，请参阅 Amazon CodeCatalyst 用户指南中的[链接源存储库](#)。

打开开发环境设置

在 CodeCatalyst 控制台中创建开发环境后，您可以查看特定的开发环境设置：

1. 在 CodeCatalyst 控制台中，通过 AWS Cloud9 IDE 导航到您的开发环境。
2. 从 AWS Cloud9 侧边栏中选择 aws-explorer。
3. 在开发者工具导航窗格中，展开 CodeCatalyst 并选择打开设置以打开开发环境设置视图。
4. 在开发环境设置视图中，以下部分包含开发环境的选项：
 - 别名：查看和更改分配给您的开发环境的别名。
 - 状态：查看您当前的开发环境状态、已分配给它的项目以及停止您的开发环境。
 - Devfile：查看开发环境的 Devfile 的名称和位置。选择在编辑器中打开可打开 Devfile。
 - 计算设置：更改开发环境的大小和默认超时长度。

Note

开发环境在创建之后就无法更改分配到其中的存储空间量。

Note

在终端上使用亚马逊 CodeCatalyst AWS CLI 时，必须确保在运行任何命令之前设置了 `aws_profile=code Catalyst`。CodeCatalyst

恢复开发环境

开发环境 \$HOME 目录中的所有内容都将持久存储。如果需要，您可以停止在开发环境中工作，并在以后恢复开发环境中的工作。如果开发环境处于空闲状态的时间达到了在创建开发环境时为超时字段选择的时间长度，此时会话将自动停止。

您只能从中恢复开发环境 CodeCatalyst。有关如何恢复开发环境的更多信息，请参阅 Amazon CodeCatalyst 指南中的[恢复开发环境](#)。

Note

恢复开发环境可能需要几分钟的时间。

删除开发环境

处理完存储在开发环境中的内容后，您可以将其删除。在删除开发环境之前，请确保提交代码更改并推送到原始源存储库。删除开发环境后，将停止对开发环境的计算和存储计费。

您只能从中的开发环境页面中删除开发环境 CodeCatalyst。有关如何删除开发环境的更多信息，请参阅 Amazon CodeCatalyst 指南中的[删除开发环境](#)。

编辑开发环境的存储库 devfile

要更改开发环境的配置，请编辑 devfile。您可以利用 devfiles 标准化整个团队的开发环境。您可以 devfile 从中源存储库的根目录进行编辑 CodeCatalyst。或者，您可以在支持的 IDE 中编辑 devfile。如果您在支持的 IDE 中编辑 devfile，请提交更改并将其推送到源存储库或创建拉取请求。这样，团队成员就可以审查和批准对 devfile 的编辑。

Note

您只能在 devfile 中包含公共容器镜像。

Note

如果缺少依赖项，则某些 AWS Cloud9 IDE 功能可能无法在自定义模式下运行 devfile。要使它们在 Linux x64 以外的某些平台上运行，可能需要额外的工作。

要在中编辑开发环境的存储库 devfileAWS Cloud9

1. 在 CodeCatalyst 控制台中，通过 AWS Cloud9 IDE 导航到您的开发环境。
2. 从 AWS Cloud9 侧栏中选择 aws- explorer。
3. 在“开发者工具”导航窗格中，选择CodeCatalyst 工具包菜单。
4. 选择打开 Devfile。
5. 编辑 devfile，然后保存文件。
6. 选择源代码控制，这是菜单侧边栏中的 Git 扩展。
7. 在消息文本字段中，输入一条消息，然后暂存更改。
8. 要准备提交，请选择暂存所有更改 (+) 图标。
9. 要查看 Git 命令，请选择存储库名称旁边的菜单图标。
10. 选择提交和推送。
11. 从 AWS Toolkit 菜单中选择更新开发环境。

选择提交和推送。已保存更新的 devfile，并且已提交并推送更改。

Note

假设您想要使用自定义 devfile 启动的开发环境无法正常工作。这可能是由于 devfile 与 AWS Cloud9不兼容。要排除故障，请检查 devfile。如果问题仍然存在，请将其删除并尝试创建一个新 devfile。

您也可以通过devfile编辑开发环境的 CodeCatalyst。有关更多信息，请参阅 Amazon CodeCatalyst 指南中的[配置您的开发环境](#)。

克隆存储库

要有效地处理源存储库中的多个文件、分支和提交，可以将源存储库克隆到本地计算机上。然后，使用 Git 客户端或 IDE 进行更改。从 CodeCatalyst，您可以像使用任何其他Git主机提供程序一样使用 AWS Cloud9 IDE Git 扩展，也可以使用命令行。要了解如何克隆第三方存储库，请参阅 [Initialize or clone a Git repository](#)。

有关从源存储库创建开发环境并使用其克隆的更多信息 CodeCatalyst，请参阅 Amazon CodeCatalyst 指南中的[源存储库概念](#)。

对开发环境进行故障排除

如果您的开发环境遇到问题，请参阅 Amazon CodeCatalyst 指南中的[开发环境问题疑难解答](#)。

Note

在终端上使用亚马逊 CodeCatalyst AWS CLI 时，必须确保在运行任何命令之前设置了 `aws_profile=code Catalyst`。CodeCatalyst

如果您的开发环境遇到问题，请参阅 Amazon CodeCatalyst 指南中的[开发环境问题疑难解答](#)。

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CDK

通过 AWS CDK 服务，您可以使用 [AWS Cloud Development Kit \(AWS CDK\)](#) 应用程序或应用。您可以在《AWS Cloud Development Kit (AWS CDK) 开发人员指南》中找到有关 AWS CDK 的详细信息。

AWS CDK 应用程序由称为[构造](#)的构建块组成。这些构建块包括对于您的 AWS CloudFormation 堆栈及其中的 AWS 资源的定义。使用 AWS CDK Explorer，您可以看到在 AWS CDK 树视图中定义的[堆栈](#)和[资源](#)。您可以在 AWS Cloud9 编辑器的 Developer Tools (开发人员工具) 窗格中访问此视图。

本节提供有关如何在 AWS Cloud9 编辑器中访问和使用 AWS CDK 的信息。

使用 AWS CDK 应用程序

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CDK Explorer 可视化和处理 AWS CDK 应用程序。

先决条件

安装 AWS CDK 命令行界面。有关说明，请参阅《AWS Cloud Development Kit (AWS CDK) 开发人员指南》中的[AWS CDK 入门](#)。

Important

您安装的 AWS CDK 版本必须是 1.17.0 或更高版本。您可以使用 `cdk --version` 命令检查您正在运行哪个版本。

可视化 AWS CDK 应用程序

使用 AWS Cloud9 IDE AWS CDK Explorer，您可以管理存储在应用程序的 CDK 结构中的[堆栈](#)和[资源](#)。AWS CDK Explorer 使用在 `tree.json` 文件中定义的信息在树视图中显示您的资源。此文件是您运行 `cdk synth` 命令时创建的。原定设置情况下，`tree.json` 文件位于应用程序的 `cdk.out` 目录中。

要开始使用工具包的 AWS CDK Explorer，请创建 CDK 应用程序。

1. 完成《AWS CDK 开发人员指南》中 [Hello World 教程](#) 的前几个步骤。

Important

当您到达部署堆栈步骤时，请停止并返回本指南。

Note

您可以在操作系统命令行界面上或在 VS Code 编辑器内的 Terminal (终端) 窗口中运行本教程中提供的命令，例如 `mkdir` 和 `cdk init`。

2. 完成 CDK 教程的所需步骤后，打开您在 AWS Cloud9 IDE 编辑器中创建的 CDK 内容。
3. 在 AWS 导航窗格中，展开 CDK 标题。您的 CDK 应用程序及其关联资源现在显示在 CDK Explorer 树视图中。您还可以在 AWS Cloud9 内的终端中运行以下命令来确认 CDK 功能正常运行：

```
mkdir mycdkapp
cd mycdkapp
cdk init app --language=typescript
cdk synth
cdk bootstrap
```

重要提示

- 将 CDK 应用程序加载到 AWS Cloud9 编辑器中时，您可以一次加载多个文件夹。每个文件夹可以包含多个 CDK 应用程序，如上图所示。AWS CDK 资源管理器在项目根目录及其直接子目录中查找应用程序。

- 执行此教程的前几个步骤时，您可能会注意到所运行的最后一个命令是 **cdk synth**。此命令通过将您的 AWS CDK 应用程序转换为 CFN 来合成 CloudFormation 模板。作为副产品，它还会生成 `tree.json` 文件。如果您对 CDK 应用程序进行了更改，请再次运行 **cdk synth** 命令以查看树视图中反映的更改。一个示例更改是向应用程序添加更多资源。

对 AWS CDK 应用程序执行其他操作

您可以使用 AWS Cloud9 编辑器在 CDK 应用程序上执行其他操作，就像使用命令行界面一样。例如，您可以更新编辑器中的代码文件，并使用 AWS Cloud9 Terminal (终端) 窗口部署应用程序。

要尝试这些类型的操作，请使用 AWS Cloud9 编辑器继续学习《AWS CDK 开发人员指南》中的 [Hello World 教程](#)。确保执行最后一个步骤，即销毁应用程序的资源。否则，您的 AWS 账户 可能会产生意料之外的费用。

使用 Git 面板进行可视化源代码控制

适用于 AWS Cloud9 的 Git 面板为使用基本 Git 功能提供了便捷的可视化界面。

使用 Git 面板界面中的选项，您可以管理完整的源代码控制生命周期：初始化存储库或克隆远程存储库、将文件添加到暂存区、将暂存文件提交到工作目录，然后将更改推送到上游存储库。

Git 的核心协作和项目管理功能，例如创建和合并分支，只需在 Git 面板界面中单击几下即可快速实现。此外，可以使用 IDE 的编辑器窗口识别和解决合并冲突。

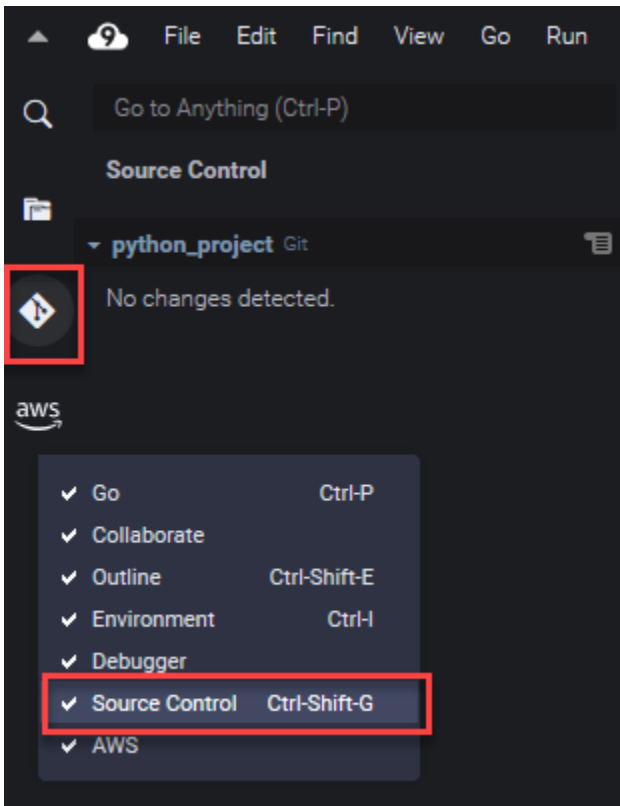
Important

Git 面板仅在使用 Amazon EC2 实例创建的 AWS Cloud9 环境中可用。如果使用的是 [SSH 开发环境](#) 而不是 EC2 环境，则无法访问此功能。

此外，Git 面板默认仅在 2020 年 12 月 11 日之后创建的新 AWS Cloud9 环境中可用。我们正致力于为在此日期之前创建的开发环境启用 Git 面板。

要访问界面并与其交互，请依次选择 Window (窗口)、Source Control (源代码控制)。或者，您也可以右键单击 IDE 侧面板中的任意位置，然后选择 Source Control (源代码控制) 来访问源代码控制。然后，选择 IDE 界面中显示的 Git 图标。

组合键 Ctrl-Shift-G 也可用于切换 Git 面板的显示。



Note

Git 面板文档的屏幕截图显示应用了 Jett Dark 主题的 AWS Cloud9 IDE。如果您使用具有不同主题的 IDE，则某些界面元素的显示方式会有所不同。要打开 Git 面板，您可以选择带有 Source Control（源代码控制）标注的链接而不是 Git 图标。

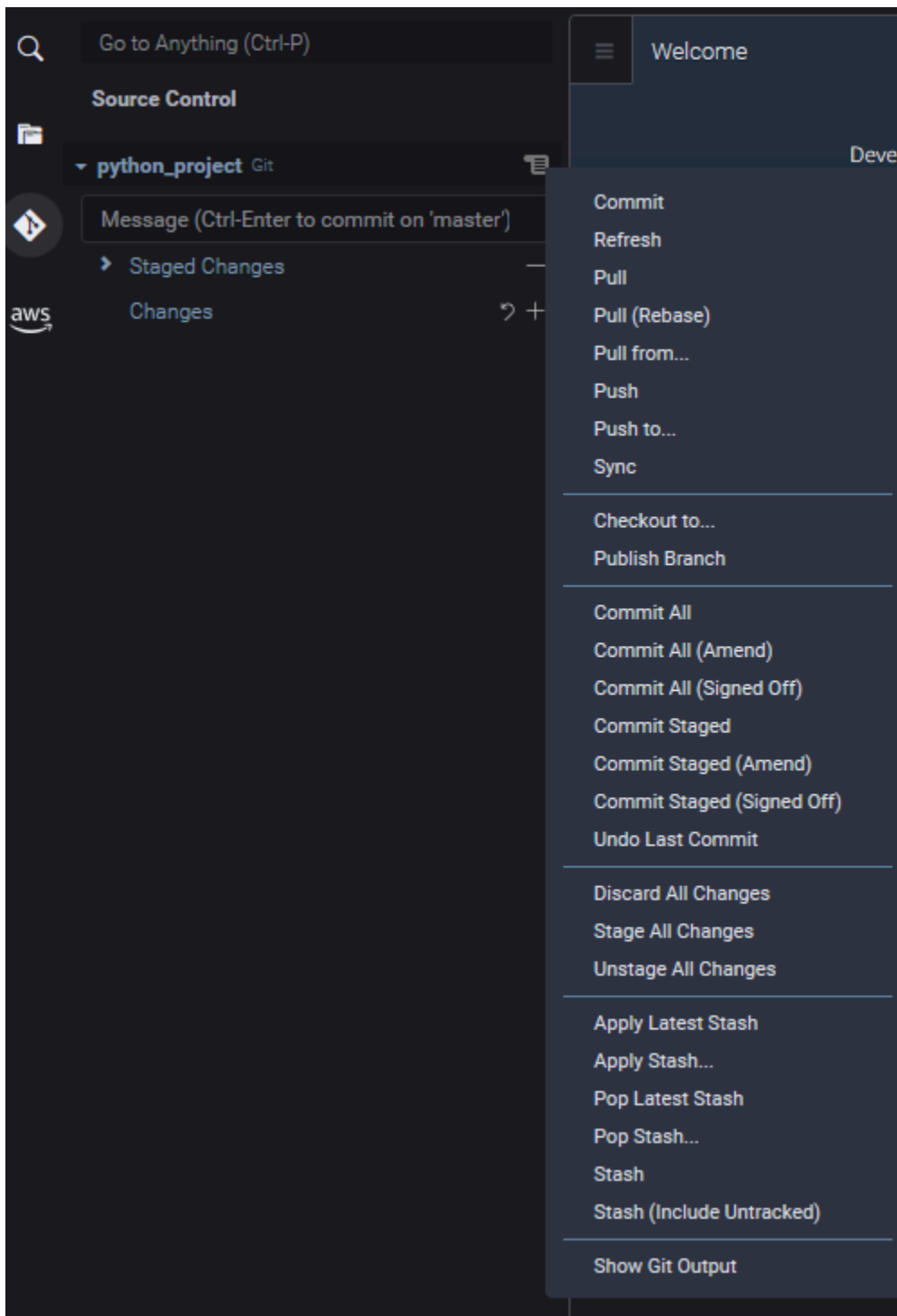
主题

- [使用 Git 面板管理源代码控制](#)
- [参考：Git 面板中可用的 Git 命令](#)

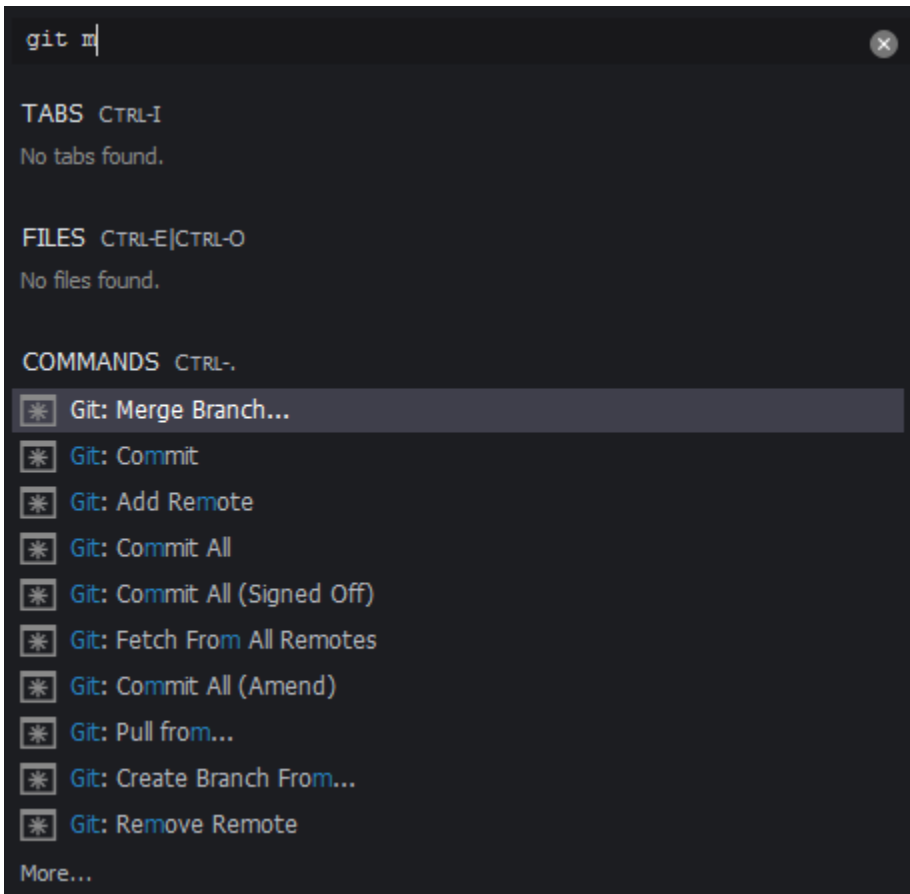
使用 Git 面板管理源代码控制

AWS Cloud9 的 Git 面板扩展提供了用户界面对核心和高级 Git 命令的便捷访问。

本节演示如何访问关键 Git 功能来管理源代码控制。这些过程侧重于使用 Git 面板菜单以针对您的存储库及其内容运行 Git 命令。



您还可以通过在 Git 面板搜索框中开始输入名称来访问任何受支持的 Git 命令：



您还可以查看与 Git 面板界面交互时运行的实际 Git 命令。要查看命令行活动，请转至 Git panel (Git 面板) 菜单，然后选择 Show Git Output (显示 Git 输出)。

```
Git x (+)

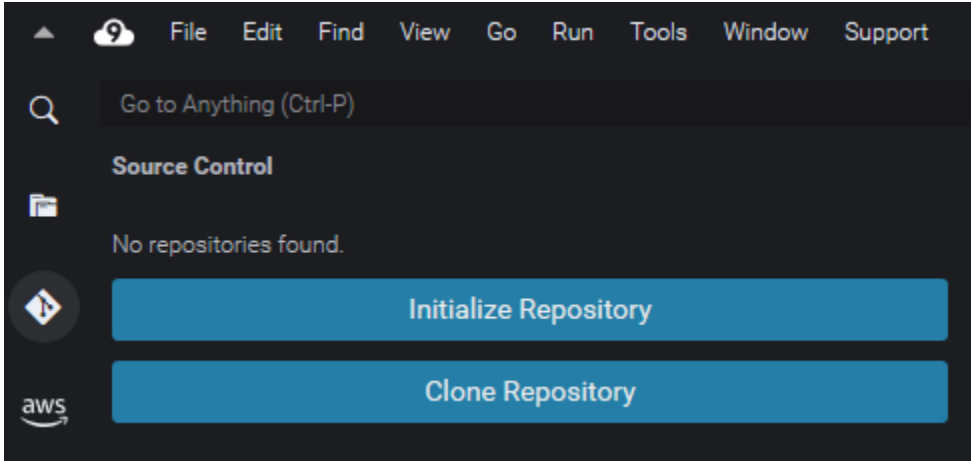
git.stage 1
git.stage.scmResources 1
> git add -A -- /home/ec2-user/environment/python_project/pythonfile.py
> git status -z -u
> git symbolic-ref --short HEAD
> git rev-parse master
> git rev-parse --symbolic-full-name master@{u}
fatal: no upstream configured for branch 'master'
> git for-each-ref --format %(refname) %(objectname) --sort -committerdate
> git remote --verbose
> git show :pythonfile.py
> git show :pythonfile.py
```

初始化或克隆 Git 存储库

Git 存储库 (“存储库”) 包含项目从头至尾的完整历史记录。存储库包含每次将暂存文件提交至该存储库时所捕获的项目内容的所有快照。

Git 面板支持两种获取 Git 存储库的方式：

- 将现有目录初始化为 Git 存储库。
- 克隆现有存储库并将其复制到本地目录。



Note

仅当您尚未将 Git 存储库添加到环境中的工作区文件夹时，用于初始化或克隆存储库的界面选项方可用。如果您已有存储库的工作目录，则 Git 面板窗口会显示工作目录和暂存区的状态。Git panel (Git 面板) 菜单也可用于访问您可以针对存储库运行的 Git 命令。

初始化或克隆存储库

1. 如果 Git 面板尚不可用，请依次选择 Window (窗口)、Source Control (源代码管理)，然后选择 Git 图标，以对其进行访问。

Note

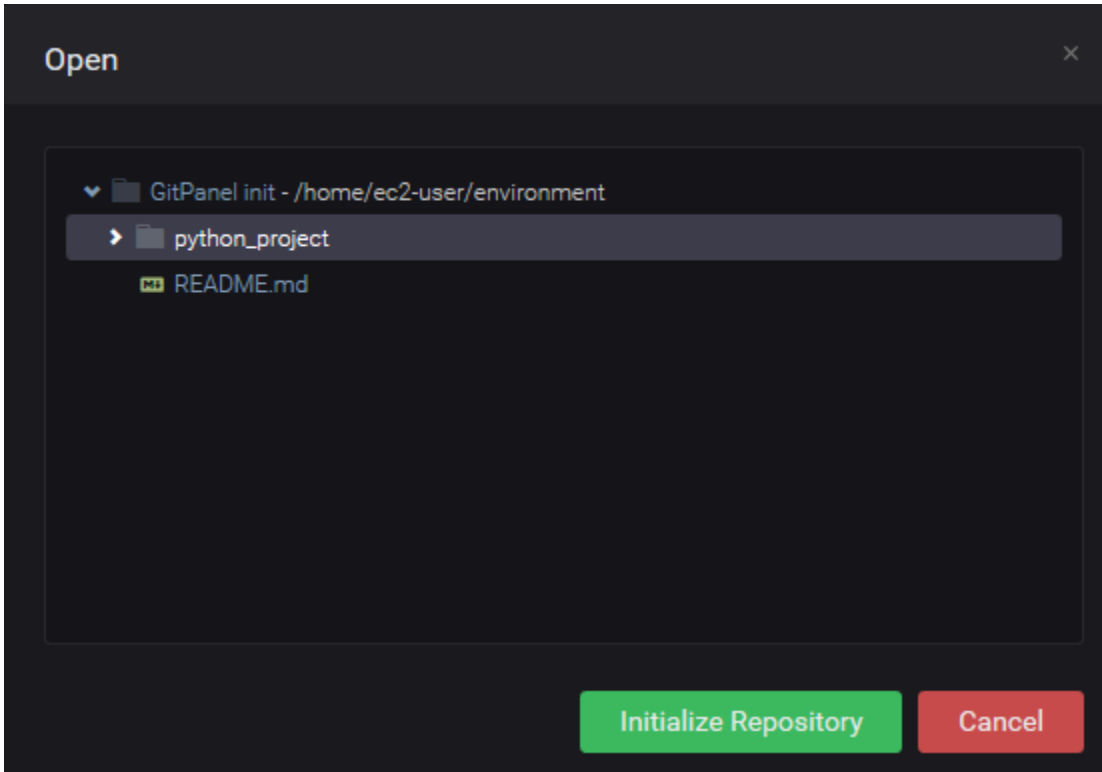
您还可以使用键盘快捷键 `Ctrl+Shift+G` 打开 Git 面板。

2. 选择初始化一个新的存储库或克隆一个现有的存储库。

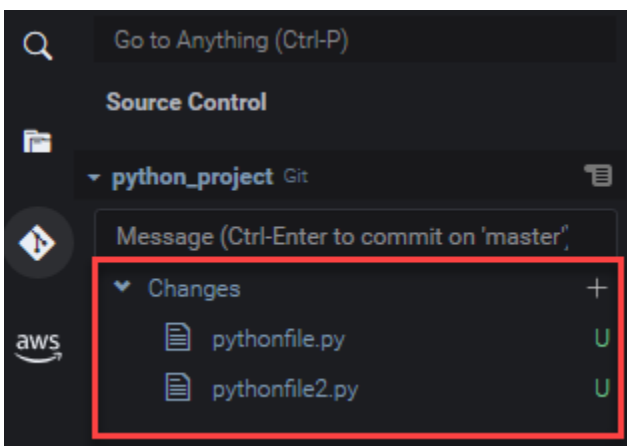
Initialize a repository

- 在 Git 面板中，选择 Initialize Repository (初始化存储库)。

- 接下来，选择一个工作区文件夹，您的 Git 存储库将在其中进行初始化。您可以输入文件夹的路径、选择路径或在对话框中选择文件夹。
- 如果您使用的是对话框，请选择目标文件夹并选择 Initialize Repository (初始化存储库)。



在选定的文件夹中初始化 Git 存储库后，Git 面板会将该文件夹中已有的所有文件显示为未跟踪并准备添加到 Git 暂存区。



Clone a repository

- 在 Git 面板窗口中，选择 Clone Repository (克隆存储库)。

- 接下来，输入要克隆的远程存储库的 URL（例如 `https://github.com/my-own-repo/my-repo-project-name.git`，以克隆托管在 GitHub 上的存储库）并按 Return（返回）。
- 在显示的对话框中，为克隆的存储库选择工作区文件夹，然后选择 Select Repository Location（选择存储库位置）。

Note

如果您正在访问托管在外部站点（例如 GitHub）上的存储库，您还需要输入该站点的登录凭证才能完成该过程。

在选定的文件夹中克隆远程存储库后，您可以运行 `git pull` 命令将本地存储库与远程存储库中的最新更改同步。有关更多信息，请参阅[使用远程存储库](#)。

暂存和提交文件

获取 Git 存储库后，您可以使用两步过程开始为其填充内容：

1. 将未跟踪或最近修改的内容添加到暂存区。
2. 将暂存区中的文件提交到工作目录。

Important

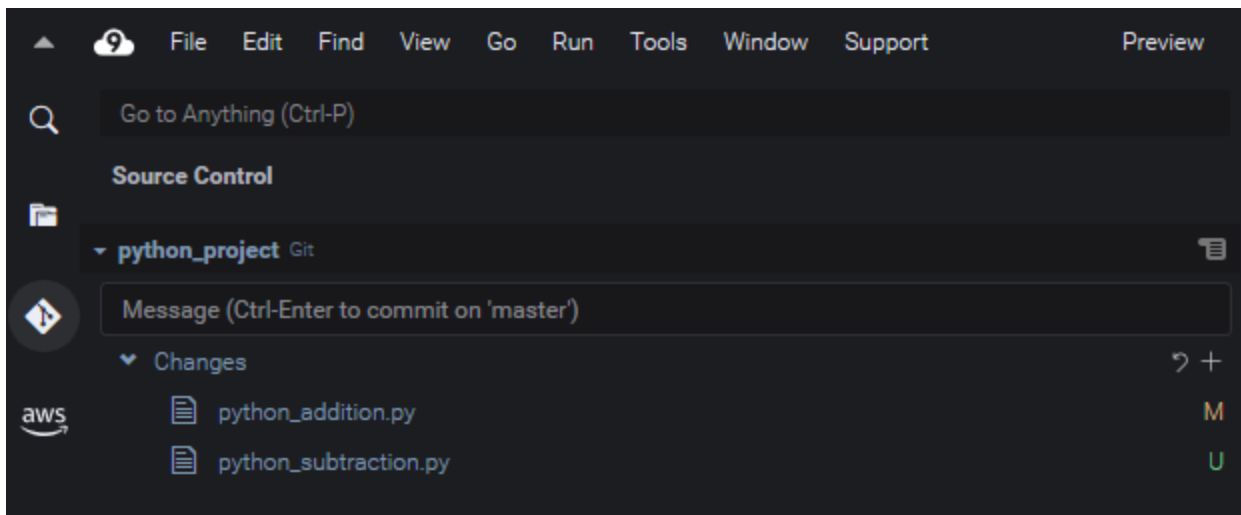
您可能不希望将工作目录中的每个文件都提交到存储库。例如，您不太可能希望将运行时生成的文件添加到项目的存储库中。使用 Git 面板，您可以通过将文件添加到 `.gitignore` 文件中的列表，对要忽略的文件进行标记。

要更新 `.gitignore` 中的列表，请右键单击尚未添加到暂存区的文件，然后选择 Add File to `.gitignore`（将文件添加到 `.gitignore`）。IDE 会打开 `.gitignore` 文件，并将选定文件的名称添加到要忽略的文件列表中。

有关使用 `.gitignore` 中的模式匹配以排除文件类型的信息，请参阅git-scm.com 站点中的[参考](#)的相关内容。

Stage files

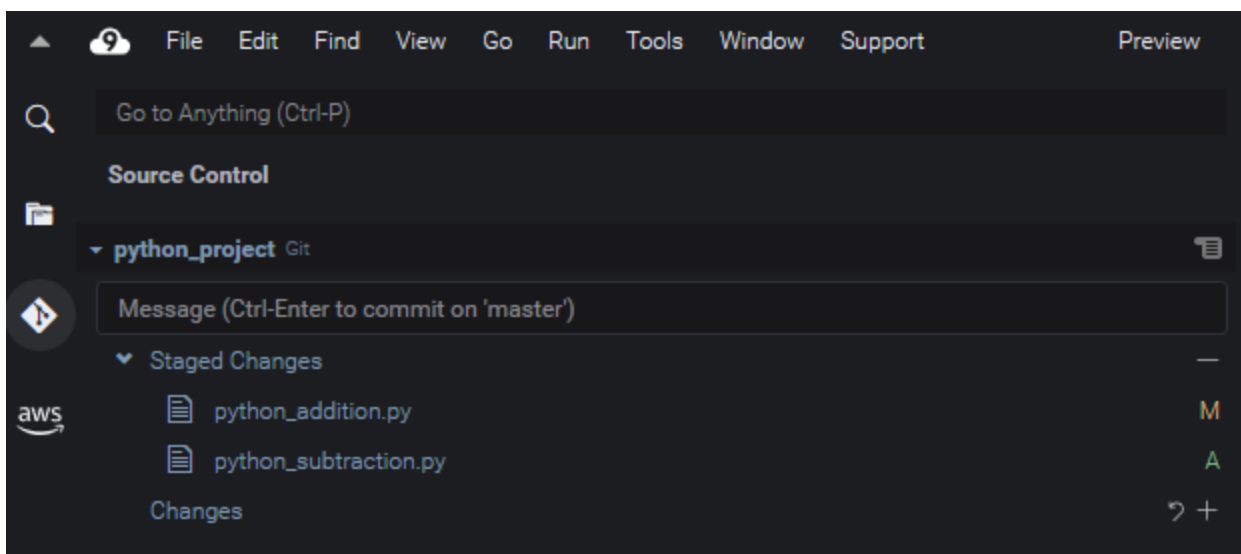
尚未添加到暂存区的未跟踪文件（标记为“U”）和已修改文件（标记为“M”）列在 Git 面板窗格中的 Changes（更改）下。



使用 Git 面板界面，您可以将特定文件或所有未跟踪和已修改的文件添加到暂存区：

- 特定文件：在文件上暂停，然后选择 + 将其添加到暂存区。或者，右键单击该文件并选择 Stage Changes（暂存的更改）。
- 所有文件：转到 Git panel（Git 面板）菜单，然后选择 Stage All Changes（暂存所有更改）。

添加到存储库索引的文件列在 Staged Changes（暂存的更改）下。以前未跟踪的文件标记为“A”，以表明其已被暂存。



Note

您还可以取消暂存特定更改或所有更改。对于单个文件，在文件上暂停，然后选择 `-`。或者，右键单击该文件，然后选择 `Unstage Changes`（取消暂存更改）。要取消暂存所有更改，请转到 `Git panel`（Git 面板）菜单，然后选择 `Unstage All Changes`（取消暂存所有更改）。

Commit files

您可以使用 Git 的 `commit` 命令将暂存文件捕获为存储库中的永久快照。使用 Git 面板界面，您可以选择要提交的文件：

- 提交暂存区的文件：转到 `Git panel`（Git 面板）菜单，然后选择 `Commit`（提交）或 `Commit Staged`（提交暂存）。
- 提交工作目录中的所有文件：转到 `Git panel`（Git 面板）菜单，然后选择 `Commit All`（提交全部）。（此选项使用 `git add` 将文件添加到暂存区，然后再调用 `git commit`。）

Note

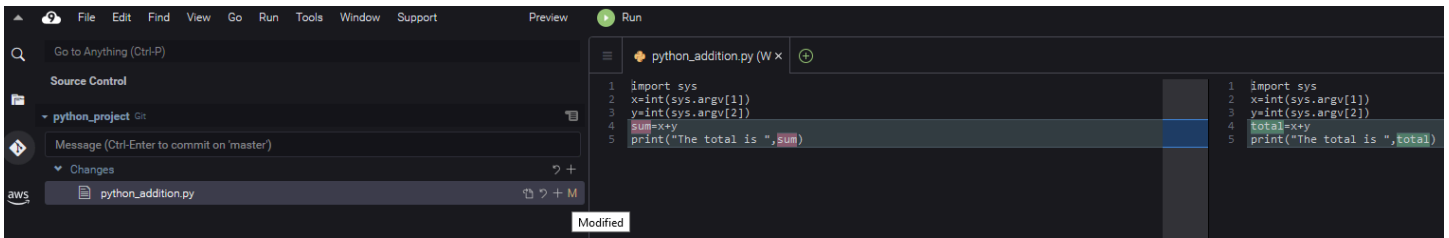
您还可以在使用 Git 面板提交文件时使用 `amend` 和 `signed-off` 选项。`amend` 选项会修改最近提交的提交消息。`sign-off` 选项可以识别在 Git 日志中执行了提交的用户。您还可以通过转到 `Git panel`（Git 面板）菜单，然后选择 `Undo Last Commit`（撤销上次提交）以撤销提交。

查看不同的文件版本

您可以比较在暂存或提交后修改的文件的版本。

- `Changes`（更改）下列出的文件：选择“M”以查看工作目录中的版本与上次暂存或提交到存储库的版本之间的差异。
- `Staged Changes`（暂存的更改）下列出的文件：选择“M”以查看暂存区中的版本与上次提交到存储库的版本之间的差异。

选择“M”后，IDE 窗口会显示两个版本的文件之间的差异。一侧显示会在存储库中被跟踪为当前版本的版本。另一侧会显示尚未提交的修改版本。



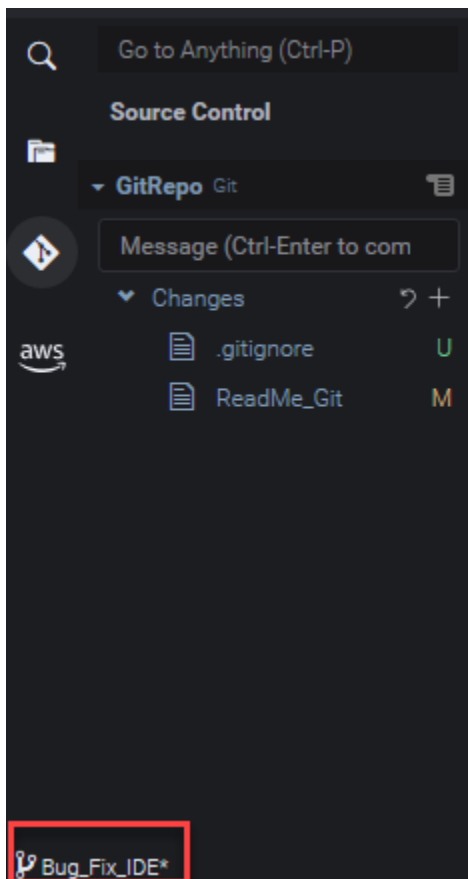
使用分支

Git 允许您在独立于存储库主分支的分支中处理新功能，从而极大地促进了 workflow 管理。您可以在多个分支之间无缝切换，同时确保您始终在主分支中拥有可供构建的源代码。

创建分支

创建分支涉及命名分支并选择其起点。

1. 在 Git panel (Git 面板) 菜单中，选择 Checkout to (签出到)。或者，您可以选择显示在 Git 面板底部的当前分支的名称。



2. 选择用于创建新分支的选项：

- Create new branch (创建新分支) : 新分支从当前分支的最后一次提交开始。
 - Create new branch from (创建新分支自) : 新分支从您在后续屏幕中选择的分支的最后一次提交开始。
3. 输入新分支的名称。
 4. 如果您将特定分支指定为分支的起始点，请从列表选择一个分支。

切换到新分支后，您可以通过查看 Git 面板的底部来检查当前分支的名称。

Note

如果您正在使用远程存储库，请[将新分支发布](#)到上游远程存储库以允许其他人访问您的内容。

切换分支

使用 Git 管理源代码控制的一个关键优势是，您只需切换分支即可在不同项目之间跳转。

Important

如果当前分支中有尚未提交到存储库的文件，则无法切换分支。您必须首先清理您的工作目录，可通过 [committing \(提交 \)](#) 或 [stashing \(存储 \)](#) 您的工作来执行。

1. 在 Git 面板底部选择当前分支的名称。或者，转到 Git panel (Git 面板) ，然后选择 Checkout to (签出到) 。
2. 从显示的列表选择一个分支。

切换后，系统会使用最近提交到选定分支的文件版本更新存储库的工作目录。

合并分支

完成离散分支中功能的处理工作后，您通常希望将更改集成到主项目中。使用 Git，通过将分支 (例如功能分支) 合并到另一个分支 (通常是存储库的主分支或默认分支) 来促进这种集成。

1. 要选择将合并到另一个分支的分支，请转到 Git panel (Git 面板) 菜单，然后选择 Checkout to (签出到) 。

或者，在 Git 面板底部选择当前分支的名称。

2. 从显示的列表中，选择要切换到分支。
3. 在 Git 面板的 Search (搜索) 框中，开始输入“merge”一词。

当 Git: Merge Branch 显示在 Commands (命令) 列表下时，选择它。

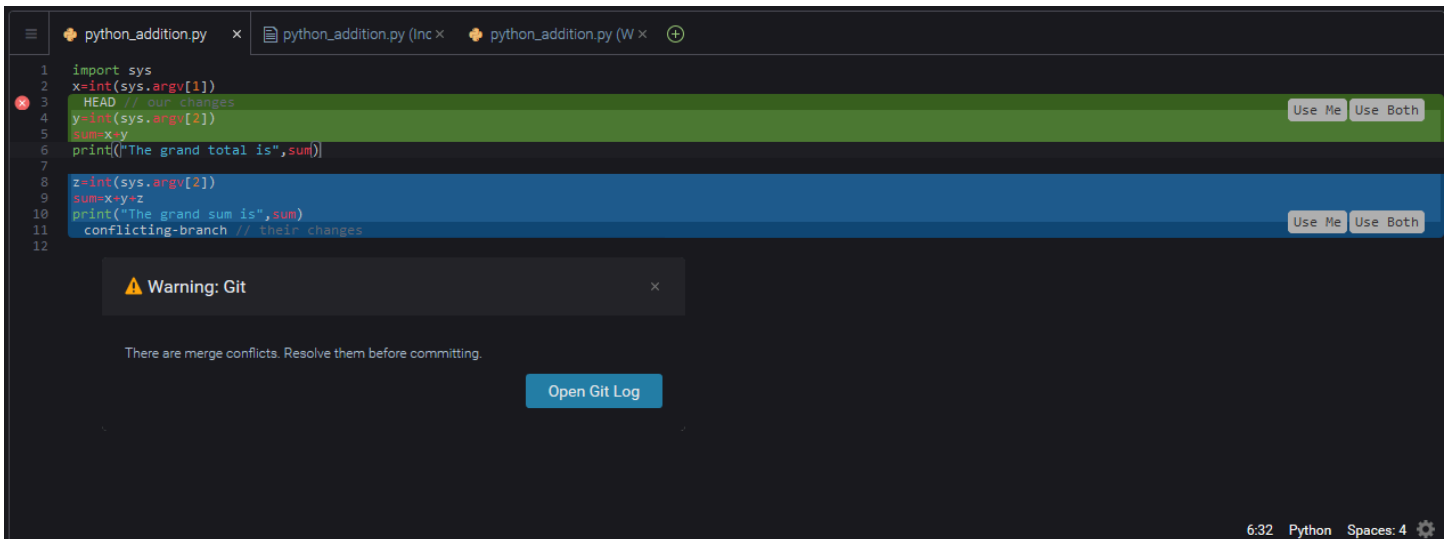


4. 从显示的列表中，选择要合并到目标分支的分支。

如果合并完成时没有冲突，Git 面板界面会刷新以显示包含合并更改的目标分支。

合并分支时，您可能会遇到由于对相同内容所做的不兼容更改而导致的合并冲突。如果发生这种情况，系统会警告您务必在提交合并之前解决冲突。

您可以使用 IDE 的代码编辑器窗口来识别两个分支中的冲突内容，然后进行更改以解决冲突。



```
1 import sys
2 x=int(sys.argv[1])
3 HEAD // our changes
4 y=int(sys.argv[2])
5 sum=x+y
6 print("The grand total is",sum)
7
8 z=int(sys.argv[2])
9 sum=x+y+z
10 print("The grand sum is",sum)
11 conflicting-branch // their changes
12
```

Warning: Git

There are merge conflicts. Resolve them before committing.

Open Git Log

6:32 Python Spaces: 4

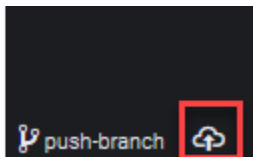
使用远程存储库

托管在互联网或网络上的远程存储库允许团队成员共享其向本地责任所提交的更改，从而促进了协作。通过使用上传和下载数据的 Git 命令，您可以确保“下游”（本地）存储库的内容与“上游”（远程）存储库的内容同步。

将分支发布到远程存储库

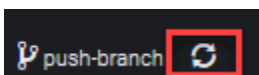
为本地存储库创建分支后，该分支对您而言是私有的，并且在您将其“上游”推送到远程存储库之前，您的协作者无法使用该分支。

1. 要发布当前分支，请转到 Git panel (Git 面板) 菜单，然后选择 Publish Branch (发布分支)。或者，单击 Git 面板底部分支名称旁边的云符号。



2. 如果需要，请输入您的登录凭证以访问远程存储库。

如果分支成功发布到远程存储库，则 Git 面板底部的分支名称旁边会显示同步符号。选择它以同步本地和远程存储库的内容。



在本地和远程存储库之间推送和拉取内容

使用 Git 协作处理共享项目时，您通常首先将其他团队成员的最新更改从远程存储库拉取到本地存储库。在您提交对本地存储库的更改后，您可以将其推送到远程存储库，以便团队的其他成员对其进行访问。这些操作由命令 `git pull` 和 `git push` 执行。

Note

在向大多数托管存储库（例如 GitHub 上的存储库）推送更改和从中拉取更改时，您需要输入登录凭证。

Pull changes from remote

通过 Git 面板界面使用 `git pull` 命令，您可以使用提交到远程存储库中分支的最新更改来更新本地存储库。

1. 在 Git panel (Git 面板) 菜单中，选择 Checkout to (签出到)。
2. 在分支列表中，选择要将更改提交到其中的本地分支。
3. 接下来，转至 Git panel (Git 面板) 菜单，然后选择 Pull from (从中拉取)。
4. 选择一个远程存储库，然后选择该存储库中的一个分支以从中拉取更改。

执行拉取操作后，您可以在存储库工作目录中访问从远程存储库检索的文件。修改文件后，您可以将更改推送到远程分支。

Push changes to remote

通过 Git 面板界面使用 `git push` 命令，您可以使用本地存储库中指定分支中的最新更改来更新远程存储库。

1. 在 Git panel (Git 面板) 菜单中，选择 Checkout to (签出到)。
2. 在分支列表中，选择要从中推送更改的本地分支。
3. 接下来，转到 Git panel (Git 面板) 菜单，然后选择 Push to (推送到)。
4. 选择一个远程存储库，然后选择该存储库中的一个分支来推送更改。

执行推送后，其他团队成员可以通过将更改下拉到他们自己的存储库本地副本来访问您的更改。

存储和检索文件

借助 Git 的存储功能，您无需预先提交暂存或修改的文件即可切换分支。存储功能会捕获工作目录和暂存区的当前状态，并将其保存以供日后使用。当您仍在处理未完成的内容并需要立即切换分支时，此功能非常有用。

存储工作

1. 要存储您的工作目录的当前状态，请转到 Git panel (Git 面板) 菜单，然后选择以下任一选项：
 - Stash (存储)：将工作目录中的所有修改或暂存文件都添加到存储。不会添加未跟踪的文件。
 - Stash (include Untracked) (存储 (包括未跟踪))：将工作目录中的所有文件 (包括那些尚未跟踪的文件) 都添加到存储。
2. 输入可帮助您识别存储以供将来检索的可选消息。

存储后，Git 面板界面会刷新以显示已清理的工作目录。

检索存储

1. 要检索存储并将其应用到您的工作目录，请转到 Git panel (Git 面板) 菜单，然后选择以下任一选项：
 - Apply Stash (应用存储)：将选定的存储应用到您的工作目录，并保留存储以供日后使用。
 - Pop Stash (弹出存储)：将选定的存储应用到您的工作目录，并从存储堆栈中删除该存储。

Note

您还可以选择应用或弹出添加到存储堆栈的最后一个存储。

2. 选择要应用于工作目录的存储。

Git 面板界面会刷新，以显示已应用存储的工作目录。

参考：Git 面板中可用的 Git 命令

AWS Cloud9 的 Git 面板菜单提供了用户界面对核心和高级 git 命令的便捷访问。

某些 git 命令 (例如用于合并和删除分支的命令) 只能通过 Git 面板搜索字段使用。

您还可以自定义 Git 面板运行命令以及与存储库交互的方式。要修改默认设置，请首先选择 AWS Cloud9 > Preferences (首选项)。然后，在 Preferences (首选项) 窗口中，在 Project Settings (项目设置) 下方，选择 Git。

在信息图标上暂停以阅读设置的简要说明。

The screenshot shows the AWS Cloud9 Preferences window, specifically the Project Settings section for Git. The settings are as follows:

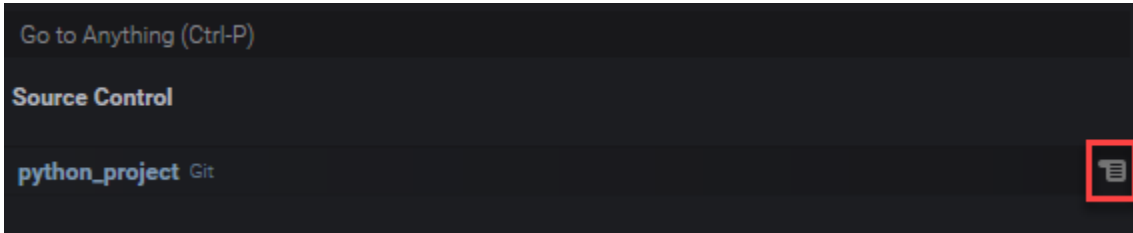
- Aws: Log Level:** Errors, Warnings, and Info
- Aws: Telemetry:** Enabled (toggle)
- Git: Enabled:** Enabled (toggle). A yellow highlight box surrounds this setting with the text "Whether git is enabled." and an information icon.
- Git: Path:** Edit in project.settings
- Git: Auto Repository Detection:** Scan for both subfolders of t (dropdown)
- Git: Autorefresh:** Enabled (toggle)
- Git: Autofetch:** Disabled (toggle)
- Git: Autofetch Period:** 180 (input field)
- Git: Branch Validation Regex:** (input field)
- Git: Branch Whitespace Char:** - (input field)
- Git: Confirm Sync:** Enabled (toggle)
- Git: Count Badge:** Count all changes. (dropdown)
- Git: Checkout Type:** Show all references. (dropdown)
- Git: Ignore Legacy Warning:** Disabled (toggle)
- Git: Ignore Missing Git Warning:** Disabled (toggle)
- Git: Ignore Limit Warning:** Disabled (toggle)
- Git: Default Clone Directory:** (input field)

Note

您可以访问 Git 官方网站 (<https://git-scm.com/doc>) 上列出的 Git 命令的详细文档。

Git 面板菜单中可用的 Git 命令的参考

您可以通过选择与存储库名称相反的符号来访问 Git 面板菜单上的选项。



Git 面板菜单

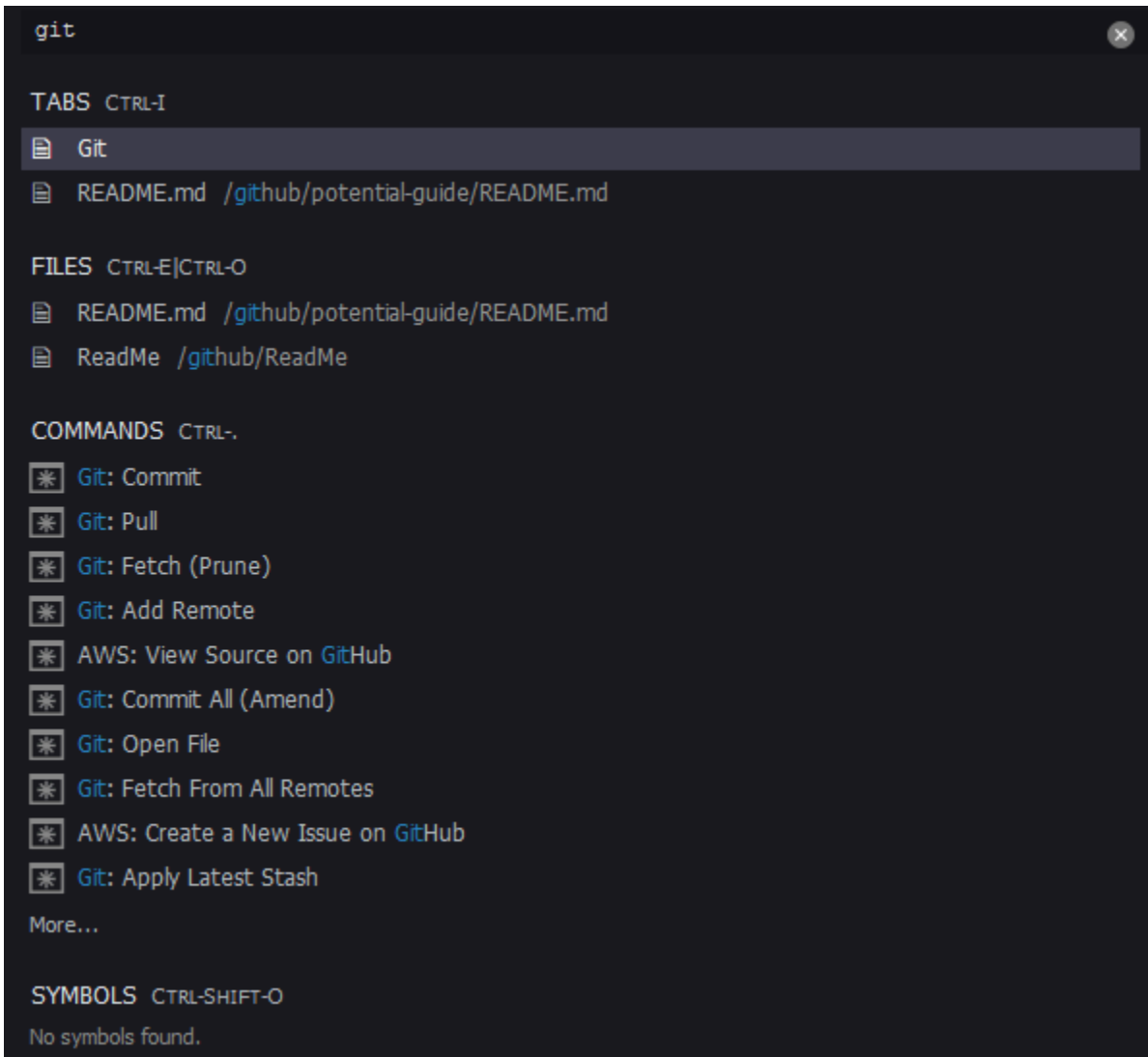
菜单选项	描述
提交	将添加到暂存区的内容提交到存储库的工作目录。添加提交消息。
刷新	刷新 GitPanel 界面以显示工作目录和暂存区的状态。
拉取	将最新的更改从远程存储库拉取到本地存储库。
拉取 (变基)	将本地更改重新应用于从远程分支拉取的远程更改。
从...推送	将提交给本地存储库中分支的更改推送到远程存储库中的分支。
Push	将提交到本地存储库的更改推送到远程存储库。
推送到...	将提交给本地存储库中分支的更改推送到远程存储库中的分支。
同步	运行 <code>git pull</code> 命令后再运行 <code>git push</code> 命令，以同步本地和远程存储库的内容。
签出到...	切换到现有分支或创建分支并切换到该分支。

菜单选项	描述
发布分支	将在本地存储库上创建的私有分支发布，并使其在远程存储库中可用。
全部提交	将暂存文件和未暂存文件均提交到存储库。(在 <code>git commit</code> 命令运行前， <code>git add -A</code> 命令将文件添加到暂存区。)
全部提交 (修改)	修改上次提交的消息。(运行 <code>git commit</code> 命令时添加 <code>-amend</code> 选项。)
全部提交 (已签字)	标识在 Git 日志中执行了提交的人员。(运行 <code>git commit</code> 命令时添加 <code>-signed-off</code> 选项。)
提交已暂存	仅将暂存文件提交到存储库。
提交已暂存 (修改)	修改上次提交的消息。(运行 <code>git commit</code> 命令时添加 <code>-amend</code> 选项。)
提交已暂存 (已签字)	标识在 Git 日志中执行了提交的人员。(运行 <code>git commit</code> 命令时添加 <code>-signed-off</code> 选项。)
Undo Last Commit (撤销上次提交)	撤消上一次提交。文件将被移回暂存区。
丢弃所有更改	从存储库的暂存区域中删除所有文件和文件夹。
暂存所有更改	将未跟踪和已修改的内容添加到暂存区。
取消暂存所有更改	将所有文件移出暂存区。未暂存的文件将无法提交到存储库。
应用最新存储	将添加到堆栈存储的最近一个存储应用到工作目录。该存储将保留在堆栈上。
应用存储...	将从存储堆栈选择的存储应用到工作目录。该存储将保留在堆栈上。

菜单选项	描述
弹出最新存储	将添加到堆栈存储的最近一个存储应用到工作目录。该存储随后将从堆栈中删除。
弹出存储...	将所选存储应用到工作目录。该存储随后将从堆栈中删除。
存储	将工作目录中的已修改和已暂存文件添加到已命名的存储中。
存储 (包括未跟踪)	将工作目录中的所有文件 (包括未跟踪的文件) 添加到已命名的存储中。
显示 Git 输出	显示一个窗口, 以显示当您与 Git 面板界面交互时运行的 Git 命令。

Git 面板搜索字段中可用的 Git 命令

您还可以通过在搜索框中键入“git”来访问一些受支持但在 Git 面板菜单中不可用的 Git 命令：



下表提供了您可以用以访问所选 Git 命令的方式说明。

所选 Git 命令

菜单选项	描述
Git : 添加远程	将远程存储库的连接添加到 Git Config 文件
Git : 删除分支	删除指定分支。
Git : 取回	从远程存储库中的分支下载内容。相反，有了 <code>git pull</code> ，远程更改不会合并到本地存储库中。

菜单选项	描述
Git : 合并分支	将一个分支中所做的更改集成到另一个分支中。有关更多信息，请参阅 合并分支流程 。

AWS Toolkit

为什么使用 AWS Toolkit？

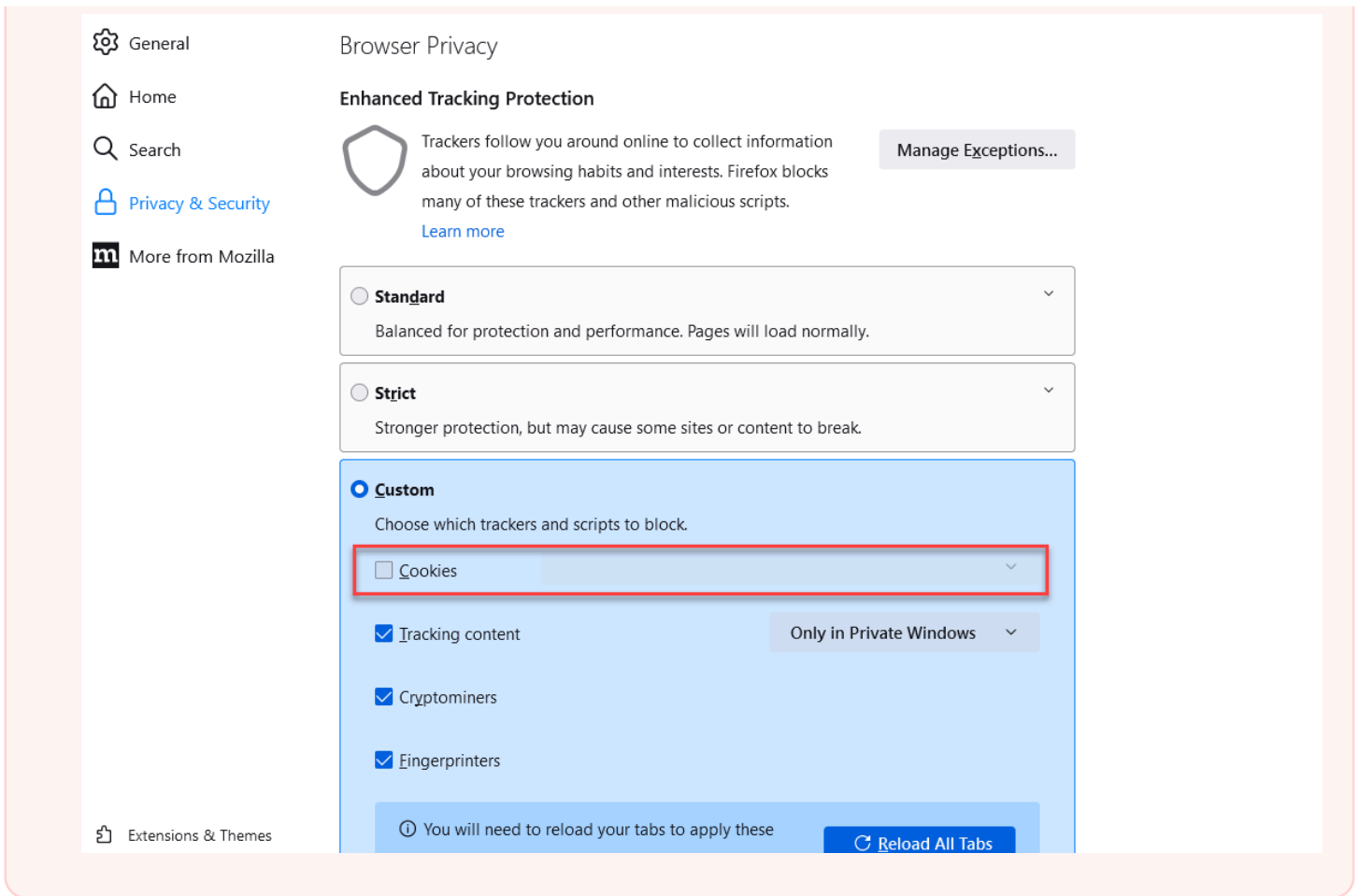
AWS Toolkit 是 AWS Cloud9 集成开发环境 (IDE) 的扩展。您可以通过此扩展来访问和处理广泛的 AWS 服务。AWS Toolkit 替换了 AWS Cloud9 中 Lambda 插件提供的功能。有关更多信息，请参阅[禁用 AWS Toolkit](#)。

Important

AWS Toolkit 支持是 AWS Cloud9 的集成功能。目前，您无法使用第三方扩展来自定义 AWS Cloud9 IDE。

Warning

如果您将 Mozilla Firefox 作为首选浏览器与 AWS Cloud9 IDE 一起使用，第三方 Cookie 设置会阻止 AWS Cloud9 Webview 和 AWS Toolkit 在浏览器中正常运行。要解决此问题，您必须确保在浏览器设置的“隐私和安全”部分没有阻止 Cookie，如下图所示。



目前可以通过 AWS Toolkit 扩展访问以下 AWS 服务和资源：

- [AWS App Runner](#)
- [API Gateway](#)
- [AWS CloudFormation 堆栈](#)
- [CloudWatch Logs](#)
- [AWS Lambda](#)
- [资源](#)
- [Amazon S3 存储桶和对象](#)
- [AWS Serverless Application Model 应用程序](#)
- [Step Functions 和状态机](#)
- [Systems Manager 自动化文档](#)
- [在 AWS Cloud9 IDE 中使用 Amazon ECR](#)

- [AWS IoT](#)
- [???](#)
- [Amazon EventBridge](#)
- [使用 Amazon CodeWhisperer](#)
- [使用 AWS Cloud Development Kit \(AWS CDK\)](#)

启用 AWS Toolkit

如果 AWS Toolkit 在您的环境中不可用，您可以在 Preferences (首选项) 选项卡中将其启用。

启用 AWS Toolkit

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 AWS Settings (Amazon 设置)。
3. 在 AWS 资源窗格中，启用 AWS Toolkit，以便在绿色背景上显示复选标记。

启用 AWS Toolkit 时，集成式开发环境 (IDE) 将刷新，以显示更新后的启用 AWS Toolkit 设置。还会在 Environment (环境) 选项下方的 IDE 旁边显示 AWS Toolkit 选项。

Important

如果您的 AWS Cloud9 环境的 EC2 实例无法访问 Internet (即不允许出站流量)，则在您启用 AWS Toolkit 并重新启动 IDE 后可能会显示一条消息。此消息指出 AWS Toolkit 所需的依赖项无法下载。如果是这种情况，您也不能使用 AWS Toolkit。

要解决此问题，请为 Amazon S3 创建 VPC 终端节点。这会授予对您 AWS 区域中 Amazon S3 存储桶的访问权限，该存储桶中包含使 IDE 保持最新状态所需的依赖项。

有关更多信息，请参阅 [Amazon S3 配置 VPC 终端节点以下载依赖项](#)。

管理 AWS Toolkit 的访问凭证

AWS Toolkit 可与广泛的 AWS 服务 进行交互。要管理访问权限控制，请确保 AWS Toolkit 服务的 IAM 实体拥有这一系列服务的必要权限。如需快速开始，请使用 [AWS 托管式临时凭证](#) 来获得必要的权限。这些托管凭证的工作方式是代表 AWS 实体 (例如 IAM 用户)，向 AWS 服务 授予对您 EC2 环境的访问权限。

但是，如果您已将开发环境的 EC2 实例启动到私有子网中，则 AWS 托管式临时凭证对您将不可用。作为替代方案，您可以通过手动创建您自己的一组凭证，允许 AWS Toolkit 访问您的 AWS 服务。这组凭证称为配置文件。配置文件具有称为访问密钥的长期凭证。您可以从 IAM 控制台获取这些访问密钥。

创建配置文件以提供 AWS Toolkit 的访问凭证

1. 要获取您的访问密钥（由访问密钥 ID 和秘密访问密钥组成），请通过 <https://console.aws.amazon.com/iam> 转到 IAM 控制台。
2. 从导航栏中选择 Users（用户），然后选择您的 AWS 用户名（不是复选框）。
3. 选择 Security Credentials（安全凭证）选项卡，然后选择 Create access key（创建访问密钥）。

Note

如果您已经拥有访问密钥但无法访问您的私有密钥，请将旧密钥设为非活动状态并创建一个新密钥。

4. 在显示您的访问密钥 ID 和秘密访问密钥的对话框中，选择 Download .csv file（下载 .csv 文件）以将此信息存储在安全位置。
5. 下载访问密钥后，启动 AWS Cloud9 环境并通过选择 Window（窗口）、New Terminal（新终端）启动终端会话。
6. 在终端窗口中，运行以下命令。

```
aws configure --profile toolkituser
```

在本例中，toolkituser 是正在使用的配置文件名称，但您可以选择自己的配置文件名称。

7. 在命令行中，输入您之前从 IAM 控制台下载的 AWS Access Key ID 和 AWS Secret Access Key。
 - 对于 Default region name，请指定 AWS 区域（例如，us-east-1）。
 - 对于 Default output format，请指定文件格式（例如，json）。

Note

有关对配置文件进行配置时选项的更多信息，请参阅《AWS Command Line Interface 用户指南》中的[配置基础](#)。

- 创建配置文件后，启动 AWS Toolkit，转到 [AWS Toolkit 菜单](#)，然后选择连接到 AWS。
- 对于选择 AWS 凭证配置文件字段，选择您刚刚在终端中创建的配置文件（例如 `profile:toolkituser`）。

如果所选配置文件包含有效的访问凭证，则 AWS Explorer 窗格将刷新以显示您现在可以访问的 AWS 服务。

使用 IAM 角色向在 EC2 实例上运行的应用程序授予权限

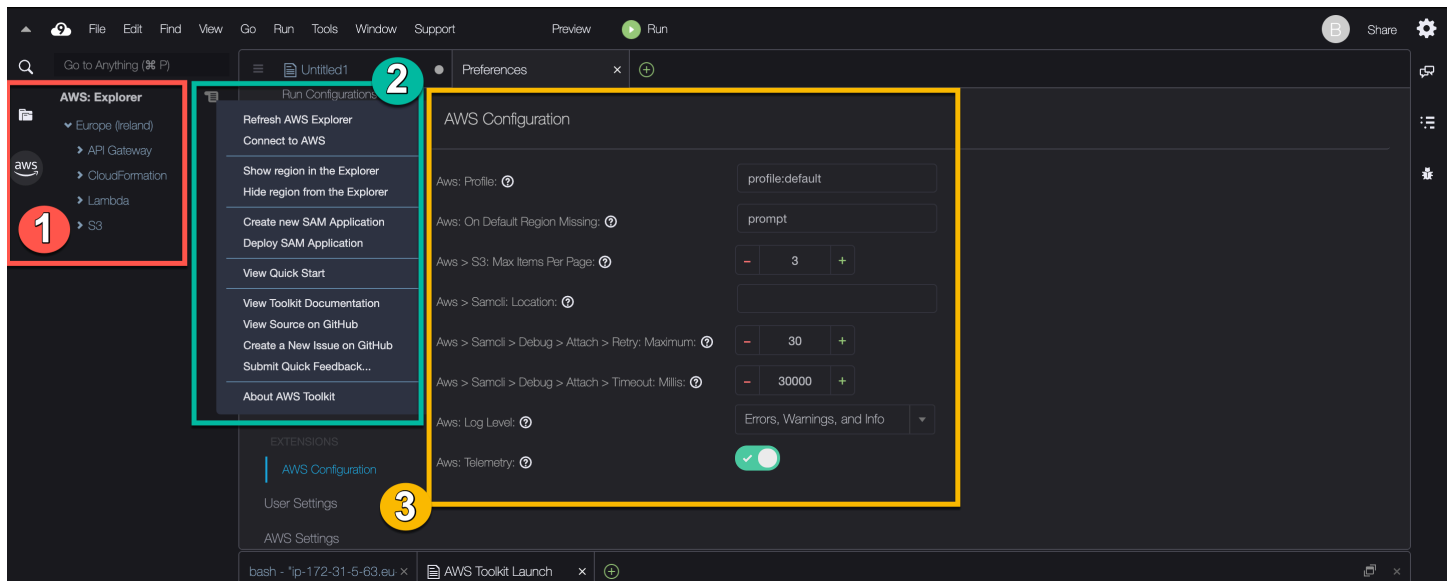
您也可以使用 IAM 角色管理在 EC2 实例上运行的应用程序的临时凭证。此角色可提供临时权限供应用程序在调用其他 AWS 资源时使用。当您启动 EC2 实例时，可指定要与实例关联的 IAM 角色。在针对 AWS 服务发出 API 请求时，在实例上运行的应用程序可以使用角色提供的临时凭证。

在您创建角色后，通过创建实例配置文件，将此角色及其关联的权限分配给实例。实例配置文件附加到实例，并且可以向在实例上运行的应用程序提供角色的临时凭证。

有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

识别 AWS Toolkit 组件

下面的屏幕截图显示了 AWS Toolkit 的三个关键 UI 组件：



1. AWS Explorer 窗口：用于通过 Toolkit 与可访问的 AWS 服务 进行交互。您可以使用集成式开发环境（IDE）左侧的 AWS 选项，在显示和隐藏 AWS Explorer 之间切换。有关使用此接口组件和访问

- 不同 AWS 区域的 AWS 服务的更多信息，请参阅[使用 AWS Explorer 处理多个区域中的服务和资源](#)。
2. Toolkit 菜单：用于管理与 AWS 的连接、自定义 AWS Explorer 窗口的显示、创建和部署无服务器应用程序、使用 GitHub 存储库以及访问文档。有关更多信息，请参阅[访问和使用 AWS Toolkit 菜单](#)。
 3. AWS 配置窗格：用于自定义通过 Toolkit 与 AWS 服务交互的行为。有关更多信息，请参阅[使用“AWS Configuration \(Amazon 配置 \)”窗格修改 AWS Toolkit 设置](#)。

禁用 AWS Toolkit

您可以在 Preferences (首选项) 选项卡中禁用 AWS Toolkit。

禁用 AWS Toolkit

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项)。
2. 在 Preferences (首选项) 选项卡上，在侧面导航窗格中选择 AWS Settings (Amazon 设置)。
3. 在 AWS Resources (Amazon 资源) 窗格中，关闭 AWS AWS Toolkit。

当您禁用 AWS Toolkit 时，集成式开发环境 (IDE) 会刷新以移除 Environment (环境) 选项下方 IDE 旁边的 AWS Toolkit 选项。

AWS Toolkit 主题

- [导航和配置 AWS Toolkit](#)
- [将 AWS App Runner 与 AWS Toolkit 配合使用](#)
- [通过 AWS Toolkit 使用 API Gateway](#)
- [通过 AWS CloudFormation Toolkit 使用 AWS 堆栈](#)
- [通过 AWS Toolkit 使用 AWS Lambda 函数](#)
- [使用资源](#)
- [通过 AWS Toolkit 使用 Amazon S3](#)
- [通过 AWS Toolkit 使用 AWS 无服务器应用程序](#)
- [与亚马逊合作 CodeCatalyst](#)

- [在 AWS Cloud9 IDE 中使用 Amazon ECR](#)

导航和配置 AWS Toolkit

您可以通过以下 AWS Toolkit 界面元素访问资源和修改设置：

- [AWS Explorer 窗口](#)：从不同 AWS 区域 访问 AWS 服务。
- [AWS Toolkit 菜单](#)：创建和部署无服务器应用程序、显示或隐藏 AWS 区域、访问用户帮助以及与 Git 存储库交互。
- [AWS Configuration \(Amazon 配置 \) 窗格](#)：修改影响您与 AWS Toolkit 中的 AWS 服务 的交互方式的设置。

使用 AWS Explorer 处理多个区域中的服务和资源

使用 AWS Explorer 窗口，您可以选择 AWS 服务 并使用与该服务关联的特定资源。在 AWS Explorer 中，选择服务名称节点（例如，API Gateway 或 Lambda）。然后，选择与该服务关联的特定资源（例如，REST API 或 Lambda 函数）。当您选择特定资源时，菜单会显示可用的交互选项，例如上传或下载、调用或复制。

考虑以下 示例。如果您的 AWS 账户 账户凭证可以访问 Lambda 函数，则展开为某个 AWS 区域 列出的 Lambda 节点，然后选择可以作为代码调用或上传到 AWS Cloud9 IDE 的特定 Lambda 函数。您还可以打开节点标题的上下文（右键单击）菜单，开始创建使用 AWS Serverless Application Model 的应用程序。

Note

如果您在集成式开发环境（IDE）中没有看到查看 AWS Explorer 窗口的选项，请确保您已启用 AWS Toolkit。然后，在确认已启用后，再试一次。有关更多信息，请参阅[启用 AWS Toolkit](#)。

AWS Explorer 窗口还可以显示托管在多个 AWS 区域 中的服务。

从选定的区域访问 AWS 服务

1. 在 AWS Explorer 窗口中，依次选择 Toolkit 菜单、Show region in the Explorer（在 Explorer 中显示区域）。
2. 在在 AWS Explorer 中选择区域以显示列表中，选择一个 AWS 区域。

选定的区域将添加到 AWS Explorer 窗口。要访问可用的服务和资源，请选择 区域名称前面的箭头 (>)。

Note

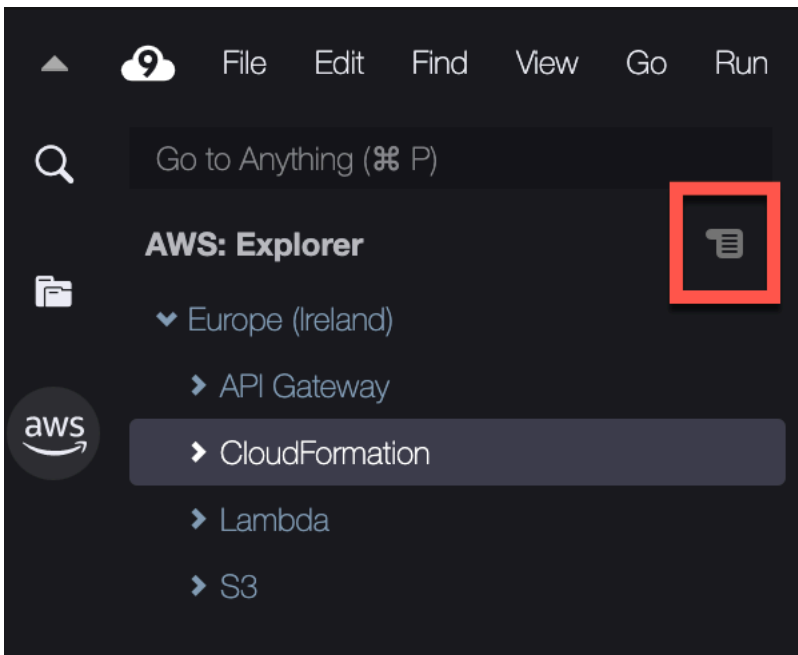
您还可以使用以下选项隐藏 AWS Explorer 窗口中选定的 AWS 区域：

- 打开区域的上下文（右键单击）菜单，然后选择 Hide region from the Explorer（从 Explorer 中隐藏区域）。
- 在 AWS Toolkit 菜单中，选择 Hide region from the Explorer（从 Explorer 中隐藏区域）并选择要隐藏的区域。

访问和使用 AWS Toolkit 菜单

AWS Toolkit 支持对创建和部署[无服务器应用程序](#)选项的访问。您可以使用此菜单管理连接、更新 AWS: Explorer 窗口、访问文档以及与 GitHub 存储库交互。

要访问 Toolkit 菜单，请选择 AWS Explorer 窗口中与 AWS: Explorer 标题相反方向的滚动图标。



下表概述了 Toolkit 菜单上的可用选项。

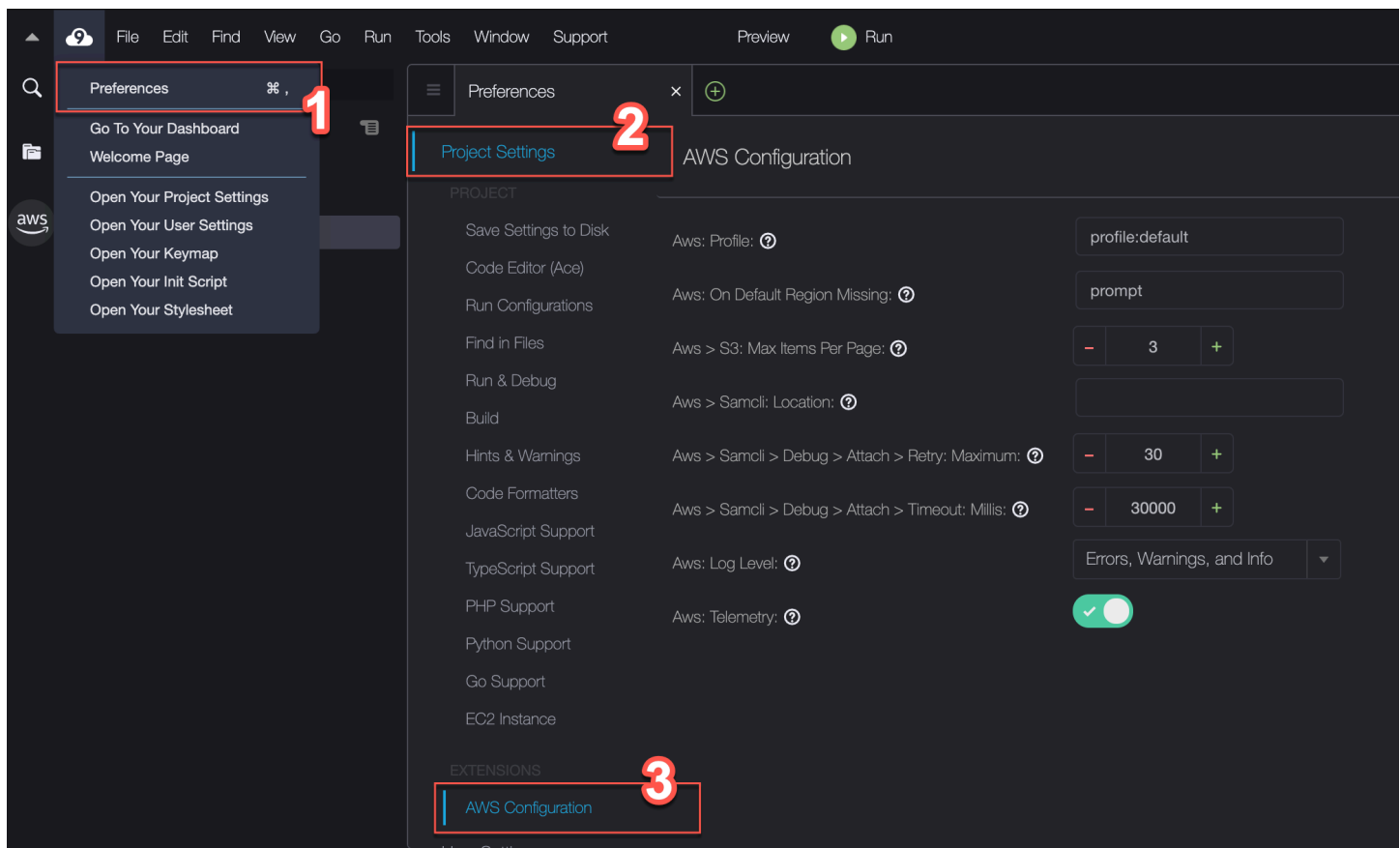
工具包菜单选项

菜单选项	描述
刷新 AWS Explorer	选择此选项可刷新 AWS Explorer 以显示自上次您打开此窗口以来修改过的所有 AWS 服务。
连接到 AWS	使用存储在配置文件中的凭证将 AWS Toolkit 连接到 AWS 账户。有关更多信息，请参阅 管理 AWS Toolkit 的访问凭证 。
在 Explorer 中显示区域	在 AWS Explorer 窗口中显示 AWS 区域。有关更多信息，请参阅 使用 AWS Explorer 处理多个区域中的服务和资源 。
从 Explorer 中隐藏区域	在 AWS Explorer 窗口中隐藏 AWS 区域。有关更多信息，请参阅 使用 AWS Explorer 处理多个区域中的服务和资源 。
创建新的 SAM 应用程序	为新的 AWS 无服务器应用程序生成一组代码文件。有关如何创建和部署 SAM 应用程序的更多信息，请参阅 通过 AWS Toolkit 使用 AWS 无服务器应用程序 。
部署 SAM 应用程序	将无服务器应用程序部署到 AWS。有关如何创建和部署 SAM 应用程序的更多信息，请参阅 通过 AWS Toolkit 使用 AWS 无服务器应用程序 。
查看快速入门	打开此快速入门指南。
查看 Toolkit 文档	打开 AWS Toolkit 的用户指南。
查看 GitHub 上的源	打开 AWS Toolkit 的 GitHub 存储库。
在 GitHub 上创建一个新问题	在 Github 上打开 AWS Toolkit 的“New Issue (新问题)”页面
提交快速反馈	向 AWS Toolkit 开发团队提交私密的单向反馈。对于需要对话或错误修复的问题，请通过选择

