

使用者指南

AWS Cloud9



AWS Cloud9: 使用者指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任從何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS Cloud9 ?	1
AWS Cloud9 如何運作?	1
AWS Cloud9 環境	1
環境與運算資源	2
我可以用 AWS Cloud9 做什麼?	2
我該如何開始?	2
其他主題	3
我可以用它做什麼?	3
其他資訊	5
相關影片	5
AWS 網站上的相關主題	5
定價	6
我有其他問題或需要協助	6
設定 AWS Cloud9	7
個別使用者設定	7
註冊 AWS 帳戶	8
建立管理使用者	8
其他驗證方式	9
後續步驟	10
小組設定	10
註冊 AWS 帳戶	8
建立管理使用者	8
步驟 2：建立 IAM 群組和使用者並將使用者新增至群組	13
步驟 3：為群組新增 AWS Cloud9 存取許可	18
步驟 4：登入 AWS Cloud9 主控台	22
後續步驟	22
企業設定	23
步驟 1：建立組織的管理帳戶	25
步驟 2：為管理帳戶建立組織	26
步驟 3：將成員帳戶新增至組織	26
步驟 4：在整個組織啟用 IAM Identity Center	27
步驟 5. 在組織內設定群組和使用者	28
步驟 6：讓組織內的群組和使用者使用 AWS Cloud9	28
步驟 3：開始使用 AWS Cloud9	30

後續步驟	31
其他設定選項 (小組和企業)	32
步驟 1：建立客戶管理政策	32
步驟 2：將客戶受管政策新增至群組	34
使用 AWS Cloud9 的小組適用的客戶受管政策範例	35
後續步驟	40
入門：基本教學課程	41
Hello AWS Cloud9 (主控台)	41
必要條件	41
步驟	41
步驟 1：建立環境	42
步驟 2：基本導覽	46
步驟 3：清理	51
相關資訊	53
Hello AWS Cloud9 (CLI)	55
必要條件	55
步驟	55
步驟 1：建立環境	56
步驟 2：基本導覽	58
步驟 3：清理	63
相關資訊	64
使用環境	66
建立環境	66
建立 EC2 環境	67
建立 SSH 環境	82
使用 Systems Manager 存取無輸入的 EC2 執行個體	87
為 EC2 環境使用 Systems Manager 的好處	88
管理 Systems Manager 許可	90
讓使用者存取由 Session Manager 管理的執行個體	91
使用 AWS CloudFormation 建立無輸入 EC2 環境	94
將 Amazon S3 的 VPC 端點設為下載相依性	96
將 VPC 端點設為使用私有連線	100
開啟環境	100
從環境呼叫 AWS 服務	102
建立及使用執行個體描述檔來管理臨時憑證	103
在環境中建立並存放永久存取憑證	109

變更環境設定	112
變更環境偏好設定	113
使用主控台變更環境設定	113
使用程式碼變更環境設定	115
使用共享環境	116
共用環境使用案例	116
關於環境成員的存取角色	117
邀請所在帳戶與環境相同的使用者	119
讓所在帳戶與環境相同的 AWS Cloud9 管理員邀請自己或其他使用者	121
開啟共用環境	122
參閱環境成員清單	123
開啟環境成員的作用中檔案	124
開啟環境成員的開放檔案	124
移至環境成員的作用中游標	125
與其他環境成員聊天	125
檢視共用環境的聊天訊息	125
從共用環境刪除聊天訊息	125
從共用環境刪除所有聊天訊息	126
變更環境成員的存取角色	126
從共用環境移除使用者	127
移除其他環境成員	129
共用環境最佳實務	130
移動環境並調整 Amazon EBS 磁碟區大小或予以加密	130
如何移動環境	131
將 AWS Cloud9 EC2 環境移動到不同的 Amazon 機器映像 (AMI)	133
為環境使用的 Amazon EBS 磁碟區調整大小	138
加密使 AWS Cloud9 用的 Amazon EBS 磁碟區	140
刪除環境	143
使用主控台刪除環境	144
使用程式碼刪除環境	146
使用 IDE	148
導覽 IDE	149
先決條件	150
步驟 1：選單列	150
步驟 2：儀表板	152
步驟 3：環境視窗	152

步驟 4：編輯器、標籤和窗格	153
步驟 5：主控台	155
步驟 6：開啟檔案區段	156
步驟 7：裝訂邊	156
步驟 8：狀態列	157
步驟 9：大綱視窗	158
步驟 10：執行視窗	159
步驟 11：即時運算標籤	161
步驟 12：程序清單	162
步驟 13：偏好設定	163
步驟 14：終端機	164
步驟 15：除錯器視窗	165
結論	171
語言支援	171
AWS Cloud9IDE 中支援的程式設計語言版本	173
增強的語言支援	174
增強的 Java 支援	174
增強的TypeScript支援	182
選單命令參考	186
AWS Cloud9 選單	186
File (檔案) 選單	187
Edit (編輯) 選單	188
Find (尋找) 選單	192
View (檢視) 選單	193
Go (執行) 選單	194
Run (執行) 選單	195
Tools (工具) 選單	196
Window (視窗) 選單	197
Support (支援) 選單	199
Preview (預覽) 選單	200
其他選單列命令	200
尋找及取代文字	201
尋找單一檔案中的文字	201
取代單一檔案中的文字	201
尋找多個檔案中的文字	202
取代多個檔案中的文字	203

尋找及取代選項	204
預覽檔案	205
開啟檔案進行預覽	205
重新載入檔案預覽	206
變更檔案預覽類型	207
在個別的 Web 瀏覽器標籤中開啟檔案預覽	207
切換至不同的檔案預覽	207
預覽執行中的應用程式	207
執行應用程式	208
預覽執行中的應用程式	210
重新載入應用程式預覽	211
變更應用程式預覽類型	211
另由其他 Web 瀏覽器標籤開啟應用程式預覽	212
切換到另一個預覽 URL	212
透過網際網路共用執行中的應用程式	212
使用檔案修訂	217
使用影像檔案	219
檢視或編輯影像	219
調整影像大小	220
裁剪影像	220
旋轉影像	221
翻轉影像	221
縮放影像	221
平滑影像	221
使用建置器、執行器和除錯器	221
內建組建、執行和除錯支援	222
建置專案檔案	222
執行您的程式碼	223
除錯您的程式碼	223
變更內建執行器	224
建立執行組態	225
建立建置器或執行器	225
定義建置器或執行器	226
使用自訂環境變數	230
設定命令層級自訂環境變數	230
在 <code>~/.bash_profile</code> 中設定自訂使用者環境變數	230

設定本機自訂環境變數	231
在 ~/.bashrc 中設定自訂使用者環境變數	231
在 ENV 清單中設定自訂環境變數	231
使用專案設定	232
檢視或變更專案設定	232
將一個環境的目前專案設定套用至另一個環境	233
您可以變更的專案設定	233
手動停止環境的 EC2 執行個體	240
使用使用者設定	241
檢視或變更您的使用者設定	242
與其他使用者共享您的使用者設定	242
您所做的使用者設定變更	242
使用 AWS 專案和使用者設定	251
專案層級設定	251
使用者層級設定	252
處理按鍵繫結關係	252
檢視或變更您的按鍵繫結關係	252
與另一名使用者共享您的按鍵繫結關係	253
變更您的鍵盤模式	253
變更您的作業系統按鍵繫結關係	254
變更特定的按鍵繫結關係	254
移除您的所有自訂按鍵繫結關係	255
使用佈景主題	256
檢視或變更您的佈景主題	256
您可以變更的整體佈景主題設定	256
佈景主題覆寫	257
管理初始化指令碼	257
開啟初始化指令碼	257
MacOS 預設按鍵繫結關係參考	258
一般	258
標籤	261
面板	263
程式碼編輯器	264
emmet	270
終端機	271
執行與除錯	271

MacOS Vim 按鍵繫結關係參考	272
一般	273
標籤	275
面板	277
程式碼編輯器	278
emmet	285
終端機	285
執行與除錯	286
MacOS Emacs 按鍵繫結關係參考	286
一般	287
標籤	290
面板	292
程式碼編輯器	292
emmet	299
終端機	300
執行與除錯	300
MacOS Sublime 按鍵繫結關係參考	301
一般	301
標籤	305
面板	307
程式碼編輯器	308
emmet	315
終端機	315
執行與除錯	316
Windows / Linux 預設按鍵繫結關係參考	316
一般	317
標籤	320
面板	322
程式碼編輯器	322
emmet	329
終端機	330
執行與除錯	330
Windows / Linux Vim 按鍵繫結關係參考	330
一般	331
標籤	334
面板	336