菜单选项	描述
	Create a New Issue on Github (在 Github 上创建新问题) 菜单选项，在 Github 中提交问题。
关于 AWS Toolkit	显示有关正在运行的 Toolkit 版本及其配置的 Amazon 操作系统的信息。

使用“AWS Configuration (Amazon 配置)”窗格修改 AWS Toolkit 设置

要访问 AWS Configuration (Amazon 配置) 窗格，依次选择 AWS Cloud9、Preferences (首选项)。然后，在 Preferences (首选项) 窗口中，在 Project Settings (项目设置) 下，选择 AWS Configuration (Amazon 配置)。



下表概述了 AWS 配置菜单上的可用选项。

菜单选项	描述
AWS : 配置文件	设置要从中获取凭证的凭证配置文件的名称。
AWS : 默认区域缺失	<p>指示当所选凭证配置文件的默认 AWS 区域在 AWS Explorer 窗口中不可用时要采取的操作。您可以从三个选项中进行选择：</p> <ul style="list-style-type: none"> • prompt (提示) (默认设置) : 询问您想进行何种操作。 • add (添加) : 区域显示在 AWS Explorer 窗口中。 • ignore (忽略) : 不采取任何操作。
AWS > S3: Max Items Per Page (S3 : 每页最大项目数)	<p>指定在 AWS Explorer 窗口中一次显示多少 Amazon S3 对象或文件夹。当显示最大数量时，您可以选择 Load More (加载更多) 以显示下一个批次。</p> <p>该字段接受值的范围介于 3 到 1000 之间。此设置仅适用于同时显示的对象或文件夹的数量。您创建的所有存储桶都会一次性显示。默认情况下，您可以在每个 AWS 账户中创建多达 100 个桶。</p>
AWS > Samcli: Location (Samcli : 位置)	指示用于创建、构建、打包和部署 无服务器应用程序 的 SAM CLI 的位置。
AWS > Samcli > Debug > Attach > Retry: Maximum: (Samcli > 调试 > 连接 > 重试 : 最大值 :)	<p>指定在放弃之前，Toolkit 尝试连接 SAM CLI 调试程序的次数。默认限制为 30 次尝试。</p> <p>当您在 AWS SAMCLI 中以调试模式在本地调用 Lambda 函数时，您可以将其连接到调试程序。</p>
AWS > Samcli > Debug > Attach > Timeout: Millis: (Samcli > 调试 > 连接 > 超时 : Millis :)	指定在放弃之前，Toolkit 尝试连接 SAM CLI 调试程序的时间。默认超时为 30000 毫秒 (30 秒)。

菜单选项	描述
	当您在 AWS SAMCLI 中以调试模式在本地调用 Lambda 函数时，您可以将其连接到调试程序。
AWS : 日志级别 ;	<p>设置录入的工作流事件的类别。可用级别如下：</p> <ul style="list-style-type: none"> • 仅错误 • 错误和警告 • 错误、警告和信息（默认选项） • 错误、警告和信息、详细和调试
AWS : 遥测	启用或禁用将使用数据发送到 AWS。默认情况下启用

通过 AWS Toolkit 使用 API Gateway

您可以使用 API Gateway 来创建 RESTful API 和 WebSocket API，以实现实时双向通信应用程序。有关使用 API Gateway 创建和管理 API 的更多信息，请参阅 [API Gateway 开发人员指南](#)。

使用 AWS Toolkit，您可以通过指定 REST 资源、方法类型和作为输入传入的数据来配置对 REST API 的调用。

调用 API Gateway 中的 REST API

Important

使用 AWS Toolkit 调用 API 方法可能会对资源导致无法撤消的更改。例如，如果您调用 POST 方法，则调用成功时，API 的资源将更新。

您可以从 AWS Toolkit 在 AWS 上调用 API Gateway。

调用 REST API

1. 在 AWS Explorer 窗口中，选择 API Gateway 节点以查看当前 AWS 区域中可用的 REST API 列表。
2. 右键单击 REST API，然后选择 Invoke on AWS（在 Amazon 上调用）。

Note

您可以使用上下文菜单复制 REST API 的 URL、名称和 Amazon 资源名称 (ARN)。

此时将显示 Invoke methods (调用方法) 窗口。您可以配置对 API 的调用。

3. 对于 Select a resource (选择资源)，选择要与之交互的 REST 资源。
4. 对于 Select authentication method (选择身份验证方法)，请选择以下方法之一：
 - GET：从通过 API 访问的后端服务获取资源。
 - OPTIONS：请求有关 API Gateway 支持的方法和操作的信息。
 - POST：在通过 API 访问的后端服务中创建新资源。
5. 要向 API 方法调用提供输入，可以使用查询字符串或 JSON 格式的负载：
 - 查询字符串：使用格式 `parameter1=value1¶meter2=value2` 输入查询字符串。(在使用查询字符串之前，请创建[映射模板](#)，以便在传入的 Web 请求发送到集成后端之前对其进行转换。)
 - JSON 格式：您可以在 Invoke methods (调用方法) 窗口中的大文本字段中定义 JSON 格式的负载。

例如，您可以利用包含以下负载的 POST 方法来添加新的资源：

```
{"type": "soda", "price" : 3.99}
```

6. 选择 Invoke (调用) 按钮调用 REST API 资源。

REST API 响应将显示在 AWS Remote Invocations (Amazon 远程调用) 选项卡中。响应正文包含 JSON 格式的资源数据。

将 AWS App Runner 与 AWS Toolkit 配合使用

[AWS App Runner](#) 提供了一种快速且经济高效的方式，从源代码或容器镜像直接部署到 AWS Cloud 中可扩展的安全 Web 应用程序。有了它，您不需要学习新技术、决定要使用的计算服务，也不需要了解如何预置和配置 AWS 资源。

您可以使用 AWS App Runner 创建和管理基于源镜像或源代码的服务。如果您使用源镜像，则可以选择存储在镜像存储库中的公有或私有容器镜像。App Runner 支持以下镜像存储库提供商：

- Amazon Elastic Container Registry (Amazon ECR) : 在您的 AWS 账户 中存储私有镜像。
- Amazon Elastic Container Registry Public (Amazon ECR Public) : 存储公开可读的镜像。

如果选择源代码选项，则可以从受支持的存储库提供商维护的源代码存储库进行部署。目前，App Runner 支持将 [GitHub](#) 作为源代码存储库提供商。

先决条件

要使用 AWS Toolkit 与 App Runner 交互，需要具有：

- 一个 AWS 账户
- 具有 AWS App Runner 的 AWS Toolkit 版本

除了这些核心要求之外，请确保所有相关的 IAM 用户都有权与 App Runner 服务进行交互。另外请确保您获取有关服务源的特定信息，例如容器镜像 URI 和与 GitHub 存储库的连接。创建 App Runner 服务时，您需要使用此信息。

为 App Runner 配置 IAM 权限

要快速授予 App Runner 所需的权限，请将现有的 AWS 托管策略附加到相关的 AWS Identity and Access Management (IAM) 实体。特别是，您可以将策略附加到用户或组。App Runner 提供两种可附加到 IAM 用户的托管策略：

- `AWSAppRunnerFullAccess` : 允许用户执行所有 App Runner 操作。
- `AWSAppRunnerReadOnlyAccess` : 允许用户列出和查看有关 App Runner 资源的详细信息。

如果您从 Amazon Elastic Container Registry (Amazon ECR) 中选择私有存储库作为服务源，则必须为 App Runner 服务创建以下访问角色：

- `AWSAppRunnerServicePolicyForECRAccess` : 允许 App Runner 访问您账户中的 Amazon Elastic Container Registry (Amazon ECR) 镜像。

使用 AWS Toolkit 的命令窗格配置服务实例时，您可以自动创建此角色。

Note

`AWSServiceRoleForAppRunner` 服务相关角色允许 AWS App Runner 完成以下操作：

- 将日志推送到 Amazon CloudWatch Logs 日志组。
- 创建 Amazon CloudWatch Events 规则，以订阅 Amazon Elastic Container Registry (Amazon ECR) 镜像推送。

无需手动创建服务相关角色。当您在 AWS Management Console 中创建 AWS App Runner 或使用 AWS Toolkit 调用的 API 操作时，AWS App Runner 将为您创建此服务相关角色。

有关更多信息，请参阅 AWS App Runner 开发人员指南中的[适用于 App Runner 的 Identity and Access Management](#)。

获取 App Runner 的服务源

您可以使用 AWS App Runner 从源镜像或源代码部署服务。

Source image

如果您从源镜像进行部署，则可以从私有或公有 AWS 镜像注册表中获取指向该镜像存储库的链接。

- Amazon ECR 私有注册表：复制使用 Amazon ECR 控制台的私有存储库的 URI：<https://console.aws.amazon.com/ecr/repositories>。
- Amazon ECR 公共注册表：复制使用 Amazon ECR Public Gallery 的公有存储库的 URI：<https://gallery.ecr.aws/>。

Note

您还可以直接从 AWS Toolkit 中的 AWS Explorer 获取私有 Amazon ECR 存储库的 URI：

- 打开 AWS Explorer，然后展开 ECR 节点以查看该 AWS 区域的存储库列表。
- 打开存储的上下文（右键单击）菜单，然后选择 Copy Repository URI（复制存储库 URI）以将链接复制到剪贴板。

使用 AWS Toolkit 的命令窗格配置服务实例时，您可以指定镜像存储库的 URI。

有关更多信息，请参阅 AWS App Runner 开发人员指南中的[基于源镜像的 App Runner 服务](#)。

Source code

要将源代码部署到 AWS App Runner 服务，该代码必须存储在 Git 存储库中。此 Git 存储库必须由支持的存储库提供商维护。App Runner 支持一个源代码存储库提供商：[GitHub](#)。

有关设置 GitHub 存储库的信息，请参阅 GitHub 上的[入门文档](#)。

要将源代码从 GitHub 存储库部署到 App Runner 服务，App Runner 需要建立与 GitHub 的连接。如果您的存储库为私有（即它不能在 GitHub 上公开访问），则必须向 App Runner 提供连接详细信息。

Important

要创建 GitHub 连接，必须使用 App Runner 控制台 (<https://console.aws.amazon.com/apprunner>) 创建将 GitHub 链接到 AWS 的连接。使用 AWS Toolkit 的命令窗格配置服务实例时，您可以选择 GitHub connections (GitHub 连接) 页面上可用的连接。

有关更多信息，请参阅 AWS App Runner 开发人员指南中的[管理 App Runner 连接](#)。

App Runner 服务实例提供了允许构建和运行代码的托管运行时。AWS App Runner 目前支持以下运行时：

- Python 托管运行时
- Node.js 托管运行时

作为服务配置的一部分，您可以提供有关 App Runner 服务如何构建和启动服务的信息。您可以使用命令调色板输入此信息，或指定 YAML 格式的 [App Runner 配置文件](#)。此文件中的值指示 App Runner 如何构建和启动服务以及提供运行时上下文。这包括相关的网络设置和环境变量。配置文件名为 `apprunner.yaml`。它会自动添加到应用程序存储库的根目录中。

定价

您需要为应用程序使用的计算和内存资源付费。此外，如果选择自动执行部署，则还需要为每个应用程序支付一笔固定的月费，其中涵盖该月的所有自动化部署。如果您选择从源代码进行部署，则需要为 App Runner 从源代码构建容器所用的时间支付构建费用。

有关更多信息，请参阅[AWS App Runner 定价](#)。

主题

- [创建 App Runner 服务](#)
- [管理 App Runner 服务](#)

创建 App Runner 服务

您可以使用 AWS Explorer 在 AWS Toolkit 中创建 App Runner 服务。选择在特定 AWS 区域 中创建服务后，AWS Toolkit 命令窗格描述如何配置运行您应用程序的服务实例。

在创建 App Runner 服务之前，请确保您已完成[先决条件](#)。这包括提供相关的 IAM 权限以及确认要部署的特定源存储库。

创建 App Runner 服务

1. 打开 AWS Explorer (如果尚未打开)。
2. 右键单击 App Runner 节点，然后选择 Create Service (创建服务)。

此时将显示 AWS Toolkit 命令窗格。

3. 对于 Select a source code location type (选择源代码位置类型)，请选择 ECR 或 Repository (存储库)。

如果选择 ECR，则可以在 Amazon Elastic Container Registry 维护的存储库中指定容器镜像。如果选择 Repository (存储库)，则可以指定由受支持的存储库提供商维护的源代码存储库。目前，App Runner 支持将 [GitHub](#) 作为源代码存储库提供商。

从 ECR 部署

1. 对于 Select or enter an image repository (选择或输入镜像存储库)，请选择或输入由 Amazon ECR 私有注册表或 Amazon ECR Public Gallery 维护的镜像存储库的 URL。

Note

如果您从 Amazon ECR Public Gallery 指定存储库，请确保关闭自动部署功能。App Runner 不支持为 ECR 公共存储库中的镜像进行自动部署。

默认情况下关闭自动部署功能。命令窗格标题上的图标显示一条对角线时即指示此状态。

如果选择开启自动部署，则会显示一条消息，通知您此选项可能会产生额外费用。

2. 如果命令窗格中的步骤报告 No tags found (未找到标签) ，则需要后退一步才能选择包含已标记的容器镜像的存储库。
3. 对于 Port (端口) ，输入服务使用的 IP 端口 (例如端口 8000) 。
4. (可选) 对于 Configure environment variables (配置环境变量) ，请指定一个文件，其中包含用于自定义服务实例中的行为的环境变量。
5. 如果您使用的是 Amazon ECR 私有注册表，则需要 AppRunnerECRAccessRole ECR 访问角色。此角色允许 App Runner 访问您账户中的 Amazon Elastic Container Registry (Amazon ECR) 镜像。选择命令窗格标题上的“+”图标以创建此角色。如果您的镜像存储在 Amazon ECR Public 中，其中镜像是公开可用的，则不需要访问角色。
6. 对于 Name your service (命名您的服务) ，请输入一个唯一名称，然后按 Enter。名称中不得包含空格。
7. 对于 Select instance configuration (选择实例配置) ，请为您的服务实例选择 CPU 单元和内存的组合 (均以 GB 为单位) 。

创建服务时，其状态将从正在创建更改为正在运行。

8. 服务开始运行后，打开其上下文 (右键单击) 菜单并选择 Copy Service URL (复制服务 URL) 。
9. 要访问已部署的应用程序，请将复制的 URL 粘贴到 Web 浏览器的地址栏中。

从远程存储库进行部署

1. 对于 Select a connection (选择连接) ，请选择将 GitHub 链接到 AWS 的连接。可供选择的连接列在 App Runner 控制台的 GitHub connections (GitHub 连接) 页面上。
2. 对于 Select a remote GitHub repository (选择远程 GitHub 存储库) ，请选择或输入远程存储库的 URL。

已配置 AWS Cloud9 源控制管理的远程存储库可供选择。如果存储库没有列出，您也可以粘贴指向存储库的链接。

3. 对于 Select a branch (选择分支) ，请选择要将源代码部署到哪个 Git 分支。
4. 对于 Choose configuration source (选择配置源) ，请指定希望如何定义运行时配置。

如果选择 Use configuration file (使用配置文件) ，则您的服务实例将通过 `apprunner.yaml` 配置文件定义的设置进行配置。此文件位于应用程序存储库的根目录中。

如果选择 Configure all settings here (在此处配置所有设置) ，请使用命令窗格指定以下项：

- Runtime (运行时) ：选择 Python 3 或 Nodejs 12。

- Build command (构建命令) : 输入命令以在服务实例的运行时环境中构建应用程序。
 - Start command (启动命令) : 输入命令以在服务实例的运行时环境中启动应用程序。
5. 对于 Port (端口) , 输入服务使用的 IP 端口 (例如 , 端口 8000) 。
 6. (可选) 对于 Configure environment variables (配置环境变量) , 请指定一个文件 , 其中包含用于自定义服务实例中的行为的环境变量。
 7. 对于 Name your service (命名您的服务) , 请输入一个唯一名称 , 然后按 Enter。名称中不得包含空格。
 8. 对于 Select instance configuration (选择实例配置) , 请为您的服务实例选择 CPU 单位和内存的组合 (以 GB 为单位) 。

在创建服务时 , 其状态将从 Creating (正在创建) 更改为 Running (正在运行) 。

9. 服务开始运行后 , 打开其上下文 (右键单击) 菜单并选择 Copy Service URL (复制服务 URL) 。
10. 要访问已部署的应用程序 , 请将复制的 URL 粘贴到 Web 浏览器的地址栏中。

Note

如果您尝试创建 App Runner 服务失败 , 则该服务将在 AWS Explorer 中显示状态 Create failed (创建失败) 。有关故障排除信息 , 请参阅《App Runner 开发人员指南》中的[服务创建失败时](#)。

管理 App Runner 服务

创建 App Runner 服务后 , 您可以使用 AWS Explorer 窗格来管理它 , 以执行以下活动 :

- [暂停和恢复 App Runner 服务](#)
- [部署 App Runner 服务](#)
- [查看 App Runner 的日志流](#)
- [删除 App Runner 服务](#)

暂停和恢复 App Runner 服务

如果您需要暂时禁用 Web 应用程序并停止代码运行 , 则可以暂停 AWS App Runner 服务。App Runner 会将服务的计算容量降至零。当你准备好再次运行应用程序时 , 请恢复 App Runner 服务。App Runner 将预置新的计算容量 , 为其部署应用程序 , 然后运行该应用程序。

Important

仅当 App Runner 运行时，您才需要为它付费。因此，您可以根据需要暂停和恢复应用程序，以便控制成本。这在开发和测试方案中特别有用。

暂停 App Runner 服务

1. 打开 AWS Explorer (如果尚未打开)。
2. 展开 App Runner 以查看服务列表。
3. 右键单击服务并选择 Pause (暂停)。
4. 在显示的对话框中，选择 Confirm (确认)。

在服务暂停期间，服务状态将从 Running (正在运行) 变为 Pausing (正在暂停)，然后变为 Paused (已暂停)。

恢复 App Runner 服务

1. 打开 AWS Explorer (如果尚未打开)。
2. 展开 App Runner 以查看服务列表。
3. 右键单击服务并选择 Resume (恢复)。

在服务恢复期间，服务状态将从 Resuming (正在恢复) 变为 Running (正在运行)。

部署 App Runner 服务

如果为服务选择手动部署选项，则需要明确启动服务的每个部署。

1. 打开 AWS Explorer (如果尚未打开)。
2. 展开 App Runner 以查看服务列表。
3. 右键单击服务并选择 Start Deployment (开始部署)。
4. 在应用程序部署期间，服务状态将从 Deploying (正在部署) 变为 Running (正在运行)。
5. 要确认应用程序已成功部署，请右键单击同一服务，然后选择 Copy Service URL (复制服务 URL)。
6. 要访问已部署的 Web 应用程序，请将复制的 URL 粘贴到 Web 浏览器的地址栏中。

查看 App Runner 的日志流

使用 CloudWatch Logs 监控、存储和访问 App Runner 等服务的日志流。日志流是共享同一来源的一系列日志事件。

1. 展开 App Runner 以查看服务实例列表。
2. 展开特定服务实例，以查看日志组列表。（日志组是一组具有相同保留期、监控和访问控制设置的日志流。）
3. 右键单击日志组并选择 View Log Streams（查看日志流）。
4. 从命令窗格中，请从组中选择日志流。

AWS Cloud9 IDE 将显示组成流的日志事件列表。您可以选择将较旧或更新的事件加载到编辑器中。

删除 App Runner 服务

Important

如果您删除 App Runner 服务，则它将被永久删除，并且您存储的数据也将被删除。如果您需要重新创建服务，则 App Runner 需要重新获取您的源代码并构建它（如果它是代码存储库）。您的 Web 应用程序将获得一个新的 App Runner 域。

1. 打开 AWS Explorer（如果尚未打开）。
2. 展开 App Runner 以查看服务列表。
3. 右键单击服务，然后选择 Delete Service（删除服务）。
4. 在 AWS Toolkit 命令窗格中，输入 delete，然后按 Enter 以确认。

已删除的服务将显示 Deleting（正在删除）状态，然后这些服务将从列表中消失。

通过 AWS CloudFormation Toolkit 使用 AWS 堆栈

AWS Toolkit 提供对 [AWS CloudFormation](#) 堆栈的支持。通过使用 AWS 工具包，您可以删除 AWS CloudFormation 堆栈。

删除 AWS CloudFormation 堆栈

您可以使用 AWS Toolkit 查看和删除 AWS CloudFormation 堆栈。

先决条件

- 确保您在 AWS Cloud9 环境中使用的凭证包含对 AWS CloudFormation 服务的适当读/写访问权限。如果在 AWS Explorer 中的 CloudFormation 下，您看到类似于“Error loading CloudFormation resources”（加载 CloudFormation 资源时出错）的消息，请检查附加到这些凭证的权限。对权限所做的更改需要几分钟时间才会对 AWS Explorer 生效。

删除 AWS CloudFormation 堆栈

1. 在 AWS Explorer 中，打开要删除的 AWS CloudFormation 堆栈的上下文（右键单击）菜单。
2. 选择 Delete CloudFormation Stack（删除 CloudFormation 堆栈）。
3. 在出现的消息中，选择 Yes（是）以确认删除。

删除堆栈后，它不再在 AWS Explorer 中列出。

通过 AWS Toolkit 使用 CloudWatch Logs

您可以使用 Amazon CloudWatch Logs，将所有系统、应用程序和您使用的 AWS 服务中的日志集中在单个具有高可扩展性的服务中。您可以轻松地查看它们、在其中搜索特定错误代码或模式、根据特定字段对其进行筛选，或者安全地将其归档以供将来分析。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的[什么是 Amazon CloudWatch Logs？](#)。

以下主题介绍通过 AWS Toolkit 使用 AWS 账户中的 CloudWatch Logs：

主题

- [使用 AWS Toolkit 查看 CloudWatch 日志组和日志流](#)
- [通过 AWS Toolkit 使用日志流中的 CloudWatch 日志事件](#)

使用 AWS Toolkit 查看 CloudWatch 日志组和日志流

日志流是共享同一个源的一系列日志事件。每个流向 CloudWatch Logs 的独立日志源构成一个独立的日志流。

日志组是一组具有相同保留期、监控和访问控制设置的日志流。您可以定义日志组并指定向各组中放入哪些流。可以属于一个日志组的日志流数没有限制。

有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南中的[使用日志组和日志流](#)。

主题

- [通过 CloudWatch Logs 节点查看 CloudWatch 日志组和日志流](#)

通过 CloudWatch Logs 节点查看 CloudWatch 日志组和日志流

1. 打开 AWS Explorer (如果尚未打开)。
2. 单击 CloudWatch Logs (CloudWatch 日志) 节点以展开日志组列表。

当前 AWS 区域 的日志组显示在 CloudWatch Logs 节点下。

3. 若要查看特定日志组中的日志流，请在该日志组名称上打开上下文 (右键单击) 菜单，然后选择 View Log Streams (查看日志流)。
4. 日志组的内容显示在 Select a log stream (选择日志流) 标题下。

您可以从列表中选择特定的流，也可以通过在字段中输入文本来对流进行筛选。

选择流后，该流中的事件将显示在 IDE 的 Log Streams (日志流) 窗口中。有关与每个流中的日志事件交互的信息，请参阅[使用 CloudWatch 日志事件](#)。

通过 AWS Toolkit 使用日志流中的 CloudWatch 日志事件

在打开日志流窗口后，您可以访问每个流中的日志事件。日志事件是对受监控的应用程序或资源记录的活动的记录。

主题

- [查看和复制日志流信息](#)
- [将日志流编辑器的内容保存到本地文件](#)

查看和复制日志流信息

当您打开日志流时，Log Stream (日志流) 窗口将显示该流的日志事件序列。

1. 若要查找要查看的日志流，请打开 Log Stream (日志流) 窗口。有关更多信息，请参阅[查看 CloudWatch 日志组和日志流](#)。

列出事件的每一行都有时间戳，以显示事件的录入时间。

2. 您可以使用以下选项查看和复制有关流的事件信息：

- 按时间查看事件：通过选择 Load newer events (加载更新的事件) 或 Load older events (加载更早的事件)，显示最新的日志事件或更早的日志事件。

Note

Log Stream (日志流) 编辑器最初会加载一批最近 1 万行日志事件或 1 MB 的日志数据 (以较小者为准)。如果选择 Load newer events (加载更新的事件)，编辑器会显示上一批事件加载后记录的事件。如果选择 Load older events (加载更早的事件)，编辑器会显示在当前显示的事件之前发生的一批事件。

- 复制日志事件：选择要复制的事件，然后打开上下文 (右键单击) 菜单并从菜单中选择 Copy (复制)。
- 复制日志流的名称：打开 Log Stream (日志流) 窗口的选项卡的上下文 (右键单击) 菜单，然后选择 Copy Log Stream Name (复制日志流名称)。

将日志流编辑器的内容保存到本地文件

您可以将 CloudWatch 日志流编辑器的内容下载为 log 文件，保存在本地计算机上。

Note

您可以使用此选项，仅将日志流编辑器中当前显示的这些日志事件保存到文件。例如，假设日志流的总大小为 5 MB，编辑器中仅加载了 2 MB。您保存的文件也仅包含 2 MB 的日志数据。要显示更多要保存的数据，请在编辑器中选择 Load newer events (加载更新的事件) 或 Load older event (加载更早的事件)。

1. 若要查找要复制的日志流，请打开 Log Streams (日志流) 窗口 (请参阅 [查看 CloudWatch 日志组和日志流](#))。
2. 打开 Log Stream (日志流) 窗口选项卡的上下文 (右键单击) 菜单，然后选择 Save Current Log Content to File (将当前日志内容保存为文件)

3. 使用该对话框为日志文件选择或创建下载文件夹，然后单击 Save (保存)。

通过 AWS Toolkit 使用 AWS Lambda 函数

AWS Toolkit 支持 [AWS Lambda](#) 函数。AWS 工具包取代了以前由 AWS Cloud9 中 Lambda 插件提供的功能。使用 AWS Toolkit，您可以为属于[无服务器应用程序](#)的 Lambda 函数编写代码。此外，您可以在本地或在 AWS 上调用 Lambda 函数。

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或来自各种 AWS 服务的事件。这些服务包括 Amazon Simple Storage Service (Amazon S3)、Amazon DynamoDB、Amazon Kinesis、Amazon Simple Notification Service (Amazon SNS) 和 Amazon Cognito。

Important

如果要构建使用 Serverless Application Model (SAM) 提供的资源的 Lambda 应用程序，请参阅[通过 AWS Toolkit 使用 AWS 无服务器应用程序](#)。

主题

- [调用远程 Lambda 函数](#)
- [下载、上传和删除 Lambda 函数](#)

调用远程 Lambda 函数

使用 AWS Toolkit，您可以通过各种方式与 [AWS Lambda](#) 函数进行交互。

有关 Lambda 的更多信息，请参阅 [AWS Lambda 开发人员指南](#)。

Note

假设您已使用 AWS Management Console 或以其他方式创建了 Lambda 函数。您可以从 AWS Toolkit 中调用这些函数。要使用 AWS Toolkit 创建可部署到 AWS Lambda 的新函数，必须首先[创建无服务器应用程序](#)。

先决条件

- 请确保您在其中配置的凭证包含对 AWS Lambda 服务的适当读/写访问权限。如果在 AWS Explorer 中的 Lambda 下，您看到类似于“Error loading Lambda resources”（加载 Lambda 资源时出错）的消息，请检查附加到这些凭证的权限。对权限所做的更改需要几分钟时间才会对 AWS Toolkit 中的 AWS Explorer 生效。

调用 Lambda 函数

Important

使用 AWS Toolkit 调用 API 方法可能会对资源导致无法撤消的更改。例如，如果您调用 POST 方法，则调用成功时，API 的资源将更新。

您可以使用 AWS Toolkit 在 AWS 上调用 Lambda 函数。

- 在 AWS Explorer 中，选择要调用的 Lambda 函数的名称，然后打开其上下文菜单。
- 选择 Invoke on AWS（在 Amazon 上调用）。
- 在打开的 Invoke function（调用函数）窗口中，为 Lambda 函数所需的负载选择一个选项。（该负载为您想要作为输入提供到您的 Lambda 函数的 JSON。）您可以选择 Browse（浏览）来选择要用作负载的文件，或使用下拉字段为负载选择模板。在这种情况下，Lambda 函数可能会将字符串显示为输入，如文本框所示。

选择 Invoke（调用）来调用 Lambda 并传入负载。

在 AWS Lambda 选项卡中，您将看到 Lambda 函数的输出。

下载、上传和删除 Lambda 函数

AWS 工具包提供了用于在 AWS Cloud9 IDE 中导入和上传 Lambda 函数的选项。

下载 Lambda 函数

通过下载 Lambda 函数，您还可以从 AWS Cloud 中下载描述该函数的项目文件，并在 AWS Cloud9 IDE 中使用这些文件。

下载 Lambda 函数

1. 在 AWS Explorer 中的 Lambda 节点下，打开函数的上下文（右键单击）菜单，然后选择 Download（下载）。
2. 当被要求 Select a workspace folder for your new project（为新项目选择工作区文件夹），您可以执行下列操作之一：
 - 选择建议创建与 Lambda 项目名称相同子文件夹的文件夹。
 - 选择 Select a different folder（选择其他文件夹）打开一个对话框，为项目子文件夹浏览并选择其他父文件夹。

IDE 将打开一个新的编辑器窗口。

配置下载的 Lambda 函数以运行和调试

要将下载的 Lambda 函数作为无服务器应用程序运行和调试，需要在 `launch.json` 文件中定义启动配置。在 AWS Management Console 中创建的 Lambda 函数可能不包括在启动配置中。因此，您需要手动添加它。

将 Lambda 函数添加到启动配置

1. 下载 Lambda 函数后，打开 Environment（环境）窗口以查看其文件夹和文件。
2. 接下来，检查 Lambda 函数是否包含在 `/home/ec2-user/.c9/launch.json` 文件中。如果它不存在，请执行以下操作以向函数的代码添加 CodeLens 链接：
 1. 打开定义 Lambda 函数的源代码文件（例如，`.js` 或 `.py` 文件）。然后，检查您是否有 CodeLens 链接可以用来将 lambda 函数添加到 `launch.json` 文件中。CodeLens 显示在函数上方并包括 Add Debug Config 链接。
 2. 选择 IDE 左侧的 Go（转到）（放大镜图标），然后输入“sam hint”以显示 AWS: Toggle SAM hints in source files 命令。选择此命令以运行。
 3. 关闭 Lambda 源代码文件，然后重新打开它。
 4. 如果重新打开文件后源代码中有 CodeLens，请选择 Add Debug Config 以添加启动配置。
3. 如果即使在切换 SAM hint 选项之后您仍无法添加 CodeLens，请执行以下操作以添加启动配置：
 1. 选择 IDE 左侧的 Go（转到）（放大镜图标），然后键入“config”以显示 AWS: SAM Debug Configuration Editor 命令。选择此命令以运行。

2. 此时将显示 SAM Debug Configuration Editor (SAM 调试配置编辑器)。您可以使用此编辑器定义启动配置属性。有关信息，请参阅[使用 SAM 模板运行和调试无服务器应用程序](#)中用于[configuring launch properties](#)的步骤。

Note

如果 Lambda 函数没有 SAM 应用程序的 `template.yaml`，则需要添加一个。有关更多信息，请参阅[创建 AWS SAM 模板](#)。

3. 在编辑器中输入所需的配置信息后，启动配置将添加到 `launch.json` 文件中。

在为 Lambda 函数定义启动配置后，您可以通过执行以下操作来运行它：

1. 在 IDE 顶部，选择 Auto (自动) 旁边的箭头，然后选择相关的启动配置。
2. 接下来，选择 Run (运行)。

上传 Lambda 函数

您可以使用本地代码更新现有的 Lambda 函数。以这种方式更新代码不会使用 AWS Serverless Application Model CLI 进行部署，并且不会创建 AWS CloudFormation 堆栈。通过这种方法，您可以使用 Lambda 支持的任何运行时上传 Lambda 函数。

有几个接口选项可用于通过 AWS Toolkit 上传 Lambda 函数。

从 Environment (环境) 窗口或 Command pane (命令窗格) 上传

1. 在项目文件的 Environment (环境) 窗口中，选择要上传的 Lambda 应用程序的 `template.yaml` 的上下文 (右键单击) 菜单，然后选择 Upload Lambda (上传 Lambda)。

或者，按 `Ctrl+P` 以打开转到任意内容窗格，然后输入“lambda”以访问 AWS 上传 Lambda 命令。接下来，选择它以开始上传过程。

2. 下一步，选择您希望上传到的 AWS 区域。
3. 现在，选择用于上传 Lambda 函数的选项：

上传 zip 格式归档

1. 从菜单中选择 ZIP Archive (ZIP 格式归档)。
2. 从您的 AWS Cloud9 文件系统选择一个 `.zip` 格式文件，然后选择 Open (打开)。

按原样上载目录

1. 从菜单中选择 Directory (目录)。
2. 从您的 AWS Cloud9 文件系统选择目录，然后选择 Open (打开)。
4. 指定处理事件的 Lambda 函数处理程序。当调用函数时，Lambda 运行此处理程序方法。

Note

选择 Lambda 函数时，您可以从显示的列表中进行选择。如果您不知道要选择哪个函数，可以输入此 Toolkit 中提供的 Lambda 函数的 Amazon 资源名称 (ARN)。

将显示一个对话框，询问您是否希望将此代码发布为 Lambda 函数的最新版本。选择 Yes (是) 以确认发布。

Note

您还可以打开该文件夹的父文件夹上下文 (右键单击) 菜单，然后选择 Upload Lambda (上传 Lambda) 来上传 Lambda 应用程序。将自动选择该父文件夹进行上传。

从 AWS Explorer 上载

1. 在 AWS Explorer 中，在要导入的 Lambda 函数的名称上打开上下文 (右键单击) 菜单。
2. 选择 Upload Lambda (上载 Lambda)。
3. 从上传 Lambda 函数的三个选项中进行选择。

上载预先制作的 zip 格式归档

1. 从菜单中选择 ZIP Archive (ZIP 格式归档)。
2. 从您的 AWS Cloud9 文件系统选择一个 .zip 格式文件，然后选择 Open (打开)。
3. 使用模态对话框确认上传。这将上传 .zip 格式文件，并在部署后立即更新 Lambda。

按原样上载目录

1. 从菜单中选择 Directory (目录)。

2. 从您的 AWS Cloud9 文件系统选择目录，然后选择 Open (打开)。
3. 当系统提示构建目录时，选择 No (否)。
4. 使用模态对话框确认上传。这将按原样上传目录，并在部署后立即更新 Lambda。

构建和上载目录

1. 从菜单中选择 Directory (目录)。
2. 从您的 AWS Cloud9 文件系统选择目录，然后选择 Open (打开)。
3. 当系统提示构建目录时，选择 Yes (是)。
4. 使用模态对话框确认上传。这将在目录中使用 AWS SAM CLI `sam build` 命令构建代码并在部署后立即更新 Lambda。

部署 Lambda 函数以进行远程访问

您可以通过将本地函数部署为无服务器 SAM 应用程序来远程使用它们。

将 Lambda 函数部署为 SAM 应用程序

1. 在 AWS Explorer 中，打开 Lambda 节点的上下文 (右键单击) 菜单，然后选择 Deploy SAM Application (部署 SAM 应用程序)。
2. 在命令窗格中，选择将函数定义为无服务器应用程序的 [YAML template](#) (YAML 模板)。
3. 接下来，选择 Amazon S3 存储桶来进行 Lambda 部署。也可以选择创建存储桶来进行部署。
4. 现在输入您要部署到的 AWS CloudFormation 堆栈的名称。如果指定现有堆栈，则该命令将更新堆栈。如果指定了新堆栈，则该命令将创建它。

输入堆栈的名称后，Lambda 函数会开始部署为 SAM 应用程序。成功部署后，SAM Lambda 应用程序可以远程使用。这样，您就可以从其他 AWS Cloud9 开发环境下载或调用该应用程序。

如果您想从头开始创建 Lambda 函数，我们建议按照以下步骤操作以[使用 AWS Toolkit 创建无服务器应用程序](#)。

删除 Lambda 函数

您也可以使用相同的上下文 (右键单击) 菜单删除 Lambda 函数。

Warning

请勿使用此过程删除与 [AWS CloudFormation](#) 关联的 Lambda 函数。例如，不要删除本指南前文中 [创建无服务器应用程序](#) 时创建的 Lambda 函数。这些函数必须通过 AWS CloudFormation 堆栈删除。

1. 在 AWS Explorer 中，选择要删除的 Lambda 函数的名称，然后打开其上下文（右键单击）菜单。
2. 选择 Delete（删除）。
3. 在出现的消息中，选择 Yes（是）以确认删除。

删除此函数后，它不再在 AWS Explorer 视图中列出。

使用资源

除了访问 AWS Explorer 中默认列出的 AWS 服务外，您还可以转到 Resources（资源）并从数百种资源中选择要添加到界面的资源。在 AWS 中，资源是您可以使用的实体。添加的一些资源包括 Amazon AppFlow、Amazon Kinesis Data Streams、AWS IAM 角色、Amazon VPC 和 Amazon CloudFront 分配。

要查看可用资源，请转到 Resources（资源），然后展开资源类型以列出该类型的可用资源。例如，如果选择 `AWS::Lambda::Function` 资源类型，则可以访问用于定义不同函数、其属性和特性的资源。

将资源类型添加到 Resources（资源）后，您可以通过以下方式与它及其资源进行交互：

- 查看当前 AWS 区域中可用于此资源类型的现有资源列表。
- 查看用于描述资源的 JSON 文件的只读版本。
- 复制资源的资源标识符。
- 查看说明资源类型用途和模式（JSON 和 YAML 格式）的 AWS 文档，以对资源进行建模。

用于访问资源的 IAM 权限

您需要特定 AWS Identity and Access Management 权限才能访问与 AWS 服务关联的资源。例如，IAM 实体（如用户或角色）需要 Lambda 权限才能访问 `AWS::Lambda::Function` 资源。

除了服务资源的权限外，IAM 实体还需要允许 AWS Toolkit 调用 AWS Cloud Control API 操作的权限。Cloud Control API 操作允许 IAM 用户或角色访问和更新远程资源。

您可以将 AWS 托管策略 PowerUserAccess 附加到使用 Toolkit 接口调用这些 API 操作的 IAM 实体，从而快速授予权限。此托管策略将授予执行应用程序开发任务的一系列权限，包括调用 API 操作。

有关定义远程资源上允许的 API 操作的特定权限，请参阅 [AWS Cloud Control API 用户指南](#)。

与现有资源交互

1. 在 AWS Explorer 中，选择 Resources (资源)。

资源类型列表显示在 Resources (资源) 节点下方。

2. 有一些文档描述了定义资源类型的模板的语法。要访问此文档，请打开该资源类型的上下文 (右键单击) 菜单，然后选择 View Documentation (查看文档)。

Note

系统可能会要求您关闭浏览器的弹出窗口拦截器，以便您能够访问文档页面。

3. 要查看某种资源类型已存在的资源，请展开该类型的条目。

可用资源列表显示在其资源类型下方。

4. 要与特定资源交互，请在其名称上打开上下文 (右键单击) 菜单，然后选择以下选项之一：

- Copy Identifier (复制标识符)：将特定资源的标识符复制到剪贴板。例如，可以使用 TableName 属性标识 AWS::DynamoDB::Table 资源。
- Preview (预览)：查看用于描述资源的 JSON 格式模板的只读版本。

通过 AWS Toolkit 使用 Amazon S3

以下主题介绍如何通过 AWS Toolkit 使用 AWS 账户 中的 [Amazon S3](#) 存储桶和对象。

主题

- [使用 Amazon S3 存储桶](#)
- [使用 Amazon S3 对象](#)

使用 Amazon S3 存储桶

存储于 Amazon S3 中的每个数据元都存储在存储段中。如同通过目录对文件系统中的文件进行分组一样，您也可以使用存储桶对相关对象进行分组。

主题

- [创建 Amazon S3 存储桶](#)
- [在 Amazon S3 存储桶中添加文件夹](#)
- [删除 Amazon S3 存储桶](#)
- [配置 Amazon S3 项目的显示](#)

创建 Amazon S3 存储桶

1. 在 AWS Explorer 中，打开 S3 节点的上下文（右键单击）菜单，然后选择 Create Bucket（创建存储桶）。
2. 在 Bucket Name（存储桶名称）字段中，输入一个有效的存储桶名称。按 Enter 键以确认。

新存储桶将显示在 S3 节点下方。

注意

由于 S3 存储桶可用作提供公开访问的 URL，因此您选择的存储桶名称必须具有全局唯一性。如果您选择的名称已被其他一些账户用于创建存储桶，则必须使用其他名称。如果您无法创建存储桶，可以检查输出选项卡中的 AWS Toolkit 日志。例如，如果您使用的存储桶名称已被使用，则会出现 BucketAlreadyExists 错误。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[存储桶限制](#)。

创建存储桶后，您可以将其名称和 Amazon Resource Name (ARN) 复制到剪贴板。打开存储桶条目的上下文（右键单击）菜单，然后从菜单中选择相关选项。

在 Amazon S3 存储桶中添加文件夹

您可以通过队文件夹中的对象分组来组织存储桶的内容。您也可以在其他文件夹中创建文件夹。

1. 在 AWS Explorer 中，选择 S3 节点以查看存储桶列表。
2. 打开存储桶或文件夹的上下文（右键单击）菜单，然后选择 Create Folder（创建文件夹）。

3. 输入 Folder Name (文件夹名称) , 然后按 Enter 键。

新文件夹现在显示在 AWS Explorer 窗口中已选择存储桶或文件夹下方。

删除 Amazon S3 存储桶

删除存储桶时，还会删除其中包含的文件夹和对象。在删除存储桶之前，系统将要求您确认是否要执行此操作。

Note

仅删除文件夹，而不是整个存储桶，请使用 AWS Management Console。

1. 在 AWS Explorer 中，选择 S3 节点以展开存储桶列表。
2. 打开要删除的存储桶的上下文菜单，然后选择 Delete (删除) 。
3. 输入存储桶的名称来确认您希望删除，然后按 Enter 键。

Note

如果存储桶包含对象，则会先清空存储桶，然后再删除它。如果需要删除数千个对象的所有版本，可能需要一些时间。删除过程完成后将显示通知。

配置 Amazon S3 项目的显示

如果您正在使用大量 Amazon S3 对象或文件夹，则指定一次显示的数量会很有帮助。当显示最大数量时，您可以选择 Load More (加载更多) 以显示下一个批次。

1. 在菜单栏上，依次选择 AWS Cloud9、Preferences (首选项) 。
2. 在 Preferences (首选项) 窗口中，展开 Project Settings (项目设置) ，然后转至 EXTENSIONS (扩展) 部分，选择 AWS Configuration (Amazon 配置) 。
3. 在 AWS Configuration (Amazon 配置) 窗格中，转至 AWS > S3 : Max Items Per Page (每页最大项目数) 设置。
4. 在选择加载更多内容之前，将默认值更改为您要显示的 S3 项目数。

Note

接受值的范围介于 3 到 1000 之间。此设置仅适用于同时显示的对象或文件夹的数量。您创建的所有存储桶都会一次性显示。默认情况下，您可以在每个 AWS 账户中创建多达 100 个桶。

使用 Amazon S3 对象

数据元是 Amazon S3 中存储的基础实体。对象由对象数据和元数据组成。

主题

- [将文件上传到 Amazon S3 存储桶](#)
- [下载 Amazon S3 对象](#)
- [删除 Amazon S3 对象](#)
- [为 Amazon S3 对象生成预签名 URL](#)

将文件上传到 Amazon S3 存储桶

您可以使用 Toolkit 界面或命令将文件上传到存储桶

这两种方法都允许您从用户环境中上传文件，并将其作为 S3 对象存储在 AWS 云中。您可以将文件上传到存储桶或组织该存储桶内容的文件夹。

使用界面将文件上传到 S3 存储桶

1. 在 AWS Explorer 中，选择 S3 节点以查看存储桶列表。
2. 打开存储桶或存储桶中的文件夹的上下文菜单（右键单击），然后选择 Upload File（上传文件）。

注意

如果打开 S3 对象的上下文菜单（右键单击），则可以选择 Upload to Parent（上传至父级）。这使您可以将文件添加到包含所选文件的文件夹或存储桶中。

3. 使用环境的文件管理器，选择一个文件，然后选择 Upload（上传）。

选定的文件将作为 S3 对象上传到存储桶或文件夹。每个对象的条目描述了存储对象的大小以及该对象上传时间距离现在的时间。您可以暂停在对象的列表上，以查看上次修改对象的路径、大小和时间。

使用命令将当前文件上传到 S3 存储桶

1. 若要选择要上传的文件，请选择该文件的选项卡。
2. 按 Ctrl+P 以显示 Commands (命令) 窗格。
3. 对于 Go To Anything (转到任意内容) ，开始输入短语 `upload file` 显示 `AWS: Upload File` 命令。当命令出现时，选择该命令。
4. 对于步骤 1：选择要上传的文件，您可以选择已选定的文件或浏览其他文件。
5. 对于步骤 2：选择要上传到的 S3 存储桶，从列表中选择存储桶。

选定的文件将作为 S3 对象上传到存储桶或文件夹。每个对象的条目描述了存储对象的大小以及该对象上传时间距离现在的时间。您可以暂停在对象的列表上，以查看上次修改对象的路径、大小和时间。

下载 Amazon S3 对象

您可以从 AWS 云下载 Amazon S3 存储桶中的对象到您的 AWS Cloud9 环境中的文件夹。

1. 在 AWS Explorer 中，选择 S3 节点以查看存储桶列表。
2. 在存储桶中或存储桶中的文件夹中，打开对象的上下文菜单 (右键单击) ，然后选择 `Download As` (下载为) 。
3. 使用环境的文件管理器，选择目标文件夹，输入文件名，然后选择 `Download` (下载) 。

下载文件后，您可以在 AWS Cloud9 中打开。

删除 Amazon S3 对象

如果对象位于不受版本控制的存储桶中，您可以永久删除该对象。但是，对于启用版本控制的存储桶，删除请求不会永久删除该对象。但是，Amazon S3 将在存储桶中插入一个删除标记。有关更多信息，请参阅 [Amazon Simple Storage Service 用户指南中的删除对象版本](#)。

1. 在 AWS Explorer 中，选择 S3 节点以查看存储桶列表。

2. 在存储桶或存储桶中的文件夹中，打开对象的上下文菜单（右键单击），然后选择 Delete（删除）。
3. 要确认删除，请选择 Delete（删除）。

为 Amazon S3 对象生成预签名 URL

使用预签名 URL，对象所有者可以通过授予下载对象的限时权限来与其他人共享私有 Amazon S3 对象。有关更多信息，请参阅 Amazon S3 用户指南中的[使用预签名 URL 共享对象](#)。

1. 在 AWS Explorer 中，选择 S3 节点以查看存储桶列表。
2. 在存储桶或存储桶中的文件夹中，右键单击某个对象，然后选择 Generate Presigned URL（生成预签名 URL）。
3. 在 AWS Toolkit 命令窗格中，输入可使用该 URL 访问对象的分钟数。按 Enter 键以确认。

IDE 底部的状态确认对象的预签名 URL 已复制到剪贴板。

通过 AWS Toolkit 使用 AWS 无服务器应用程序

AWS Toolkit 提供对[无服务器应用程序](#)的支持。使用 AWS Toolkit，您可以创建包含 [AWS Lambda](#) 函数的无服务器应用程序，然后将应用程序部署到 AWS CloudFormation 堆栈。

主题

- [创建无服务器应用程序](#)
- [运行和调试无服务器应用程序](#)
- [同步无服务器应用程序](#)
- [启用 AWS Toolkit 代码镜头](#)
- [从 AWS 云中删除无服务器应用程序](#)
- [调试无服务器应用程序的配置选项](#)

创建无服务器应用程序

此示例演示如何使用 AWS Toolkit 创建无服务器应用程序。有关如何运行和调试无服务器应用程序的信息，请参阅[运行和调试无服务器应用程序](#)。

创建无服务器应用程序的必要先决条件包括 AWS SAM CLI 和 AWS CLI。这些都包含在 AWS Cloud9 中。如果 AWS SAM CLI 未安装或已过期，则可能需要运行安装或升级。有关如何安装 AWS SAM CLI

的说明，请参阅[安装 AWS SAM CLI](#)；有关如何升级 AWS SAM CLI 的说明，请参阅[升级 AWS SAM CLI](#)。

使用 AWS Toolkit 创建无服务器应用程序

此示例演示如何使用 [AWS Serverless Application Model \(AWS SAM\)](#) 通过 AWS Toolkit 创建无服务器应用程序。

1. 在 AWS Explorer 中，打开 Lambda 节点的上下文（右键单击）菜单，然后选择 Create Lambda SAM Application（创建 Lambda SAM 应用程序）。

Note

或者，您也可以单击 AWS: Explorer 标题中的菜单图标，然后选择 Create Lambda SAM Application（创建 Lambda SAM 应用程序）。

2. 选择 SAM 应用程序的运行时。对于此示例，请选择 nodejs12.x。

Note

如果您选择一个带有“(Image)”的运行时，则您的应用程序是 Image 型软件包。如果您选择一个不带有“(Image)”的运行时，则您的应用程序是 Zip 型。有关 Image 和 Zip 软件包类型差异的更多信息，请参阅 AWS Lambda 开发人员指南中的 [Lambda 部署软件包](#)。

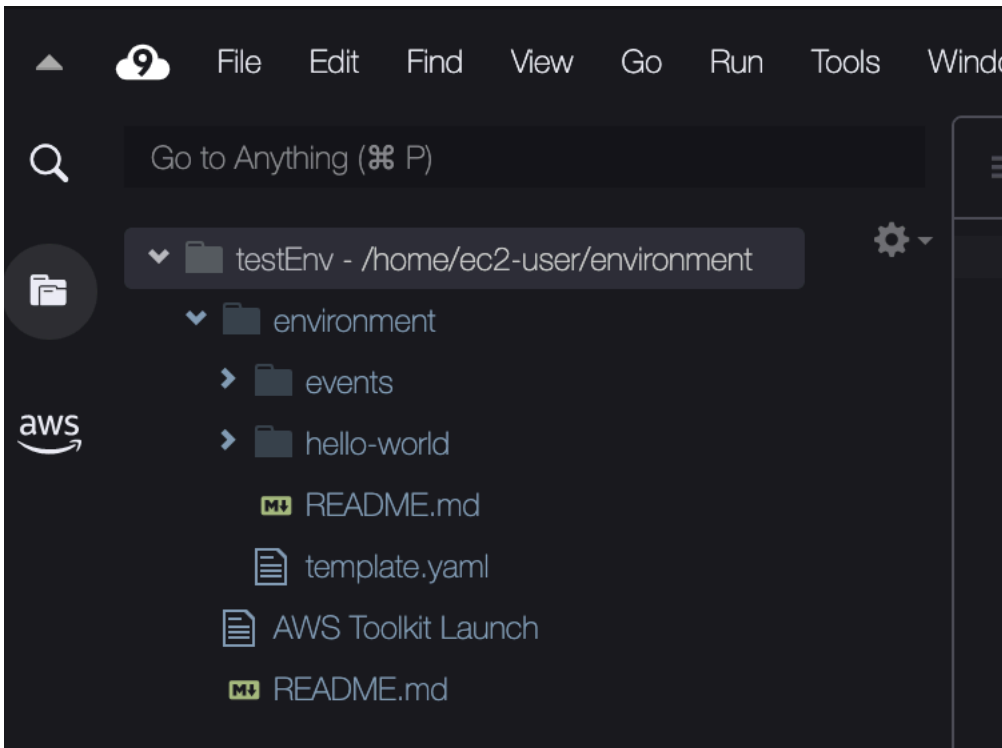
3. 为您的无服务器应用程序选择以下模板之一：

- AWS SAM Hello World：带有 Lambda 函数的基本模板，返回经典的“Hello World（你好，世界）”消息。
- AWS Step Functions 示例应用程序：运行股票交易工作流的示例应用程序。Step Functions 编排所涉及 Lambda 函数的交互。

4. 为新项目选择一个位置。如果有工作区文件夹可用，您可以选择现有的工作区文件夹。否则，请浏览其他文件夹。如果您选择 Select a different folder（选择其他文件夹），将显示一个对话框，允许您选择文件夹位置。

5. 输入新应用程序的名称。对于本示例，请使用 my-sam-app-nodejs。按 Enter 键后，AWS Toolkit 需要几分钟才能创建项目。

创建项目后，您可以在环境窗口中查看应用程序的文件。在 Explorer 窗口的列表中找到该文件夹。



运行和调试无服务器应用程序

您可以使用 AWS Toolkit 配置如何调试无服务器应用程序并在开发环境中本地运行它们。您可以调试由 AWS Serverless Application Model (AWS SAM) 模板定义的无服务器应用程序。此模板使用简单的 YAML 语法来描述构成无服务器应用程序的函数、API、数据库和事件源映射等资源。

为了更详细地了解 AWS SAM 模板，请参阅 [AWS Serverless Application Model 开发人员指南](#) 中的 [AWS SAM 模板剖析](#)。

或者，您可以快速调试尚未提交到 SAM 模板的无服务器应用程序。

您开始通过使用内联操作来配置调试行为，以识别符合条件 AWS Lambda 函数。要使用 SAM 模板定义的基础设施，请在相关 YAML 格式的文件中使用内联操作。要在不使用模板的情况下直接测试函数，请使用应用程序文件中 Lambda 处理程序的上下文感知链接。

Note

在此示例中，我们调试使用 JavaScript 的应用程序。但是，您可以使用 AWS Toolkit 中可用的，具有以下语言和运行时的调试功能：

- JavaScript – Node.js 10.x、12.x、14.x

- Python – 3.7、3.8、3.9、3.10 (Python 2.7 和 3.6 无服务器应用程序可以运行，但不能通过 AWS 工具包调试。)

您的语言选择还会影响上下文感知链接指示符合条件的 Lambda 处理程序的方式。有关更多信息，请参阅 [直接从代码运行和调试无服务器函数](#)。

使用 SAM 模板运行和调试无服务器应用程序

对于使用 SAM 模板运行和调试的应用程序，YAML 格式的文件描述了应用程序的行为及其使用的资源。如果您使用 AWS 工具包创建无服务器应用程序，您的项目会自动生成一个名为 `template.yaml` 的文件。

在此步骤中，使用在 [创建无服务器应用程序](#) 中创建的示例应用程序。

使用 SAM 模板运行和调试无服务器应用程序

1. 要查看构成无服务器应用程序的应用程序文件，请转到 Environment (环境) 窗口中。
2. 从应用程序文件夹 (如 `my-sample-app` (我的样本应用程序))，打开 `template.yaml` 文件。
3. 对于 `template.yaml`，选择 Edit Launch Configuration (编辑启动配置)。

新编辑器将显示 `launch.json` 文件，该文件提供具有默认属性的调试配置。

4. 编辑或确认以下配置属性的值：
 - "name" – 输入一个易于阅读的名称，以显示在 Run (运行) 视图中的 Configuration (配置) 下拉字段中。
 - "target" – 确保该值为 "template"。这样，SAM 模板便成为调试会话的入口点。
 - "templatePath" – 输入 `template.yaml` 文件的相对路径或绝对路径。
 - "logicalId" – 确保名称与 SAM 模板的 Resource (资源) 部分中指定的名称一致。在这种情况下，它是类型 `AWS::Serverless::Function` 的 `HelloWorldFunction`。

有关 `launch.json` 文件中这些以及其他条目的更多信息，请参阅 [调试无服务器应用程序的配置选项](#)。

5. 如果您对调试配置满意，请保存 `launch.json`。然后选择 RUN (运行) 旁边的绿色“play” (播放) 按钮以启动调试。

Note

如果 SAM 应用程序无法运行，请检查 Output (输出) 窗口以查看错误是否因 Docker 镜像未构建引起。您可能需要释放环境中的磁盘空间。

有关更多信息，请参阅[由于 AWS Cloud9 环境没有足够的磁盘空间，因此在 AWS Toolkit 中本地运行 SAM 应用程序时出错](#)。

当调试会话启动时，DEBUG CONSOLE (调试控制台) 面板显示调试输出，并显示 Lambda 函数返回的任何值。调试 SAM 应用程序时，选择 AWS Toolkit 作为 Output (输出) 面板中的 Output (输出) 通道。

Note

对于 Windows 用户，如果在此过程中遇到 Docker 安装错误，则可能需要在 Docker Settings (Docker 设置) 中刷新共享驱动器的凭证。Docker 安装错误类似于以下内容。

```
Fetching lambci/lambda:nodejs10.x Docker container image.....
2019-07-12 13:36:58 Mounting C:\Users\\AppData\Local\Temp\ ...
as /var/task:ro,delegated inside runtime container
Traceback (most recent call last):
...requests.exceptions.HTTPError: 500 Server Error: Internal Server
Error ...
```

直接从代码运行和调试无服务器函数

在测试 AWS SAM 应用程序时，您可以选择仅运行和调试 Lambda 函数。排除 SAM 模板定义的其他资源。此方法涉及使用内联操作来识别源代码中可以直接调用的 Lambda 函数处理程序。

上下文感知链接检测到的 Lambda 处理程序取决于您为应用程序使用的语言和运行时。

语言/运行时	上下文感知链接识别 Lambda 函数的条件
JavaScript (Node.js 10.x、12.x、14.x)	该函数具有以下功能： <ul style="list-style-type: none"> • 这是一个导出的函数，最多有三个参数。

语言/运行时	上下文感知链接识别 Lambda 函数的条件 <ul style="list-style-type: none"> 它在工作区文件夹中的父文件夹中具有 <code>package.json</code> 文件。
Python (3.7、3.8、3.9 和 3.10)	该函数具有以下功能： <ul style="list-style-type: none"> 这是一个顶级函数。 它在工作区文件夹中的父文件夹中具有 <code>requirements.txt</code> 文件。

直接从应用程序代码运行和调试无服务器应用程序

- 要查看无服务器应用程序文件，请选择编辑器旁边的文件夹图标导航到应用程序文件夹。
- 从应用程序文件夹（如 `my-sample-app`），展开函数文件夹（在本例中为 `hello-world`），然后打开 `app.js` 文件。
- 在识别符合条件的 Lambda 处理程序函数的内联操作中，选择 `Add Debug Configuration`。如果未出现添加调试配置选项，则必须启用代码透镜。要启用代码镜头，请参阅[the section called “启用 AWS Toolkit 代码镜头”](#)。
- 选择运行 SAM 应用程序的运行时。
- 在 `launch.json` 文件的编辑器中，编辑或确认以下配置属性的值：
 - `"name"` – 输入易于阅读的名称。
 - `"target"` – 确保值为 `"code"`，以便直接调用 Lambda 函数处理程序。
 - `"lambdaHandler"` – 输入代码中方法的名称，Lambda 可使用该方法来调用您的函数。例如，对于 JavaScript 中的应用程序，默认为 `app.lambdaHandler`。
 - `"projectRoot"` – 输入包含 Lambda 函数的应用程序文件的路径。
 - `"runtime"` – 输入或确认 Lambda 执行环境的有效运行时（例如 `"nodejs.12x"`）。
 - `"payload"` – 选择以下选项之一以定义要作为输入提供给 Lambda 函数的事件负载：
 - `"json"`：JSON 格式的键值对，用于定义事件负载。
 - `"path"`：用作事件负载的文件路径。
- 如果您对调试配置满意，请选择 `RUN`（运行）旁边的绿色播放箭头启动调试。

当调试会话启动时，DEBUG CONSOLE（调试控制台）面板显示调试输出，并显示 Lambda 函数返回的任何值。调试 SAM 应用程序时，选择 AWS Toolkit 作为 Output（输出）面板中的 Output（输出）通道。

Note

如果您看到错误消息中提到的 Docker，请参阅此[备注](#)。

运行和调试本地 Amazon API Gateway 资源

您可以运行或调试在 `template.yaml` 中指定的 AWS SAM API Gateway 本地资源。为此，请运行 AWS Cloud9 启动配置 `type=aws-sam` 和 `invokeTarget.target=api`。

Note

API Gateway 支持两种类型的 API。它们是 REST 和 HTTP API。但是，带有 AWS 工具包的 API Gateway 功能仅支持 REST API。有时候 HTTP API 被称为“API Gateway V2 API”。

运行和调试本地 API Gateway 资源

1. 选择以下方法之一以创建 AWS SAM API Gateway 资源的启动配置：

- 选项 1：访问 AWS SAM 项目中的处理程序源代码（具体而言，是 `.js`、`.cs` 或 `.py` 文件），将鼠标悬停在 Lambda 处理程序上，然后选择 Add Debug Configuration（添加调试配置）。如果未出现添加调试配置选项，请启用代码透镜。要启用代码透镜，请参阅[the section called “启用 AWS Toolkit 代码镜头”](#)。然后，在菜单中，选择标记为 API 事件的项目。
- 选项 2 编辑 `launch.json` 并使用以下语法创建新的启动配置。

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
  "invokeTarget": {
    "target": "api",
    "templatePath": "n12/template.yaml",
    "logicalId": "HelloWorldFunction"
  },
}
```

```
"api": {
  "path": "/hello",
  "httpMethod": "post",
  "payload": {
    "json": {}
  }
},
"sam": {},
"aws": {}
}
```

2. 在 Run (运行) 按钮旁的下拉菜单中, 选择启动配置 (在前例中名为 myConfig)。
3. (可选) 将断点添加到您的 Lambda 项目代码中。
4. 选择绿色 play (播放) 按钮旁边的 Run (运行) 按钮。
5. 在输出窗格中, 查看结果。

配置

在使用 `invokeTarget.target` 属性值 `api` 时, Toolkit 会更改启动配置验证和行为, 以支持 `api` 字段。

```
{
  "type": "aws-sam",
  "request": "direct-invoke",
  "name": "myConfig",
  "invokeTarget": {
    "target": "api",
    "templatePath": "n12/template.yaml",
    "logicalId": "HelloWorldFunction"
  },
  "api": {
    "path": "/hello",
    "httpMethod": "post",
    "payload": {
      "json": {}
    },
    "queryString": "abc=def&qrs=tuv",
    "headers": {
      "cookie": "name=value; name2=value2; name3=value3"
    }
  },
}
```

```
"sam": {},  
"aws": {}  
}
```

按以下示例的方式替换值：

`invokeTarget.logicalId`

API 资源。

`path`

启动配置请求的 API 路径（例如 `"path": "/hello"`）。

必须是从由 `invokeTarget.templatePath` 指定的 `template.yaml` 解析出的有效 API 路径。

`httpMethod`

使用以下任一动词：“delete”、“get”、“head”、“options”、“patch”、“post”和“put”。

`payload`

要在请求中发送的 JSON 负载（HTTP 正文），其结构和规则与 `lambda.payload` 字段相同。

`payload.path` 指向包含 JSON 负载的文件。

`payload.json` 指定内联 JSON 负载。

标头

可选的名称/值对映射。使用它来指定要包含在请求中的 HTTP 标头。

```
"headers": {  
  "accept-encoding": "deflate, gzip;q=1.0, *;q=0.5",  
  "accept-language": "fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5",  
  "cookie": "name=value; name2=value2; name3=value3",  
  "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)  
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36",  
}
```

`querystring`

（可选）使用此字符串设置请求的 `querystring`（例如 `"querystring": "abc=def&ghi=jkl"`）。

AWS

AWS 连接信息提供的方式。有关更多信息，请参阅 [调试无服务器应用程序的配置选项](#) 中的 AWS 连接 (**aws**) 属性表。

sam

AWS SAM CLI 构建应用程序的方式。有关更多信息，请参阅 [调试无服务器应用程序的配置选项](#) 中的 AWS SAM CLI (**sam**) 属性。

同步无服务器应用程序

此示例演示如何使用 AWS Toolkit for Visual Studio Code 将在上一主题 ([创建无服务器应用程序](#)) 中创建的无服务器应用程序同步到 AWS。

先决条件

- 请务必选择全局唯一的 Amazon S3 存储桶名称。
- 确保您配置的凭证包含对以下服务的适当读/写访问权限：Amazon S3、AWS CloudFormation、AWS Lambda 和 Amazon API Gateway。
- 对于具有部署类型 Image 的应用程序，请确保您拥有全局唯一的 Amazon S3 存储桶名称和用于部署的 Amazon ECR 存储库 URI。

同步无服务器应用程序

1. 在 AWS Explorer 窗口中，打开 Lambda 节点的上下文（右键单击）菜单，然后选择同步 SAM 应用程序。
2. 选择要部署到的 AWS 区域。
3. 选择要用于部署的 `template.yaml` 文件。
4. 输入此部署可使用的 Amazon S3 存储桶的名称。存储桶必须位于您要部署到的区域中。

Warning

Amazon S3 存储桶名称在 Amazon S3 中的所有现有存储桶名称之间必须是全局唯一的。向以下示例给出的名称添加唯一标识符，或者选择其他名称。

5. 如果您的无服务器应用程序包含具有类型 Image 的函数，输入此部署可使用的 Amazon ECR 存储库的名称。存储库必须位于您要部署到的区域中。

6. 输入已部署堆栈的名称：新堆栈名称或现有堆栈名称。
7. 在 Console (控制台) 的 AWS Toolkit 选项卡验证部署是否成功。

如果发生错误，右下角将弹出一条消息。

如果发生这种情况，请查看 AWS Toolkit 选项卡中的文本以了解详细信息。以下是错误详细信息的示例。

```
Error with child process: Unable to upload artifact HelloWorldFunction referenced
  by CodeUri parameter of HelloWorldFunction resource.
S3 Bucket does not exist. Execute the command to create a new bucket
aws s3 mb s3://pbart-my-sam-app-bucket
An error occurred while deploying a SAM Application. Check the logs for more
information by running the "View AWS Toolkit Logs" command from the Command
Palette.
```

在此示例中，出现错误的原因是 Amazon S3 存储桶不存在。

完成部署时，您将看到 AWS Explorer 窗口中列出了您的应用程序。要了解如何调用作为应用程序的一部分创建的 Lambda 函数，请参阅 [调用远程 Lambda 函数](#)。

启用 AWS Toolkit 代码镜头

1. 在菜单栏上，依次选择 AWS Cloud9 和 Preferences (首选项) 。
2. 在 Preferences (首选项) 选项卡上，在侧边栏中选择 AWS Toolkit。
3. 要启用代码镜头，请选择 Enable Code Lenses (启用代码镜头) 。

从 AWS 云中删除无服务器应用程序

删除无服务器应用程序涉及删除之前部署到 AWS 云的 AWS CloudFormation 堆栈。请注意，此过程不会从本地主机中删除您的应用程序目录。

1. 打开 AWS Explorer。
2. 在 AWS Explorer 窗口中，展开包含希望删除的已部署应用程序的区域，然后展开 AWS CloudFormation。
3. 对于所要删除的无服务器应用程序，打开与其对应的 AWS CloudFormation 堆栈名称的上下文 (右键单击) 菜单。然后，选择 Delete CloudFormation Stack (删除 CloudFormation 堆栈) 。

4. 要确认删除已选择的堆栈，请选择 Delete（删除）。

如果堆栈删除成功，AWS Toolkit 将从 AWSExplorer 中的 AWS CloudFormation 列表中删除堆栈名称。

调试无服务器应用程序的配置选项

通过内联操作，您可以轻松地查找和定义直接调用 Lambda 函数或使用 SAM 模板的属性。您还可以定义属性 "lambda"（函数运行方式）、"sam"（AWS SAM CLI 构建应用程序的方式），以及 "aws"（提供 AWS 连接信息的方式）。

AWS SAM：Lambda 处理程序直接调用/基于模板的 Lambda 调用

属性	描述
type	指定哪个扩展管理启动配置。始终设置为 <code>aws-sam</code> 来使用 AWS SAM CLI 在本地构建和调试。
name	指定一个易于阅读的名称，以显示在 Debug launch configuration（调试启动配置）列表中。
request	指定要由指定扩展程序执行的配置类型 (<code>aws-sam</code>)。始终设置为 <code>direct-invoke</code> 以启动 Lambda 函数。
invokeTarget	<p>指定资源调用的入口点。</p> <p>为了直接调用 Lambda 函数，请为以下 <code>invokeTarget</code> 字段设置值：</p> <ul style="list-style-type: none"> <code>target</code> – 设置为 <code>code</code>。 <code>lambdaHandler</code> – 要调用的 Lambda 函数处理程序的名称。 <code>projectRoot</code> – 包含 Lambda 处理程序的应用程序文件的路径。 <p>为了使用 SAM 模板调用 Lambda 资源，请为以下 <code>invokeTarget</code> 字段设置值：</p> <ul style="list-style-type: none"> <code>target</code> – 设置为 <code>template</code>。 <code>templatePath</code> – SAM 模板文件的路径。

属性	描述
	<ul style="list-style-type: none"> <code>logicalId</code> – 要调用的 <code>AWS::Lambda::Function</code> 或 <code>AWS::Serverless::Function</code> 的资源名称。您可以在 YAML 格式的 SAM 模板中找到资源名称。

Lambda ("**lambda**") 属性

属性	描述
<code>environmentVariables</code>	将操作参数传递到您的函数。例如，如果您要写入 Amazon S3 存储桶，请将存储桶名称配置为环境变量。不要对您所写入的存储桶名称进行硬编码。
<code>payload</code>	为作为输入提供给 Lambda 函数的事件负载提供两个选项。 <ul style="list-style-type: none"> <code>"json"</code>：JSON 格式的键值对，用于定义事件负载。 <code>"path"</code>：用作事件负载的文件路径。
<code>memoryMB</code>	指定为运行一个已调用 Lambda 函数所提供的内存（以兆为单位）。
<code>runtime</code>	指定 Lambda 函数使用的运行时。有关更多信息，请参阅 AWS Lambda 运行时 。
<code>timeoutSec</code>	设置调试会话超时之前的允许时间（以秒为单位）。

AWS Toolkit 扩展使用 AWS SAM CLI 在本地构建和调试无服务器应用程序。您可以使用 `launch.json` 文件中的 `"sam"` 配置属性来配置 AWS SAM CLI 命令的行为。

AWS SAM CLI ("**sam**") 属性

属性	描述	默认值
<code>buildArguments</code>	配置 <code>sam build</code> 命令构建 Lambda 源代码的方式。若要查看构建选项，请参阅 AWS	空字符串

属性	描述	默认值
	Serverless Application Model 开发人员指南中的 sam 构建 。	
<code>containerBuild</code>	指示是否在类似 AWS Lambda 的 Docker 容器内构建函数。	<code>false</code>
<code>dockerNetwork</code>	Lambda Docker 容器应连接到的现有 Docker 网络的名称或 ID，以及默认桥接网络。如果未指定此项，Lambda 容器将仅连接到默认的桥接 Docker 网络。	空字符串
<code>localArguments</code>	其他本地调用参数。	空字符串
<code>skipNewImageCheck</code>	指定命令是否应跳过下拉最新 Docker 镜像获取 Lambda 运行时的操作。	<code>false</code>
<code>template</code>	通过使用参数向 SAM 模板输入客户值来对其进行自定义。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的 参数 。	<code>"parameters": {}</code>

AWS 连接 ("aws") 属性

属性	描述	默认值
<code>credentials</code>	从凭证文件中选择一个特定的配置文件（如 <code>profile:default</code> ）以获取 AWS 凭证。	由现有共享 AWS Config 文件或共享 AWS 凭证提供的 AWS 凭证。
<code>Region</code>	设置服务的 AWS 区域（如 <code>us-east-1</code> ）。	与活动凭证配置文件关联的默认 AWS 区域。

通过 AWS 工具包使用 AWS Step Functions

AWS 工具包提供对 [AWS Step Functions](#) 的支持。Step Functions 允许您创建状态机，为支持业务关键型应用程序的 AWS Lambda 函数和其他 AWS 服务定义 workflow。

您可以通过 Step Functions 使用 AWS 工具包来执行以下操作：

- 创建并发布状态机，它是由各个步骤组成的 workflow。
- 下载定义状态机 workflow 的文件。
- 使用已输入或选择的输入运行状态机 workflow。

主题

- [先决条件](#)
- [创建和发布状态机](#)
- [在 AWS 工具包中运行状态机](#)
- [下载状态机定义文件并可视化其 workflow](#)

先决条件

Step Functions 可以运行代码并访问 AWS 资源（例如调用 Lambda 函数）。为了保持安全性，您必须使用 IAM 角色为 Step Functions 授予对这些资源的访问权限。

使用 AWS 工具包，您可以利用自动生成的 IAM 角色，这些角色对您创建状态机的 AWS 区域有效。要为状态机创建您自己的 IAM 角色，请参阅 AWS Step Functions 开发人员指南中的 [AWS Step Functions 如何与 IAM 协同工作](#)。

创建和发布状态机

当您使用 AWS 工具包创建一个状态机时，您可以选择一个为业务案例定义 workflow 的起始模板。然后，您可以编辑或替换该模板以满足您的特定需求。有关在表示状态机结构的文件中定义状态机的详细信息，请参阅 AWS Step Functions 开发人员指南中的 [Amazon 状态语言](#)。

1. 在 AWS Explorer 窗格中，打开上下文（右击）菜单以访问 Step Functions，然后选择 Create a new Step Function state machine（新建 Step Function 状态机）。
2. 在命令面板中，为状态机的 workflow 选择入门模板。
3. 接下来，为定义状态机的 Amazon 状态机 (ASL) 文件选择格式。

打开一个编辑器以显示定义状态机工作流的 ASL 文件。

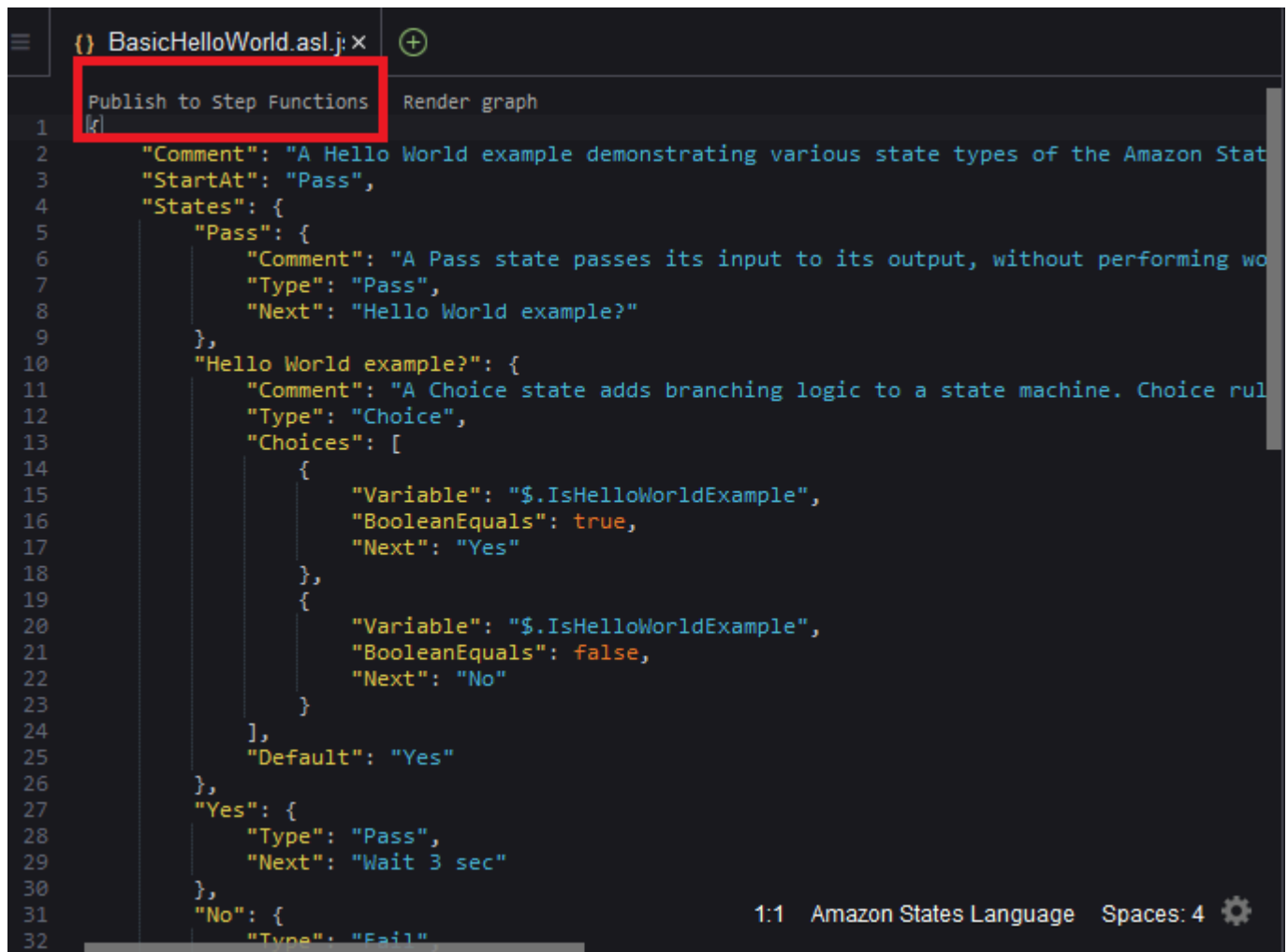
Note

有关编辑 ASL 文件以自定义工作流程的信息，请参阅 [State Machine Structure](#) (状态机结构)。

4. 在 ASL 文件中，选择发布到 Step Functions，以将状态机添加到 AWS 云。

Note

您也可以在 ASL 文件中选择 Render graph (渲染图) 来显示状态机工作流的可视化表示。



```
{} BasicHelloWorld.asl.j: x +
Publish to Step Functions Render graph
1 [
2   "Comment": "A Hello World example demonstrating various state types of the Amazon Stat
3   "StartAt": "Pass",
4   "States": {
5     "Pass": {
6       "Comment": "A Pass state passes its input to its output, without performing wo
7       "Type": "Pass",
8       "Next": "Hello World example?"
9     },
10    "Hello World example?": {
11      "Comment": "A Choice state adds branching logic to a state machine. Choice rul
12      "Type": "Choice",
13      "Choices": [
14        {
15          "Variable": "$.IsHelloWorldExample",
16          "BooleanEquals": true,
17          "Next": "Yes"
18        },
19        {
20          "Variable": "$.IsHelloWorldExample",
21          "BooleanEquals": false,
22          "Next": "No"
23        }
24      ],
25      "Default": "Yes"
26    },
27    "Yes": {
28      "Type": "Pass",
29      "Next": "Wait 3 sec"
30    },
31    "No": {
32      "Type": "Fail",
```

5. 在命令面板中，选择 AWS 区域以托管您的 Step Function。
6. 接下来，您可以选择创建新的步骤函数，或选择更新现有的函数。

Quick Create

此选项允许您使用 [step-functions/latest/dg/concepts-standard-vs-express.html](https://docs.aws.amazon.com/step-functions/latest/dg/concepts-standard-vs-express.html) 通过 ASL 文件新建 Step Function。系统要求您指定以下内容：

- IAM 角色，允许您的 Step Function 运行代码和访问 AWS 资源。（您可以选择一个自动生成的 IAM 角色，该角色对您创建状态机的 AWS 区域有效。）
- 新函数的名称。

您可以检查您的状态机是否已成功创建，并在 AWS 工具包输出选项卡中获取其 ARN。

Quick Update

如果状态机已经存在于 AWS 区域，您可以选择一个以使用当前 ASL 文件进行更新。

您可以检查您的状态机是否已成功更新，并在 AWS 工具包输出选项卡中获取其 ARN。

创建状态机之后，它会显示在 AWS Explorer 窗格中的 Step Functions 下方。如果没有立即出现，请选择 Toolkit (工具包) 菜单，Refresh Explorer (刷新 Explorer)。

在 AWS 工具包中运行状态机

您可以使用 AWS 工具包运行远程状态机。运行状态机接收 JSON 文本作为输入，并将该输入传递给工作流中的第一个状态。各个状态将接收输入形式的 JSON 并通常将 JSON 以输出形式传递到下一个状态。有关更多信息，请参阅 [Step Functions 中输入和输出的处理](#)。

1. 在 AWS Explorer 窗格中，请选择 Step Functions。然后打开特定状态机的上下文（右键单击）菜单，并选择 Start Execution (开始执行)。
2. 在 Start Execution (开始执行) 窗格中，通过直接在下面的字段中输入文本或从本地设备上传文件，为状态机的工作流添加 JSON 格式的输入。
3. 选择 Execute (执行)。

AWS 工具包输出选项卡显示工作流已启动的确认和流程标识的 ARN。您可以使用该进程 ID 来签入 AWS Step Functions 控制台，以检查工作流是否成功运行。您还可以查看工作流程开始和结束的时间戳。

下载状态机定义文件并可视化其 workflow

下载状态机意味着您下载一个包含表示该状态机结构的 JSON 文本的文件。然后，您可以编辑此文件以创建新状态机或更新现有状态机。有关更多信息，请参阅 [AWS Step Functions 开发人员指南中的 Amazon 状态语言](#)。

1. 在 AWS Explorer 窗格中，请选择 Step Functions。然后打开特定状态机的上下文（右键单击）菜单，并选择 Download Definition（下载定义）。

Note

上下文菜单还提供了若干选项，可复制名称和复制 ARN。

2. 在 Save（保存）对话框中，选择您在环境中存储已下载状态机文件的文件夹，然后选择 Save（保存）。

定义状态机 workflow 的 JSON 格式文件将显示在编辑器中。

3. 要显示 workflow 的可视化表示，请选择 Render graph（渲染图）。

窗口显示一个流程图，其中显示状态机 workflow 中的状态顺序。

使用 Systems Manager 自动化文档

使用 AWS Systems Manager，您可以查看和控制 AWS 上的基础设施。Systems Manager 提供一个统一的用户界面，供您查看多种 AWS 服务的操作数据，并在 AWS 资源上自动执行操作任务。

[Systems Manager 文档](#) 定义 Systems Manager 对您的托管式实例执行的操作。自动化文档是一种 Systems Manager 文档，用于执行常见的维护和部署任务。这包括创建或更新 Amazon 机器映像（AMI）。本主题概述了如何通过 AWS 工具包创建、编辑、发布和删除自动化文档。

主题

- [假设和先决条件](#)
- [Systems Manager 自动化文档的 IAM 权限](#)
- [创建新的 Systems Manager 自动化文档](#)
- [发布 Systems Manager 自动化文档](#)
- [编辑现有 Systems Manager 自动化文档](#)

- [使用版本](#)
- [删除 Systems Manager 自动化文档](#)
- [运行 Systems Manager 自动化文档](#)
- [排除 AWS 工具包中 Systems Manager 自动化文档的问题](#)

假设和先决条件

在开始之前，请确保您满足以下条件：

- 您熟悉 Systems Manager。有关更多信息，请参阅 [AWS Systems Manager 用户指南](#)。
- 你熟悉 Systems Manager 自动化使用案例。有关更多信息，请参阅 AWS Systems Manager 用户指南中的 [AWS Systems Manager 自动化](#)。

Systems Manager 自动化文档的 IAM 权限

要创建、编辑、发布和删除 Systems Manager 自动化文档，您必须具有包含所需 AWS Identity and Access Management (IAM) 权限的凭证配置文件。以下策略文档定义了可在主体策略中使用的必要 IAM 权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListDocuments",
        "ssm:ListDocumentVersions",
        "ssm:DescribeDocument",
        "ssm:GetDocument",
        "ssm:CreateDocument",
        "ssm:UpdateDocument",
        "ssm:UpdateDocumentDefaultVersion",
        "ssm>DeleteDocument"
      ],
      "Resource": "*"
    }
  ]
}
```

有关如何更新 IAM policy 的信息，请参阅《IAM 用户指南》中的[创建 IAM policy](#)。

创建新的 Systems Manager 自动化文档

您可以使用 AWS 工具包在 JSON 或 YAML 中创建自动化文档。创建自动化文档时，它将显示在无标题文件中。您可以命名文件并将其保存。但是，在发布文件之前，文件不会上传到 AWS。

要新建自动化文档

1. 请在左侧导航窗格中选择搜索图标或按 Ctrl+P 以打开搜索窗格。
2. 在搜索窗格中，开始输入术语“systems manager”并于显示时选择 AWS: Create a new Systems Manager Document Locally 命令。
3. 为“Hello World”示例选择其中一个初学者模板。
4. 请选择 JSON 或 YAML 作为文档的模板。

编辑器将显示新的自动化文档。

Note

当第一次创建本地自动化文档时，它不会自动显示在 AWS 中。您必须将其发布至 AWS 后方可运行。

发布 Systems Manager 自动化文档

在 AWS 工具包中创建或编辑自动化文档后，您可以将其发布到 AWS。

要发布您的自动化文档

1. 打开您想要使用 [编辑现有 Systems Manager 自动化文档](#) 中概述的流程进行发布的自动化文档。
2. 在左侧导航窗格中选择搜索图标或按 Ctrl+P 以打开搜索窗格。
3. 在搜索窗格中，开始输入术语 "systems manager" 并于显示时选择 AWS: Publish a new Systems Manager Document 命令。
4. 对于 Step 1 of 3 [步骤 1 (共 3 步)]，选择想要发布文档的 AWS 区域。
5. 对于步骤 2 (共 3 步)，选择 Quick Create (快速创建) 以创建一个自动化文档。或者选择 Quick Update (快速更新) 以在该区域中更新现有的自动化文档。

Note

您只能更新您拥有的自动化文档。如果选择 Quick Update (快速更新) 而且您不拥有该区域中的任何文档，会有一条消息通知您在更新文档之前发布文档。

- 对于步骤 3 (共 3 步) ，根据您在上一步中的选择，输入新自动化文档的名称或选择要更新的现有文档。

Note

当您向 AWS 中现有的自动化文档发布更新时，将向文档中添加一个新版本。如果一个文档有多个版本，则可以设置[默认版本](#)。

编辑现有 Systems Manager 自动化文档

您可以使用 AWS Explorer 查找现有 Systems Manager 自动化文档。当您打开现有文档时，它将显示为 AWS Cloud9 编辑器中的无标题文件。您可以下载三种类型的自动化文档：

- Owned by Amazon：可通过在运行时指定参数来使用的预配置 SSM 文档。
- Owned by me：我已创建并发布至 AWS 的文档。
- Shared with me (已与我共享)：拥有者根据您的 AWS 账户 ID 与您共享的文档。

AWS 中唯一可以更新的文档类型是我拥有的文档。您还可以下载亚马逊共享或拥有的自动化文档，并在 AWS Cloud9 中进行编辑。但是，当您发布到 AWS 时，您必须使用创建新文档或更新您拥有的现有文档。您不能创建具有其他所有者或由亚马逊拥有的文档的新版本。

有关更多信息，请参阅 AWS Systems Manager 用户指南中的 [AWS Systems Manager 文档](#)。

- 在 AWS Explorer 中，对于 Systems Manager，请选择您想要下载的 SSM 文档类型：Owned by Amazon、Owned by me 或 Shared with me。
- 对于特定文档，请打开上下文 (右键单击) 菜单并选择 Download as YAML (下载为 YAML) 或者 Download as JSON (下载为 JSON) 。

格式化的 SSM 文档将显示在新的编辑器选项卡中。

完成编辑后，可以使用 AWS : 发布新 Systems Manager 文档命令以在 AWS Cloud 中创建新文档或更新您拥有的现有文档。

使用版本

Systems Manager 自动化文档使用版本进行变更管理。借助 AWS 工具包，您可以设置文档的默认版本，即运行文档时使用的版本。

要设置默认版本

- 在 AWS Explorer 中，导航到要设置默认版本的文档，请打开该文档的上下文(右键单击)菜单，然后选择 Set default version (设置默认版本)。

Note

如果选择的文档只有一个版本，则无法更改默认版本。

删除 Systems Manager 自动化文档

您可以删除自己在 AWS 工具包中拥有的自动化文档。删除自动化文档将删除该文档以及该文档的所有版本。

Important

- 删除是一个无法撤消的破坏性操作。
- 删除已经启动的自动化文档不会删除运行时创建或修改的 AWS 资源。
- 只有在您拥有文档的情况下才允许删除。

要删除您的自动化文档

- 在 AWS 窗格中，对于 Systems Manager，扩展 Owned by Me 以列出您的文档。
- 打开要删除的文档的上下文 (右键单击) 菜单，然后选择 Delete document (删除文档)。
- 在显示的警告对话框中，选择 Delete (删除) 以确认。

运行 Systems Manager 自动化文档

将自动化文档发布到 AWS 之后，您可以将其运行以在 AWS 账户中代表您执行任务。要运行自动化文档，请使用 AWS Management Console、Systems Manager API、AWS CLI 或者 AWS Tools for PowerShell。有关如何运行 Automation 文档的说明，请参阅 AWS Systems Manager 用户指南中的[运行简单的自动化](#)。

或者，如果您希望使用其中一个带有 Systems Manager API 的 AWS SDK 来运行您的自动化文档，请参阅[AWS SDK 参考](#)。

Important

运行自动化文档可以在 AWS 中创建新的资源，并可能产生计费成本。我们强烈建议您在运行自动化文档之前了解自动化文档将在账户中创建什么。

排除 AWS 工具包中 Systems Manager 自动化文档的问题

我已将自动化文档保存在 AWS 工具包中，但是我没有在 AWS Management Console 中看到。

将自动化文档保存在 AWS 工具包中不会将自动化文档发布到 AWS。有关发布 Automation 文档的更多信息，请参阅[发布 Systems Manager 自动化文档](#)。

发布自动化文档失败，出现权限错误。

确保您的 AWS 凭证配置文件具有发布自动化文档的必要权限。有关权限策略的示例，请参阅[Systems Manager 自动化文档的 IAM 权限](#)。

我已将自动化文档发布至 AWS，但是我没有在 AWS Explorer 窗格中看到。

确保您已将文档发布到您在 AWS Explorer 窗格中浏览的同一 AWS 区域。

我已经删除了自动化文档，但是我仍在为其创建的资源付费。

删除自动化文档不会删除它创建或修改的资源。您可以从[AWS 计费管理控制台](#)识别您创建的资源、浏览您的费用，并选择要从中删除的 AWS 资源。

在 AWS Cloud9 IDE 中使用 Amazon ECR

Amazon Elastic Container Registry (Amazon ECR) 是 AWS 托管式容器注册表服务，它安全且可扩展。可以从 AWS Toolkit Explorer 访问多项 Amazon ECR 服务功能：

- 创建存储库。
- 为您的存储库或已标记的映像创建 AWS App Runner 服务。
- 访问映像标签和存储库 URI 或 ARN。
- 删除映像标签和存储库。

还可以通过安装 AWS CLI 和其他平台，使用 AWS Cloud9 控制台访问全方位的 Amazon ECR 功能。

有关 Amazon ECR 的更多信息，请参阅《Amazon Elastic Container Registry 用户指南》中的[什么是 Amazon ECR ?](#)

先决条件

以下内容已预安装在适用于 AWS Cloud9 Amazon EC2 环境的 AWS Cloud9 IDE 中。从 AWS Cloud9 IDE 中访问 Amazon ECR 服务时需要使用这些内容。

IAM 凭证

您在 AWS 控制台中创建并用于进行身份验证的 IAM 角色。有关 IAM 的更多信息，请参阅[AWS Identity and Access Management 用户指南](#)。

Docker 配置

Docker 已预安装在适用于 AWS Cloud9 Amazon EC2 环境的 AWS Cloud9 IDE 中。有关 Docker 的更多信息，请参阅[安装 Docker 引擎](#)。

AWS CLI 版本 2 配置

AWS CLI 版本 2 已预安装在适用于 AWS Cloud9 Amazon EC2 环境的 AWS Cloud9 IDE 中。有关 AWS CLI 版本 2 的更多信息，请参阅[安装、更新和卸载 AWS CLI 版本 2](#)。

主题

- [在 Amazon Elastic 容器注册表服务中使用 AWS Cloud9](#)

在 Amazon Elastic 容器注册表服务中使用 AWS Cloud9

您可以直接从 AWS IDE 中的资源管理器访问亚马逊弹性容器注册表 (Amazon ECR) Container Registry 服务。AWS Cloud9 您可以使用 Amazon ECR 将程序映像推送到 Amazon ECR 存储库。要开始使用，请执行以下步骤：

1. 创建一个 Dockerfile ，其中包含构建映像所需的信息。
2. 从该 Dockerfile 生成映像并标记该映像以供处理。
3. 创建一个位于 Amazon ECR 实例内部的存储库。
4. 将标记的映像推送到此存储库。

Sections

- [先决条件](#)
- [1. 创建 Dockerfile](#)
- [2. 从您的 Dockerfile 构建映像](#)
- [3. 创建新的存储库](#)
- [4. 推送、提取和删除映像](#)

先决条件

在使用该 AWS 工具包的 Amazon ECR 功能之前 AWS Cloud9 ，请确保先满足这些[先决条件](#)。这些先决条件已预先安装在适用于 AWS Cloud9 Amazon EC2 环境的 AWS Cloud9 IDE 中，并且是访问 Amazon ECR 所必需的。

1. 创建 Dockerfile

Docker 使用名为 Dockerfile 的文件来定义可以推送和存储在远程存储库中的映像。在将映像上载到 ECR 存储库之前，请先创建一个 Dockerfile ，然后从该 Dockerfile 构建映像。

创建 Dockerfile

1. 要导航到要在其中存储 Dockerfile 的目录，请在 AWS Cloud9 IDE 的左侧导航栏中选择 Toggle Tree (切换树) 选项。
2. 创建名为 Dockerfile 的新文件。

Note

AWS Cloud9 IDE 可能会提示您选择文件类型或文件扩展名。如果出现这种情况，请选择纯文本。AWS Cloud9 IDE 有一个“dockerfile”扩展名。但是，我们建议您不要使用它。这是因为该扩展名可能会导致与某些版本的 Docker 或其他关联应用程序发生冲突。

使用 IDE 编辑你的 Dockerfile AWS Cloud9

如果 Dockerfile 具有文件扩展名，请打开该文件的上下文（右键单击）菜单，然后移除文件扩展名。带有扩展名的 Dockerfile 可能会导致与某些版本的 Docker 或其他关联应用程序发生冲突。

从 Dockerfile 中删除文件扩展名后：

1. 直接在 IDE 中打开空的 Dockerfile。AWS Cloud9
2. 将以下示例的内容复制到您的 Dockerfile 中。

Example Dockerfile 映像模板

```
FROM ubuntu:22.04

# Install dependencies
RUN apt-get update && \
    apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
    echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
    echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

这是使用 Ubuntu 22.04 映像的 Dockerfile。RUN 指令将更新软件包缓存。安装一些适用于 Web 服务器的软件包，然后将“Hello World!”内容写入到 Web 服务器的文档根目录。EXPOSE 指令在容器上公开端口 80，而 CMD 指令启动 Web 服务器。

3. 保存您的 Dockerfile。

2. 从您的 Dockerfile 构建映像

您创建的 Dockerfile 包含为程序构建映像所需的信息。在将该映像推送到您的 Amazon ECR 实例之前，请先构建该映像。

从您的 Dockerfile 构建映像

1. 要导航到包含您的 Dockerfile 的目录，请使用 Docker CLI 或与您的 Docker 实例集成的 CLI。
2. 要构建在 Dockerfile 中定义的映像，请在与 Dockerfile 相同的目录中运行 Docker build 命令。

```
docker build -t hello-world .
```

3. 要验证是否已正确创建映像，请运行 Docker images 命令。

```
docker images --filter reference=hello-world
```

Example

输出如下所示。

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 241MB	latest	e9ffedc8c286	4 minutes ago

4. 要运行基于 Ubuntu 22.04 的新构建映像，请使用 echo 命令。

Note

创建或推送映像不需要执行此步骤。但是，您可以看到程序映像在运行时是如何工作的。

```
FROM ubuntu:22.04  
CMD ["echo", "Hello from Docker in Cloud9"]
```

然后，运行并构建 dockerfile。必须从与 dockerfile 相同的目录中运行此命令。

```
docker build -t hello-world .
docker run --rm hello-world
```

Example

输出如下所示。

```
Hello from Docker in Cloud9
```

有关 Docker run 命令的更多信息，请参阅 Docker 网站上的 [Docker Run 参考](#)。

3. 创建新的存储库

要将您的映像上传到您的 Amazon ECR 实例，请创建一个新的存储库来存储它。

创建新的 Amazon ECR 存储库

1. 从 AWS Cloud9 IDE 导航栏中选择“AWS 工具包”图标。
2. 展开 AWS Explorer 菜单。
3. 找到与您的关联 AWS 区域 的默认值 AWS 账户。然后，选择它以查看通过 AWS Cloud9 IDE 提供的服务的列表。
4. 打开 ECR 选项的上下文（右键单击）菜单，以启动 Create new repository（创建新的存储库）过程。然后，选择 Create Repository（创建存储库）。
5. 要完成该过程，请按照提示操作。
6. 该过程完成后，您可以从 AWS 资源管理器菜单的 ECR 部分访问您的新存储库。

4. 推送、提取和删除映像

从 Dockerfile 构建映像并创建存储库后，您可以将映像推送到 Amazon ECR 存储库中。此外，使用带有 Docker 的 AWS Explorer 和 AWS CLI，您可以执行以下操作：

- 从存储库中提取映像。
- 删除存储在存储库中的映像。

- 删除您的存储库。

使用您的原定设置注册表对 Docker 进行身份验证

在 Amazon ECR 实例和 Docker 实例之间交换数据要求进行身份验证。使用注册表对 Docker 进行身份验证：

1. 在 AWS Cloud9 IDE 中打开终端。
2. 使用该 `get-login-password` 方法对您的私有 ECR 注册表进行身份验证，然后输入您的地区和 AWS 账户 ID。

```
aws ecr get-login-password \  
  --region <region> \  
| docker login \  
  --username AWS \  
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Important

在前面的命令中，将 **region** 和 **AWS_account_id** 替换为特定于您的 AWS 账户的信息。有效的 **region** 值为 `us-east-1`。

标记映像并将其推送到存储库

使用您的实例对 Docker 进行身份验证后 AWS，将映像推送到您的存储库。

1. 使用 `docker images` 命令查看您本地存储的映像，并确定要标记的映像。

```
docker images
```

Example

输出如下所示。

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
--------------------	-----	----------	---------


```
hello-world          latest          e9ffedc8c286      4 minutes ago
  241MB
```

2. 使用 `Docker tag` 命令标记映像。

```
docker tag hello-world:latest AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

3. 使用 `Docker push` 命令将标记的映像推送到您的存储库。

Important

请确保您的本地存储库名称与 AWS Amazon EC2 存储库的名称相同。在此示例中，这两个存储库必须称为 `hello-world`。有关使用 `docker` 推送映像的更多信息，请参阅[推送 Docker 映像](#)。

```
docker push AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example

输出如下所示。

```
The push refers to a repository [AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest:
  sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

成功将带标签的图像上传到存储库后，从“资源管理器”选项卡中选择“刷新资源管理器”，刷新 AWS Too AWS Ikit。然后它就会显示在 AWS Cloud9 IDE 的“AWS 资源管理器”菜单中。

从 Amazon ECR 提取映像

- 您可以将映像提取到您的 Docker tag 命令的本地实例。

```
docker pull AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example

输出如下所示。

```
amazonaws.com/hello-world:latest
latest: Pulling from hello-world
Digest: sha256:e02c521fd65eae4ef1acb746883df48de85d55fc85a4172a09a124b11b339f5e
Status: Image is up to date for 922327013870.dkr.ecr.us-west-2.amazonaws.com/hello-world:latest
```

从 Amazon ECR 存储库中删除映像

从 AWS Cloud9 IDE 中删除图像有两种方法。第一种方法是使用 AWS 资源管理器。

1. 在 AWS 资源管理器中，展开 ECR 菜单。
2. 展开要从中删除映像的存储库。
3. 打开与要删除的映像关联的映像标签的上下文（右键单击）菜单。
4. 要删除与该标签关联的所有存储的映像，请选择 Delete Tag...（删除标签...）。

使用 AWS CLI 删除镜像

- 您也可以使用 AWS `ecr batch-delete-image` 命令从存储库中删除图像。

```
aws ecr batch-delete-image \
  --repository-name hello-world \
  --image-ids imageTag=latest
```

Example

输出如下所示。

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "latest",
      "imageDigest":
"sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
    }
  ]
}
```

从 Amazon ECR 实例中删除存储库

从 AWS Cloud9 IDE 中删除存储库有两种方法。第一种方法是使用 AWS 资源管理器：

1. 在 AWS 资源管理器中，展开 ECR 菜单。
2. 打开要删除的存储库的上下文（右键单击）菜单。
3. 选择 Delete Repository...（删除存储库...）。

从 CLI 中删除 Amazon ECR 存储库 AWS

- 您可以使用 `AWS ecr delete-repository` 命令删除存储库。

Note

通常，如果不先删除存储库中包含的映像，则无法删除存储库。但是，如果添加 `--force` 标志，则可以一步删除存储库及其所有映像。

```
aws ecr delete-repository \  
--repository-name hello-world \  
--force
```

Example

输出如下所示。

```
--repository-name hello-world --force
{
  "repository": {
    "repositoryUri": "922327013870.dkr.ecr.us-west-2.amazonaws.com/hello-
world",
    "registryId": "922327013870",
    "imageTagMutability": "MUTABLE",
    "repositoryArn": "arn:aws:ecr:us-west-2:922327013870:repository/hello-
world",
    "repositoryName": "hello-world",
    "createdAt": 1664469874.0
  }
}
```

在 AWS Cloud9 IDE 中使用 AWS IoT

利用 AWS Cloud9 IDE 中的 AWS IoT，您可以与 AWS IoT 服务交互，同时最大限度地减少对 AWS Cloud9 中工作流的干扰。本指南介绍如何开始使用 AWS Cloud9 IDE 中提供的 AWS IoT 服务功能。有关更多信息，请参阅 AWS IoT 开发人员指南中的[什么是 AWS IoT？](#)

AWS IoT 先决条件

要开始在 AWS Cloud9 IDE 中使用 AWS IoT，请确保您的 AWS 账户和 AWS Cloud9 设置满足所有要求。有关特定于 AWS IoT 服务的 AWS 账户要求以及 AWS 用户权限的信息，请参阅《AWS IoT 开发人员指南》中的[开始使用 AWS IoT Core](#)。

AWS IoT 事物

AWS IoT 将设备连接到 AWS 服务和 AWS 资源。您可以使用名为事物的对象，将设备连接到 AWS IoT。事物是特定设备或逻辑实体的表示形式。它可以是物理设备或传感器（例如，灯泡或墙壁上的开关）。有关 AWS IoT 事物的更多信息，请参阅《AWS IoT 开发人员指南》中的[使用 AWS IoT 管理设备](#)。

管理 AWS IoT 事物

AWS Cloud9 IDE 有多项功能可以提高您的事物管理效率。要管理您的 AWS IoT 事物，请按照以下步骤操作：

- [Create a thing](#)
- [Attach a certificate to a thing](#)
- [Detach a certificate from a thing](#)
- [Delete a thing](#)

创建事物

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 打开该事物的上下文（右键单击）菜单，然后选择 Create Thing（创建事物）。
3. 在 Thing Name（事物名称）字段中输入事物的名称，然后按照提示操作。
4. 此步骤完成后，在 Things（事物）部分中可以看到事物图标后跟您指定的名称。

将证书附加到事物

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 在 Things（事物）子部分中，找到您要附加证书的事物。
3. 打开该事物的上下文（右键单击）菜单，然后从上下文菜单中选择 Attach Certificate（附加证书）以打开输入选择器，其中提供了您的证书列表。
4. 从列表中，选择与您要附加到事物的证书相对应的证书 ID。
5. 此步骤完成后，即可在 AWS Explorer 中，将证书作为事物附加到的项目进行访问。

从事物分离证书

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 在 Things（事物）子部分中，找到您要与之分离证书的事物。
3. 打开该事物的上下文（右键单击）菜单，然后选择 Attach Certificate（附加证书）。
4. 此步骤完成后，已分离证书将不再显示在 AWS Explorer 中的事物下。但是，仍然可以从 Certificate（证书）子部分访问该证书。

删除事物

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 在 Things (事物) 子部分中，找到您要删除的事物。
3. 打开该事物的上下文 (右键单击) 菜单，然后选择 Delete Thing (删除事物) 。
4. 完成此步骤后，Things (事物) 子部分中不再提供已删除的事物。

Note

您只能删除没有附加证书的事物。

AWS IoT 证书

证书是用于在您的 AWS IoT 服务与设备之间创建安全连接的常用方法。X.509 证书是一个数字证书，它使用 X.509 公有密钥基础设施标准将公有密钥与证书中包含的身份相关联。有关 AWS IoT 证书的更多信息，请参阅《AWS IoT 开发人员指南》中的 [身份验证 \(IoT \)](#)。

管理证书

AWS Toolkit 提供了多种直接从 AWS Explorer 管理 AWS IoT 证书的方法。以下步骤概述了这些方法：

- [Create a certificate](#)
- [Change a certificate status](#)
- [Attach a policy to a certificate](#)
- [Delete a certificate](#)

创建 AWS IoT 证书

X.509 证书用于连接到您的 AWS IoT 实例。

1. 在 AWS Explorer 中，展开 IoT 服务部分，打开 (右键单击) Certificates (证书) 。
2. 要打开对话框，请从上下文菜单中选择 Create Certificate (创建证书) 。
3. 要保存 RSA 密钥对和 X.509 证书，请在本地文件系统中选择一个目录。

Note

- 默认文件名包含证书 ID 作为前缀。
- 只有 X.509 证书通过 AWS IoT 服务存储在您的 AWS 账户中。
- 您的 RSA 密钥对只能颁发一次，请在出现提示时将其保存到文件系统的安全位置。
- 如果证书或密钥对无法保存到您的文件系统，则 AWS Toolkit 会从 AWS 账户中删除证书。

修改证书状态

单个证书的状态显示在 AWS Explorer 中的证书 ID 旁边，可以设置为 active (活动)、inactive (不活动) 或 revoked (已撤销)。

Note

- 您的证书需要处于 active (活动) 状态才能用来将设备连接到 AWS IoT 服务。
- 无论证书是之前已停用还是默认处于不活动状态，都可以激活 inactive (不活动) 的证书。
- Revoked (已撤销) 的证书无法重新激活。

1. 在 AWS Explorer 中，展开 IoT 服务部分。
 2. 在 Certificates (证书) 子部分中，找到要修改的证书。
 3. 打开证书的上下文 (右键单击) 菜单，该菜单显示对该证书可用的状态更改选项。
- 如果证书的状态为 inactive (不活动)，选择 activate (激活) 可将其状态更改为 active (活动)。
 - 如果证书的状态为 active (活动)，选择 deactivate (停用) 可将其状态更改为 inactive (不活动)。
 - 如果证书的状态为 active (活动) 或 inactive (不活动)，选择 revoke (撤销) 可将其状态更改为 revoked (已撤销)。

Note

如果您选择了附加到 Things (事物) 子部分所显示事物的证书，则所有这些状态更改操作都可用。

将 IoT 策略附加到证书

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 在 Certificates (证书) 子部分中，找到要修改的证书。
3. 打开证书的上下文 (右键单击) 菜单，然后选择 Attach Policy (附加策略) 以打开输入选择器，其中提供了可用策略的列表。
4. 选择要将策略附加到的证书。
5. 完成此步骤后，您选择的策略将作为子菜单项添加到证书中。

将 IoT 策略与证书分离

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 在 Certificates (证书) 子部分中，找到要修改的证书。
3. 展开证书并找到要分离的策略。
4. 打开策略的上下文 (右键单击) 菜单，然后从上下文菜单中选择 Detach (分离)。
5. 完成此步骤后，将无法再从您的证书访问该策略，该策略可从 Policy (策略) 子部分获取。

删除证书

1. 在 AWS Explorer 中，展开 IoT 服务标题。
2. 在 Certificates (证书) 子部分中，找到要删除的证书。
3. 打开证书的上下文 (右键单击) 菜单，然后从上下文菜单中选择 Delete Certificate (删除证书)。

Note

无法删除已经附加到某个事物或者处于活动状态的证书。您可以删除附加了策略的证书。

AWS IoT 策略

AWS IoT Core 策略通过 JSON 文档定义。每个文档必须包括至少一个策略语句。策略定义了 AWS IoT、AWS 和您的设备如何相互交互。有关如何创建策略文档的更多信息，请参阅《AWS IoT 开发人员指南》中的 [IoT 策略](#)。

Note

命名策略采取了版本控制，因此您可以回滚它们。在 AWS Explorer 中，您的 IoT 策略列在 AWS IoT 服务的 Policies (策略) 子部分下。您可以通过展开策略来查看策略版本。默认版本由星号 (*) 表示。

管理策略

AWS Cloud9 IDE 为您提供了多种管理 AWS IoT 服务策略的方法。以下是直接在 VS Code 中通过 AWS Explorer 管理或修改策略的方法：

- [Create a policy](#)
- [Upload a new policy version](#)
- [Edit a policy version](#)
- [Change the policy version default](#)
- [Change the policy version default](#)

创建 AWS IoT策略

Note

您可以从 AWS Explorer 创建新策略。但是，定义策略的 JSON 文档必须已经存在于您的文件系统中。

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 打开 Policies (策略) 子部分的上下文 (右键单击) 菜单，要打开 Policy Name (策略名称) 输入字段，请选择 Create Policy from Document (从文档创建策略)。
3. 输入名称并按照提示打开一个对话框，其中会要求您从文件系统中选择 JSON 文档。

4. 选择包含您的策略定义的 JSON 文件，策略完成后将在 AWS Explorer 中可用。

上传新的 AWS IoT 策略版本

您可以通过向策略上传 JSON 文档来创建策略的新版本。

Note

要使用 AWS Explorer 创建新版本，新的 JSON 文档必须存在于您的文件系统中。

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 展开 Policies (策略) 子部分以查看您的 AWS IoT 策略。
3. 打开要更新的策略的上下文 (右键单击) 菜单，然后选择 Create new version from Document (从文档创建新版本)。
4. 对话框打开时，选择包含策略定义更新的 JSON 文件。

新版本可在 AWS Explorer 中从策略进行访问。

编辑 AWS IoT 策略版本

您可以使用 AWS Cloud9 打开和编辑策略文档。完成文档编辑后，请将其保存到文件系统中。然后，从 AWS Explorer 将其上传到您的 AWS IoT 服务。

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 展开 Policies (策略) 子部分并找到要更新的策略。
3. 要打开 Policy Name (策略名称)，请从 Document (文档) 选择 Create Policy (创建策略)。
4. 展开要更新的策略，然后打开要编辑的策略版本的上下文 (右键单击) 菜单。
5. 要在 AWS Cloud9 中打开策略版本，请从上下文菜单中选择 View (查看) 以打开策略版本。
6. 打开策略文档后，编辑并保存更改。

Note

此时，您对策略所做的更改仅保存到本地文件系统中。要更新版本并使用 AWS Explorer 对其进行跟踪，请重复 [Upload a new policy version](#) 中的步骤。

选择新的默认策略版本

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 展开 Policies (策略) 子部分并找到要更新的策略。
3. 展开要更新的策略，然后打开要设置的策略版本的上下文 (右键单击) 菜单并选择 Set as Default (设置为默认)。

完成后，您选择的新默认版本旁边会显示一个星号。

删除策略

Note

在删除策略或策略版本之前，请确保满足以下条件：

- 您无法删除已附加到证书的策略。
- 如果策略有任何非默认版本，则无法删除该策略。
- 只有选择了新的默认版本或删除了整个策略，您才能删除策略的默认版本。
- 在删除整个策略之前，必须删除该策略的所有非默认版本。

1. 在 AWS Explorer 中，展开 IoT 服务部分。
2. 展开 Policies (策略) 子部分并找到要更新的策略。
3. 展开要更新的策略，然后打开要删除的策略版本的上下文 (右键单击) 菜单并选择 Delete (删除)。
4. 删除版本后，它在 AWS Explorer 中不再可见。
5. 如果策略只剩下默认版本，请打开父策略的上下文 (右键单击) 菜单，然后选择 Delete (删除)。

使用 Amazon Elastic Container Service

AWS Cloud9 IDE 为 [Amazon Elastic Container Service \(Amazon ECS\)](#) 提供一些支持。您可以使用 AWS Cloud9 IDE 管理 Amazon ECS 资源。例如，您可以创建任务定义。

主题

- [适用于 AWS Cloud9 的 AWS Toolkit 中的 Amazon Elastic Container Service Exec](#)

适用于 AWS Cloud9 的 AWS Toolkit 中的 Amazon Elastic Container Service Exec

您可以使用适用于 AWS Cloud9 的 AWS Toolkit 在 Amazon Elastic Container Service (Amazon ECS) 容器中发出单条命令。您可以使用 Amazon ECS Exec 功能完成此操作。

Important

启用和禁用 Amazon ECS Exec 会更改您的 AWS 账户中 ECS 资源的状态。更改包括停止和重新启动服务。此外，在 Amazon ECS Exec 处于启用状态时更改资源状态可能会导致不可预测的结果。有关 Amazon ECS 的更多信息，请参阅《Amazon ECS 开发人员指南》中的[使用 Amazon ECS Exec 进行调试](#)。

Amazon ECS Exec 先决条件

在使用 Amazon ECS Exec 功能之前，您必须满足某些先决条件。

Amazon ECS 要求

根据您的任务是托管在 Amazon EC2 上还是 AWS Fargate (Fargate) 上，Amazon ECS Exec 具有不同的版本要求。

- 如果您使用 Amazon EC2，必须使用 2021 年 1 月 20 日之后发布的经 Amazon ECS 优化的 AMI，代理版本为 1.50.2 或更高。有关更多信息，请参阅《Amazon ECS 开发人员指南》中的[经 Amazon ECS 优化的 AMI](#)。
- 如果您使用 AWS Fargate，则必须使用平台版本 1.4.0 或更高版本。有关更多信息，请参阅《Amazon ECS 开发人员指南》中的[AWS Fargate 平台版本](#)。

AWS 账户配置和 IAM 权限

要使用 Amazon ECS Exec 功能，您必须拥有与您的 AWS 账户关联的现有 Amazon ECS 集群。Amazon ECS Exec 使用 Systems Manager 与集群中的容器建立连接。Amazon ECS 需要特定的任务 IAM 角色权限才能与 SSM 服务通信。

有关特定于 Amazon ECS Exec 的 IAM 角色和策略的信息，请参阅《Amazon ECS 开发人员指南》中的[ECS Exec 所需的 IAM 权限](#)。

使用 Amazon ECS Exec

您可以通过适用于 AWS Cloud9 的 AWS Toolkit 中的 AWS Explorer 来启用或禁用 Amazon ECS Exec。启用 Amazon ECS Exec 后，从 Amazon ECS 菜单中选择容器，然后对它们运行命令。

启用 Amazon ECS Exec

1. 从 AWS Explorer 中，找到并展开 Amazon ECS 菜单。
2. 展开包含您要修改的服务的集群。
3. 打开服务的上下文菜单（右键单击），然后选择 Enable Command Execution（启用命令执行）。

Important

此步骤将启动服务的新部署，可能需要几分钟。有关更多信息，请参阅本部分开头的注释。

禁用 Amazon ECS Exec

1. 从 AWS Explorer 中，找到并展开 Amazon ECS 菜单。
2. 展开包含所需服务的集群。
3. 打开服务的上下文菜单（右键单击），然后选择 Disable Command Execution（禁用命令执行）。

Important

此步骤将启动服务的新部署，可能需要几分钟。有关更多信息，请参阅本部分开头的注释。

对容器运行命令

要使用 AWS Explorer 对容器运行命令，必须启用 Amazon ECS Exec。如果未启用，请参阅本节中的 [启用 Amazon ECS Exec](#) 步骤。

1. 从 AWS Explorer 中，找到并展开 Amazon ECS 菜单。
2. 扩展具有所需服务的集群。
3. 扩展此服务以列出关联的容器。

4. 打开容器的上下文菜单（右键单击），然后选择 Run Command in Container（在容器中运行命令）。
5. 将打开一个提示符，其中包含正在运行的任务列表。选择您想要的任务 ARN。

Note

如果只有一个任务在运行，则不会打开提示符，而是自动选择该任务。

6. 出现提示时，输入要运行的命令，然后按 Enter 以继续。

使用 Amazon EventBridge

AWS Toolkit for AWS Cloud9 为 [Amazon EventBridge](#) 提供支持。使用 AWS Toolkit for AWS Cloud9，您可以处理 EventBridge 的某些方面，例如架构。

主题

- [使用 Amazon EventBridge Schemas](#)

使用 Amazon EventBridge Schemas

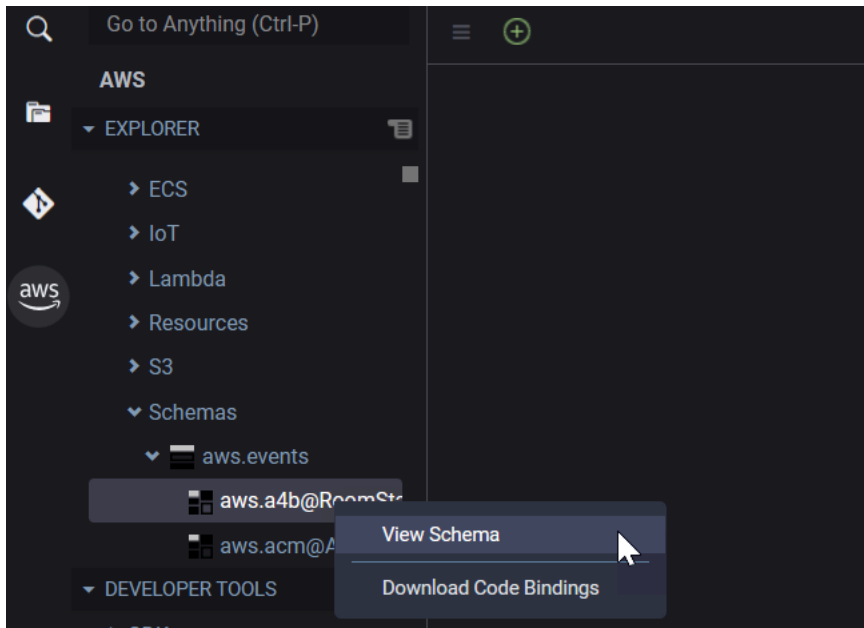
您可以使用 AWS Toolkit for AWS Cloud9，对 [Amazon EventBridge 架构](#) 执行各种操作。

先决条件

要使用的 EventBridge 架构必须在您的 AWS 账户中可用。如果不可用，请创建或上传架构。有关更多信息，请参阅 [Amazon EventBridge 用户指南](#) 中的 [Amazon EventBridge 架构](#)。

查看可用架构

1. 在 AWS Explorer 中，展开 Schemas (架构)。
2. 展开包含您希望查看的架构的注册表的名称。例如，AWS 提供的许多架构位于 aws.events 注册表中。
3. 要在编辑器中查看架构，请打开该架构的上下文（右键单击）菜单，然后选择 View Schema（查看架构）。



查找可用架构

在 AWS Explorer 中，执行以下一项或多项操作：

- 首先输入要查找的架构的标题。AWS Explorer 突出显示包含匹配项的架构标题。（必须扩展注册表，您才能看到突出显示的标题。）
- 打开 Schemas (架构) 的上下文 (右键单击) 菜单，然后选择 Search Schemas (搜索架构)。或者展开 Schemas (架构)，打开包含您希望查找的架构的注册表的上下文 (右键单击) 菜单，然后选择 Search Schemas in Registry (在注册表中搜索架构)。在 EventBridge Schemas Search (EventBridge 架构搜索) 对话框中，开始输入要查找的架构的标题。对话框中将显示包含匹配项的架构标题。

要在对话框中显示架构，请选择架构标题。

为可用架构生成代码

1. 在 AWS Explorer 中，展开 Schemas (架构)。
2. 展开包含您希望生成代码的架构的注册表的名称。
3. 在架构标题上打开上下文 (右键单击) 菜单，然后选择 Download code bindings (下载代码绑定)。

4. 在显示的向导页面中，选择以下内容：

- 架构的版本
- 代码绑定语言
- 本地开发计算机上要将生成的代码存储到的工作区文件夹

AWS Cloud9 的教程

您是首次使用 AWS Cloud9 吗？在 [入门：基本教程](#) 中浏览 IDE 相关信息。

试验这些教程和示例代码来增进了解并充满信心地将 AWS Cloud9 与各种编程语言以及 AWS 服务结合使用。

主题

- [适用于 AWS Cloud9 的 AWS Command Line Interface 和 aws-shell 教程](#)
- [适用于 AWS Cloud9 的 AWS CodeCommit 教程](#)
- [适用于 AWS Cloud9 的 Amazon DynamoDB 教程](#)
- [适用于 AWS Cloud9 的 AWS CDK 教程](#)
- [适用于 AWS Cloud9 的 LAMP 教程](#)
- [适用于 AWS Cloud9 的 WordPress 教程](#)
- [适用于 AWS Cloud9 的 Java 教程](#)
- [适用于 AWS Cloud9 的 C++ 教程](#)
- [适用于 AWS Cloud9 的 Python 教程](#)
- [适用于 AWS Cloud9 的 .NET 教程](#)
- [Node.js 的教程 AWS Cloud9](#)
- [的 PHP 教程 AWS Cloud9](#)
- [AWS Cloud9 中的 Ruby](#)
- [适用于 AWS Cloud9 的 Go 教程](#)
- [适用于 AWS Cloud9 的 TypeScript 教程](#)
- [适用于 Docker 的教程 AWS Cloud9](#)
- [相关教程](#)

适用于 AWS Cloud9 的 AWS Command Line Interface 和 aws-shell 教程

按照以下教程操作，您可以在 AWS Cloud9 开发环境中设置 AWS Command Line Interface (AWS CLI) 和/或 aws-shell。AWS CLI 和 aws-shell 是统一工具，这些工具提供用于与 AWS 的所有部分交

互的一致界面。您可以使用 AWS CLI 而不是 AWS Management Console 来快速运行命令以与 AWS 交互，其中一些命令可以通过 AWS CLI 或 AWS CloudShell 来运行。

有关 AWS CLI 的更多信息，请参阅 [AWS Command Line Interface 用户指南](#)。对于 aws-shell，请参阅以下资源：

- GitHub 网站上的 [aws-shell](#)
- pip 网站上的 [aws-shell](#)

有关可通过 AWS CLI 运行以与 AWS 交互的命令列表，请参阅 [AWS CLI 命令参考](#)。您可以通过 AWS CloudShell 使用相同的命令，不同之处在于命令开头没有 aws 前缀。

创建此示例可能会导致您的 AWS 账户产生相关费用。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：在您的环境中安装 AWS CLI 和/或 aws-shell](#)
- [步骤 2：在环境中设置凭证管理](#)
- [步骤 3：在环境中使用 AWS CLI 或 aws-shell 运行基本命令](#)
- [步骤 4：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅 [在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅 [在 AWS Cloud9 中打开环境](#)。

步骤 1：在您的环境中安装 AWS CLI 和/或 aws-shell

在此步骤中，您将在环境中使用 AWS Cloud9 IDE 安装 AWS CLI 和/或 aws-shell，这样您就可以运行命令来与 AWS 交互。

如果您使用的是 AWS Cloud9 EC2 开发环境并且仅希望使用 AWS CLI，则可以向前跳至 [步骤 3：在环境中使用 AWS CLI 或 aws-shell 运行基本命令](#)。这是因为，EC2 环境中已经安装了 AWS CLI，并且环境中已经设置了一组 AWS 访问凭证。有关更多信息，请参阅 [AWS 托管的临时证书](#)。

如果您不使用 EC2 环境，请执行以下操作安装 AWS CLI：

1. 在您的环境处于打开状态时，在 IDE 中检查是否已安装 AWS CLI。在终端中运行 `aws --version` 命令。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口) > New Terminal (新建终端)。）如果已安装 AWS CLI，则将显示版本号以及其他信息，例如 Python 的版本号、您的 Amazon EC2 实例或您自己的服务器的操作系统版本号。如果已安装 AWS CLI，请向前跳至 [步骤 2：在环境中设置凭证管理](#)。
2. 要安装 AWS CLI，请参阅 AWS Command Line Interface 用户指南中的 [安装 AWS Command Line Interface](#)。例如，对于运行 Amazon Linux 的 EC2 环境，请在终端运行以下三个命令（一次运行一条命令）以安装 AWS CLI。

```
sudo yum -y update           # Install the latest system updates.
sudo yum -y install aws-cli  # Install the AWS CLI.
aws --version                # Confirm the AWS CLI was installed.
```

例如，对于运行 Ubuntu Server 的 EC2 环境，请在终端运行以下三个命令（一次运行一个命令）以安装 AWS CLI。

```
sudo apt update             # Install the latest system updates.
sudo apt install -y awscli  # Install the AWS CLI.
aws --version               # Confirm the AWS CLI was installed.
```

如果要安装 aws-shell，请执行以下操作：

1. 在环境处于打开状态时，在 IDE 中检查是否已安装 aws-shell。在终端中运行 `aws-shell` 命令。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口) > New Terminal (新建终端)。）如果已安装 aws-shell，则将显示 `aws>` 提示。如果已安装 aws-shell，请向前跳至 [步骤 2：在环境中设置凭证管理](#)。
2. 要安装 aws-shell，您可以使用 pip。要使用 pip，您必须已安装 Python。

要检查是否已安装 Python（并在需要时安装），请按照 Python 示例中 [步骤 1：安装 Python](#) 中的说明操作，然后返回此主题。

要检查是否已安装 pip，请在终端中运行 **pip --version** 命令。如果已安装 pip，则会显示版本号。如果未安装 pip，请在终端中运行以下三个命令（一次运行一个命令）以安装它。

```
wget https://bootstrap.pypa.io/get-pip.py # Get the pip install file.
sudo python get-pip.py                  # Install pip. (You might need to run
'sudo python2 get-pip.py' or 'sudo python3 get-pip.py' instead, depending on how
Python is installed.)
rm get-pip.py                           # Delete the pip install file, as it is
no longer needed.
```

3. 要使用 pip 安装 aws-shell，请运行以下命令。

```
sudo pip install aws-shell
```

步骤 2：在环境中设置凭证管理

每次您使用 AWS CLI 或 aws-shell 调用 AWS 服务时，都必须提供一组凭证才能进行调用。这些凭证决定 AWS CLI 或 aws-shell 是否具有相应的权限来进行该调用。如果凭证没有包括合适的权限，调用将失败。

如果您使用的是 AWS Cloud9 EC2 开发环境，则可以向前跳至 [步骤 3：在环境中使用 AWS CLI 或 aws-shell 运行基本命令](#)。这是因为 EC2 环境中已设置了凭证。有关更多信息，请参阅[AWS 托管的临时证书](#)。

如果您不使用 EC2 环境，则必须在环境中手动存储凭证。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

步骤 3：在环境中使用 AWS CLI 或 aws-shell 运行基本命令

在此步骤中，您将在环境中使用 AWS CLI 或 aws-shell 以在 Amazon S3 中创建存储桶，列出可用存储桶，然后删除存储桶。

1. 如果您想要使用 aws-shell 但尚未启动它，请运行 `aws-shell` 命令启动 aws-shell。此时将显示 `aws>` 提示。
2. 创建存储桶。使用 AWS CLI 运行 `aws s3 mb` 命令或使用 aws-shell 运行 `s3 mb` 命令，提供要创建的存储桶的名称。在此示例中，我们使用名为 `cloud9-123456789012-bucket` 的存储桶，其中 `123456789012` 是您的 AWS 账户 ID。如果您使用了其他名称，请在整个步骤中使用该名称进行替换。

```
aws s3 mb s3://cloud9-123456789012-bucket # For the AWS CLI.
s3 mb s3://cloud9-123456789012-bucket    # For the aws-shell.
```

Note

存储桶名称在整个 AWS 中都必须唯一，而不仅仅在您的 AWS 账户中唯一。前面建议的存储桶名称可帮助您起一个唯一的存储桶名称。如果您收到消息，其中包含错误 `BucketAlreadyExists`，则必须使用其他存储桶名称再次运行命令。

3. 列出您的可用存储桶。使用 AWS CLI 运行 `aws s3 ls` 命令，或者使用 `aws-shell` 运行 `s3 ls` 命令。此时将显示您的可用存储桶列表。
4. 删除存储桶。使用 AWS CLI 运行 `aws s3 rb` 命令或使用 `aws-shell` 运行 `s3 rb` 命令，提供要删除的存储桶的名称。

```
aws s3 rb s3://cloud9-123456789012-bucket # For the AWS CLI.
s3 rb s3://cloud9-123456789012-bucket    # For the aws-shell.
```

要确认存储桶是否已删除，请使用 AWS CLI 再次运行 `aws s3 ls` 命令或使用 `aws-shell` 再次运行 `s3 ls` 命令。已删除的存储桶的名称应不再显示在列表中。

Note

如果您要继续使用存储桶，则无需删除它。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的[向存储桶添加对象](#)。另请参阅 AWS CLI 命令参考中的[s3 命令](#)。（请记住，如果您不删除存储桶，则可能会导致向您的 AWS 账户持续收取费用。）

要继续试用 AWS CLI，请参阅 AWS Command Line Interface 用户指南中的[使用 Amazon Web Services](#) 以及 [AWS CLI 命令参考](#)。要继续试用 `aws-shell`，请参阅 [AWS CLI 命令参考](#)，注意命令开头不需要 `aws` 前缀。

步骤 4：清除

如果您在使用 `aws-shell`，则可以通过运行 `.exit` 或 `.quit` 命令来停止使用它。

为防止在您使用完该示例后一直对您的 AWS 账户收费，应删除环境。有关说明，请参阅 [在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 AWS CodeCommit 教程

您可以使用 AWS CodeCommit 教程设置 AWS Cloud9 开发环境以与 CodeCommit 中的远程代码存储库进行交互。CodeCommit 是一个源代码控制服务，可用于在 AWS Cloud 中专门存储和管理 Git 存储库。有关 CodeCommit 的更多信息，请参阅 [AWS CodeCommit 用户指南](#)。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 CodeCommit 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [AWS CodeCommit 定价](#)。

- [先决条件](#)
- [步骤 1：设置具有所需访问权限的 IAM 组](#)
- [步骤 2：在 AWS CodeCommit 中创建存储库](#)
- [步骤 3：将您的环境连接到远程存储库](#)
- [步骤 4：将远程存储库克隆到您的环境](#)
- [步骤 5：向存储库添加文件](#)
- [步骤 6：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：设置具有所需访问权限的 IAM 组

假设您的 AWS 凭证与您的 AWS 账户中的管理员用户相关联，并且您希望通过该用户使用 CodeCommit，则请向前跳到[步骤 2：在 AWS CodeCommit 中创建存储库](#)。

您可以使用 [AWS Management Console](#) 或 [AWS Command Line Interface \(AWS CLI\)](#) 完成此步骤。

使用控制台设置具有所需访问权限的 IAM 组

1. 登录 AWS Management Console (如果尚未登录) 。

对于该步骤，我们建议使用您的 AWS 账户 中的管理员用户的凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户 管理员。

2. 打开 IAM 控制台。为此，请在控制台的导航栏中选择 Service (服务)。然后，选择 IAM。

3. 选择 Groups (组) 。

4. 选择组名。

5. 在 Permissions (权限) 选项卡中，在 Managed Policies (托管式策略) 处，选择 Attach Policy (附加策略) 。

6. 在策略名称列表中，选中以下框之一：

- 选择 AWSCodeCommitPowerUser 可访问 CodeCommit 的所有功能和存储库相关资源。但是，这不允许您删除 CodeCommit 存储库或在其他 AWS 服务 (如 Amazon CloudWatch Events) 中创建或删除存储库相关资源。
- 选中 AWSCodeCommitFullAccess 以完全控制 AWS 账户 中的 CodeCommit 存储库和相关资源。这包括可以删除存储库。

如果在列表中看不到任一策略名称，请在 Filter (筛选条件) 框中输入策略名称以显示它们。

7. 选择 Attach Policy (附加策略) 。

要查看这些 AWS 托管式策略为组授予的访问权限列表，请参阅 AWS CodeCommit 用户指南中的 [适用于 AWS CodeCommit 的 AWS 托管式 \(预定义 \) 策略](#)。

向前跳到 [步骤 2：在 AWS CodeCommit 中创建存储库](#)。

使用 AWS CLI 设置具有所需访问权限的 IAM 组

运行 IAM attach-group-policy 命令，并指定组的名称以及描述所需访问权限的 AWS 托管式策略的 Amazon 资源名称 (ARN)。语法如下所示。

```
aws iam attach-group-policy --group-name MyGroup --policy-arn POLICY_ARN
```

在上述命令中，将 MyGroup 替换为组的名称。将 POLICY_ARN 替换为 AWS 托管式策略的 ARN：

- `arn:aws:iam::aws:policy/AWSCodeCommitPowerUser`，以访问 CodeCommit 的所有功能和存储库相关资源。但是，它不允许您删除 CodeCommit 存储库或在其他 AWS 服务（如 Amazon CloudWatch Events）中创建或删除存储库相关资源。
- `arn:aws:iam::aws:policy/AWSCodeCommitFullAccess`，以完全控制 AWS 账户中的 CodeCommit 存储库和相关资源。这包括可以删除存储库。

要查看这些 AWS 托管式策略为组授予的访问权限列表，请参阅 AWS CodeCommit 用户指南中的[适用于 AWS CodeCommit 的 AWS 托管式（预定义）策略](#)。

步骤 2：在 CodeCommit 中创建存储库

在此步骤中，您使用 CodeCommit 控制台在 CodeCommit 中创建一个远程代码存储库。

如果已具有一个 CodeCommit 存储库，请向前跳到[步骤 3：将您的环境连接到远程存储库](#)。

您可以使用 [AWS Management Console](#) 或 [AWS Command Line Interface \(AWS CLI\)](#) 完成此步骤。

使用控制台在 CodeCommit 中创建存储库

1. 假设您在上一步中以管理员用户身份登录到 AWS Management Console，并且不希望使用管理员用户创建存储库。然后，退出 AWS Management Console。
2. 打开 CodeCommit 控制台 (<https://console.aws.amazon.com/codecommit>)。
3. 在控制台的导航栏中，使用区域选择器选择要在其中创建存储库的 AWS 区域（如美国东部（俄亥俄州））。
4. 如果显示欢迎页面，请选择 Get started（开始使用）。否则，选择 Create repository（创建存储库）。
5. 在 Create repository（创建存储库）页面中，为 Repository name（存储库名称）键入新存储库的名称（例如 MyDemoCloud9Repo）。如果您选择不同的名称，请在本示例中替换为该名称。
6. （可选）对于 Description（描述），请键入有关该存储库的信息。例如，您可以输入：This is a demonstration repository for the AWS Cloud9 sample.
7. 选择 Create repository（创建存储库）。将显示 Connect to your repository（连接到您的存储库）窗格。选择 Close（关闭），因为将在本主题后面以不同方式连接到存储库。

向前跳到[步骤 3：将您的环境连接到远程存储库](#)。

使用 AWS CLI 在 CodeCommit 中创建存储库

运行 `create-repository` 命令 AWS CodeCommit。指定存储库的名称、可选描述以及要在其中创建存储库的 AWS 区域。

```
aws codecommit create-repository --repository-name MyDemoCloud9Repo --repository-  
description "This is a demonstration repository for the AWS Cloud9 sample." --region  
us-east-2
```

在上述命令中，将 `us-east-2` 替换为要在其中创建存储库的 AWS 区域的 ID。有关受支持的区域的列表，请参阅《Amazon Web Services 一般参考》中的 [AWS CodeCommit](#)。

如果您选择使用不同的存储库名称，请在本示例中替换为该名称。

步骤 3：将您的环境连接到远程存储库

在该步骤中，您使用 AWS Cloud9 IDE 连接到在上一步中创建或确定的 CodeCommit 存储库。

Note

如果您更喜欢通过可视化界面使用 Git，可以克隆远程存储库。然后，您可以使用 IDE 中提供的 [Git 面板](#) 功能添加文件。

完成以下若干组过程之一，具体取决于您的 AWS Cloud9 开发环境类型。

环境类型	执行以下过程
EC2 环境	<ol style="list-style-type: none">从 IDE 的终端会话中，运行以下两个命令：<pre>git config --global credential l.helper '!aws codecommit credential l-helper \$@' git config --global credential l.UseHttpPath true</pre> <p>有关更多信息，请参阅 AWS CodeCommit 用户指南中集成 AWS Cloud9 与 AWS CodeCommit 部分的 步骤 2：在 AWS Cloud9</p>

环境类型	执行以下过程
	<p>EC2 开发环境上配置 AWS CLI 凭证辅助程序。</p> <ol style="list-style-type: none">2. 向前跳到本主题后面的步骤 4：将远程存储库克隆到您的环境中。
SSH 环境	<ol style="list-style-type: none">1. 如果在环境中尚未安装 Git，请在 IDE 中使用终端会话安装它。有关更多信息，请参阅《AWS CodeCommit 用户指南》中的在 Linux、macOS 或 Unix 上设置到 AWS CodeCommit 存储库中的 SSH 连接的步骤部分的步骤 2：安装 Git。2. 完成 AWS CodeCommit 用户指南中的在 Linux、macOS 或 Unix 上设置到 AWS CodeCommit 存储库的 SSH 连接的步骤部分的步骤 3：在 Linux、macOS 或 Unix 上配置凭证。 <p>在指示您登录到 AWS Management Console 并打开 IAM 控制台时，我们建议使用您的 AWS 账户中的管理员用户的凭证登录。如果您无法执行此操作，请咨询您的 AWS 账户管理员。</p> <ol style="list-style-type: none">3. 向前跳到本主题后面的步骤 4：将远程存储库克隆到您的环境中。

步骤 4：将远程存储库克隆到您的环境

在该步骤中，您使用 AWS Cloud9 IDE 将 CodeCommit 中的远程存储库克隆到您的环境中。

要克隆存储库，请运行 **git clone** 命令。将 `CLONE_URL` 替换为存储库的克隆 URL。

```
git clone CLONE_URL
```

对于 EC2 环境，请提供以 `https://` 开头的 HTTPS 克隆 URL。对于 SSH 环境，请提供以 `ssh://` 开头的 SSH 克隆 URL。

要获取存储库的完整克隆 URL，请参阅 AWS CodeCommit 用户指南中的[使用 AWS CodeCommit 控制台查看存储库详细信息](#)。

如果在您的存储库中没有任何文件，则会显示一条警告消息，如 `You appear to have cloned an empty repository.`。这是预期行为。您将稍后处理该问题。

步骤 5：向存储库添加文件

在这一步中，您将在 AWS Cloud9 环境的克隆存储库中创建三个简单文件。接下来，将文件添加到克隆存储库的 Git 暂存区域。最后，您提交暂存的文件，并将提交内容推送到 CodeCommit 中的远程存储库。

如果在克隆的存储库中已具有文件，则该过程已完成，您可以跳过本示例的其余部分。

在存储库中添加文件

1. 创建新的文件。在菜单栏上，选择 File (文件)、New File (新建文件)。
2. 在文件中输入以下内容，然后选择 File (文件) > Save (保存)，以在您的 AWS Cloud9 环境的 MyDemoCloud9Repo 目录中将该文件保存为 `bird.txt`。

```
bird.txt
-----
Birds are a group of endothermic vertebrates, characterized by feathers,
toothless beaked jaws, the laying of hard-shelled eggs, a high metabolic
rate, a four-chambered heart, and a lightweight but strong skeleton.
```

Note

要确认您将该文件保存到正确的目录中，请在 Save As (另存为) 对话框中选择 MyDemoCloud9Repo 文件夹。然后，确保 Folder (文件夹) 显示 / MyDemoCloud9Repo。

3. 再创建两个名为 `insect.txt` 和 `reptile.txt` 的文件，内容如下。将这些文件保存到相同的 MyDemoCloud9Repo 目录中。

```
insect.txt
-----
Insects are a class of invertebrates within the arthropod phylum that
have a chitinous exoskeleton, a three-part body (head, thorax, and abdomen),
three pairs of jointed legs, compound eyes, and one pair of antennae.
```

```
reptile.txt
-----
Reptiles are tetrapod (four-limbed vertebrate) animals in the class
Reptilia, comprising today's turtles, crocodilians, snakes,
amphisbaenians, lizards, tuatara, and their extinct relatives.
```

4. 在终端中，运行 **cd** 命令以切换到 MyDemoCloud9Repo 目录。

```
cd MyDemoCloud9Repo
```

5. 通过运行 **git status** 命令，确认这些文件已成功保存到 MyDemoCloud9Repo 目录中。所有三个文件将作为未跟踪的文件列出。

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    bird.txt
    insect.txt
    reptile.txt
```

6. 运行 **git add** 命令，以将这些文件添加到 Git 暂存区域中。

```
git add --all
```

7. 再次运行 **git status** 命令，以确认这些文件已成功添加到 Git 暂存区域中。所有三个文件现在列为更改待提交。

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   bird.txt
    new file:   insect.txt
    new file:   reptile.txt
```

8. 运行 **git commit** 命令以提交暂存的文件。

```
git commit -m "Added information about birds, insects, and reptiles."
```

9. 运行 **git push** 命令，以将提交内容推送到 CodeCommit 中的远程存储库。

```
git push -u origin master
```

10. 确认是否成功推送这些文件。如果尚未打开 CodeCommit 控制台，请将其打开 (<https://console.aws.amazon.com/codecommit>)。
11. 在顶部导航栏的右侧边缘附近，选择您在其中创建了存储库的 AWS 区域（如美国东部（俄亥俄州））。
12. 在 Dashboard（控制面板）页中，选择 MyDemoCloud9Repo。将显示这三个文件。

要继续试验 CodeCommit 存储库，请参阅 [AWS CodeCommit 用户指南](#) 中的浏览存储库的内容。

如果不熟悉 Git 并且不希望毁坏您的 CodeCommit 存储库，请试验 [Try Git](#) 网站上的示例 Git 存储库。

步骤 6：清除

为防止在您使用完该示例后一直对您的 AWS 账户收费，请删除 CodeCommit 存储库。有关说明，请参阅 AWS CodeCommit 用户指南中的 [删除 AWS CodeCommit 存储库](#)。

同时确保删除环境。有关说明，请参阅 [删除环境](#)。

适用于 AWS Cloud9 的 Amazon DynamoDB 教程

此教程让您能够设置 AWS Cloud9 开发环境以与 Amazon DynamoDB 一起使用。

DynamoDB 是一种完全托管式 NoSQL 数据库服务。您可以使用 DynamoDB 创建一个数据库表来存储和检索任意量级的数据，并支持任何级别的请求流量。DynamoDB 自动将表的数据和流量分布到足够数量的服务器上，以处理指定的请求容量和存储的数据量，同时保持一致且快速的性能。有关更多信息，请参阅 AWS 网站上的 [Amazon DynamoDB](#)。

创建此示例可能会导致您的 AWS 账户产生相关费用。其中包括可能对 Amazon EC2 和 DynamoDB 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon DynamoDB 定价](#)。

有关其他 AWS 数据库服务的信息，请参阅 AWS 网站上的 [Amazon Relational Database Service \(RDS\)](#)、[Amazon ElastiCache](#) 和 [Amazon Redshift](#)。另请参阅 AWS 网站上的 [AWS Database Migration Service](#)。

- [先决条件](#)
- [步骤 1：在您的环境中安装和配置 AWS CLI 和/或 AWS CloudShell](#)

- [步骤 2：创建表](#)
- [步骤 3：向表添加项目](#)
- [步骤 4：向表添加多个项目](#)
- [步骤 5：创建全局二级索引](#)
- [步骤 6：获取表中的项目](#)
- [步骤 7：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：在您的环境中安装和配置 AWS CLI 和/或 AWS CloudShell

在此步骤中，您将在环境中使用 AWS Cloud9 IDE 安装和配置 AWS CLI 和/或 AWS CloudShell，这样您就可以运行命令来与 DynamoDB 交互。然后，您将使用 AWS CLI 运行基本 DynamoDB 命令来测试安装和配置。

1. 要在您的环境中设置 AWS CLI 或 AWS CloudShell 的凭证管理以及安装 AWS CLI 和/或 AWS CloudShell，请按照[AWS CLI 和 AWS CloudShell 示例](#)中的步骤 1 和 2 操作，然后返回此主题。如果您已在环境中安装并配置了 AWS CLI 和/或 AWS CloudShell，则无需重复此过程。
2. 在环境中，通过从终端会话运行 DynamoDB **list-tables** 命令列出现有的 DynamoDB 表（如果有），测试 AWS CLI 和/或 aws-shell 的安装和配置。要开始新的终端会话，请在菜单栏上依次选择 Window（窗口）、New Terminal（新建终端）。

```
aws dynamodb list-tables # For the AWS CLI.  
dynamodb list-tables     # For the aws-shell.
```

Note

在本示例中，如果您使用的是 `aws-shell`，请忽略以 `aws` 开头的每个命令中的 `aws`。要启动 `aws-shell`，请运行 `aws-shell` 命令。要停止使用 `aws-shell`，请运行 `.exit` 或 `.quit` 命令。

如果此命令成功，它会输出一个 `TableNames` 数组，其中包含您现在已有的 DynamoDB 表的列表。如果您还没有 DynamoDB 表，则 `TableNames` 数组将为空。

```
{
  "TableNames": []
}
```

如果您有任何 DynamoDB 表，则 `TableNames` 数组包含表名称列表。

步骤 2：创建表

在此步骤中，您在 DynamoDB 中创建表，并指定表的名称、布局、简单主键以及数据吞吐量设置。

此示例表名为 `Weather`，包含美国几个城市的天气预报信息。此表包含以下类型的信息（在 DynamoDB 中，每一条信息称为一个属性）：

- 必需的唯一城市 ID (`CityID`)
- 必需的预测日期 (`Date`)
- 城市名称 (`City`)
- 州名称 (`State`)
- 预测天气状况 (`Conditions`)
- 预测温度 (`Temperatures`)
 - 预测高温，使用华氏度 (`HighF`)
 - 预测低温，使用华氏度 (`LowF`)

要创建表，请在 AWS Cloud9 IDE 的终端会话中运行 DynamoDB `create-table` 命令。

```
aws dynamodb create-table \
```

```
--table-name Weather \  
--attribute-definitions \  
  AttributeName=CityID,AttributeType=N AttributeName=Date,AttributeType=S \  
--key-schema \  
  AttributeName=CityID,KeyType=HASH AttributeName=Date,KeyType=RANGE \  
--provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

在此命令中：

- `--table-name` 表示表名 (在本示例中为 `Weather`)。表名在您的 AWS 账户中的每个 AWS 区域内必须是唯一的。
- `--attribute-definitions` 表示用于唯一标识表项目的属性。此表的各个项目由数字 ID 属性和 `Date` 属性的组合唯一标识，表示为 ISO-8601 格式的字符串。
- `--key-schema` 表示表的键架构。此表具有复合主键 `CityID` 和 `Date`。这意味着，每个表项目必须具有一个 `CityID` 属性值和一个 `Date` 属性值，但表中的任意两个项目不能具有相同的 `CityID` 属性值和 `Date` 属性值。
- `--provisioned-throughput` 表示表的读写容量。对于最大 4 KB 的项目，DynamoDB 允许每秒最多 5 次强一致性读取，或者对于最大 4 KB 的项目，每秒最多 5 次最终一致读数。对于最大 1 KB 的项目，DynamoDB 还允许每秒最多 5 次写入。

Note

设置较高的预配置吞吐量可能会导致您的 AWS 账户产生额外费用。
有关此命令和其他 DynamoDB 命令的更多信息，请参阅 AWS CLI 命令参考中的 [dynamodb](#)。

如果此命令成功，它会显示所创建新表的摘要信息。要确认已成功创建表，请运行 DynamoDB `describe-table` 命令，在命令中指定表的名称 (`--table-name`)。

```
aws dynamodb describe-table --table-name Weather
```

成功创建表之后，`TableStatus` 值会从 `CREATING` 变为 `ACTIVE`。在成功创建表之前，请不要继续完成此步骤。

步骤 3：向表添加项目

在此步骤中，您将向刚刚创建的表中添加一个项目。

1. 使用以下内容创建名为 `weather-item.json` 的文件。要创建新文件，请在菜单栏上依次选择 File (文件)、New File (新建文件)。要保存文件，请依次选择 File (文件)、Save (保存)。

```
{
  "CityID": { "N": "1" },
  "Date": { "S": "2017-04-12" },
  "City": { "S": "Seattle" },
  "State": { "S": "WA" },
  "Conditions": { "S": "Rain" },
  "Temperatures": { "M": {
    "HighF": { "N": "59" },
    "LowF": { "N": "46" }
  }
}
}
```

在本代码中，N 表示数字属性值。S 是字符串属性值。M 是映射属性，该属性是一组属性/值对。在使用项目时，您必须指定属性的数据类型。有关其他可用属性数据类型的信息，请参阅 Amazon DynamoDB 开发人员指南中的[数据类型](#)。

2. 运行 DynamoDB `put-item` 命令，在命令中指定表的名称 (`--table-name`) 和 JSON 格式项目 (`--item`) 的路径。

```
aws dynamodb put-item \
--table-name Weather \
--item file://weather-item.json
```

如果命令成功，它将运行而没有错误，并且不显示确认消息。

3. 要确认表的当前内容，请运行 DynamoDB `scan` 命令，在命令中指定表的名称 (`--table-name`)。

```
aws dynamodb scan --table-name Weather
```

如果命令成功，则将显示有关表以及刚刚添加项目的汇总信息。

步骤 4：向表添加多个项目

在此步骤中，您将向 Weather 表中添加多个项目。

1. 使用以下内容创建名为 `more-weather-items.json` 的文件。

```
{
  "Weather": [
    {
      "PutRequest": {
        "Item": {
          "CityID": { "N": "1" },
          "Date": { "S": "2017-04-13" },
          "City": { "S": "Seattle" },
          "State": { "S": "WA" },
          "Conditions": { "S": "Rain" },
          "Temperatures": { "M": {
            "HighF": { "N": "52" },
            "LowF": { "N": "43" }
          }
        }
      }
    },
    {
      "PutRequest": {
        "Item": {
          "CityID": { "N": "1" },
          "Date": { "S": "2017-04-14" },
          "City": { "S": "Seattle" },
          "State": { "S": "WA" },
          "Conditions": { "S": "Rain" },
          "Temperatures": { "M": {
            "HighF": { "N": "49" },
            "LowF": { "N": "43" }
          }
        }
      }
    },
    {
      "PutRequest": {
        "Item": {
          "CityID": { "N": "2" },
          "Date": { "S": "2017-04-12" },
          "City": { "S": "Portland" },
          "State": { "S": "OR" },
          "Conditions": { "S": "Thunderstorms" },
          "Temperatures": { "M": {
```

```
        "HighF": { "N": "59" },
        "LowF": { "N": "43" }
      }
    }
  },
  {
    "PutRequest": {
      "Item": {
        "CityID": { "N": "2" },
        "Date": { "S": "2017-04-13" },
        "City": { "S": "Portland" },
        "State": { "S": "OR" },
        "Conditions": { "S": "Rain" },
        "Temperatures": { "M": {
          "HighF": { "N": "51" },
          "LowF": { "N": "41" }
        }
      }
    }
  },
  {
    "PutRequest": {
      "Item": {
        "CityID": { "N": "2" },
        "Date": { "S": "2017-04-14" },
        "City": { "S": "Portland" },
        "State": { "S": "OR" },
        "Conditions": { "S": "Rain Showers" },
        "Temperatures": { "M": {
          "HighF": { "N": "49" },
          "LowF": { "N": "39" }
        }
      }
    }
  },
  {
    "PutRequest": {
      "Item": {
        "CityID": { "N": "3" },
        "Date": { "S": "2017-04-12" },
```

```

    "City": { "S": "Portland" },
    "State": { "S": "ME" },
    "Conditions": { "S": "Rain" },
    "Temperatures": { "M": {
      "HighF": { "N": "59" },
      "LowF": { "N": "40" }
    }
  }
},
{
  "PutRequest": {
    "Item": {
      "CityID": { "N": "3" },
      "Date": { "S": "2017-04-13" },
      "City": { "S": "Portland" },
      "State": { "S": "ME" },
      "Conditions": { "S": "Partly Sunny" },
      "Temperatures": { "M": {
        "HighF": { "N": "54" },
        "LowF": { "N": "37" }
      }
    }
  }
},
{
  "PutRequest": {
    "Item": {
      "CityID": { "N": "3" },
      "Date": { "S": "2017-04-14" },
      "City": { "S": "Portland" },
      "State": { "S": "ME" },
      "Conditions": { "S": "Mostly Sunny" },
      "Temperatures": { "M": {
        "HighF": { "N": "53" },
        "LowF": { "N": "37" }
      }
    }
  }
}
]

```

```
}
```

在本代码中，8 个 Item 对象定义要添加到表中的 8 个项目，类似于上一步中定义的单个项目。不过，在下一步骤中运行 DynamoDB **batch-write-item** 命令时，您必须提供 JSON 格式的对象，该对象在包含的每个 PutRequest 对象中包括一个 Item 对象。然后，您必须将这些 PutRequest 对象包括在与表具有相同名称的父数组中。

2. 运行 DynamoDB **batch-write-item** 命令，在命令中指定要添加的 JSON 格式项目的路径 (`--request-items`)。

```
aws dynamodb batch-write-item \  
--request-items file://more-weather-items.json
```

如果命令成功，它会显示以下消息，确认已成功添加项目。

```
{  
  "UnprocessedItems": {}  
}
```

3. 要确认表的当前内容，请再次运行 DynamoDB **scan** 命令。

```
aws dynamodb scan --table-name Weather
```

如果命令成功，现在显示 9 个项目。

步骤 5：创建全局二级索引

运行 DynamoDB **scan** 命令以获取有关项目的速度可能会较慢，尤其在表的大小增长或者您要获取的信息类型非常复杂时。您可以创建一个或多个二级索引，以加快速度并更轻松地获取信息。在此步骤中，您将了解 DynamoDB 支持用于实现此目的两种二级索引类型。这两种索引称为本地二级索引和全局二级索引。然后创建全局二级索引。

要了解这些二级索引类型，您首先需要了解主键，它用于唯一地标识表的项目。DynamoDB 支持简单主键或复合主键。简单主键具有一个属性，该属性值对于表中的每个项目必须唯一。该属性也称为分区键（或哈希属性），DynamoDB 可以使用该属性来分区项目以实现更快地访问。表还可以具有复合主键，其中包含两个属性。第一个属性是分区键，第二个属性是排序键（也称为范围属性）。在具有复合主键的表中，任何两个项目可以具有相同的分区键值，但不能同时具有相同的排序键值。Weather 表具有复合主键。

本地二级索引具有与表相同的分区键，但此索引类型可以具有不同的排序键。全局二级索引可以具有与表本身均不相同的分区键和排序键。

例如，您可以使用主键，按照 Weather 访问 CityID 项目。要按照 Weather 访问 State，您可以创建具有分区键 CityID (必须与表本身的分区键相同) 以及排序键 State 的本地二级索引。要按照 Weather 访问 City，您可以创建具有分区键 City 以及排序键 Date 的全局二级索引。

您只能在创建表的时候创建本地二级索引。由于 Weather 表已存在，您不能向它添加任何本地二级索引。不过，您可以添加全局二级索引。现在练习添加全局二级索引。

Note

创建二级索引可能会导致您的 AWS 账户产生额外费用。

1. 使用以下内容创建名为 `weather-global-index.json` 的文件。

```
[
  {
    "Create": {
      "IndexName": "weather-global-index",
      "KeySchema": [
        {
          "AttributeName": "City",
          "KeyType": "HASH"
        },
        {
          "AttributeName": "Date",
          "KeyType": "RANGE"
        }
      ],
      "Projection": {
        "ProjectionType": "INCLUDE",
        "NonKeyAttributes": [
          "State",
          "Conditions",
          "Temperatures"
        ]
      },
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 5
      }
    }
  }
]
```

```
    }  
  }  
}  
]
```

在此代码中：

- 全局二级索引的名称为 `weather-global-index`。
- `City` 属性是分区键 (哈希属性)，`Date` 属性是排序键 (范围属性)。
- 对于使用此索引进行的表搜索所匹配的每个项目，`Projection` 定义默认情况下要检索的属性 (在哈希属性和任意范围属性之外)。在本示例中，为每个匹配项目检索 `State`、`Conditions`、`HighF` (属于 `Temperatures`) 和 `LowF` (也属于 `Temperatures`) 属性 (以及 `City` 和 `Date` 属性)。
- 与表类似，全局二级索引必须定义其预配置吞吐量设置。
- `IndexName`、`KeySchema`、`Projection` 和 `ProvisionedThroughput` 设置必须包含在 `Create` 对象中，该对象定义在下一步骤中运行 `DynamoDB update-table` 命令时，要创建的全局二级索引。

2. 运行 `DynamoDB update-table` 命令。

```
aws dynamodb update-table \  
--table-name Weather \  
--attribute-definitions \  
  AttributeName=City,AttributeType=S AttributeName=Date,AttributeType=S \  
--global-secondary-index-updates file://weather-global-index.json
```

在此命令中：

- `--table-name` 是要更新的表的名称。
- `--attribute-definitions` 是要包括在索引中的属性。分区键始终首先列出，任意排序键始终第二列出。
- `--global-secondary-index-updates` 是指向定义全局二级索引的文件的文件的路径。

如果此命令成功，它会显示所创建新全局二级索引的摘要信息。要确认已成功创建全局二级索引，请运行 `DynamoDB describe-table` 命令，在命令中指定表的名称 (`--table-name`)。

```
aws dynamodb describe-table --table-name Weather
```

成功创建了全局二级索引时，TableStatus 值从 UPDATING 更改为 ACTIVE，IndexStatus 值从 CREATING 更改为 ACTIVE。在成功创建全局二级索引之前，请不要继续完成此步骤。这个过程可能需要几分钟。

步骤 6：获取表中的项目

有多种方法可以从表中获取项目。在此步骤中，您可以使用表的主键、使用表的其他属性以及使用全局二级索引来获取项目。

根据项目的主键值从表中获取单个项目

如果您知道某个项目的主键值，则可以通过运行 DynamoDB 命令 **get-item**、**scan** 或 **query** 来获取匹配项目。以下介绍了这些命令中的主要差别：

- **get-item** 返回具有给定主键的项目的一组属性。
- **scan** 通过访问表或者二级索引中的每个项目，返回一个或多个项目和项目属性。
- **query** 基于主键值查找项目。您可查询具有复合主键 (分区键和排序键) 的任何表或二级索引。

在本示例中，下面介绍了如何使用这些各个命令来获取包含 CityID (属性值为 1) 和 Date (属性值为 2017-04-12) 的项目。

1. 要运行 DynamoDB **get-item** 命令，请指定表的名称 (`--table-name`)、主键值 (`--key`) 以及要显示的项目的属性值 (`--projection-expression`)。由于 Date 是 DynamoDB 中的保留关键字，您还必须为 Date 属性值 (`--expression-attribute-names`) 提供一个别名。(State 也是保留关键字，因此您将在后面的步骤中看到为其提供的别名。)

```
aws dynamodb get-item \  
--table-name Weather \  
--key '{ "CityID": { "N": "1" }, "Date": { "S": "2017-04-12" } }' \  
--projection-expression \  
  "City, #D, Conditions, Temperatures.HighF, Temperatures.LowF" \  
--expression-attribute-names '{ "#D": "Date" }'
```

在此命令和其他命令中，要显示项目的所有属性，请勿包括 `--projection-expression`。在本示例中，由于您未包括 `--projection-expression`，您也无需包含 `--expression-attribute-names`。


```
aws dynamodb get-item \  
--table-name Weather \  
--key '{ "CityID": { "N": "1" }, "Date": { "S": "2017-04-12" } }'
```

2. 要运行 DynamoDB **scan** 命令，请指定：

- 表的名称 (--table-name)。
- 要运行的搜索 (--filter-expression)。
- 要使用的搜索标准 (--expression-attribute-values)。
- 要为匹配项目显示的属性类型 (--select)。
- 要显示的项目的属性值 (--projection-expression)。
- 如果您的属性使用 DynamoDB 中的预留关键字，则指定这些属性的别名 (--expression-attribute-names)。

```
aws dynamodb scan \  
--table-name Weather \  
--filter-expression "(CityID = :cityID) and (#D = :date)" \  
--expression-attribute-values \  
  '{ ":cityID": { "N": "1" }, ":date": { "S": "2017-04-12" } }' \  
--select SPECIFIC_ATTRIBUTES \  
--projection-expression \  
  "City, #D, Conditions, Temperatures.HighF, Temperatures.LowF" \  
--expression-attribute-names '{ "#D": "Date" }'
```

3. 要运行 DynamoDB **query** 命令，请指定：

- 表的名称 (--table-name)。
- 要运行的搜索 (--key-condition-expression)。
- 在搜索中使用的属性值 (--expression-attribute-values)。
- 要为匹配项目显示的属性类型 (--select)。
- 要显示的项目的属性值 (--projection-expression)。
- 如果您的属性使用 DynamoDB 中的预留关键字，则指定这些属性的别名 (--expression-attribute-names)。

```
aws dynamodb query \  
--table-name Weather \  
--key-condition-expression "(CityID = :cityID) and (#D = :date)" \  
--expression-attribute-values \  
  '{ ":cityID": { "N": "1" }, ":date": { "S": "2017-04-12" } }'
```

```
{ ":cityID": { "N": "1" }, ":date": { "S": "2017-04-12" } }' \
--select SPECIFIC_ATTRIBUTES \
--projection-expression \
"City, #D, Conditions, Temperatures.HighF, Temperatures.LowF" \
--expression-attribute-names '{ "#D": "Date" }'
```

请注意，**scan** 命令需要扫描所有 9 个项目以获得所需的结果，而 **query** 命令仅扫描 1 个项目。

根据项目的主键值从表中获取多个项目

如果您知道项目的主键值，则可以通过运行 DynamoDB **batch-get-item** 命令获取匹配项目。在本示例中，下面介绍了如何获取包含 CityID (属性值为 3) 和 Date (属性值为 2017-04-13 或 2017-04-14) 的项目。

运行 DynamoDB **batch-get-item** 命令，指定描述要获取项目的文件的路径 (`--request-items`)。

```
aws dynamodb batch-get-item --request-items file://batch-get-item.json
```

在本示例中，`batch-get-item.json` 文件中的代码指定在 Weather 表中，搜索 CityID 为 3 并且 Date 为 2017-04-13 或 2017-04-14 的项目。对于找到的每个项目，将显示 City、State、Date 和 HighF (属于 Temperatures) 的属性值 (如果存在)。

```
{
  "Weather" : {
    "Keys": [
      {
        "CityID": { "N": "3" },
        "Date": { "S": "2017-04-13" }
      },
      {
        "CityID": { "N": "3" },
        "Date": { "S": "2017-04-14" }
      }
    ],
    "ProjectionExpression": "City, #S, #D, Temperatures.HighF",
    "ExpressionAttributeNames": { "#S": "State", "#D": "Date" }
  }
}
```