程式碼編輯器	336
emmet	343
終端機	343
執行與除錯	344
Windows / Linux Emacs 按鍵繫結關係參考	344
一般	345
標籤	347
面板	350
程式碼編輯器	350
emmet	357
終端機	357
執行與除錯	358
Windows / Linux Sublime 按鍵繫結關係參考	358
一般	359
標籤	363
面板	365
程式碼編輯器	365
emmet	372
終端機	372
執行與除錯	373
命令參考	373
使用其他 AWS 服務	375
使用 Amazon Lightsail 執行個體	375
步驟 1：建立 Linux 型 Lightsail 執行個體	376
步驟 2：設定執行個體以搭配 AWS Cloud9 使用	379
步驟 3：建立並連接至 AWS Cloud9 SSH 開發環境	381
步驟 4：使用 AWS Cloud9 IDE 變更執行個體上的程式碼	383
使用 AWS CodeStar 專案	384
步驟 1：準備使用 AWS CodeStar 專案	385
步驟 2：在中建立專案 AWS CodeStar	385
步驟 3：建立 AWS Cloud9 開發環境並將其連接至專案	385
使用 CodeWhisperer	386
什麼是 CodeWhisperer？	386
啟用的 IAM 許可 CodeWhisperer	386
使用 AWS CodePipeline	387
步驟 1：建立或識別來源碼儲存庫	387

步驟 2：建立 AWS Cloud9 開發環境、將其連線至程式碼儲存庫，並上傳程式碼	388
步驟 3：準備使用 AWS CodePipeline	389
步驟 4：在 AWS CodePipeline 中建立管道	389
使用 CodeCatalyst	390
開始使用 CodeCatalyst	390
在 Amazon 中複製 AWS Cloud9 代碼資源 CodeCatalyst	391
使用複製工具	402
有關複寫程序的常見問題	405
開發環境 CodeCatalyst	407
使用 AWS CDK	412
AWS CDK 應用程式	412
使用 Git 面板的視覺化來源控制	414
使用 Git 面板管理來源控制	415
初始化或複製 Git 儲存庫	417
暫存和遞交檔案	420
檢視不同的檔案版本	422
使用分支	423
使用遠端儲存庫	426
隱藏和擷取檔案	428
參考：Git 面板中提供的 Git 命令	428
Git 面板選單中提供的 Git 命令參考資料	430
Git 面板搜尋欄位中可用的 Git 命令	432
AWS Toolkit	435
為什麼要使用 AWS Toolkit？	435
啟用 AWS Toolkit	437
管理 AWS Toolkit 的存取憑證	438
使用 IAM 角色為 EC2 執行個體上的應用程式授予許可	439
識別 AWS Toolkit 元件	439
停用 AWS Toolkit	440
AWS Toolkit 主題	441
導覽和設定	441
透過 AWS Explorer 來使用多個區域的服務和資源	441
存取和使用 AWS 工具組選單	442
使用 AWS 組態窗格修改 AWS 工具組設定	444
API Gateway	447
叫用 REST API	447

AWS App Runner	448
必要條件	448
定價	451
建立 App Runner 服務	451
管理 App Runner 服務	454
AWS CloudFormation 堆疊	456
刪除 AWS CloudFormation 堆疊	456
Amazon CloudWatch Logs	457
檢視 CloudWatch 日誌群組和日誌串流	457
使用 CloudWatch log 事件	458
AWS Lambda 函式	459
叫用遠端 Lambda 函式	460
下載、上傳並刪除 Lambda 函式	461
資源	466
存取資源的 IAM 許可	466
與現有資源互動	466
Simple Storage Service (Amazon Simple Storage Service (Amazon S3))	467
使用 Amazon S3 儲存貯體	467
使用 Amazon S3 物件	469
AWS無伺服器應用	472
建立無伺服器應用程式	472
執行與除錯無伺服器應用程式	474
同步無伺服器應用程式	481
啟用 AWS Toolkit 程式碼濾鏡	482
刪除無伺服器應用程式	483
除錯無伺服器應用程式的組態選項	483
AWS Step Functions	486
先決條件	486
建立和發佈狀態機器	487
在 AWS 工具組中運行狀態機器	489
下載狀態機器定義檔案並視覺化其工作流程	489
AWS Systems Manager	490
假設和先決條件	491
Systems Manager 自動化文件的 IAM 許可	491
建立新的 Systems Manager 自動化文件	491
發佈 Systems Manager 自動化文件	492

編輯現有的 Systems Manager 自動化文件	493
處理版本	493
刪除 Systems Manager 自動化文件	494
運行 Systems Manager 自動化文件	494
故障診斷	495
Amazon ECR	495
先決條件	496
搭配 IDE 使用 Amazon ECR AWS Cloud9	496
AWS IoT	505
AWS IoT 先決條件	505
AWS IoT 物件	505
AWS IoT 憑證	507
AWS IoT 政策 :	509
Amazon ECS	512
Amazon ECS Exec	512
Amazon EventBridge	514
使用 Amazon EventBridge 結構描述	515
AWS Cloud9 教學課程	517
AWS CLI和 aws 外殼教程	517
必要條件	518
步驟 1 : 在環境中安裝 AWS CLI、aws-shell 或同時安裝	518
步驟 2 : 在環境中設定憑證管理	520
步驟 3 : 在環境中使用 AWS CLI 或 aws-shell 執行基本命令	520
步驟 4 : 清理	521
AWS CodeCommit 教學課程	522
先決條件	522
步驟 1 : 設定具有必要存取許可的 IAM 群組	522
步驟 2 : 在中建立儲存庫 CodeCommit	524
步驟 3 : 將環境連線至遠端儲存庫	525
步驟 4 : 將遠端儲存庫複製到您的環境	526
步驟 5 : 將檔案新增至儲存庫	527
步驟 6 : 清除	529
亞馬遜教學	529
先決條件	530
步驟 1 : 在環境中安裝和設定 AWS CLI、AWS CloudShell , 或兩者兼具	530
步驟 2 : 建立資料表	531

步驟 3：新增項目到資料表	533
步驟 4：新增多個項目到資料表	534
步驟 5：建立全域輔助索引	537
步驟 6：取得資料表中的項目	540
步驟 7：清除	544
AWS CDK 教學課程	545
先決條件	545
步驟 1：安裝必要工具	545
步驟 2：新增程式碼	549
步驟 3：執行程式碼	551
步驟 4：清理	553
燈泡教學	553
先決條件	554
步驟 1：安裝工具	554
步驟 2：設定 MySQL	556
步驟 3：設定網站	558
步驟 4：清理	562
WordPress 教學課程	563
先決條件	563
安裝概觀	564
步驟 1：安裝與設定 MariaDB 伺服器	564
步驟 2：安裝和配置 WordPress	565
步驟 3：設定 Apache HTTP 伺服器	566
步驟 4：預覽 WordPress 網頁內容	567
管理混合內容錯誤	567
爪哇教程	568
先決條件	569
步驟 1：安裝必要工具	569
步驟 2：新增程式碼	571
步驟 3：建置並執行程式碼	571
步驟 4：進行使用 AWS SDK for Java 的設定	572
步驟 5：在環境中設定 AWS 憑證管理	578
步驟 6：新增 AWS SDK 程式碼	578
步驟 7：建置並執行 AWS SDK 程式碼	580
步驟 8：清除	581
C ++ 教程	581

先決條件	582
步驟 1：安裝 g++ 和所需的開發套件	582
步驟 2：安裝 CMake	583
步驟 3：取得並建置適用於 C++ 的 SDK	583
第 4 步：創建 C++ 和 C MakeLists 文件	585
步驟 5：建置並執行 C++ 程式碼	589
步驟 6：清除	590
Python 教學課程	590
先決條件	590
步驟 1：安裝 Python	591
步驟 2：新增程式碼	592
步驟 3：執行程式碼	592
步驟 4：安裝並設定 AWS SDK for Python (Boto3)	593
步驟 5：新增 AWS SDK 程式碼	594
步驟 6：執行 AWS SDK 程式碼	595
步驟 7：清除	596
.NET 教程	596
先決條件	597
步驟 1：安裝必要工具	597
步驟 2 (選用)：為 Lambda 函數安裝 .NET CLI 擴充功能	599
步驟 3：建立 .NET 主控台應用程式專案	600
步驟 4：新增程式碼	600
步驟 5：建置並執行程式碼	601
步驟 6：建立並設定 .NET 主控台應用程式專案，該專案使用 AWS SDK for .NET	603
步驟 7：新增 AWS SDK 程式碼	604
步驟 8：建置並執行 AWS SDK 程式碼	606
步驟 9：清除	607
Node.js 教學課程	607
必要條件	607
步驟 1：安裝必要工具	608
步驟 2：新增程式碼	609
步驟 3：執行程式碼	610
步驟 4：JavaScript 在 Node.js 中安裝和配置 AWS 軟體開發套件	610
步驟 5：新增 AWS SDK 程式碼	612
步驟 6：執行 AWS SDK 程式碼	615
步驟 7：清除	616