获取表的所有匹配项目

如果您知道表中属性值的一些信息，则可以通过运行 DynamoDB **scan** 命令获取匹配项目。在本示例中，下面介绍了当 Conditions 属性值包含 Sunny 并且 HighF 属性值 (属于 Temperatures) 大于 53 时，如何获取日期。

运行 DynamoDB **scan** 命令，指定：

- 表的名称 (--table-name)。
- 要运行的搜索 (--filter-expression)。
- 要使用的搜索标准 (--expression-attribute-values)。
- 要为匹配项目显示的属性类型 (--select)。
- 要显示的项目的属性值 (--projection-expression)。
- 如果您的属性使用 DynamoDB 中的预留关键字，则指定这些属性的别名 (--expression-attribute-names)。

```
aws dynamodb scan \  
--table-name Weather \  
--filter-expression \  
  "(contains (Conditions, :sun)) and (Temperatures.HighF > :h)" \  
--expression-attribute-values \  
  '{ ":sun": { "S" : "Sunny" }, ":h": { "N" : "53" } }' \  
--select SPECIFIC_ATTRIBUTES \  
--projection-expression "City, #S, #D, Conditions, Temperatures.HighF" \  
--expression-attribute-names '{ "#S": "State", "#D": "Date" }'
```

从全局二级索引获取所有匹配项目

要使用全局二级索引搜索，请使用 DynamoDB **query** 命令。在本示例中，下面介绍了如何使用 weather-global-index 二级索引，获取名为 Portland 的城市在日期 2017-04-13 和 2017-04-14 的天气预报。

运行 DynamoDB **query** 命令，指定：

- 表的名称 (--table-name)。
- 全局二级索引的名称 (--index-name)。
- 要运行的搜索 (--key-condition-expression)。
- 在搜索中使用的属性值 (--expression-attribute-values)。

- 要为匹配项目显示的属性类型 (--select)。
- 如果您的属性使用 DynamoDB 中的预留关键字，则指定这些属性的别名 (--expression-attribute-names)。

```
aws dynamodb query \  
--table-name Weather \  
--index-name weather-global-index \  
--key-condition-expression "(City = :city) and (#D between :date1 and :date2)" \  
--expression-attribute-values \  
  '{ ":city": { "S" : "Portland" }, ":date1": { "S": "2017-04-13" }, ":date2": { "S":  
  "2017-04-14" } }' \  
--select SPECIFIC_ATTRIBUTES \  
--projection-expression "City, #S, #D, Conditions, Temperatures.HighF" \  
--expression-attribute-names '{ "#S": "State", "#D": "Date" }'
```

步骤 7：清除

为防止在您使用完该示例后一直对您的 AWS 账户收费，应删除表。删除表还将删除全局二级索引。您还应该删除环境。

要删除表，请运行 DynamoDB **delete-table** 命令，在命令中指定表的名称 (--table-name)。

```
aws dynamodb delete-table --table-name Weather
```

如果命令成功，则会显示有关表的信息，包括值为 TableStatus 的 DELETING。

要确认已成功删除表，请运行 DynamoDB **describe-table** 命令，在命令中指定表的名称 (--table-name)。

```
aws dynamodb describe-table --table-name Weather
```

如果成功删除此表，将显示包含短语 Requested resource not found 的消息。

要删除您的环境，请参阅[删除环境](#)。

适用于 AWS Cloud9 的 AWS CDK 教程

此教程演示如何在 AWS Cloud9 开发环境中使用 AWS Cloud Development Kit (AWS CDK)。AWS CDK 是一组软件工具和库，开发人员可以使用它将 AWS 基础设施组件建模为代码。

AWS CDK 包括您可用于在 AWS 上快速完成多个任务的 AWS 构造库。例如，您可以使用 Fleet 构造将代码完全且安全地部署到一队主机。您可以创建自己的构造为架构的各种元素建模，与其他人共享这些构造或将其发布到社区。有关更多信息，请参阅 [AWS Cloud Development Kit 开发人员指南](#)。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2、Amazon SNS 和 Amazon SQS 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#)、[Amazon SNS 定价](#) 和 [Amazon SQS 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [步骤 4：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，您在环境中安装 AWS CDK 运行以 TypeScript 编程语言编写的示例所需的所有工具。

1. [节点版本管理器](#)（或 `nvm`），用于稍后安装 Node.js。
2. [Node.js](#)，示例所需并包含节点程序包管理器（即 `npm`），用于稍后安装 TypeScript 和 AWS CDK。
3. [TypeScript](#)，此示例所需。（AWS CDK 还支持多种其他编程语言。）
4. 这些区域有：[AWS CDK](#)。

步骤 1.1：安装节点版本管理器 (nvm)

1. 在 AWS Cloud9 IDE 的终端会话中，确保已安装最新的安全更新和错误修复。要执行此操作，请运行 **yum update**（适用于 Amazon Linux）或 **apt update** 命令（适用于 Ubuntu Server）。（要开始新的终端会话，请在菜单栏上依次选择 Window（窗口）> New Terminal（新建终端）。）

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu Server：

```
sudo apt update
```

2. 确认是否已安装 **nvm**。为此，请运行带 **--version** 选项的 **nvm** 命令。

```
nvm --version
```

如果成功，则输出包含 **nvm** 版本号，并且您可以向前跳至 [步骤 1.2：安装 Node.js](#)。

3. 下载并安装 **nvm**。要执行此操作，请运行安装脚本。在此示例中，安装了 v0.33.0，但您可以在[此处](#)检查 **nvm** 的最新版本。

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

4. 开始使用 **nvm**。您可用关闭终端会话并重新启动，或者查找包含命令以加载 **nvm** 的 `~/.bashrc` 文件。

```
. ~/.bashrc
```

步骤 1.2：安装 Node.js

1. 确认是否已安装 Node.js，如果已安装，请确认安装的版本为 16.17.0 或更高版本。此示例使用 Node.js 16.17.0 进行测试。要检查结果，请在终端会话仍在 IDE 中打开的情况下，运行带有 **--version** 选项的 **node** 命令。

```
node --version
```

如果已安装 Node.js，则输出将包含版本号。如果该版本号是 v16.17.0，请向前跳至 [步骤 1.3：安装 TypeScript](#)。

2. 安装 Node.js 16，方法是运行带有 **install** 操作的 **nvm** 命令。

Note

您还可以运行 **nvm install node** 来安装长期支持 (LTS) 版本的 Node.js。AWS Cloud9 支持跟踪 LTS 版本的 Node.js。

```
nvm install v16
```

3. 开始使用 Node.js 16。为此，请运行 **nvm** 命令与 **alias** 操作、别名的版本号以及要用于该别名的版本，如下所示。

```
nvm alias default 16
```

Note

上述命令将 Node.js 16 设置为 Node.js 的默认版本。或者，您可以运行带有 **use** 操作（如 **nvm use 16.17.0**）而非 **alias** 操作的 **nvm** 命令。不过，**use** 操作会导致该版本的 Node.js 仅在当前终端会话运行时运行。

4. 要确认您使用的是 Node.js 16，请再次运行 **node --version** 命令。如果已安装正确的版本，则输出将包含版本 v16。

步骤 1.3：安装 TypeScript

1. 确认是否已安装 TypeScript。为此，请在终端会话仍在 IDE 中打开的情况下，带 **--version** 选项运行命令行 TypeScript 编译器。

```
tsc --version
```

如果已安装 TypeScript，则输出将包含 TypeScript 版本号。如果已安装 TypeScript，请向前跳至 [步骤 1.4：安装 AWS CDK](#)。

2. 安装 TypeScript。为此，请运行带有 **install** 操作、**-g** 选项以及 TypeScript 包名称的 **npm** 命令。这会在环境中作为全局程序包安装 TypeScript。

```
npm install -g typescript
```

3. 确认已安装 TypeScript。为此，请带 **--version** 选项运行命令行 TypeScript 编译器。

```
tsc --version
```

如果已安装 TypeScript，则输出将包含 TypeScript 版本号。

步骤 1.4：安装 AWS CDK

1. 确认是否已安装 AWS CDK。为此，请在终端会话仍在 IDE 中打开的情况下，运行带有 **--version** 选项的 **cdk** 命令。

```
cdk --version
```

如果已安装 AWS CDK，则输出将包含 AWS CDK 版本号和内部版本号。向前跳至 [步骤 2：添加代码](#)。

2. 安装 AWS CDK，方法是通过运行带有 **install** 操作、AWS CDK 软件包名称以及 **-g** 选项的 **npm** 命令以环境中全局安装软件包。

```
npm install -g aws-cdk
```

3. 确认 AWS CDK 已安装并已正确引用。为此，请运行带 **--version** 选项的 **cdk** 命令。

```
cdk --version
```

如果成功，将显示 AWS CDK 版本号和内部版本号。

步骤 2：添加代码

在此步骤中，您将创建一个示例 TypeScript 项目，其中包含编程部署 AWS CloudFormation 堆栈的 AWS CDK 所需的所有源代码。此堆栈在您的 AWS 账户中创建 Amazon SNS 主题和 Amazon SQS 队列，然后为队列订阅该主题。

1. 在终端会话仍在 IDE 中打开的情况下，创建存储项目源代码的目录，例如您环境中的 `~/environment/hello-cdk` 目录。然后切换到该目录。

```
rm -rf ~/environment/hello-cdk # Remove this directory if it already exists.
mkdir ~/environment/hello-cdk # Create the directory.
cd ~/environment/hello-cdk # Switch to the directory.
```

2. 将该目录设置为 AWS CDK 的 TypeScript 语言项目。为此，请运行带有 `init` 操作、`sample-app` 模板和 `--language` 选项以及编程语言名称的 `cdk` 命令。

```
cdk init sample-app --language typescript
```

这将在该目录中创建以下文件和子目录。

- 隐藏的 `.git` 子目录和隐藏的 `.gitignore` 文件，可使项目与 Git 等源控制工具兼容。
 - `lib` 子目录，其中包含 `hello-cdk-stack.ts` 文件。此文件包含 AWS CDK 堆栈的代码。此代码将在此过程的下一步骤中介绍。
 - `bin` 子目录，其中包含 `hello-cdk.ts` 文件。此文件包含 AWS CDK 应用程序的入口点。
 - `node_modules` 子目录，其中包含应用程序和堆栈可根据需要使用的支持代码包。
 - 隐藏的 `.npmignore` 文件，其中列出了 `npm` 生成代码时不需要的子目录和文件类型。
 - `cdk.json` 文件，其中包含使 `cdk` 命令运行更轻松的信息。
 - `package-lock.json` 文件，其中包含 `npm` 可用于减少可能的生成和运行错误的信息。
 - `package.json` 文件，其中包含使 `npm` 命令运行更轻松以及可能减少构建和运行错误的信息。
 - `README.md` 文件，其中列出了可以与 `npm` 和 AWS CDK 一起运行的有用命令。
 - `tsconfig.json` 文件，其中包含使 `tsc` 命令运行更轻松以及可能减少构建和运行错误的信息。
3. 在 Environment (环境) 窗口中，打开 `lib/hello-cdk-stack.ts` 文件，并浏览该文件中的以下代码。

```
import sns = require('@aws-cdk/aws-sns');
import sqs = require('@aws-cdk/aws-sqs');
import cdk = require('@aws-cdk/cdk');

export class HelloCdkStack extends cdk.Stack {
  constructor(parent: cdk.App, name: string, props?: cdk.StackProps) {
    super(parent, name, props);
  }
}
```

```
const queue = new sqs.Queue(this, 'HelloCdkQueue', {
  visibilityTimeoutSec: 300
});

const topic = new sns.Topic(this, 'HelloCdkTopic');

topic.subscribeQueue(queue);
}
}
```

- Stack、App、StackProps、Queue 和 Topic 类分别表示 AWS CloudFormation 堆栈及其属性、可执行程序、Amazon SQS 队列和 Amazon SNS 主题。
 - HelloCdkStack 类表示此应用程序的 AWS CloudFormation 堆栈。此堆栈包含用于此应用程序的新 Amazon SQS 队列和 Amazon SNS 主题。
4. 在 Environment (环境) 窗口中，打开 bin/hello-cdk.ts 文件，并浏览该文件中的以下代码。

```
#!/usr/bin/env node
import cdk = require('@aws-cdk/cdk');
import { HelloCdkStack } from '../lib/hello-cdk-stack';

const app = new cdk.App();
new HelloCdkStack(app, 'HelloCdkStack');
app.run();
```

此代码从 HelloCdkStack 文件中加载、实例化然后运行 lib/hello-cdk-stack.ts 类。

5. 使用 **npm** 运行 TypeScript 编译器以检查编码错误，然后启用 AWS CDK 以执行项目的 bin/hello-cdk.js 文件。为此，请从项目的根目录中运行带有 **run** 操作的 **npm** 命令，指定 package.json 文件中的 **build** 命令值，如下所示。

```
npm run build
```

前面的命令运行 TypeScript 编译器，其中增加了支持 bin/hello-cdk.d.ts 和 lib/hello-cdk-stack.d.ts 文件。该编译器还可将 hello-cdk.ts 和 hello-cdk-stack.ts 文件转换为 hello-cdk.js 和 hello-cdk-stack.js 文件。

步骤 3：运行代码

在此步骤中，您指示 AWS CDK 基于 `bin/hello-cdk.js` 文件中的代码创建 AWS CloudFormation 堆栈模板。然后，您指示 AWS CDK 部署堆栈，从而创建 Amazon SNS 主题和 Amazon SQS 队列，并为队列订阅该主题。然后，通过从主题向队列发送消息，确认已成功部署主题和队列。

1. 使用 AWS CDK 创建 AWS CloudFormation 堆栈模板。为此，在终端会话仍在 IDE 中打开的情况下，从项目的根目录，运行带有 **synth** 操作和堆栈名称的 **cdk** 命令。

```
cdk synth HelloCdkStack
```

如果成功，则输出将显示 AWS CloudFormation 堆栈模板的 `Resources` 部分。

2. 首次将 AWS CDK 应用程序部署到特定 AWS 账户和 AWS 区域的组合环境中时，必须安装引导堆栈。该堆栈包括 AWS CDK 完成各种操作所需要的资源。例如，此堆栈包含一个 Amazon S3 存储桶，AWS CDK 在部署过程中使用该存储桶来存储模板和资产。要安装引导堆栈，请运行带有 **bootstrap** 操作的 **cdk** 命令。

```
cdk bootstrap
```

Note

如果您运行 `cdk bootstrap` 而不指定任何选项，则使用默认 AWS 账户和 AWS 区域。您还可以通过指定配置文件和账户/区域组合来引导特定环境。例如：

```
cdk bootstrap --profile test 123456789012/us-east-1
```

3. 使用 AWS CDK 运行 AWS CloudFormation 堆栈模板以部署堆栈。为此，从项目的根目录，运行带有 **deploy** 操作和堆栈名称的 **cdk** 命令。

```
cdk deploy HelloCdkStack
```

如果成功，则输出将显示 `HelloCdkStack` 堆栈已部署且没有错误。

Note

如果输出显示的消息表明堆栈未定义环境且无法从标准位置获取 AWS 凭证或未配置任何区域，请确保在 IDE 中正确设置了 AWS 凭证，然后再次运行 **cdk deploy** 命令。有关更多信息，请参阅[从 AWS Cloud9 中的环境调用 AWS 服务](#)。

4. 要确认 Amazon SNS 主题和 Amazon SQS 队列已成功部署，请发送消息到该主题，然后检查队列是否收到消息。为此，您可以使用 AWS Command Line Interface (AWS CLI) 或 AWS CloudShell 等工具。有关这些工具的更多信息，请参阅[适用于 AWS Cloud9 的 AWS Command Line Interface](#) 和 [aws-shell 教程](#)。

例如，要向该主题发送消息，请在终端会话仍在 IDE 中打开的情况下，使用 AWS CLI 运行 Amazon SNS **publish** 命令，提供消息的主题和正文、主题的 AWS 区域以及主题的 Amazon Resource Name (ARN)。

```
aws sns publish --subject "Hello from the AWS CDK" --message "This is a message from the AWS CDK." --topic-arn arn:aws:sns:us-east-2:123456789012:HelloCdkStack-HelloCdkTopic1A234567-8BCD9EFGHIJ0K
```

在上述命令中，将 `arn:aws:sns:us-east-2:123456789012:HelloCdkStack-HelloCdkTopic1A234567-8BCD9EFGHIJ0K` 替换为 AWS CloudFormation 分配给该主题的 ARN。为获取该 ID，您可以运行 Amazon SNS **list-topics** 命令。

```
aws sns list-topics --output table --query 'Topics[*].TopicArn'
```

如果成功，则 **publish** 命令的输出将显示所发布消息的 MessageId 值。

要检查队列是否收到消息，请运行 Amazon SQS **receive-message** 命令，提供队列的 URL。

```
aws sqs receive-message --queue-url https://queue.amazonaws.com/123456789012/HelloCdkStack-HelloCdkQueue1A234567-8BCD9EFGHIJ0K
```

在上述命令中，将 `https://queue.amazonaws.com/123456789012/HelloCdkStack-HelloCdkQueue1A234567-8BCD9EFGHIJ0K` 替换为 AWS CloudFormation 分配给该队列的 ARN。为获取该 URL，您可以运行 Amazon SQS **list-queues** 命令。

```
aws sqs list-queues --output table --query 'QueueUrls[*]'
```

如果成功，则 `receive-message` 命令的输出将显示收到的消息的相关信息。

步骤 4：清除

为防止在您使用完该示例后一直对您的 AWS 账户收费，应删除 AWS CloudFormation 堆栈。此步骤将删除 Amazon SNS 主题和 Amazon SQS 队列。您还应该删除环境。

步骤 4.1：删除堆栈

在终端会话仍在 IDE 中打开的情况下，从项目的根目录，运行 `cdk` 命令及 `destroy` 操作和堆栈名称。

```
cdk destroy HelloCdkStack
```

当系统提示删除堆栈时，请键入 `y`，然后按 `Enter` 键。

如果成功，则输出将显示 `HelloCdkStack` 堆栈已删除且没有错误。

步骤 4.2：删除环境

要删除环境，请参阅 [在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 LAMP 教程

此教程可让您在 AWS Cloud9 开发环境内设置和运行 LAMP (Linux、Apache HTTP Server、MySQL 和 PHP)。

按照本教程操作并创建此示例可能会对您的 AWS 账户 收费。这些包括可能对 Amazon Elastic Compute Cloud (Amazon EC2) 等 AWS 服务 收取费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装工具](#)
- [步骤 2：设置 MySQL](#)
- [步骤 3：设置网站](#)

- [步骤 4：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装工具

在本步骤中，您将安装以下工具：

- Apache HTTP Server，Web 服务器主机。
- PHP，一种脚本语言，特别适用于 Web 开发并可嵌入到 HTML 中。
- MySQL，数据库管理系统。

然后，您可以通过按顺序启动 Apache HTTP Server 和 MySQL 完成此步骤。

1. 确保已在实例上安装了最新的安全更新和错误修复。要执行此操作，在 AWS Cloud9 IDE 的终端会话中，运行 **yum update**（适用于 Amazon Linux）或 **apt update**（适用于 Ubuntu Server）命令。（要开始新的终端会话，请在菜单栏上依次选择 Window（窗口）> New Terminal（新建终端）。）

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu 服务器：

```
sudo apt -y update
```

2. 检查是否已安装 Apache HTTP Server。要执行此操作，请运行 **httpd -v**（适用于 Amazon Linux）或 **apache2 -v**（适用于 Ubuntu Server）命令。

如果成功，输出会包含 Apache HTTP Server 版本号。

如果您看到错误，则通过运行 **install** 命令安装 Apache HTTP Server。

对于 Amazon Linux：

```
sudo yum install -y httpd24
```

对于 Ubuntu Server：

```
sudo apt install -y apache2
```

3. 通过运行 **php -v** 命令确认是否已安装 PHP。

如果成功，输出将包含 PHP 版本号。

如果您看到错误，则通过运行 **install** 命令安装 PHP。

对于 Amazon Linux：

```
sudo yum install -y php56
```

对于 Ubuntu Server：

```
sudo apt install -y php libapache2-mod-php php-xml
```

4. 通过运行 **mysql --version** 命令确认是否已安装 MySQL。

如果成功，输出将包含 MySQL 版本号。

如果您看到错误，则通过运行 **install** 命令安装 MySQL。

对于 Amazon Linux：

```
sudo yum install -y mysql-server
```

对于 Ubuntu 服务器：

```
sudo apt install -y mysql-server
```

5. 安装 Apache HTTP Server、PHP 和 MySQL 后，启动 Apache HTTP Server，然后通过运行以下命令确认它已启动。

对于 Amazon Linux (可能需要运行此命令两次) :

```
sudo service httpd start && sudo service httpd status
```

对于 Ubuntu Server (要返回到命令提示符，按 q) :

```
sudo service apache2 start && sudo service apache2 status
```

6. 启动 MySQL，然后通过运行以下命令确认它已启动。

对于 Amazon Linux :

```
sudo service mysqld start && sudo service mysqld status
```

对于 Ubuntu Server (要返回到命令提示符，按 q) :

```
sudo service mysql start && sudo service mysql status
```

步骤 2：设置 MySQL

在此步骤中，您设置 MySQL 以遵循 MySQL 安全最佳实践。这些安全最佳实践包括为根账户设置密码和删除可从本地主机外部访问的根账户。其他需要注意的最佳实践是删除匿名用户的权限，删除测试数据库，以及删除允许任何人访问名称以 `test_` 开头的数据库的权限。

然后，在此步骤最后练习启动 MySQL 命令行客户端，然后退出。

1. 通过在 AWS Cloud9 IDE 中的终端会话中运行以下命令来实施 MySQL 安装的 MySQL 安全最佳实践。

```
sudo mysql_secure_installation
```

2. 在提示时，按照说明回答以下问题。

对于 Amazon Linux :

1. 输入当前的根密码 (如无密码，按 Enter) – 按 Enter (如无密码)。

2. 设置根密码 – 键入 Y，然后按 Enter。
3. 新建密码 – 键入密码，然后按 Enter。
4. 重新输入新密码 – 再次键入密码，然后按 Enter。（请务必将密码存储在安全位置以便以后使用。）
5. 删除匿名用户 – 键入 Y，然后按 Enter。
6. 禁止远程根登录 – 键入 Y，然后按 Enter。
7. 删除测试数据库以及对它的访问 – 键入 Y，然后按 Enter。
8. 立即重新加载权限表 – 键入 Y，然后按 Enter。

对于 Ubuntu 服务器：

1. 是否要设置“验证密码”插件 – 输入 y，然后按 Enter。
 2. 有三个密码验证策略级别 – 输入 0、1 或 2，然后按 Enter。
 3. 新建密码 – 输入密码，然后按 Enter。
 4. 重新输入新密码 – 再次输入密码，然后按 Enter。务必将密码存储在安全位置以便以后使用。
 5. 是否希望继续使用提供的密码 – 输入 y，然后按 Enter。
 6. 删除匿名用户 – 输入 y，然后按 Enter。
 7. 禁止远程根登录 – 输入 y，然后按 Enter。
 8. 删除测试数据库以及对它的访问 – 输入 y，然后按 Enter。
 9. 立即重新加载权限表 – 输入 y，然后按 Enter。
3. 要与 MySQL 直接进行交互，请以根用户身份，通过运行以下命令启动 MySQL 命令行客户端。在提示时，键入您之前设置的根用户的密码，然后按 Enter。当您进入 MySQL 命令行客户端时，提示符将更改为 `mysql>`。

```
sudo mysql -uroot -p
```

4. 要退出 MySQL 命令行客户端，请运行以下命令。提示符变回 `$`。

```
exit;
```

步骤 3：设置网站

在此步骤中，您使用建议的拥有者和访问权限为 Apache HTTP Server 设置原定设置网站根目录。然后，在该原定设置的网站根目录内创建基于 PHP 的网页。

然后，启用传入 Web 流量来查看该网页，方法是通过在 Amazon EC2 中设置安全组，并在与此 EC2 环境关联的 Amazon Virtual Private Cloud (Amazon VPC) 中设置网络访问控制列表（网络 ACL）。每个 EC2 环境都必须与 Amazon EC2 中的安全组和 Amazon VPC 中的网络 ACL 关联。但是，即使 AWS 账户中的原定设置网络 ACL 允许环境的所有传入和传出流量，原定设置安全组也仅允许端口 22 上使用 SSH 的传入流量。有关更多信息，请参阅[the section called “Amazon VPC 设置”](#)。

然后，在此步骤最后从 AWS Cloud9 IDE 外部成功查看网页。

1. 使用建议的拥有者和访问权限为 Apache HTTP Server 设置原定设置网站根目录（`/var/www/html`）。要执行此操作，请在 AWS Cloud9 IDE 的终端会话中运行以下六个命令，按照以下顺序每次运行一个命令。要了解每个命令的作用，请阅读每个命令后面 # 字符后的信息。

对于 Amazon Linux：

```
sudo groupadd web-content # Create a group named web-content.

sudo usermod -G web-content -a ec2-user # Add the user ec2-user (your default user
for this environment) to the group web-content.

sudo usermod -G web-content -a apache # Add the user apache (Apache HTTP Server) to
the group web-content.

sudo chown -R ec2-user:web-content /var/www/html # Change the owner of /var/www/
html and its files to user ec2-user and group web-content.

sudo find /var/www/html -type f -exec chmod u=rw,g=rx,o=rx {} \; # Change all file
permissions within /var/www/html to user read/write, group read-only, and others
read/execute.

sudo find /var/www/html -type d -exec chmod u=rwx,g=rx,o=rx {} \; # Change /var/
www/html directory permissions to user read/write/execute, group read/execute, and
others read/execute.
```

对于 Ubuntu 服务器：

```
sudo groupadd web-content # Create a group named web-content.
```

```
sudo usermod -G web-content -a ubuntu # Add the user ubuntu (your default user for
this environment) to the group web-content.

sudo usermod -G web-content -a www-data # Add the user www-data (Apache HTTP
Server) to the group web-content.

sudo chown -R ubuntu:web-content /var/www/html # Change the owner of /var/www/html
and its files to user ubuntu and group web-content.

sudo find /var/www/html -type f -exec chmod u=rw,g=rx,o=rx {} \; # Change all file
permissions within /var/www/html to user read/write, group read-only, and others
read/execute.

sudo find /var/www/html -type d -exec chmod u=rwx,g=rx,o=rx {} \; # Change /var/
www/html directory permissions to user read/write/execute, group read/execute, and
others read/execute.
```

2. 通过运行以下命令，在 Apache HTTP Server 的原定设置网站根文件夹（即 `/var/www/html`）中创建名为 `index.php` 的基于 PHP 的网页。

对于 Amazon Linux：

```
sudo touch /var/www/html/index.php && sudo chown -R ec2-user:web-content /var/www/
html/index.php && sudo chmod u=rw,g=rx,o=rx /var/www/html/index.php && sudo printf
'%s\n%s\n%s' '<?php' ' phpinfo();' '?>' >> /var/www/html/index.php
```

Amazon Linux 的上述命令还会将文件的拥有者更改为 `ec2-user`，将文件的组更改为 `web-content`，并更改用户的文件读取/写入权限以及组等的读取/执行权限。

对于 Ubuntu 服务器：

```
sudo touch /var/www/html/index.php && sudo chown -R ubuntu:web-content /var/www/
html/index.php && sudo chmod u=rw,g=rx,o=rx /var/www/html/index.php && sudo printf
'%s\n%s\n%s' '<?php' ' phpinfo();' '?>' >> /var/www/html/index.php
```

Ubuntu Server 的上述命令还会将文件的拥有者更改为 `ubuntu`，将文件的组更改为 `web-content`，并更改用户的文件读取/写入权限以及组等的读取/执行权限。

如果成功，上述命令将使用以下内容创建 `index.php` 文件。

```
<?php
  phpinfo();
?>
```

3. 启用端口 80 上的传入 Web 流量以查看新网页，方法是通过在 Amazon VPC 和与此 EC2 环境关联的安全组 Amazon EC2 中设置网络 ACL。要执行此操作，按照以下顺序运行下面的八个命令，一次运行一个命令。要了解每个命令的作用，请阅读每个命令的 # 字符后的信息。

Important

运行以下命令将为所有 EC2 环境以及与此环境的安全组和网络 ACL 关联的 Amazon EC2 实例启用端口 80 上的传入 Web 流量。这可能导致意外为其他的 EC2 环境和 Amazon EC2 实例启用端口 80 上的传入 Web 流量。

Note

下面的第二到第四个命令启用安全组来允许端口 80 上的传入 Web 流量。如果您有默认安全组，其仅允许端口 22 上的传入 SSH 流量，则必须在运行第二到第四个命令后运行第一个命令。但是，如果您有已经允许端口 80 上的传入 Web 流量的自定义安全组，则可以跳过运行这些命令。

下面的第五到第八个命令会启用网络 ACL 来允许端口 80 上的传入 Web 流量。如果您有默认网络 ACL，其允许所有端口的所有传入流量，则可以安全地跳过运行这些命令。但是，假设您有一个自定义网络 ACL，它不允许通过端口 80 传入 Web 流量。然后，运行第一个命令，然后运行第五到第八个命令。

```
MY_INSTANCE_ID=$(curl http://169.254.169.254/latest/meta-data/instance-id) # Get
the ID of the instance for the environment, and store it temporarily.

MY_SECURITY_GROUP_ID=$(aws ec2 describe-instances --instance-id $MY_INSTANCE_ID
--query 'Reservations[].Instances[0].SecurityGroups[0].GroupId' --output text)
# Get the ID of the security group associated with the instance, and store it
temporarily.

aws ec2 authorize-security-group-ingress --group-id $MY_SECURITY_GROUP_ID --
protocol tcp --cidr 0.0.0.0/0 --port 80 # Add an inbound rule to the security group
to allow all incoming IPv4-based traffic over port 80.
```

```
aws ec2 authorize-security-group-ingress --group-id $MY_SECURITY_GROUP_ID --ip-
permissions IpProtocol=tcp,Ipv6Ranges='[CidrIpv6=::/0]',FromPort=80,ToPort=80 #
Add an inbound rule to the security group to allow all incoming IPv6-based traffic
over port 80.
```

```
MY_SUBNET_ID=$(aws ec2 describe-instances --instance-id $MY_INSTANCE_ID --query
'Reservations[].Instances[0].SubnetId' --output text) # Get the ID of the subnet
associated with the instance, and store it temporarily.
```

```
MY_NETWORK_ACL_ID=$(aws ec2 describe-network-acls --filters
Name=association.subnet-id,Values=$MY_SUBNET_ID --query
'NetworkAcls[].Associations[0].NetworkACLId' --output text) # Get the ID of the
network ACL associated with the subnet, and store it temporarily.
```

```
aws ec2 create-network-acl-entry --network-acl-id $MY_NETWORK_ACL_ID --ingress --
protocol tcp --rule-action allow --rule-number 10000 --cidr-block 0.0.0.0/0 --port-
range From=80,To=80 # Add an inbound rule to the network ACL to allow all IPv4-
based traffic over port 80. Advanced users: change this suggested rule number as
desired.
```

```
aws ec2 create-network-acl-entry --network-acl-id $MY_NETWORK_ACL_ID --ingress --
protocol tcp --rule-action allow --rule-number 10100 --ipv6-cidr-block ::/0 --port-
range From=80,To=80 # Add an inbound rule to the network ACL to allow all IPv6-
based traffic over port 80. Advanced users: change this suggested rule number as
desired.
```

4. 获取 Web 服务器根目录内的 `index.php` 文件的 URL。要执行此操作，请运行以下命令，并使用新的 Web 浏览器标签页或与 AWS Cloud9 IDE 无关的其他 Web 浏览器转到显示的 URL。如果成功，网页将显示有关 Apache HTTP Server、MySQL、PHP 及其他相关设置的信息。

```
MY_PUBLIC_IP=$(curl http://169.254.169.254/latest/meta-data/public-ipv4) && echo
http://$MY_PUBLIC_IP/index.php # Get the URL to the index.php file within the web
server root.
```

步骤 4：清除

假设您希望继续使用此环境，但希望禁用端口 80 上的传入 Web 流量。那么，按照以下顺序运行下面的八个命令（一次运行一个），以删除您之前在与环境关联的安全组和网络 ACL 中设置的对应的传入流量规则。要了解每个命令的作用，请阅读每个命令的 # 字符后的信息。

⚠ Important

运行以下命令将为所有 EC2 环境以及与此环境的安全组和网络 ACL 关联的 Amazon EC2 实例禁用端口 80 上的传入 Web 流量。这可能导致意外为其他的 EC2 环境和 Amazon EC2 实例禁用端口 80 上的传入 Web 流量。

📌 Note

下面的第五到第八个命令删除现有规则，以阻止网络 ACL 允许端口 80 上的传入 Web 流量。如果您有原定设置网络 ACL，其允许所有端口的所有传入流量，则可以跳过运行这些命令。但是，假设您有一个自定义网络 ACL，其现有规则允许通过端口 80 传入 Web 流量，而您想要删除这些规则。那么，您需要运行第一个命令，然后运行第五到第八个命令。

```
MY_INSTANCE_ID=$(curl http://169.254.169.254/latest/meta-data/instance-id) # Get the ID
of the instance for the environment, and store it temporarily.

MY_SECURITY_GROUP_ID=$(aws ec2 describe-instances --instance-id $MY_INSTANCE_ID --query
'Reservations[].Instances[0].SecurityGroups[0].GroupId' --output text) # Get the ID of
the security group associated with the instance, and store it temporarily.

aws ec2 revoke-security-group-ingress --group-id $MY_SECURITY_GROUP_ID --protocol tcp
--cidr 0.0.0.0/0 --port 80 # Delete the existing inbound rule from the security group
to block all incoming IPv4-based traffic over port 80.

aws ec2 revoke-security-group-ingress --group-id $MY_SECURITY_GROUP_ID --ip-permissions
IpProtocol=tcp,Ipv6Ranges='[{"CidrIpv6=":::/0}]',FromPort=80,ToPort=80 # Delete the
existing inbound rule from the security group to block all incoming IPv6-based traffic
over port 80.

MY_SUBNET_ID=$(aws ec2 describe-instances --instance-id $MY_INSTANCE_ID --query
'Reservations[].Instances[0].SubnetId' --output text) # Get the ID of the subnet
associated with the instance, and store it temporarily.

MY_NETWORK_ACL_ID=$(aws ec2 describe-network-acls --filters Name=association.subnet-
id,Values=$MY_SUBNET_ID --query 'NetworkAcls[0].Associations[0].NetworkAclId' --output
text) # Get the ID of the network ACL associated with the subnet, and store it
temporarily.
```

```
aws ec2 delete-network-acl-entry --network-acl-id $MY_NETWORK_ACL_ID --ingress --rule-number 10000 # Delete the existing inbound rule from the network ACL to block all IPv4-based traffic over port 80. Advanced users: if you originally created this rule with a different number, change this suggested rule number to match.
```

```
aws ec2 delete-network-acl-entry --network-acl-id $MY_NETWORK_ACL_ID --ingress --rule-number 10100 # Delete the existing inbound rule from the network ACL to block all IPv6-based traffic over port 80. Advanced users: if you originally created this rule with a different number, change this suggested rule number to match.
```

如果您已使用完此环境，请将其删除，以免一直对您的 AWS 账户 收费。有关说明，请参阅[在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 WordPress 教程

此教程让您能够在 AWS Cloud9 开发环境内安装并运行 WordPress。WordPress 是一个广泛应用于交付 Web 内容的开源内容管理系统 (CMS)。

Note

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。这些包括可能对 Amazon Elastic Compute Cloud (Amazon EC2) 等服务收取的费用。有关更多信息，请参阅[Amazon EC2 定价](#)。

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。
- 您拥有一个包含所有最新软件包的最新 EC2 实例。在 AWS Cloud9 IDE 终端窗口中，您可以使用 `-y` 选项运行 `yum update` 以安装更新而无需确认。如果您希望在安装前检查更新，则可以忽略该选项。

```
sudo yum update -y
```

安装概览

在环境的 EC2 实例上安装 WordPress 包含以下步骤：

1. 安装和配置 MariaDB Server，这是一个开源关系数据库，用于存储 WordPress 安装的信息
2. 安装和配置 WordPress，其中包括编辑 `wordpress.conf` 配置文件
3. 配置托管 WordPress 站点的 Apache 服务器
4. 预览由 Apache 服务器托管的 WordPress web 内容

步骤 1：安装和配置 MariaDB Server

1. 在 AWS Cloud9 IDE 中，依次选择 Window (窗口)、New Terminal (新建终端) 并输入以下命令来安装和启动 MariaDB Server 安装：

```
sudo yum install -y mariadb-server
sudo systemctl start mariadb
```

2. 接下来，运行 `mysql_secure_installation` 脚本以提高 MariaDB Server 安装的安全性。

在提供对脚本的响应时，对第一个问题按 Enter 以保持根密码为空。对于 Set root password?，按 n，然后对于其余的每个安全选项，按 y。

```
mysql_secure_installation
```

3. 现在使用 MariaDB 客户端创建一个数据库表来存储 WordPress 信息。

(当系统要求您输入密码时，按 Enter。)

```
sudo mysql -u root -p
MariaDB [(none)]> create database wp_test;
MariaDB [(none)]> grant all privileges on wp_test.* to root@localhost identified by
';'
```

4. 若要注销 MariaDB 客户端，请运行 `exit` 命令。

步骤 2：安装和配置 WordPress

1. 在 IDE 终端窗口中，导航到 `environment` 目录，然后创建目录 `config` 和 `wordpress`。然后运行 `touch` 命令以在 `config` 目录中创建一个名为 `wordpress.conf` 的文件：

```
cd /home/ec2-user/environment
mkdir config wordpress
touch config/wordpress.conf
```

2. 使用 IDE 编辑器或 `vim`，通过主机配置信息更新 `wordpress.conf`，该信息允许 Apache 服务器为 WordPress 内容提供服务：

```
# Ensure that Apache listens on port 80
Listen 8080
<VirtualHost *:8080>
    DocumentRoot "/var/www/wordpress"
    ServerName www.example.org
    # Other directives here
</VirtualHost>
```

3. 现在运行以下命令来检索所需的归档文件并安装 WordPress：

```
cd /home/ec2-user/environment
wget https://wordpress.org/latest.tar.gz
tar xvf latest.tar.gz
```

4. 运行 `touch` 命令以在 `environment/wordpress` 目录中创建一个名为 `wp-config.php` 的文件：

```
touch wordpress/wp-config.php
```

5. 使用 IDE 编辑器或 `vim` 更新 `wp-config.php`，然后使用您的设置替换示例数据：

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wp_test' );

/** MySQL database username */
define( 'DB_USER', 'wp_user' );

/** MySQL database password */
define( 'DB_PASSWORD', 'YourSecurePassword' );
```

```
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

define('FORCE_SSL', true);

if ($_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https') $_SERVER['HTTPS'] = 'on';
```

步骤 3：配置 Apache HTTP Server

1. 在 AWS Cloud9 IDE 终端窗口中，请确保已安装 Apache：

```
httpd -v
```

如果 Apache 服务器并未安装，请运行以下命令：

```
sudo yum install -y httpd
```

2. 导航到 `/etc/httpd/conf.d` 目录，该目录是 Apache 虚拟主机配置文件的位置。然后使用 `ln` 命令将您之前创建的 `wordpress.conf` 链接到当前工作目录 (`/etc/httpd/conf.d`)：

```
cd /etc/httpd/conf.d
sudo ln -s /home/ec2-user/environment/config/wordpress.conf
```

3. 现在导航到 `/var/www` 目录，该目录是 Apache 服务器的默认根文件夹。然后使用 `ln` 命令将您之前创建的 `wordpress` 目录链接到当前工作目录 (`/var/www`)：

```
cd /var/www
sudo ln -s /home/ec2-user/environment/wordpress
```

4. 运行 `chmod` 命令以允许 Apache 服务器在 `wordpress` 子目录中运行内容：

```
sudo chmod +x /home/ec2-user/
```

5. 现在重启 Apache 服务器以允许其检测新的配置：

```
sudo service httpd restart
```

步骤 4：预览 WordPress web 内容

1. 使用 AWS Cloud9 IDE，在以下目录创建名为 `index.html` 的新文件：`environment/wordpress`。
2. 将 HTML 格式的文本添加到 `index.html`。例如：

```
<h1>Hello World!</h1>
```

3. 在环境窗口中，选择 `index.html` 文件，然后依次选择预览、预览运行的应用程序。

显示 Hello World！（你好世界！）消息的网页，会显示在应用程序预览选项卡中。要在首选浏览器中查看 Web 内容，请选择 Pop Out Into a New Window（在新的弹出窗口中显示）。

如果您删除 `index.html` 文件并刷新应用程序预览选项卡，会显示 WordPress 配置页面。

管理混合内容错误

如果 WordPress 站点同时加载 HTTPS 和 HTTP 脚本或内容，Web 浏览器会显示 WordPress 站点的混合内容错误。错误消息的措辞取决于您正在使用的 Web 浏览器，但会通知您与站点的连接不安全或不完全安全。您的 Web 浏览器会阻止访问混合内容。

Important

默认情况下，您在 AWS Cloud9 IDE 的应用程序预览选项卡中访问的所有网页都会自动使用 HTTPS 协议。如果页面的 URI 具有不安全的 http 协议，则会自动替换为 https。并且您无法通过将 https 手动更改回 http 的方式来访问不安全的内容。有关为您的网站实施 HTTPS 的指南，请参阅 [WordPress 文档](#)。

适用于 AWS Cloud9 的 Java 教程

Important

如果您使用的是由具有 2 GiB 或更多内存的 EC2 实例支持的 AWS Cloud9 开发环境，我们建议您激活增强型 Java 支持。这提供了对生产率特性的访问，比如代码完成、错误的 linting、特定于上下文的操作，以及调试选项，比如断点和步进。

有关更多信息，请参阅[增强了对 Java 开发的支持](#)。

该教程让您能够在 AWS Cloud9 开发环境中运行部分 Java 代码。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅[Amazon EC2 定价](#)和[Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2：添加代码](#)
- [步骤 3：生成并运行代码](#)
- [步骤 4：设置以使用 AWS SDK for Java](#)
- [步骤 5：在环境中设置 AWS 凭证管理](#)
- [步骤 6：添加 AWS 开发工具包代码](#)
- [步骤 7：构建并运行 AWS 开发工具包代码](#)
- [步骤 8：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，您将在 AWS Cloud9 开发环境中安装一组 Java 开发工具。如果您已在环境中安装了一组 Java 开发工具（如 Oracle JDK 或 OpenJDK），则可以向前跳到 [步骤 2：添加代码](#)。此示例是使用 OpenJDK 8 开发的，您可以通过完成以下过程在环境中安装它。

1. 确认是否已安装 OpenJDK 8。为此，请在 AWS Cloud9 IDE 的终端会话中，运行带 **-version** 选项的 Java 运行程序的命令行版本。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口) > New Terminal (新建终端) 。

```
java -version
```

根据上述命令的输出，执行以下操作之一：

- 如果输出说明未找到 java 命令，请继续此过程中的步骤 2 来安装 OpenJDK 8。
 - 如果输出包含以 Java(TM)、Java Runtime Environment、Java SE、J2SE 或 Java2 开头的值，则表明未安装 OpenJDK 或未将 OpenJDK 设置为默认 Java 开发工具集。继续此过程中的步骤 2 来安装 OpenJDK 8，然后切换为使用 OpenJDK 8。
 - 如果输出包含以 java version 1.8 和 OpenJDK 开头的值，请向前跳到 [步骤 2：添加代码](#)。已为此示例正确安装 OpenJDK 8。
 - 如果输出包含低于 java version 的 1.8 和以 OpenJDK 开头的值，请继续此过程中的步骤 2 来将已安装的 OpenJDK 版本升级到 OpenJDK 8。
2. 确保已安装最新的安全更新和错误修复。为此，请使用 **update** 命令运行 yum 工具（适用于 Amazon Linux）或 apt 工具（适用于 Ubuntu Server）。

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu Server：

```
sudo apt update
```

3. 安装 OpenJDK 8。为此，请使用 **install** 命令运行 yum 工具（适用于 Amazon Linux）或 apt 工具（适用于 Ubuntu Server），指定 OpenJDK 8 程序包。

对于 Amazon Linux：

```
sudo yum -y install java-1.8.0-openjdk-devel
```

对于 Ubuntu Server :

```
sudo apt install -y openjdk-8-jdk
```

有关更多信息，请参阅 OpenJDK 网站上的[如何下载并安装预先构建的 OpenJDK 程序包](#)。

4. 将默认的 Java 开发工具集切换或升级到 OpenJDK 8。为此，请运行带 **--config** 选项的 **update-alternatives** 命令。运行此命令两次来切换或升级 Java 运行程序和编译器的命令行版本。

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

在每个提示符处，键入 OpenJDK 8 的选择编号（包含 java-1.8 的那一个）。

5. 确认 Java 运行程序和编译器的命令行版本使用的是 OpenJDK 8。为此，请运行带 **-version** 选项的 Java 运行程序和编译器的命令行版本。

```
java -version
javac -version
```

如果已正确安装并设置 OpenJDK 8，则 Java 运行程序版本输出会包含以 `openjdk version 1.8` 开头的值，并且 Java 编译器版本输出会以值 `javac 1.8` 开头。

步骤 2：添加代码

在 AWS Cloud9 IDE 中，创建一个包含以下代码的文件，并使用文件名 `hello.java` 保存该文件。（要创建文件，请在菜单栏上依次选择 File（文件）、New File（新建文件）。要保存文件，请依次选择 File（文件）、Save（保存）。）

```
public class hello {

    public static void main(String []args) {
        System.out.println("Hello, World!");

        System.out.println("The sum of 2 and 3 is 5.");
    }
}
```

```
int sum = Integer.parseInt(args[0]) + Integer.parseInt(args[1]);

System.out.format("The sum of %s and %s is %s.\n",
    args[0], args[1], Integer.toString(sum));
}
}
```

步骤 3：生成并运行代码

1. 使用 Java 编译器的命令行版本将 `hello.java` 文件编译为 `hello.class` 文件。为此，请使用 AWS Cloud9 IDE 中的终端，从 `hello.java` 文件所在的相同目录中，运行 Java 编译器，指定 `hello.java` 文件。

```
javac hello.java
```

2. 使用 Java 运行程序的命令行版本来运行 `hello.class` 文件。为此，请从 `hello.class` 文件所在的相同目录中，运行 Java 运行程序，指定在 `hello` 文件中声明的 `hello.java` 类的名称，并添加两个整数（例如 5 和 9）。

```
java hello 5 9
```

3. 比较您的输出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```

步骤 4：设置以使用 AWS SDK for Java

您可以改进此示例，以使用 AWS SDK for Java 创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除刚刚创建的存储桶。

在此步骤中，您将在环境中安装 [Apache Maven](#) 或 [Gradle](#)。Maven 和 Gradle 是可用于 Java 项目的常见构建自动化系统。在您安装 Maven 或 Gradle 后，您可以使用它来生成新的 Java 项目。在此新项目中，您将添加对 AWS SDK for Java 的引用。此 AWS SDK for Java 提供了从 Java 代码与 AWS 服务（如 Amazon S3）交互的便捷方式。

主题

- [使用 Maven 设置](#)

- [使用 Gradle 设置](#)

使用 Maven 设置

1. 在环境中安装 Maven。要查看是否已安装 Maven，请使用 AWS Cloud9 IDE 中的终端，运行带 **-version** 选项的 Maven。

```
mvn -version
```

如果成功，输出将包含 Maven 版本号。如果已安装 Maven，请向前跳到此过程中的步骤 4 来使用 Maven 在环境中生成新的 Java 项目。

2. 通过使用终端运行以下命令来安装 Maven。

对于 Amazon Linux，以下命令将获取有关 Maven 存储所在的程序包存储库的信息，然后使用此信息来安装 Maven。

```
sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
sudo yum install -y apache-maven
```

有关上述命令的更多信息，请参阅 Fedora 项目 Wiki 网站上的 [Extra Packages for Enterprise Linux \(EPEL\)](#)。

对于 Ubuntu Server，则运行以下命令。

```
sudo apt install -y maven
```

3. 带 **-version** 选项运行 Maven 以确认安装。

```
mvn -version
```

4. 使用 Maven 生成新的 Java 项目。为此，请使用终端从您希望 Maven 在其中生成项目的目录（例如，环境的根目录）中运行以下命令。

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```


上述命令在环境中为项目创建以下目录结构。

```
my-app
|- src
|  `-- main
|     `-- java
|         `-- com
|             `-- mycompany
|                 `-- app
|                     `--App.java
|- test
|  `-- java
|     `-- com
|         `-- mycompany
|             `-- app
|                 `-- AppTest.java
`-- pom.xml
```

有关上述目录结构的更多信息，请参阅 Apache Maven 项目网站上的 [Maven Quickstart 原型和标准目录布局简介](#)。

5. 修改项目的项目对象模型 (POM) 文件。(POM 文件定义 Maven 项目的设置。) 为此，请从 Environment (环境) 窗口中打开 my-app/pom.xml 文件。在编辑器中，将该文件的当前内容替换为以下代码，然后保存 pom.xml 文件。

```
<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/
maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-assembly-plugin</artifactId>
        <version>3.6.0</version>
        <configuration>
          <descriptorRefs>
```

```
        <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
    <archive>
        <manifest>
            <mainClass>com.mycompany.app.App</mainClass>
        </manifest>
    </archive>
</configuration>
<executions>
    <execution>
        <phase>package</phase>
        <goals>
            <goal>single</goal>
        </goals>
    </execution>
</executions>
</plugin>
</plugins>
</build>
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk</artifactId>
        <version>1.11.330</version>
    </dependency>
</dependencies>
</project>
```

上述 POM 文件包含指定声明的项目设置，如下所示：

- artifactId 的 my-app 设置将设置项目的根目录名称，group-id 的 com.mycompany.app 设置将设置 com/mycompany/app 子目录结构以及 package 和 App.java 文件中的 AppTest.java 声明。
- artifactId 的 my-app 设置、packaging 的 jar 设置、version 的 1.0-SNAPSHOT 设置和 descriptorRef 的 jar-with-dependencies 设置将设置 my-app-1.0-SNAPSHOT-jar-with-dependencies.jar 的输出 JAR 文件的名称。

- `plugin` 部分声明将构建单个 JAR，其中包括所有依赖项。
- 具有 `com.amazon.aws` 的 `groupId` 设置和 `aws-java-sdk` 的 `artifactId` 设置的 `dependency` 部分包含 AWS SDK for Java 库文件。通过 `version` 设置声明要使用的 AWS SDK for Java 版本。要使用其他版本，请替换此版本号。

向前跳至 [步骤 5：在环境中设置 AWS 凭证管理](#)。

使用 Gradle 设置

1. 在环境中安装 Gradle。要查看是否已安装 Gradle，请使用 AWS Cloud9 IDE 中的终端，运行带 `-version` 选项的 Gradle。

```
gradle -version
```

如果成功，输出将包含 Gradle 版本号。如果已安装 Gradle，请向前跳到此过程中的步骤 4 来使用 Gradle 在环境中生成新的 Java 项目。

2. 通过使用终端运行以下命令来安装 Gradle。这些命令安装并运行 SDKMAN! 工具，然后使用 SDKMAN! 安装最新版本的 Gradle。

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk install gradle
```

有关上述命令的更多信息，请参阅 SDKMAN! 网站上的[安装](#)以及 Gradle 网站上的[使用包管理器进行安装](#)。

3. 运行带 `-version` 选项的 Gradle 以确认安装。

```
gradle -version
```

4. 使用 Gradle 在环境中生成新的 Java 项目。为此，请使用终端运行以下命令来为项目创建目录，然后切换到该目录。

```
mkdir my-app
cd my-app
```

5. 运行以下命令来让 Gradle 在环境中的 `my-app` 目录中生成新的 Java 应用程序项目。

```
gradle init --type java-application
```

上述命令在环境中为项目创建以下目录结构。

```
my-app
|- .gradle
|  `-- (various supporting project folders and files)
|- gradle
|  `-- (various supporting project folders and files)
|- src
|  |- main
|  |   `-- java
|  |       `-- App.java
|  `-- test
|       `-- java
|           `-- AppTest.java
|- build.gradle
|- gradlew
|- gradlew.bat
`-- settings.gradle
```

6. 修改项目的 `AppTest.java`。(如果不这样做,则项目可能不会按预期方式构建或运行。)为此,请从 `Environment` (环境) 窗口中打开 `my-app/src/test/java/AppTest.java` 文件。在编辑器中,将该文件的当前内容替换为以下代码,然后保存 `AppTest.java` 文件。

```
import org.junit.Test;
import static org.junit.Assert.*;

public class AppTest {
    @Test public void testAppExists () {
        try {
            Class.forName("com.mycompany.app.App");
        } catch (ClassNotFoundException e) {
            fail("Should have a class named App.");
        }
    }
}
```

7. 修改项目的 `build.gradle` 文件。(`build.gradle` 文件定义 Gradle 项目的设置。)为此,请从 `Environment` (环境) 窗口中打开 `my-app/build.gradle` 文件。在编辑器中,将该文件的当前内容替换为以下代码,然后保存 `build.gradle` 文件。

```
apply plugin: 'java'
apply plugin: 'application'

repositories {
    jcenter()
    mavenCentral()
}

buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath "io.spring.gradle:dependency-management-plugin:1.0.3.RELEASE"
    }
}

apply plugin: "io.spring.dependency-management"

dependencyManagement {
    imports {
        mavenBom 'com.amazonaws:aws-java-sdk-bom:1.11.330'
    }
}

dependencies {
    compile 'com.amazonaws:aws-java-sdk-s3'
    testCompile group: 'junit', name: 'junit', version: '4.12'
}

run {
    if (project.hasProperty("appArgs")) {
        args Eval.me(appArgs)
    }
}

mainClassName = 'App'
```

上述 `build.gradle` 文件包含指定声明的项目设置，如下所示：

- `io.spring.dependency-management` 插件用于导入AWS SDK for Java Maven 材料清单 (BOM) 以管理项目的AWS SDK for Java依赖项。`classpath` 声明要使用的版本。要使用其他版本，请替换此版本号。
- `com.amazonaws:aws-java-sdk-s3` 包含 AWS SDK for Java 库文件的 Amazon S3 部分。`mavenBom` 声明要使用的版本。如果您希望使用其他版本，请替换此版本号。

步骤 5：在环境中设置 AWS 凭证管理

每次您使用 AWS SDK for Java 调用 AWS 服务时，都必须提供一组 AWS 凭证才能进行调用。这些凭证确定 AWS SDK for Java 是否具有相应的权限以进行该调用。如果凭证没有包括合适的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关其他信息，请参阅 AWS SDK for Java 开发人员指南中的 [设置用于开发的 AWS 凭证和区域](#)。

步骤 6：添加 AWS 开发工具包代码

在此步骤中，您添加用于与 Amazon S3 交互的代码，以创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。

从 Environment (环境) 窗口中，打开 `my-app/src/main/java/com/mycompany/app/App.java` 文件 (适用于 Maven) 或 `my-app/src/main/java/App.java` 文件 (适用于 Gradle)。在编辑器中，将该文件的当前内容替换为以下代码，然后保存 `App.java` 文件。

```
package com.mycompany.app;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.CreateBucketRequest;

import java.util.List;

public class App {

    private static AmazonS3 s3;
```

```
public static void main(String[] args) {
    if (args.length < 2) {
        System.out.format("Usage: <the bucket name> <the AWS Region to use>\n" +
            "Example: my-test-bucket us-east-2\n");
        return;
    }

    String bucket_name = args[0];
    String region = args[1];

    s3 = AmazonS3ClientBuilder.standard()
        .withCredentials(new ProfileCredentialsProvider())
        .withRegion(region)
        .build();

    // List current buckets.
    ListMyBuckets();

    // Create the bucket.
    if (s3.doesBucketExistV2(bucket_name)) {
        System.out.format("\nCannot create the bucket. \n" +
            "A bucket named '%s' already exists.", bucket_name);
        return;
    } else {
        try {
            System.out.format("\nCreating a new bucket named '%s'...\n\n",
bucket_name);
            s3.createBucket(new CreateBucketRequest(bucket_name, region));
        } catch (AmazonS3Exception e) {
            System.err.println(e.getErrorMessage());
        }
    }

    // Confirm that the bucket was created.
    ListMyBuckets();

    // Delete the bucket.
    try {
        System.out.format("\nDeleting the bucket named '%s'...\n\n", bucket_name);
        s3.deleteBucket(bucket_name);
    } catch (AmazonS3Exception e) {
        System.err.println(e.getErrorMessage());
    }
}
```

```
        // Confirm that the bucket was deleted.
        ListMyBuckets();
    }

    private static void ListMyBuckets() {
        List<Bucket> buckets = s3.listBuckets();
        System.out.println("My buckets now are:");

        for (Bucket b : buckets) {
            System.out.println(b.getName());
        }
    }
}
```

步骤 7：构建并运行 AWS 开发工具包代码

要运行上一步中的代码，请从终端中运行以下命令。这些命令使用 Maven 或 Gradle 为项目创建可执行的 JAR 文件，然后使用 Java 运行程序来运行 JAR。JAR 使用要在 Amazon S3 中创建的存储桶的名称（如 `my-test-bucket`）和要在其中创建存储桶的 AWS 区域的 ID（如 `us-east-2`）作为输入来运行。

对于 Maven，运行以下命令。

```
cd my-app
mvn package
java -cp target/my-app-1.0-SNAPSHOT-jar-with-dependencies.jar com.mycompany.app.App my-test-bucket us-east-2
```

对于 Gradle，运行以下命令。

```
gradle build
gradle run -PappArgs="['my-test-bucket', 'us-east-2']"
```

将结果与以下输出进行比较。

```
My buckets now are:

Creating a new bucket named 'my-test-bucket'...
```



```
My buckets now are:  
  
my-test-bucket  
  
Deleting the bucket named 'my-test-bucket'...  
  
My buckets now are:
```

步骤 8：清除

为防止在使用完该示例后一直对您的 AWS 账户收费，应删除环境。有关说明，请参阅 [在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 C++ 教程

在本教程中，您可以在 AWS Cloud9 开发环境中运行 C++ 代码。该代码还使用 [AWS SDK for C++](#) 提供的资源，这是一个模块化、跨平台的开源库，您可以使用它连接到 Amazon Web Services。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装 g++ 和所需的开发软件包](#)
- [步骤 2：安装 CMake](#)
- [步骤 3：获取并构建 SDK for C++](#)
- [步骤 4：创建 C++ 和 CMakeLists 文件](#)
- [步骤 5：构建并运行 C++ 代码](#)
- [步骤 6：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅 [在中创建环境 AWS Cloud9](#)。

- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装 g++ 和所需的开发软件包

要构建和运行 C++ 应用程序，您需要一个实用工具，例如 g++，这是一个由 [GNU 编译器集合 \(GCC\)](#) 提供的 C++ 编译器。

您还需要为 libcurl、libopenssl、libuuid、zlib 添加标头文件（-dev 程序包），并为 Amazon Polly 支持添加 libpulse（可选）。

安装开发工具的过程略有不同，具体取决于您使用的是 Amazon Linux/Amazon Linux 2 实例还是 Ubuntu 实例。

Amazon Linux-based systems

你可以检查你是否已经通过运行以下命令在 AWS Cloud9 终端安装 gcc：

```
g++ --version
```

如果 g++ 未安装，您可以轻松地将其安装为名为“开发工具”的软件包组的一部分。这些工具已添加到带有 yum groupinstall 命令的实例：

```
sudo yum groupinstall "Development Tools"
```

再次运行 g++ --version 以确认编译器已安装。

现在，使用系统的软件包管理器安装所需库的软件包：

```
sudo yum install libcurl-devel openssl-devel libuuid-devel pulseaudio-libs-devel
```

Ubuntu-based systems

你可以检查你是否已经通过运行以下命令在 AWS Cloud9 终端安装 gcc：

```
g++ --version
```

如果 gcc 未安装，则可以通过运行以下命令将其安装在基于 Ubuntu 的系统上：

```
sudo apt update
```

```
sudo apt install build-essential
sudo apt-get install manpages-dev
```

再次运行 `g++ --version` 以确认编译器已安装。

现在，使用系统的软件包管理器安装所需库的软件包：

```
sudo apt-get install libcurl4-openssl-dev libssl-dev uuid-dev zlib1g-dev libpulse-dev
```

步骤 2：安装 CMake

您需要安装 `cmake` 工具，该工具可自动执行从源代码构建可执行文件的流程。

1. 在 IDE 终端窗口中，运行以下命令以获取所需的归档文件：

```
wget https://cmake.org/files/v3.18/cmake-3.18.0.tar.gz
```

2. 从归档文件中提取文件并导航到包含解压缩文件的目录：

```
tar xzf cmake-3.18.0.tar.gz
cd cmake-3.18.0
```

3. 接下来，通过运行以下命令来运行引导程序脚本并安装 `cmake`：

```
./bootstrap
make
sudo make install
```

4. 通过运行以下命令以确认您已安装该工具：

```
cmake --version
```

步骤 3：获取并构建 SDK for C++

要设置 AWS SDK for C++，您可以直接从源代码构建开发工具包，也可以使用软件包管理器下载库。您可以在 [AWS SDK for C++ 开发人员指南](#) 中的 [AWS SDK for C++ 使用入门](#) 找到有关可用选项的详细信息。

本示例演示了使用 `git` 来克隆开发工具源代码，以及使用 `cmake` 构建 SDK for C++。

1. 在终端中运行以下命令，为您的 AWS Cloud9 环境克隆远程存储库并递归地获取所有 `git` 子模块：

```
git clone --recurse-submodules https://github.com/aws/aws-sdk-cpp
```

2. 导航到新的 `aws-sdk-cpp` 目录中，创建一个子目录来构建 AWS SDK for C++，然后导航到：

```
cd aws-sdk-cpp
mkdir sdk_build
cd sdk_build
```

- 3.

Note

为节省时间，该步骤仅构建 AWS SDK for C++ 的 Amazon S3 部分。如果要构建完整的开发工具包，请省略 `cmake` 命令中的 `-DBUILD_ONLY=s3`。

构建完整的 SDK for C++ 可能需要一小时以上才能完成，具体取决于 Amazon EC2 实例或您自己的服务器上的可用计算资源。

运行以下命令，使用 `cmake` 将 SDK for C++ 的 Amazon S3 部分构建到 `sdk_build` 目录下：

```
cmake .. -DBUILD_ONLY=s3
```

4. 现在，运行 `make install` 命令，以便可以访问构建的开发工具包：

```
sudo make install
cd ..
```

步骤 4：创建 C++ 和 CMakeLists 文件

在本步骤中，您将创建 C++ 文件，该文件允许项目的用户与 Amazon S3 存储桶进行交互。

您还可以创建 `CMakeLists.txt` 文件，该文件提供了 `cmake` 用于构建您的 C++ 库的说明。

1. 在 AWS Cloud9 IDE 中，创建一个包含该内容的文件，并在您的环境的根目录 (`/`) 中将该文件保存为 `s3-demo.cpp`。

```
#include <iostream>
```

```
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <aws/s3/model/Bucket.h>
#include <aws/s3/model/CreateBucketConfiguration.h>
#include <aws/s3/model/CreateBucketRequest.h>
#include <aws/s3/model/DeleteBucketRequest.h>

// Look for a bucket among all currently available Amazon S3 buckets.
bool FindTheBucket(const Aws::S3::S3Client &s3Client,
                  const Aws::String &bucketName) {

    Aws::S3::Model::ListBucketsOutcome outcome = s3Client.ListBuckets();

    if (outcome.IsSuccess()) {

        std::cout << "Looking for a bucket named '" << bucketName << "'..."
                  << std::endl << std::endl;

        Aws::Vector<Aws::S3::Model::Bucket> bucket_list =
            outcome.GetResult().GetBuckets();

        for (Aws::S3::Model::Bucket const &bucket: bucket_list) {
            if (bucket.GetName() == bucketName) {
                std::cout << "Found the bucket." << std::endl << std::endl;

                return true;
            }
        }

        std::cout << "Could not find the bucket." << std::endl << std::endl;
    }
    else {
        std::cerr << "ListBuckets error: "
                  << outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}

// Create an Amazon S3 bucket.
bool CreateTheBucket(const Aws::S3::S3Client &s3Client,
                    const Aws::String &bucketName,
                    const Aws::String& region) {
```

```
std::cout << "Creating a bucket named '"
    << bucketName << "'..." << std::endl << std::endl;

Aws::S3::Model::CreateBucketRequest request;
request.SetBucket(bucketName);

if (region != "us-east-1") {
    Aws::S3::Model::CreateBucketConfiguration createBucketConfig;
    createBucketConfig.SetLocationConstraint(
        Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
            region));
    request.SetCreateBucketConfiguration(createBucketConfig);
}

Aws::S3::Model::CreateBucketOutcome outcome =
    s3Client.CreateBucket(request);

if (outcome.IsSuccess()) {
    std::cout << "Bucket created." << std::endl << std::endl;
}
else {
    std::cerr << "CreateBucket error: "
        << outcome.GetError().GetMessage() << std::endl;
}

return outcome.IsSuccess();
}

// Delete an existing Amazon S3 bucket.
bool DeleteTheBucket(const Aws::S3::S3Client &s3Client,
    const Aws::String &bucketName) {

    std::cout << "Deleting the bucket named '"
        << bucketName << "'..." << std::endl << std::endl;

    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
        s3Client.DeleteBucket(request);

    if (outcome.IsSuccess()) {
        std::cout << "Bucket deleted." << std::endl << std::endl;
    }
}
```

```
    }
    else {
        std::cerr << "DeleteBucket error: "
                  << outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}

#ifdef TESTING_BUILD
// Create an S3 bucket and then delete it.
// Before and after creating the bucket, and again after deleting the bucket,
// try to determine whether that bucket still exists.
int main(int argc, char *argv[]) {

    if (argc < 3) {
        std::cout << "Usage: s3-demo <bucket name> <AWS Region>" << std::endl
                  << "Example: s3-demo my-bucket us-east-1" << std::endl;
        return 1;
    }

    Aws::SDKOptions options;
    Aws::InitAPI(options);
    {
        Aws::String bucket_name = argv[1];
        Aws::String region = argv[2];

        Aws::Client::ClientConfiguration config;

        config.region = region;

        Aws::S3::S3Client s3_client(config);

        if (!FindTheBucket(s3_client, bucket_name)) {
            return 1;
        }

        if (!CreateTheBucket(s3_client, bucket_name, region)) {
            return 1;
        }

        if (!FindTheBucket(s3_client, bucket_name)) {
            return 1;
        }
    }
}
```

```
        if (!DeleteTheBucket(s3_client, bucket_name)) {
            return 1;
        }

        if (!FindTheBucket(s3_client, bucket_name)) {
            return 1;
        }
    }
    Aws::ShutdownAPI(options);

    return 0;
}
#endif // TESTING_BUILD
```

2. 创建一个包含该内容的文件，并在您的环境的根目录 (CMakeLists.txt) 中将该文件保存为 /。通过使用该文件，您可以将您的代码生成成为一个可执行文件。

```
# A minimal CMakeLists.txt file for the AWS SDK for C++.

# The minimum version of CMake that will work.
cmake_minimum_required(VERSION 2.8)

# The project name.
project(s3-demo)

# Locate the AWS SDK for C++ package.
set(AWSSDK_ROOT_DIR, "/usr/local/")
set(BUILD_SHARED_LIBS ON)
find_package(AWSSDK REQUIRED COMPONENTS s3)

# The executable name and its source files.
add_executable(s3-demo s3-demo.cpp)

# The libraries used by your executable.
target_link_libraries(s3-demo ${AWSSDK_LINK_LIBRARIES})
```

步骤 5：构建并运行 C++ 代码

1. 在您已在其中保存 s3-demo.cpp 和 CMakeLists.txt 的环境根目录中，运行 cmake 来构建您的项目：


```
cmake .  
make
```

2. 您现在可以从命令行运行程序。在下面的命令中，将 `my-unique-bucket-name` 替换为 Amazon S3 存储桶的唯一名称，如有必要，将 `us-east-1` 替换为您想要创建存储桶的其他 AWS 区域的标识符。

```
./s3-demo my-unique-bucket-name us-east-1
```

如果程序成功运行，则将返回类似于以下内容的输出：

```
Looking for a bucket named 'my-unique-bucket-name'...  
  
Could not find the bucket.  
  
Creating a bucket named 'my-unique-bucket-name'...  
  
Bucket created.  
  
Looking for a bucket named 'my-unique-bucket-name'...  
  
Found the bucket.  
  
Deleting the bucket named 'my-unique-bucket-name'...  
  
Bucket deleted.  
  
Looking for a bucket named 'my-unique-bucket-name'...  
  
Could not find the bucket.
```

步骤 6：清除

为防止在您使用完该示例后一直对您的 AWS 账户收费，请删除环境。有关说明，请参阅[在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 Python 教程

本教程为您介绍如何在 AWS Cloud9 开发环境中运行 Python 代码。

按照本教程操作可能会对您的 AWS 账户收费。其中包括可能对 Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon Simple Storage Service (Amazon S3) 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装 Python](#)
- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [步骤 4：安装和配置 AWS SDK for Python \(Boto3\)](#)
- [步骤 5：添加 AWS 开发工具包代码](#)
- [步骤 6：运行 AWS 开发工具包代码](#)
- [步骤 7：清除](#)

先决条件

使用本教程之前，确保满足以下要求。

- 您有 AWS Cloud9 EC2 开发环境

本教程假设您已有 EC2 环境，并且该环境已连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例。有关详细信息，请参阅 [创建 EC2 环境](#)。

如果您有不同类型的环境或操作系统，您可能需要调整本教程的说明。

- 您已为该环境打开 AWS Cloud9 IDE

当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关详细信息，请参阅 [在 AWS Cloud9 中打开环境](#)。

步骤 1：安装 Python

1. 在 AWS Cloud9 IDE 的终端会话中，运行 **python --version** 命令以确认是否已安装 Python。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口)、New Terminal (新建终端)。）
如果已安装 Python，请向前跳至 [步骤 2：添加代码](#)。
2. 运行 **yum update** (适用于 Amazon Linux) 或 **apt update** (适用于 Ubuntu Server) 命令，以帮助确保已安装最新的安全更新和错误修复。

对于 Amazon Linux :

```
sudo yum -y update
```

对于 Ubuntu Server :

```
sudo apt update
```

3. 通过运行 **install** 命令来安装 Python。

对于 Amazon Linux :

```
sudo yum -y install python3
```

对于 Ubuntu Server :

```
sudo apt-get install python3
```

步骤 2 : 添加代码

在 AWS Cloud9 IDE 中，创建一个包含以下内容的文件，并使用文件名 `hello.py` 保存该文件。（要创建文件，请在菜单栏上依次选择 File（文件）、New File（新建文件）。要保存文件，请依次选择 File（文件）、Save（保存）。）

```
import sys

print('Hello, World!')

print('The sum of 2 and 3 is 5.')

sum = int(sys.argv[1]) + int(sys.argv[2])

print('The sum of {0} and {1} is {2}.'.format(sys.argv[1], sys.argv[2], sum))
```

步骤 3 : 运行代码

1. 在 AWS Cloud9 IDE 中的菜单栏上依次选择 Run（运行）、Run Configurations（运行配置）、New Run Configuration（新建运行配置）。

- 在 [New] - Stopped ([新建] - 已停止) 选项卡中，在 Command 中输入 `hello.py 5 9`。在代码中，5 表示 `sys.argv[1]`，9 表示 `sys.argv[2]`。
- 选择 Run (运行)，然后比较输出。

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```

- 默认情况下，AWS Cloud9 自动为您的代码选择运行程序。要更改运行程序，请选择 Runner (运行程序)，然后选择 Python 2 或 Python 3。

Note

您可以为特定版本的 Python 创建自定义运行程序。有关详细信息，请参阅 [创建生成器或运行程序](#)。

步骤 4：安装和配置 AWS SDK for Python (Boto3)

AWS SDK for Python (Boto3) 使您可以使用 Python 代码与 Amazon S3 等 AWS 服务进行交互。例如，您可以使用开发工具包创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除刚刚创建的存储桶。

安装 pip

在 AWS Cloud9 IDE 中，通过运行 `python -m pip --version` 命令确认 pip 是否已经安装 Python 的活动版本。如果已安装 pip，请跳到下一节。

要安装 pip，请运行以下命令。由于 `sudo` 与用户处于不同的环境中，因此，如果 Python 的版本与当前别名版本不同，则必须指定要使用的 Python 版本。

```
curl -O https://bootstrap.pypa.io/get-pip.py # Get the install script.  
sudo python3 get-pip.py                    # Install pip for Python 3.  
python -m pip --version                    # Verify pip is installed.  
rm get-pip.py                              # Delete the install script.
```

有关更多信息，请参阅 pip 网站上的[安装](#)。

安装 AWS SDK for Python (Boto3)

安装 pip 后，通过运行 **pip install** 命令安装 AWS SDK for Python (Boto3)。

```
sudo python3 -m pip install boto3 # Install boto3 for Python 3.
python -m pip show boto3         # Verify boto3 is installed for the current version
of Python.
```

有关更多信息，请参阅 [中的快速入门](#) AWS SDK for Python (Boto3) 的“安装”部分。

在环境中设置凭证

每次您使用 AWS SDK for Python (Boto3) 调用 AWS 服务时，都必须提供一组凭证才能进行调用。这些凭证确定开发工具包是否具有必需的权限以进行调用。如果凭证没有包括必需的权限，调用将失败。

要在环境中存储凭证，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明进行操作，然后返回到该主题。

有关其他信息，请参阅 [中的凭证](#) AWS SDK for Python (Boto3)。

步骤 5：添加 AWS 开发工具包代码

添加代码，该代码使用 Amazon S3 创建存储桶、列出可用存储桶并（可选）删除刚刚创建的存储桶。

在 AWS Cloud9 IDE 中，创建一个包含以下内容的文件，并使用文件名 `s3.py` 保存该文件。

```
import sys
import boto3
from botocore.exceptions import ClientError

def list_my_buckets(s3_resource):
    print("Buckets:\n\t", *[b.name for b in s3_resource.buckets.all()], sep="\n\t")

def create_and_delete_my_bucket(s3_resource, bucket_name, keep_bucket):
    list_my_buckets(s3_resource)

    try:
        print("\nCreating new bucket:", bucket_name)
        bucket = s3_resource.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
```

```
        "LocationConstraint": s3_resource.meta.client.meta.region_name
    },
)
except ClientError as e:
    print(
        f"Couldn't create a bucket for the demo. Here's why: "
        f"{e.response['Error']['Message']}"
    )
    raise

bucket.wait_until_exists()
list_my_buckets(s3_resource)

if not keep_bucket:
    print("\nDeleting bucket:", bucket.name)
    bucket.delete()

    bucket.wait_until_not_exists()
    list_my_buckets(s3_resource)
else:
    print("\nKeeping bucket:", bucket.name)

def main():
    import argparse

    parser = argparse.ArgumentParser()
    parser.add_argument("bucket_name", help="The name of the bucket to create.")
    parser.add_argument("region", help="The region in which to create your bucket.")
    parser.add_argument(
        "--keep_bucket",
        help="Keeps the created bucket. When not "
        "specified, the bucket is deleted "
        "at the end of the demo.",
        action="store_true",
    )

    args = parser.parse_args()
    s3_resource = (
        boto3.resource("s3", region_name=args.region)
        if args.region
        else boto3.resource("s3")
    )
    try:
```

```
    create_and_delete_my_bucket(s3_resource, args.bucket_name, args.keep_bucket)
except ClientError:
    print("Exiting the demo.")

if __name__ == "__main__":
    main()
```

步骤 6：运行 AWS 开发工具包代码

1. 在菜单栏上依次选择 Run (运行) > Run Configurations (运行配置) > New Run Configuration (新建运行配置)。
2. 对于 Command (命令)，输入 `s3.py my-test-bucket us-west-2`，其中 `my-test-bucket` 是要创建的存储桶的名称，`us-west-2` 是在其中创建存储桶的 AWS 区域的 ID。默认情况下，您的存储桶会在脚本退出之前被删除。要保留您的存储桶，请将 `--keep_bucket` 添加到您的命令中。有关 AWS 区域 ID 的列表，请参阅《AWS 一般参考》中的 [Amazon Simple Storage Service Endpoints and Quotas](#)。

Note

Amazon S3 存储桶名称在整个 AWS 中都必须唯一的，而不仅仅在您的 AWS 账户中唯一。

3. 选择 Run (运行)，然后比较输出。

Buckets:

```
a-pre-existing-bucket
```

Creating new bucket: my-test-bucket

Buckets:

```
a-pre-existing-bucket
my-test-bucket
```

Deleting bucket: my-test-bucket

Buckets:

```
a-pre-existing-bucket
```

步骤 7：清除

为防止在使用完此教程后一直对您的 AWS 账户收费，请删除 AWS Cloud9 环境。有关说明，请参阅[在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 .NET 教程

在本教程中，您可以在 AWS Cloud9 开发环境中运行 .NET 代码。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅[Amazon EC2 定价](#)和[Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2 \(可选 \)：为 Lambda 函数安装 .NET CLI 扩展](#)
- [步骤 3：创建 .NET 控制台应用程序项目](#)
- [步骤 4：添加代码](#)
- [步骤 5：生成并运行代码](#)
- [步骤 6：创建并设置使用 AWS SDK for .NET 的 .NET 控制台应用程序项目](#)
- [步骤 7：添加 AWS 开发工具包代码](#)
- [步骤 8：构建并运行 AWS 开发工具包代码](#)
- [步骤 9：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，将运行此示例需要的 .NET 开发工具包安装到环境中。

1. 确认环境中是否已安装 .NET 开发工具包的最新版本。要执行此操作，请在 AWS Cloud9 IDE 的终端会话中，运行带 `--version` 选项的 .NET Core 命令行界面 (CLI)。

```
dotnet --version
```

如果显示 .NET 命令行工具版本并且版本为 2.0 或更高版本，请向前跳至 [步骤 3：创建 .NET 控制台应用程序项目](#)。如果版本低于 2.0 或显示 `bash: dotnet: command not found` 等错误，请继续安装 .NET 开发工具包。

2. 对于 Amazon Linux，在 AWS Cloud9 IDE 的终端会话中，运行以下命令，确保已安装最新的安全更新和缺陷修复，并安装 .NET 开发工具包需要的 `libunwind` 程序包。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口)、New Terminal (新建终端)。）

```
sudo yum -y update
sudo yum -y install libunwind
```

对于 Ubuntu Server，在 AWS Cloud9 IDE 的终端会话中，运行以下命令，确保已安装最新的安全更新和错误修复。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口)、New Terminal (新建终端)。）

```
sudo apt -y update
```

3. 通过运行以下命令，将 .NET 开发工具包安装程序脚本下载到环境中。

```
wget https://dot.net/v1/dotnet-install.sh
```

4. 通过运行以下命令，使安装程序脚本可由当前用户执行。

```
sudo chmod u=rx dotnet-install.sh
```

5. 通过运行以下命令，运行安装程序脚本，该脚本将下载并安装 .NET 开发工具包。

```
./dotnet-install.sh -c Current
```

6. 将 .NET 开发工具包添加到 PATH。为此，在环境的 shell 配置文件（例如，`.bashrc` 文件）中，将 `$HOME/.dotnet` 子目录添加到环境的 PATH 变量，如下所示。

- a. 使用 `.bashrcvi` 命令打开 文件以供编辑。

```
vi ~/.bashrc
```

- b. 对于 Amazon Linux，使用向下键或 `j` 键，移至以 `export PATH` 开头的行。

对于 Ubuntu Server，通过键入 `G` 移动至文件最后一行。

- c. 使用向右键或 `$` 键移至行尾。
- d. 通过按 `i` 键切换到插入模式。(`-- INSERT ---` 将出现在显示内容的末尾。)
- e. 对于 Amazon Linux，通过键入 `:$HOME/.dotnet` 将 `$HOME/.dotnet` 子目录添加至 **PATH** 变量。请务必包含冒号字符 (`:`)。现在，行应类似于以下内容。

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet
```

对于 Ubuntu Server，请按向右箭头键，然后按 `Enter` 两次，然后在文件末尾单独输入以下代码行。

```
export PATH=$HOME/.dotnet:$PATH
```

- f. 保存该文件。要执行此操作，请按 `Esc` 键 (`-- INSERT ---` 将从显示内容的末尾消失)，键入 `:wq` (以写入然后退出文件)，然后按 `Enter`。
7. 通过提供 `.bashrc` 文件原来加载 `.NET` 开发工具包。

```
. ~/.bashrc
```

8. 通过运行带 `--help` 选项的 `.NET CLI` 确认已加载 `.NET` 开发工具包。

```
dotnet --help
```

如果成功，则显示 `.NET` 开发工具包版本号，以及附加的使用信息。

9. 如果不想再在环境中保留 `.NET` 开发工具包安装程序脚本，则可按如下所述将其删除。

```
rm dotnet-install.sh
```

步骤 2 (可选) : 为 Lambda 函数安装 .NET CLI 扩展

虽然本教程不是必需的，但如果您还安装了 Amazon.Lambda.Tools 程序包，则可以使用 .NET CLI 部署 AWS Lambda 函数和 AWS Serverless Application Model 应用程序。

1. 要安装此程序包，请运行以下命令：

```
dotnet tool install -g Amazon.Lambda.Tools
```

2. 现在设置 PATH 和 DOTNET_ROOT 环境变量，以指向已安装的 Lambda 工具。在 .bashrc 文件，查找 export PATH 部分，然后对其进行编辑，使其类似于以下内容（有关编辑此文件的详细信息，请参阅步骤 1）：

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet:$HOME/.dotnet/tools
export DOTNET_ROOT=$HOME/.dotnet
```

步骤 3 : 创建 .NET 控制台应用程序项目

在此步骤中，将使用 .NET 创建一个名为 hello 的项目。此项目包含 .NET 在 IDE 中从终端运行简单应用程序所需的全部文件。该应用程序的代码是用 C# 编写的。

创建 .NET 控制台应用程序项目。要执行此操作，请用 **new** 命令运行 .NET CLI，并指定要使用的控制台应用程序项目模板类型和编程语言（在本示例中为 C#）。

-n 选项表示将项目输出到新目录，hello。然后，导航到该目录。

```
dotnet new console -lang C# -n hello
cd hello
```

前面的命令会将一个名为 obj 且包含几个文件的子目录以及一些附加的独立文件添加至 hello 目录。您应注意以下两个关键文件：

- hello/hello.csproj 文件包含有关控制台应用程序项目的信息。
- hello/Program.cs 文件包含要运行的应用程序代码。

步骤 4 : 添加代码

在此步骤中，您将向应用程序中添加一些代码。

从 AWS Cloud9 IDE 的 Environment (环境) 窗口中打开 hello/Program.cs 文件。

在编辑器中，将该文件的当前内容替换为以下代码，然后保存 Program.cs 文件。

```
using System;

namespace hello
{
    class Program
    {
        static void Main(string[] args)
        {
            if (args.Length < 2) {
                Console.WriteLine("Please provide 2 numbers");
                return;
            }

            Console.WriteLine("Hello, World!");

            Console.WriteLine("The sum of 2 and 3 is 5.");

            int sum = Int32.Parse(args[0]) + Int32.Parse(args[1]);

            Console.WriteLine("The sum of {0} and {1} is {2}.",
                args[0], args[1], sum);

        }
    }
}
```

步骤 5：生成并运行代码

在此步骤中，您会将该项目及其依赖项生成为一组二进制文件，包括一个可运行的应用程序文件。然后运行该应用程序。

1. 在 IDE 中，为 .NET 创建一个生成器，如下所示。
 - a. 在菜单栏中，依次选择 Run (运行)、Build System (生成系统) 和 New Build System (新建生成系统)。
 - b. 在 My Builder.build (我的 Builder.build) 选项卡上，将选项卡的内容替换为以下代码。

```
{
```

```
"cmd" : ["dotnet", "build"],
"info" : "Building..."
}
```

- c. 依次选择 File (文件)、Save As (另存为)。
 - d. 对于 Filename (文件名)，键入 .NET.build。
 - e. 对于 Folder (文件夹)，键入 /.c9/builders。
 - f. 选择 Save (保存)。
2. 编辑器中显示 Program.cs 文件的内容后，依次选择 运行、生成系统和 .NET。然后依次选择 Run (运行)、Build (生成)。

此生成器将向 hello/obj 子目录中添加一个名为 bin 的子目录和一个名为 Debug 的子目录。请注意以下三个关键文件。

- hello/bin/Debug/netcoreapp3.1/hello.dll 文件是可运行的应用程序文件。
- hello/bin/Debug/netcoreapp3.1/hello.deps.json 文件列出了应用程序的依赖项。
- hello/bin/Debug/netcoreapp3.1/hello.runtimeconfig.json 文件为应用程序指定共享的运行时及其版本。

Note

文件夹名称 netcoreapp3.1，反映了本示例中使用的 .NET 开发工具包的版本。根据您的安装的版本，您可能会在文件夹名称中看到不同的编号。

3. 为 .NET 创建一个运行程序，如下所示。
 - a. 在菜单栏上，依次选择 Run (运行)、Run With (运行方式) 和 New Runner (新建运行程序)。
 - b. 在 My Runner.run (我的 Runner.run) 选项卡上，将该选项卡的内容替换为以下代码。

```
{
  "cmd" : ["dotnet", "run", "$args"],
  "working_dir": "$file_path",
  "info" : "Running..."
}
```

- c. 依次选择 File (文件)、Save As (另存为)。
- d. 对于 Filename (文件名)，键入 .NET.run。

- e. 对于 Folder (文件夹), 键入 `/.c9/runners`。
 - f. 选择 Save (保存)。
4. 运行应用程序, 将两个整数相加 (例如, 5 和 9), 如下所示。
 - a. 当编辑器中显示 `Program.cs` 文件的内容时, 依次选择 Run (运行)、Run Configurations (运行配置) 和 New Run Configuration (新建运行配置)。
 - b. 在 [新建] - 空闲 选项卡中, 选择运行程序: 自动, 然后选择 .NET。
 - c. 在 Command (命令) 框中, 键入 `hello 5 9`。
 - d. 选择 运行。

默认情况下, 此运行程序将指示 .NET 运行 `hello.dll` 目录中的 `hello/bin/Debug/netcoreapp3.1` 文件。

将输出与以下内容进行比较。

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```

步骤 6 : 创建并设置使用 AWS SDK for .NET 的 .NET 控制台应用程序项目

您可以改进此示例, 以使用 AWS SDK for .NET 创建 Amazon S3 存储桶, 列出您的可用存储桶, 然后删除刚刚创建的存储桶。

在此新项目中, 您将添加对 AWS SDK for .NET 的引用。AWS SDK for .NET 提供了从 .NET 代码与 AWS 服务 (如 Amazon S3) 交互的便捷方式。然后在环境中设置 AWS 凭证管理。AWS SDK for .NET 需要使用这些凭证来与 AWS 服务进行交互。

创建项目

1. 创建 .NET 控制台应用程序项目。要执行此操作, 请用 `new` 命令运行 .NET CLI, 并指定要使用的控制台应用程序项目模板类型和编程语言。

`-n` 选项表示将项目输出到新目录, `s3`。然后, 导航到该目录。

```
dotnet new console -lang C# -n s3  
cd s3
```

- 在 AWS SDK for .NET 中添加对 Amazon S3 程序包的项目引用。要执行此操作，请用 **add package** 命令运行 .NET CLI，同时指定 NuGet 中的 Amazon S3 程序包名称。（NuGet 定义用于 .NET 的程序包的创建、托管和使用方式，并为每个角色提供工具。）

```
dotnet add package AWSSDK.S3
```

在添加对 Amazon S3 程序包的项目引用时，NuGet 也会向 AWS SDK for .NET 的其余部分添加项目引用。

Note

有关 NuGet 中其他 AWS 相关程序包的名称和版本，请参阅 NuGet 网站上的[用 `aws-sdk` 贴标签的 NuGet 程序包](#)。

设置 AWS 凭证管理

每次您使用 AWS SDK for .NET 调用 AWS 服务时，都必须提供一组 AWS 凭证才能进行调用。这些凭证确定 AWS SDK for .NET 是否具有相应的权限以进行该调用。如果凭证没有包括合适的权限，调用将失败。

要在环境中存储凭证，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明进行操作，然后返回到该主题。

有关其他信息，请参阅 AWS SDK for .NET 开发人员指南中的[配置 AWS 凭证](#)。

步骤 7：添加 AWS 开发工具包代码

在此步骤中，您添加用于与 Amazon S3 交互的代码，以用于创建存储桶、删除刚刚创建的存储桶，然后列出可用的存储桶。

从 AWS Cloud9 IDE 的 Environment（环境）窗口中打开 `s3/Program.cs` 文件。在编辑器中，将该文件的当前内容替换为以下代码，然后保存 `Program.cs` 文件。

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;
```

```
namespace s3
{
    class Program
    {
        async static Task Main(string[] args)
        {
            if (args.Length < 2) {
                Console.WriteLine("Usage: <the bucket name> <the AWS Region to use>");
                Console.WriteLine("Example: my-test-bucket us-east-2");
                return;
            }

            if (args[1] != "us-east-2") {
                Console.WriteLine("Cannot continue. The only supported AWS Region ID is " +
                    "'us-east-2'.");
                return;
            }

            var bucketRegion = RegionEndpoint.USEast2;
            // Note: You could add more valid AWS Regions above as needed.

            using (var s3Client = new AmazonS3Client(bucketRegion)) {
                var bucketName = args[0];

                // Create the bucket.
                try
                {
                    if (await AmazonS3Util.DoesS3BucketExistV2Async(s3Client, bucketName))
                    {
                        Console.WriteLine("Cannot continue. Cannot create bucket. \n" +
                            "A bucket named '{0}' already exists.", bucketName);
                        return;
                    } else {
                        Console.WriteLine("\nCreating the bucket named '{0}'...", bucketName);
                        await s3Client.PutBucketAsync(bucketName);
                    }
                }
                catch (AmazonS3Exception e)
                {
                    Console.WriteLine("Cannot continue. {0}", e.Message);
                }
                catch (Exception e)
                {

```



```
    Console.WriteLine("Cannot continue. {0}", e.Message);
}

// Confirm that the bucket was created.
if (await AmazonS3Util.DoesS3BucketExistV2Async(s3Client, bucketName))
{
    Console.WriteLine("Created the bucket named '{0}'.", bucketName);
} else {
    Console.WriteLine("Did not create the bucket named '{0}'.", bucketName);
}

// Delete the bucket.
Console.WriteLine("\nDeleting the bucket named '{0}'...", bucketName);
await s3Client.DeleteBucketAsync(bucketName);

// Confirm that the bucket was deleted.
if (await AmazonS3Util.DoesS3BucketExistV2Async(s3Client, bucketName))
{
    Console.WriteLine("Did not delete the bucket named '{0}'.", bucketName);
} else {
    Console.WriteLine("Deleted the bucket named '{0}'.", bucketName);
};

// List current buckets.
Console.WriteLine("\nMy buckets now are:");
var response = await s3Client.ListBucketsAsync();

foreach (var bucket in response.Buckets)
{
    Console.WriteLine(bucket.BucketName);
}
}
}
}
```

步骤 8：构建并运行 AWS 开发工具包代码

在此步骤中，您会将该项目及其依赖项生成为一组二进制文件，包括一个可运行的应用程序文件。然后运行该应用程序。

1. 构建项目。要执行此操作，在编辑器中显示 `s3/Program.cs` 文件的内容后，请在菜单栏上依次选择 **Run (运行)**、**Build (生成)**。

2. 用要创建的 Amazon S3 存储桶的名称和要在其中创建存储桶的 AWS 区域的 ID (例如 , my-test-bucket 和 us-east-2) 运行应用程序 , 如下所示。
 - a. 当编辑器中仍显示 s3/Program.cs 文件的内容时 , 依次选择 Run (运行)、Run Configurations (运行配置)、New Run Configuration (新建运行配置)。
 - b. 在 [新建] - 空闲 选项卡中 , 选择运行程序: 自动 , 然后选择 .NET 。
 - c. 在 Command (命令) 框中 , 键入应用程序的名称、要创建的 Amazon S3 存储桶的名称、要在其中创建存储桶的 AWS 区域的 ID (例如 , s3 my-test-bucket us-east-2) 。
 - d. 选择 运行。

默认情况下 , 此运行程序将指示 .NET 运行 s3.dll 目录中的 s3/bin/Debug/netcoreapp3.1 文件。

将结果与以下输出进行比较。

```
Creating a new bucket named 'my-test-bucket'...
Created the bucket named 'my-test-bucket'.

Deleting the bucket named 'my-test-bucket'...
Deleted the bucket named 'my-test-bucket'.

My buckets now are:
```

步骤 9 : 清除

为防止在使用完该示例后一直对您的 AWS 账户收费 , 应删除环境。有关说明 , 请参阅 [在 AWS Cloud9 中删除环境](#)。

Node.js 的教程 AWS Cloud9

本教程使你能够在 AWS Cloud9 开发环境中运行一些 Node.js 脚本。

按照本教程并创建此示例可能会导致您的 AWS 账户被扣款。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息 , 请参阅 [Amazon EC2 定价](#)和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1 : 安装所需工具](#)

- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [第 4 步：在 Node.js JavaScript 中安装和配置 AWS 开发工具包](#)
- [第 5 步：添加 AWS SDK 代码](#)
- [第 6 步：运行 S AWS DK 代码](#)
- [步骤 7：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须拥有现有 AWS Cloud9 的 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅[在中创建环境 AWS Cloud9](#)。
- 您已经打开了现有环境的 AWS Cloud9 IDE。打开环境时，会在 Web 浏览器中 AWS Cloud9 打开该环境的 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在该步骤中，您安装运行本示例所需的 Node.js。

1. 在 AWS Cloud9 IDE 的终端会话中，通过运行 `node --version` 命令来确认是否已安装 Node.js。（要开始新的终端会话，请在菜单栏上依次选择 Window（窗口）> New Terminal（新建终端）。）如果成功，输出将包含 Node.js 版本号。如果 Node.js 已安装，请向前跳至[步骤 2：添加代码](#)。
2. 运行 `yum update`（适用于 Amazon Linux）或 `apt update`（适用于 Ubuntu Server）命令，以帮助确保已安装最新的安全更新和错误修复。

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu Server：

```
sudo apt update
```

- 要安装 Node.js，首先运行此命令下载节点版本管理器 (nvm)。(nvm 是一个简单的 Bash shell 脚本，可用于安装和管理 Node.js 版本。有关更多信息，请参阅 GitHub 网站上的[节点版本管理器](#)。)

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash
```

- 要开始使用 nvm，请关闭终端会话并重新启动，或者查找包含命令以加载 nvm 的 ~/.bashrc 文件。

```
. ~/.bashrc
```

- 运行此命令在 Amazon Linux 2、Amazon Linux 1 和 Ubuntu 18.04 上安装 Node.js 16。Amazon Linux 1 和 Ubuntu 18.04 实例最高仅支持 Node.js 版本 16。

```
nvm install 16
```

运行此命令在 Amazon Linux 2023 和 Ubuntu 22.04 上安装最新版本的 Node.js：

```
nvm install --lts && nvm alias default lts/*
```

Note

最新的 AL2023 AWS Cloud9 镜像安装了 Node.js 20，最新的亚马逊 Linux 2 AWS Cloud9 镜像安装了 Node.js 18。如果你想在 Amazon Linux 2 上 AWS Cloud9 手动安装 Node.js 18，请在 AWS Cloud9 IDE 终端中运行以下命令：

```
C9_NODE_INSTALL_DIR=~/.nvm/versions/node/v18.17.1
C9_NODE_URL=https://d3kgj69l4ph6w4.cloudfront.net/static/node-amazon/node-
v18.17.1-linux-x64.tar.gz
mkdir -p $C9_NODE_INSTALL_DIR
curl -fSs1 $C9_NODE_URL | tar xz --strip-components=1 -C
"$C9_NODE_INSTALL_DIR"
nvm alias default v18.17.1
nvm use default
echo -e 'nvm use default' >> ~/.bash_profile
```

步骤 2：添加代码

在 AWS Cloud9 IDE 中，使用此内容创建一个文件，然后使用该名称保存该文件 `hello.js`。（要创建文件，请在菜单栏上依次选择 File（文件）、New File（新建文件）。要保存文件，请依次选择 File（文件）、Save（保存）。）

```
console.log('Hello, World!');

console.log('The sum of 2 and 3 is 5.');
```

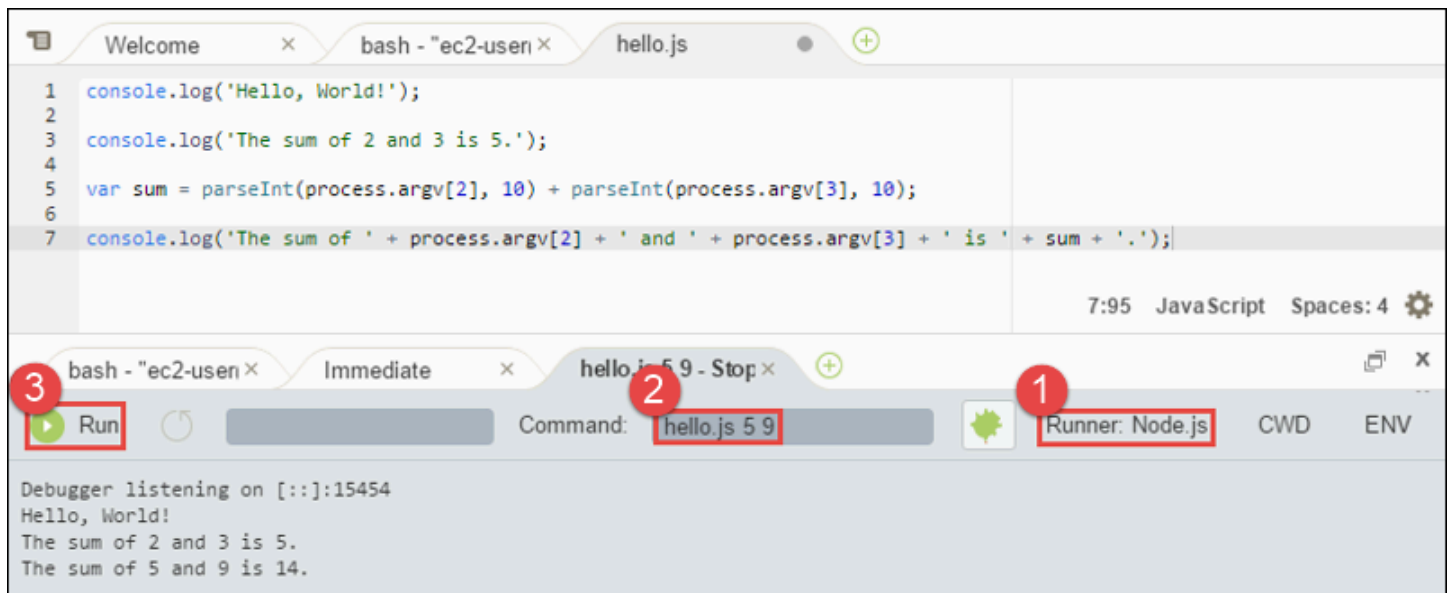
```
var sum = parseInt(process.argv[2], 10) + parseInt(process.argv[3], 10);

console.log('The sum of ' + process.argv[2] + ' and ' +
  process.argv[3] + ' is ' + sum + '.');
```

步骤 3：运行代码

1. 在 AWS Cloud9 IDE 的菜单栏上，选择“运行”、“运行配置”、“新建运行配置”。
2. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Node.js。
3. 在 Command (命令) 中，键入 `hello.js 5 9`。在代码中，5 表示 `process.argv[2]`，9 表示 `process.argv[3]`。（`process.argv[0]` 表示运行时的名称 (node)，`process.argv[1]` 表示文件的名称 (hello.js)。）
4. 选择 Run (运行) 按钮，然后比较输出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```



第 4 步：在 Node.js JavaScript 中安装和配置 AWS 开发工具包

在中运行 Node.js 脚本时 AWS Cloud9，可以在版本 3 (V3) 的 AWS SDK 和 JavaScript 版本 2 (V2) 的旧 JavaScript 版 AWS SDK 之间进行选择。与 V2 一样，V3 使您可以轻松地使用 Amazon Web Services，但它已经写入 TypeScript 并添加了几项经常要求的功能，例如模块化软件包。

AWS SDK for JavaScript (V3)

您可以增强此示例，使用 Node.js JavaScript 中的 AWS 软件开发工具包创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除您刚刚创建的存储桶。

在此步骤中，您将在 Node.js 中安装和配置 AWS 软件开发工具包的 Amazon S3 服务客户端模块，该模块提供了一种通过 JavaScript 代码与 Amazon S3 AWS 服务进行交互的便捷方式。JavaScript

如果要使用其他 AWS 服务，则需要单独安装它们。有关安装 AWS 模块的更多信息，请参阅 [AWS 开发人员指南 \(V3\)](#)。有关如何开始使用 Node.js 和 AWS 适用于 JavaScript (V3) 的 SDK 的信息，请参阅《JavaScript 开发者开发 AWS 工具包指南》(V3) 中的 [Node.js 入门](#)。

JavaScript 在 Node.js 中安装 AWS 适用的 SDK 后，您必须在自己的环境中设置凭据管理。Node.js JavaScript 中的 S AWS DK 需要这些凭据才能与 AWS 服务进行交互。

在 Node.js JavaScript 中安装 AWS 适用的 SDK

可以使用 npm 运行 **install** 命令。

```
npm install @aws-sdk/client-s3
```

有关更多信息，请参阅 [《AWS SDK for JavaScript 开发人员指南》 JavaScript 中的安装软件开发工具包](#)。

在环境中设置凭证管理

每次使用 Node.js JavaScript 中的 AWS SDK 调用 AWS 服务时，都必须在调用时提供一组凭据。这些凭证决定了 Node.js JavaScript 中的 S AWS DK 是否具有进行该调用的相应权限。如果这些凭证不包含相应的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关其他信息，请参阅 AWS SDK for JavaScript 开发人员指南中的 [在 Node.js 中设置凭证](#)。

AWS SDK for JavaScript (V2)

您可以增强此示例，使用 Node.js JavaScript 中的 AWS 软件开发工具包创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除您刚刚创建的存储桶。

在此步骤中，您将在 Node.js JavaScript 中安装和配置 AWS 软件开发工具包，它提供了一种通过 JavaScript 代码与 Amazon S3 等 AWS 服务进行交互的便捷方式。JavaScript 在 Node.js 中安装 AWS 适用的 SDK 后，您必须在自己的环境中设置凭据管理。Node.js JavaScript 中的 S AWS DK 需要这些凭据才能与 AWS 服务进行交互。

在 Node.js JavaScript 中安装 AWS 适用的 SDK

可以使用 npm 运行 **install** 命令。

```
npm install aws-sdk
```

有关更多信息，请参阅 [《AWS SDK for JavaScript 开发人员指南》 JavaScript 中的安装软件开发工具包](#)。

在环境中设置凭证管理

每次使用 Node.js JavaScript 中的 AWS SDK 调用 AWS 服务时，都必须在调用时提供一组凭据。这些凭证决定了 Node.js JavaScript 中的 S AWS DK 是否具有进行该调用的相应权限。如果这些凭证不包含相应的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关其他信息，请参阅 AWS SDK for JavaScript 开发人员指南中的 [在 Node.js 中设置凭证](#)。

第 5 步：添加 AWS SDK 代码

AWS SDK for JavaScript (V3)

在此步骤中，您会添加一些代码，此次用于与 Amazon S3 交互来创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。您将稍后运行此代码。

在 AWS Cloud9 IDE 中，使用此内容创建一个文件，然后使用该名称保存该文件 `s3.js`。

```
import {
  CreateBucketCommand,
  DeleteBucketCommand,
  ListBucketsCommand,
  S3Client,
} from "@aws-sdk/client-s3";

const wait = async (milliseconds) => {
  return new Promise((resolve) => setTimeout(resolve, milliseconds));
};

export const main = async () => {
  const client = new S3Client({});
  const now = Date.now();
  const BUCKET_NAME = `easy-bucket-${now.toString()}`;

  const createBucketCommand = new CreateBucketCommand({ Bucket: BUCKET_NAME });
  const listBucketsCommand = new ListBucketsCommand({});
  const deleteBucketCommand = new DeleteBucketCommand({ Bucket: BUCKET_NAME });

  try {
    console.log(`Creating bucket ${BUCKET_NAME}.`);
    await client.send(createBucketCommand);
    console.log(`${BUCKET_NAME} created`);

    await wait(2000);

    console.log(`Here are your buckets:`);
```



```
const { Buckets } = await client.send(listBucketsCommand);
Buckets.forEach((bucket) => {
  console.log(` • ${bucket.Name}`);
});

await wait(2000);

console.log(`Deleting bucket ${BUCKET_NAME}.`);
await client.send(deleteBucketCommand);
console.log(`${BUCKET_NAME} deleted`);
} catch (err) {
  console.error(err);
}
};

main();
```

AWS SDK for JavaScript (V2)

在此步骤中，您会添加一些代码，此次用于与 Amazon S3 交互来创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。您将稍后运行此代码。

在 AWS Cloud9 IDE 中，使用此内容创建一个文件，然后使用该名称保存该文件 `s3.js`。

```
if (process.argv.length < 4) {
  console.log(
    "Usage: node s3.js <the bucket name> <the AWS Region to use>\n" +
    "Example: node s3.js my-test-bucket us-east-2"
  );
  process.exit(1);
}

var AWS = require("aws-sdk"); // To set the AWS credentials and region.
var async = require("async"); // To call AWS operations asynchronously.

AWS.config.update({
  region: region,
});

var s3 = new AWS.S3({ apiVersion: "2006-03-01" });
var bucket_name = process.argv[2];
var region = process.argv[3];
```

```
var create_bucket_params = {
  Bucket: bucket_name,
  CreateBucketConfiguration: {
    LocationConstraint: region,
  },
};

var delete_bucket_params = { Bucket: bucket_name };

// List all of your available buckets in this AWS Region.
function listMyBuckets(callback) {
  s3.listBuckets(function (err, data) {
    if (err) {
    } else {
      console.log("My buckets now are:\n");

      for (var i = 0; i < data.Buckets.length; i++) {
        console.log(data.Buckets[i].Name);
      }
    }

    callback(err);
  });
}

// Create a bucket in this AWS Region.
function createMyBucket(callback) {
  console.log("\nCreating a bucket named " + bucket_name + "...");

  s3.createBucket(create_bucket_params, function (err, data) {
    if (err) {
      console.log(err.code + ": " + err.message);
    }

    callback(err);
  });
}

// Delete the bucket you just created.
function deleteMyBucket(callback) {
  console.log("\nDeleting the bucket named " + bucket_name + "...");

  s3.deleteBucket(delete_bucket_params, function (err, data) {
```

```
    if (err) {
      console.log(err.code + ": " + err.message);
    }

    callback(err);
  });
}

// Call the AWS operations in the following order.
async.series([
  listMyBuckets,
  createMyBucket,
  listMyBuckets,
  deleteMyBucket,
  listMyBuckets,
]);
```

第 6 步：运行 S AWS DK 代码

1. 使用 npm 运行 **install** 命令，以允许代码异步调用 Amazon S3 操作。

```
npm install async
```

2. 在 AWS Cloud9 IDE 的菜单栏上，选择“运行”、“运行配置”、“新建运行配置”。
3. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Node.js。
4. 如果您使用的是 JavaScript (V3) 的 AWS SDK，则使用命令类型 `s3.js`。如果您使用的是适用于 Javascript 的 AWS SDK (v2) `s3.js my-test-bucket us-east-2`，则对于命令类型，其中 `my-test-bucket` 是您要创建然后删除的存储桶的名称，`us-east-2` 也是您要在其中创建存储桶的 AWS 区域的 ID。有关更多 ID，请参阅 [中的](#) Amazon Simple Storage Service (Amazon S3) Amazon Web Services 一般参考。

Note

Amazon S3 存储桶名称必须是唯一的，而 AWS 不仅仅是您的 AWS 账户。

5. 选择 Run (运行) 按钮，然后比较输出。

```
My buckets now are:
```

```
Creating a new bucket named 'my-test-bucket'...
```

```
My buckets now are:
```

```
my-test-bucket
```

```
Deleting the bucket named 'my-test-bucket'...
```

```
My buckets now are:
```

步骤 7：清除

为了防止在您使用完此示例后继续向您的 AWS 账户收费，您应该删除该环境。有关说明，请参阅 [在 AWS Cloud9 中删除环境](#)。

的 PHP 教程 AWS Cloud9

本教程使你能够在 AWS Cloud9 开发环境中运行一些 PHP 脚本。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [步骤 4：安装和配置 AWS SDK for PHP](#)
- [第 5 步：添加 AWS SDK 代码](#)
- [第 6 步：运行 S AWS DK 代码](#)
- [步骤 7：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须拥有现有 AWS Cloud9 的 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅 [在中创建环境 AWS Cloud9](#)。
- 您已经打开了现有环境的 AWS Cloud9 IDE。打开环境时，会在 Web 浏览器中 AWS Cloud9 打开该环境的 IDE。有关更多信息，请参阅 [在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，您将安装运行此示例需要的 PHP。

Note

以下过程仅安装 PHP。要安装相关工具，例如 Apache Web 服务器和 MySQL 数据库，请参阅亚马逊 EC2 用户指南中的教程：[在亚马逊 Linux 上安装 LAMP Web 服务器](#)。

1. 在 AWS Cloud9 IDE 的终端会话中，通过运行 **php --version** 命令来确认 PHP 是否已安装。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口) > New Terminal (新建终端)。）如果成功，输出将包含 PHP 版本号。如果已安装 PHP，请向前跳至 [步骤 2：添加代码](#)。
2. 运行 **yum update** (适用于 Amazon Linux) 或 **apt update** (适用于 Ubuntu Server) 命令，以帮助确保已安装最新的安全更新和错误修复。

对于 Amazon Linux 2 和 Amazon Linux :

```
sudo yum -y update
```

对于 Ubuntu Server :

```
sudo apt update
```

3. 通过运行 **install** 命令安装 PHP。

对于 Amazon Linux 2 :

```
sudo amazon-linux-extras install -y php7.2
```

对于 Amazon Linux :

```
sudo yum -y install php72
```

Note

您可以使用以下命令查看 Amazon Linux 的版本：

```
cat /etc/system-release
```

对于 Ubuntu Server：

```
sudo apt install -y php php-xml
```

有关更多信息，请参阅 PHP 网站上的[安装和配置](#)。

步骤 2：添加代码

在 AWS Cloud9 IDE 中，使用此内容创建一个文件，然后使用该名称保存该文件 hello.php。（要创建文件，请在菜单栏上依次选择 File（文件）、New File（新建文件）。要保存文件，请依次选择 File（文件）、Save（保存），并为 Filename（文件名）键入 hello.php、然后选择 Save（保存）。）

```
<?php
print('Hello, World!');

print("\nThe sum of 2 and 3 is 5.");

$sum = (int)$argv[1] + (int)$argv[2];

print("\nThe sum of $argv[1] and $argv[2] is $sum.");
?>
```

Note

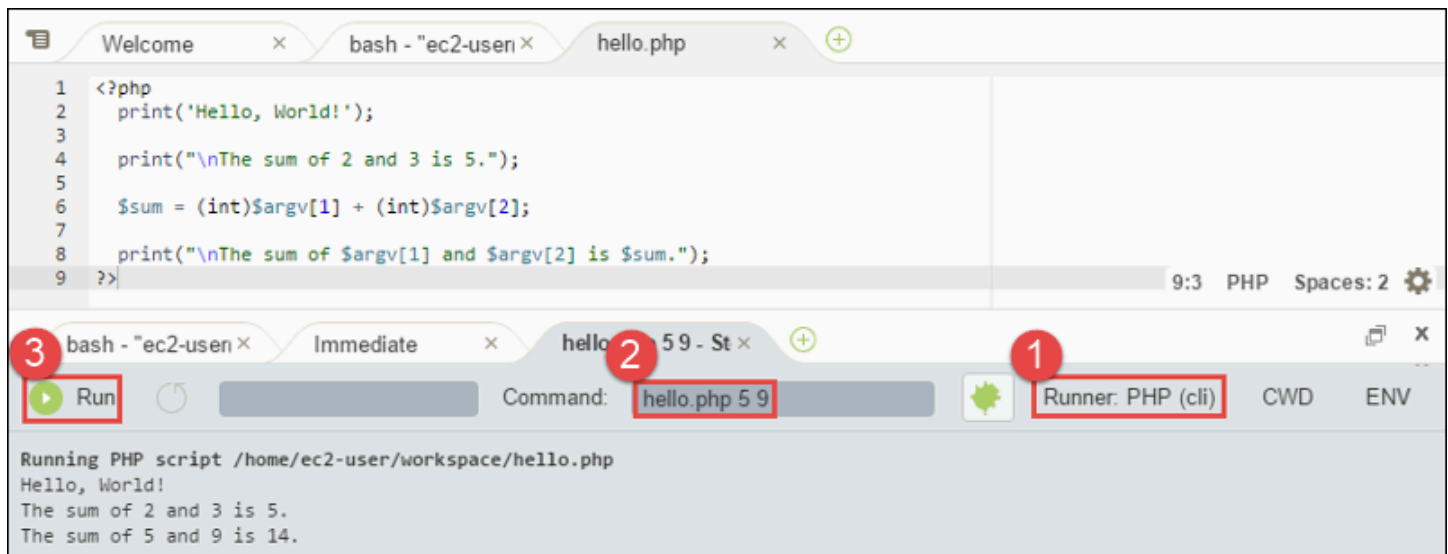
上述代码不依赖于任何外部文件。但是，如果您曾经在文件中包含或要求使用其他 PHP 文件，并且想要 AWS Cloud9 在键入时使用这些文件完成代码，请打开“首选项”中的“项目”、“PHP 支持”、“启用 PHP 代码完成”设置，然后将这些文件的路径添加到“项目”、“PHP

支持”、“PHP 完成包含路径”设置中。(要查看和更改您的首选项，请在菜单栏上选择 AWS Cloud9、Preferences (首选项)。)

步骤 3：运行代码

1. 在 AWS Cloud9 IDE 的菜单栏上，选择“运行”、“运行配置”、“新建运行配置”。
2. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序：自动)，然后选择 PHP (cli)。
3. 在 Command (命令) 中，键入 `hello.php 5 9`。在代码中，5 表示 `$argv[1]`，9 表示 `$argv[2]`。(`$argv[0]` 表示文件 (`hello.php`) 的名称。)
4. 选择 Run (运行) 按钮，然后比较输出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```



步骤 4：安装和配置 AWS SDK for PHP

您可以增强此示例，使用创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除您刚刚创建的存储桶。AWS SDK for PHP

在此步骤中，您将安装和配置 AWS SDK for PHP，它提供了一种通过您的 PHP 代码与 Amazon S3 等 AWS 服务进行交互的便捷方式。在安装之前 AWS SDK for PHP，您应该先安装 C [omposer](#)。安装

后 AWS SDK for PHP，您必须在您的环境中设置凭据管理。AWS SDK for PHP 需要这些凭据才能与 AWS 服务进行交互。

安装 Composer

使用静默 (`-s`) 和显示错误 (`-S`) 选项运行 `curl` 命令，将 Composer 安装程序传送到 PHP 档案 (PHAR) 文件，按约定名为 `composer.phar`。

```
curl -sS https://getcomposer.org/installer | php
```

要安装 AWS SDK for PHP

对于 Ubuntu Server，请安装 Composer 安装 AWS SDK for PHP 所需的其他软件包。

```
sudo apt install -y php-xml php-curl
```

对于 Amazon Linux 或 Ubuntu Server，请使用 `php` 命令运行 Composer 安装程序以安装 AWS SDK for PHP。

```
php composer.phar require aws/aws-sdk-php
```

此命令在环境中创建多个文件夹和文件。您将使用的主要文件为 `autoload.php`，位于环境中的 `vendor` 文件夹中。

Note

安装后，Composer 可能会建议您安装其他依赖项。您可以使用如下命令执行此操作，指定要安装的依赖项列表。例如，以下命令会指示 Composer 安装以下依赖项列表。

```
php composer.phar require psr/log ext-curl doctrine/cache aws/aws-php-sns-message-validator
```

有关更多信息，请参阅 AWS SDK for PHP 开发人员指南中的 [安装](#)。

在环境中设置凭证管理

每次使用调 AWS SDK for PHP 用 AWS 服务时，都必须为呼叫提供一组凭据。这些证书决定了是否 AWS SDK for PHP 具有进行该呼叫的相应权限。如果凭证没有包括合适的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关更多信息，请参阅 AWS SDK for PHP 开发人员指南中的 [基本用法](#) 的“创建客户端”部分。

第 5 步：添加 AWS SDK 代码

在此步骤中，您会添加一些代码，此次用于与 Amazon S3 交互来创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。您将稍后运行此代码。

在 AWS Cloud9 IDE 中，使用此内容创建一个文件，然后使用该名称保存该文件 s3.php。

```
<?php
require './vendor/autoload.php';

if ($argc < 4) {
    exit("Usage: php s3.php <the time zone> <the bucket name> <the AWS Region to use>
\n" .
        "Example: php s3.php America/Los_Angeles my-test-bucket us-east-2");
}

$timeZone = $argv[1];
$bucketName = $argv[2];
$region = $argv[3];

date_default_timezone_set($timeZone);

$s3 = new Aws\S3\S3Client([
    'region' => $region,
    'version' => '2006-03-01'
]);

# Lists all of your available buckets in this AWS Region.
function listMyBuckets($s3)
{
    print("\nMy buckets now are:\n");

    $promise = $s3->listBucketsAsync();

    $result = $promise->wait();

    foreach ($result['Buckets'] as $bucket) {
        print("\n");
    }
}
```

```
        print($bucket['Name']);
    }
}

listMyBuckets($s3);

# Create a new bucket.
print("\n\nCreating a new bucket named '$bucketName'...\n");

try {
    $promise = $s3->createBucketAsync([
        'Bucket' => $bucketName,
        'CreateBucketConfiguration' => [
            'LocationConstraint' => $region
        ]
    ]);

    $promise->wait();
} catch (Exception $e) {
    if ($e->getCode() == 'BucketAlreadyExists') {
        exit("\nCannot create the bucket. " .
            "A bucket with the name '$bucketName' already exists. Exiting.");
    }
}

listMyBuckets($s3);

# Delete the bucket you just created.
print("\n\nDeleting the bucket named '$bucketName'...\n");

$promise = $s3->deleteBucketAsync([
    'Bucket' => $bucketName
]);

$promise->wait();


listMyBuckets($s3);

?>
```

第 6 步：运行 S AWS DK 代码

1. 在 AWS Cloud9 IDE 的菜单栏上，选择“运行”、“运行配置”、“新建运行配置”。

- 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序：自动)，然后选择 PHP (cli)。
- 在 Command (命令) 中，键入 `s3.php America/Los_Angeles my-test-bucket us-east-2`，其中：
 - `America/Los_Angeles` 是默认时区 ID。有关更多 ID，请参阅 PHP 网站上[支持的时区列表](#)。
 - `my-test-bucket` 是您要创建，然后删除的存储桶的名称。

 Note

Amazon S3 存储桶名称必须是唯一的，而 AWS 不仅仅是您的 AWS 账户。

- `us-east-2` 是您要在其中创建存储桶的 AWS 区域的 ID。有关更多 ID，请参阅 [中的](#) Amazon Simple Storage Service (Amazon S3) Amazon Web Services 一般参考。
- 选择 Run (运行) 按钮，然后比较输出。

```
My buckets now are:

Creating a new bucket named 'my-test-bucket'...

My buckets now are:

my-test-bucket

Deleting the bucket named 'my-test-bucket'...

My buckets now are:
```

步骤 7：清除

为了防止在您使用完此示例后继续向您的 AWS 账户收费，您应该删除该环境。有关说明，请参阅[在 AWS Cloud9 中删除环境](#)。

针对 AWS Cloud9 的 PHP 运行程序问题的故障排除

如果您遇到 PHP CLI 运行程序的问题，则必须确保已将运行程序设置为 PHP，并确保启用调试器模式。

AWS Cloud9 中的 Ruby

有关将 AWS Cloud9 与适用于 Ruby 的 AWS SDK 结合使用的信息，请参阅《适用于 Ruby 的 AWS 开发人员指南》中的 [Using AWS Cloud9 with the AWS SDK for Ruby](#)。

Note

按照本教程操作可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

适用于 AWS Cloud9 的 Go 教程

该教程让您能够在 AWS Cloud9 开发环境中运行部分 Go 代码。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [步骤 4：安装和配置 AWS SDK for Go](#)
- [步骤 5：添加 AWS 开发工具包代码](#)
- [步骤 6：运行 AWS 开发工具包代码](#)
- [步骤 7：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅 [在中创建环境 AWS Cloud9](#)。

- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅[在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，安装和配置运行此示例需要的 Go。

1. 在 AWS Cloud9 IDE 的终端会话中，通过运行 **go version** 命令确认是否已安装 Go。（要开始新的终端会话，请在菜单栏上依次选择 Window（窗口）> New Terminal（新建终端）。）如果成功，则输出应包含 Go 版本号。否则，会输出错误消息。如果已安装 GO，请向前跳至[步骤 2：添加代码](#)。
2. 运行 **yum update**（对于 Amazon Linux）或 **apt update**（对于 Ubuntu Server）命令，以帮助确保已安装最新的安全更新和错误修复。

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu Server：

```
sudo apt update
```

3. 要安装 Go，请运行以下命令（一次运行一个命令）。

```
wget https://storage.googleapis.com/golang/go1.9.3.linux-amd64.tar.gz # Download
the Go installer.
sudo tar -C /usr/local -xzf ./go1.9.3.linux-amd64.tar.gz           # Install Go.
rm ./go1.9.3.linux-amd64.tar.gz                                   # Delete the
installer.
```

上述命令假定安装在编写本主题时最新的稳定版本 Go。有关更多信息，请在 Go Programming Language 网站上参阅 [Downloads](#)（下载）页面。

4. 将指向 Go 二进制文件的路径添加到 PATH 环境变量，如下所示。
 - a. 打开 Shell 配置文件（例如 `~/.bashrc`）以进行编辑。
 - b. 在这一行代码的末尾键入以下内容，现在的代码如下所示。

```
PATH=$PATH:/usr/local/go/bin
```

- c. 保存该文件。
5. 给出 `~/.bashrc` 文件来源，使终端现在可以找到您刚刚引用的 Go 二进制文件。

```
. ~/.bashrc
```

6. 通过运行 `go version` 命令确认此时已成功安装和配置 Go。如果成功，输出会包含 Go 版本号。

步骤 2：添加代码

在 AWS Cloud9 IDE 中，创建一个包含以下内容的文件，并使用文件名 `hello.go` 保存该文件。（要创建文件，请在菜单栏上依次选择 File（文件）、New File（新建文件）。要保存文件，请依次选择 File（文件）、Save（保存）。）

```
package main

import (
    "fmt"
    "os"
    "strconv"
)

func main() {
    fmt.Printf("Hello, World!\n")

    fmt.Printf("The sum of 2 and 3 is 5.\n")

    first, _ := strconv.Atoi(os.Args[1])
    second, _ := strconv.Atoi(os.Args[2])
    sum := first + second

    fmt.Printf("The sum of %s and %s is %s.",
        os.Args[1], os.Args[2], strconv.Itoa(sum))
}
```

步骤 3：运行代码

1. 在 AWS Cloud9 IDE 中的菜单栏上依次选择 Run（运行）、Run Configurations（运行配置）、New Run Configuration（新建运行配置）。
2. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Go。

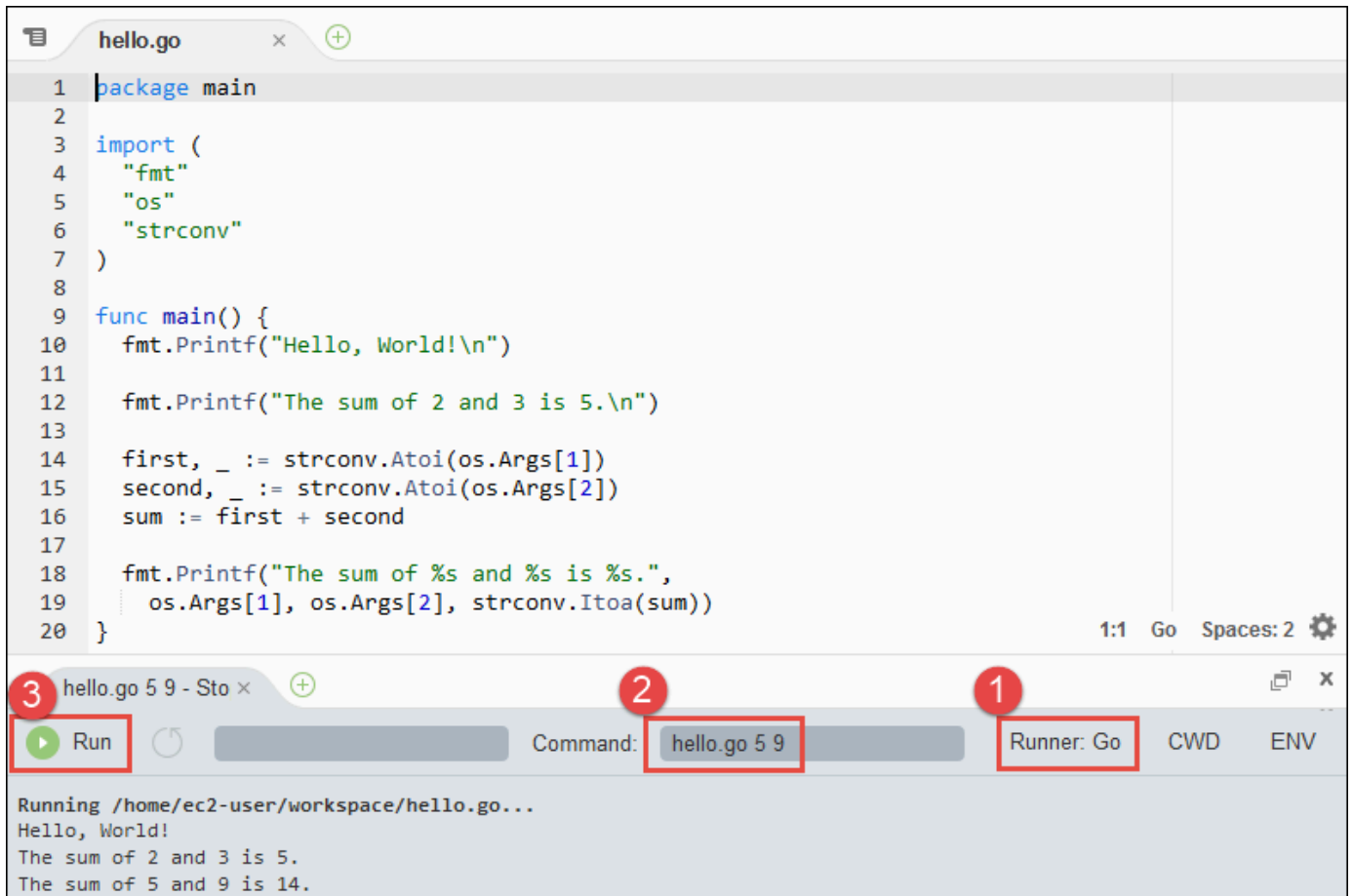
Note

如果 Go 不可用，您可以为 Go 创建自定义运行程序。

1. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 New Runner (新建运行程序)。
2. 在 My Runner.run (我的 Runner.run) 选项卡上，将选项卡的内容替换为以下代码。

```
{
  "cmd" : ["go", "run", "$file", "$args"],
  "info" : "Running $project_path$file_name...",
  "selector" : "source.go"
}
```

3. 在菜单栏上依次选择 File (文件)、Save As (另存为)，然后在 `/.c9/runners` 文件夹中将该文件保存为 `Go.run`。
 4. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Go。
 5. 选择 `hello.go` 选项卡以将其激活。
3. 在 Command (命令) 中，键入 `hello.go 5 9`。在代码中，5 表示 `os.Args[1]`，9 表示 `os.Args[2]`。



```
1 package main
2
3 import (
4     "fmt"
5     "os"
6     "strconv"
7 )
8
9 func main() {
10     fmt.Printf("Hello, World!\n")
11
12     fmt.Printf("The sum of 2 and 3 is 5.\n")
13
14     first, _ := strconv.Atoi(os.Args[1])
15     second, _ := strconv.Atoi(os.Args[2])
16     sum := first + second
17
18     fmt.Printf("The sum of %s and %s is %s.",
19         os.Args[1], os.Args[2], strconv.Itoa(sum))
20 }
```

1:1 Go Spaces: 2

3 Run Command: hello.go 5 9 Runner: Go CWD ENV

Running /home/ec2-user/workspace/hello.go...
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.

4. 选择 Run (运行) 按钮，然后比较输出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```

步骤 4：安装和配置 AWS SDK for Go

您可以改进此示例，以使用 AWS SDK for Go 创建 Amazon S3 存储桶，列出您的可用存储桶，然后删除刚刚创建的存储桶。

在该步骤中，您安装并配置 AWS SDK for Go，它提供了一种简便方法以从 Go 代码中与 AWS 服务（如 Amazon S3）交互。在安装之 AWS SDK for Go 前，必须设置 GOPATH 环境变量。安装 AWS SDK for Go 并设置 GOPATH 环境变量后，必须在环境中设置凭证管理。AWS SDK for Go 需要使用这些凭证来与 AWS 服务进行交互。

设置 GOPATH 环境变量

1. 打开您的 `~/.bashrc` 文件以进行编辑。
2. 在文件中的最后一行后面，键入以下代码。

```
GOPATH=~/.environment/go  
  
export GOPATH
```

3. 保存该文件。
4. 给出 `~/.bashrc` 文件来源，使终端现在可以找到您刚刚引用的 GOPATH 环境变量。

```
. ~/.bashrc
```

5. 通过运行 `echo $GOPATH` 命令，确认已成功设置 GOPATH 环境变量。如果成功，输出应为 `/home/ec2-user/environment/go` 或 `/home/ubuntu/environment/go`。

安装 AWS SDK for Go

运行 `go get` 命令，指定 AWS SDK for Go 源文件的位置。

```
go get -u github.com/aws/aws-sdk-go/...
```

Go 会将 AWS SDK for Go 源文件安装到 GOPATH 环境变量指定的位置（环境中的 `go` 文件夹）。

在环境中设置凭证管理

每次您使用 AWS SDK for Go 调用 AWS 服务时，都必须提供一组凭证才能进行调用。这些凭证确定 AWS SDK for Go 是否具有相应的权限以进行该调用。如果凭证没有包括合适的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关其他信息，请参阅 AWS SDK for Go 开发人员指南中的 [指定凭证](#)。

步骤 5：添加 AWS 开发工具包代码

在此步骤中，您会添加一些代码，此次用于与 Amazon S3 交互来创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。您将稍后运行此代码。

在 AWS Cloud9 IDE 中，创建一个包含以下内容的文件，并使用文件名 `s3.go` 保存该文件。

```
package main

import (
    "fmt"
    "os"

    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/s3"
)

func main() {

    if len(os.Args) < 3 {
        fmt.Printf("Usage: go run s3.go <the bucket name> <the AWS Region to use>\n" +
            "Example: go run s3.go my-test-bucket us-east-2\n")
        os.Exit(1)
    }

    sess := session.Must(session.NewSessionWithOptions(session.Options{
        SharedConfigState: session.SharedConfigEnable,
    }))
    svc := s3.New(sess, &aws.Config{
        Region: aws.String(os.Args[2]),
    })

    listMyBuckets(svc)
    createMyBucket(svc, os.Args[1], os.Args[2])
    listMyBuckets(svc)
    deleteMyBucket(svc, os.Args[1])
    listMyBuckets(svc)
}

// List all of your available buckets in this AWS Region.
func listMyBuckets(svc *s3.S3) {
    result, err := svc.ListBuckets(nil)

    if err != nil {
        exitErrorf("Unable to list buckets, %v", err)
    }

    fmt.Println("My buckets now are:\n")
}
```

```
for _, b := range result.Buckets {
    fmt.Printf(aws.StringValue(b.Name) + "\n")
}

fmt.Printf("\n")
}

// Create a bucket in this AWS Region.
func createMyBucket(svc *s3.S3, bucketName string, region string) {
    fmt.Printf("\nCreating a new bucket named '" + bucketName + "'...\n\n")

    _, err := svc.CreateBucket(&s3.CreateBucketInput{
        Bucket: aws.String(bucketName),
        CreateBucketConfiguration: &s3.CreateBucketConfiguration{
            LocationConstraint: aws.String(region),
        },
    })

    if err != nil {
        exitErrorf("Unable to create bucket, %v", err)
    }

    // Wait until bucket is created before finishing
    fmt.Printf("Waiting for bucket %q to be created...\n", bucketName)

    err = svc.WaitUntilBucketExists(&s3.HeadBucketInput{
        Bucket: aws.String(bucketName),
    })
}

// Delete the bucket you just created.
func deleteMyBucket(svc *s3.S3, bucketName string) {
    fmt.Printf("\nDeleting the bucket named '" + bucketName + "'...\n\n")

    _, err := svc.DeleteBucket(&s3.DeleteBucketInput{
        Bucket: aws.String(bucketName),
    })

    if err != nil {
        exitErrorf("Unable to delete bucket, %v", err)
    }

    // Wait until bucket is deleted before finishing
```

```
fmt.Printf("Waiting for bucket %q to be deleted...\n", bucketName)

err = svc.WaitUntilBucketNotExists(&s3.HeadBucketInput{
    Bucket: aws.String(bucketName),
})
}

// If there's an error, display it.
func exitErrorf(msg string, args ...interface{}) {
    fmt.Fprintf(os.Stderr, msg+"\n", args...)
    os.Exit(1)
}
```

步骤 6：运行 AWS 开发工具包代码

1. 在 AWS Cloud9 IDE 中的菜单栏上依次选择 Run (运行)、Run Configurations (运行配置)、New Run Configuration (新建运行配置)。
2. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Go。
3. 对于 Command (命令)，请键入 `s3.go YOUR_BUCKET_NAME THE_AWS_REGION`，其中 `YOUR_BUCKET_NAME` 是要创建并随后删除的存储桶的名称，`THE_AWS_REGION` 是要在其中创建存储桶的 AWS 区域的 ID。例如，对于美国东部 (俄亥俄) 区域，请使用 `us-east-2`。有关更多 ID，请参阅 [中的 Amazon Simple Storage Service \(Amazon S3\)](#) Amazon Web Services 一般参考。

Note

Amazon S3 存储桶名称在整个 AWS 中都必须唯一的，而不仅仅在您的 AWS 账户中唯一。

4. 选择 Run (运行) 按钮，然后比较输出。

```
My buckets now are:

Creating a new bucket named 'my-test-bucket'...

My buckets now are:

my-test-bucket
```

```
Deleting the bucket named 'my-test-bucket'...
```

```
My buckets now are:
```

步骤 7：清除

为防止在使用完该示例后一直对您的 AWS 账户收费，应删除环境。有关说明，请参阅 [在 AWS Cloud9 中删除环境](#)。

适用于 AWS Cloud9 的 TypeScript 教程

此教程演示如何在 AWS Cloud9 开发环境中使用 TypeScript。

按照本教程操作并创建此示例可能会对您的 AWS 账户收费。其中包括可能对 Amazon EC2 和 Amazon S3 等服务收取的费用。有关更多信息，请参阅 [Amazon EC2 定价](#) 和 [Amazon S3 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装所需工具](#)
- [步骤 2：添加代码](#)
- [步骤 3：运行代码](#)
- [步骤 4：安装并配置 AWS SDK for JavaScript in Node.js](#)
- [步骤 5：添加 AWS 开发工具包代码](#)
- [步骤 6：运行 AWS 开发工具包代码](#)
- [步骤 7：清除](#)

先决条件

在使用此示例之前，请确保您的设置满足以下要求：

- 您必须有现成的 AWS Cloud9 EC2 开发环境。本示例假设您已经有连接到运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例的 EC2 环境。如果您有不同类型的环境或操作系统，可能需要按照本示例的说明来设置相关的工具。有关更多信息，请参阅 [在中创建环境 AWS Cloud9](#)。
- 您已将适用于现有环境的 AWS Cloud9 IDE 打开。当您打开环境时，AWS Cloud9 会在 Web 浏览器中为该环境打开 IDE。有关更多信息，请参阅 [在 AWS Cloud9 中打开环境](#)。

步骤 1：安装所需工具

在此步骤中，您使用节点程序包管理器 (**npm**) 安装 TypeScript。要安装 **npm**，请使用节点版本管理器 (**nvm**)。如果您没有 **nvm**，则在此步骤中首先安装它。

1. 在 AWS Cloud9 IDE 的终端会话中，运行带 **--version** 选项的命令行 TypeScript 编译器以确认是否已安装 TypeScript。（要开始新的终端会话，请在菜单栏上依次选择 Window (窗口) > New Terminal (新建终端)。）如果成功，输出将包含 TypeScript 版本号。如果已安装 TypeScript，请向前跳至 [步骤 2：添加代码](#)。

```
tsc --version
```

2. 通过运行带 **--version** 选项的 **npm** 以确认是否已安装 **npm**。如果成功，输出将包含 **npm** 版本号。如果已安装 **npm**，请向前跳到此过程中的步骤 10 来使用 **npm** 安装 TypeScript。

```
npm --version
```

3. 运行 **yum update** (适用于 Amazon Linux) 或 **apt update** (适用于 Ubuntu Server) 命令，以帮助确保已安装最新的安全更新和错误修复。

对于 Amazon Linux：

```
sudo yum -y update
```

对于 Ubuntu Server：

```
sudo apt update
```

4. 要安装 **npm**，请先运行以下命令以下载 Node Version Manager (**nvm**)。（**nvm** 是一个简单的 Bash Shell 脚本，用于安装和管理 Node.js 版本。有关更多信息，请参阅 GitHub 网站上的 [Node Version Manager](#)。）

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

5. 要开始使用 **nvm**，请关闭终端会话并重新启动，或者查找包含命令以加载 `~/.bashrcnvm` 的文件。

```
. ~/.bashrc
```

6. 运行带 **--version** 选项的 **nvm** 以确认已安装 **nvm**。

```
nvm --version
```

7. 运行 **nvm** 以安装最新版本的 Node.js 16。 (**npm** 包含在 Node.js 中。)

```
nvm install v16
```

8. 带 **--version** 选项运行命令行版本的 Node.js 以确认已安装 Node.js。

```
node --version
```

9. 运行带 **--version** 选项的 **npm** 以确认已安装 **npm**。

```
npm --version
```

10. 通过运行带 **-g** 选项的 **npm** 安装 TypeScript。这会在环境中作为全局程序包安装 TypeScript。

```
npm install -g typescript
```

11. 带 **--version** 选项运行命令行 TypeScript 编译器以确认已安装 TypeScript。

```
tsc --version
```

步骤 2：添加代码

1. 在 AWS Cloud9 IDE 中，创建名为 `hello.ts` 的文件。(要创建文件，请在菜单栏上依次选择 File (文件)、New File (新建文件)。要保存文件，请依次选择 File (文件)、Save (保存)。)
2. 在 IDE 的终端中，从 `hello.ts` 文件所在的相同目录中，运行 **npm** 以安装 `@types/node` 库。

```
npm install @types/node
```

这会在 `node_modules/@types/node` 文件所在的相同目录中添加 `hello.ts` 文件夹。此新文件夹包含 TypeScript 在此过程的后面需要的 Node.js 类型定义，用于您将添加到 `hello.ts` 文件中的 `console.log` 和 `process.argv` 属性。

3. 将以下代码添加到 `hello.ts` 文件：

```
console.log('Hello, World!');
```

```
console.log('The sum of 2 and 3 is 5.');
```

```
const sum: number = parseInt(process.argv[2], 10) + parseInt(process.argv[3], 10);
```

```
console.log('The sum of ' + process.argv[2] + ' and ' +  
  process.argv[3] + ' is ' + sum + '.');
```

步骤 3：运行代码

1. 在终端中，从 `hello.ts` 文件所在的相同目录中，运行 TypeScript 编译器。指定 `hello.ts` 文件以及要包括的其他库。

```
tsc hello.ts --lib es6
```

TypeScript 使用 `hello.ts` 文件和一组 ECMAScript 6 (ES6) 库文件将 `hello.ts` 文件中的 TypeScript 代码转换为名为 `hello.js` 的文件中的等效 JavaScript 代码。

2. 在 Environment (环境) 窗口中，打开 `hello.js` 文件。
3. 在菜单栏上依次选择 Run (运行) > Run Configurations (运行配置) > New Run Configuration (新建运行配置)。
4. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Node.js。
5. 在 Command (命令) 中，键入 `hello.js 5 9`。在代码中，5 表示 `process.argv[2]`，9 表示 `process.argv[3]`。(`process.argv[0]` 表示运行时的名称 (node)，`process.argv[1]` 表示文件的名称 (`hello.js`)。)
6. 选择 Run (运行)，然后比较输出。完成此操作后，选择 Stop (停止)。

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```




Note

不必在 IDE 中创建新的运行配置，您还可以从终端运行命令 **node hello.js 5 9** 以执行此代码。

步骤 4：安装并配置 AWS SDK for JavaScript in Node.js

您可以改进此示例以使用 AWS SDK for JavaScript in Node.js 创建 Amazon S3 存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。

在本步骤中，您将安装并配置 AWS SDK for JavaScript in Node.js。此开发工具包提供了从 JavaScript 代码与 AWS 服务（如 Amazon S3）交互的便捷方式。在安装 AWS SDK for JavaScript in Node.js 后，您必须在环境中设置凭证管理。开发工具包需要这些凭证才能与 AWS 服务交互。

安装 AWS SDK for JavaScript in Node.js

在 AWS Cloud9 IDE 的终端会话中，从 [步骤 3：运行代码](#) 中 `hello.js` 文件所在的相同目录中，运行 **npm** 以安装 AWS SDK for JavaScript in Node.js。

```
npm install aws-sdk
```

此命令从 [步骤 3：运行代码](#) 中向 `node_modules` 文件夹添加多个文件夹。这些文件夹包含 AWS SDK for JavaScript in Node.js 的源代码和依赖项。有关更多信息，请参阅 AWS SDK for JavaScript 开发人员指南中的 [安装 SDK for JavaScript](#)。

在环境中设置凭证管理

每次您使用 AWS SDK for JavaScript in Node.js 调用 AWS 服务时，都必须提供一组凭证才能进行调用。这些凭证确定 AWS SDK for JavaScript in Node.js 是否具有相应的权限以进行该调用。如果凭证没有包括合适的权限，调用将失败。

在此步骤中，您将凭证存储在环境内。为此，请按照 [从 AWS Cloud9 中的环境调用 AWS 服务](#) 中的说明操作，然后返回到本主题。

有关其他信息，请参阅 AWS SDK for JavaScript 开发人员指南中的 [在 Node.js 中设置凭证](#)。

步骤 5：添加 AWS 开发工具包代码

在此步骤中，您会添加一些代码，此次用于与 Amazon S3 交互来创建存储桶，列出可用的存储桶，然后删除刚刚创建的存储桶。您将稍后运行此代码。

1. 在 AWS Cloud9 IDE 中，在以前步骤中的 `hello.js` 文件所在的同一目录中，创建名为 `s3.ts` 的文件。
2. 从 AWS Cloud9 IDE 中的终端中，在 `s3.ts` 文件所在的同一目录中，启用代码以异步调用 Amazon S3 操作，方法是运行 `npm` 两次以安装适用于 TypeScript 的异步库，然后再次安装适用于 JavaScript 的异步库。

```
npm install @types/async # For TypeScript.
npm install async        # For JavaScript.
```

3. 将以下代码添加到 `s3.ts` 文件：

```
import * as async from 'async';
import * as AWS from 'aws-sdk';

if (process.argv.length < 4) {
  console.log('Usage: node s3.js <the bucket name> <the AWS Region to use>\n' +
    'Example: node s3.js my-test-bucket us-east-2');
  process.exit(1);
}

const AWS = require('aws-sdk'); // To set the AWS credentials and AWS Region.
const async = require('async'); // To call AWS operations asynchronously.

const s3: AWS.S3 = new AWS.S3({apiVersion: '2006-03-01'});
const bucket_name: string = process.argv[2];
const region: string = process.argv[3];
```

```
AWS.config.update({
  region: region
});

const create_bucket_params: any = {
  Bucket: bucket_name,
  CreateBucketConfiguration: {
    LocationConstraint: region
  }
};

const delete_bucket_params: any = {
  Bucket: bucket_name
};

// List all of your available buckets in this AWS Region.
function listMyBuckets(callback): void {
  s3.listBuckets(function(err, data) {
    if (err) {

    } else {
      console.log("My buckets now are:\n");

      for (let i: number = 0; i < data.Buckets.length; i++) {
        console.log(data.Buckets[i].Name);
      }
    }

    callback(err);
  });
}

// Create a bucket in this AWS Region.
function createMyBucket(callback): void {
  console.log("\nCreating a bucket named '" + bucket_name + "'...\n");

  s3.createBucket(create_bucket_params, function(err, data) {
    if (err) {
      console.log(err.code + ": " + err.message);
    }

    callback(err);
  });
};
```

```
}

// Delete the bucket you just created.
function deleteMyBucket(callback): void {
  console.log("\nDeleting the bucket named '" + bucket_name + "'...\n");

  s3.deleteBucket(delete_bucket_params, function(err, data) {
    if (err) {
      console.log(err.code + ": " + err.message);
    }

    callback(err);
  });
}

// Call the AWS operations in the following order.
async.series([
  listMyBuckets,
  createMyBucket,
  listMyBuckets,
  deleteMyBucket,
  listMyBuckets
]);
```

步骤 6：运行 AWS 开发工具包代码

1. 在终端中，从 `s3.ts` 文件所在的相同目录中，运行 TypeScript 编译器。指定 `s3.ts` 文件以及要包括的其他库。

```
tsc s3.ts --lib es6
```

TypeScript 使用 `s3.ts` 文件、AWS SDK for JavaScript in Node.js、异步库和一组 ECMAScript 6 (ES6) 库文件，将 `s3.ts` 文件中的 TypeScript 代码转换为名为 `s3.js` 的文件中的等效 JavaScript 代码。

2. 在 Environment (环境) 窗口中，打开 `s3.js` 文件。
3. 在菜单栏上依次选择 Run (运行) > Run Configurations (运行配置) > New Run Configuration (新建运行配置)。
4. 在 [New] - Idle ([新建] - 空闲) 选项卡上，选择 Runner: Auto (运行程序: 自动)，然后选择 Node.js。

- 对于 Command (命令), 请键入 `s3.js YOUR_BUCKET_NAME THE_AWS_REGION`, 其中 `YOUR_BUCKET_NAME` 是要创建并随后删除的存储桶的名称, `THE_AWS_REGION` 是要在其中创建存储桶的 AWS 区域的 ID。例如, 对于美国东部 (俄亥俄) 区域, 请使用 `us-east-2`。有关更多 ID, 请参阅 [中的](#) Amazon Simple Storage Service (Amazon S3) Amazon Web Services 一般参考。

 Note

Amazon S3 存储桶名称在整个 AWS 中都必须唯一的, 而不仅仅在您的 AWS 账户中唯一。

- 选择 Run (运行), 然后比较输出。完成此操作后, 选择 Stop (停止)。

```
My buckets now are:

Creating a new bucket named 'my-test-bucket'...

My buckets now are:

my-test-bucket

Deleting the bucket named 'my-test-bucket'...

My buckets now are:
```

步骤 7：清除

为防止在使用完该示例后一直对您的 AWS 账户收费, 应删除环境。有关说明, 请参阅 [在 AWS Cloud9 中删除环境](#)。

适用于 Docker 的教程 AWS Cloud9

本教程向您展示如何将 AWS Cloud9 SSH 开发环境连接到亚马逊 EC2 中亚马逊 Linux 实例中正在运行的 Docker 容器。这使您可以使用 AWS Cloud9 IDE 处理 Docker 容器内的代码和文件, 并在该容器上运行命令。有关 Docker 的信息, 请参阅 Docker 网站上的 [Docker 是什么](#)。

按照本教程并创建此示例可能会导致您的 AWS 账户被扣款。其中包括可能对 Amazon EC2 等服务收取的费用。有关更多信息, 请参阅 [Amazon EC2 定价](#)。

主题

- [先决条件](#)
- [步骤 1：安装并运行 Docker](#)
- [步骤 2：构建镜像](#)
- [步骤 3：运行容器](#)
- [步骤 4：创建环境](#)
- [步骤 5：运行代码](#)
- [步骤 6：清理](#)

先决条件

- 您应该有一个运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例。此示例假设您的账户中已经有一个运行亚马逊 Linux 或 Ubuntu 服务器的 Amazon EC2 实例。AWS 要启动 Amazon EC2 实例，请参阅[启动 Linux 虚拟机](#)。在向导的 Choose an Amazon Machine Image (AMI) (选择 Amazon Machine Image (AMI)) 页面中，选择其显示名称以 Amazon Linux AMI 或 Ubuntu Server 开头的 AMI。
- 如果 Amazon EC2 实例在 Amazon VPC 中运行，则存在额外的要求。请参阅[AWS Cloud9 开发环境的 VPC 设置](#)。
- Amazon EC2 实例应至少具有 8 到 16 GB 的可用磁盘空间。此示例使用 3 GB 以上的 Docker 映像，并且可使用 3 GB 或更多磁盘空间的额外增量来构建映像。如果您尝试在可用空间为 8 GB 或更少的磁盘上运行此示例，我们发现，Docker 映像可能不会构建或 Docker 容器可能不会运行。要检查实例的可用磁盘空间，可以在实例上运行诸如 `df -h` 这样的命令（针对“用户可读格式的磁盘文件系统信息”）。要增加现有实例的磁盘大小，请参阅 Amazon EC2 用户指南中的[修改卷](#)。

步骤 1：安装并运行 Docker

在此步骤中，检查 Docker 是否安装在 Amazon EC2 实例上，如果未安装 Docker，则请安装它。安装 Docker 后，在实例上运行它。

1. 通过使用 SSH 客户端（如 `ssh` 实用工具或 PuTTY）连接到正在运行的 Amazon EC2 实例。为此，请参阅[启动 Linux 虚拟机](#)中的“步骤 3：连接到您的实例”。
2. 检查是否在实例上安装了 Docker。为此，请在实例上将 `docker` 命令与 `--version` 选项一起运行。

```
docker --version
```

如果已安装 Docker，则将显示 Docker 版本和版本号。在此情况下，请向前跳至此过程中后面的步骤 5。

3. 安装 Docker。为此，请将 **yum** 命令或 **apt** 命令与 **install** 操作一起运行，并指定要安装的 **docker** 或 **docker.io** 程序包。

对于 Amazon Linux：

```
sudo yum install -y docker
```

对于 Ubuntu Server：

```
sudo apt install -y docker.io
```

4. 确认已安装 Docker。为此，请再次运行 **docker --version** 命令。将显示 Docker 版本和版本号。
5. 运行 Docker。为此，请将 **service** 命令与 **docker** 服务和 **start** 操作一起运行。

```
sudo service docker start
```

6. 确认 Docker 正在运行。为此，请将 **docker** 命令与 **info** 操作一起运行。

```
sudo docker info
```

如果 Docker 正在运行，则将显示有关 Docker 的信息。

步骤 2：构建镜像

在此步骤中，您使用 Dockerfile 在实例上构建一个 Docker 映像。此示例使用包括 Node.js 和示例聊天服务器应用程序的映像。

1. 在实例上，创建 Dockerfile。为此，在 SSH 客户端仍连接到实例的情况下，在实例上的 `/tmp` 目录中，创建一个名为 Dockerfile 的文件。例如，运行 **touch** 命令，如下所示。

```
sudo touch /tmp/Dockerfile
```

2. 将以下内容添加到 Dockerfile 文件。

```
# Build a Docker image based on the Amazon Linux 2 Docker image.
FROM amazonlinux:2

# install common tools
RUN yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
RUN yum update -y
RUN yum install -y sudo bash curl wget git man-db nano vim bash-completion tmux
gcc gcc-c++ make tar

# Enable the Docker container to communicate with AWS Cloud9 by
# installing SSH.
RUN yum install -y openssh-server

# Ensure that Node.js is installed.
RUN yum install -y nodejs

# Create user and enable root access
RUN useradd --uid 1000 --shell /bin/bash -m --home-dir /home/ubuntu ubuntu && \
    sed -i 's/%wheel\s.*\s/%wheel ALL=NOPASSWD:ALL/' /etc/sudoers && \
    usermod -a -G wheel ubuntu

# Add the AWS Cloud9 SSH public key to the Docker container.
# This assumes a file named authorized_keys containing the
# AWS Cloud9 SSH public key already exists in the same
# directory as the Dockerfile.
RUN mkdir -p /home/ubuntu/.ssh
ADD ./authorized_keys /home/ubuntu/.ssh/authorized_keys
RUN chown -R ubuntu /home/ubuntu/.ssh /home/ubuntu/.ssh/authorized_keys && \
    chmod 700 /home/ubuntu/.ssh && \
    chmod 600 /home/ubuntu/.ssh/authorized_keys

# Update the password to a random one for the user ubuntu.
RUN echo "ubuntu:$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1)"
    | chpasswd

# pre-install Cloud9 dependencies
USER ubuntu
RUN curl https://d2j6vhu5uywtq3.cloudfront.net/static/c9-install.sh | bash
```



```
USER root
# Start SSH in the Docker container.
CMD ssh-keygen -A && /usr/sbin/sshd -D
```

要将上述内容添加到 Dockerfile 文件，您可以在实例上使用 **vi** 实用工具，如下所示。

- a. 使用 AWS Cloud9 打开和编辑/tmp/Dockerfile文件。

```
sudo vi /tmp/Dockerfile
```

- b. 将上述内容粘贴到 Dockerfile 文件。如果不确定如何执行此操作，请参阅 SSH 客户端文档。
- c. 切换到命令模式。要执行此操作，请按 Esc 键。（-- INSERT -- 将从窗口底部消失。）
- d. 键入 :wq（以写入 /tmp/Dockerfile 文件，保存此文件，然后退出 **vi**），然后按 Enter。

Note

您可以从 AWS CodeBuild 中访问经常更新的 Docker 镜像列表。有关更多信息，请参阅《AWS CodeBuild 用户指南》CodeBuild 中提供的 [Docker 镜像](#)。

3. 在实例上，创建一个包含供 Docker 容器使用的 AWS Cloud9 SSH 公钥的文件。为此，请在 Dockerfile 文件所在的目录中，创建一个名为 authorized_keys 的文件，例如，通过运行 **touch** 命令。

```
sudo touch /tmp/authorized_keys
```

4. 将 AWS Cloud9 SSH 公钥添加到 authorized_keys 文件中。要获取 AWS Cloud9 SSH 公钥，请执行以下操作：
 - a. 打开 AWS Cloud9 控制台，[网址为 https://console.aws.amazon.com/cloud9/](https://console.aws.amazon.com/cloud9/)。
 - b. 在 AWS 导航栏的 AWS 区域选择器中，选择您要在本主题后面创建 AWS Cloud9 开发环境的 AWS 区域。
 - c. 如果显示欢迎页面，请在“新建 AWS Cloud9 环境”中选择“创建环境”。如未显示，则选择 Create environment（创建环境）。
 - d. 在 Name environment（命名环境）页面上，为 Name（名称）键入环境的名称。（在这里键入的名称不重要。稍后您将选择其他名称。）

- e. 选择下一步。
 - f. 对于 Environment type (环境类型) , 请选择 Connect and run in remote server (SSH) (连接并在远程服务器中运行 (SSH)) 。
 - g. 展开 View public SSH key (查看公有 SSH 密钥)。
 - h. 选择 Copy key to clipboard (将密钥复制到剪贴板) 。 (这位于 View public SSH key (查看公有 SSH 密钥) 和 Advanced settings (高级设置) 之间。)
 - i. 选择取消。
 - j. 将剪贴板的内容粘贴到 authorized_keys 文件中 , 然后保存此文件。例如 , 可以使用 **vi** 实用工具 , 如此步骤前面所述。
5. 通过将 **docker** 命令与 **build** 操作一起运行 , 将标签 cloud9-image:latest 添加到镜像并指定要使用的 Dockerfile 文件的路径来构建镜像。

```
sudo docker build -t cloud9-image:latest /tmp
```

如果成功 , 生成输出的最后两行将显示 Successfully built 和 Successfully tagged。

要确认 Docker 已成功构建映像 , 请将 **docker** 命令与 image ls 操作一起运行。

```
sudo docker image ls
```

如果成功 , 输出将显示一个条目 (其中 , REPOSITORY 字段设置为 cloud9-image , TAG 字段设置为 latest) 。

6. 记下 Amazon EC2 实例的公有 IP 地址。您将在[步骤 4 : 创建环境](#)中需要它。如果您不确定实例的公有 IP 地址 , 则可在实例上运行以下命令来获取此地址。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

步骤 3 : 运行容器

在此步骤中 , 在实例上运行 Docker 容器。此容器基于在上一步中构建的映像。

1. 要运行 Docker 容器 , 请在实例上将 **docker** 命令与 **run** 操作和以下选项一起运行。

```
sudo docker run -d -it --expose 9090 -p 0.0.0.0:9090:22 --name cloud9 cloud9-image:latest
```

- `-d` 在已分离模式下运行容器，以便在用于运行容器的根进程（此示例中为 SSH 客户端）退出时退出。
- `-it` 运行带分配的伪 TTY 的容器并使 STDIN 保持打开状态，即使未附加容器也是如此。
- `--expose` 使指定端口（此示例中为端口 9090）可通过容器使用。
- `-p` 使指定端口在内部通过指定 IP 地址和端口可供 Amazon EC2 实例使用。在此示例中，可通过 Amazon EC2 实例上的端口 22 内部访问容器上的端口 9090。
- `--name` 是容器的用户可读名称（此示例中为 `cloud9`）。
- `cloud9-image:latest` 是要用于运行容器的已构建映像的用户可读名称。

要确认 Docker 正在成功运行容器，请将 **docker** 命令与 `container ls` 操作一起运行。

```
sudo docker container ls
```

如果成功，输出将显示一个条目（其中，`IMAGE` 字段设置为 `cloud9-image:latest`，`NAMES` 字段设置为 `cloud9`）。

2. 登录运行的容器。为此，请将 **docker** 命令与 **exec** 操作和以下选项一起运行。

```
sudo docker exec -it cloud9 bash
```

- `-it` 运行带分配的伪 TTY 的容器并使 STDIN 保持打开状态，即使未附加容器也是如此。
- `cloud9` 是运行的容器的用户可读名称。
- `bash` 在运行的容器中启动标准 Shell。

如果成功，则终端提示符将更改为显示登录用户的容器名称和容器 ID。

Note

如果您需要从运行的容器注销，请运行 **exit** 命令。终端提示符将更改回显示登录用户的实例名称和实例的私有 DNS。容器应仍在运行中。

3. 对于正在运行的容器上 AWS Cloud9 要在登录后启动的目录，请将其访问权限设置为 **rwxr-xr-x**。这意味着所有者的 read-write-execute 权限、群组的读取执行权限以及其他人的读取执行权限。例如，如果目录的路径为 `~`，则可以通过在运行的容器中运行 **chmod** 命令来设置对应目录的这些权限，如下所示。

```
sudo chmod u=rwx,g=rx,o=rx ~
```

- 记下运行的容器上包含 Node.js 二进制文件的目录的路径，因为您在[步骤 4：创建环境](#)中将需要它。如果您不确定此路径，请在运行的容器上运行以下命令来获取它。

```
which node
```

步骤 4：创建环境

在此步骤中，您将使用 AWS Cloud9 创建 AWS Cloud9 SSH 开发环境并将其连接到正在运行的 Docker 容器。AWS Cloud9 创建环境后，它会显示 AWS Cloud9 IDE，以便您可以开始使用容器中的文件和代码。

您可以使用 AWS Cloud9 控制台创建 AWS Cloud9 SSH 开发环境。您无法使用 CLI 创建 SSH 环境。

先决条件

- 首先确保您已完成 [设置 AWS Cloud9](#) 中的步骤。这样，您便可以登录到 AWS Cloud9 控制台并创建环境。
- 确定要连接到环境的现有云计算实例（例如，您的中的 Amazon EC2 实例 AWS 账户）或您 AWS Cloud9 自己的服务器。
- 确保现有实例或您自己的服务器满足所有 [SSH 托管要求](#)。这包括安装特定版本的 Python、Node.js 和其他组件；对于您希望的登录后 AWS Cloud9 开始目录设置特定权限；以及设置任何关联的 Amazon Virtual Private Cloud。

创建 SSH 环境

- 确保您已完成上述先决条件。
- 使用 SSH 客户端连接到现有实例或您自己的服务器（如果您尚未连接）。这确保了您可以向实例或服务器添加必要的公有 SSH 密钥值。这些内容将在此过程的稍后介绍。

Note

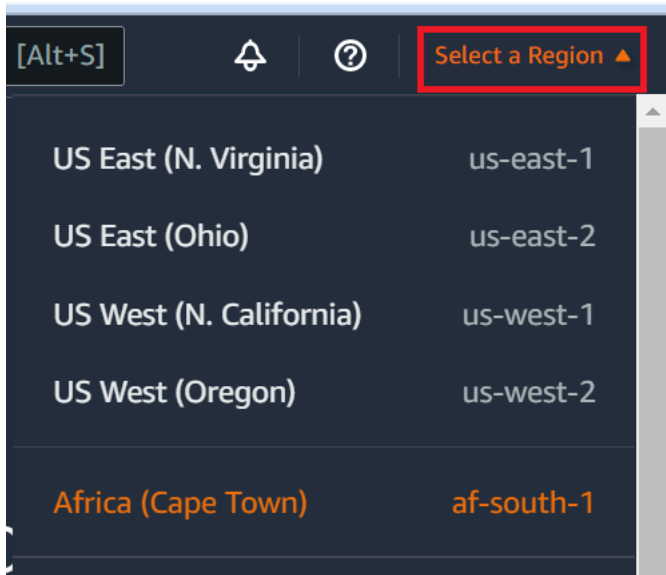
要连接到现有的 AWS Cloud 计算实例，请参阅以下一项或多项资源：

- 对于亚马逊 EC2，请参阅《亚马逊 EC2 用户指南》中的“[连接到您的 Linux 实例](#)”。

- 有关 Amazon Lightsail，请参阅 [Amazon Lightsail 文档](#) 中的连接到基于 Linux/Unix 的 Lightsail 实例。
- 有关信息 AWS Elastic Beanstalk，请参阅《AWS Elastic Beanstalk 开发人员指南》中的 [列出和连接到服务器实例](#)。
- 有关信息 AWS OpsWorks，请参阅《AWS OpsWorks 用户指南》中的 [“使用 SSH 登录 Linux 实例”](#)。
- 有关其他 AWS 服务，请参阅该特定服务的文档。