PHP 教程	616
必要條件	617
步驟 1：安裝必要工具	617
步驟 2：新增程式碼	619
步驟 3：執行程式碼	619
步驟 4：安裝並設定 AWS SDK for PHP	620
步驟 5：新增 AWS SDK 程式碼	621
步驟 6：執行 AWS SDK 程式碼	623
步驟 7：清除	624
Ruby	624
前往教學	624
必要條件	625
步驟 1：安裝必要工具	625
步驟 2：新增程式碼	626
步驟 3：執行程式碼	627
步驟 4：安裝並設定 AWS SDK for Go	628
步驟 5：新增 AWS SDK 程式碼	630
步驟 6：執行 AWS SDK 程式碼	632
步驟 7：清除	633
TypeScript 教學課程	633
先決條件	633
步驟 1：安裝必要工具	634
步驟 2：新增程式碼	635
步驟 3：執行程式碼	636
步驟 4：安裝和配置AWS適用於的 SDKJavaScript在 Node.js 中	637
步驟 5：新增 AWS SDK 程式碼	638
步驟 6：執行 AWS SDK 程式碼	640
步驟 7：清除	641
泊塢視窗教程	641
必要條件	642
步驟 1：安裝和執行 Docker	642
步驟 2：建置影像	643
步驟 3：執行容器	646
步驟 4：建立環境	648
步驟 5：執行程式碼	653
步驟 6：清除	653

相關教學	655
AWS Cloud9 進階主題	656
比較 EC2 環境與 SSH 環境	656
Amazon VPC 設定	657
AWS Cloud9 的 Amazon VPC 需求	658
建立 VPC 以及其他 VPC 資源	670
僅建立 VPC	671
為 AWS Cloud9 建立子網路	672
將子網路設定為公有或私有	674
SSH 環境主機要求	675
建立 SSH 環境環境的時機與方法	676
SSH 主機要求	677
AWS Cloud9 安裝程式	679
下載並執行 AWS Cloud9 安裝程式	680
AWS Cloud9 安裝程式疑難排解	680
傳入 SSH IP 地址範圍	682
不在 ip-ranges.json 中的 IP 地址	683
AMI 內容	684
Amazon Amazon	684
Ubuntu Server	686
服務連結角色	687
AWS Cloud9 的服務連結角色許可	687
為 AWS Cloud9 建立服務連結角色	691
為 AWS Cloud9 編輯服務連結角色	691
為 AWS Cloud9 刪除服務連結角色	691
AWS Cloud9 服務連結角色的支援區域	692
使用 CloudTrail 日誌記錄 API 呼叫	692
CloudTrail 中的 AWS Cloud9 資訊	692
了解 AWS Cloud9 日誌檔案項目	694
標籤	709
將標籤更新傳播至基礎資源	710
安全性 AWS Cloud9	712
資料保護	712
資料加密	713
身分和存取權管理	715
物件	716

使用身分驗證	716
使用政策管理存取權	719
如何與 IAM AWS Cloud9 搭配使用	720
身分型政策範例	726
故障診斷	729
如何使 AWS Cloud9 用 IAM 資源和作業	730
AWS 受管理政策	734
建立客戶管理的政策 AWS Cloud9	744
AWS Cloud9 權限參考	758
AWS 受管理的臨時憑	763
日誌記錄和監控	768
監視活動 CloudTrail	768
監控 EC2 環境效能	769
法規遵循驗證	769
恢復能力	773
基礎架構安全	774
軟體更新與修補	774
安全最佳實務	775
AWS Cloud9 疑難排解	776
Installer (安裝程式)	776
AWS Cloud9 安裝程式停止回應或失敗	776
AWS Cloud9 安裝程式未在顯示「Package Cloud9 IDE 1」(套件 Cloud9 IDE 1) 之後完成 ...	776
安裝相依性失敗	777
SSH 環境錯誤:Python 安裝 pty.js 需要三版本」	778
AWS Cloud9 環境	778
環境建立錯誤：「我們無法建立 EC2 執行個體...」	778
環境建立錯誤：「未授權執行 sts : AssumeRole」	778
聯合身分無法建立環境	779
主控台錯誤：「User is not authorized to perform action on resource」(使用者未獲授權對資源執行動作)	779
無法連線到環境	780
無法開啟環境	780
無法開啟 AWS Cloud9 環境：「This environment cannot be currently accessed by collaborators. Please wait until the removal of managed temporary credentials is complete, or contact the owner of this environment.」(此環境目前無法供協作者存取。請等候受管臨時憑證的移除作業完成，或連絡此環境的擁有者。)	782

環境刪除錯誤：「One or more environments failed to delete」(無法刪除一或多個環境)	783
變更 AWS Cloud9 IDE 中環境的逾時時間	783
在本機 AWS Toolkit 執行 SAM 應用程式時發生錯誤，因為 AWS Cloud9 環境沒有足夠的磁碟空間	784
無法使用舊版 Microsoft Edge 瀏覽器載入 IDE	785
無法在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment。	785
無法在 IDE 的文件資源管理器中創建子文件夾結構/項目/項目。AWS Cloud9 CodeCatalyst .	785
因為 tmux 工作階段發生錯誤，所以無法與 AWS Cloud9 中的終端機視窗互動。	786
Amazon EC2	787
Amazon EC2 執行個體未自動更新	787
EC2 環境中的 AWS CLI 或 AWS-shell 錯誤：「The security token included in the request is invalid」(包含在請求中的安全權杖無效)	788
無法連線到 EC2 環境，因為 VPC 的 IP 地址是由 Docker 使用	788
無法在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment。	785
無法從主控台啟動 AWS Cloud9，原因是 AWS License Manager 授權組態與 Amazon EC2 執行個體相關聯	789
無法在 EC2 環境中執行某些命令或指令碼	790
使用創建 EC2 環境時，錯誤消息報告「實例配置文 AWSCloud9SSMInstanceProfile 件不存在於帳戶中」AWS CloudFormation	790
使用 AWS CloudFormation 建立 EC2 環境時，有錯誤訊息回報「未授權對資源執行 perform: ssm:StartSession」	790
使用 AWS CLI 建立 EC2 環境時，有錯誤訊息回報未授權「在資源：執行個體設定檔 AWSCloud9SSMInstanceProfile 上執行：iam:GetInstanceProfile」	791
將預設加密套用至 Amazon EBS 磁碟區時，無法建立環境	791
EC2-Classic 帳戶的 VPC 錯誤：「無法存取您的環境」	792
其他 AWS 服務	792
無法在 IDE 的文件資源管理器中創建子文件夾結構/項目/項目。AWS Cloud9 CodeCatalyst .	785
無法從 IDE 外部顯示執行中的應用程式	793
執行 AWS Toolkit 時發生錯誤：「Your environment is running out of inodes, please increase 'fs.inotify.max_user_watches' limit.」(您的環境即將用盡 Inode，請提高『fs.inotify.max_user_watches』的上限。)	795
Lambda 本機函數執行錯誤：無法安裝 SAM Local	795
AWS Control Tower嘗試使用以下方式創建 Amazon EC2 環境時出錯AWS Cloud9：「環境創建失敗，出現錯誤：以下鉤子失敗：[:ControlTower: Guard:: Hook]」。	796

將預設加密套用至 Amazon EBS 磁碟區時，無法建立環境	791
無法從主控台啟動 AWS Cloud9，原因是 AWS License Manager 授權組態與 Amazon EC2 執行個體相關聯	789
申請預覽	797
重新載入環境後，您必須重新整理應用程式預覽	797
應用程式預覽或檔案預覽通知：「Third-party cookies disabled」(已停用第三方 Cookie)	797
應用程式預覽標籤顯示錯誤或一片空白	801
無法在 IDE 中預覽 Web 內容，因為與網站的連線不安全	803
預覽檔案時傳回 499 錯誤	803
效能	803
AWS Cloud9IDE 凍結大量時間	803
主控台警告：「Switching to the minimal code completion engine...」(切換到最少程式碼完成 引擎...)	804
IDE 警告：「This environment is running low on memory」(此環境的記憶體過低) 或「This environment has high CPU load」(此環境有高負載 CPU)	804
無法在 AWS Cloud9 IDE 中上傳檔案	805
AWS Cloud9IDE 中的下載速度慢	805
無法在 IDE 中預覽 Web 內容，因為與網站的連線不安全	803
第三方應用程式和服務	806
因為 tmux 工作階段發生錯誤，所以無法與 AWS Cloud9 中的終端機視窗互動。	786
無法使用舊版 Microsoft Edge 瀏覽器載入 IDE	785
除錯 C++ 專案時發生 gdb 錯誤	808
PHP 運行器的問題 AWS Cloud9	809
與 Node.js 相關的格林錯誤	809
支援的瀏覽器	810
限制	812
AWS Cloud9 限制	812
AWS Cloud9IDE 下載限制	813
相關 AWS 服務限制	813
文件歷史紀錄	814
.....	dcccxxv

什麼是 AWS Cloud9 ？

AWS Cloud9 是整合開發環境，或稱 IDE。

AWS Cloud9 IDE 提供豐富的程式碼編輯體驗，可支援多種程式設計語言和執行時間除錯器，以及內建終端機。其中包含用於編碼、建置、執行、測試、除錯以及在雲端中發行軟體的工具集合。

您可以透過 Web 瀏覽器存取 AWS Cloud9 IDE。您可以將 IDE 設為您的偏好設定。您可以切換色彩主題、繫結快速鍵、啟用程式設計語言特定語法著色和程式碼格式編排等等。

(瞭解！我已經準備好試用 AWS Cloud9 了。[我該如何開始？](#))