要连接到您自己的服务器，请使用 SSH。已经在 macOS 和 Linux 操作系统上安装了 SSH。要在 Windows 上使用 SSH 连接到服务器，您必须安装 [PuTTY](#)。

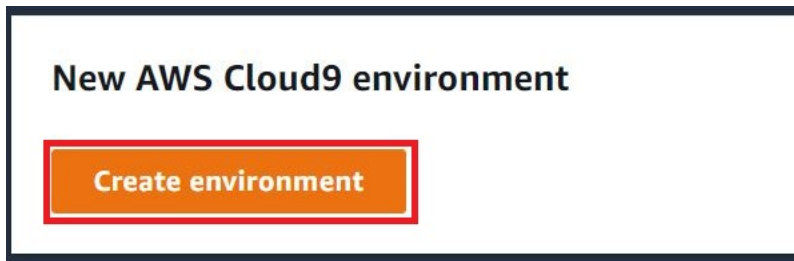
3. 登录 AWS Cloud9 主机，[网址为 https://console.aws.amazon.com/cloud9/](https://console.aws.amazon.com/cloud9/)。
4. 登录 AWS Cloud9 控制台后，在顶部导航栏中选择一个 AWS 区域 来创建环境。有关可用列表 AWS 区域，请参阅[AWS Cloud9](#)中的 AWS 一般参考。



5. 如果您是第一次创建开发环境，则将显示欢迎页面。在“新建 AWS Cloud9 环境”面板中，选择“创建环境”。

如果您之前创建过开发环境，您也可以展开屏幕左侧的窗格。选择 Your environments (您的环境)，然后选择 Create environment (创建环境)。

在欢迎页面：



或在 Your environment (您的环境) 页面内 :



6. 在 Create environment (创建环境) 页面上，输入环境的名称。
7. 在 Description (描述) 中输入对环境的描述。在本教程中，请使用 `This environment is for the AWS Cloud9 tutorial.`
8. 对于 Environment type (环境类型)，请从以下选项中选择 Existing Compute (现有计算)：
 - 新的 EC2 实例 — 启动 AWS Cloud9 可以直接通过 SSH 连接的 Amazon EC2 实例。
 - 现有计算-启动不需要任何开放入站端口的 Amazon EC2 实例。AWS Cloud9 通过连接到实例 [AWS Systems Manager](#)。
 - 如果您选择 Existing compute (现有计算) 选项，将创建服务角色和 IAM 实例配置文件，以允许 Systems Manager 代表您与 EC2 实例进行交互。您可以在界面下方的 Systems Manager 访问的服务角色和实例配置文件部分查看两者的名称。有关更多信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

⚠ Warning

为您的环境创建 EC2 实例可能会导致您 AWS 账户收取 Amazon EC2 费用。使用 Systems Manager 管理与 EC2 实例的连接不会产生额外费用。

⚠ Warning

AWS Cloud9 使用 SSH 公钥安全地连接到您的服务器。要建立安全连接，请将我们的公有密钥添加到您的 `~/.ssh/authorized_keys` 文件中，并在以下步骤中提供您的登

录凭证。选择 Copy key to clipboard (将密钥复制到剪贴板) 以复制 SSH 密钥，或选择 View public SSH key (查看 SSH 公有密钥) 以查看密钥。

9. 在 Existing compute (现有计算) 面板上，对于 User (用户)，请输入您在此过程中之前连接到实例或服务器时使用的登录名。例如，对于 AWS Cloud 计算实例，该登录名可能为 ec2-user、ubuntu 或 root。

Note

我们建议将登录名与实例或服务器上的管理权限或管理员用户关联。更具体地说，我们建议此登录名应在实例或服务器上拥有 Node.js 安装。要进行此检查，请在实例或服务器的终端上运行命令 `ls -l $(which node)` (如果您使用的是 nvm，请运行 `ls -l $(nvm which node)`)。此命令会显示 Node.js 安装的拥有者名称。它还会显示安装的权限、组名称和位置。

10. 对于 Host (主机)，请输入公有 IP 地址 (首选) 或者实例或服务器的主机名。
11. 在“端口”中，输入您 AWS Cloud9 要用来尝试连接到实例或服务器的端口。您也可保留默认端口。
12. 选择 Additional details - optional (其他详细信息 – 可选) 以显示环境路径、node.js 二进制文件路径和 SSH 跳转主机信息。
13. 在环境路径中，输入您要 AWS Cloud9 从中启动的实例或服务器的目录路径。您之前在此步骤的先决条件中确定了这一点。如果将此项留空，AWS Cloud9 将使用实例或服务器在登录之后通常开始使用的目录。这通常为主目录或默认目录。
14. 对于 Path to Node.js binary path (Node.js 二进制文件路径)，请输入路径信息以指定实例或服务器上指向 Node.js 二进制文件的路径。要获取路径，您可以在实例或服务器上运行命令 `which node` (在使用 nvm 时为 `nvm which node`)。例如，路径可能为 `/usr/bin/node`。如果将此项留空，AWS Cloud9 在尝试连接时，将尝试猜测 Node.js 二进制文件所在的位置。
15. 对于 SSH jump host (SSH 跳转主机)，输入有关实例或服务器使用的跳转主机的信息。使用格式 `USER_NAME@HOSTNAME:PORT_NUMBER` (例如 `ec2-user@ip-192-0-2-0:22`)。

跳转主机必须满足以下要求：

- 它必须可使用 SSH 通过公有互联网访问。
- 它必须允许任意 IP 地址通过指定端口的入站访问。
- 复制到现有实例或服务器上 `~/.ssh/authorized_keys` 文件中的公有 SSH 密钥值还必须复制到跳转主机上的 `~/.ssh/authorized_keys` 文件中。

- 必须安装有 Netcat。
16. 通过为每个标签提供键和值，添加最多 50 个标签。要执行此操作，请选择 Add new tag (添加新标签)。这些标签作为资源标签附加到 AWS Cloud9 环境，并传播到以下基础资源：AWS CloudFormation 堆栈、Amazon EC2 实例和 Amazon EC2 安全组。要了解有关标签的更多信息，请参阅 [IAM 用户指南中的使用 AWS 资源标签控制访问权限以及本指南中有关标签的高级信息](#)。

Warning

如果您在创建标签后更新这些标签，则更改不会传播到底层资源。有关更多信息，请参阅 [标签高级信息中的将标签更新传播到底层资源](#)。

17. 选择 Create (创建) 以创建您的环境，然后系统会将您重定向到主页。成功创建帐户后，AWS Cloud9 控制台顶部会出现一个绿色的闪光条。您可以选择新环境并选择 Open in Cloud9 (在 Cloud9 中打开) 以启动 IDE。



如果未能创建账户，则 AWS Cloud9 控制台顶部会出现一个红色的闪光条。由于您的网络浏览器、AWS 访问权限、实例或关联网络存在问题，您的账户可能无法创建。您可以在 [AWS Cloud9 故障排除](#) 部分找到有关可能修复导致账户失败问题的信息。

Note

如果您的环境使用代理来访问 Internet，则必须向其提供代理详细信息，AWS Cloud9 这样它才能安装依赖关系。有关更多信息，请参阅 [无法安装依赖项](#)。

步骤 5：运行代码

在此步骤中，您将使用 AWS Cloud9 IDE 在正在运行的 Docker 容器中运行示例应用程序。

1. 显示正在运行的容器的 AWS Cloud9 IDE 后，启动示例聊天服务器。为此，请在 Environment (环境) 窗口中，右键单击示例 workspace/server.js 文件，然后选择 Run (运行)。
2. 预览示例应用程序。为此，请在 Environment (环境) 窗口中，打开 workspace/client/index.html 文件。然后，在菜单栏上，依次选择 Tools (工具)、Preview (预览)、Preview Running Application (预览运行的应用程序)。

3. 在应用程序预览标签页上，对于 Your Name (您的姓名)，键入您的姓名。对于 Message (消息)，请输入消息。然后选择 Send (发送)。聊天服务器会将您的姓名和消息添加到列表中。

步骤 6：清理

在此步骤中，您将删除环境并从 Amazon EC2 实例中移除 AWS Cloud9 和 Docker 支持文件。此外，为了防止在您使用完此示例后继续向您的 AWS 账户收费，您应该终止运行 Docker 的 Amazon EC2 实例。

步骤 6.1：删除环境

要删除环境，请参阅 [在 AWS Cloud9 中删除环境](#)。

步骤 6.2：删除容器中的 AWS Cloud9 支持文件

删除环境后，一些 AWS Cloud9 支持文件仍保留在容器中。如果您想继续使用容器但不再需要这些支持文件，请在容器登录后从您指定的 AWS Cloud9 目录中删除该文件 .c9 夹。例如，如果目录为 ~，请将 **rm** 命令与 **-r** 选项一起运行，如下所示。

```
sudo rm -r ~/.c9
```

步骤 6.3：删除实例中的 Docker 支持文件

如果您不想再在 Amazon EC2 实例上保留 Docker 容器、Docker 镜像和 Docker，但想要保留此实例，则可删除这些 Docker 支持文件，如下所示。

1. 从实例中删除 Docker 容器。为此，请在实例上将 **docker** 命令与 **stop** 和 **rm** 停止操作以及容器的用户可读名称一起运行。

```
sudo docker stop cloud9
sudo docker rm cloud9
```

2. 从实例中删除 Docker 映像。为此，请在实例上将 **docker** 命令与 **image rm** 操作和映像的标签一起运行。

```
sudo docker image rm cloud9-image:latest
```

3. 删除可能仍存在的任何其他 Docker 支持文件。为此，请在实例上将 **docker** 命令与 **system prune** 操作一起运行。

```
sudo docker system prune -a
```

4. 卸载 Docker。为此，请在实例上将 **yum** 命令与 **remove** 操作一起运行，以指定要卸载的 **docker** 程序包。

对于 Amazon Linux：

```
sudo yum -y remove docker
```

对于 Ubuntu Server：

```
sudo apt -y remove docker
```

还可以删除之前创建的 Dockerfile 和 authorized_keys 文件。例如，在实例上运行 **rm** 命令。

```
sudo rm /tmp/Dockerfile
sudo rm /tmp/authorized_keys
```

步骤 6.4：终止实例

要终止 Amazon EC2 实例，请参阅 Amazon EC2 用户指南中的[终止您的实例](#)。

相关教程

- AWS RoboMaker 开发人员指南中的[开始使用 AWS RoboMaker](#)。此教程使用 AWS Cloud9 修改、生成和捆绑示例机器人应用程序。

AWS Cloud9 的高级主题

这些主题包含以下信息：

- 用于高级配置和决策的信息。
- 与特定任务相关并有助于您更好地了解 AWS Cloud9 但对完成该任务并不关键的信息。

主题

- [AWS Cloud9 中 EC2 环境与 SSH 环境的比较](#)
- [AWS Cloud9 开发环境的 VPC 设置](#)
- [SSH 环境主机要求](#)
- [在 AWS Cloud9 SSH 环境中使用 AWS Cloud9 安装程序](#)
- [AWS Cloud9 的入站 SSH IP 地址范围](#)
- [AWS Cloud9 EC2 开发环境的 Amazon Machine Image \(AMI\) 内容](#)
- [将服务相关角色用于 AWS Cloud9](#)
- [使用 AWS Cloud9 记录 AWS CloudTrail API 调用](#)
- [Tags](#)

AWS Cloud9 中 EC2 环境与 SSH 环境的比较

如[环境和计算资源简介](#)以及[使用环境](#)中所述，您的 AWS Cloud9 环境可以设置为 EC2 环境或 SSH 环境。

下表突出显示了在 AWS Cloud9 中使用 EC2 环境和 SSH 环境之间的相似之处和差异。

EC2 环境	SSH 环境
AWS Cloud9 创建关联的 Amazon EC2 实例并管理该实例的生命周期。这包括启动、停止和终止操作。	使用现有云计算实例或您自己的服务器。您负责管理其生命周期。
实例在 Amazon Linux 或 Ubuntu Server 上运行。	您可以使用运行 Linux 的任何云计算实例或您自己的运行 Linux 的服务器。

EC2 环境	SSH 环境
AWS Cloud9 自动设置实例以开始使用 AWS Cloud9。	必须手动配置实例或您自己的服务器才能使用 AWS Cloud9。
AWS Cloud9 自动在实例上设置 AWS Command Line Interface (AWS CLI)。	如果希望在实例或您自己的服务器上使用 AWS CLI，必须负责自行对其进行设置。
该实例可访问数百个有用的程序包，并且已安装并配置一些常用程序包。示例包括 Git、Docker、Node.js 和 Python。	您可能需要下载、安装并配置其他程序包才能完成常见任务。
您需要维护实例，例如，定期应用系统更新。	您需要维护实例或您自己的服务器。
在删除环境时，AWS Cloud9 自动终止关联的实例。	在删除环境时，实例或您自己的服务器保持启动状态。
<p>AWS 托管式临时凭证 可用于 EC2 环境中。使用这些凭证，您可以在某些限制下，为调用方的 AWS 账户中的所有 AWS 资源开启或关闭所有 AWS 操作。您无需为环境的 Amazon EC2 实例配置实例配置文件，也无需存储 AWS 实体（例如 IAM 用户）的永久 AWS 访问凭证。目前，如果环境的 Amazon EC2 实例启动到私有子网，则无法使用 AWS 托管式临时凭证来允许 Amazon EC2 环境代表 AWS 实体（例如 IAM 用户）访问 AWS 服务。</p>	<p>AWS 托管式临时凭证 在 SSH 环境中不可用。您必须使用 AWS Identity and Access Management 管理允许您使用 AWS Cloud9 以及其他 AWS 服务和资源的权限。</p>
<p>AWS Toolkit、Git 面板 和 增强版的 Java 技术支持 可供使用。</p>	<p>AWS Toolkit、Git 面板和增强版的 Java 技术支持不可用。</p>

AWS Cloud9 开发环境的 VPC 设置

与亚马逊虚拟私有云（亚马逊 VPC）关联的每个 AWS Cloud9 开发环境都必须满足特定的 VPC 要求。这些环境包括 EC2 环境以及与 VPC 内运行的 AWS Cloud 计算实例关联的 SSH 环境。示例包括 Amazon EC2 和 Amazon Lightsail 实例。

亚马逊 VPC 对以下内容的要求 AWS Cloud9

AWS Cloud9 使用的亚马逊 VPC 需要以下设置。如果您已熟悉这些要求，并且只想创建一个兼容的 VPC，则向前跳至 [创建 VPC 以及其他 VPC 资源](#)。

使用以下核对清单来确认 VPC 满足以下所有要求：

- VPC 可以 AWS 账户与 AWS 区域 AWS Cloud9 开发环境位于同一环境中，也可以是 VPC 与环境 AWS 账户不同的共享 VPC。但是，VPC 必须与环境位于 AWS 区域同一环境中。有关的 Amazon VPC 的更多信息 AWS 区域，请参阅[查看 AWS 区域的 VPC 列表](#)。有关为创建 Amazon VPC 的更多说明 AWS Cloud9，请参阅[创建 VPC 以及其他 VPC 资源](#)。有关使用共享 Amazon VPC 的更多信息，请参阅《Amazon VPC 用户指南》中的[使用共享 VPC](#)。
- VPC 必须具有公有子网。如果子网的流量路由到互联网网关，则该子网是公有子网。有关 Amazon VPC 的子网列表，请参阅[查看 VPC 的子网列表](#)。
- 如果您的环境通过 SSH 直接访问其 EC2 实例，则该实例只能在公有子网中启动。有关确认子网是否为公有子网的信息，请参阅[确认子网是否是公有子网](#)。
- 如果您使用 Systems Manager 访问[非入口 Amazon EC2 实例](#)，则可以在公有子网或私有子网中启动实例。
- 如果您使用的是公有子网，请将互联网网关附加到 VPC。这样，实例的 AWS Systems Manager Agent (SSM Agent) 就可以连接到 Systems Manager。
- 如果您使用的是私有子网，请允许子网的实例通过在公有子网中托管 NAT 网关与互联网通信。有关查看或更改互联网网关设置的更多信息，请参阅[查看或更改互联网网关的设置](#)。
- 公有子网必须具有包含一组最少路由的路由表。要了解如何确认子网是否具有路由表，请参阅[确认子网是否具有路由表](#)。有关如何创建路由表的信息，请参阅[创建路由表](#)。
- VPC (或 AWS Cloud 计算实例，取决于您的架构) 的关联安全组必须允许最少的入站和出站流量。有关适用于 Amazon VPC 的安全组列表，请参阅[查看 VPC 的安全组列表](#)。有关在 Amazon VPC 中创建安全组的更多信息，请参阅[在 VPC 中创建安全组](#)。
- 对于附加的安全层，如果 VPC 具有网络 ACL，则网络 ACL 必须允许至少一组入站和出站流量。要确认 Amazon VPC 是否具有至少一个网络 ACL，请参阅[确认 VPC 是否具有至少一个网络 ACL](#)。有关创建网络 ACL 的信息，请参阅[创建网络 ACL](#)。
- 如果您的开发环境[使用 SSM 访问 EC2 实例](#)，请确保该实例被启动到的公有子网分配了公有 IP 地址。为此，您必须为公有子网启用自动分配公有 IP 地址选项，并将其设置为 Yes。您可以在子网设置页面中创建 AWS Cloud9 环境之前，在公有子网上启用此选项。有关修改公有子网中自动分配 IP 设置的步骤，请参阅《Amazon VPC 用户指南》中的[修改子网的公有 IPv4 寻址属性](#)。有关配置共有子网和私有子网的更多信息，请参阅[将子网配置为公有子网或私有子网](#)。

Note

要完成以下过程，请登录 AWS Management Console 并使用管理员凭证打开 Amazon VPC 控制台 (<https://console.aws.amazon.com/vpc>) 或亚马逊 EC2 控制台 (<https://console.aws.amazon.com/ec2>)。

如果您使用 AWS CLI 或 AWS CloudShell，我们建议您在其中使用管理员凭据配置 AWS CLI 或 AWS 账户。AWS CloudShell 如果您无法执行此操作，请咨询您的 AWS 账户 管理员。

查看 AWS 区域的 VPC 列表

要使用 Amazon VPC 控制台，请在 AWS 导航栏中选择 AWS Cloud9 用于创建环境的。AWS 区域 然后，在导航窗格中选择 Your VPCs (您的 VPC)。

例如，要使用 AWS CLI 或 AWS CloudShell，请运行 Amazon EC2 **describe-vpcs** 命令，如下所示。

```
aws ec2 describe-vpcs --output table --query 'Vpcs[*].VpcId' --region us-east-2
```

在前面的命令中，us-east-2 替换为 AWS 区域 在中 AWS Cloud9 创建环境的。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 aws-shell 一起运行，请忽略 aws。

输出包含 VPC ID 的列表。

查看 VPC 的子网列表

要使用 Amazon VPC 控制台，请选择导航窗格中的 Your VPCs (您的 VPC)。记下 VPC ID 列中的 VPC ID。然后选择导航窗格中的 Subnets (子网)，并在 VPC 列中查找包含该 ID 的子网。

例如，要使用 AWS CLI 或 aws-shell，请运行 Amazon EC2 **describe-subnets** 命令，如下所示。

```
aws ec2 describe-subnets --output table --query 'Subnets[*].[SubnetId,VpcId]' --region us-east-2
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含子网的。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 aws-shell 一起运行，请忽略 aws。

在输出中，查找与 VPC ID 匹配的子网。

确认子网是否是公有子网

Important

假设您正在将环境的 EC2 实例启动到私有子网中。确保允许该实例的出站流量，以便它能够连接到 SSM 服务。对于私有子网，出站流量通常通过网络地址转换 (NAT) 网关或 VPC 终端节点进行配置。(NAT 网关需要公有子网)。

假设您选择 VPC 端点 (而不是 NAT 网关) 来访问 SSM。如果实例的自动更新和安全补丁依赖于互联网访问，则它们可能无法正常工作。您可以使用其他应用程序 (例如 S [AWS systems Manager Patch Manager](#)) 来管理您的环境可能需要的任何软件更新。AWS Cloud9 软件将照常更新。

要使用 Amazon VPC 控制台，请选择导航窗格中的 Subnets (子网)。选中要使用的子网旁边的 AWS Cloud9 复选框。在 Route Table (路由表) 选项卡上，如果在 Target (目标) 列中有一个以 igw- 开头的条目，则该子网是公有子网。

要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **describe-route-tables** 命令。

```
aws ec2 describe-route-tables --output table --query 'RouteTables[*].Routes[*].
{GatewayIds:GatewayId}' --region us-east-2 --filters Name=association.subnet-
id,Values=subnet-12a3456b
```

在前面的命令中，`us-east-2` 替换为 AWS 区域 包含子网的，然后 `subnet-12a3456b` 替换为子网 ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

在输出中，如果至少有一个结果以 `igw-` 开头，则子网为公有子网。

在输出中，如果没有结果，则路由表可能与 VPC 而不是子网关联。要确认这一点，请针对 VPC 的相关子网而不是子网本身运行 Amazon EC2 **describe-route-tables** 命令，如下所示。

```
aws ec2 describe-route-tables --output table --query 'RouteTables[*].Routes[*].
{GatewayIds:GatewayId}' --region us-east-1 --filters Name=vpc-id,Values=vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包 AWS 区域 含 VPC 的，然后 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

在输出中，如果至少有一个结果以 `igw-` 开头，则 VPC 包含互联网网关。

查看或更改互联网网关的设置

要使用 Amazon VPC 控制台，请选择导航窗格中的 Internet Gateways (互联网网关)。选中 Internet 网关旁边的框。要查看这些设置，请查看每个选项卡。要更改某个选项卡上的设置，请选择 Edit (编辑) (如果适用)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或查看设置，请运行 Amazon EC2 **describe-internet-gateways** 命令。aws-shell

```
aws ec2 describe-internet-gateways --output table --region us-east-2 --internet-gateway-id igw-1234ab5c
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含互联网网关的，并 igw-1234ab5c 替换为互联网网关 ID。要将上述命令与 aws-shell 一起运行，请忽略 aws。

创建 Internet 网关

要使用 Amazon VPC 控制台，请选择导航窗格中的 Internet Gateways (互联网网关)。选择 Create internet gateway (创建 Internet 网关)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或 aws-shell，请运行 Amazon EC2 **create-internet-gateway** 命令。

```
aws ec2 create-internet-gateway --output text --query 'InternetGateway.InternetGatewayId' --region us-east-2
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含新 Internet 网关的。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 aws-shell 一起运行，请忽略 aws。

输出包含新 Internet 网关的 ID。

将 Internet 网关附加到 VPC

要使用 Amazon VPC 控制台，请选择导航窗格中的 Internet Gateways (互联网网关)。选中 Internet 网关旁边的框。选择 Actions, Attach to VPC (操作，附加到 VPC) (如果可用)，然后按照屏幕上的说明进行操作。

例如，要使用 AWS CLI 或 aws-shell，请运行 Amazon EC2 **attach-internet-gateway** 命令，如下所示。

```
aws ec2 attach-internet-gateway --region us-east-2 --internet-gateway-id igw-a1b2cdef --vpc-id vpc-1234ab56
```


在前面的命令中，`us-east-2` 替换为 AWS 区域 包含互联网网关的。将 `igw-a1b2cdef` 替换为互联网网关 ID。并将 `vpc-1234ab56` 替换为 VPC ID。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

确认子网是否具有路由表

要使用 Amazon VPC 控制台，请选择导航窗格中的 Subnets (子网)。选中要使用的 VPC 的公有子网旁边的 AWS Cloud9 复选框。在 Route table (路由表) 选项卡上，如果有 Route Table (路由表) 的值，则公有子网具有一个路由表。

要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **describe-route-tables** 命令。

```
aws ec2 describe-route-tables --output table --query 'RouteTables[*].Associations[*].
{RouteTableIds:RouteTableId}' --region us-east-2 --filters Name=association.subnet-
id,Values=subnet-12a3456b
```

在前面的命令中，`us-east-2` 替换为 AWS 区域 包含公有子网的，并 `subnet-12a3456b` 替换为公有子网 ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

如果输出中存在值，则公有子网具有至少一个路由表。

在输出中，如果没有结果，则路由表可能与 VPC 而不是子网关联。要进行确认，请针对子网的相关 VPC 而不是子网本身运行 Amazon EC2 **describe-route-tables** 命令，如下所示。

```
aws ec2 describe-route-tables --output table --query 'RouteTables[*].Associations[*].
{RouteTableIds:RouteTableId}' --region us-east-2 --filters Name=vpc-
id,Values=vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包含 VPC 的 AWS 区域，然后 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

在输出中，如果存在至少一个结果，则 VPC 至少具有一个路由表。

将路由表附加到子网

要使用 Amazon VPC 控制台，请选择导航窗格中的 Route Tables (路由表)。选中要附加的路由表旁边的框。在 Subnet Associations (子网关联) 选项卡上，选择 Edit (编辑)，选中要将路由表附加到的子网旁边的框，然后选择 Save (保存)。

例如，要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **associate-route-table** 命令，如下所示。

```
aws ec2 associate-route-table --region us-east-2 --subnet-id subnet-12a3456b --route-table-id rtb-ab12cde3
```

在前面的命令中，`us-east-2` 替换为 AWS 区域 包含路由表的。将 `subnet-12a3456b` 替换为子网 ID。并将 `rtb-ab12cde3` 替换为路由表 ID。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

创建路由表

要使用 Amazon VPC 控制台，请选择导航窗格中的 Route Tables (路由表)。选择 Create Route Table (创建路由表)，然后按照屏幕上的说明进行操作。

例如，要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **create-route-table** 命令，如下所示。

```
aws ec2 create-route-table --output text --query 'RouteTable.RouteTableId' --region us-east-2 --vpc-id vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为 AWS 区域 包含新路由表的，并 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (`'`) 替换为双引号 (`"`)。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

输出包含新路由表的 ID。

查看或更改路由表的设置

要使用 Amazon VPC 控制台，请选择导航窗格中的 Route Tables (路由表)。选中路由表旁边的框。要查看这些设置，请查看每个选项卡。要更改某个选项卡中的设置，请选择 Edit (编辑)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或查看设置，请运行 Amazon EC2 **describe-route-tables** 命令，例如，如下所示。`aws-shell`

```
aws ec2 describe-route-tables --output table --region us-east-2 --route-table-ids rtb-ab12cde3
```

在前面的命令中，`us-east-2` 替换为 AWS 区域 包含路由表的，并 `rtb-ab12cde3` 替换为路由表 ID。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

建议的最低路由表设置 AWS Cloud9

目标位置	目标	状态	已传播
CIDR-BLOCK	本地	处于活动状态	否
0.0.0.0/0	igw-INTERNET-GATEWAY-ID	处于活动状态	否

在这些设置中，*CIDR-BLOCK* 是子网的 CIDR 块，*igw-INTERNET-GATEWAY-ID* 是兼容的互联网网关的 ID。

查看 VPC 的安全组列表

要使用 Amazon VPC 控制台，请选择导航窗格中的 Security Groups (安全组)。在 Search Security Groups (搜索安全组) 框中，输入 VPC ID 或名称，然后按 Enter。将在搜索结果列表中显示该 VPC 的安全组。

要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **describe-security-groups** 命令。

```
aws ec2 describe-security-groups --output table --query 'SecurityGroups[*].GroupId' --region us-east-2 --filters Name=vpc-id,Values=vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包含 AWS 区域含 VPC 的，然后 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

输出包含 VPC 的安全组 ID 的列表。

查看 AWS Cloud 计算实例的安全组列表

要使用 Amazon EC2 控制台，请展开导航窗格中的 Instances (实例)，然后选择 Instances (实例)。在实例列表中，选择实例旁边的框。该实例的安全组将出现在 Description (描述) 选项卡中的 Security groups (安全组) 旁边。

例如，要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **describe-security-groups** 命令，如下所示。

```
aws ec2 describe-instances --output table --query
'Reservations[*].Instances[*].NetworkInterfaces[*].Groups[*].GroupId' --region us-
east-2 --instance-ids i-12a3c456d789e0123
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含实例的，然后 i-12a3c456d789e0123 替换为实例 ID。要在 Windows 中运行上述命令，请将单引号 (') 替换为双引号 (")。要将上述命令与 aws-shell 一起运行，请忽略 aws。

输出包含实例的安全组 ID 的列表。

查看或更改 VPC 中的安全组的设置

要使用 Amazon VPC 控制台，请选择导航窗格中的 Security Groups (安全组)。选中安全组旁边的框。要查看这些设置，请查看每个选项卡。要更改某个选项卡上的设置，请选择 Edit (编辑) (如果适用)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或查看设置，请运行 Amazon EC2 **describe-security-groups** 命令，例如，如下所示。aws-shell

```
aws ec2 describe-security-groups --output table --region us-east-2 --group-ids
sg-12a3b456
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含实例的，然后 sg-12a3b456 替换为安全组 ID。要将上述命令与 aws-shell 一起运行，请忽略 aws。

查看或更改 AWS Cloud 计算实例安全组的设置

要使用 Amazon EC2 控制台，请展开导航窗格中的 Instances (实例)，然后选择 Instances (实例)。在实例列表中，选中实例旁边的框。在 Description (描述) 选项卡中，对于 Security groups (安全组)。查看每个选项卡。要更改某个选项卡上的设置，请选择 Edit (编辑) (如果适用)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或查看设置，请运行 Amazon EC2 **describe-security-groups** 命令，例如，如下所示。aws-shell

```
aws ec2 describe-security-groups --output table --region us-east-2 --group-ids
sg-12a3b456
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含实例的，然后 sg-12a3b456 替换为安全组 ID。要将上述命令与 aws-shell 一起运行，请忽略 aws。

的最低入站和出站流量设置 AWS Cloud9

⚠ Important

实例的 IA 安全组可能没有入站规则。如果发生这种情况，这意味着不允许从另一台主机发送到实例的传入流量。有关使用非入口 EC2 实例的信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)。

- 入站：通过端口 22 使用 SSH 的所有 IP 地址。但是，您可以将这些 IP 地址限制为仅限 AWS Cloud9 使用的 IP 地址。有关更多信息，请参阅 [AWS Cloud9 的入站 SSH IP 地址范围](#)。

📘 Note

对于 2018 年 7 月 31 日当天或之后创建的 EC2 环境，AWS Cloud9 使用安全组通过端口 22 通过 SSH 限制入站 IP 地址。具体而言，这些入站 IP 地址只是 AWS Cloud9 使用的地址。有关更多信息，请参阅 [AWS Cloud9 的入站 SSH IP 地址范围](#)。

- 入站（仅限网络 ACL）：对于 EC2 环境以及与运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例关联的 SSH 环境，所有 IP 地址均通过端口 32768-61000 使用 TCP。有关更多信息以及其他 Amazon EC2 实例类型的端口范围，请参阅 Amazon VPC 用户指南中的 [临时端口](#)。
- 出站：使用任何协议和端口的所有流量源。

您可以在安全组级别设置该行为。要获得更高级别的安全性，您还可以使用网络 ACL。有关更多信息，请参阅 Amazon VPC 用户指南中的 [安全组和网络 ACL 的比较](#)。

例如，要将入站和出站规则添加到安全组中，您可以按如下方式设置这些规则。

入站规则

类型	协议	端口范围	源
SSH (22)	TCP (6)	22	0.0.0.0 (但请参阅以下注释以及 AWS Cloud9 的入站 SSH IP 地址范围 。)

Note

对于 2018 年 7 月 31 日当天或之后创建的 EC2 环境，AWS Cloud9 添加入站规则，以限制通过端口 22 使用 SSH 的入站 IP 地址。这仅限于 AWS Cloud9 使用的地址。有关更多信息，请参阅 [AWS Cloud9 的入站 SSH IP 地址范围](#)。

出站规则

类型	协议	端口范围	源
所有流量	ALL	ALL	0.0.0.0/0

如果您还选择将入站和出站规则添加到网络 ACL 中，您可以按如下方式设置这些规则。

入站规则

规则 #	类型	协议	端口范围	源	允许/拒绝
100	SSH (22)	TCP (6)	22	0.0.0.0 (但请参阅 AWS Cloud9 的入站 SSH IP 地址范围 。)	允许
200	自定义 TCP 规则	TCP (6)	32768-61000 (适用于 Amazon Linux 和 Ubuntu Server 实例。有关其他实例类型，请参阅 临时端口 。)	0.0.0.0/0	允许
*	所有流量	ALL	ALL	0.0.0.0/0	DENY

出站规则

规则 #	类型	协议	端口范围	源	允许/拒绝
100	所有流量	ALL	ALL	0.0.0.0/0	允许
*	所有流量	ALL	ALL	0.0.0.0/0	DENY

有关安全组和网络 ACL 的更多信息，请参阅 Amazon VPC 用户指南中的以下内容。

- [安全性](#)
- [您的 VPC 的安全组](#)
- [网络 ACL](#)

在 VPC 中创建安全组

要使用 Amazon VPC 或 Amazon EC2 控制台，请执行以下任一操作：

- 在 Amazon VPC 控制台中，选择导航窗格中的 Security Groups (安全组)。选择 Create Security Group (创建安全组)，然后按照屏幕上的说明进行操作。
- 在 Amazon EC2 控制台中，展开导航窗格中的 Network & Security (网络与安全)，然后选择 Security Groups (安全组)。选择 Create Security Group (创建安全组)，然后按照屏幕上的说明进行操作。

例如，要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **create-security-group** 命令，如下所示。

```
aws ec2 create-security-group --region us-east-2 --vpc-id vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包含 AWS 区域 VPC 的，然后 `vpc-1234ab56` 替换为 VPC ID。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

确认 VPC 是否具有至少一个网络 ACL

要使用 Amazon VPC 控制台，请选择导航窗格中的 Your VPCs (您的 VPC)。选择要使用的 VPC 旁边的 AWS Cloud9 复选框。在 Summary (摘要) 选项卡上，如果具 Network ACL (网络 ACL) 的值，则 VPC 具有至少一个网络 ACL。

要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **`describe-network-acls`** 命令。

```
aws ec2 describe-network-acls --output table --query
'NetworkAcls[*].Associations[*].NetworkAclId' --region us-east-2 --filters Name=vpc-
id,Values=vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包含 AWS 区域 含 VPC 的，然后 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (`'`) 替换为双引号 (`"`)。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

如果输出包含列表中的至少一个条目，则 VPC 至少具有一个网络 ACL。

查看 VPC 的网络 ACL 列表

要使用 Amazon VPC 控制台，请选择导航窗格中的 Network ACLs (网络 ACL)。在 Search Network ACLs (搜索网络 ACL) 框中，输入 VPC ID 或名称，然后按 Enter。将在搜索结果列表中显示该 VPC 的网络 ACL。

要使用 AWS CLI 或 `aws-shell`，请运行 Amazon EC2 **`describe-network-acls`** 命令。

```
aws ec2 describe-network-acls --output table --query
'NetworkAcls[*].Associations[*].NetworkAclId' --region us-east-2 --filters Name=vpc-
id,Values=vpc-1234ab56
```

在前面的命令中，`us-east-2` 替换为包含 AWS 区域 含 VPC 的，然后 `vpc-1234ab56` 替换为 VPC ID。要在 Windows 中运行上述命令，请将单引号 (`'`) 替换为双引号 (`"`)。要将上述命令与 `aws-shell` 一起运行，请忽略 `aws`。

输出包含 VPC 的网络 ACL 的列表。

查看或更改网络 ACL 的设置

要使用 Amazon VPC 控制台，请选择导航窗格中的 Network ACLs (网络 ACL)。选择网络 ACL 旁边的框。要查看这些设置，请查看每个选项卡。要更改某个选项卡上的设置，请选择 Edit (编辑) (如果适用)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或查看设置，请运行 Amazon EC2 **`describe-network-acls`** 命令。 `aws-shell`

```
aws ec2 describe-network-acls --output table --region us-east-2 --network-acl-ids
acl-1234ab56
```


在前面的命令中，us-east-2 替换为 AWS 区域 包含网络 ACL 的，然后 ac1-1234ab56 替换为网络 ACL ID。要将上述命令与 aws-shell 一起运行，请忽略 aws。

创建网络 ACL

要使用 Amazon VPC 控制台，请选择导航窗格中的 Network ACLs (网络 ACL)。选择 Create Network ACL (创建网络 ACL)，然后按照屏幕上的说明进行操作。

要使用 AWS CLI 或 aws-shell，请运行 Amazon EC2 **create-network-acl** 命令。

```
aws ec2 create-network-acl --region us-east-2 --vpc-id vpc-1234ab56
```

在前面的命令中，us-east-2 替换为 AWS 区域 包含要将新网络 ACL 连接到的 VPC 的。另外，将 vpc-1234ab56 替换为 VPC ID。要将上述命令与 aws-shell 一起运行，请忽略 aws。

创建 VPC 以及其他 VPC 资源

按照以下过程创建 VPC 以及运行应用程序所需的其他 VPC 资源。VPC 资源包括子网、路由表、互联网网关和 NAT 网关。

使用控制台创建 VPC、子网和其他 VPC 资源

1. 通过 <https://console.aws.amazon.com/vpc/> 打开 Amazon VPC 控制台。
2. 在 VPC 控制面板上，选择创建 VPC。
3. 对于要创建的资源，选择 VPC 等。
4. 要为 VPC 资源创建名称标签，请选中名称标签自动生成。要为 VPC 资源提供自己的名称标签，请取消选中该选项。
5. 对于 IPv4 CIDR 块，必须输入 VPC 的 IPv4 地址范围。推荐的 IPv4 范围 AWS Cloud9 为 10.0.0.0/16。
6. (可选) 要支持 IPv6 流量，请选择 IPv6 CIDR 块，然后选择 Amazon 提供的 IPv6 CIDR 块。
7. 选择租赁选项。此选项定义您启动到此 VPC 中的 EC2 实例是在与其他 AWS 账户 共享的硬件上运行，还是在专供您使用的硬件上运行。如果您选择将 VPC 的租赁设为 Default，则在此 VPC 中启动的 EC2 实例将使用您在启动实例时指定的租赁属性。有关更多信息，请参阅 Amazon EC2 用户指南中的 [使用定义参数启动实例](#)。

如果您选择 VPC 的租赁为 Dedicated，则这些实例将始终在专供您使用的硬件上作为 [专用实例](#) 运行。如果您使用的是 AWS Outposts，则您的 Outpost 需要私有连接；您必须使用 Default 租赁。

8. 对于可用区 (AZ) 数量，我们建议生产环境至少在两个 Availability Zones 中预置子网。要为您的子网选择可用区，请展开自定义可用区。否则，您可以让 AWS 自己选择可用区。
9. 要配置子网，请选择公有子网的数量和私有子网的数量的值。要选择子网的 IP 地址范围，请展开自定义子网 CIDR 块。否则，让我们为您 AWS 选择它们。
10. (可选) 如果私有子网中的资源需要通过 IPv4 访问公共互联网：对于 NAT 网关，请选择要在其中创建 NAT 网关的可用区数量。在生产环境中，我们建议您每个可用区部署一个 NAT 网关，其中包含需要访问公共互联网的资源。
11. (可选) 如果私有子网中的资源需要通过 IPv6 访问公共互联网：对于仅限出口的互联网网关，请选择是。
12. (可选) 如果要直接从 VPC 访问 Amazon S3，请选择 VPC 端点、S3 网关。这将为 Amazon S3 创建一个网关 VPC 端点。有关更多信息，请参阅《AWS PrivateLink 指南》中的 [网关 VPC 端点](#)。
13. (可选) 对于 DNS 选项，默认情况下，两个域名解析选项均处于启用状态。如果默认设置无法满足您的需求，您可以停用这些选项。
14. (可选) 要向 VPC 添加标签，请展开其他标签，选择添加新标签，然后输入标签键和标签值。
15. 在预览窗格中，您可以直观地显示您所配置的 VPC 资源之间的关系。实线表示资源之间的关系。虚线表示指向 NAT 网关、互联网网关和网关端点的网络流量。创建 VPC 后，您可以使用资源地图选项卡，随时以此格式直观地显示 VPC 的资源。
16. VPC 配置完成后，选择创建 VPC。

仅创建 VPC

按照以下过程，使用 Amazon VPC 控制台创建无额外 VPC 资源的 VPC。

使用控制台创建无额外 VPC 资源的 VPC。

1. 通过 <https://console.aws.amazon.com/vpc/> 打开 Amazon VPC 控制台。
2. 在 VPC 控制面板上，选择创建 VPC。
3. 对于要创建的资源，选择 仅 VPC。
4. (可选) 对于名称标签，输入 VPC 的名称。这样做可创建具有 Name 键以及您指定的值的标签。
5. 对于 IPv4 CIDR block (IPv4 CIDR 块)，请执行以下操作之一：
 - 选择 IPv4 CIDR 手动输入，然后输入您的 VPC 的 IPv4 地址范围。推荐的 IPv4 范围 AWS Cloud9 为 10.0.0.0/16。

- 选择 IPAM 分配的 IPv4 CIDR 块，然后选择一个 Amazon VPC IP 地址管理器 (IPAM) IPv4 地址池和网络掩码。CIDR 块的大小受 IPAM 池上的分配规则限制。IPAM 是一项 VPC 功能，可帮助您规划、跟踪和监控工作负载的 IP 地址。AWS 有关更多信息，请参阅《Amazon Virtual Private Cloud 管理员指南》中的[什么是 IPAM？](#)。

如果您使用 IPAM 来管理 IP 地址，我们建议您选择此选项。否则，您为 VPC 指定的 CIDR 块可能与 IPAM CIDR 分配重叠。

6. (可选) 要创建双堆栈 VPC，请为您的 VPC 指定一个 IPv6 地址范围。对于 IPv6 CIDR block (IPv6 CIDR 块)，请执行以下操作之一：

- 选择 IPAM 分配的 IPv6 CIDR 块，然后选择您的 IPAM IPv6 地址池。CIDR 块的大小受 IPAM 池上的分配规则限制。
- 要从 Amazon 的 IPv6 地址池请求 IPv6 CIDR 块，请选择 Amazon 提供的 IPv6 CIDR 块。对于网络边界组，选择从中 AWS 通告 IP 地址的组。Amazon 提供 /56 固定大小的 IPv6 CIDR 块。
- 选择我拥有的 IPv6 CIDR，使用你[自带的 IP 地址 \(BYOIP\) AWS 使用你带来的 IP v6 CIDR 块](#)。对于 Pool (池)，选择要从中分配 IPv6 CIDR 块的 IPv6 地址池。

7. (可选) 选择租赁选项。此选项定义了您在 VPC 中启动的 EC2 实例是在与其他人共享的硬件上运行 AWS 账户 还是仅供您使用的硬件上运行。如果您选择将 VPC 的租赁设为 Default，则在此 VPC 中启动的 EC2 实例将使用您在启动实例时指定的租赁属性。有关更多信息，请参阅 Amazon EC2 用户指南中的[使用定义参数启动实例](#)。

如果您选择 VPC 的租赁为 Dedicated，则这些实例将始终在专供您使用的硬件上作为[专用实例](#)运行。如果您使用的是 AWS Outposts，则您的 Outpost 需要私有连接；您必须使用 Default 租赁。

8. (可选) 要向 VPC 添加标签，请选择添加新标签，然后输入标签键和标签值。
9. 选择创建 VPC。
10. 创建 VPC 后，您可以添加子网。

为创建子网 AWS Cloud9

您可以使用 Amazon VPC 控制台为与兼容的 VPC 创建子网 AWS Cloud9。您是否可以为 EC2 实例创建私有子网或公有子网，取决于环境与其连接的方式：

- 通过 SSH 直接访问：仅限公有子网
- 通过 Systems Manager 访问：公有子网或私有子网

在私有子网启动环境的 EC2 这一选项仅在使用[控制台、命令行或 AWS CloudFormation](#) 创建“非入口”EC2 环境时可用。

您可以按照[相同的步骤创建子网](#)，该子网可设为公有或私有。如果子网的关联路由表包含指向互联网网关的路由，则该子网是公有子网。但是如果子网的关联路由表没有指向互联网网关的路由，则该子网是私有子网。有关更多信息，请参阅[将子网配置为公有子网或私有子网](#)

如果您按照前面的步骤为其创建 VPC AWS Cloud9，则无需执行此过程。这是因为新建 VPC 向导自动为您创建子网。

Important

- AWS 账户 必须已经在同 AWS 区域 一个环境中具有兼容的 VPC。有关更多信息，请参阅[亚马逊 VPC 对以下内容的要求 AWS Cloud9](#)中的 VPC 要求。
- 对于此过程，我们建议您在中使用 IAM 管理员的证书登录 AWS Management Console 并打开 Amazon VPC 控制台 AWS 账户。如果您无法执行此操作，请咨询您的 AWS 账户 管理员。
- 某些企业可能不允许您自行创建子网。如果您无法创建子网，请咨询您的 AWS 账户 管理员或网络管理员。

创建子网

1. 如果亚马逊 VPC 控制台尚未打开，请登录 AWS Management Console 并打开亚马逊 VPC 控制台，[网址为 https://console.aws.amazon.com/vpc](https://console.aws.amazon.com/vpc)。
2. 在导航栏中 AWS 区域，如果与环境的区域不同，请选择正确的区域。
3. 如果尚未显示子网页，请在导航窗格中选择子网。
4. 选择创建子网。
5. 在 Create Subnet (创建子网) 对话框中，在 Name tag (名称标签) 中输入子网的名称。
6. 对于 VPC，请选择要与子网关联的 VPC。
7. 对于可用区，选择子网要使用的可用区，或者选择无首选项让您 AWS 选择可用区。AWS 区域
8. 对于 IPv4 CIDR 块，请使用 CIDR 格式输入供子网使用的 IP 地址范围。该 IP 地址范围必须是 VPC 中的 IP 地址子集。

有关 CIDR 块的信息，请参阅 Amazon VPC 用户指南中的[VPC 和子网大小调整](#)。另请参阅[3.1. RFC 4632 中的基本概念和前缀表示法](#)，或 Wikipedia 中的[IPv4 CIDR 块](#)。

创建子网后，[将其配置为公有子网或私有子网](#)。

将子网配置为公有子网或私有子网

创建子网后，您可以通过指定子网与互联网通信的方式将其设为公有子网或私有子网。

公有子网有一个公有 IP 地址，并且连接了一个互联网网关 (IGW)，以允许子网的实例与互联网以及其他 AWS 服务进行通信。

私有子网中的实例具有私有 IP 地址，并且网络地址转换 (NAT) 网关用于在子网的实例与互联网以及其他 AWS 服务之间来回发送流量。NAT 网关必须托管在公有子网中。

Public subnets

Note

即使您环境的实例是在私有子网中启动的，您的 VPC 也必须至少具有一个公有子网。这是因为转发进出实例的流量的 NAT 网关必须托管在公有子网中。

将子网配置为公有子网，包括将互联网网关 (IGW) 附加到该子网、配置路由表以指定通往该 IGW 的路由以及在安全组中定义用于控制入站和出站流量的设置。

有关执行这些任务的指导，请参阅 [创建 VPC 以及其他 VPC 资源](#)。

Important

如果您的开发环境[使用 SSM 访问 EC2 实例](#)，请确保该实例被启动到的公有子网分配了公有 IP 地址。为此，您必须为公有子网启用自动分配公有 IP 地址选项，并将其设置为 Yes。在子网设置页面中创建 AWS Cloud9 环境之前，可以在公有子网上启用此功能。有关修改公有子网中自动分配 IP 设置的步骤，请参阅《Amazon VPC 用户指南》中的[修改子网的公有 IPv4 寻址属性](#)。有关配置共有子网和私有子网的更多信息，请参阅[将子网配置为公有子网或私有子网](#)。

Private subnets

如果您正在创建通过 Systems Manager 访问的非入口实例，则可以在私有子网中将其启动。私有子网没有公有 IP 地址。因此，您需要一个 NAT 网关以将私有 IP 地址映射到公有地址来进行请求，并且还需要将公有 IP 地址映射回私有地址以进行响应。

⚠ Warning

您在账户中创建和使用 NAT 网关会产生费用。NAT 网关小时使用费率和数据处理费率适用于此。Amazon EC2 数据传输费同样适用。有关更多信息，请参阅 [Amazon VPC 定价](#)。

在创建和配置 NAT 网关之前，您必须执行以下操作：

- 创建一个公有 VPC 子网以托管 NAT 网关。
- 预置可分配给 NAT 网关的[弹性 IP 地址](#)。
- 对于私有子网，清除 Enable auto-assign public IPv4 address (启用自动分配公有 IPv4 地址) 复选框，以便为启动到其中的实例分配一个私有 IP 地址。有关更多信息，请参阅 Amazon VPC 用户指南中的[您的 VPC 中的 IP 地址](#)。

有关此任务中的步骤，请参阅 Amazon VPC 用户指南中的[使用 NAT 网关](#)。

⚠ Important

目前，如果您的环境的 EC2 实例启动到私有子网，则无法使用[AWS 托管临时证书](#)允许 EC2 环境 AWS 服务 代表 AWS 实体 (例如 IAM 用户) 进行访问。

SSH 环境主机要求

要指示将环境 AWS Cloud9 连接到现有的云计算实例或您自己的服务器，请创建一个 AWS Cloud9 SSH 开发环境。然而，在创建 SSH 环境之前，请考虑改为创建 EC2 环境的益处。

创建 EC2 环境时，AWS Cloud9 会创建新环境并请求 Amazon EC2 启动一个新实例，然后将新启动的实例连接到该新环境。创建 EC2 环境具有以下益处：

- 自动启动实例 在创建 EC2 环境时，会 AWS Cloud9 请求 Amazon EC2 同时创建一个新实例。在 SSH 环境中，您必须提供现有的云计算实例 (例如 Amazon EC2 实例) 或您自己的服务器。
- 自动关闭实例 默认情况下，当为 EC2 环境连接到 IDE 的所有 Web 浏览器实例都关闭后，AWS Cloud9 会在 30 分钟后自动关闭 EC2 环境。您可以随时更改此行为。这有助于减少因使用 Amazon EC2 而向您的 AWS 账户 收取额外费用的可能性。

- 自动实例清理。当您删除 EC2 环境时，会自动删除连接的 Amazon EC2 实例。这还有助于减少因使用 Amazon EC2 AWS 账户 而向您收取额外费用的可能性。在连接到云计算实例的 SSH 环境中，您必须记住自行删除该实例。
- AWS 托管的临时证书。对于 EC2 环境，您可以轻松地开启或关闭调用方中所有 AWS 资源的所有 AWS 操作 AWS 账户（有一些限制）。您无需为环境的 Amazon EC2 实例配置实例配置文件或存储 AWS 实体（例如，IAM 用户）的永久 AWS 访问证书。

有关更多信息，请参阅 [AWS 托管的临时证书](#)。

- AWS 工具包和 Git 面板。这些用于与可视源代码控制交互 AWS 服务 和使用可视源代码控制的工具 仅在使用 Amazon EC2 实例创建的 AWS Cloud9 环境中可用。

如果要改为创建 EC2 环境，请参阅 [创建 EC2 环境](#)。否则，继续阅读以获得有关创建 SSH 环境的信息。

何时以及如何创建 SSH 环境

每当您有以下任一要求时，您必须创建 SSH 环境而不是 EC2 环境：

要求	说明
您不希望因使用 AWS Cloud 计算实例 AWS 账户 而产生额外费用。因此，您决定改为 AWS Cloud9 连接到服务器外部 AWS 或您自己的服务器之外的现有云计算实例。	<ol style="list-style-type: none"> 1. 确保您的实例或服务器满足本主题下文所述的 要求。 2. 创建 SSH 环境，以便 AWS Cloud9 将您的实例或服务器连接到该环境。
您想在中使用现有的 AWS 云计算实例（例如，Amazon EC2 实例），AWS 账户 而不必 AWS Cloud9 在创建环境的同时启动新实例。	<ol style="list-style-type: none"> 1. 确保该实例满足本主题下文所述的 要求。 2. 创建 SSH 环境，以便 AWS Cloud9 将实例连接到该环境。
您想使用 AWS Cloud9 目前不支持 EC2 环境的 Amazon EC2 实例类型（例如，R4）。	<ol style="list-style-type: none"> 1. 根据所需实例类型 启动 Amazon EC2 实例。或者，确定您 AWS 账户 中运行所需实例类型的现有实例。 2. 确保该实例满足本主题下文所述的 要求。 3. 创建 SSH 环境，以便 AWS Cloud9 将实例连接到该环境。

要求	说明
您希望使用基于 Amazon 机器映像 (AMI) 而不是基于 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例。	<ol style="list-style-type: none"> 1. 根据所需 AMI 启动 Amazon EC2 实例。或者，根据所需 AWS 账户的 AMI 确定您的现有实例。 2. 确保该实例满足本主题下文所述的要求。 3. 创建 SSH 环境，以便 AWS Cloud9 将实例连接到该环境。
您希望将多个环境连接到一个现有云计算实例或您自己的服务器。	<ol style="list-style-type: none"> 1. 确保该实例或服务器满足本主题下文所述的要求。 2. 为要将实例或服务器 AWS Cloud9 连接到的每个环境@@ 创建 SSH 环境。

Note

启动 Amazon EC2 实例可能会对您的 AWS 账户收取 Amazon EC2 费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

SSH 托管要求

现有的云计算实例或您自己的服务器必须满足以下要求 AWS Cloud9 才能将其连接到 SSH 环境。

- 它必须运行 Linux。（AWS Cloud9 不支持 Windows。）
- 它不得使用基于 Arm 的架构。（对围绕 Arm 处理器构建的系统的支持正在审查中。）
- 它必须可使用 SSH 通过公共 Internet 访问。如果只能通过 Virtual Private Cloud (VPC) 或 Virtual Private Network (VPN) 来访问，则该 VPC 或 VPN 必须有权访问公有互联网。
- 如果主机是属于亚马逊虚拟私有 AWS 云 (Amazon VPC) 的现有云计算实例，则还有其他要求。有关更多信息，请参阅 [Amazon VPC 设置](#)。
- 它必须已 Python3 安装并设置为默认 Python 版本，并且在安装 pip3 时也是如此 AWS Cloud9。要检查版本，请在现有实例或服务器的终端上运行命令 `python --version`。要在实例或服务器上安装 Python，请参阅以下任一资源：
 - Python 示例中的 [步骤 1：安装所需工具](#)。
 - 从 Python 网站 [下载 Python](#)。

Note

要连接到现有 AWS Cloud 计算实例以验证并满足要求，请参阅以下一项或多项资源：

- 对于亚马逊 EC2，请参阅《亚马逊 EC2 用户指南》中的“[连接到您的 Linux 实例](#)”。
- 有关 Amazon Lightsail，请参阅 [Amazon Lightsail 文档](#) 中的连接到基于 Linux/Unix 的 Lightsail 实例。
- 有关信息 AWS Elastic Beanstalk，请参阅《AWS Elastic Beanstalk 开发人员指南》中的 [列出和连接到服务器实例](#)。
- 有关信息 AWS OpsWorks，请参阅《AWS OpsWorks 用户指南》中的“[使用 SSH 登录 Linux 实例](#)”。
- 有关其他 AWS 服务，请参阅该服务的 [文档](#)。

要连接到您自己的服务器以验证并满足要求，请使用短语搜索互联网，如“使用 ssh 命令连接到服务器”（对于 macOS 或 Linux）或“使用 PuTTY 连接到服务器”（对于 Windows）。

- 要安装所有所需的程序包，请运行以下命令：

对于 Amazon Linux：

```
sudo yum install -y make glibc-devel gcc gcc-c++
```

对于 Ubuntu Server：

```
sudo apt install build-essential
```

- 必须已安装 Node.js。我们建议安装主机操作系统支持的最新版本的 Node.js。

Warning

AWS Cloud9 如果您使用不支持的 Node.js 版本，则在创建 SSH 环境时可能会出现安装问题 AWS Cloud9。

要查看版本，请在现有实例或服务器的终端上运行命令 **node --version**。要在实例或服务器的上安装 Node.js，请参阅以下任一资源：

- Node.js 示例中的 [步骤 1：安装所需工具](#)。
- Node.js 网站上的 [通过程序包管理器安装 Node.js](#)。

- [节点版本管理器](#)已开启 GitHub。
- 对于现有实例或服务器上您在登录后希望 AWS Cloud9 从中开始操作的目录路径，其访问权限必须设置为 `rwxr-xr-x`。这意味着所有者的 read-write-run 权限与您在[创建环境向导](#)中在“配置设置”页面上为用户指定的登录名相对应，该所有者所属组的读取运行权限以及其他所有者的读取运行权限。

例如，如果目录的路径是 `~`（其中 `~` 代表您在 Configure settings（配置设置）页面上为 User（用户）指定的登录名的主目录），您可以通过在实例或服务器上使用以下命令和说明来运行 `chmod` 命令，以在目录上设置这些权限。

```
sudo chmod u=rwx,g=rx,o=rx ~
```

- 在现有实例或服务器上[下载并运行 AWS Cloud9 安装程序](#)。
- 或者，您可以将通过 SSH 的入站流量限制为仅限 AWS Cloud9 使用的 IP 地址。要执行该操作，请按照[AWS Cloud9 的入站 SSH IP 地址范围](#)中所述，将入站 SSH 流量设置为对应 IP 范围。

确定您的实例或服务器满足上述要求后，请为[创建一个 SSH 环境](#) AWS Cloud9 以将其连接到。

在 AWS Cloud9 SSH 环境中使用 AWS Cloud9 安装程序

在创建 AWS Cloud9 SSH 开发环境之前，您希望连接至环境的云计算实例（例如 Amazon EC2 实例）或您自己的服务器必须满足 [SSH 主机要求](#)。其中一个要求是必须在该实例或服务器中下载并运行 AWS Cloud9 安装程序。AWS Cloud9 安装程序是一个 Linux shell 脚本，用于检查正在运行实例或服务器的操作系统平台和架构是否受 AWS Cloud9 支持。如果此检查成功，该脚本将尝试安装 AWS Cloud9 要求实例或服务器中具有组件及其依赖项。

本主题介绍如何在目标实例或服务器上下载和运行此安装程序脚本。

- [下载并运行 AWS Cloud9 安装程序](#)
- [排除 AWS Cloud9 安装程序故障](#)

下载并运行 AWS Cloud9 安装程序

1. 确保你希望连接到环境的云计算实例或您自己的服务器满足 [SSH 主机要求](#)。这包括安装特定版本的 Python 和 Node.js；对于您希望的登录后 AWS Cloud9 开始目录设置特定权限；以及设置任何关联的 Amazon Virtual Private Cloud。
2. 在连接到实例或服务器后，请在该实例或服务器上运行以下命令之一。在运行其中一个命令之前，你需要先安装 `gcc`。

```
curl -L https://d3kgj69l4ph6w4.cloudfront.net/static/c9-install-2.0.0.sh | bash
wget -O - https://d3kgj69l4ph6w4.cloudfront.net/static/c9-install-2.0.0.sh | bash
```

3. 如果显示 Done (完成) 消息且没有任何错误, 则可[创建 SSH 环境](#)。

如果显示错误消息, 请参阅下一部分了解故障排除信息。

AWS Cloud9 安装程序故障排除

本节介绍常见问题、可能原因以及用于对 AWS Cloud9 安装程序错误进行故障排除的建议解决方案。

如果您的问题没有列出, 或者您需要更多帮助, 请参阅 [AWS Cloud9 开发论坛](#)。(当您进入此论坛时, AWS 可能会要求您登录。) 您也可以直接[联系我们](#)。

- [-bash: wget: command not found](#)
- [错误 : please install make to proceed](#)
- [错误 : please install gcc to proceed](#)
- [configure: error: curses not found](#)

`-bash: wget: command not found`

问题 : 当您运行安装程序脚本时, 显示以下消息 : `-bash: wget: command not found`。

可能的原因 : 实例或服务器上未安装 **wget** 实用程序。

建议的解决方案 : 改用 **curl** 实用程序在实例或服务器上运行安装程序脚本。

错误 : `please install make to proceed`

问题 : 当您运行安装程序脚本时, 显示以下消息 : `Error: please install make to proceed`。

可能的原因 : 实例或服务器上未安装 **make** 实用程序。

建议的解决方案 : 安装 **make** 实用程序, 然后在该实例或服务器上尝试再次运行安装程序脚本。

要安装 **make** 实用程序, 可在实例或服务器上运行以下命令之一。

- 对于在 Amazon EC2 中运行的 Amazon Linux、Amazon Linux 2 和 Red Hat Enterprise Linux (RHEL) : **sudo yum -y groupinstall "Development Tools"**
- 对于在 Amazon EC2 中运行的 Ubuntu Server : **sudo apt install -y build-essential**
- 对于 SUSE : **sudo zypper install -y make**

错误 : please install gcc to proceed

问题 : 当您运行安装程序脚本时 , 显示以下消息 : Error: please install gcc to proceed.

可能的原因 : 实例或服务器上未安装 **gcc** 实用程序。

建议的解决方案 : 安装 **gcc** 实用程序 , 然后在该实例或服务器上尝试再次运行安装程序脚本。

要安装 **gcc** 实用程序 , 可在实例或服务器上运行以下命令之一。

- 对于在 Amazon EC2 中运行的 Amazon Linux、Amazon Linux 2 和 Red Hat Enterprise Linux (RHEL) : **sudo yum -y groupinstall "Development Tools"**
- 对于在 Amazon EC2 中运行的 Ubuntu Server : **sudo apt install -y build-essential**
- 对于 SUSE : **sudo zypper install -y gcc**
- 对于其他操作系统 , 请参阅[安装 GCC](#)。

configure: error: curses not found

问题 : 当您运行安装程序脚本时 , 显示以下消息 : configure: error: curses not found.

可能的原因 : 实例或服务器上未安装 **ncurses** 终端控制库。

建议的解决方案 : 安装 **ncurses** 终端控制库 (在某些操作系统上 , 需安装 **glibc-static** 库) , 然后在该实例或服务器上尝试再次运行安装程序脚本。

要安装 **ncurses** 终端控制库 (在某些操作系统上 , 需安装 **glibc-static** 库) , 请在您的实例或服务器上运行以下命令之一 :

- 对于在 Amazon EC2 中运行的 Amazon Linux、Amazon Linux 2 和 Red Hat Enterprise Linux (RHEL) : **sudo yum -y install ncurses-devel**
- 对于 SUSE : **sudo zypper install -y ncurses-devel** 和 **sudo zypper install -y glibc-static**

AWS Cloud9 的入站 SSH IP 地址范围

可以将传入流量限制为仅 AWS Cloud9 用来通过 SSH 连接 Amazon VPC 中的 AWS 云计算实例（如 Amazon EC2 实例）或您的网络中您自己的服务器的 IP 地址范围。

Note

您可以限制传入流量限制，仅允许 AWS Cloud9 通过 SSH 进行连接的 IP 地址范围。对于在 2018 年 7 月 31 日或之后创建的 EC2 环境，您可以跳过本主题。这是因为，AWS Cloud9 会自动将该环境的入站 SSH 流量限制为仅本主题稍后描述的那些 IP 地址。AWS Cloud9 通过自动在与 Amazon EC2 实例相关联的安全组中为该环境添加一条规则来实现这一点。此规则将通过端口 22 的入站 SSH 流量限制为关联的 AWS 区域的 IP 地址。对于您网络中自己的服务器，您仍然必须遵循本主题后面描述的步骤。

如《AWS 一般参考》中的 [AWS IP 地址范围](#) 所述，大多数 AWS 区域的 IP 地址范围都在 `ip-ranges.json` 文件中。

Note

亚太地区（香港）、欧洲（米兰）和中东（巴林）等 `ip-ranges.json` 文件未包含的区域的 IP 地址范围见 [下方](#)。

要在 `ip-ranges.json` 文件中查找 IP 范围，请执行以下操作：

- 对于 Windows，使用 AWS Tools for Windows PowerShell 运行以下命令。

```
Get-AWSPublicIpAddressRange -ServiceKey CLOUD9
```

- 对于 Linux，下载 [ip-ranges.json](#) 文件。然后，可以使用诸如 `jq` 的工具，通过运行以下命令来查询它。

```
jq '.prefixes[] | select(.service=="CLOUD9")' < ip-ranges.json
```

这些 IP 范围有时可能会发生变化。只要发生更改，我们会向 AmazonIpSpaceChanged 主题的订阅者发送通知。要接收这些通知，请参阅《AWS 一般参考》中的 [AWS IP 地址范围通知](#)。

要在配置使用 AWS 云计算实例的环境时使用这些 IP 地址范围，请参阅 [AWS Cloud9 开发环境的 VPC 设置](#)。此外，如果您选择限制 EC2 环境的传入流量，或者限制与运行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 实例关联的 SSH 环境的传入流量，请确保至少允许所有在端口 32768-61000 上使用 TCP 的 IP 地址。有关更多信息以及其他 AWS 云计算实例类型的端口范围，请参阅 Amazon VPC 用户指南中的 [临时端口](#)。

要在配置使用您自己的网络的 SSH 环境时使用这些 IP 地址范围，请参阅您的网络的文档或者联系网络管理员。

未在 `ip-ranges.json` 中的 IP 地址

以下 AWS 区域的 AWS Cloud9 IP 地址范围目前未在 `ip-ranges.json` 文件中提供：亚太地区（香港）、欧洲（米兰）和中东（巴林）。下表列出了这些区域的 IP 范围。

Note

每个区域都有两个 IP 地址范围来支持 AWS Cloud9 控制层面（信息路由）和数据层面（信息处理）服务。

AWS 区域	代码	IP 范围（CIDR 表示法）
Asia Pacific (Hong Kong)	ap-east1	18.163.201.96/27
		18.163.139.32/27
Europe (Milan)	eu-south-1	15.161.135.64/27
		15.161.135.96/27
中东（巴林）	me-south-1	15.185.141.160/27
		15.185.91.32/27

AWS Cloud9 EC2 开发环境的 Amazon Machine Image (AMI) 内容

使用以下信息可获取 AWS Cloud9 用于 EC2 环境的 Amazon Machine Image (AMI) 相关详细信息。

⚠ Important

如果您环境的 Amazon EC2 实例基于 Amazon Linux 2023 AMI 或 Amazon Linux 2 AMI 模板，则实例启动后，安全更新会在实例上立即安装。然后每小时自动将安全补丁应用于实例。这些更新由后台进程应用，不会影响您对实例的使用。

在 Ubuntu EC2 环境中，实例启动后，安全更新同样会在其上立即安装。然后 `unattended-upgrades` 软件包会每天自动安装可用的更新。

主题

- [Amazon Linux 2023/Amazon Linux 2](#)
- [Ubuntu Server](#)

Amazon Linux 2023/Amazon Linux 2

⚠ Important

建议在[使用控制台创建 Amazon EC2 环境](#)时选择 Amazon Linux 2023 选项。除了提供安全、稳定和高性能的运行时系统环境外，Amazon Linux 2023 AMI 还包括到 2024 年的长期支持。

要显示 Amazon Linux 实例的版本，请从 AWS Cloud9 IDE 中针对连接的环境运行以下命令，或者从 SSH 实用工具（例如 `ssh` 命令或 PuTTY）中运行以下命令。

```
cat /etc/system-release
```

要显示在 Amazon Linux 实例上安装的软件包的列表，请运行以下一个或多个命令。

要将所有已安装软件包显示为单个列表，请执行以下操作：

```
sudo yum list installed
```

要显示软件包名称中包含指定文本的已安装软件包列表，请执行以下操作：

```
sudo yum list installed | grep YOUR_SEARCH_TERM
```

在上述命令中，将 `YOUR_SEARCH_TERM` 替换为软件包名称的一部分。例如，要显示名称中包含 `sql` 的所有已安装软件包，请执行以下操作：

```
sudo yum list installed | grep sql
```

要一次性在一页中显示所有已安装软件包的列表，请执行以下操作：

```
sudo yum list installed | less
```

要滚动查看所显示的页面，请执行以下操作：

- 要下移一行，请按 **j**。
- 要上移一行，请按 **k**。
- 要向后翻一页，请按 **Ctrl-F**。
- 要向前翻一页，请按 **Ctrl-B**。
- 要退出，请按 **q**。

Note

利用 Amazon Linux 2，您可以使用 Extras 库以在您的实例上安装应用程序和软件更新。这些软件更新称为主题。有关更多信息，请参阅[亚马逊 EC2 用户指南中的附加库 \(Amazon Linux 2\)](#)。

如需了解其他选项，请运行 `man yum` 命令。同时请参阅以下资源：

- Amazon Linux 2023：[AMI 页](#)。
- Amazon Linux：[Amazon Linux AMI 2018.03 软件包](#)。

Ubuntu Server

要显示 Ubuntu Server 实例的版本，请从已连接环境的 AWS Cloud9 IDE 或者从 SSH 实用工具（例如 `ssh` 命令或 PuTTY）运行以下命令。

```
lsb_release -a
```

版本将显示在 Description（描述）字段旁。

要显示在 Ubuntu Server 上安装软件包的列表，请运行以下一个或多个命令。

要将所有已安装软件包显示为单个列表，请执行以下操作：

```
sudo apt list --installed
```

要显示软件包名称中包含指定文本的已安装软件包列表，请执行以下操作：

```
sudo apt list --installed | grep YOUR_SEARCH_TERM
```

在上述命令中，将 YOUR_SEARCH_TERM 替换为软件包名称的一部分。例如，要显示名称中包含 sql 的所有已安装软件包，请执行以下操作：

```
sudo apt list --installed grep sql
```

要一次性在一页中显示所有已安装软件包的列表，请执行以下操作：

```
sudo apt list --installed | less
```

要滚动查看所显示的页面，请执行以下操作：

- 要下移一行，请按 **j**。
- 要上移一行，请按 **k**。
- 要向后翻一页，请按 **Ctrl-F**。
- 要向前翻一页，请按 **Ctrl-B**。
- 要退出，请按 **q**。

如需了解其他选项，请运行 `man apt` 命令。另请参阅 Ubuntu 网站上的 [Ubuntu 软件包搜索](#)。

将服务相关角色用于 AWS Cloud9

AWS Cloud9 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种与 AWS Cloud9 直接关联的独特类型的 IAM 角色。服务相关角色由 AWS Cloud9 预定义，并包含该服务代表您调用其他 AWS 服务所需的一切权限。

服务相关角色使 AWS Cloud9 的设置更轻松，因为您不必添加必要的权限。AWS Cloud9 会定义其服务相关角色的权限，且仅 AWS Cloud9 可以代入其角色。定义的权限包括信任策略和权限策略，以及不能附加到任何其它 IAM 实体的权限策略。

只有在首先删除角色的相关资源后，才能删除角色。这将保护您的 AWS Cloud9 资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅[使用 IAM 的 AWS 服务](#)并查找 Service-Linked Role (服务相关角色) 列中显示为 Yes (是) 的服务。请选择 Yes 与查看该服务的服务相关角色文档的链接。

- [AWS Cloud9 的服务相关角色权限](#)
- [为 AWS Cloud9 创建服务相关角色](#)
- [为 AWS Cloud9 编辑服务相关角色](#)
- [删除 AWS Cloud9 的服务相关角色](#)
- [AWS Cloud9 服务相关角色的受支持区域](#)

AWS Cloud9 的服务相关角色权限

AWS Cloud9 使用名为 AWSServiceRoleForAWSCloud9 的服务相关角色。该服务相关角色信任 cloud9.amazonaws.com 服务担任该角色。

此服务相关角色的权限策略名为 AWSCloud9ServiceRolePolicy，它允许 AWS Cloud9 对指定资源完成策略中罗列的操作。

Important

如果使用的是 License Manager 并且收到 unable to access your environment 错误，您需要用支持 License Manager 的版本替换旧的服务相关角色。只需删除旧角色即可替换旧角色。之后将自动创建更新后的角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
```

```

    "ec2:DescribeInstances",
    "ec2:DescribeInstanceStatus",
    "cloudformation:CreateStack",
    "cloudformation:DescribeStacks",
    "cloudformation:DescribeStackEvents",
    "cloudformation:DescribeStackResources"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:TerminateInstances",
    "ec2>DeleteSecurityGroup",
    "ec2:AuthorizeSecurityGroupIngress"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "cloudformation>DeleteStack"
  ],
  "Resource": "arn:aws:cloudformation:*:*:stack/aws-cloud9-*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:security-group/*"
  ],
  "Condition": {
    "StringLike": {
      "aws:RequestTag/Name": "aws-cloud9-*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StartInstances",

```

```

    "ec2:StopInstances"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "ec2:ResourceTag/aws:cloudformation:stack-name": "aws-cloud9-*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances"
  ],
  "Resource": [
    "arn:aws:license-manager:*:*:license-configuration:*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:ListInstanceProfiles",
    "iam:GetInstanceProfile"
  ],
  "Resource": [
    "arn:aws:iam:*:*:instance-profile/cloud9/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam:*:*:role/service-role/AWSCloud9SSMAccessRole"
  ],
  "Condition": {
    "StringLike": {
      "iam:PassedToService": "ec2.amazonaws.com"
    }
  }
}
]

```

```
}
```

您必须配置权限以允许 AWS Cloud9 代表 IAM 实体（如用户、组或角色）创建服务相关角色。

要允许 AWS Cloud9 创建 `AWSServiceRoleForAWSCloud9` 服务相关角色，请将以下语句添加到 IAM 实体（AWS Cloud9 需要代表其创建服务相关角色）的权限策略中。

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "cloud9.amazonaws.com"
    }
  }
}
```

或者，您也可以将 AWS 托管式策略 `AWSCloud9User` 或 `AWSCloud9Administrator` 添加到 IAM 实体中。

要允许 IAM 实体删除 `AWSServiceRoleForAWSCloud9` 服务相关角色，请将以下语句添加到需要删除服务相关角色的 IAM 实体的权限策略中。

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "cloud9.amazonaws.com"
    }
  }
}
```

为 AWS Cloud9 创建服务相关角色

您无需创建服务相关角色。您创建 AWS Cloud9 开发环境时，AWS Cloud9 会为您创建服务相关角色。

为 AWS Cloud9 编辑服务相关角色

您无法在 AWS Cloud9 中编辑 AWSServiceRoleForAWSCloud9 服务相关角色。例如，在创建服务相关角色后，您无法更改角色的名称，因为不同的实体可能会引用该角色。但是可以使用 IAM 编辑角色说明。有关更多信息，请参见 IAM 用户指南中的[编辑服务相关角色](#)。

删除 AWS Cloud9 的服务相关角色

如果不再需要使用某个需要服务相关角色的特征或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。

在 IAM 中删除服务相关角色

您必须先删除服务相关角色使用的所有 AWS Cloud9 资源，然后才能使用 IAM 删除该角色。要删除 AWS Cloud9 资源，请参见[删除环境](#)。

您可以使用 IAM 控制台删除 AWSServiceRoleForAWSCloud9 服务相关角色。有关更多信息，请参见 IAM 用户指南中的[删除服务相关角色](#)。

AWS Cloud9 服务相关角色的受支持区域

AWS Cloud9 支持在服务可用的所有区域中使用服务相关角色。有关更多信息，请参见 Amazon Web Services 一般参考 中的 [AWS Cloud9](#)。

使用 AWS Cloud9 记录 AWS CloudTrail API 调用

AWS Cloud9 与 CloudTrail 集成，后者是在 AWS Cloud9 中提供用户、角色或 AWS 服务 所采取操作的记录的服务。CloudTrail 将 AWS Cloud9 的所有 API 调用作为事件捕获。捕获的调用包含来自 AWS Cloud9 控制台和来自对 AWS Cloud9 API 的代码调用的调用。如果您创建跟踪记录，则可以使 CloudTrail 事件持续传送到 Amazon Simple Storage Service (Amazon S3) 存储桶（包括 AWS Cloud9 事件）。如果您不配置跟踪记录，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS Cloud9 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参见 [《AWS CloudTrail 用户指南》](#)。

CloudTrail 中的 AWS Cloud9 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 AWS Cloud9 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history (事件历史记录) 中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅[使用 CloudTrail 事件历史记录查看事件](#)。

对于 AWS 账户中的事件的持续记录 (包括 AWS Cloud9 的事件)，请创建跟踪记录。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service (Amazon S3) 存储桶。预设情况下，在控制台中创建跟踪记录时，此跟踪记录将适用于所有 AWS 区域。跟踪记录 AWS 分区所有区域的事件，将日志文件传送至指定的 S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- [创建跟踪概览](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件](#)

AWS Cloud9 支持将以下操作记录为 CloudTrail 日志文件中的事件：

- CreateEnvironmentEC2
- CreateEnvironmentSSH
- CreateEnvironmentMembership
- DeleteEnvironment
- DeleteEnvironmentMembership
- DescribeEnvironmentMemberships
- DescribeEnvironments
- DescribeEnvironmentStatus
- ListEnvironments
- ListTagsForResource
- TagResource
- UntagResource
- UpdateEnvironment
- UpdateEnvironmentMembership

Note

AWS Cloud9 的某些 CloudTrail 事件不是由公有 API 操作导致的。相反，以下事件是由影响用户身份验证和托管式临时凭证的内部更新启动的：

- `DisableManagedCredentialsByCollaborator`
- `EnvironmentTokenSuccessfullyCreated`
- `ManagedCredentialsUpdatedOnEnvironment`

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management IAM 用户凭证发出的。
- 请求是使用角色还是联合身份用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务 发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

了解 AWS Cloud9 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间以及请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

- [CreateEnvironmentEC2](#)
- [CreateEnvironmentSSH](#)
- [CreateEnvironmentMembership](#)
- [DeleteEnvironment](#)
- [DeleteEnvironmentMembership](#)
- [DescribeEnvironmentMemberships](#)
- [DescribeEnvironments](#)
- [DescribeEnvironmentStatus](#)
- [ListEnvironments](#)
- [ListTagsForResource](#)

- [TagResource](#)
- [UntagResource](#)
- [UpdateEnvironment](#)
- [UpdateEnvironmentMembership](#)

CreateEnvironmentEC2

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 CreateEnvironmentEC2 操作。

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    {
      "eventTime": "2019-01-14T11:33:27Z",
      "eventSource": "cloud9.amazonaws.com",
      "eventName": "CreateEnvironmentEC2",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "signin.amazonaws.com",
      "requestParameters": {
        "instanceType": "t2.small",
        "subnetId": "subnet-1d4a9eEX",
        "description": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "dryRun": true,
        "automaticStopTimeMinutes": 30,
        "name": "my-test-environment",

```

```

    "clientRequestToken": "cloud9-console-f8e37272-e541-435d-a567-5c684EXAMPLE"
  },
  "responseElements": null,
  "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
  "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]
}

```

CreateEnvironmentSSH

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 CreateEnvironmentSSH 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    {
      "eventTime": "2019-01-14T11:33:27Z",
      "eventSource": "cloud9.amazonaws.com",
      "eventName": "CreateEnvironmentSSH",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "signin.amazonaws.com",
      "requestParameters": {
        "host": "198.51.100.0",
        "port": 22,

```

```

    "name": "my-ssh-environment",
    "description": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "clientRequestToken": "cloud9-console-b015a0e9-469e-43e3-be90-6f432EXAMPLE",
    "loginName": "ec2-user"
  },
  "responseElements": {
    "environmentId": "5c39cc4a85d74a8bbb6e23ed6EXAMPLE"
  },
  "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
  "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]
}

```

CreateEnvironmentMembership

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 CreateEnvironmentMembership 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2019-01-14T11:33:27Z",
    "eventSource": "cloud9.amazonaws.com",
    "eventName": "CreateEnvironmentMembership",

```

```

"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
  "userArn": "arn:aws:iam::111122223333:user/MyUser",
  "permissions": "read-write"
},
"responseElements": {
  "membership": {
    "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
    "permissions": "read-write",
    "userId": "AIDACKCEVSQ6C2EXAMPLE",
    "userArn": "arn:aws:iam::111122223333:user/MyUser"
  }
},
"requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
"eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
]
}

```

DeleteEnvironment

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DeleteEnvironment 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      }
    }
  ]
}

```

```

    }
  },
  "invokedBy": "signin.amazonaws.com"
},
"eventTime": "2019-01-14T11:33:27Z",
"eventSource": "cloud9.amazonaws.com",
"eventName": "DeleteEnvironment",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE"
},
"responseElements": null,
"requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
"eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
]
}

```

DeleteEnvironmentMembership

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DeleteEnvironmentMembership 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      }
    }
  ]
}

```

```

    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2019-01-14T11:33:27Z",
  "eventSource": "cloud9.amazonaws.com",
  "eventName": "DeleteEnvironmentMembership",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
    "userArn": "arn:aws:iam::111122223333:user/MyUser",
  },
  "responseElements": null,
  "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
  "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]
}

```

DescribeEnvironmentMemberships

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DescribeEnvironmentMemberships 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      }
    }
  ]
}

```

```

    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2019-01-14T11:33:27Z",
  "eventSource": "cloud9.amazonaws.com",
  "eventName": "DescribeEnvironmentMemberships",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "nextToken": "NEXT_TOKEN_EXAMPLE",
    "permissions": [ "owner" ],
    "maxResults": 15
  },
  "responseElements": null,
  "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
  "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
]
}

```

DescribeEnvironments

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DescribeEnvironments 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      }
    }
  ]
}

```

```

    }
  },
  "invokedBy": "signin.amazonaws.com"
},
"eventTime": "2019-01-14T11:33:27Z",
"eventSource": "cloud9.amazonaws.com",
"eventName": "DescribeEnvironments",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "signin.amazonaws.com",
"requestParameters": {
  "environmentIds": [
    "2f5ff70a640f49398f67e3bdeb811ab2"
  ]
},
"responseElements": null,
"requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
"eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
"readOnly": true,
"eventType": "AwsApiCall",
"recipientAccountId": "111122223333"
}
]
}