AWS Cloud9 如何運作？

下圖顯示 AWS Cloud9 運作方式的高階概觀。

在圖表 (從底部開始) 中，您會使用在本機電腦的 Web 瀏覽器中執行的 AWS Cloud9 IDE 與您的 AWS Cloud9 環境互動。運算資源 (例如 Amazon EC2 執行個體或您自己的伺服器) 會連接到該環境。最後，您的工作會存放在 AWS CodeCommit 儲存庫或其他類型的遠端儲存庫。



AWS Cloud9 環境

AWS Cloud9 環境是您存放專案檔案以及執行應用程式開發工具的位置。

使用 AWS Cloud9 IDE，您可以：

- 將專案的檔案存放在執行個體或伺服器本機。
- 將遠端程式碼儲存庫 (例如 AWS CodeCommit 中的儲存庫) 複製到環境。
- 在環境中使用本機與複製檔案的組合。

您可以建立與切換多個環境，其中的每個環境都已為特定開發專案做好設定。在雲端中存放環境，您的專案就不再需要繫結至單一電腦或伺服器設定。這可讓您執行數個事項，例如輕鬆地切換電腦，以及更快地讓開發人員融入您的團隊。

環境與運算資源

在幕後，您有幾種方式可將您的環境連接到運算資源：

- 您可以指示 AWS Cloud9 建立 Amazon EC2 執行個體，然後將環境連接至新建立的 EC2 執行個體。我們將此類型的設定稱為 EC2 環境。
- 您可以指示 AWS Cloud9 將環境連線至現有雲端運算執行個體或自有伺服器。我們將此類型的設定稱為 SSH 環境。

EC2 環境和 SSH 環境彼此有些異同之處。如果您剛開始使用 AWS Cloud9，建議您使用 EC2 環境，因為 AWS Cloud9 會為您處理大部分的組態。當您進一步了解 AWS Cloud9，並且想要了解這些異同點時，請參閱 [比較 AWS Cloud9 中的 EC2 環境與 SSH 環境](#)。

如需 AWS Cloud9 運作方式的詳細資訊，請參閱這些相關的[影片](#)和[網頁](#)。

我可以用 AWS Cloud9 做什麼？

您可以使用 AWS Cloud9，在許多令人興奮的案例和相似案例中編寫、建置、執行、測試、偵錯和發行軟體。其中包括 (但不限於)：

- 使用以多種程式設計語言和 AWS Cloud Development Kit (AWS CDK) 撰寫的程式碼。
- 在執行中的 Docker 容器中使用程式碼。
- 使用線上程式碼儲存庫。
- 與他人即時協作。
- 與各種資料庫和網站技術互動。
- 以 AWS Lambda、Amazon API Gateway 和 AWS 無伺服器應用程式為目標。
- 利用其他 AWS 產品，如 Amazon Lightsail, AWS CodeStar, 和 AWS CodePipeline。

如需更詳細的清單，請參閱 [我可以用 AWS Cloud9 做什麼？](#)

我該如何開始？

若要開始使用 AWS Cloud9，請遵循 [設定 AWS Cloud9](#) 中的步驟，然後進行[基本教學課程](#)。

其他主題

- [我可以利用 AWS Cloud9 做什麼？](#)
- [AWS Cloud9 的其他相關資訊](#)

我可以利用 AWS Cloud9 做什麼？

探索下列資源，以了解在一些常見情況下如何使用 AWS Cloud9。

主要案例

情況	資源
使用 AWS Toolkit 建立、執行與除錯 AWS Lambda 函數中的程式碼及無伺服器應用程式。	透過 AWS Toolkit 使用 AWS Lambda 函式
使用 Amazon Lightsail 實例預先配置了流行的應用程式和框架 WordPress，例如 LAMP (Linux, 阿帕奇, MySQL 和 PHP)，Node.js，尼金克斯，Drupal 和 Joomla，以及 Linux 發行版，如亞馬遜 Linux，Ubuntu，Debian，FreeBSD 和 openSUSE。	在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體
在 AWS CodeStar 的 AWS 軟體開發專案與工具鏈中使用程式碼。	在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CodeStar 專案
在 AWS CodePipeline 的持續交付解決方案中使用程式碼。	在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline
使用 AWS CLI 和 AWS CloudShell 自動化 AWS 服務。	AWS Command Line Interface 和 aws-shell 教程 AWS Cloud9
在 AWS CodeCommit 中使用來源碼儲存庫。	AWS CodeCommit 教學課程 AWS Cloud9
使用 Git 面板介面來處理中 GitHub 的原始程式碼儲存庫。	使用 Git 面板的視覺化來源控制
在 Amazon DynamoDB 使用 NoSQL 資料庫。	亞馬遜動態 B 教學 AWS Cloud9

情況	資源
使用 LAMP (Linux、Apache HTTP Server、MySQL 和 PHP) 堆疊。	燈泡教程 AWS Cloud9
使用 WordPress 網站。	WordPress教學課程AWS Cloud9
使用適用於 Java 和 AWS SDK for Java 的程式碼。	爪哇教學 AWS Cloud9
使用適用於 C++ 和 AWS SDK for C++ 的程式碼。	C ++ 教程 AWS Cloud9
使用適用於 Python 和 AWS SDK for Python (Boto) 的程式碼。	AWS Cloud9 的 Python 教學課程
使用適用於 .NET Core 和 AWS SDK for .NET 的程式碼。	.NET 教學課程AWS Cloud9
使用適用於 Node.js 和 AWS SDK for JavaScript 的程式碼。	Node.js 教學課程 AWS Cloud9
使用適用於 PHP 和 AWS SDK for PHP 的程式碼。	對於 PHP 的教程 AWS Cloud9
使用適用於 Ruby 和 AWS SDK for Ruby 的程式碼。	紅寶石AWS Cloud9
使用適用於 Go 和 AWS SDK for Go 的程式碼。	前往的教學課程 AWS Cloud9
使用 TypeScript 和的程式碼AWS SDK for JavaScript。	TypeScript教學課程AWS Cloud9
使用適用於 AWS Cloud Development Kit (AWS CDK) 的程式碼。	AWS CDK教學課程AWS Cloud9
在執行中 Docker 容器使用程式碼。	碼頭教程 AWS Cloud9
邀請其他人與您即時透過文字聊天支援功能一同使用環境。	使用 AWS Cloud9 的共用環境

情況	資源
使用中智慧型機器人應用程式的程式碼AWS RoboMaker。	在開發人 AWS RoboMaker 員指南 AWS Cloud9 中進行開發

AWS Cloud9 的其他相關資訊

本主題提供詳細資訊以協助您了解 AWS Cloud9。

主題

- [相關影片](#)
- [AWS 網站上的相關主題](#)
- [定價](#)
- [我有其他問題或需要協助](#)

相關影片

- [AWSRE：發明 2017 年-介紹AWS Cloud9：沃納·沃格爾斯主題演講 \(9 分鐘，網站 \)](#) YouTube
- [AWSRE：發明啟動板 2017 年-AWS Cloud9，\(15 分鐘，網站 \)](#) YouTube
- [介紹 AWS Cloud9-AWS 網上技術講座 \(33 分鐘，YouTube 網站 \)](#)
- [AWS悉尼峰會 2018：AWS Cloud9和 AWS CodeStar \(25 分鐘，YouTube 網站 \)](#)

AWS 網站上的相關主題

- [AWS Cloud9 簡介](#)
- [AWS Cloud9 – Cloud Developer 環境](#)
- [AWS Cloud9 概觀](#)
- [AWS Cloud9 功能](#)
- [AWS Cloud9 常見問答集](#)

定價

AWS Cloud9 無須額外收費。如果您為 AWS Cloud9 開發環境使用 Amazon EC2 執行個體，您只需為用於執行及存放程式碼的運算與儲存資源 (例如 Amazon EC2 執行個體、Amazon EBS 磁碟區) 付費。您也可以透過 SSH，將環境連接至現有的 Linux 伺服器 (例如內部部署伺服器)，無須另外付費。

您只需按實際用量付費；既沒有最低費用，也無須預付款項。您需為在環境中建立或使用的任何 AWS 資源 (例如，AWS Lambda 函數) 支付正常 AWS 費率。

符合 AWS 免費方案資格的 AWS 新客戶，可以免費使用 AWS Cloud9。如果環境超出 AWS 免費方案的資源使用量，您需為這些資源支付正常 AWS 費率。

如需詳細資訊，請參閱下列內容。

- AWS Cloud9 定價：請參閱 [AWS Cloud9 定價](#)。
- AWS 服務定價：請參閱 [Amazon EC2 定價](#)、[Amazon EBS 定價](#)、[AWS Lambda 定價](#)，以及 [AWS 定價](#)。
- AWS 免費方案：請參閱 AWS Billing and Cost Management 使用者指南中的 [使用 AWS 免費方案](#) 和 [追蹤您的免費方案使用量](#)。
- 教育定價：請參閱 [AWS Educate](#) 計劃。

我有其他問題或需要協助

若要提問或尋求 AWS Cloud9 社群的協助，請參閱 [AWS Cloud9 開發論壇](#)。(當您進入此論壇時，AWS 可能會要求您登入。)