```

DescribeEnvironmentStatus

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 DescribeEnvironmentStatus 操作。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:sts::123456789012:myuser_role",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:sts::123456789012:myuser_role",
        "accountId": "123456789012",

```



```

        "userName": "barshane_role"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-03-12T15:10:54Z"
    }
}
},
"eventTime": "2021-03-12T15:13:31Z",
"eventSource": "cloud9.amazonaws.com",
"eventName": "DescribeEnvironmentStatus",
"awsRegion": "us-east-1",
"sourceIPAddress": "XX.XX.XXX.XX",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.951
Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.282-b08
java/1.8.0_282 vendor/Oracle_Corporation",
"requestParameters": {
    "environmentId": "31ea8a12746a4221b7d8e07d9ef6ee21"
},
"responseElements": null,
"requestID": "68b163fb-aa88-4f40-bafd-4a18bf24cbd5",
"eventID": "c0fc52a9-7331-4ad0-a8ee-157995dfb5e6",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012"
}

```

ListEnvironments

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 ListEnvironments 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",

```

```

    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "MyUser",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-01-14T11:29:47Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2019-01-14T11:33:27Z",
  "eventSource": "cloud9.amazonaws.com",
  "eventName": "ListEnvironments",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "nextToken": "NEXT_TOKEN_EXAMPLE",
    "maxResults": 15
  },
  "responseElements": null,
  "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
  "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
]
}

```

ListTagsForResource

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 ListTagsForResource 操作。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:sts::123456789012:myuser_role",
    "accountId": "123456789012",
    "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
    "sessionContext": {

```

```

    "sessionIssuer": {
      "type": "Role",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE",
      "arn": "123456789012:myuser_role",
      "accountId": "123456789012",
      "userName": "barshane_role"
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-03-23T16:41:51Z"
    }
  }
},
"eventTime": "2021-03-23T16:42:58Z",
"eventSource": "cloud9.amazonaws.com",
"eventName": "ListTagsForResource",
"awsRegion": "us-east-1",
"sourceIPAddress": "XX.XX.XXX.XX",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.976
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.282-b08
java/1.8.0_282 vendor/Oracle_Corporation cfg/retry-mode/legacy",
"requestParameters": {
  "resourceARN": "arn:aws:cloud9:us-
east-1:123456789012:environment:3XXXXXXXXXX6a4221b7d8e07d9ef6ee21"
},
"responseElements": {
  "tags": "HIDDEN_DUE_TO_SECURITY_REASONS"
},
"requestID": "5750a344-8462-4020-82f9-f1d500a75162",
"eventID": "188d572d-9a14-4082-b98b-0389964c7c30",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012"
}

```

TagResource

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 TagResource 操作。

```
{
```

```
"eventVersion": "1.08",
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AIDACKCEVSQ6C2EXAMPLE",
  "arn": "arn:aws:sts::123456789012:myuser_role",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE",
      "arn": "arn:aws:iam::123456789012:role/myuser_role",
      "accountId": "123456789012",
      "userName": "MyUser"
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-03-23T15:03:57Z"
    }
  }
},
"eventTime": "2021-03-23T15:08:16Z",
"eventSource": "cloud9.amazonaws.com",
"eventName": "TagResource",
"awsRegion": "us-east-1",
"sourceIPAddress": "54.XXX.XXX.XXX",
"userAgent": "aws-internal/3 aws-sdk-java/1.11.976
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.282-b08
java/1.8.0_282 vendor/Oracle_Corporation cfg/retry-mode/legacy",
"requestParameters": {
  "resourceARN": "arn:aws:cloud9:us-
east-1:123456789012:environment:3XXXXXXXXX6a4221b7d8e07d9ef6ee21",
  "tags": "HIDDEN_DUE_TO_SECURITY_REASONS"
},
"responseElements": null,
"requestID": "658e9d70-91c2-41b8-9a69-c6b4cc6a9456",
"eventID": "022b2893-73d1-44cb-be6f-d3faa68e83b1",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012"
```

```
}
```

UntagResource

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 UntagResource 操作。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:sts::123456789012/MyUser",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:MyUser",
        "accountId": "123456789012",
        "userName": "MyUser"
      },
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2021-03-23T15:58:36Z"
      }
    }
  },
  "eventTime": "2021-03-23T16:05:08Z",
  "eventSource": "cloud9.amazonaws.com",
  "eventName": "UntagResource",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "3.XX.XX.XXX",
  "userAgent": "aws-internal/3 aws-sdk-java/1.11.976
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.282-b08
java/1.8.0_282 vendor/Oracle_Corporation cfg/retry-mode/legacy",
  "requestParameters": {
    "resourceARN": "arn:aws:cloud9:us-
east-1:123456789012:environment:3XXXXXXXXX6a4221b7d8e07d9ef6ee21",
    "tagKeys": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "responseElements": null,
```

```

"requestID": "0eadaef3-dc0a-4cd7-85f6-135b8529f75f",
"eventID": "41f2f2e2-4b17-43d4-96fc-9857981ca1de",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012"
}

```

UpdateEnvironment

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 UpdateEnvironment 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "MyUser",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2019-01-14T11:29:47Z"
          }
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    {
      "eventTime": "2019-01-14T11:33:27Z",
      "eventSource": "cloud9.amazonaws.com",
      "eventName": "UpdateEnvironment",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "signin.amazonaws.com",
      "requestParameters": {
        "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
        "description": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "name": "my-test-environment-renamed"
      }
    },
  ],
}

```

```

    "responseElements": null,
    "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
    "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
]
}

```

UpdateEnvironmentMembership

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 UpdateEnvironmentMembership 操作。

```

{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/MyUser",
        "accountId": "111122223333",
      },
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "MyUser",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2019-01-14T11:29:47Z"
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    {
      "eventTime": "2019-01-14T11:33:27Z",
      "eventSource": "cloud9.amazonaws.com",
      "eventName": "UpdateEnvironmentMembership",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "signin.amazonaws.com",
      "requestParameters": {
        "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
        "userArn": "arn:aws:iam::111122223333:user/MyUser",
        "permissions": "read-only"
      }
    }
  ]
}

```

```
    },
    "responseElements": {
      "membership": {
        "environmentId": "2f5ff70a640f49398f67e3bdeEXAMPLE",
        "permissions": "read-only",
        "userId": "AIDACKCEVSQ6C2EXAMPLE",
        "userArn": "arn:aws:iam::111122223333:user/MyUser"
      }
    },
    "requestID": "f0e629fb-fd37-49bd-b2cc-e9822EXAMPLE",
    "eventID": "8a906445-1b2a-47e9-8d7c-5b242EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  }
}]}
```

Tags

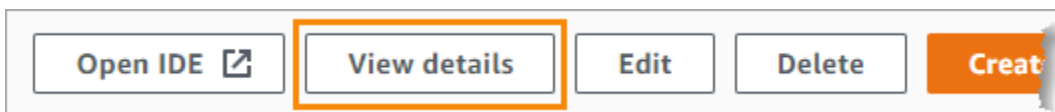
标签是一个您或 AWS 附加到 AWS 资源的标注或属性。每个标签均包含一个键 和一个配对的值。您可以使用标签来控制对 AWS Cloud9 资源的访问，如 [IAM 用户指南](#) 中的 [使用 AWS 资源标签控制访问](#) 中所述。标签还可以帮助您管理账单信息，如 [用户定义的成本分配标签](#) 中所述。

当您 [创建 AWS Cloud9 EC2 开发环境](#) 后，AWS Cloud9 包括管理环境所需的某些系统标签。系统标签以“aws:”开头。在创建过程中，还可以添加您自己的资源标签。

创建环境后，您可以查看附加到环境的标签，向环境中添加新的资源标签，或者修改或删除先前添加的标签。您最多可以将 50 个用户定义的标签附加到一个 AWS Cloud9 环境中。

使用以下一种或多种方法来查看或更新标签。

- 在 [AWS Cloud9 控制台](#) 中，选择所关注的环境，然后选择 View Details (查看详细信息)。



- 使用以下 AWS Cloud9 CLI 命令：[list-tags-for-resource](#)、[tag-resource](#) 和 [untag-resource](#)。
- 使用以下 AWS Cloud9 API 操作：[ListTagsForResource](#)、[TagResource](#) 和 [UntagResource](#)。

⚠ Warning

您使用上述方法为 AWS Cloud9 创建或更新的标签不会自动传播到底层资源。有关如何执行此操作的信息，请参阅[将标签更新传播到底层资源](#)。

将标签更新传播到底层资源

当您使用 AWS Cloud9 CLI 命令或 API 操作来添加、修改或删除附加到 AWS Cloud9 环境的标签时，这些更改不会自动传播到底层资源（例如 AWS CloudFormation 堆栈、Amazon EC2 实例和 Amazon EC2 安全组）。必须手动传播这些更改。

为了更轻松地使用以下过程，您可以获取所关注环境的环境 ID。如果要进行此操作，请执行以下步骤：

1. 在 [AWS Cloud9 控制台](#) 中，选择所关注的环境，然后选择 View Details（查看详细信息）。
2. 查找 Environment ARN (环境 ARN) 属性并记录环境 ID，这是环境 ARN 的一部分，位于“environment:”的后面。

您必须将标签更新传播到一个或多个位置，这具体取决于您将标签用于哪些位置。

将标签更新传播到 AWS CloudFormation 堆栈

i Note

当您更新标签到 AWS CloudFormation 堆栈时，这些更新将自动传播到与该堆栈关联的 Amazon EC2 实例和 Amazon EC2 安全组。

1. 导航到 [AWS CloudFormation 控制台](#)。
2. 查找并选择与您所关注的 AWS Cloud9 环境相对应的堆栈。如果您记录了环境 ID，则可以使用该 ID 筛选环境。
3. 在 Stack info (堆栈信息) 选项卡的 Tags (标签) 部分中，查看标签的列表。
4. 如果需要更新标签，请选择页面顶部附近的 Update (更新)，然后按照相应的说明操作。有关更多信息，请参阅 [AWS CloudFormation 用户指南](#) 中的 [直接更新堆栈](#)。

您还可以使用 [describe-stacks](#) 和 [update-stack](#) CLI 命令更新标签。

将标签更新传播到 Amazon EC2 实例

1. 导航到 [Amazon EC2 实例](#) 控制台。
2. 查找并选择与您所关注的 AWS Cloud9 环境相对应的 Amazon EC2 实例。如果您先前记录了环境 ID，则可以使用该 ID 筛选环境。
3. 在 Tags (标签) 选项卡上，根据需要查看和更新标签。

您还可以使用 [describe-tags](#)、[create-tags](#) 和 [delete-tags](#) CLI 命令更新标签。

将标签更新传播到 Amazon EC2 安全组

1. 导航到 [Amazon EC2 安全组](#) 控制台。
2. 查找并选择与您所关注的 AWS Cloud9 环境相对应的安全组。如果您先前记录了环境 ID，则可以使用该 ID 筛选环境。
3. 打开 Tags (标签) 选项卡，根据需要查看和更新标签。

您还可以使用 [describe-tags](#)、[create-tags](#) 和 [delete-tags](#) CLI 命令更新标签。

的安全性 AWS Cloud9

云安全性一直是 Amazon Web Services (AWS) 的重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性。

云安全 — AWS 负责保护运行 AWS 云中提供的所有服务的基础架构，并为您提供可以安全使用的服务。我们的安全责任是重中之重 AWS，作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。

云端安全 — 您的责任由您使用的 AWS 服务以及其他因素决定，包括数据的敏感性、组织的要求以及适用的法律和法规。

AWS Cloud9 通过其支持的特定 AWS 服务遵循[分担责任模式](#)。有关 AWS 服务安全信息，请参阅[AWS 服务安全文档页面](#)和[合规计划合 AWS 规工作范围内的AWS 服务](#)。

以下主题向您介绍如何进行配置 AWS Cloud9 以满足您的安全和合规性目标。

主题

- [中的数据保护 AWS Cloud9](#)
- [适用于 Identity and Access Managem AWS Cloud9](#)
- [登录和监控 AWS Cloud9](#)
- [合规性验证 AWS Cloud9](#)
- [韧性在 AWS Cloud9](#)
- [中的基础设施安全 AWS Cloud9](#)
- [软件更新和修补](#)
- [以下方面的安全最佳实践 AWS Cloud9](#)

中的数据保护 AWS Cloud9

分 AWS [担责任模型](#)适用于中的数据保护 AWS Cloud9。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础设施上的内容的控制。您还负责您所使用的 AWS 服务的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。有关欧洲数据保护的信息，请参阅 AWS 安全性博客上的 [AWS 责任共担模式和 GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅 [《美国联邦信息处理标准 \(FIPS \) 第 140-2 版》](#)。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API AWS Cloud9 或 SDK 或以其他 AWS 服务方式使用控制台 AWS CLI、API 或 AWS SDK 的情况。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

数据加密

数据加密是指保护传输中的数据（当数据在您的 AWS 账户之间 AWS Cloud9 传输时）和静态数据（当数据存储在 AWS Cloud9 配置存储和 AWS 云计算实例中时）。

在的背景下 AWS Cloud9，以下类型的数据可能需要通过加密进行保护：

您的内容和数据

您处理、收集和存储的信息。下面是此类型的数据的示例：

- 您的代码文件
- 附加 EC2 环境或 SSH 环境的配置、应用程序和数据

AWS Cloud9 元数据

AWS Cloud9 操纵、收集和存储的数据。下面是此类型的数据的示例：

- 各种 IDE 设置，例如选项卡状态、打开的文件以及 IDE 首选项
- AWS Cloud9 开发环境元数据，例如环境名称和描述
- AWS Cloud9 服务 API 和控制台日志
- 服务日志，例如 HTTP 请求

AWS Cloud9 还会通过其数据平面服务传输您的一些内容和数据。这包括您的文件、终端输入、输出文本和一些 IDE 命令（例如，用于保存文件的命令）。

静态加密

静态加密是指通过对存储数据进行加密来保护您的数据免受未经授权的访问。存储在 AWS Cloud9 环境中的任何客户数据（例如代码文件、软件包或依赖关系）始终存储在客户的资源中。如果客户使用亚马逊 EC2 环境，则数据将存储在其账户中存在的关联亚马逊弹性区块存储 (Amazon EBS) Block Store 卷中。AWS 如果客户使用 SSH 环境，则数据将存储在客户的 Linux 服务器上的本地存储中。

为 AWS Cloud9 开发环境创建 Amazon EC2 实例时，会创建一个未加密的 Amazon EBS 卷并将其附加到该实例。想要加密数据的客户需要创建一个加密的 EBS 卷并将其连接到 EC2 实例。AWS Cloud9 附加的 Amazon EBS 卷支持 Amazon EBS 默认加密，默认情况下，这是特定于区域的设置。有关更多信息，请参阅《AWS Elastic Compute Cloud 用户指南》中的[默认加密](#)。

有关 AWS Cloud9 开发环境的元数据（例如环境名称、环境成员和 IDE 设置）由客户资源存储 AWS，而不是存储在客户资源中。客户特定的信息（例如环境说明和 IDE 设置）将会进行加密。

传输中加密

传输中加密是指在通信终端节点之间移动数据时，保护您的数据免遭拦截。客户机与 AWS Cloud9 服务之间传输的所有数据均通过 HTTPS、WSS 和加密的 SSH 进行加密。

- HTTPS — 确保客户的 Web 浏览器和 AWS Cloud9 服务之间的请求安全。AWS Cloud9 还会加载通过客户浏览器通过 HTTPS CloudFront 发送的 Amazon 资产。
- WSS (WebSocket 安全) — 通过客户的网络浏览器和 AWS Cloud9 服务 WebSockets 之间实现安全的双向通信。
- 加密 SSH (安全外壳) : 支持在客户端的 Web 浏览器和 AWS Cloud9 服务之间安全传输数据。

HTTPS、WSS 和 SSH 协议的使用取决于您使用支持的浏览器 AWS Cloud9。请参阅 [AWS Cloud9 支持的浏览器](#)。

Note

默认情况下，加密协议在 AWS Cloud9 中实施。客户无法更改 encryption-in-transit 设置。

密钥管理

AWS Key Management Service (AWS KMS) 是一项托管服务 AWS KMS keys，用于创建和控制用于加密客户数据的加密密钥。AWS Cloud9 代表客户生成和管理用于加密数据的加密密钥。

互连网络流量隐私保护

SSH 环境连接到客户拥有的本地计算和存储。加密的 SSH、HTTPS 和 WSS 连接支持该服务与 SSH 环境之间的数据传输。

您可以将 AWS Cloud9 EC2 开发环境（由 Amazon EC2 实例提供支持）配置为在特定 VPC 和子网中启动。有关 Amazon Virtual Private Cloud 设置的更多信息，请参阅 [AWS Cloud9 开发环境的 VPC 设置](#)。

适用于 Identity and Access Management AWS Cloud9

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（拥有权限）使用 AWS Cloud9 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [如何 AWS Cloud9 与 IAM 配合使用](#)
- [适用于 AWS Cloud9 的基于身份的策略示例](#)
- [对 AWS Cloud9 身份和访问进行故障排除](#)
- [如何 AWS Cloud9 使用 IAM 资源和操作](#)
- [AWS 的托管策略 AWS Cloud9](#)

- [为以下各项创建客户托管策略 AWS Cloud9](#)
- [AWS Cloud9 权限参考](#)
- [AWS 托管的临时证书](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您所做的工作 AWS Cloud9。

服务用户-如果您使用 AWS Cloud9 服务完成工作，则管理员会为您提供所需的凭证和权限。当您使用更多 AWS Cloud9 功能来完成工作时，您可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS Cloud9 中的特征，请参阅 [对 AWS Cloud9 身份和访问进行故障排除](#)。

服务管理员-如果您负责公司的 AWS Cloud9 资源，则可能拥有完全访问权限 AWS Cloud9。您的工作是确定您的服务用户应访问哪些 AWS Cloud9 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何将 IAM 与配合使用 AWS Cloud9，请参阅[如何 AWS Cloud9 与 IAM 配合使用](#)。

IAM 管理员：如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS Cloud9 的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS Cloud9 基于身份的策略示例，请参阅。[适用于 AWS Cloud9 的基于身份的策略示例](#)

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当您使用联合访问 AWS 时，您就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》[中的如何登录到您 AWS 账户的](#)。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity Center 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。AWS Directory Service 当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center ?](#)。

IAM 用户和群组

[IAM 用户](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以使用 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- **联合用户访问** – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的[为第三方身份提供商创建角色](#)。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的[权限集](#)。
- **临时 IAM 用户权限** – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- **跨账户存取** – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。
- **跨服务访问** — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
 - **转发访问会话 (FAS)** — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。
- **服务角色** - 服务角色是服务代表您在您的账户中执行操作而分派的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。
- **服务相关角色**-服务相关角色是一种与服务相关联的服务角色。AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- **在 Amazon EC2 上运行的应用程序** — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文

件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的 [何时创建 IAM 角色（而不是用户）](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS，当与身份或资源关联时，它会定义其权限。AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 [JSON 策略概览](#)。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的 [创建 IAM 策略](#)。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的 [在托管策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可

以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表 (ACL) 控制哪些主体 (账户成员、用户或角色) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。AWS WAF 要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的[访问控制列表 \(ACL\) 概览](#)。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体 (IAM 用户或角色) 授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户项进行分组和集中管理的服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体 (包括每个 AWS 账户根用户实体) 的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的[SCP 的工作原理](#)。
- 会话策略 - 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

如何 AWS Cloud9 与 IAM 配合使用

在使用 IAM 管理访问权限之前 AWS Cloud9，请先了解有哪些 IAM 功能可供使用 AWS Cloud9。

您可以搭配使用的 IAM 功能 AWS Cloud9

IAM 功能	AWS Cloud9 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键 (特定于服务)	是
ACL	否
ABAC (策略中的标签)	是
临时凭证	是
转发访问会话 (FAS)	是
服务角色	是
服务相关角色	是

要全面了解 AWS Cloud9 以及其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM [配合使用的AWS 服务](#)。

基于身份的策略 AWS Cloud9

支持基于身份的策略	是
-----------	---

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅 IAM 用户指南中的[创建 IAM 策略](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

基于身份的策略示例 AWS Cloud9

要查看 AWS Cloud9 基于身份的策略的示例，请参阅。[适用于 AWS Cloud9 的基于身份的策略示例](#)

内部基于资源的政策 AWS Cloud9

支持基于资源的策略	否
-----------	---

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service (Amazon S3) 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户存取，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。

AWS Cloud9 不支持基于资源的策略，但您仍然可以通过 AWS Cloud9 API 和 AWS Cloud9 IDE 控制 AWS Cloud9 AWS Cloud9 环境成员的环境资源权限。

的政策行动 AWS Cloud9

支持策略操作	是
--------	---

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 AWS Cloud9 操作列表，请参阅《服务授权参考》AWS Cloud9中[定义的操作](#)。

正在执行的策略操作在操作前 AWS Cloud9 使用以下前缀：

```
account
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [  
  "account:action1",  
  "account:action2"  
]
```

要查看 AWS Cloud9 基于身份的策略的示例，请参阅。[适用于 AWS Cloud9的基于身份的策略示例](#)

的政策资源 AWS Cloud9

支持策略资源

是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 AWS Cloud9 资源类型及其 ARN 的列表，请参阅《服务授权参考》[AWS Cloud9中定义的资源](#)。要了解可以在哪些操作中指定每个资源的 ARN，请参阅[AWS Cloud9定义的操作](#)。

要查看 AWS Cloud9 基于身份的策略的示例，请参阅。[适用于 AWS Cloud9的基于身份的策略示例](#)
的策略条件密钥 AWS Cloud9

支持特定于服务的策略条件键	是
---------------	---

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素 (或 Condition 块) 中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用[条件运算符](#) (例如，等于或小于) 的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 策略元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

要查看 AWS Cloud9 条件键列表，请参阅《服务授权参考》AWS Cloud9中的[条件密钥](#)。要了解可以使用条件键的操作和资源，请参阅[由定义的操作 AWS Cloud9](#)。

要查看 AWS Cloud9 基于身份的策略的示例，请参阅。[适用于 AWS Cloud9的基于身份的策略示例](#)
输入的 ACL AWS Cloud9

支持 ACL	否
--------	---

访问控制列表 (ACL) 控制哪些主体 (账户成员、用户或角色) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

ABAC with AWS Cloud9

支持 ABAC (策略中的标签) 是

基于属性的访问控制 (ABAC) 是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以将标签附加到 IAM 实体 (用户或角色) 和许多 AWS 资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的 [条件元素](#) 中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息,请参阅《IAM 用户指南》中的[什么是 ABAC ?](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的[使用基于属性的访问权限控制 \(ABAC \)](#)。

将临时证书与 AWS Cloud9

支持临时凭证 是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的[AWS 服务与 IAM 配合使用的信息](#)。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的 [切换到角色 \(控制台 \)](#)。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 [IAM 中的临时安全凭证](#)。

转发访问会话 AWS Cloud9

支持转发访问会话 (FAS) 是

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详细信息，请参阅[转发访问会话](#)。

AWS Cloud9的服务角色

支持服务角色 是

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。

Warning

更改服务角色的权限可能会中断 AWS Cloud9 功能。只有在 AWS Cloud9 提供操作指导时才编辑服务角色。

的服务相关角色 AWS Cloud9

支持服务相关角色 是

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅[能够与 IAM 搭配使用的AWS 服务](#)。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

适用于 AWS Cloud9 的基于身份的策略示例

默认情况下，用户和角色没有创建或修改 AWS Cloud9 资源的权限。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅 IAM 用户指南中的 [创建 IAM 策略](#)。

有关由定义的操作和资源类型的详细信息 AWS Cloud9，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》AWS Cloud9 中的 [操作、资源和条件密钥](#)。

主题

- [策略最佳实践](#)
- [使用 AWS Cloud9 控制台](#)
- [允许用户查看他们自己的权限](#)

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS Cloud9 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限 – 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM

Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM Access Analyzer 策略验证](#)。

- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [配置受 MFA 保护的 API 访问](#)。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实操](#)。

使用 AWS Cloud9 控制台

要访问 AWS Cloud9 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您的 AWS Cloud9 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 AWS Cloud9 控制台，还需要将 AWS Cloud9 *ConsoleAccess* 或 *ReadOnly* AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的 [为用户添加权限](#)。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    }
  ]
}
```

```
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

对 AWS Cloud9 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS Cloud9 和 IAM 时可能遇到的常见问题。

主题

- [我无权在以下位置执行操作 AWS Cloud9](#)
- [我无权执行 iam : PassRole](#)
- [我想允许我以外的人 AWS 账户 访问我的 AWS Cloud9 资源](#)

我无权在以下位置执行操作 AWS Cloud9

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 *awes:GetWidget* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
awes:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 `aws:GetWidget` 操作访问 `my-example-widget` 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS Cloud9。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 AWS Cloud9 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 AWS Cloud9 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS Cloud9 支持这些功能，请参阅 [如何 AWS Cloud9 与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问 [权限 AWS 账户](#)，请参阅 [IAM 用户指南中的向第三方提供访问权限。AWS 账户](#)
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的 [为经过外部身份验证的用户 \(身份联合验证 \) 提供访问权限](#)。

- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

如何 AWS Cloud9 使用 IAM 资源和操作

AWS Identity and Access Management 用于管理允许您同时使用 AWS Cloud9 开发环境 AWS 服务 和其他资源的权限。

AWS Cloud9 资源和运营

在中 AWS Cloud9，主要资源是 AWS Cloud9 开发环境。在策略中，您可以使用 Amazon Resource Name (ARN) 标识策略应用到的资源。下表列出了环境 ARN。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#)。

资源类型	ARN 格式
环境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i></code>
指定 AWS 区域中的指定账户拥有的每个环境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :environment:*</code>
指定区域中的指定账户拥有的每个环境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :*</code>
每种 AWS Cloud9 资源，无论账户和地区如何	<code>arn:aws:cloud9:*</code>

例如，您可以在语句中使用 Amazon 资源名称 (ARN) 指示特定的环境，如下所示。

```
"Resource": "arn:aws:cloud9:us-east-2:123456789012:environment:70d899206236474f9590d93b7c41dfEX"
```

要指定所有资源，请在 Resource 元素中使用通配符 (*)。

```
"Resource": "*"
```

要在单个语句中指定多个资源，请使用逗号分隔其 Amazon 资源名称 (ARN)。

```
"Resource": [  
  "arn:aws:cloud9:us-east-2:123456789012:environment:70d899206236474f9590d93b7c41dfEX",  
  "arn:aws:cloud9:us-east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"  
]
```

AWS Cloud9 提供了一组使用 AWS Cloud9 资源的操作。有关列表，请参阅 [AWS Cloud9 权限参考](#)。

了解资源所有权

该 AWS 账户 账户拥有在账户中创建的资源，无论谁创建了这些资源。

请考虑以下使用案例和场景：

- 假设您使用您的根账户证书 AWS 账户 来创建 AWS Cloud9 开发环境。尽管可能，但不建议这样做。在这种情况下，您 AWS 账户 就是环境的所有者。
- 假设您在中创建了一个 IAM 用户，AWS 账户 并向该用户授予了创建环境的权限。然后，用户可以创建环境。但是，用户所属的您 AWS 账户 仍然拥有该环境。
- 假设您在中创建了一个 AWS 账户 拥有创建环境权限的 IAM 角色。然后，可以代入该角色的任何人都可以创建环境。角色所属的 AWS 账户 拥有此环境。

Note

如果您删除一个或多个 AWS Cloud9 环境的 ARN 所有者的用户账户，则这些环境将没有所有者。这种情况的解决方法是使用 AWS Cloud9 软件开发工具包通过 `CreateEnvironmentMembership` 操作和 `EnvironmentMember` 数据类型添加另一位具有读写权限的 IAM 用户。添加此 IAM 用户后，您可以将环境文件复制到新 AWS Cloud9 环境中，并将该所有者设为 ARN 所有者。有关此操作的更多信息 [CreateEnvironmentMembership](#)，请参阅；有关此数据类型的更多信息，请参阅 AWS Cloud9 API 参考指南 [EnvironmentMember](#) 中的。

管理对资源的访问

权限策略规定谁可以访问哪些资源。

Note

本节讨论如何在 AWS Cloud9 中使用 IAM。这里不提供有关 IAM 服务的详细信息。有关完整的 IAM 文档，请参阅 IAM 用户指南中的[什么是 IAM？](#)。有关 IAM 策略语法和说明的信息，请参阅 IAM 用户指南中[IAM JSON 策略参考](#)。

附加到 IAM 身份的策略称为基于身份的策略（或 IAM 策略）。附加到资源的策略称为基于资源的策略。AWS Cloud9 支持基于身份和基于资源的策略。

以下各项 API 操作仅要求将一个 IAM 策略附加到要调用这些 API 操作的 IAM 身份：

- `CreateEnvironmentEC2`
- `DescribeEnvironments`

以下 API 操作需要使用基于资源的策略。IAM 策略不是必需的，但如果它附加到想要调 AWS Cloud9 用这些 API 操作的 IAM 身份，则使用 IAM 策略。必须将基于资源的策略应用于所需的 AWS Cloud9 资源：

- `CreateEnvironmentMembership`
- `DeleteEnvironment`
- `DeleteEnvironmentMembership`
- `DescribeEnvironmentMemberships`
- `DescribeEnvironmentStatus`
- `UpdateEnvironment`
- `UpdateEnvironmentMembership`

有关每个 API 操作执行的任务的更多信息，请参阅《AWS Cloud9 API 参考》。

您不能将基于资源的策略直接附加到 AWS Cloud9 资源。相反，在添加、修改、更新或删除环境成员时，将相应的基于 AWS Cloud9 资源的策略 AWS Cloud9 附加到资源。

要向用户授予对 AWS Cloud9 资源执行操作的权限，您需要向该用户所属的 IAM 群组附加权限策略。我们建议您 AWS Cloud9 尽可能为其附加 AWS 托管（预定义）策略。AWS 托管策略包含针对常见使用场景和用户类型的预定义访问权限集，例如对环境的全面管理、环境用户以及对环境只有只读访问权限的用户。有关 AWS 托管策略的列表 AWS Cloud9，请参阅[AWS 的托管策略 AWS Cloud9](#)。

对于更详细的使用方案和独特的用户类型，您可以创建并附加自己的客户托管策略。请参阅 [适用于 AWS Cloud9 的其他设置选项（团队和企业）](#) 和 [为以下各项创建客户托管策略 AWS Cloud9](#)。

要将 IAM 策略（AWS 托管或客户托管）附加到 IAM 身份，请参阅 [IAM 用户指南中的附加 IAM 策略（控制台）](#)。

API 操作的会话权限

使用 AWS CLI 或 AWS API 以编程方式为角色或联合用户创建临时会话时，您可以将会话策略作为参数传递以扩展角色会话的范围。这意味着会话的有效权限是 [角色基于身份的策略与会话策略的交集](#)。

在会话期间请求访问资源时，如果没有适用的 Deny 语句，也没有适用的 Allow 语句，则策略评估的结果是 [隐式拒绝](#)。（有关更多信息，请参阅 IAM 用户指南中的 [确定账户内请求是否被允许或拒绝](#)。）

但是，对于需要基于资源的策略的 AWS Cloud9 API 操作（见上文），如果资源策略 Principal 中指定为，则会向正在调用的 IAM 实体授予权限。此显式权限优先于隐式拒绝会话策略，从而允许会话成功调用 AWS Cloud9 API 操作。

AWS 的托管策略 AWS Cloud9

AWS 托管策略是由创建和管理的独立策略 AWS。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的 [客户管理型策略](#) 来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#)。

AWS 托管策略：AWSCloud9Administrator

您可以将 AWSCloud9Administrator 策略附加到 IAM 身份。

此策略授予 ## 权限，提供管理员访问权限 AWS Cloud9。

权限详细信息

该策略包含以下权限。

- AWS Cloud9 — 所有 AWS Cloud9 动作都在他们身上 AWS 账户。
- Amazon EC2 — 获取有关其中的多个 Amazon VPC 和子网资源的信息 AWS 账户。
- IAM — 获取有关其中的 IAM 用户的信息 AWS 账户，并根据需要在他们 AWS 账户 中创建 AWS Cloud9 服务相关角色。
- Systems Manager-允许用户 StartSession 通过呼叫启动与实例的连接以进行会话管理器会话。打开通过 Systems Manager 与其 EC2 实例通信的环境的用户需要此权限。有关更多信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:*",
        "iam:GetUser",
        "iam:ListUsers",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "cloud9.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:StartSession",
        "ssm:GetConnectionStatus"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringLike": {
        "ssm:resourceTag/aws:cloud9:environment": "*"
      },
      "StringEquals": {
        "aws:CalledViaFirst": "cloud9.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession"
    ],
    "Resource": [
      "arn:aws:ssm:*:*:document/*"
    ]
  }
]
}

```

AWS 托管策略：AWSCloud9User

您可以将 AWSCloud9User 策略附加到 IAM 身份。

此策略授予##进行创建 AWS Cloud9 开发环境和管理拥有的环境的权限。

权限详细信息

该策略包含以下权限。

- AWS Cloud9 — 创建和获取有关其环境的信息，以及获取和更改其环境的用户设置。
- Amazon EC2 — 获取有关其中的多个 Amazon VPC 和子网资源的信息 AWS 账户。
- IAM — 获取有关其中的 IAM 用户的信息 AWS 账户，并根据需要在他们 AWS 账户 中创建 AWS Cloud9 服务相关角色。
- Systems Manager-允许用户 StartSession 通过呼叫启动与实例的连接以进行会话管理器会话。打开通过 Systems Manager 与其 EC2 实例通信的环境的用户需要此权限。有关更多信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:UpdateUserSettings",
        "cloud9:GetUserSettings",
        "iam:GetUser",
        "iam:ListUsers",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentEC2",
        "cloud9:CreateEnvironmentSSH"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloud9:OwnerArn": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:GetUserPublicKey"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloud9:UserArn": "true"
        }
      }
    }
  ],
  {
```

```

    "Effect": "Allow",
    "Action": [
      "cloud9:DescribeEnvironmentMemberships"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "Null": {
        "cloud9:UserArn": "true",
        "cloud9:EnvironmentId": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "cloud9.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession",
      "ssm:GetConnectionStatus"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringLike": {
        "ssm:resourceTag/aws:cloud9:environment": "*"
      },
      "StringEquals": {
        "aws:CalledViaFirst": "cloud9.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",

```

```

        "Action": [
            "ssm:StartSession"
        ],
        "Resource": [
            "arn:aws:ssm:*:*:document/*"
        ]
    }
]
}

```

AWS 托管策略：AWSCloud9EnvironmentMember

您可以将 AWSCloud9EnvironmentMember 策略附加到 IAM 身份。

此策略授予##权限，从而能够加入 AWS Cloud9 共享环境。

权限详细信息

该策略包含以下权限：

- AWS Cloud9 — 获取有关其环境的信息，并获取和更改其环境的用户设置。
- IAM — 获取有关其中的 IAM 用户的信息 AWS 账户，并根据需要在他们 AWS 账户 中创建 AWS Cloud9 服务相关角色。
- Systems Manager-允许用户 StartSession 通过呼叫启动与实例的连接以进行会话管理器会话。打开通过 Systems Manager 与其 EC2 实例通信的环境的用户需要此权限。有关更多信息，请参阅 [使用 AWS Systems Manager 访问非入口 EC2 实例](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:GetUserSettings",
        "cloud9:UpdateUserSettings",
        "iam:GetUser",
        "iam:ListUsers"
      ],
      "Resource": "*"
    },
    {

```

```

    "Effect": "Allow",
    "Action": [
      "cloud9:DescribeEnvironmentMemberships"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "Null": {
        "cloud9:UserArn": "true",
        "cloud9:EnvironmentId": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession",
      "ssm:GetConnectionStatus"
    ],
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringLike": {
        "ssm:resourceTag/aws:cloud9:environment": "*"
      },
      "StringEquals": {
        "aws:CalledViaFirst": "cloud9.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:StartSession"
    ],
    "Resource": [
      "arn:aws:ssm:*:*:document/*"
    ]
  }
]
}

```

AWS 托管策略 : **AWSCloud9ServiceRolePolicy**

服务相关角色 `AWSServiceRoleForAWSCloud9` 使用此策略允许 AWS Cloud9 环境与 Amazon EC2 和 AWS CloudFormation 资源进行交互。

权限详细信息

`AWSCloud9ServiceRolePolicy` 授予 `AWSServiceRoleForAWSCloud9` 9 个必要的权限，AWS Cloud9 允许他们与创建和运行开发环境所需的 AWS 服务（Amazon EC2 和 AWS CloudFormation）进行交互。

AWS Cloud9 定义其服务相关角色的权限，并且 AWS Cloud9 只能担任其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

有关如何 AWS Cloud9 使用服务相关角色的更多信息，请参阅 [将服务相关角色用于 AWS Cloud9](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "cloudformation:CreateStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResources"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:TerminateInstances",
        "ec2>DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress"
      ],
    }
  ]
}
```



```
"Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "cloudformation:DeleteStack"
  ],
  "Resource": "arn:aws:cloudformation:*:*:stack/aws-cloud9-*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:security-group/*"
  ],
  "Condition": {
    "StringLike": {
      "aws:RequestTag/Name": "aws-cloud9-*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "ec2:ResourceTag/aws:cloudformation:stack-name": "aws-cloud9-*"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances"
  ],
  "Resource": [
```

```

    "arn:aws:license-manager:*:*:license-configuration:*"
  ],
},
{
  "Effect": "Allow",
  "Action": [
    "iam:ListInstanceProfiles",
    "iam:GetInstanceProfile"
  ],
  "Resource": [
    "arn:aws:iam:*:*:instance-profile/cloud9/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam:*:*:role/service-role/AWSCloud9SSMAccessRole"
  ],
  "Condition": {
    "StringLike": {
      "iam:PassedToService": "ec2.amazonaws.com"
    }
  }
}
]
}

```

AWS Cloud9AWS 托管策略的更新

查看 AWS Cloud9 自该服务开始跟踪这些更改以来 AWS 托管策略更新的详细信息。要获得有关此页面变更的自动提醒，请订阅“AWS Cloud9 文档历史记录”页面上的 RSS feed。

更改	描述	日期
已向AWSCloud9 User、AWSCloud9Administrator和AWSCloud9EnvironmentMember策略中添加了新操作。	该ssm:GetConnectionStatus 操作已添加到AWSCloud9 UserAWSCloud9Administrator和AWSCloud9EnvironmentMember策略中。	2023 年 10 月 12 日

更改	描述	日期
	ntMember策略。此操作将授予用户检查 SSM 连接状态的权限。该 cloud9:ValidateEnvironmentName API 已从AWSCloud9User政策中删除，因为它已被弃用。	
API 已添加到AWSCloud9User和AWSCloud9Administrator政策。	AWSCloud9User和AWSCloud9Administrator策略中添加了两个新API，这些API是ec2:DescribeInstanceTypeOfferings和ec2:DescribeRouteTables。这些API的AWS Cloud9目的是验证默认子网是否支持客户在创建AWS Cloud9环境时选择的实例类型。	2023年8月02日
更新到 AWSCloud9ServiceRolePolicy	AWSCloud9ServiceRolePolicy 已更新，AWS Cloud9允许启动和停止由License Manager许可证配置管理的Amazon EC2实例。	2022年1月12日
AWS Cloud9 开始跟踪更改	AWS Cloud9 开始跟踪其 AWS 托管策略的更改。	2021年3月15日

为以下各项创建客户托管策略 AWS Cloud9

如果所有 AWS 托管策略都不符合您的访问控制要求，则可以创建并附加自己的客户托管策略。

要创建客户托管策略，请参阅 IAM 用户指南中的 [创建 IAM 策略 \(控制台\)](#)。

主题

- [指定策略元素：效果、委托人、操作和资源](#)
- [客户管理型策略示例](#)

指定策略元素：效果、委托人、操作和资源

对于每种 AWS Cloud9 资源，该服务都定义了一组 API 操作。要授予这些 API 操作的权限，请 AWS Cloud9 定义一组可在策略中指定的操作。

以下是基本的策略元素：

- **Effect** – 用于指定在用户请求操作时的效果（可以是允许或拒绝）。如果没有显式授予（允许）对资源的访问权限，则隐式拒绝访问。您也可显式拒绝对资源的访问。您可以执行该操作以确保用户无法访问资源，即使其他策略授予了访问权限。
- **Principal** – 在基于身份的策略（IAM 策略）中，策略附加到的用户是隐式委托人。对于基于资源的策略，您可以指定您希望将权限授予的用户、账户、服务或其他实体。
- **Resource** – 使用 Amazon 资源名称（ARN）以确定策略应用到的资源。
- **Action** – 使用操作关键字来标识要允许或拒绝的资源操作。例如，`cloud9:CreateEnvironmentEC2` 权限授予用户执行 `CreateEnvironmentEC2` 操作的权限。

有关 IAM 策略语法和介绍的更多信息，请参阅 IAM 用户指南的 [IAM JSON 策略参考](#)。

有关显示所有 AWS Cloud9 API 操作及其适用的资源的表格，请参阅 [AWS Cloud9 权限参考](#)。

客户管理型策略示例

在本节中，您可以找到为 AWS Cloud9 操作授予权限的示例策略。您可以修改以下示例 IAM 策略以允许或明确拒绝您的 IAM 身份访问 AWS Cloud9。

要创建客户托管式策略或将其附加到 IAM 身份，请参阅 IAM 用户指南中的 [创建 IAM 策略（控制台）](#) 和 [附加 IAM 策略（控制台）](#)。

Note

以下示例使用美国东部（俄亥俄州）区域（`us-east-2`）、虚构的 AWS 账户 ID（`123456789012`）和虚构的 AWS Cloud9 开发环境 ID（`81e900317347585a0601e04c8d52eaEX`）。

主题

- [获取有关环境的信息](#)
- [创建 EC2 环境](#)
- [使用特定的 Amazon EC2 实例类型创建 EC2 环境](#)
- [在特定的 Amazon VPC 子网中创建 EC2 环境](#)
- [使用特定的环境名称创建 EC2 环境](#)
- [仅创建 SSH 环境](#)
- [更新环境或禁止更新环境](#)
- [获取环境成员列表](#)
- [仅与特定用户共享环境](#)
- [禁止共享环境](#)
- [更改或禁止更改环境成员设置](#)
- [删除或禁止删除环境成员](#)
- [删除或阻止删除环境](#)
- [用于创建 SSM 环境的自定义 IAM 策略](#)

获取有关环境的信息

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体获取有关其账户中的任何环境的信息。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:DescribeEnvironments",
      "Resource": "*"
    }
  ]
}
```

Note

上述访问权限已包含在 AWS 托管策略 `AWSCloud9Administrator` 和 `AWSCloud9User`。

创建 EC2 环境

以下 IAM 策略声明示例（附加到 IAM 实体）允许该实体在其账户中创建 AWS Cloud9 EC2 开发环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*"
    }
  ]
}
```

Note

上述访问权限已包含在 AWS 托管策略 `AWSCloud9Administrator` 和 `AWSCloud9User`。

使用特定的 Amazon EC2 实例类型创建 EC2 环境

以下 IAM 策略声明示例（附加到 IAM 实体）允许该实体在其账户中创建 AWS Cloud9 EC2 开发环境。不过，EC2 环境只能使用指定类别的 Amazon EC2 实例类型。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "cloud9:InstanceType": "t3.*"
        }
      }
    }
  ]
}
```

Note

如果 AWS 托管策略 `AWSCloud9Administrator` 或已附加到 IAM 实体，`AWSCloud9User` 则该 AWS 托管策略将覆盖前面的 IAM 策略声明的行为。这是因为这些 AWS 托管策略更宽松。

在特定的 Amazon VPC 子网中创建 EC2 环境

以下 IAM 策略声明示例（附加到 IAM 实体）允许该实体在其账户中创建 AWS Cloud9 EC2 开发环境。不过，EC2 环境只能使用指定的 Amazon VPC 子网。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "cloud9:SubnetId": [
            "subnet-12345678",
            "subnet-23456789"
          ]
        }
      }
    }
  ]
}
```

Note

如果 AWS 托管策略 `AWSCloud9Administrator` 或已附加到 IAM 实体，`AWSCloud9User` 则该 AWS 托管策略将覆盖前面的 IAM 策略声明的行为。这是因为这些 AWS 托管策略更宽松。

使用特定的环境名称创建 EC2 环境

以下 IAM 策略声明示例（附加到 IAM 实体）允许该实体在其账户中创建 AWS Cloud9 EC2 开发环境。不过，EC2 环境只能使用指定的名称。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloud9:EnvironmentName": "my-demo-environment"
        }
      }
    }
  ]
}
```

Note

如果 AWS 托管策略 `AWSCloud9Administrator` 或已附加到 IAM 实体，`AWSCloud9User` 则该 AWS 托管策略将覆盖前面的 IAM 策略声明的行为。这是因为这些 AWS 托管策略更宽松。

仅创建 SSH 环境

以下 IAM 策略声明示例（附加到 IAM 实体）允许该实体在其账户中创建 AWS Cloud9 SSH 开发环境。但是，该实体无法创建 AWS Cloud9 EC2 开发环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentSSH",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*"
    }
  ]
}
```



```
}
```

更新环境或禁止更新环境

以下 IAM 政策声明示例（附加到 IAM 实体）允许该实体更改有关其账户中任何 AWS Cloud9 开发环境的信息。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:UpdateEnvironment",
      "Resource": "*"
    }
  ]
}
```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。

以下示例 IAM policy 语句附加到一个 IAM 实体，此语句显式防止该实体更改有关具有指定 Amazon 资源名称（ARN）的环境的信息。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:UpdateEnvironment",
      "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}
```

获取环境成员列表

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体获取其账户中的任何环境的成员列表。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:DescribeEnvironmentMemberships",
      "Resource": "*"
    }
  ]
}
```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。此外，前面的访问权限比 AWS 托管策略 `AWSCloud9User` 中的等效访问权限更具宽松性。

仅与特定用户共享环境

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体仅与指定用户共享其账户中的任何环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentMembership"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloud9:UserArn": "arn:aws:iam::123456789012:user/MyDemoUser"
        }
      }
    }
  ]
}
```

```

    }
  ]
}

```

Note

如果 AWS 托管策略 `AWSCloud9Administrator` 或已附加到 IAM 实体，`AWSCloud9User` 则这些 AWS 托管策略将覆盖前面的 IAM 策略声明的行为。这是因为这些 AWS 托管策略更宽松。

禁止共享环境

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句禁止该实体共享其账户中的任何环境。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloud9:CreateEnvironmentMembership",
        "cloud9:UpdateEnvironmentMembership"
      ],
      "Resource": "*"
    }
  ]
}

```

更改或禁止更改环境成员设置

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体更改其账户的任何环境中的成员设置。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:UpdateEnvironmentMembership",
      "Resource": "*"
    }
  ]
}

```

```
    }  
  ]  
}
```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。

以下示例 IAM policy 语句附加到一个 IAM 实体，此语句显式防止该实体更改具有指定 Amazon 资源名称 (ARN) 的环境中的成员设置。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Deny",  
      "Action": "cloud9:UpdateEnvironmentMembership",  
      "Resource": "arn:aws:cloud9:us-  
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"  
    }  
  ]  
}
```

删除或禁止删除环境成员

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体从其账户的任何环境中删除任何成员。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "cloud9:DeleteEnvironmentMembership",  
      "Resource": "*"   
    }  
  ]  
}
```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。

以下示例 IAM policy 语句附加到一个 IAM 实体，此语句显式防止该实体从具有指定 Amazon 资源名称 (ARN) 的环境中删除任何成员。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:DeleteEnvironmentMembership",
      "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}
```

删除或阻止删除环境

以下示例 IAM 策略语句附加到一个 IAM 实体，此语句允许该实体删除其账户中的任何环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:DeleteEnvironment",
      "Resource": "*"
    }
  ]
}
```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。

以下示例 IAM policy 语句附加到一个 IAM 实体，此语句显式防止该实体删除具有指定 Amazon 资源名称 (ARN) 的环境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:DeleteEnvironment",
      "Resource": "arn:aws:cloud9:us-east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}
```

用于创建 SSM 环境的自定义 IAM 策略

当前存在一个权限问题，会在创建附加 AWSCloud9Administrator 或 AWSCloud9User 策略的 SSM 环境时出现。以下示例 IAM 策略声明在附加到 IAM 实体时，允许用户附加和使用 AWS 托管策略 AWSCloud9Administrator 或 AWSCloud9User。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:UpdateUserSettings",
        "cloud9:GetUserSettings",
        "iam:GetUser",
        "iam:ListUsers",
        "iam:ListRoles",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentEC2",
        "cloud9:CreateEnvironmentSSH"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "cloud9:OwnerArn": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "cloud9:GetUserPublicKey"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "cloud9:UserArn": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "cloud9:DescribeEnvironmentMemberships"
    ],
    "Resource": [
      "*"
    ],
    "Condition": {
      "Null": {
        "cloud9:UserArn": "true",
        "cloud9:EnvironmentId": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "cloud9.amazonaws.com"
      }
    }
  }
}

```

```

    }
  }
},
{
  "Effect": "Allow",
  "Action": "ssm:StartSession",
  "Resource": "arn:aws:ec2:*:*:instance/*",
  "Condition": {
    "StringLike": {
      "ssm:resourceTag/aws:cloud9:environment": "*"
    },
    "StringEquals": {
      "aws:CalledViaFirst": "cloud9.amazonaws.com"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:StartSession"
  ],
  "Resource": [
    "arn:aws:ssm:*:*:document/*"
  ]
},
{
  "Effect": "Allow",
  "Action": ["iam:ListInstanceProfilesForRole", "iam:CreateRole"],
  "Resource": ["arn:aws:iam:*:*:role/service-role/AWSCloud9SSMAccessRole"]
},
{
  "Effect": "Allow",
  "Action": ["iam:AttachRolePolicy"],
  "Resource": ["arn:aws:iam:*:*:role/service-role/AWSCloud9SSMAccessRole"],
  "Condition": {
    "StringEquals": {
      "iam:PolicyARN": "arn:aws:iam::aws:policy/
AWSCloud9SSMInstanceProfile"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "iam:PassRole",

```



```

    "Resource": "arn:aws:iam::*:role/service-role/AWSCloud9SSMAccessRole",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "ec2.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:CreateInstanceProfile",
      "iam:AddRoleToInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::*:instance-profile/cloud9/AWSCloud9SSMInstanceProfile"
    ]
  }
]
}

```

AWS Cloud9 权限参考

您可以在 AWS Cloud9 策略中使用 AWS 宽条件密钥来表达条件。有关列表，请参阅 IAM 用户指南中的 [IAM JSON 策略元素：条件](#)。

请在策略的 Action 字段中指定这些操作。要指定操作，请在 API 操作名称之前使用 cloud9: 前缀（例如，"Action": "cloud9:DescribeEnvironments"）。要在单个语句中指定多项操作，请使用逗号将它们隔开（例如，"Action": ["cloud9:UpdateEnvironment", "cloud9>DeleteEnvironment"]）。

使用通配符

您可以在策略的 * 字段中将带或不带通配符 (Resource) 的 ARN 指定为资源值。您可以使用通配符指定多个操作或资源。例如，cloud9:* 指定所有 AWS Cloud9 操作并 cloud9:Describe* 指定以开头的所有 AWS Cloud9 操作 Describe。

以下示例允许 IAM 实体获取有关其账户中的任何环境和环境成员的信息。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Effect": "Allow",
    "Action": [
      "cloud9:Describe*"
    ],
    "Resource": "*"
  }
]
}

```

Note

上述访问权限已包含在 AWS 托管策略中 `AWSCloud9Administrator`。此外，前面的访问权限比 AWS 托管策略 `AWSCloud9User` 中的等效访问权限更具宽松性。

AWS Cloud9 API 操作和操作所需的权限

Note

在设置访问控制以及编写要附加到 IAM 身份的权限策略（基于身份的策略）时，可以使用下表作为参考。

[Public API operations](#) 表列出了使用开发工具包和 AWS Command Line Interface 可由客户调用的 API 操作。

[Permission-only API operations](#) 列出了客户代码或 AWS Command Line Interface 无法直接调用的 API 操作。但 IAM 用户确实需要这些操作的权限，这些操作在执行 AWS Cloud9 操作时使用控制台调用。

公有 API 操作

AWS Cloud9 操作	所需的权限 (API 操作)	资源
CreateEnvironmentEC2	cloud9:CreateEnvironmentEC2 创建 AWS Cloud9 EC2 开发环境所必需的。	*
CreateEnvironmentMembership	cloud9:CreateEnvironmentMembership	arn:aws:cloud9:REGION_ID

AWS Cloud9 操作	所需的权限 (API 操作)	资源
	需要使用该权限在环境中添加成员。	: <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
DeleteEnvironment	cloud9:DeleteEnvironment 需要使用该权限删除环境。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
DeleteEnvironmentMembership	cloud9:DeleteEnvironmentMembership 需要使用该权限从环境中移除成员。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
DescribeEnvironmentMemberships	cloud9:DescribeEnvironmentMemberships 需要使用该权限以获取环境中的成员列表。	*
DescribeEnvironments	cloud9:DescribeEnvironments 需要使用该权限以获取有关环境的信息。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
DescribeEnvironmentStatus	cloud9:DescribeEnvironmentStatus 需要使用该权限以获取有关环境状态的信息。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
UpdateEnvironment	cloud9:UpdateEnvironment 需要使用该权限以更新环境设置。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>

AWS Cloud9 操作	所需的权限 (API 操作)	资源
UpdateEnvironmentMembership	cloud9:UpdateEnvironmentMembership 需要使用该权限以更新环境中的成员设置。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>

仅限 权限的 API 操作

AWS Cloud9 操作	描述	控制台文档
ActivateEC2Remote	cloud9:ActivateEC2Remote 启动您的 AWS Cloud9 IDE 连接到的 Amazon EC2 实例。	在 AWS Cloud9 中打开环境
CreateEnvironmentSSH	cloud9:CreateEnvironmentSSH 创建 AWS Cloud9 SSH 开发环境。	创建 SSH 环境
CreateEnvironmentToken	cloud9:CreateEnvironmentToken 创建身份验证令牌，以允许 AWS Cloud9 IDE 和用户环境之间的连接。	创建 EC2 环境
DescribeEC2Remote	cloud9:DescribeEC2Remote 获取有关 EC2 开发环境 (包括主机、用户和端口) 连接详细信息。	创建 EC2 环境

AWS Cloud9 操作	描述	控制台文档
DescribeSSHRemote	<p>cloud9:DescribeSSHRemote</p> <p>获取有关 SSH 开发环境 (包括主机、用户和端口) 连接详细信息。</p>	创建 SSH 环境
GetEnvironmentConfig	<p>cloud9:GetEnvironmentConfig</p> <p>获取用于初始化 AWS Cloud9 IDE 的配置信息。</p>	使用 AWS Cloud9 集成开发环境 (IDE)
GetEnvironmentSettings	<p>cloud9:GetEnvironmentSettings</p> <p>获取指定开发环境的 AWS Cloud9 IDE 设置。</p>	使用 AWS Cloud9 集成开发环境 (IDE)
GetMembershipSettings	<p>cloud9:GetMembershipSettings</p> <p>获取指定环境成员的 AWS Cloud9 IDE 设置。</p>	在 AWS Cloud9 中使用共享环境
GetUserPublicKey	<p>cloud9:GetUserPublicKey</p> <p>获取用户的 SSH 公钥, 该密钥用于连接 AWS Cloud9 到 SSH 开发环境。</p>	创建 SSH 环境
GetUserSettings	<p>cloud9:GetUserSettings</p> <p>获取指定用户的 AWS Cloud9 IDE 设置。</p>	使用 AWS Cloud9 集成开发环境 (IDE)

AWS Cloud9 操作	描述	控制台文档
ModifyTemporaryCredentialsOnEnvironmentEC2	<p>cloud9:ModifyTemporaryCredentialsOnEnvironmentEC2</p> <p>在 AWS Cloud9 集成开发环境 (IDE) 使用的 Amazon EC2 实例上设置 AWS 托管临时证书。</p>	AWS 托管的临时证书
UpdateEnvironmentSettings	<p>cloud9:UpdateEnvironmentSettings</p> <p>更新指定开发环境的 AWS Cloud9 IDE 设置。</p>	使用 AWS Cloud9 集成开发环境 (IDE)
UpdateMembershipSettings	<p>cloud9:UpdateMembershipSettings</p> <p>更新指定环境成员的 AWS Cloud9 IDE 设置。</p>	在 AWS Cloud9 中使用共享环境
UpdateSSHRemote	<p>cloud9:UpdateSSHRemote</p> <p>更新有关 SSH 开发环境的 (包括主机、用户和端口) 连接详细信息。</p>	创建 SSH 环境
UpdateUserSettings	<p>cloud9:UpdateUserSettings</p> <p>更新指定用户的 AWS Cloud9 IDE 设置。</p>	使用 AWS Cloud9 集成开发环境 (IDE)

AWS Cloud9 操作	描述	控制台文档
GetMigrationExperiences	<code>cloud9:GetMigrationExperiences</code> 向 AWS Cloud9 用户授予从 AWS Cloud9 到的迁移体验的权限 CodeCatalyst。	

AWS 托管的临时证书

如果您只是在寻找 AWS 托管临时证书支持的操作列表，请直接跳至[AWS 托管临时证书支持的操作](#)。

对于 AWS Cloud9 EC2 开发 AWS Cloud9 环境，您可以在环境中 AWS 使用临时访问证书。我们将这些凭证称为 AWS 托管式临时凭证。这些凭证具有以下优势：

- 您无需在环境中的任何地方存储实 AWS 体（例如 IAM 用户）的永久 AWS 访问证书。这会在您不知情和未批准的情况下禁止环境成员访问这些凭证。
- 您不需要手动设置或管理实例配置文件或将其附加到连接到环境的 Amazon EC2 实例。实例配置文件是管理临时 AWS 访问证书的另一种方法。
- AWS Cloud9 不断更新其临时证书，因此一组证书只能在有限的时间内使用。这是一种 AWS 安全最佳实践。有关更多信息，请参阅 [创建和更新 AWS 托管临时证书](#)。
- AWS Cloud9 对如何使用其临时证书从环境中访问 AWS 操作和资源施加了额外限制。这也是一种 AWS 安全最佳实践。

Important

目前，如果您的环境的 EC2 实例启动到私有子网，则无法使用 AWS 托管临时证书允许 EC2 环境代表 AWS 实体（例如，IAM 用户）访问 AWS 服务。

有关何时可以将 EC2 实例启动到私有子网中的更多信息，请参阅 [为创建子网 AWS Cloud9](#)。

Note

在使用 AWS 托管临时证书时，请考虑使用 AWS 托管策略而不是内联策略。

每当 EC2 环境尝试代表 AWS 实体（例如，IAM 用户）访问时，AWS 托管临时证书的工作原理如下：AWS 服务

1. AWS Cloud9 检查调用 AWS 实体（例如，IAM 用户）是否有权对中请求的资源执行请求的操作 AWS。如果没有或明确拒绝该权限，请求将失败。
 2. AWS Cloud9 检查 AWS 托管的临时证书，以查看其权限是否允许对中请求的资源执行请求的操作 AWS。如果没有或明确拒绝该权限，请求将失败。有关 AWS 托管临时证书支持的权限列表，请参阅[AWS 托管临时证书支持的操作](#)。
- 如果 AWS 实体和 AWS 托管临时证书都允许对请求的资源执行请求的操作，则请求成功。
 - 如果 AWS 实体或 AWS 托管临时证书明确拒绝或未明确允许对请求的资源执行请求的操作，则请求将失败。这意味着，即使调用 AWS 实体具有正确的权限，如果 AWS Cloud9 不明确允许，请求也会失败。同样，如果 AWS Cloud9 允许对特定资源采取特定操作，则如果该 AWS 实体不明确允许，则请求将失败。

EC2 环境的所有者可以随时为该环境开启或关闭 AWS 托管临时证书，如下所示：

1. 打开环境后，在 AWS Cloud9 IDE 的菜单栏中选择 AWS Cloud9“首选项”。
2. 在 Preferences (首选项) 选项卡中，在导航窗格中选择 AWS Settings > Credentials (Amazon 设置 > 凭证)。
3. 使用 AWS 托管式临时凭证以启用或禁用 AWS 托管式临时凭证。

Note

您还可以通过调用 AWS Cloud9 API 操作[UpdateEnvironment](#)并为 managedCredentialsAction 参数分配值来开启或关闭 AWS 托管临时证书。您可以使用标准 AWS 工具（例如 AWS SDK 和）来请求此 API 操作。AWS CLI

如果您关闭 AWS 托管临时证书，则无论是哪个 AWS 实体提出请求 AWS 服务，环境都无法访问任何证书。但是，假设你不能或不想为环境开启 AWS 托管临时证书，但你仍然需要环境才能访问 AWS 服务。则考虑以下替代方案：

- 将实例配置文件附加到连接到环境的 Amazon EC2 实例。有关说明，请参阅[创建和使用实例配置文件管理临时凭证](#)。
- 例如，通过设置特殊的环境变量或运行 `aws configure` 命令将您的永久 AWS 访问凭证存储在环境中。有关说明，请参阅[在环境中创建和存储永久访问凭证](#)。

上述替代方法覆盖 EC2 环境中 AWS 托管式临时凭证允许（或拒绝）的所有权限。

AWS 托管临时证书支持的操作

对于 AWS Cloud9 EC2 开发环境，AWS 托管临时证书允许对调用者中的所有 AWS 资源执行所有 AWS 操作 AWS 账户，但有以下限制：

- 对于 AWS Cloud9，仅允许执行以下操作：
 - `cloud9:CreateEnvironmentEC2`
 - `cloud9:CreateEnvironmentSSH`
 - `cloud9:DescribeEnvironmentMemberships`
 - `cloud9:DescribeEnvironments`
 - `cloud9:DescribeEnvironmentStatus`
 - `cloud9:UpdateEnvironment`
- 对于 IAM，仅允许执行以下操作：
 - `iam:AttachRolePolicy`
 - `iam:ChangePassword`
 - `iam:CreatePolicy`
 - `iam:CreatePolicyVersion`
 - `iam:CreateRole`
 - `iam:CreateServiceLinkedRole`
 - `iam>DeletePolicy`
 - `iam>DeletePolicyVersion`
 - `iam>DeleteRole`
 - `iam>DeleteRolePolicy`

- iam:DeleteSSHPublicKey
 - iam:DetachRolePolicy
 - iam:GetInstanceProfile
 - iam:GetPolicy
 - iam:GetPolicyVersion
 - iam:GetRole
 - iam:GetRolePolicy
 - iam:GetSSHPublicKey
 - iam:GetUser
 - iam:List*
 - iam:PassRole
 - iam:PutRolePolicy
 - iam:SetDefaultPolicyVersion
 - iam:UpdateAssumeRolePolicy
 - iam:UpdateRoleDescription
 - iam:UpdateSSHPublicKey
 - iam:UploadSSHPublicKey
- 仅以 Cloud9- 开头的角色名称允许执行与角色交互的所有 IAM 操作。不过，iam:PassRole 适用于所有角色名称。
 - 对于 AWS Security Token Service (AWS STS)，仅允许执行以下操作：
 - sts:GetCallerIdentity
 - sts:DecodeAuthorizationMessage
 - 所有支持的 AWS 操作仅限于环境的 IP 地址。这是一种 AWS 安全最佳实践。

如果 AWS Cloud9 不支持您需要 EC2 环境才能访问的操作或资源，或者如果 EC2 环境的 AWS 托管临时证书已关闭，而您无法将其重新打开，请考虑以下替代方案：

- 将实例配置文件附加到连接到 EC2 环境的 Amazon EC2 实例。有关说明，请参阅[创建和使用实例配置文件管理临时凭证](#)。
- 例如，通过设置特殊环境变量或运行aws configure命令将您的永久 AWS 访问凭证存储在 EC2 环境中。有关说明，请参阅[在环境中创建和存储永久访问凭证](#)。

上述替代方法覆盖 EC2 环境中 AWS 托管式临时凭证允许（或拒绝）的所有权限。

创建和更新 AWS 托管临时证书

对于 AWS Cloud9 EC2 开发环境，AWS 托管临时证书是在您首次打开环境时创建的。

AWS 在以下任一条件下更新托管临时证书：

- 每次经过特定的一段时间。目前，该时间段为 5 分钟。
- 每次重新加载显示环境 IDE 的 Web 浏览器选项卡时。
- 在到达环境的 `~/.aws/credentials` 文件中列出的时间戳时。
- 如果每次重新启用设置时，AWS 托管式临时凭证设置为禁用。（要查看或更改该设置，请选择 IDE 的菜单栏中的 AWS Cloud9，Preferences（首选项）。在 Preferences（首选项）选项卡上，在导航窗格中，选择 AWS Settings > Credentials（Amazon 设置 > 凭证）。）
- 出于安全考虑，AWS 托管临时证书会在 15 分钟后自动过期。要刷新凭证，环境所有者必须通过 IDE 连接到 AWS Cloud9 环境。有关环境所有者角色的更多信息，请参阅 [控制对 AWS 托管式临时凭证的访问](#)。

控制对 AWS 托管式临时凭证的访问

拥有 AWS 托管临时证书的合作者可以使用与其他 AWS Cloud9 AWS 服务人互动。为了确保仅向受信任的协作者提供 AWS 托管式临时凭证，如果环境所有者以外的任何人添加了新成员，则这些凭证将被禁用。通过删除 `~/.aws/credentials` 文件禁用凭证。

Important

AWS 托管临时证书也每 15 分钟自动过期。要刷新凭据以便协作者可以继续使用它们，必须通过 IDE 将环境所有者连接到 AWS Cloud9 环境。

只有环境所有者才能重新启用 AWS 托管临时证书，以便与其他成员共享这些证书。当环境所有者打开 IDE 时，会出现一个对话框，确认 AWS 托管临时凭证已禁用。环境所有者可以为所有成员重新启用凭证，或禁用所有成员的凭证。

Warning

要遵守最佳安全实践，如果您不确定上一个添加到环境中的用户的身份，请保持禁用托管式临时凭证。您可以在 [Collaborate（协作）](#) 窗口中检查有读/写权限的成员列表。

登录和监控 AWS Cloud9

使用监控活动 CloudTrail

AWS Cloud9 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在中执行的操作的记录 AWS Cloud9。CloudTrail 将所有 API 调用捕获 AWS Cloud9 为事件。捕获的调用包括来自 AWS Cloud9 控制台的调用和对 AWS Cloud9 API 的代码调用。

如果您创建跟踪，则可以允许持续向亚马逊简单存储服务 (Amazon S3) Storage Service 存储桶传送 CloudTrail 事件，包括的事件。AWS Cloud9

如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的“事件历史记录”中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向哪个请求发出 AWS Cloud9、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

有关更多信息，请参阅 [使用 AWS Cloud9 记录 AWS CloudTrail API 调用](#)。

监控 EC2 环境性能

如果您使用的是 AWS Cloud9 EC2 开发环境，则可以监控关联的 Amazon EC2 实例的可靠性、可用性和性能。例如，使用实例状态监控，您可以快速确定 Amazon EC2 是否已经检测到可能阻止您的实例运行应用程序的任何问题。

有关更多信息，请参阅 [亚马逊 EC2 用户指南中的监控 Amazon EC2](#)。

合规性验证 AWS Cloud9

作为多个合规计划的一部分，第三方审计师对 AWS 服务的安全性和 AWS 合规性进行评估。

AWS Cloud9 符合以下合规计划：

SOC

AWS 系统和组织控制 (SOC) 报告是独立的第三方检查报告，用于展示如何 AWS 实现关键合规控制和目标。

服务	SDK	SOC 1、2、3
AWS Cloud9	cloud9	✓

PCI

支付卡行业数据安全标准 (PCI DSS) 是由PCI安全标准委员会管理的专有信息安全标准，该委员会由美国运通、Discover Financial Services、JCB International、Worldw MasterCard ide和Visa Inc.创立。

服务	SDK	PCI
AWS Cloud9	cloud9	✓

FedRAMP

联邦风险与授权管理计划 (FedRAMP) 是一项美国政府层面的计划，它提供一种标准方法来对云产品和云服务进行安全性评估、授权以及持续监控。

通过 FedRAMP 评估和授权的服务将具有以下状态：

- 第三方评估组织 (3PAO) 评估：该服务目前正在接受我们的第三方评估员的评估。
- 联合授权委员会 (JAB) 审查：该服务目前正在接受 JAB 审查。

服务	SDK	FedRAMP 中等 (东部/西部)	FedRamp High () GovCloud
AWS Cloud9	cloud9	JAB 审查	不适用

DoD CC SRG

国防部 (DoD) 云计算安全需求指南 (SRG) 为云服务提供商 (CSP) 提供了一个标准化的评估和授权流程，获得国防部临时授权，让他们能够为国防部客户服务。

通过 DoD CC SRG 评估和授权的服务将具有以下状态：

- 第三方评估组织 (3PAO) 评估：该服务目前正在接受我们的第三方评估员的评估。
- 联合授权委员会 (JAB) 审查：该服务目前正在接受 JAB 审查。
- 美国国防信息系统局 (DISA) 审查：该服务目前正在接受 DISA 审查。

服务	软件开发工具包	DoD CC SRG IL2 (东部/西部)	国防部 CC SRG IL2 () GovCloud	国防部 CC SRG IL4 () GovCloud	国防部 CC SRG IL5 () GovCloud	国防部 CC SRG IL6 (秘密区域) AWS
AWS Cloud9	cloud9	JAB 审查	不适用	不适用	不适用	不适用

HIPAA BAA

1996 年的《健康保险流通与责任法案》(HIPAA) 是一项联邦法律，要求制定国家标准以保护敏感的患者健康信息，避免此类信息在未经患者同意或知情的情况下遭到披露。

AWS 使受 HIPAA 约束的受保实体及其业务伙伴能够安全地处理、存储和传输受保护的健康信息 (PHI)。此外，自 2013 年 7 月起，AWS 为此类客户提供标准化的商业伙伴附录 (BAA)

服务	SDK	HIPAA BAA
AWS Cloud9	cloud9	✓

IRAP

通过信息安全注册评估员计划 (IRAP)，澳大利亚政府客户能够验证适当的控制措施是否到位，并确定适当的责任模式，以满足澳大利亚网络安全中心 (ACSC) 编制的《澳大利亚政府信息安全手册》(ISM) 的要求。

服务	命名空间*	IRAP 受保护
AWS Cloud9	cloud9	✓

*命名空间可帮助您识别环境中的服务。AWS 例如，在创建 IAM 策略时，使用 Amazon 资源名称 (ARN) 并读取 AWS CloudTrail 日志。

C5

云计算合规性控制目录 (C5) 是由德国联邦信息安全办公室 (BSI) 在德国推出的一项由德国政府支持的认证计划，旨在帮助企业在德国政府的“云提供商安全建议”所规定的环境中使用云服务时，展示企业在运营方面防范常见网络攻击的安全措施。

服务	SDK	C5
AWS Cloud9	cloud9	✓

FINMA

FINMA 是瑞士的独立金融市场监管机构。Amazon Web Services (AWS) 已完成 FINMA ISAE 3000 类型 2 报告。

服务	SDK	FINMA
AWS Cloud9	cloud9	✓

GSMA

GSM 协会是一家代表全球移动网络运营商利益的行业组织。Amazon Web Services (AWS) 欧洲 (巴黎) 和美国东部 (俄亥俄) 区域现已由 GSM 协会 (GSMA) 依据其安全认证计划订阅管理 (SAS-SM) 进行认证，其范围涉及数据中心运营和管理 (DCOM)。这种与 GSMA 要求的一致性表明，我们持续致力于满足业界对云服务提供商的更高期望。

服务	美国东部 (俄亥俄)	欧洲 (巴黎)
AWS Cloud9	✓	✓

PiTuKri

AWS 与 PiTuKri 需求保持一致表明了我们对满足芬兰交通和通信局Traficom对云服务提供商更高的期望的持续承诺。

服务	SDK	PiTuKri
AWS Cloud9	cloud9	✓

要了解是否属于特定合规计划的范围，请参阅AWS 服务“[按合规计划划分的范围](#)”，然后选择您感兴趣的合规计划。AWS 服务 有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的“[下载报告](#)”中的“[AWS Artifact](#)”。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。AWS 提供了以下资源来帮助实现合规性：

- [安全与合规性快速入门指南](#) — 这些部署指南讨论了架构注意事项，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 [Amazon Web Services 上构建 HIPAA 安全与合规性](#) — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

Note

并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅[符合 HIPAA 要求的服务参考](#)。

- [AWS 合规资源](#) — 此工作簿和指南集合可能适用于您的行业和所在地区。
- [AWS 客户合规指南](#) — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践，AWS 服务 并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- [使用AWS Config 开发人员指南中的规则评估资源](#) — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- [AWS Security Hub](#) — 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 [Security Hub 控件参考](#)。
- [Amazon GuardDuty](#) — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务 检测您的工作负载、容器和数据面临的潜在威胁。GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。

- [AWS Audit Manager](#)— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

韧性在 AWS Cloud9

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅[AWS 全球基础设施](#)。

除 AWS 全球基础设施外，还 AWS Cloud9 支持特定功能，以支持您的数据弹性和备份需求。

- AWS Cloud9 与 AWS CodeCommit Amazon Web Services 托管的版本控制服务集成，您可以使用该服务在云中私下存储和管理资产（例如文档、源代码和二进制文件）。有关更多信息，请参阅《AWS CodeCommit 用户指南》AWS CodeCommit 中的[AWS Cloud9 与集成](#)。
- 在 AWS Cloud9 开发环境中使用 Git 版本控制系统备份远程 GitHub 仓库上的文件和数据。有关更多信息，请参阅 [使用 Git 面板进行可视化源代码控制](#)。

中的基础设施安全 AWS Cloud9

作为一项托管服务 AWS Cloud9，受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS security Pillar Well-Architected Framework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用 AWS Cloud9 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 [AWS Security Token Service](#)（AWS STS）生成临时安全凭证来对请求进行签名。

Note

默认情况下，AWS Cloud9 EC2 开发环境会自动为实例的系统包安装安全补丁。

软件更新和修补

AWS Cloud9 开发环境在云计算资源之上运行。云计算资源可以是 Amazon EC2 实例（用于 EC2 环境）或您自己的云计算资源（用于 SSH 环境）。有关这些选项的更多信息，请参阅[环境和计算资源](#)部分。

AWS Cloud9 EC2 环境在环境启动后会自动安装操作系统安全补丁和更新。AWS Cloud9 环境还包含运行和支持 IDE 功能所需的 AWS Cloud9 软件包。加载环境时会自动修补这些软件包。特定的开发工具已预先安装在 AWS Cloud9 环境中。AWS Cloud9 在 AMI 中更新这些工具，但我们不会在您的环境中自动更新它们。有关如何更新这些工具的更多信息，请参阅以下概述的部分：

- 《AWS Command Line Interface 用户指南》中的[安装或更新最新版本的 AWS CLI](#)。
- 在《AWS Serverless Application Model 开发者指南》中[@@ 管理 AWS SAM CLI 版本](#)。
- 《AWS Cloud Development Kit (AWS CDK) 开发人员指南》中的[安装 AWS CDK](#)。

无论底层的云计算资源或自动更新的频率如何，AWS Cloud9 用户或其 AWS Cloud9 管理员都有责任确保云计算资源已修补并处于最新状态。

有关客户在[责任共担模式](#)下负责内容的更多信息，请参阅[中的数据保护 AWS Cloud9](#)。

以下方面的安全最佳实践 AWS Cloud9

以下最佳实践是一般指导原则，并不代表完整安全解决方案。由于这些最佳实践可能不适合您的环境或不满足您的环境要求，因此将其视为有用的考虑因素，而不要视为惯例。

以下方面的一些安全最佳实践 AWS Cloud9

- 将您的代码安全地存储在版本控制系统中，如[AWS CodeCommit](#)。
- 对于您 AWS Cloud9 的 EC2 开发环境，请配置和使用[Amazon Elastic Block Store](#) 加密卷。
- 对于您的 EC2 环境，使用[标签](#)控制对 AWS Cloud9 资源的访问权限。
- 对于共享的 AWS Cloud9 开发环境，请遵循[最佳实践](#)。

故障排除 AWS Cloud9

使用以下信息来识别和解决问题 AWS Cloud9。

如果没有列出您的问题，或者您需要更多帮助，请参阅 [AWS Cloud9 开发论坛](#)。当您进入此论坛时，可能会要求您登录。您也可以直接[联系我们](#)。

主题

- [Installer \(安装程序 \)](#)
- [AWS Cloud9 环境](#)
- [Amazon EC2](#)
- [其他 AWS 服务](#)
- [应用程序预览](#)
- [Performance](#)
- [第三方应用程序和服务](#)

Installer (安装程序)

以下部分概述了与 AWS Cloud9 安装程序相关的故障排除问题。

AWS Cloud9 安装程序挂起或失败

问题：[下载并运行 AWS Cloud9 安装程序](#)时，会发生一个或多个错误，并且安装程序脚本不显示Done。

原因：AWS Cloud9 安装程序遇到了一个或多个无法恢复的错误，因此失败了。

解决方案：有关更多信息，请参阅[AWS Cloud9 安装程序故障排除](#)。请参阅常见问题、可能原因及提供的建议解决方案。

AWS Cloud9 显示 “Package Cloud9 IDE 1” 后，安装程序无法完成

问题：AWS Cloud9 在创建 SSH 开发环境的过程中安装在您现有的 Amazon EC2 实例或您自己的服务器上。如果 AWS Cloud9 Installer 对话框中显示以下消息，则表示安装已暂停：“Package Cloud9 IDE 1 (程序包 Cloud9 IDE 1)”。如果您选择 Cancel (取消)，则会显示以下消息：“Installation Failed (安装失败)”。当无法在客户的 SSH 主机上安装 AWS Cloud9 软件包时，就会发生此错误。

原因：SSH 主机要求您安装 Node.js。我们建议安装主机操作系统支持的最新版本的 Node.js。如果您的主机 Node.js 上有 AWS Cloud9 不支持的版本，则可能会出现安装错误。

推荐的解决方案：在 SSH 主机上安装 AWS Cloud9 支持的 Node.js 版本。

无法安装依赖项

问题：AWS Cloud9 需要互联网访问才能下载依赖项。

可能的原因：

- 如果您的 AWS Cloud9 环境使用代理访问互联网，则 AWS Cloud9 需要代理详细信息才能安装依赖关系。如果您没有向提供代理详细信息 AWS Cloud9，则会出现此错误。
- 另一个原因可能是您的环境不允许出站流量。

建议的解决方案：

- 要向提供您的代理详细信息 AWS Cloud9，请将以下代码附加到您的环境 `~/.bashrc` 文件中：

```
export http_proxy=[proxy url for http]
export https_proxy=[proxy url for https]
#Certificate Authority used by your proxy
export NODE_EXTRA_CA_CERTS=[path_to_pem_certificate]
```

例如，如果您的 HTTP 代理 URL 是 `https://172.31.26.80:3128`，而您的 HTTPS 代理 URL 是 `https://172.31.26.80:3129`，则将以下行添加到您的 `~/.bashrc` 文件中，并将 `NODE_EXTRA_CA_CERTS` 设置为 PEM 格式的证书颁发机构文件的路径。有关此变量的更多信息，请参阅 https://nodejs.org/api/cli.html#node_extra_ca_certfile。

```
export http_proxy=http://172.31.26.80:3128
export https_proxy=https://172.31.26.80:3129
export NODE_EXTRA_CA_CERTS=[path_to_pem_certificate]
```

- 如果您使用的是非入口 Amazon EC2 实例，则必须确保为 Amazon S3 配置了 Amazon VPC 端点。有关更多信息，请参阅 [为 Amazon S3 配置 VPC 端点以下载依赖项](#)。

SSH 环境错误：“需要 Python 版本 3 才能安装 pty.js”

问题：打开 AWS Cloud9 SSH 开发环境后，AWS Cloud9 IDE 中的终端会显示一条以“安装 pty.js 需要 Python 版本 3”开头的消息。

原因：要按预期工作，SSH 环境需要安装 Python 版本 3。

解决方案：在环境中安装 Python 版本 3。要检查您的版本，请在您服务器的终端上运行命令 `python --version`。要在您的服务器上安装 Python 3，请参阅以下内容之一：

- Python 示例中的 [步骤 1：安装 Python](#)。
- 在 Python 网站上[下载 Python](#)。

AWS Cloud9 环境

以下部分概述了与 AWS Cloud9 环境相关的故障排除问题。

环境创建错误：“我们无法创建 EC2 实例……”

问题：当您尝试创建 AWS Cloud9 开发环境时，会出现一条消息，上面写着“在账户验证和激活期间，我们无法在您的账户中创建 EC2 实例”。

原因：当前 AWS 正在验证并激活您的 AWS 账户。在激活完成之前（可能需要最多 24 小时），您无法创建此环境或其他环境。

解决方案：稍后重新尝试创建环境。如果您在 24 小时后仍收到此消息，请联系[支持](#)。除此之外，重要的是要知道，即使创建环境的尝试失败，AWS CloudFormation 也会在您的账户中创建相关堆栈。这些堆栈计入您账户的堆栈创建配额。为了避免耗尽堆栈创建配额，您可以安全地删除这些失败的堆栈。有关更多信息，请参阅 AWS CloudFormation 用户指南中的[在 AWS CloudFormation 控制台上删除堆栈](#)。

环境创建错误：“未授权执行 sts:AssumeRole”

问题：尝试创建新环境时，会看到以下错误：“未授权执行 sts:AssumeRole，并且环境未创建。”