另請參閱 [常見問答集](#) (FAQ)，或 [直接聯絡我們](#)。

設定 AWS Cloud9

若要開始使用 AWS Cloud9，請根據您計劃使用 AWS Cloud9 的方式，依照下列其中一組程序進行。

使用模式	遵循下列程序
我是使用 AWS 帳戶的唯一個別使用者，我不是學生。	個別使用者設定
我屬於在單一 AWS 帳戶內具有多個使用者的小組。	小組設定
我屬於在單一組織中具有一或多個 AWS 帳戶的企業。	企業設定

如需關於 AWS Cloud9 的一般資訊，請參閱[什麼是 AWS Cloud9？](#)。

主題

- [AWS Cloud9 的個別使用者設定](#)
- [AWS Cloud9 的小組設定](#)
- [AWS Cloud9 的企業設定](#)
- [AWS Cloud9 的其他設定選項 \(小組和企業\)](#)

AWS Cloud9 的個別使用者設定

本主題說明如何在您不是學生 AWS 帳戶時設定和使用 AWS Cloud9 為唯一的使用者。您可以設定 AWS Cloud9 任何其他使用模式。如需說明，請參閱[設定 AWS Cloud9](#)。

要 AWS Cloud9 作為您的唯一用戶使用 AWS 帳戶，請註冊 (AWS 帳戶如果您還沒有)。接著，登入 AWS Cloud9 主控台。

主題

- [註冊 AWS 帳戶](#)
- [建立管理使用者](#)

- [其他驗證方式](#)
- [後續步驟](#)

註冊 AWS 帳戶

如果您還沒有 AWS 帳戶，請完成以下步驟建立新帳戶。

註冊 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

註冊 AWS 帳戶時，會建立 AWS 帳戶根使用者。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

註冊程序完成後，AWS 會傳送一封確認電子郵件給您。您可以隨時登錄 <https://aws.amazon.com/> 並選擇 [我的帳戶](#)，以檢視您目前的帳戶活動並管理帳戶。

建立管理使用者

當您註冊 AWS 帳戶之後，請保護您的 AWS 帳戶根使用者，啟用 AWS IAM Identity Center，並建立管理使用者，讓您可以不使用根使用者處理日常作業。

保護您的 AWS 帳戶根使用者

1. 選擇 [根使用者](#) 並輸入您的 AWS 帳戶電子郵件地址，以帳戶擁有者身分登入 [AWS Management Console](#)。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

建立管理使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，將管理權限授予管理使用者。

若要取得有關使用 IAM Identity Center 目錄做為身分識別來源的教學課程，請參閱《使用 AWS IAM Identity Center 使用者指南中的[以預設 IAM Identity Center 目錄設定使用者存取權限](#)。

以管理員的身分登入

- 若要使用您的 IAM 身分中心使用者登入，請使用建立 IAM 身分中心使用者時傳送至您電子郵件地址的登入 URL。

如需有關如何使用 IAM Identity Center 使用者登入的說明，請參閱《AWS 登入 使用者指南》中的[登入 AWS存取入口網站](#)。

其他驗證方式

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

管理存取權限 AWS 帳戶

作為安全性最佳實務，我們建議您AWS Organizations搭配 IAM 身分中心管理所有人的存取AWS 帳戶。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的安全性最佳實務](#)。

您可以在身分識別中心建立使用者、使用 Microsoft 作用中目錄、使用 SAML 2.0 身分識別提供者 (IdP)，或個別將您的 IdP 聯合至。AWS 帳戶使用其中一種方法，您可以為使用者提供單一登入體驗。您也可以強制執行多重要素驗證 (MFA)，並使用臨時登入資料進行存取AWS 帳戶。這與 IAM 使用者不同，IAM 使用者是可共用的長期登入資料，可能會增加AWS資源的安全風險。

僅針對沙箱環境建立 IAM 使用者

如果您不熟悉AWS，可以建立測試 IAM 使用者，然後使用它來執行教學課程並探索可提供的內容AWS。學習時可以使用這種類型的認證，但我們建議您避免在沙箱環境之外使用它。

對於以下使用案例，在中開始使用 IAM 使用者可能很有意義AWS：

- 開始使用 AWS SDK 或工具，並在沙箱環境AWS 服務中進行探索。
- 執行排程的指令碼、工作和其他不支援人工登入程序的自動化程序，做為學習的一部分。

如果您在這些使用案例以外使用 IAM 使用者，請盡快轉換至 IAM 身分中心或聯合身分提供者。AWS 帳戶如需詳細資訊，請參閱[關中的AWS](#)。

安全的 IAM 使用者存取金鑰

您應該定期輪換 IAM 使用者存取金鑰。請遵循 IAM 使用者指南中[輪換存取金鑰](#)中的指引。如果您認為自己不小心共用了 IAM 使用者存取金鑰，請輪換存取金鑰。

IAM 使用者存取金鑰應儲存在本機電腦的共用AWScredentials檔案中。請勿在程式碼中儲存 IAM 使用者存取金鑰。請勿在任何原始程式碼管理軟體中包含 IAM 使用者存取金鑰的設定檔。外部工具（例如開源項目 [git-secrets](#)）可以幫助您無意中將敏感信息提交到 Git 存儲庫。如需詳細資訊，請參閱[IAM 使用者指南中的 IAM 身分 \(使用者、使用者群組和角色\)](#)。

後續步驟

學習任務	主題
了解如何使用 AWS Cloud9 IDE。	入門：基本教學課程 和 使用 IDE
更進階的工作	主題
建立 AWS Cloud9 開發環境，然後使用 AWS Cloud9 IDE 在您的新環境中使用程式碼。	建立環境
邀請其他人與您即時透過文字聊天支援功能一起使用您的新環境。	使用共享環境

AWS Cloud9 的小組設定

此主題說明如何使用 [AWS IAM Identity Center](#)，讓單一 AWS 帳戶內中的多個使用者可以使用 AWS Cloud9。若要設定將 AWS Cloud9 用於任何其他使用模式，請參閱 [設定 AWS Cloud9](#) 以取得正確的指示。

這些指示假設您具有或將有單一 AWS 帳戶的管理存取權。如需詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 帳戶根使用者](#) 和 [建立您的第一個管理員和群組](#)。如果您已有 AWS 帳戶，但沒有帳戶的管理存取權，請洽詢您的 AWS 帳戶管理員。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

Note

您可以使用 [IAM Identity Center](#) (而不是 IAM)，讓單一 AWS 帳戶內的多個使用者可以使用 AWS Cloud9。在這種使用模式中，單一 AWS 帳戶在 AWS Organizations 中充當組織的管理帳戶。此外，該組織沒有成員帳戶。若要使用 IAM Identity Center，請跳過這個主題，改按照 [企業設定](#) 中的指示操作。如需相關資訊，請參閱下列資源：

- AWS Organizations 使用者指南 (IAM Identity Center 需要使用 AWS Organizations) 中的 [AWS Organizations 是什麼](#)
- 《AWS IAM Identity Center 使用者指南》中的 [什麼是 AWS IAM Identity Center](#)
- 4 分鐘的視頻 [AWS 知識中心視頻：如何 AWS Organizations 開始](#) YouTube
- 7 分鐘的影片 [使用 IAM 身分中心管理使用者對多個 AWS 帳戶的存取](#) YouTube
- 9 分鐘的影片 [如何為您的內部部署 Active Directory 使用者設定身分識別中心](#) YouTube

若要讓單一 AWS 帳戶中的多個使用者可以開始使用 AWS Cloud9，請啟動針對您所具有之 AWS 資源的步驟。

您是否有 AWS 帳戶？	您在該帳戶中是否至少有一個 IAM 群組和使用者？	由此步驟開始
否	—	步驟 1：註冊 AWS 帳戶
是	否	步驟 2：建立 IAM 群組和使用者並將使用者新增至群組

您是否有 AWS 帳戶？	您在該帳戶中是否至少有一個 IAM 群組和使用者？	由此步驟開始
是	是	步驟 3：為群組新增 AWS Cloud9 存取許可

主題

- [註冊 AWS 帳戶](#)
- [建立管理使用者](#)
- [步驟 2：建立 IAM 群組和使用者並將使用者新增至群組](#)
- [步驟 3：為群組新增 AWS Cloud9 存取許可](#)
- [步驟 4：登入 AWS Cloud9 主控台](#)
- [後續步驟](#)

註冊 AWS 帳戶

如果您還沒有 AWS 帳戶，請完成以下步驟建立新帳戶。

註冊 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

註冊 AWS 帳戶時，會建立 AWS 帳戶根使用者。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為最佳安全實務，[將管理存取權指派給管理使用者](#)，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

註冊程序完成後，AWS 會傳送一封確認電子郵件給您。您可以隨時登錄 <https://aws.amazon.com/> 並選擇 [我的帳戶](#)，以檢視您目前的帳戶活動並管理帳戶。

建立管理使用者

在您註冊 AWS 帳戶之後，請保護您的 AWS 帳戶根使用者、啟用 AWS IAM Identity Center，以及建立管理使用者，讓您可以不使用根使用者處理日常作業。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入您的 AWS 帳戶電子郵件地址，以帳戶擁有者身分登入 [AWS Management Console](#)。在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

建立管理使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，將管理存取權授予管理使用者。

有關如何使用 IAM Identity Center 目錄作為身分來源的教學課程，請參閱《AWS IAM Identity Center 使用者指南》中的[以預設 IAM Identity Center 目錄設定使用者存取權](#)。

以管理員的身分登入

- 若要使用您的 IAM 身分中心使用者登入，請使用建立 IAM 身分中心使用者時傳送至您電子郵件地址的登入 URL。

如需有關如何使用 IAM Identity Center 使用者登入的說明，請參閱《AWS 登入使用者指南》中的[登入 AWS 存取入口網站](#)。

步驟 2：建立 IAM 群組和使用者並將使用者新增至群組

在此步驟中，您將於 AWS Identity and Access Management (IAM) 建立群組和使用者，將使用者新增至群組，然後使用該使用者來存取 AWS Cloud9。這是 AWS 安全最佳實務。如需詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 IAM 最佳實務。

如果已有您需要的所有 IAM 群組和使用者，請跳至 [步驟 3：為群組新增 AWS Cloud9 存取許可](#)。

Note

您的組織可能已為您設定了 IAM 群組和使用者。若您的組織內有 AWS 帳戶管理員，開始進行以下程序前請先向該名管理員確認。

您可以使用 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#) 完成這些任務。

若要觀看與下列主控台程序相關的 9 分鐘影片，請參閱 [如何在上設定 IAM 使用者並登入 AWS Management Console 使用 IAM 登入資料](#) YouTube。

步驟 2.1：使用主控台建立 IAM 群組

1. 前往 <https://console.aws.amazon.com/codecommit> 登入 AWS Management Console (若您尚未登入)。

Note

您可以使用建立 AWS 帳戶時所提供的電子郵件地址和密碼登入 AWS Management Console。這稱為以根使用者身分登入。不過，這不是 AWS 安全最佳實務。日後，建議您在 AWS 帳戶中使用管理員使用者的憑證來登入。管理員使用者具備與 AWS 帳戶根使用者類似的 AWS 存取許可，能夠避免某些相關安全風險。若您無法以管理員使用者身分登入，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱《IAM 使用者指南》中的 [建立您的第一個 IAM 使用者和群組](#)。

2. 開啟 IAM 主控台。方法如下：從 AWS 導覽列選擇 Services (服務)。然後選擇 IAM。
3. 從 IAM 主控台的導覽窗格中選擇 Groups (群組)。
4. 選擇 Create New Group (建立新群組)。
5. 在 Set Group Name (設定群組名稱) 頁面的 Group Name (群組名稱) 中，輸入新群組的名稱。
6. 選擇 Next Step (後續步驟)。
7. 在 Attach Policy (附加政策) 頁面上選擇 Next Step (下一個步驟)，無須附加任何政策 您將在 [步驟 3：將 AWS Cloud9 存取許可新增至群組](#) 中附加政策。
8. 選擇 Create Group (建立群組)。

Note

我們建議您重複這個程序來建立至少兩個個群組：一個群組用於 AWS Cloud9 使用者，另一個群組則用於 AWS Cloud9 管理員。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

跳至 [步驟 2.2：使用主控台建立 IAM 使用者並將使用者新增至群組](#)。

步驟 2.1：使用 AWS CLI 建立 IAM 群組

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

1. 在您的電腦上安裝及設定 AWS CLI (如果您尚未這樣做)。若要執行此作業，請參閱以下 AWS Command Line Interface 使用者指南中的說明：

- [安裝 AWS Command Line Interface](#)
- [快速組態](#)

Note

您可以使用與建立 AWS 帳戶時所提供之電子郵件地址和密碼相關聯的憑證來設定 AWS CLI。這稱為以根使用者身分登入。不過，這不是 AWS 安全最佳實務。反之，建議您使用 AWS 帳戶中 IAM 管理員使用者的憑證來設定 AWS CLI。IAM 管理員使用者具備與 AWS 帳戶根使用者類似的 AWS 存取許可，能夠避免某些相關安全風險。若您無法以 IAM 管理員使用者身分設定 AWS CLI，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱 [IAM 使用者指南](#) 中的建立您的第一個 IAM 管理員使用者和群組。

2. 執行 IAM `create-group` 命令，指定新群組的名稱 (例如 `MyCloud9Group`)。

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

Note

我們建議您重複這個程序來建立至少兩個個群組：一個群組用於 AWS Cloud9 使用者，另一個群組則用於 AWS Cloud9 管理員。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

跳至 [步驟 2.2：使用 AWS CLI 建立 IAM 使用者並將使用者新增至群組](#)。

步驟 2.2：使用主控台建立 IAM 使用者並將使用者新增至群組

1. 在之前的程序開啟的 IAM 主控台中，從導覽窗格中選擇 Users (使用者)。
2. 選擇新增使用者。
3. 對於 User name (使用者名稱)，輸入新使用者的名稱。

Note

選擇 Add another user (新增另一個使用者) 可同時建立多名使用者。本程序進行的其他設定會套用到當中的每一名新使用者。

4. 選取 Programmatic access (程式設計存取權) 和 AWS Management Console access (主控台存取權) 核取方塊。這可讓新使用者使用各種 AWS 開發人員工具和服務主控台。
5. 保留預設選擇 Autogenerated password (自動產生的密碼)。這會建立隨機密碼讓新的使用者登入主控台。或者，選擇 Custom password (自訂密碼)，並為新使用者輸入特定密碼。
6. 保留預設選擇 Require password reset (需要密碼重設)。這會在新使用者第一次登入主控台之後提示他們變更密碼。
7. 選擇 Next: Permissions (下一步：許可)。
8. 保留預設選項 Add user to group (新增使用者至群組)，如有多名使用者則為 Add users to group (新增各使用者至群組)。
9. 在群組清單中，選取您要新增使用者的該群組旁的核取方塊 (而非名稱)。
10. 選擇 Next: Review (下一步：檢閱)。
11. 選擇 Create user (建立使用者)。或者，若有多個使用者，選擇 Create users (建立使用者)。
12. 在精靈的最後一頁，執行下列其中一項：

- 在每一個新使用者旁，選擇 Send email (傳送電子郵件)，並依照螢幕上的指示，透過電子郵件將主控台登入 URL 和使用者名稱傳送給新使用者。接著，分別向每一個新使用者告知其主控台登入密碼、AWS 存取金鑰 ID 及 AWS 私密存取金鑰。
- 選擇 Download .csv (下載 .csv)。接著，向每一個新使用者告知其主控台登入 URL、主控台登入密碼、AWS 存取金鑰 ID 及 AWS 私密存取金鑰 (位於下載的檔案中)。
- 選擇每一名新使用者旁的 Show (顯示) 以查看 Secret access key (私密存取金鑰) 及 Password (密碼) 資訊。接著，向每一個新使用者告知其主控台登入 URL、主控台登入密碼、AWS 存取金鑰 ID 及 AWS 私密存取金鑰。

Note

如果未選擇 Download .csv (下載 .csv)，您就只有這麼一次機會可以查看新使用者的 AWS 私密存取金鑰與主控台登入密碼。若要為新使用者產生新的 AWS 私密存取金鑰或主控台登入密碼，請參閱以下 IAM 使用者指南中的章節。

- [建立、修改和檢視存取金鑰 \(主控台\)](#)
- [建立、變更或刪除 IAM 使用者密碼 \(主控台\)](#)

13 為您要建立的每個其他每個 IAM 使用者重複操作這個程序，然後跳到 [步驟 3：為群組新增 AWS Cloud9 存取許可](#)。

步驟 2.2：使用 AWS CLI 建立 IAM 使用者並將使用者新增至群組

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

1. 執行 IAM create-user 命令，指定新使用者的名稱 (例如 MyCloud9User) 以建立使用者。

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

2. 執行 IAM create-login-profile 命令，指定使用者的名稱及初始登入密碼 (例如 MyC10ud9Us3r!)，為使用者建立新的主控台登入密碼。使用者登入後，AWS 將要求使用者變更其登入密碼。

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

如果您稍後需要為使用者產生替代主控台登入密碼，請參閱 [《IAM 使用者指南》](#) 中的 [建立、變更或刪除 IAM 使用者密碼 \(API、CLI PowerShell\)](#)。

3. 執行 IAM `create-access-key` 命令，為使用者建立新的 AWS 存取金鑰及對應的 AWS 私密存取金鑰。

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

記下畫面上顯示的 `AccessKeyId` 和 `SecretAccessKey` 值。在您執行 IAM `create-access-key` 命令後，就只有這麼一次機會可以查看使用者的 AWS 私密存取金鑰。如果您稍後需要為使用者產生新的 AWS 秘密存取金鑰，請參閱 IAM 使用者指南中的 [建立、修改和檢視存取金鑰 \(API、CLI, PowerShell\)](#)。