可能的原因：您的 AWS 账户中不存在 AWS Cloud9 服务相关角色。

推荐的解决方案：在中创建 AWS Cloud9 AWS 账户服务相关角色。您可以使用以下命令在 AWS Command Line Interface（AWS CLI）或 AWS CloudShell 中进行这项操作。

```
aws iam create-service-linked-role --aws-service-name cloud9.amazonaws.com # For the
AWS CLI.
iam create-service-linked-role --aws-service-name cloud9.amazonaws.com      # For the
aws-shell.
```

如果您无法执行此操作，请咨询您的 AWS 账户 管理员。

运行此命令后，请尝试再次创建环境。

联合身份无法创建环境

问题：当您尝试使用 AWS 联合身份创建 AWS Cloud9 开发环境时，会显示访问错误消息，并且不会创建环境。

原因：AWS Cloud9 使用服务相关角色。在首次使用 `iam:CreateServiceLinkedRole` 调用在账户中创建环境时，会创建服务相关角色。但联合身份用户无法调用 IAM API。有关更多信息，请参阅 [AWS Security Token Service API 参考 `GetFederationToken`](#) 中的。

解决方案：要求 AWS 账户 管理员在 IAM 控制台中 AWS Cloud9 或通过使用 AWS Command Line Interface (AWS CLI) 运行以下命令来创建服务相关角色：

```
aws iam create-service-linked-role --aws-service-name cloud9.amazonaws.com
```

或者这个带有 AWS-shell 的命令：

```
iam create-service-linked-role --aws-service-name cloud9.amazonaws.com
```

有关更多信息，请参阅 IAM 用户指南 中的 [使用服务相关角色](#)。

控制台错误：“用户无权对资源执行操作”

问题：当你尝试使用 AWS Cloud9 控制台创建或管理 AWS Cloud9 开发环境时，你会看到一个错误，其中包含类似于“用户 `arn:aws:iam::123456789012:user/MyUser` 无权使用资源” `cloud9:action` 的短语 `arn:aws:cloud9:us-east-2:123456789012:environment:12a34567b8cd9012345ef67abcd890e1`，其中：

- `arn:aws:iam::123456789012:user/MyUser` 是请求用户的 Amazon Resource Name (ARN)。
- `action` 是用户请求的操作的名称。

- `arn:aws:cloud9:us-east-2:123456789012:environment:12a34567b8cd9012345ef67abcd890e1` 是用户请求用于运行操作的环境的 ARN。

原因：您登录 AWS Cloud9 控制台时使用的用户没有执行该操作的正确 AWS 访问权限。

解决方案：确保用户具有正确的 AWS 访问权限，然后再次尝试执行此操作。有关更多信息，请参阅下列内容：

- 团队设置中的 [步骤 3：向群组添加 AWS Cloud9 访问权限](#)
- 企业设置中的 [步骤 6：在企业中允许组和用户使用 AWS Cloud9](#)
- 使用共享环境中的 [关于环境成员访问角色](#)

无法连接到环境

问题：用户无法连接到环境，并且在连接阶段停滞不前。

原因：如果您更改文件的权限、从该 `~/ .ssh/authorized_keys` 文件中删除 AWS Cloud9 密钥或完全删除该文件，则可能会出现此问题。

解决方案：请勿删除此文件。如果您将其删除，则必须重新创建环境，并且可能需要将现有环境的 [EBS 卷](#) 附加到新的 EC2 环境。这是为了找回丢失的数据。如果缺少权限，请确保文件具有 Read-Write 权限。这是为了允许 SSH 进程守护程序读取它。

无法打开环境

问题：尝试打开环境时，IDE 在超过五分钟的时间内都未显示。

可能的原因：

- 登录 AWS Cloud9 控制台的 IAM 用户没有打开环境所需的 AWS 访问权限。
- 如果环境与 AWS 云计算实例（例如，Amazon EC2 实例）关联，则可能出现以下情况：
 - 与该实例关联的 VPC 的设置不正确 AWS Cloud9。
 - 该实例正在状态之间转换，或者在尝试连接实例 AWS Cloud9 时无法进行自动状态检查。
- 如果环境是 SSH 环境，则关联的云计算实例或您自己的服务器的设置不正确，无法 AWS Cloud9 允许对其进行访问。

建议的解决方案：

- 确保登录 AWS Cloud9 控制台的 IAM 用户拥有打开环境所需的 AWS 访问权限。然后尝试再次打开环境。有关更多信息，请参阅以下内容或咨询您的 AWS 账户 管理员：
 - 团队设置中的 [步骤 3：向群组添加 AWS Cloud9 访问权限](#)
 - 身份验证和访问控制中的 [AWS 的托管策略 AWS Cloud9](#)
 - 高级团队设置中的 [使用 AWS Cloud9 的团队的客户托管式策略示例](#)
 - 身份验证和访问控制中的 [客户管理型策略示例](#)
 - 《IAM 用户指南》中的 [更改 IAM 用户的权限](#)
 - IAM 用户指南中的 [IAM 策略故障排除](#)

如果已登录的 IAM 用户仍然无法打开环境，请尝试注销，然后以账户中的 AWS 账户 根用户或管理员用户身份重新登录。然后尝试再次打开环境。如果您无法以这种方式打开环境，问题最有可能与 IAM 用户的访问权限有关。

- 如果环境与 AWS 云计算实例（例如，Amazon EC2 实例）关联，请执行以下操作：
 - 确保将与该实例关联的 VPC 设置为正确的设置 AWS Cloud9，然后再次尝试打开环境。有关更多信息，请参阅 [亚马逊 VPC 对以下内容的要求 AWS Cloud9](#)。

如果将与 AWS 云计算实例关联的 VPC 设置为 AWS Cloud9 正确设置，但您仍然无法打开环境，则该实例的安全组可能会阻止访问该实例 AWS Cloud9。仅作为一种故障排除技术，请检查安全组，以确保至少允许所有 IP 地址（Anywhere 或 0.0.0.0/0）的入站 SSH 流量通过端口 22。有关说明，请参阅 Amazon EC2 用户指南中的 [描述您的安全组和更新安全组规则](#)。

要了解更多 VPC 故障排除步骤，请观看相关的 5 分钟视频 [AWS 知识中心视频：如果我无法连接到 VPC 中的实例，我可以检查什么？](#) 上 YouTube。

Warning

完成故障排除后，请确保将入站规则设置为适当的地址范围。有关更多信息，请参阅 [the section called “入站 SSH IP 地址范围”](#)。

- 重新启动实例，确保实例正在运行，并已通过所有系统检查，然后尝试再次打开环境。有关更多信息，请参阅 Amazon EC2 用户指南中的 [重启实例](#) 和 [查看状态检查](#)。
- 如果环境是 SSH 环境，请确保与其关联的云计算实例或您自己的服务器已正确设置为 AWS Cloud9 允许访问该实例。然后尝试再次打开环境。有关更多信息，请参阅 [SSH 环境主机要求](#)。

无法打开 AWS Cloud9 环境：“协作者目前无法访问此环境。Please wait until the removal of managed temporary credentials is complete, or contact the owner of this environment.”（协作者当前无法访问此环境。请等待托管式临时凭证删除完成，或联系此环境的拥有者。）

问题：如果不是环境所有者的人员将新的协作者添加到环境中，则 AWS 托管临时证书将被禁用。删除 ~/.aws/credentials 文件时将禁用凭证。~/.aws/credentials 文件被删除时，新的协作者无法访问 AWS Cloud9 环境。

原因：在删除 AWS 托管式临时凭证期间，防止环境访问是一种安全措施。这允许环境所有者确认只有受信任的协作者方可访问托管式凭证。如果满足协作者列表有效，则环境所有者可以重新启用托管式凭证，以便共享。有关更多信息，请参阅 [控制对 AWS 托管式临时凭证的访问](#)。

推荐的解决方案：等待 ~/.aws/credentials 文件完全删除，然后再尝试打开 AWS Cloud9 环境。凭证到期的最长等待时间为 15 分钟。或者，请求环境所有者重新启用或禁用托管式临时凭证。重新启用或禁用凭证后，协作者可以立即访问环境。通过将托管式凭证的状态切换为“ENABLED”（启用）或“DISABLED”（禁用），环境所有者可确保凭证不会保持在中间状态。中间统计数据可以阻止协作者访问环境。

Note

假设环境所有者和合作者属于同一 AWS 账户，则协作者可以通过在控制台上的 Your environments（您的环境）页面中查看环境卡片，来确定要联系的环境所有者。环境所有者也在 Environment details（环境详细信息）页面中列出。

环境删除错误：“One or more environments failed to delete（一个或多个环境删除失败）”

问题：当您尝试在 AWS Cloud9 控制台中删除一个或多个环境时，会显示一条消息，上面写着“一个或多个环境删除失败”，并且至少有一个环境未被删除。

可能的原因：删除一个或多个环境时 AWS CloudFormation 可能出现错误。AWS Cloud9 依赖 AWS CloudFormation 于创建和删除环境。

推荐的解决方案：AWS CloudFormation 尝试使用删除每个未删除的环境。

1. 打开 AWS CloudFormation 控制台，[网址为 https://console.aws.amazon.com/cloudformation](https://console.aws.amazon.com/cloudformation)。

2. 在 AWS 导航栏上，AWS 区域 为环境选择。
3. 在堆栈列表中，选择堆栈名称包含未删除的环境名称且状态为 DELETE_FAILED 的条目。AWS CloudFormation 例如，如果环境名称为 **my-demo-environment**，则选择以名称 aws-cloud 9-开头的堆栈。my-demo-environment (选择环境名称旁的复选框或选项，而不是环境名称本身。)
4. 选择 Actions (操作)、Delete Stack (删除堆栈)。
5. 在系统提示时，选择 Yes, Delete (是，删除)。

删除堆栈的过程可能需要几分钟时间。

如果堆栈从列表中消失，则环境现已删除。

如果在几分钟后，堆栈仍显示为 DELETE_FAILED，则环境仍未删除。您可以尝试手动删除各个失败的堆栈资源。

Note

手动删除失败堆栈的资源并不能将堆栈本身从您的资源中移除 AWS 账户。

要手动删除这些资源，请执行以下操作。在 AWS CloudFormation 控制台中，选择失败的堆栈，然后选择资源部分。访问此列表中 AWS 每个资源的控制台，然后使用该控制台删除该资源。

在 AWS Cloud9 IDE 中更改环境的超时时间

问题：用户想要更新 Amazon EC2 环境的超时时间。

原因：默认超时时间为 30 分钟。对于某些用户来说，这个时间可能太短了。

建议的解决方案：

1. 打开要配置的环境。
2. 在 AWS Cloud9 IDE 中，在菜单栏上依次选择 AWS Cloud9、首选项。
3. 在首选项窗口中，滚动至 Amazon EC2 实例部分。
4. 从可用列表中选择超时值并更新。

由于 AWS Cloud9 环境没有足够的磁盘空间，因此在 AWS Toolkit 中本地运行 SAM 应用程序时出错

问题：当您使用 AWS 工具包为 SAM 模板定义的应用程序运行 AWS SAM CLI 命令时，会出现错误。

可能的原因：使用 AWS Toolkit 在本地运行和调试无服务器应用程序时，会 AWS SAM 使用 Docker 镜像。这些映像提供一个运行时环境和构建工具，它们可以模拟您计划部署到的 Lambda 环境。

但是，如果您的环境缺乏足够的磁盘空间，则无法构建提供这些功能的 Docker 映像，并且您的本地 SAM 应用程序将无法运行。如果发生这种情况，您可能会在 Output（输出）选项卡中收到与以下内容类似的错误。

```
Error: Could not find amazon/aws-sam-cli-emulation-image-python3.7:rapid-1.18.1 image locally and failed to pull it from docker.
```

此错误与使用 Python 运行时构建的 SAM 应用程序有关。根据您为应用程序选择的运行时，您收到的消息可能略有不同。

建议的解决方案：释放环境中的磁盘空间，以构建 Docker 映像。通过在 IDE 的终端中运行以下命令来删除所有未使用的 Docker 映像。

```
docker image prune -a
```

如果由于磁盘空间限制而反复出现 SAM CLI 命令问题，请切换到使用不同[实例类型](#)的开发环境。

([回到顶部](#))

无法使用早期版本的 Microsoft Edge 浏览器加载 IDE

问题：尝试使用 Microsoft Edge Web 浏览器加载 AWS Cloud9 IDE 时返回 HTTP403: FORBIDDEN 错误。

可能的原因： AWS Cloud9 IDE 不支持某些较早版本的 Microsoft Edge。

建议的解决方案：要更新浏览器，请选择 Microsoft Edge 工具栏中的省略号 (...) 按钮。从菜单中选择 Settings（设置），然后选择 About Microsoft Edge（关于 Microsoft Edge）。如果需要更新，则会自动下载并安装。

([回到顶部](#))

无法在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment。

问题：在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment 时，会收到一条错误消息，指出无法打开此目录。

可能的原因：目前无法使用 IDE 的文件系统在同名文件夹中创建子文件夹结构 /home/ec2-user/environment。AWS Cloud9 您将无法从 AWS Cloud9 IDE 文件资源管理器访问此目录中的任何文件，但可以使用命令行访问它们。此问题仅影响文件路径 /home/ec2-user/environment/home/ec2-user/environment，/test/home/ec2-user/environment 和 /home/ec2-user/environment/test 等文件路径应该有效。这是一个已知问题，仅影响 AWS Cloud9 IDE 文件资源管理器。

建议的解决方案：使用不同的文件名和结构。

([回到顶部](#))

无法在 IDE 的文件资源管理器中为其创建子文件夹结构 /projects/ projects。 AWS Cloud9 CodeCatalyst

问题：在 AWS Cloud9 IDE 文件资源管理器中为其创建子文件夹结构 /projects/projects 时 CodeCatalyst，会收到一条错误消息，指出无法打开此目录。

可能的原因：目前无法使用 AWS Cloud9 IDE 的文件资源管理器在同名文件夹中创建子文件夹结构 /projects。CodeCatalyst 您将无法从 AWS Cloud9 IDE 文件资源管理器访问此目录中的任何文件，但可以使用命令行访问它们。此问题仅影响文件路径 /projects/projects，/test/projects 和 /projects/test 等此类文件路径应有效。这是一个已知问题，仅影响的 AWS Cloud9 IDE 文件资源管理器 CodeCatalyst。

建议的解决方案：使用不同的文件名和结构。

([回到顶部](#))

由于 tmux 会话错误，无法与 AWS Cloud9 中的终端窗口交互

问题：当您尝试在中启动新的终端窗口时 AWS Cloud9，预期的命令行界面不可用。没有命令提示符，您无法输入文本。将返回 tmux: need UTF-8 locale (LC_CTYPE) 和 invalid LC_ALL, LC_CTYPE or LANG 等错误消息。

可能的原因：终端无响应可能是由 tmux 错误引起的。AWS Cloud9 使用 [tmux 实用程序](#)。这样，即使在页面重新加载或重新连接到开发环境时，终端中显示的信息也会保持不变。

在 tmux 会话中，终端窗口中显示的内容由客户端处理。客户端与可以管理多个会话的服务器通信。服务器和客户端通过位于 tmp 文件夹中的套接字进行通信。如果您的开发环境中缺少该 tmp 文件夹或对其应用了过于严格的权限，则 tmux 会话将无法运行。如果发生这种情况，IDE 中的终端窗口将变得没有响应。

建议的解决方案：如果 tmux 错误阻止您与终端窗口进行交互，请使用另一种方法来创建具有正确权限的 tmp 文件夹。这样一来就可以运行 tmux 会话。一种解决方案是导出 .bash_profile 中或 .bashrc 文件中的 LC_CTYPE。另一个推荐的解决方案是使用 AWS Systems Manager 来设置主机管理配置。这允许通过 Amazon EC2 控制台访问相关实例。

设置主机管理

1. 首先，在 AWS Cloud9 控制台中，找到您的环境实例的名称。为此，您可以选择 Your environments (您的环境) 页面中的相关面板并选择 View details (查看详细信息)。在 Environment details (环境详细信息) 页面上，选择 Go to Instance (转到实例)。在 Amazon EC2 控制台中，确认需要访问的实例的名称。
2. 现在进入 AWS Systems Manager 控制台，在导航窗格中选择“快速设置”。
3. 在 Quick Setup (快速设置) 页面上，选择 Create (创建)。
4. 对于 Configuration types (配置类型)，请转到 Host Management (主机管理)，然后选择 Create (创建)。
5. 对于 Customize Host Management configuration options (自定义主机管理配置选项)，在 Targets (目标) 部分中，选择 Manual (手动)。
6. 选择要访问的 EC2 实例，然后选择 Create (创建)。

连接到实例并运行命令

Note

以下步骤适用于新的 EC2 控制台。

1. 在 Amazon EC2 控制台的导航窗格中，选择 Instances (实例)，然后选择要连接的实例。
2. 选择连接。

如果 Connect (连接) 未激活，您可能需要先启动实例。

3. 在 Connect to your instance (连接到您的实例) 窗格中，对于 Connection method (连接方法)，请选择 Session Manager (会话管理器)，然后选择 Connect (连接)。

4. 在出现的终端会话窗口中，输入以下命令。这些命令会创建具有正确权限的 tmp 文件夹，以便 tmux 套接字可用。

```
sudo mkdir /tmp
sudo chmod 777 /tmp
sudo rmdir /tmp/tmux-*
```

([回到顶部](#))

Amazon EC2

以下部分概述了与 Amazon EC2 相关的故障排除问题。

Amazon EC2 实例未自动更新

问题：最近的系统更新不会自动应用于连接到 AWS Cloud9 开发环境的 Amazon EC2 实例。

原因：自动应用最近的系统更新可能导致您的代码或 Amazon EC2 实例在您事先不知情或未经您批准的情况下以非预期方式运行。

建议的解决方案：

按照 Amazon EC2 用户指南中更新实例软件中的说明定期对 Amazon EC2 [实例](#)进行系统更新。

要在实例上运行命令，可以在 AWS Cloud9 IDE 中使用与实例连接的环境中的终端会话。

或者，您可以使用 SSH 远程访问实用程序（如 ssh 或 PuTTY）连接到该实例。要执行此操作，请在您的本地计算机中，使用诸如 ssh-keygen 或 PuTTYgen 之类的 SSH 密钥对创建实用程序。使用与实例连接的环境中的 AWS Cloud9 IDE 将生成的公钥存储在实例上。然后，使用 SSH 远程访问实用程序以及生成的私有密钥访问该实例。有关更多信息，请参阅您的实用程序的文档。

AWS CLI 或者 AWS-shell 错误：在 EC2 环境中“请求中包含的安全令牌无效”

问题：当您尝试使用 AWS Command Line Interface (AWS CLI) 或 AWS-shell 在 AWS Cloud9 IDE 中为 EC2 环境运行命令时，会显示一条错误：“请求中包含的安全令牌无效。”

原因：如果您启用了 AWS 托管式临时凭证并且发生了以下任一情况，可能会导致安全令牌无效：

- 您试图运行 AWS 托管临时证书不允许的命令。有关允许的命令列表，请参阅[AWS 托管临时证书支持的操作](#)。
- AWS 托管的临时证书会在 15 分钟后自动过期。
- 共享环境的 AWS 托管临时证书已停用，因为环境所有者之外的其他人添加了新成员。

建议的解决方案：

- 仅运行 AWS 托管临时证书允许的命令。如果您需要运行 AWS 托管临时证书不允许的命令，请在环境中使用一组永久凭证配置 AWS CLI 或 AWS-shell。这样可消除此限制。有关说明，请参阅[在环境中创建和存储永久访问凭证](#)。
- 对于已停用或过期的证书，请确保环境所有者打开环境，以便 AWS Cloud9 可以刷新环境中的临时证书。有关更多信息，请参阅[控制对 AWS 托管式临时凭证的访问](#)。

无法连接到 EC2 环境，因为 VPC 的 IP 地址被 Docker 使用

问题：对于 EC2 环境，如果您将 EC2 实例启动到使用 IPv4 无类别域间路由 (CIDR) 块 172.17.0.0/16 的 Amazon VPC 中，则在您尝试打开该环境时连接可能会停止。

原因：Docker 使用一种称为桥接网络的链路层设备，该设备使连接到同一桥接网络的容器能够进行通信。AWS Cloud9 创建使用默认桥进行容器通信的容器。默认网桥通常使用 172.17.0.0/16 子网进行容器联网。

如果环境实例的 VPC 子网使用的地址范围与 Docker 使用的相同，则可能会出现 IP 地址冲突。因此，当 AWS Cloud9 尝试连接到其实例时，该连接由网关路由表路由到 Docker 网桥。这样可以 AWS Cloud9 防止连接到支持开发环境的 EC2 实例。

建议的解决方案：要解决由 Amazon VPC 和 Docker 使用相同 IPv4 CIDR 地址块导致的 IP 地址冲突，请为支持您的 EC2 环境的实例配置一个新的 VPC。为这个新的 VPC，配置一个不同于 172.17.0.0/16 的 CIDR 块。（您不能更改现有 VPC 或子网的 IP 地址范围。）

有关更多配置信息，请参阅 Amazon VPC 用户指南 中的 [VPC 和子网大小调整](#)。

无法在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment。

问题：在 AWS Cloud9 IDE 文件资源管理器中创建子文件夹结构 /home/ec2-user/environment/home/ec2-user/environment 时，会收到一条错误消息，指出无法打开此目录。

可能的原因：目前无法使用 IDE 的文件系统在同名文件夹中创建子文件夹结构 `/home/ec2-user/environment`。AWS Cloud9 您将无法从 AWS Cloud9 IDE 文件资源管理器访问此目录中的任何文件，但可以使用命令行访问它们。此问题仅影响文件路径 `/home/ec2-user/environment/home/ec2-user/environment`，`/test/home/ec2-user/environment` 和 `/home/ec2-user/environment/test` 等文件路径应该有效。这是一个已知问题，仅影响 AWS Cloud9 IDE 文件资源管理器。

建议的解决方案：使用不同的文件名和结构。

当 AWS License Manager 许可证配置与 Amazon EC2 实例关联时，无法 AWS Cloud9 从控制台启动

问题：当您尝试从控制台启动 AWS Cloud9 EC2 环境时，会返回 `unable to access your environment` 一条错误消息。

可能的原因：AWS License Manager 简化了整个软件供应商许可证的管理。AWS Cloud 设置 License Manager 时，您可以创建许可证配置，它们是基于企业协议条款的许可规则集。这些许可配置可以附加到机制，例如 Amazon 系统映像 (AMI) 或 AWS CloudFormation。您可以使用其中一种机制启动 EC2 实例。

`f AWSCloud9ServiceRolePolicy` 或 `AWSServiceRoleForAWSCloud9` 服务相关角色 (SLR) 的旧版本目前不包括 `license-configuration` 资源条件。因此，AWS Cloud9 不允许启动和停止其实例。因此，AWS Cloud9 被拒绝访问其 Amazon EC2 实例，并返回错误。

推荐的解决方案：如果您无法访问现有 AWS Cloud9 环境和使用 License Manager，请将旧的 `AWSCloud9ServiceRolePolicy` 服务相关角色替换为 [在 `license-configuration` 适用于实例时明确允许 EC2 操作的 SLR 版本](#)。只需删除旧角色即可替换旧角色。之后将自动创建更新后的角色。

无法在 EC2 环境中运行某些命令或脚本

问题：打开 AWS Cloud9 EC2 开发环境后，您将无法安装某些类型的软件包、无法运行 `yum` 或之类的命令 `apt`，也无法运行包含通常适用于其他 Linux 操作系统的命令的脚本。

原因：AWS Cloud9 用于 EC2 环境的亚马逊 EC2 实例依赖于亚马逊 Linux (基于红帽企业 Linux (RHEL)) 或 Ubuntu 服务器。

解决方案：如果您在用于 EC2 环境的 IDE 中安装或管理软件包或运行命令或脚本，根据该环境的实例，请确保它们与 RHEL (对于 Amazon Linux) 或 Ubuntu Server 兼容。

使用创建 EC2 环境时报告错误消息“账户中 AWSCloud9SSMInstanceProfile 不存在实例配置文件” AWS CloudFormation

问题：使用 [AWS::Cloud9::EnvironmentEC2](#) AWS CloudFormation 资源创建 EC2 环境时，用户会收到一条错误消息，提示账户中 AWSCloud9SSMInstanceProfile 不存在实例配置文件。

原因：在创建非入口 EC2 环境时，您必须创建服务角色 AWSCloud9SSMAccessRole 和实例配置文件 AWSCloud9SSMInstanceProfile。这些 IAM 资源使得 Systems Manager 能够对支持您的开发环境的 EC2 实例进行管理。

如果您使用控制台创建非入口环境，AWSCloud9SSMAccessRole 和 AWSCloud9SSMInstanceProfile 会自动创建。但是，在使用 AWS CloudFormation 或 AWS CLI 创建您的第一个无入口环境时，必须手动创建这些 IAM 资源。

推荐的解决方案：有关编辑 AWS CloudFormation 模板和更新 IAM 权限的信息，请参阅 [使用 AWS CloudFormation 创建非入口 EC2 环境](#)

使用 AWS CloudFormation 创建 EC2 环境时，报告错误消息“未授权对资源执行 **perform: ssm:StartSession**”

问题：使用 [AWS::Cloud9::EnvironmentEC2](#) AWS CloudFormation 资源创建 EC2 环境时，用户会收到 AccessDeniedException 并被告知他们“无权 ssm:StartSession 在资源上执行：”。

原因：用户缺乏调用 StartSession API 的权限，该 API 是将 Systems Manager 用于非入口实例的 EC2 环境配置的一部分。

推荐的解决方案：有关编辑 AWS CloudFormation 模板和更新 IAM 权限的信息，请参阅 [使用 AWS CloudFormation 创建非入口 EC2 环境](#)。

使用 AWS CLI 创建 EC2 环境时，报告错误消息“no authorization to perform: on resource: instance profile” (未授权对资源执行 **iam:GetInstanceProfile** : 实例配置文件 AWSCloud9SSMInstanceProfile)

问题：使用创建 EC2 环境时，用户会收到 AccessDeniedException 并被告知其 AWS Cloud9 环境无权“GetInstanceProfile 对资源：实例配置文件执行 iam:” AWSCloud9SSMInstanceProfile。 [AWS CLI](#)

原因：对于使用 Systems Manager 进行无入口实例的 EC2 环境，AWS Cloud9 缺少调用配置所需的 StartSession API 的权限。

推荐的解决方案：有关AWSCloud9SSMInstanceProfile向您的 AWS Cloud9 环境添加所需AWSCloud9SSMAccessRole服务角色的信息，请参阅[使用 AWS CLI 管理 Systems Manager 的实例配置文件](#)。

将默认加密应用于 Amazon EBS 卷时，无法创建环境

问题：尝试创建 Amazon EC2 环境时会返回 Failed to create environments. The development environment '[environment-ID]' failed to create 错误。

可能的原因：如果您的 AWS Cloud9 IDE 使用默认加密的 Amazon EBS 卷，则 AWS Identity and Access Management 服务相关角色 AWS Cloud9 需要访问这些 EBS 卷 AWS KMS keys 的。如果未提供访问权限，AWS Cloud9 IDE 可能无法启动，并且可能很难调试问题。

推荐的解决方案：要提供访问权限，请将、的服务相关角色添加到您的 Amazon EBS 卷使用的客户托管密钥中。AWS Cloud9AWSServiceRoleForAWSCloud9

有关此任务的更多信息，请参阅AWS 规范指导模式中的[创建使用 Amazon EBS 卷和默认加密的 Amazon EBS 卷](#)。AWS Cloud9

EC2-Classic 账户的 VPC 错误“Unable to access your environment (无法访问您的环境)”

问题：EC2-Classic 是在 Amazon EC2 的初始版本中引入的。如果您使用在 2013 年 12 月 4 日之前设置的，那么如果您在创建 AWS Cloud9 EC2 开发环境时未配置 Amazon VPC 和子网，则可能会发生此错误。AWS 账户

如果您接受原定设置 VPC 设置，则 Amazon EC2 实例将启动到 EC2-Classic 网络中。该实例不会启动到原定设置 VPC 的子网中。当创建环境失败时，将显示以下消息：

Environment Error (环境错误)

Unable to access your environment (无法访问您的环境)

The environment creation failed with the error: The following resource(s) failed to create: [Instance]. . Rollback requested by user.. (环境创建失败，错误为: 无法创建以下资源: [Instance]. . 用户已请求回滚..)

您可以确认错误是因 EC2 实例未位于默认 VPC 中导致的。AWS CloudFormation 用于查看开发环境的堆栈事件历史记录。

1. 打开控制 AWS CloudFormation 台。有关更多信息，请参阅[登录到 AWS CloudFormation 本地控制台](#)。
2. 在 AWS CloudFormation 控制台中，选择 Stacks。
3. 在 Stacks (堆栈) 页面上，选择创建失败的开发环境的名称。
4. 在 Stack details (堆栈详细信息) 页面上，选择 Events (事件) 选项卡并检查以下条目：

Status: CREATE_FAILED (状态: CREATE_FAILED)

状态原因：只有启动 VPC 时才支持该 AssociatePublicIpAddress 参数。 [...]

原因：AWS Cloud9 开发环境必须与符合特定 VPC 要求的 Amazon VPC 相关联。对于启用了 EC2-Classic 的账户，在[创建 EC2 环境](#)时接受默认网络设置意味着所需的 EC2 实例未在 VPC 中启动。相反，此实例将在 EC2-Classic 网络中启动。

建议的解决方案：对于 EC2-Classic 账户，您必须在[创建 EC2 环境](#)时选择 VPC 和子网。在 Configure settings (配置设置) 页面上的 Network settings (advanced) (网络设置(高级)) 部分中，选择可以在其中启动 EC2 实例的 VPC 和子网。

其他 AWS 服务

以下部分概述了与其他 AWS 服务相关的故障排除问题。

无法在 IDE 的文件资源管理器中为其创建子文件夹结构 /projects/ projects。 AWS Cloud9 CodeCatalyst

问题：在 AWS Cloud9 IDE 文件资源管理器中为其创建子文件夹结构 /projects/projects 时 CodeCatalyst，会收到一条错误消息，指出无法打开此目录。

可能的原因：目前无法使用 AWS Cloud9 IDE 的文件资源管理器在同名文件夹中创建子文件夹结构 /projects。CodeCatalyst 您将无法从 AWS Cloud9 IDE 文件资源管理器访问此目录中的任何文件，但可以使用命令行访问它们。此问题仅影响文件路径 /projects/projects，/test/projects 和 /projects/test 等此类文件路径应有效。这是一个已知问题，仅影响的 AWS Cloud9 IDE 文件资源管理器 CodeCatalyst。

建议的解决方案：使用不同的文件名和结构。

无法显示在 IDE 以外运行的应用程序

问题：当您或其他用户尝试在 IDE 以外的 Web 浏览器选项卡中显示正在运行的应用程序，该 Web 浏览器选项卡会显示错误或空白。

可能的原因：

- 应用程序未在 IDE 中运行。
- 应用程序正使用 IP 127.0.0.1 或 localhost 运行。
- 该应用程序在 AWS Cloud9 EC2 开发环境中运行。此外，一个或多个与相应 Amazon EC2 实例关联的安全组不允许通过应用程序需要使用的协议、端口或 IP 地址传输输入站流量。
- 该应用程序在 AWS 云计算实例（例如，Amazon EC2 实例）的 AWS Cloud9 SSH 开发环境中运行。此外，与相应实例关联的虚拟私有云 (VPC) 中子网的网络 ACL 不允许通过应用程序需要使用的协议、端口或 IP 地址传输输入站流量。
- URL 不正确。
- 正在请求应用程序预览标签页中的 URL，而不是实例的公有 IP 地址。
- 您正在尝试转到包含 IP 127.0.0.1 或 localhost 的地址。这些 IP 将尝试访问您的本地计算机上的资源，而非环境中的资源。
- 该实例的公有 IP 地址已更改。
- Web 请求源自 virtual private network (VPN)，该网络阻止通过应用程序需要使用的协议、端口或 IP 地址传输流量。
- 该应用程序正在 SSH 环境中运行。但是，您的服务器或关联的网络不允许通过应用程序需要使用的协议、端口或 IP 地址传输流量。

建议的解决方案：

- 确保此应用程序正在 IDE 中运行。
- 确保应用程序未使用 IP 127.0.0.1 或 localhost 运行。有关 Node.js 和 Python 中的示例，请参阅[运行应用程序](#)。
- 假设应用程序在 AWS 云计算实例（例如，Amazon EC2 实例）上运行。请确保与相应实例关联的所有安全组都允许通过应用程序需要使用的协议、端口和 IP 地址传输输入站流量。有关说明，请参阅[通过互联网共享正在运行的应用程序中的步骤 2：为实例设置安全组](#)。要了解更多信息，请参阅 Amazon VPC 用户指南中的[您的 VPC 的安全组](#)。
- 假设应用程序在 AWS 云计算实例上运行。此外，与相应实例关联的 VPC 中的子网存在网络 ACL。请确保网络 ACL 允许通过应用程序需要使用的协议、端口和 IP 地址传输输入站流量。有关说明，请参阅

阅通过互联网共享正在运行的应用程序中的 [步骤 3：为实例设置子网](#)。有关信息，请参阅 Amazon VPC 用户指南 中的 [网络 ACL](#)。

- 确保请求 URL 是正确的，包括协议和端口 (如果必须指定)。有关更多信息，请参阅通过互联网共享运行的应用程序中的 [步骤 4：共享运行的应用程序 URL](#)。
- 我们不建议请求格式为 `https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/` (其中 `12a34567b8cd9012345ef67abcd890e1` 是 AWS Cloud9 分配给环境的 ID，`us-east-2` 是环境的 AWS 区域 ID) 的 URL。仅当环境的 IDE 处于打开状态并且应用程序正在同一个 Web 浏览器中运行时，此 URL 才有用。
- 假设您正在尝试转到包含 IP `127.0.0.1` 或 `localhost` 的地址。尝试转到正在运行的应用程序的正确非本地地址。有关更多信息，请参阅 [通过互联网共享运行的应用程序](#)。
- 假设应用程序在 AWS 云计算实例上运行。确定该实例的公有 IP 地址是否已更改。该实例的公有 IP 地址可能在实例重新启动时随时更改。要防止此 IP 地址发生更改，您可以分配弹性 IP 地址并将其分配给正在运行的实例。有关更多信息，请参阅通过互联网共享运行的应用程序中的 [步骤 4：共享运行的应用程序 URL](#)。
- 如果 Web 请求源自 VPN，请确保 VPN 允许通过应用程序需要使用的协议、端口和 IP 地址传输流量。如果您无法更改您的 VPN，请联系您的网络管理员。或者，从其他网络发出 Web 请求 (如果可能)。
- 假设应用程序在您自己的服务器的 SSH 环境中运行。确保您的服务器和关联的网络允许通过应用程序需要使用的协议、端口和 IP 地址传输流量。如果您无法更改您的服务器或关联的网络，请联系您的服务器或网络管理员。
- 尝试通过运行 `curl` 命令并在后面跟随 URL，从环境中的终端运行应用程序。如果此命令显示错误消息，则可能存在与之无关的其他问题 AWS Cloud9。

运行 AWS Toolkit 时出错：“您的环境已用完索引节点，请增加'fs.inotify.max_user_watches'限制。”

问题：AWS Toolkit 使用的文件观察器实用程序已接近其当前可监视的文件限制或限额。

原因：AWS Toolkit 使用文件观察器实用程序来监视文件和目录的更改。当该实用程序接近其可监视文件的当前配额时，会出现一条警告消息。

建议的解决方案：要增加文件监视器可处理的最大文件数，请执行以下操作：

1. 要开始一个终端会话，请在菜单栏上依次选择 Window (窗口)、New Terminal (新建终端)。
2. 输入以下命令。

```
sudo bash -c 'echo "fs.inotify.max_user_watches=524288" >> /etc/sysctl.conf' &&  
sudo sysctl -p
```

Lambda 本地函数运行错误：无法安装 SAM Local

问题：尝试在 AWS Cloud9 IDE 中运行 AWS Lambda 函数的本地版本后，会显示一个对话框。对话框显示在安装 SAM Local 时遇到问题。AWS Cloud9 需要 SAM Local 才能在 IDE 中运行本地版本的 AWS Lambda 函数。除非安装 SAM Local，否则无法在 IDE 中运行 Lambda 函数的本地版本。

原因：在环境中的预期路径上 AWS Cloud9 找不到 SAM Local，即 ~/.c9/bin/sam。这是因为尚未安装 SAM Local；或者虽然已安装了该程序，但 AWS Cloud9 无法在该位置找到程序。

推荐的解决方案：您可以等待 AWS Cloud9 尝试完成 SAM Local 的安装，也可以自己安装。

要查看尝试安装 SAM Local 的效果如何 AWS Cloud9，请在菜单栏上选择“窗口”、“安装程序”。

要自己安装 SAM Local，请按照《AWS Serverless Application Model 开发人员指南》中在 [Linux 上安装 AWS SAM CLI](#) 中的说明进行操作。

AWS Control Tower 尝试使用以下方法创建 Amazon EC2 环境时出错 AWS Cloud9：“创建环境失败，错误为：以下挂钩失败：[:: GuardControlTower:: Hook]。”

问题：AWS Control Tower 主动控制 CT.EC2.P AWS Cloud9 R. 8 存在兼容性问题。如果启用此控件，则无法在 AWS Cloud9 中创建 EC2 环境。

原因：期 AWS Control Tower 望 AssociatePublicIpAddress 参数出现在 AWS CloudFormation 模板中。此时无法添加此参数。

推荐的解决方案：从控制 AWS Control Tower 台禁用控制 CT.EC2.PR.8，然后在 AWS Cloud9 中重新创建环境。

将默认加密应用于 Amazon EBS 卷时，无法创建环境

问题：尝试创建 Amazon EC2 环境时会返回 Failed to create environments. The development environment '[environment-ID]' failed to create 错误。

可能的原因：如果您的 AWS Cloud9 IDE 使用默认加密的 Amazon EBS 卷，则 AWS Identity and Access Management 服务相关角色 AWS Cloud9 需要访问这些 EBS 卷 AWS KMS keys 的。如果未提供访问权限，AWS Cloud9 IDE 可能无法启动，并且可能很难调试问题。

推荐的解决方案：要提供访问权限，请将、的服务相关角色添加到您的 Amazon EBS 卷使用的客户托管密钥中。AWS Cloud9AWSServiceRoleForAWSCloud9

有关此任务的更多信息，请参阅AWS 规范指导模式中的[创建使用 Amazon EBS 卷和默认加密的 Amazon EBS 卷](#)。AWS Cloud9

([回到顶部](#))

当 AWS License Manager 许可证配置与 Amazon EC2 实例关联时，无法 AWS Cloud9 从控制台启动

问题：当您尝试从控制台启动 AWS Cloud9 EC2 环境时，会返回unable to access your environment一条错误消息。

可能的原因：AWS License Manager 简化了整个软件供应商许可证的管理。AWS Cloud设置 License Manager 时，您可以创建许可证配置，它们是基于企业协议条款的许可规则集。这些许可配置可以附加到机制，例如 Amazon 系统映像 (AMI) 或 AWS CloudFormation。您可以使用其中一种机制启动 EC2 实例。

f AWSCloud9ServiceRolePolicyor AWSServiceRoleForAWSCloud 9 服务相关角色 (SLR) 的旧版本目前不包括license-configuration资源条件。因此，AWS Cloud9 不允许启动和停止其实例。因此，AWS Cloud9 被拒绝访问其 Amazon EC2 实例，并返回错误。

推荐的解决方案：如果您无法访问现有 AWS Cloud9 环境和使用 License Manager，请将旧的AWSCloud9ServiceRolePolicy服务相关角色替换为[在license-configuration适用于实例时明确允许 EC2 操作的 SLR 版本](#)。只需删除旧角色即可替换旧角色。之后将自动创建更新后的角色。

([回到顶部](#))

应用程序预览

以下部分概述了与应用程序预览相关的故障排除问题。

重新加载环境之后，必须刷新应用程序预览

问题：重新加载显示应用程序预览选项卡的环境之后，该选项卡未显示应用程序预览。

原因：有时，用户编写的代码可能会运行无限循环。或者他们的代码会占用太多的内存，以至于应用程序预览运行时 AWS Cloud9 IDE 可能会暂停或停止。为了防止这种情况 AWS Cloud9 发生，在重新加载环境时不要重新加载应用程序预览选项卡。

解决方案：在重新加载显示应用程序预览选项卡的环境之后，要显示应用程序预览，请选择该选项卡上的 Click to load the page (单击以加载页面) 按钮。

应用程序预览或文件预览通知：“Third-party cookies disabled (第三方 cookie 已禁用)”

问题：当您尝试预览[应用程序](#)或[文件](#)时，会显示一条通知，并显示以下消息：“预览功能已禁用，因为您的浏览器禁用了第三方 cookie。”

原因：打开 AWS Cloud9 IDE 不需要第三方 Cookie。但是，您必须启用第三方 Cookie 才能使用应用程序预览或文件预览功能。

解决方案：在 Web 浏览器中启用第三方 cookie，重新加载 IDE，然后再次尝试打开预览。

- Apple Safari：Apple 支持网站上的[在 Safari 中管理 Cookie 和网站数据](#)。
- Google Chrome：在 Google Chrome 帮助网站上的[在 Chrome 中清除、启用和管理 Cookie](#) 中的更改您的 Cookie 设置。
- Internet Explorer：Microsoft 支持网站上的[删除和管理 Cookie](#) 中的阻止或允许 Cookie。
- Microsoft Edge：Microsoft 支持网站上的[阻止第三方 Cookie](#)。
- Mozilla Firefox：Mozilla 支持网站上的[启用和禁用网站用于跟踪您的偏好的 Cookie](#) 中的接受第三方 Cookie 设置。
- 任何其他 Web 浏览器：请参阅该 Web 浏览器的文档。

如果您的 Web 浏览器允许这种精细度，则只能为 AWS Cloud9 启用第三方 Cookie。为此，请指定以下域，具体取决于您要在其中使用 AWS Cloud9 的受支持的 AWS 区域。

AWS 区域	域
美国东部 (弗吉尼亚州北部)	*.vfs.cloud9.us-east-1.amazonaws.com vfs.cloud9.us-east-1.amazonaws.com

AWS 区域	域
美国东部 (俄亥俄州)	*.vfs.cloud9.us-east-2.amazonaws.com vfs.cloud9.us-east-2.amazonaws.com
美国西部 (加利福尼亚北部)	*.vfs.cloud9.us-west-1.amazonaws.com vfs.cloud9.us-west-1.amazonaws.com
美国西部 (俄勒冈州)	*.vfs.cloud9.us-west-2.amazonaws.com vfs.cloud9.us-west-2.amazonaws.com
非洲 (开普敦)	*.vfs.cloud9.af-south-1.amazonaws.com vfs.cloud9.af-south-1.amazonaws.com
亚太地区 (香港)	*.vfs.cloud9.ap-east-1.amazonaws.com vfs.cloud9.ap-east-1.amazonaws.com
Asia Pacific (Mumbai)	*.vfs.cloud9.ap-south-1.amazonaws.com vfs.cloud9.ap-south-1.amazonaws.com

AWS 区域	域
亚太地区 (大阪)	*.vfs.cloud9.ap-northeast-3 .amazonaws.com vfs.cloud9.ap-northeast-3.a mazonaws.com
亚太地区 (首尔)	*.vfs.cloud9.ap-northeast-2 .amazonaws.com vfs.cloud9.ap-northeast-2.a mazonaws.com
亚太地区 (新加坡)	*.vfs.cloud9.ap-southeast-1 .amazonaws.com vfs.cloud9.ap-southeast-1.a mazonaws.com
亚太地区 (悉尼)	*.vfs.cloud9.ap-southeast-2 .amazonaws.com vfs.cloud9.ap-southeast-2.a mazonaws.com
亚太地区 (东京)	*.vfs.cloud9.ap-northeast-1 .amazonaws.com vfs.cloud9.ap-northeast-1.a mazonaws.com
加拿大 (中部)	*.vfs.cloud9.ca-central-1.a mazonaws.com vfs.cloud9.ca-central-1.ama zonaws.com

AWS 区域	域
欧洲地区 (法兰克福)	*.vfs.cloud9.eu-central-1.amazonaws.com vfs.cloud9.eu-central-1.amazonaws.com
欧洲地区 (爱尔兰)	*.vfs.cloud9.eu-west-1.amazonaws.com vfs.cloud9.eu-west-1.amazonaws.com
欧洲地区 (伦敦)	*.vfs.cloud9.eu-west-2.amazonaws.com vfs.cloud9.eu-west-2.amazonaws.com
欧洲地区 (米兰)	*.vfs.cloud9.eu-south-1.amazonaws.com vfs.cloud9.eu-south-1.amazonaws.com
欧洲地区 (巴黎)	*.vfs.cloud9.eu-west-3.amazonaws.com vfs.cloud9.eu-west-3.amazonaws.com
欧洲地区 (斯德哥尔摩)	*.vfs.cloud9.eu-north-1.amazonaws.com vfs.cloud9.eu-north-1.amazonaws.com

AWS 区域	域
中东 (巴林)	*.vfs.cloud9.me-south-1.amazonaws.com vfs.cloud9.me-south-1.amazonaws.com
南美洲 (圣保罗)	*.vfs.cloud9.sa-east-1.amazonaws.com vfs.cloud9.sa-east-1.amazonaws.com

应用程序预览选项卡显示错误或空白

问题：在 IDE 菜单栏中，当您选择 Preview, Preview Running Application (预览、预览正在运行的应用程序) 或 Tools, Preview, Preview Running Application (工具、预览、预览正在运行的应用程序)，以便尝试在 IDE 中的预览选项卡上显示您的应用程序时，该选项卡将显示错误或空白。

可能的原因：

- 您的应用程序未在 IDE 中运行。
- 您的应用程序未使用 HTTP 运行。
- 您的应用程序正通过多个端口运行。
- 您的应用程序正通过 8080、8081 或 8082 之外的端口运行。
- 您的应用程序正使用 127.0.0.1、localhost 或 0.0.0.0 之外的 IP 运行。
- 在预览标签页的 URL 中未指定端口 (8080、8081 或 8082)。
- 您的网络阻止流向端口 8080、8081 或 8082 的入站流量。
- 您正在尝试转到包含 IP 127.0.0.1、localhost 或 0.0.0.0 的地址。默认情况下，AWS Cloud9 IDE 会尝试访问您的本地计算机。它不会尝试访问连接到环境的实例或您自己的服务器。

建议的解决方案：

- 确保此应用程序正在 IDE 中运行。
- 确保此应用程序正在使用 HTTP 运行。有关 Node.js 和 Python 中的示例，请参阅[运行应用程序](#)。

- 确保应用程序只通过一个端口运行。有关 Node.js 和 Python 中的示例，请参阅[运行应用程序](#)。
- 确保应用程序正通过端口 8080、8081 或 8082 运行。有关 Node.js 和 Python 中的示例，请参阅[运行应用程序](#)。
- 确保应用程序正使用 IP 127.0.0.1、localhost 或 0.0.0.0 运行。有关 Node.js 和 Python 中的示例，请参阅[运行应用程序](#)。
- 将 :8080、:8081 或 :8082 添加到预览标签页上的 URL。
- 确保您的网络允许入站流量通过端口 8080、8081 或 8082。如果您无法更改您的网络，请联系您的网络管理员。
- 如果您正在尝试访问一个包含 IP 127.0.0.1、localhost 或 0.0.0.0 的地址，请尝试转到以下地址：<https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/>。在该地址中，12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 分配给环境的 ID。us-east-2 是环境所在 AWS 区域的 ID。您也可以尝试在 IDE 之外转到这个地址。但是，仅当环境的 IDE 处于打开状态并且应用程序正在同一个 Web 浏览器中运行时，这才会起作用。
- 当您确定满足所有之前的条件后，请尝试停止应用程序，然后重新启动它。
- 如果您已停止应用程序并重新启动，请尝试在菜单栏上重新选择 Preview (预览)、Preview Running Application(预览正在运行的应用程序) 或 Tools (工具)、Preview (预览)、Preview Running Application (预览正在运行的应用程序)。或者，如果选项卡已显示，则尝试在相应的应用程序预览标签页上选择 Refresh (刷新) 按钮 (圆形箭头)。

无法在 IDE 中预览 Web 内容，因为与站点的连接不安全

问题：当您尝试访问诸如 AWS Cloud9 EC2 环境中托管的 WordPress 网站之类的 Web 内容时，IDE 预览窗口无法显示该内容。

可能的原因：默认情况下，您在 AWS Cloud9 IDE 的“应用程序预览”选项卡中访问的所有网页都会自动使用 HTTPS 协议。如果页面的 URI 具有不安全的 http 协议，则会自动替换为 https。并且您无法通过将 https 手动更改回 http 的方式来访问不安全的内容。

建议的解决方案：从尝试在 IDE 中预览的网站中删除不安全的 HTTP 脚本或内容。按照 Web 服务器或内容管理系统的说明，获取有关实施 HTTPS 的指导。

预览文件时返回 499 错误

问题：当您尝试使用 AWS Cloud9 IDE 预览包含包含该属性的 <script> 元素且该 srctype 属性设置为 module 的文件时，会出现 499 错误，脚本无法按预期运行。

原因：AWS Cloud9 IDE 中的文件预览提取请求要求网络浏览器发送 Cookie 才能进行身份验证。默认情况下，Web 浏览器会为常规脚本请求发送 Cookie。他们不会为模块脚本请求发送 Cookie，除非您添加 `crossorigin` 属性。

解决方案：将 `crossorigin` 属性添加至 `<script>` 元素。例如，`<script type="module" src="index.js" crossorigin></script>`。然后，保存更改的文件，并尝试再次预览该文件。

Performance

以下部分概述了与性能相关的故障排除问题。

AWS Cloud9 IDE 冻结了很长时间

问题：在启动期间和执行刷新时，AWS Cloud9 IDE 终端会冻结很长时间并且无法使用。

原因：您的环境中可能含有大量文件正在被 AWS Cloud9 的文件监视模块递归监视。

推荐的解决方案：您可以降低文件监视深度（最小值为 1），并考虑将大型文件夹或与源代码无关的文件夹（构建输出/构件、第三方软件包）添加到忽略的模式中。要完成此操作，请导航至首选项 > 用户设置 > 文件监视。请注意，这将导致 AWS Toolkit CodeLenses 中无法正常运行。

另一种可能的解决方案是考虑通过减少要搜索的最大文件数来忽略与源代码无关的大型文件和文件夹。要完成此操作，请导航至首选项 > 项目设置 > 在文件中查找。请注意，这将导致被忽略的文件夹不会显示在文件搜索中。

控制台警告：“Switching to the minimal code completion engine... (切换到最小代码完成引擎.....)”

问题：在 AWS Cloud9 控制台中工作时（例如，打开 IDE 或刷新 IDE 的网页时），您会看到以下消息：“在此环境中有一个或多个会话或协作者处于活动状态。切换到最小代码完成引擎来节省内存。”结合此消息来看，代码完成行为可能很慢或间歇。

原因：运行代码完成引擎占用环境中的内存和 CPU 周期。此外，每个协作者和每个额外会话都需要单独的代码完成引擎。为避免使用太多资源，尤其是在 t2.nano 和等小实例大小上 t2.micro，请 AWS Cloud9 切换到最低限度的代码完成引擎。

建议的解决方案：如果您计划经常协作或长时间协作，请在创建 EC2 环境时选择更大的 Amazon EC2 实例。或者，也可以将 SSH 环境连接到容量更大的实例。

Note

选择更大的 Amazon EC2 实例可能会 AWS 账户 导致您产生额外费用。有关更多信息，请参阅 [Amazon EC2 定价](#)。

IDE 警告：“This environment is running low on memory (此环境内存不足)”或“this environment has high CPU load (此环境具有较高 CPU 负载)”

问题：在 IDE 运行时，出现一条类似以下短句的消息，“this environment is running low on memory (此环境内存不足)”或“this environment has high CPU load (此环境具有较高 CPU 负载)”

原因：IDE 可能没有足够的计算资源用来继续运行而不发生延迟或挂起。

建议的解决方案：

- 停止一个或多个正在运行的进程以释放可用内存。为此，请在环境的 IDE 中的菜单栏上，选择 Tools, Process List (工具、进程列表)。对于要停止的每个进程，请选择进程，然后选择 Force Kill (强制终止)。
- 在环境中创建交换文件。交换文件是操作系统可以用作虚拟内存的环境中的文件。

要确认环境当前是否正在使用交换内存，请在环境中的终端会话中运行 **top** 命令。如果正在使用交换内存，则输出将显示非零 Swap 内存统计数据 (例如 Swap: 499996k total, 1280k used, 498716 free, 110672k cached)。要停止显示实时内存信息，请按 Ctrl + C。

要创建交换文件，请在环境中运行类似如下的命令。

```
sudo fallocate --length 512MB /var/swapfile && sudo chmod 600 /var/swapfile && sudo  
mkswap /var/swapfile && echo '/var/swapfile swap swap defaults 0 0' | sudo tee -a /  
etc/fstab > /dev/null
```

上述命令执行以下操作：

1. 在 /var 目录中创建一个名为 swapfile 的 512 MB 文件。
2. 将对 swapfile 文件的访问权限更改为读写 (仅针对所有者)。
3. 将 swapfile 文件设置为交换文件。
4. 将信息写入 /etc/fstab file。这使得该交换文件在系统重新启动时可用。

运行上述命令后，要使此交换文件立即可用，请运行以下命令。

```
sudo swapon /var/swapfile
```

- 调整大小或将环境移动到具有更多计算资源的实例或服务器。要移动 Amazon EC2 实例或调整其大小，请参阅 [移动环境并对 Amazon EBS 卷调整大小或加密](#)。对于其他实例或服务器类型，请参阅您的实例或服务器的文档。

无法在 AWS Cloud9 IDE 中上传文件

问题：用户无法在 AWS Cloud9 IDE 中上传大文件。此类上传失败。

原因：AWS Cloud9 限制上传到 AWS Cloud9 IDE 的速度，结果文件上传请求超时。

推荐的解决方案：我们建议将文件上传到 Amazon S3，然后使用 Amazon S3 通过 AWS Cloud9 IDE 中的 CLI 将文件下载到环境中。有关将对象上传至 Amazon S3 的更多信息，请参阅《Amazon S3 用户指南》中的 [上传对象](#)。

AWS Cloud9 IDE 中的下载速度很慢

问题：用户在尝试从 AWS Cloud9 IDE 下载文件时遇到下载速度慢的问题。

原因：将文件从 IDE 下载到本地文件系统时，传输速度将限制为 0.1 兆字节/秒。

推荐的解决方案：要加快文件传输速度，请使用 AWS Cloud9 IDE 中的 CLI 将文件上传到 Amazon S3，然后使用 Amazon S3 从那里下载文件。

无法在 IDE 中预览 Web 内容，因为与站点的连接不安全

问题：当您尝试访问诸如 AWS Cloud9 EC2 环境中托管的 WordPress 网站之类的 Web 内容时，IDE 预览窗口无法显示该内容。

可能的原因：默认情况下，您在 AWS Cloud9 IDE 的“应用程序预览”选项卡中访问的所有网页都会自动使用 HTTPS 协议。如果页面的 URI 具有不安全的 http 协议，则会自动替换为 https。并且您无法通过将 https 手动更改回 http 的方式来访问不安全的内容。

建议的解决方案：从尝试在 IDE 中预览的网站中删除不安全的 HTTP 脚本或内容。按照 Web 服务器或内容管理系统的说明，获取有关实施 HTTPS 的指导。

([回到顶部](#))

第三方应用程序和服务

以下部分概述了与第三方应用程序和服务相关的故障排除问题。

由于 **tmux** 会话错误，无法与 AWS Cloud9 中的终端窗口交互

问题：当您尝试在中启动新的终端窗口时 AWS Cloud9，预期的命令行界面不可用。没有命令提示符，您无法输入文本。将返回 `tmux: need UTF-8 locale (LC_CTYPE) and invalid LC_ALL, LC_CTYPE or LANG` 等错误消息。

可能的原因：终端无响应可能是由 `tmux` 错误引起的。AWS Cloud9 使用 [tmux 实用程序](#)。这样，即使在页面重新加载或重新连接到开发环境时，终端中显示的信息也会保持不变。

在 `tmux` 会话中，终端窗口中显示的内容由客户端处理。客户端与可以管理多个会话的服务器通信。服务器和客户端通过位于 `tmp` 文件夹中的套接字进行通信。如果您的开发环境中缺少该 `tmp` 文件夹或对其应用了过于严格的权限，则 `tmux` 会话将无法运行。如果发生这种情况，IDE 中的终端窗口将变得没有响应。

建议的解决方案：如果 `tmux` 错误阻止您与终端窗口进行交互，请使用另一种方法来创建具有正确权限的 `tmp` 文件夹。这样一来就可以运行 `tmux` 会话。一种解决方案是导出 `.bash_profile` 中或 `.bashrc` 文件中的 `LC_CTYPE`。另一个推荐的解决方案是使用 AWS Systems Manager 来设置主机管理配置。这允许通过 Amazon EC2 控制台访问相关实例。

设置主机管理

1. 首先，在 AWS Cloud9 控制台中，找到您的环境实例的名称。为此，您可以选择 `Your environments` (您的环境) 页面中的相关面板并选择 `View details` (查看详细信息)。在 `Environment details` (环境详细信息) 页面上，选择 `Go to Instance` (转到实例)。在 Amazon EC2 控制台中，确认需要访问的实例的名称。
2. 现在进入 AWS Systems Manager 控制台，在导航窗格中选择“快速设置”。
3. 在 `Quick Setup` (快速设置) 页面上，选择 `Create` (创建)。
4. 对于 `Configuration types` (配置类型)，请转到 `Host Management` (主机管理)，然后选择 `Create` (创建)。
5. 对于 `Customize Host Management configuration options` (自定义主机管理配置选项)，在 `Targets` (目标) 部分中，选择 `Manual` (手动)。
6. 选择要访问的 EC2 实例，然后选择 `Create` (创建)。

连接到实例并运行命令

Note

以下步骤适用于新的 EC2 控制台。

1. 在 Amazon EC2 控制台的导航窗格中，选择 Instances (实例) ，然后选择要连接的实例。
2. 选择连接。

如果 Connect (连接) 未激活，您可能需要先启动实例。

3. 在 Connect to your instance (连接到您的实例) 窗格中，对于 Connection method (连接方法) ，请选择 Session Manager (会话管理器) ，然后选择 Connect (连接) 。
4. 在出现的终端会话窗口中，输入以下命令。这些命令会创建具有正确权限的 tmp 文件夹，以便 tmux 套接字可用。

```
sudo mkdir /tmp
sudo chmod 777 /tmp
sudo rmdir /tmp/tmux-*
```

无法使用早期版本的 Microsoft Edge 浏览器加载 IDE

问题：尝试使用 Microsoft Edge Web 浏览器加载 AWS Cloud9 IDE 时返回HTTP403：FORBIDDEN错误。

可能的原因：AWS Cloud9 IDE 不支持某些较早版本的Microsoft Edge。

建议的解决方案：要更新浏览器，请选择 Microsoft Edge 工具栏中的省略号 (...) 按钮。从菜单中选择 Settings (设置) ，然后选择 About Microsoft Edge (关于 Microsoft Edge) 。如果需要更新，则会自动下载并安装。

调试 C++ 项目时 gdb 出现错误

问题：尝试在 IDE 中调试 C++ 项目时，gdb 调试程序报告错误。

可能的原因：假设您的 AWS Cloud9 环境使用某些 EC2 实例类型 (例如，t3.small或m5.large) 。然后，当您尝试使用 IDE 的内置运行程序运行和调试 C++ 项目时，

可能会出现调试错误。发生此错误的原因可能是为您的环境预安装的 gdb (GNU 项目调试程序) 版本在某些处理器平台上不适用。您可能会看到以下错误代码。

```
GDB server terminated with code 1
```

建议的解决方案：gdb 不支持某些处理器平台的问题，自 3.0 版本起开始修复。应卸载旧版本的调试程序并升级到 gdb 的新版本：

1. 在 AWS Cloud9 终端中运行以下命令，删除调试器的现有版本。

```
sudo yum -y remove gdb
```

2. 检索 gdb 的归档文件、将其解压，然后通过运行以下命令导航到包含解压文件的目录。

```
wget "http://ftp.gnu.org/gnu/gdb/gdb-8.3.tar.gz"
tar xzf gdb-8.3.tar.gz
cd gdb-8.3
```

3. 通过运行以下命令生成调试器。为此，请将以下文本作为单个块复制并粘贴，然后按回车键运行make。

```
./configure --prefix=/usr \
            --with-system-readline \
            --with-python=/usr/bin/python3 &&
make
```

4. 安装调试程序。

```
sudo make -C gdb install
```

5. 确认调试程序的更新版本已安装完成。

```
gdb --version
```

PHP 运行器出现问题 AWS Cloud9

问题：用户无法在 PHP CLI 运行程序终端中查看任何输出。

原因：CLI 运行程序需要设置为 PHP，并且需要启用调试器模式。

推荐的解决方案：将 CLI 运行程序设置为 PHP，并确保启用调试器模式。

与 Node.js 相关的 GLIBC 错误

问题：用户无法运行 Node.js 并收到 GLIBC 错误。以下是这些错误消息的示例：

```
node: /lib64/libm.so.6: version `GLIBC_2.27' not found (required by node)
node: /lib64/libc.so.6: version `GLIBC_2.28' not found (required by node)
```

原因：可能是与正在使用的实例有关的 Node.js 版本问题。

推荐的解决方案：有关如何为 AWS Cloud9 安装 Node.js 的信息，请参阅 [步骤 1：安装所需工具](#) 部分。

AWS Cloud9 支持的浏览器

下表列出了 AWS Cloud9 支持的浏览器。

浏览器	版本
Google Chrome	最新的三个版本
Mozilla Firefox	最新的三个版本
Microsoft Edge	最新的三个版本
Apple Safari for macOS	最新的两个版本

Warning

如果您将 Mozilla Firefox 作为首选浏览器与 AWS Cloud9 IDE 一起使用，第三方 Cookie 设置会阻止 AWS Cloud9 Webview 和 AWS Toolkit 在浏览器中正常运行。要解决此问题，您必须确保在浏览器设置的“隐私和安全”部分没有阻止 Cookie，如下图所示。

The screenshot shows the Firefox 'Browser Privacy' settings page. On the left is a navigation menu with 'General', 'Home', 'Search', 'Privacy & Security' (highlighted), and 'More from Mozilla'. The main content area is titled 'Browser Privacy' and features 'Enhanced Tracking Protection'. A shield icon is next to the text: 'Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.' A 'Manage Exceptions...' button is to the right. Below this are three protection levels: 'Standard' (selected), 'Strict', and 'Custom'. The 'Custom' section is expanded, showing a list of trackers to block: 'Cookies' (unchecked and highlighted with a red box), 'Tracking content' (checked), 'Cryptominers' (checked), and 'Fingerprinters' (checked). A 'Only in Private Windows' dropdown is next to 'Tracking content'. At the bottom, a message states 'You will need to reload your tabs to apply these' with a 'Reload All Tabs' button.

AWS Cloud9 的限制

下表列出了 AWS Cloud9 以及相关AWS服务中的限制。

- [AWS Cloud9 限制](#)
- [相关 AWS 服务限制](#)

AWS Cloud9 限制

下表提供了 AWS Cloud9 对于 AWS 账户的默认限制。除非另有说明，否则每个区域都有各自的限制。您可以使用 AWS 管理控制台或 AWS CLI 请求提高限额。要请求提高限额，请参阅《服务限额用户指南》中的[请求提高限额](#)。

我们不会立即同意提高这些限制，因此，您的提高请求可能需要几天才能生效。

资源	默认限制	可调整
最大 AWS Cloud9 EC2 开发环境数	<ul style="list-style-type: none"> • 每个用户 100 个 • 每个账户 200 个 	可以
最大 SSH 环境数	<ul style="list-style-type: none"> • 每个用户 100 个 • 每个账户 200 个 	可以
环境中成员的最大数量	<p>默认最大成员数等于该环境的实例的内存量除以 60 MB，结果向下取整。例如，1 GiB 内存的实例最多可拥有 17 个成员（即 1 GiB 除以 60 MB，向下取整）。</p> <p>如果 AWS Cloud9 无法确定实例的内存量，它会默认为与该实例关联的每个环境最多有 8 个用户。</p> <p>环境的绝对最大成员数为 25。</p>	否 ¹

资源	默认限制	可调整
最大可编辑文件大小	8 MB	否

¹ 您可以[移动环境](#)以尝试增加最大成员数。但是，环境的绝对最大成员数仍为 25。

AWS Cloud9 IDE 下载限制

将文件从 AWS Cloud9 IDE 下载到本地文件系统时，传输速度将限制为 0.1 兆字节/秒。要提高文件传输速度，请使用 AWS Cloud9 IDE 中的 CLI 将文件上传到 Amazon S3，然后在 Amazon S3 处下载文件。

相关 AWS 服务限制

Amazon Elastic Block Store (Amazon EBS) 卷的最大数	5000 有关更多信息，请参阅 中的 Amazon Elastic Block Store (Amazon EBS) 限制Amazon Web Services 一般参考。
AWS CloudFormation 堆栈的最大数量	200 有关更多信息，请参阅 AWS CloudFormation 用户指南中的 AWS CloudFormation 限制 。
Amazon EC2 限制	请参阅 中的 Amazon Elastic Compute Cloud (Amazon EC2) 限制Amazon Web Services 一般参考。

AWS Cloud9 用户指南的文档历史记录

本主题包含对 AWS Cloud9 用户指南的重大更改的列表。如需对此文档更新的通知，您可以订阅 RSS 源。

最近的更新

下表介绍了 2019 年 3 月之后对 AWS Cloud9 用户指南的一些重要更改。

变更	说明	日期
添加了对适用于 AWS Cloud9 的 Amazon Linux 2023 的支持。	AWS Cloud9 现在支持 Amazon Linux 2023。	2023 年 12 月 15 日
对 Node.js 教程进行了更新。	对 Node.js 教程进行了与支持 Amazon Linux 2 和 Node.js 18 相关的更新。	2023 年 10 月 23 日
更新了有关使用 Amazon VPC 控制面板创建 Amazon VPC 的相关章节	更新了有关使用 Amazon VPC 控制面板创建 Amazon VPC 的相关章节。	2023 年 7 月 27 日
有关处理 Amazon EventBridge 架构的章节	添加了有关使用 AWS Toolkit for AWS Cloud9 处理 Amazon EventBridge 架构的章节。	2022 年 12 月 15 日
添加了 CodeCatalyst 章节	添加了关于新 Amazon CodeCatalyst 服务的章节。	2022 年 12 月 2 日
添加了 AWS IoT 内容	添加了有关使用 AWS IoT 的章节。	2022 年 11 月 1 日
适用于 AWS Cloud9 IDE 的 Amazon ECS 服务概述	添加了可在 AWS Cloud9 IDE 中访问的 Amazon ECS 服务的特性和功能的概述和演练。	2022 年 10 月 20 日

在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CDK	添加了有关在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS CDK 的章节。	2022 年 10 月 5 日
添加了 Amazon ECR 内容	添加了有关使用 AWS Amazon ECR 的章节。	2022 年 10 月 4 日
合规性验证	更新了其范围涵盖 AWS Cloud9 的合规性计划的列表。	2022 年 3 月 4 日
增强了 Java 支持	提供额外的语言支持，以改善使用 Java 时的开发体验。主要的生产力特性包括代码完成、错误检查、特定于上下文的操作，以及调试选项，如断点和步进。	2022 年 1 月 18 日
已更新 AWSServiceRoleForAWSCloud9	更新了服务相关角色以使用 License Manager 支持 EC2 实例。	2022 年 1 月 12 日
Step Functions 文档支持	添加了介绍如何使用 Step Functions 创建、编辑和运行状态机的内容。	2021 年 12 月 20 日
AWS Systems Manager 文档支持	添加了介绍 Systems Manager 自动化文档的内容。	2021 年 12 月 20 日
为 Amazon Elastic Container Service Exec 创建了用户指南	这是 Amazon ECS Exec 的概述。	2021 年 12 月 13 日
为 AWS IoT AWS Cloud9 IDE 服务创建了用户指南	本用户指南介绍如何开始使用适用于 AWS Cloud9 IDE 的 AWS IoT 服务。	2021 年 11 月 22 日
支持 AWS 资源	添加了对访问资源类型的支持以及用于查看资源和相关文档的界面选项。	2021 年 11 月 5 日

适用于 AWS Cloud9 IDE 的 Amazon ECR 服务概述	添加了可在 AWS Cloud9 IDE 中访问的 Amazon ECR 服务的特性和功能的概述和演练	2021 年 10 月 14 日
App Runner 支持	向 AWS Toolkit 添加了对 AWS App Runner 的支持。	2021 年 9 月 30 日
AWS Cloud9 已在非洲（开普敦）和亚太地区（大阪）区域推出	AWS Cloud9 现已在以下区域推出：非洲（开普敦）和亚太地区（大阪）。有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2021 年 9 月 1 日
AWS Toolkit 中的 CloudWatch Logs 和 Amazon S3	为适用于 AWS Cloud9 的 AWS Toolkit 增加了 CloudWatch Logs 支持。允许将当前文件上载到 Amazon S3 存储桶的新功能。	2021 年 7 月 16 日
Amazon S3 的 VPC 终端节点	添加了对为 Amazon S3 配置 VPC 终端节点的支持，以允许下载依赖项。	2021 年 4 月 22 日
通过 Git 面板提供可视化源代码控制	作为开发人员，您可以使用 Git 面板在用户界面中运行 Git 命令。	2021 年 2 月 1 日
将环境实例启动到私有子网内	添加了对通过 Systems Manager 访问的 EC2 实例的支持，以启动到私有子网内。	2021 年 1 月 21 日
集成 AWS Toolkit	现在，您可以通过 AWS Explorer 窗口，使用 AWS Toolkit 浏览 AWS 服务 并进行交互。	2020 年 12 月 11 日

AWS CloudFormation 和非入口 EC2 环境	有关使用 AWS CloudFormation 模板创建非入口 EC2 环境的已展开文档。	2020 年 10 月 29 日
基于 Amazon Linux 2 的 EC2 环境	在控制台中创建 EC2 环境时，您可以为 EC2 实例选择 Amazon Linux 2 AMI。	2020 年 10 月 7 日
带有 Systems Manager 的非入口 EC2 实例	添加了对使用 AWS Systems Manager 访问私有 EC2 实例的支持。	2020 年 8 月 12 日
增强 AWS 无服务器应用程序的本地调试	增加了对 AWS 无服务器应用程序新的本地调试功能的支持。	2020 年 7 月 30 日
AWS Cloud9 已在欧洲 (米兰) 区域推出	AWS Cloud9 现已在欧洲 (米兰) 区域推出。此有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2020 年 7 月 29 日
Amazon EBS 加密	此部分介绍了如何为 AWS Cloud9 开发环境使用的 EC2 实例加密 Amazon EBS 卷。	2020 年 7 月 3 日
增加了对 AWS Cloud9 的区域支持	AWS Cloud9 现已在以下区域推出：美国西部 (加利福尼亚北部)、亚太地区 (香港)、欧洲 (巴黎)、中东 (巴林) 和南美洲 (圣保罗)。有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2020 年 5 月 7 日

安全性	已在 AWS Cloud9 用户指南中添加“安全性”一章。	2020 年 4 月 30 日
标签	使用标签可以帮助您控制对 AWS Cloud9 资源的访问，并且可帮助您管理账单信息。	2020 年 1 月 22 日
增加了对 AWS Cloud9 的区域支持	AWS Cloud9 现已在以下区域推出：亚太地区（孟买）、亚太地区（首尔）、亚太地区（悉尼）、加拿大（中部）、欧洲（伦敦）和欧洲（斯德哥尔摩）。有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2019 年 12 月 18 日
已更新：故障排除，无法打开环境	不再需要第三方 cookie 来打开 IDE。	2019 年 11 月 6 日
已添加：故障排除，第三方 Cookie 已禁用	不再需要第三方 Cookie 来打开 IDE。但是，它们是应用程序预览或文件预览功能所必需的。您可以在故障排除主题中找到有关这方面的信息。	2019 年 11 月 6 日
文档组织	对用户指南进行了组织更改，以帮助导航，特别是对首次使用的用户。	2019 年 8 月 15 日
AWS Cloud9 已在欧洲（法兰克福）区域推出	AWS Cloud9 现已在欧洲（法兰克福）区域推出。此有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2019 年 5 月 15 日

增加了 LAMP 示例	增加了说明如何将 AWS Cloud9 与 LAMP (Linux、Apache HTTP Server、MySQL 和 PHP) 结合使用的新示例。有关更多信息，请参阅 适用于 AWS Cloud9 的 LAMP 示例 。	2019 年 5 月 10 日
增加了 WordPress 示例	增加了说明如何将 AWS Cloud9 与 WordPress 结合使用的新示例。有关更多信息，请参阅 适用于 AWS Cloud9 的 WordPress 示例 。	2019 年 4 月 19 日
AWS Cloud9 已在亚太地区 (东京) 区域推出	AWS Cloud9 现已在亚太地区 (东京) 区域推出。此有关这些区域和其他 AWS 区域服务端点和服务限额的详细信息，请参阅《Amazon Web Services 一般参考》中的 AWS Cloud9 。	2019 年 4 月 4 日
增加了有关 EC2 环境中对 Ubuntu Server 支持的信息	<p>增加了有关使用 AWS Cloud9 控制台创建连接到 Ubuntu Server 的 AWS Cloud9 EC2 开发环境的信息。有关更多信息，请参阅创建 EC2 环境。</p> <p>请注意，目前您不能使用代码创建连接到 Ubuntu Server 的 AWS Cloud9 EC2 开发环境，例如使用 AWS CLI、AWS CloudFormation、AWS 开发工具包、Tools for Windows PowerShell 或 AWS Cloud9 API。预计在将来会提供对这些方法的支持。</p>	2019 年 4 月 2 日

早期更新

下表列出了 2019 年 4 月前对 AWS Cloud9 用户指南 的一些重要更改。

更改	描述	更改日期
增加了面向学生、教育工作者和企业的入门说明	扩展了 AWS Cloud9 的入门说明，包括面向学生、教育工作者和企业的步骤。有关更多信息，请参阅 设置 AWS Cloud9 。	2019 年 2 月 7 日
增加了对 AWS CloudTrail 的支持	AWS CloudTrail 现在支持 AWS Cloud9。有关更多信息，请参阅 使用 AWS Cloud9 记录 AWS CloudTrail API 调用 。	2019 年 1 月 21 日
增加了对共享 VPC 的支持	AWS Cloud9 现在支持 Amazon VPC 中的共享 VPC。有关更多信息，请参阅 亚马逊 VPC 对以下内容的要求 AWS Cloud9 。	2018 年 12 月 7 日
增加了 AWS RoboMaker 集成	AWS Cloud9 现在支持 AWS RoboMaker，该服务可以帮助您轻松实现大规模开发、测试和部署智能机器人应用程序。有关更多信息，请参阅 AWS RoboMaker 开发人员指南中的 AWS RoboMaker 入门 和 通过 AWS Cloud9 进行开发 。	2018 年 11 月 26 日
增加了有关针对语言项目的额外生产功能的信息	AWS Cloud9 IDE 现在针对语言项目环境中的某些语言提供更多的生产力功能。有关更多信息，请参阅 增强 TypeScript 支持和特征 。	2018 年 10 月 2 日

更改	描述	更改日期
增加了 Go (前往) 窗口 ; 删除了 Navigate (导航) 和 Commands (命令) 窗口	添加了 Go (前往) 窗口到适用于 2018 年 10 月 2 日或之后创建的环境的 AWS Cloud9 IDE。这一新窗口取代了 Navigate (导航) 和 Commands (命令) 窗口 , 对于 2018 年 10 月 2 日或之后创建的环境 , 已从 IDE 中删除后二者。有关更多信息 , 请参阅 了解 IDE 中的 步骤 10 : “Go (转到)”窗口 。	2018 年 10 月 2 日
增加了 AWS CDK 示例	增加了一个新的示例 , 用来演示如何将 AWS Cloud9 与 AWS Cloud Development Kit (AWS CDK) 结合起来使用。有关更多信息 , 请参阅 适用于 AWS Cloud9 的 AWS CDK 教程 。	2018 年 8 月 30 日
增加了有关自动添加到已添加 EC2 环境中的 SSH IP 地址限制的信息	对于在 2018 年 7 月 31 日或之后创建的 AWS Cloud9 EC2 开发环境 , AWS Cloud9 现在自动将传入 SSH 流量限制为仅 AWS Cloud9 用来通过 SSH 进行连接的 IP 地址范围。有关更多信息 , 请参阅 AWS Cloud9 的入站 SSH IP 地址范围 。	2018 年 7 月 31 日
增加了 Docker 示例	增加了说明如何将 AWS Cloud9 与 Docker 结合使用的新示例。有关更多信息 , 请参阅 适用于 Docker 的教程 AWS Cloud9 。	2018 年 6 月 19 日

更改	描述	更改日期
增加了 Java、.NET Core 和 TypeScript 的示例	增加了演示如何将 AWS Cloud9 与 Java、.NET Core 和 TypeScript 一起使用的新示例。有关更多信息，请参阅 适用于 AWS Cloud9 的 Java 教程 、 适用于 AWS Cloud9 的 .NET 教程 和 适用于 AWS Cloud9 的 TypeScript 教程 。	2018 年 5 月 29 日
增加了支持的浏览器列表	增加了有关 AWS Cloud9 支持的浏览器的信息。有关更多信息，请参阅 AWS Cloud9 支持的浏览器 。	2018 年 5 月 23 日
增加了 SSH IP 流量限制信息	增加了如何将传入流量限制为仅限 AWS Cloud9 用于通过 SSH 连接到主机的 IP 地址范围的相关信息。有关更多信息，请参阅 AWS Cloud9 的入站 SSH IP 地址范围 。	2018 年 4 月 19 日
增加了针对预览应用程序和共享运行中的应用程序的故障排除技巧	添加了有关预览应用程序和共享运行中的应用程序的新故障排除技巧。有关更多信息，请参阅 应用程序预览选项卡显示错误或空白 和 无法显示在 IDE 以外运行的应用程序 。	2018 年 4 月 19 日
增加了 File Revision History (文件修订历史记录) 信息	增加了有关如何使用 IDE 中的 File Revision History (文件修订历史记录) 窗格的信息。有关更多信息，请参阅 在 AWS Cloud9 集成开发环境 (IDE) 中使用文件修订 。	2018 年 4 月 19 日

更改	描述	更改日期
增加了针对打开环境的故障排除技巧	添加了关于打开 AWS Cloud9 开发环境的新故障排除技巧。有关更多信息，请参阅 无法打开环境 。	2018 年 3 月 19 日
增加了针对 AWS Cloud9 安装程序的故障排除技巧	增加了有关 AWS Cloud9 安装程序的新故障排除技巧。有关更多信息，请参阅 AWS Cloud9 安装程序挂起或失败 。	2018 年 3 月 19 日
增加了 AWS CodePipeline 信息。	增加了有关如何将 AWS Cloud9 与 AWS CodePipeline 结合使用的信息。有关更多信息，请参阅 在 AWS Cloud9 集成开发环境 (IDE) 中使用 AWS CodePipeline 。	2018 年 2 月 13 日
增加了 AWS CloudShell 信息。	增加了有关如何将 AWS Cloud9 与 AWS CloudShell 结合使用的信息。有关更多信息，请参阅 适用于 AWS Cloud9 的 AWS Command Line Interface 和 aws-shell 教程 。	2018 年 1 月 19 日
增加了在 GitHub 上提供的文档	本指南现在通过 GitHub 提供。您也可以使用 GitHub 提交反馈和更改本指南内容的请求。有关更多信息，请在指南的导航栏中选择 Edit on GitHub (在 GitHub 上编辑) 图标，或者参阅 GitHub 网站上的 awsdocs/aws-cloud9-user-guide 存储库。	2018 年 1 月 10 日

更改	描述	更改日期
提供 Kindle 格式	本指南现在支持 Amazon Kindle 格式。有关更多信息，请在指南的导航栏中选择打开 Kindle 图标。	2018 年 1 月 2 日
增加了 Amazon Lightsail 信息。	增加了有关如何将 AWS Cloud9 与 Amazon Lightsail 结合使用的信息。有关更多信息，请参阅 在 AWS Cloud9 集成式开发环境 (IDE) 中使用 Amazon Lightsail 实例 。	2017 年 12 月 19 日
增加了针对用于 AWS 的环境设置描述	增加了对 AWS Cloud9 开发环境的指定 AWS 设置的描述。有关更多信息，请参阅 在 AWS Cloud9 集成式开发环境 (IDE) 中使用 AWS 项目和用户设置 。	2017 年 12 月 7 日
针对 AWS 账户根用户以及团队的高级设置步骤增加了入门说明	增加了使用 AWS Cloud9 账户根用户身份使用 AWS 的设置步骤。增加团队使用 AWS Cloud9 的高级设置步骤。有关更多信息，请参阅 设置 AWS Cloud9 。	2017 年 12 月 5 日
扩展了对环境要求的介绍	扩展了对 Amazon EC2 实例或您自己的服务器连接到 AWS Cloud9 SSH 开发环境的要求的介绍。有关更多信息，请参阅 SSH 环境主机要求 。	2017 年 12 月 4 日
初始文档版本	这是 AWS Cloud9 用户指南的初始版本。	2017 年 11 月 30 日

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。