4. 執行 IAM `add-user-to-group` 命令，指定群組的名稱與使用者的名稱，將使用者新增至群組。

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

5. 向使用者告知其主控台登入 URL、主控台初始登入密碼、AWS 存取金鑰 ID 及 AWS 私密存取金鑰。
6. 為您要建立的其他每個 IAM 使用者重複操作這個程序。

步驟 3：為群組新增 AWS Cloud9 存取許可

預設情況下，大多數 IAM 群組和使用者無法存取任何群組和使用者 AWS 服務 AWS Cloud9，包括（例外是 IAM 管理員群組和 IAM 管理員使用者，預設情況下可存取所有 AWS 服務群組和 IAM 管理員使用者）。在此步驟中，您會使用 IAM，直接將 AWS Cloud9 存取許可新增至一或多個使用者所屬的 IAM 群組新增。如此，您就可以確保這些使用者可以存取 AWS Cloud9。

Note

您的組織可能已為您設定了具備適當存取許可的群組。若您的組織內有 AWS 帳戶管理員，開始進行以下程序前請先向該名管理員確認。

您可以使用 [AWS Management Console](#) 主控台或 [AWS CLI](#) 執行這項任務。

使用主控台為群組新增 AWS Cloud9 存取許可

1. 前往 <https://console.aws.amazon.com/codecommit> 登入 AWS Management Console (若您尚未登入)。

Note

您可以使用建立 AWS 帳戶時所提供的電子郵件地址和密碼登入 AWS Management Console。這稱為以根使用者身分登入。不過，這不是 AWS 安全最佳實務。日後，建議您使用 AWS 帳戶中 IAM 管理員使用者的憑證來登入。管理員使用者具備與 AWS 帳戶根使用者類似的 AWS 存取許可，能夠避免某些相關安全風險。若您無法以管理員使用者身分登入，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱 [IAM 使用者指南](#) 中的建立您的第一個 IAM 管理員使用者和群組。

2. 開啟 IAM 主控台。方法如下：從 AWS 導覽列選擇 Services (服務)。然後，選擇 IAM。
3. 選擇 Groups (群組)。
4. 選擇群組的名稱。
5. 決定要為該群組新增 AWS Cloud9 使用者存取許可還是 AWS Cloud9 管理員存取許可。這些許可會套用到群組中的每個使用者。

AWS Cloud9 使用者存取許可允許群組中的每個使用者在其 AWS 帳戶內執行以下操作：

- 建立自己的 AWS Cloud9 開發環境。
- 取得自己環境的相關資訊。
- 變更自有環境的設定。

AWS Cloud9 管理員存取許可允許群組中的每個使用者在其 AWS 帳戶內執行額外操作：

- 為自己或其他人建立環境。
- 取得自己或其他人的環境相關資訊。
- 刪除自己或其他人的環境。
- 變更自己或其他人的環境設定。

Note

建議您僅新增限定人數的使用者至 AWS Cloud9 管理員群組。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

6. 在 Permissions (許可) 標籤上，針對 Managed Policies (受管政策)，選擇 Attach Policy (連接政策)。
7. 在策略名稱清單中，選擇AWS Cloud9使用者存取權限或AWS Cloud9管理員存取權限旁邊AWSCloud9Administrator的核取方塊。AWSCloud9User如果您在清單中未看到上述任一政策名稱，請在 Filter (篩選條件) 方塊中輸入政策名稱來顯示它。
8. 選擇 Attach Policy (連接政策)。

Note

如果您有多個群組要新增 AWS Cloud9存取權限，請對這些群組皆重複此程序。

如要查看上述 AWS 受管政策授予群組的存取許可清單，請參閱 [AWS 受管 \(預先定義\) 政策](#)。

除了 AWS Cloud9 必需的政策之外，您可以參閱 IAM 使用者指南中的[受管政策與內嵌政策](#)和[了解政策授與的許可](#)，以便了解其他可新增至群組的 AWS 存取許可。

跳至[步驟 4：登入 AWS Cloud9 主控台](#)。

使用 AWS CLI 將 AWS Cloud9 存取許可加入群組

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

1. 在您的電腦上安裝及設定 AWS CLI (如果您尚未這樣做)。若要執行此作業，請參閱以下 AWS Command Line Interface 使用者指南中的說明：
 - [安裝 AWS Command Line Interface](#)
 - [快速組態](#)

Note

您可以使用與建立 AWS 帳戶時所提供之電子郵件地址和密碼相關聯的憑證來設定 AWS CLI。這稱為以根使用者身分登入。不過，這不是 AWS 安全最佳實務。反之，建議您在 AWS 帳戶使用 IAM 管理員使用者的憑證來設定 AWS CLI。IAM 管理員使用者具備與 AWS

帳戶根使用者類似的 AWS 存取許可，能夠避免某些相關安全風險。若您無法以管理員使用者身分設定 AWS CLI，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱 IAM 使用者指南中的[建立您的第一個 IAM 管理員使用者和群組](#)。

2. 決定要為該群組新增 AWS Cloud9 使用者存取許可，還是 AWS Cloud9 管理員存取許可。這些許可會套用到群組中的每個使用者。

AWS Cloud9 使用者存取許可允許群組中的每個使用者在其 AWS 帳戶內執行以下操作：

- 建立自己的 AWS Cloud9 開發環境。
- 取得自己環境的相關資訊。
- 變更自有環境的設定。

AWS Cloud9 管理員存取許可允許群組中的每個使用者在其 AWS 帳戶內執行額外操作：

- 為自己或其他人建立環境。
- 取得自己或其他人的環境相關資訊。
- 刪除自己或其他人的環境。
- 變更自己或其他人的環境設定。

Note

建議您僅新增限定人數的使用者至 AWS Cloud9 管理員群組。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

3. 執行 IAM `attach-group-policy` 命令，指定群組的名稱以及所要新增的 AWS Cloud9 存取許可政策的 Amazon Resource Name (ARN)。

對於 AWS Cloud9 使用者存取許可，指定以下 ARN。

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

對於 AWS Cloud9 管理員存取許可，指定以下 ARN。

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

Note

如果您有多個群組要新增 AWS Cloud9 存取權限，請對這些群組皆重複此程序。

如要查看上述 AWS 受管政策授予群組的存取許可清單，請參閱 [AWS 受管 \(預先定義\) 政策](#)。

除了 AWS Cloud9 必需的政策之外，您可以參閱 IAM 使用者指南中的 [受管政策與內嵌政策](#) 和 [了解政策授與的許可](#)，以便了解其他可新增至群組的 AWS 存取許可。

步驟 4：登入 AWS Cloud9 主控台

完成本主題的上述步驟後，您和您的使用者就可以登入 AWS Cloud9 主控台。

1. 如果您已以 AWS 帳戶根使用者的身分登入 AWS Management Console，請登出主控台。
2. 開啟位於 <https://console.aws.amazon.com/cloud9/> 的 AWS Cloud9 主控台。
3. 輸入您先前所建立或識別之 IAM 使用者的 AWS 帳戶號碼，然後選擇 Next (下一步)。

Note

如果未看到輸入 AWS 帳號的選項，請選擇 Sign in to a different account (登入另一帳戶)。在下一頁輸入 AWS 帳戶號碼，然後選擇 Next (下一步)。

4. 輸入您先前所建立或識別之 IAM 使用者的登入憑證，然後選擇 Sign In (登入)。
5. 出現提示時，依照螢幕上的指示變更您的使用者初始登入密碼。將新的登入密碼存放到安全的位置。

AWS Cloud9 主控台隨即顯示，而您可以開始使用 AWS Cloud9。

後續步驟

任務	參閱此主題
限制 AWS 帳戶中其他人的 AWS Cloud9 用量，以控制成本。	其他設定選項

任務	參閱此主題
建立 AWS Cloud9 開發環境，然後使用 AWS Cloud9 IDE 在您的新環境中使用程式碼。	建立環境
了解如何使用 AWS Cloud9 IDE。	入門：基本教學課程 和 使用 IDE
邀請其他人與您即時透過文字聊天支援功能一起使用您的新環境。	使用共用環境

AWS Cloud9 的企業設定

此主題說明如何使用 [AWS IAM Identity Center](#)，讓一或多個 AWS 帳戶可以在企業內使用 AWS Cloud9。若要設定將 AWS Cloud9 用於任何其他使用模式，請參閱 [設定 AWS Cloud9](#) 以取得正確的指示。

Warning

為避免安全風險，在開發專用軟體或使用真實資料時，請勿使用 IAM 使用者進行身分驗證。相反地，搭配使用聯合功能和身分提供者，例如 [AWS IAM Identity Center](#)。

這些指示假設您在 AWS Organizations 中具有或將具有組織的管理存取權。如果您在 AWS Organizations 中還沒有組織的管理存取權，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱下列資源：

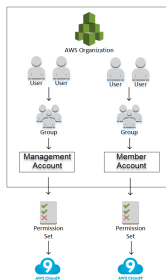
- 在 [AWS 組織使用者指南](#) 中 (IAM Identity Center 需要使用 AWS Organizations)，為您的 AWS 組織管理存取許可
- AWS IAM Identity Center 使用者指南中的 [管理 IAM Identity Center 資源存取許可概觀](#)
- [使用 AWS Control Tower](#)，這是一項您可以用來設定和控管 AWS 多帳戶環境的服務。AWS Control Tower 配合其他 AWS 服務的功能 (包括 AWS Organizations、AWS Service Catalog 和 AWS IAM Identity Center)，在不到一小時的時間內建置登陸區域。

如需本主題相關的簡介資訊，請參閱下列資源：

- AWS Organizations 使用者指南 (IAM Identity Center 需要使用 AWS Organizations) 中的 [AWS Organizations 是什麼](#)

- 《AWS IAM Identity Center 使用者指南》中的[什麼是 AWS IAM Identity Center](#)
- AWS Control Tower 使用者指南中的[開始使用 AWS Control Tower](#)
- [如何開 OrganAWS izations 使用AWSYouTube](#)
- 使用[管理多個AWS帳戶的使用AWS IAM Identity Center者存取許可YouTube](#)
- [現場部署 Active Directory 使用者設定AWS單一登入YouTube](#)

以下概念圖顯示您最後得到的結果。



若要讓一或多個 AWS 帳戶可在企業內開始使用 AWS Cloud9，請根據您已有的 AWS 資源遵循步驟。

您在 AWS Organizations 中是否有 AWS 帳戶可以或已經做為組織的管理帳戶？	您在 AWS Organizations 中是否已有所屬該管理帳戶的組織？	所有所需 AWS 帳戶都是該組織的成員嗎？	該組織是否設定為使用 IAM Identity Center？	該組織是否已設定所有想要使用 AWS Cloud9 的必要群組和使用者？	由此步驟開始
否	—	—	—	—	步驟 1：建立組織的管理帳戶
是	否	—	—	—	步驟 2：為管理帳戶建立組織
是	是	否	—	—	步驟 3：將成員帳戶新增至組織

您在 AWS Organizations 中是否有 AWS 帳戶可以或已經做為組織的管理帳戶？	您在 AWS Organizations 中是否已有所屬該管理帳戶的組織？	所有所需 AWS 帳戶都是該組織的成員嗎？	該組織是否設定為使用 IAM Identity Center？	該組織是否已設定所有想要使用 AWS Cloud9 的必要群組和使用者？	由此步驟開始
是	是	是	否	—	步驟 4：在整個組織啟用 IAM Identity Center
是	是	是	是	否	步驟 5. 在組織內設定群組和使用者
是	是	是	是	是	步驟 6：讓組織內的群組和使用者使用 AWS Cloud9

步驟 1：建立組織的管理帳戶

Note

您的企業可能已為您設定管理帳戶。如果您的企業有 AWS 帳戶管理員，在開始進行以下程序前，請先洽詢該管理員。如果您已經有管理帳戶，請跳到[步驟 2：為管理帳戶建立組織](#)。

若要使用 AWS IAM Identity Center (IAM Identity Center)，您必須具有 AWS 帳戶。您的 AWS 帳戶充當 AWS Organizations 中組織的管理帳戶。如需詳細資訊，請參閱 AWS Organizations 使用者指南中 [AWS Organizations 詞彙與概念](#) 的管理帳戶相關討論。

若要觀看與以下程序相關的 4 分鐘示範影片，請參閱上的 [建立 Amazon Web Services 帳戶](#) YouTube。

如何建立管理帳戶：

1. 轉到 <https://aws.amazon.com/>。
2. 選擇 Sign In to the Console (登入主控台)。
3. 選擇 Create a new (建立新的) AWS 帳戶。
4. 依照螢幕上的指示完成整個流程，步驟包括向 AWS 提供您的電子郵件地址和信用卡資訊。您還必須使用手機輸入 AWS 傳給您的驗證碼。

帳戶建立完成後，AWS 會傳送確認電子郵件給您。未收到該項確認前切勿前往下一個步驟。

步驟 2：為管理帳戶建立組織

Note

您的企業可能已設定 AWS Organizations 來使用管理帳戶。如果您的企業有 AWS 帳戶管理員，在開始進行以下程序前，請先洽詢該管理員。如果您已設定 AWS Organizations 來使用管理帳戶，請跳到 [步驟 3：將成員帳戶新增至組織](#)。

若要使用 IAM Identity Center，您在 AWS Organizations 中必須有使用管理帳戶的組織。如需詳細資訊，請參閱 AWS Organizations 使用者指南中 [AWS Organizations 詞彙與概念](#) 的組織相關討論。

若要在 AWS Organizations 中為管理 AWS 帳戶建立組織，請按照《AWS Organizations 使用者指南》中的這些指示進行：

1. [建立組織](#)
2. [啟用組織中的所有功能](#)

若要觀看與這些程序相關的 4 分鐘示範影片 [知識中心影片 AWS 知識中心影片 知識中心影片 知識中心影片 知識中心影片 OrganAWS izations 中心影片 YouTube](#)

步驟 3：將成員帳戶新增至組織

Note

您的企業可能已在 AWS Organizations 中設定需要的成員帳戶。如果您的企業有 AWS 帳戶管理員，在開始進行以下程序前，請先洽詢該管理員。如果您已在 AWS Organizations 中設定需要的成員帳戶，請跳到 [步驟 4：在整個組織啟用 IAM Identity Center](#)。

在此步驟中，您會在 AWS Organizations 中新增任何 AWS 帳戶來充當組織的成員帳戶。如需詳細資訊，請參閱 AWS Organizations 使用者指南中 [AWS Organizations 詞彙與概念](#) 的成員帳戶相關討論。

Note

您不需要將任何成員帳戶新增至組織。您在組織中只需要單一管理帳戶就可以使用 IAM Identity Center。之後，需要的話，您可以將成員帳戶新增至組織。如果您現在不想新增任何成員帳戶，請跳到 [步驟 4：在整個組織啟用 IAM Identity Center](#)。

若要在 AWS Organizations 中將成員帳戶新增至組織，請按照以下一組或兩組 AWS Organizations 使用者指南中的指示操作。視需要不斷重複操作這些指示，直到您需要的所有 AWS 帳戶都成為組織的成員為止：

- [在您的組織中建立 AWS 帳戶](#)
- [邀請 AWS 帳戶 加入您的組織](#)

步驟 4：在整個組織啟用 IAM Identity Center

Note

您的企業可能已設定 AWS Organizations 來使用 IAM Identity Center。如果您的企業有 AWS 帳戶管理員，在開始進行以下程序前，請先洽詢該管理員。如果您已設定 AWS Organizations 來使用 IAM Identity Center，請跳到 [步驟 5. 在組織內設定群組和使用者](#)。

在此步驟中，您將讓組織在 AWS Organizations 中使用 IAM Identity Center。若要執行此作業，請遵循 AWS IAM Identity Center 使用者指南中的這些說明：

1. [IAM Identity Center 先決條件](#)
2. [啟用 IAM Identity Center](#)

步驟 5. 在組織內設定群組和使用者

Note

您的企業可能已在 AWS Organizations 中設定來自 IAM Identity Center 目錄，或來自 AWS Managed Microsoft AD 或 AD Connector 目錄 (在 AWS Directory Service 中管理) 的群組和使用者。如果您的企業有 AWS 帳戶管理員，在開始進行以下程序前，請先洽詢該管理員。如果您已在 AWS Organizations 中設定來自 IAM Identity Center 目錄 AWS Directory Service 的使用者和群組，請跳到[步驟 6. 讓組織內的群組和使用者使用 AWS Cloud9](#)。


在此步驟中，您會在 IAM Identity Center 目錄中為組織建立群組和使用者。或者，您會連線到 AWS Directory Service 中針對組織管理的 AWS Managed Microsoft AD 或 AD Connector 目錄。在後續步驟中，您將提供必要的存取許可給群組來使用 AWS Cloud9。

- 如果您為組織使用 IAM Identity Center 目錄，請依照 AWS IAM Identity Center 使用者指南中的指示進行。視需要不斷重複以下步驟，直到有您想要的所有群組和使用者為止：
 1. [新增群組](#)。建議您針對整個組織的任何 AWS Cloud9 管理員建立至少一個群組。然後，重複此步驟，針對整個組織的所有 AWS Cloud9 使用者建立另一個群組。(選用) 您還可以重複此步驟，針對整個組織的所有使用者建立第三個群組，而與他們共用現有的 AWS Cloud9 開發環境。但是，不允許他們自行建立環境。為了簡單使用，我們建議將這些群組分別命名為 AWSCloud9Administrators、AWSCloud9Users 和 AWSCloud9EnvironmentMembers。如需詳細資訊，請參閱 [AWS Cloud9 的 AWS 受管 \(預先定義\) 政策](#)。
 2. [新增使用者](#)。
 3. [將使用者新增至群組](#)。將任何 AWS Cloud9 管理員新增至 AWSCloud9Administrators 群組，然後重複此步驟，將 AWS Cloud9 使用者新增至 AWSCloud9Users 群組。也可以選擇性地重複此步驟，將任何剩餘的使用者新增至 AWSCloud9EnvironmentMembers 群組。將使用者新增至群組是 AWS 安全最佳實務，可協助您進一步控制、追蹤和排除 AWS 資源存取的問題。
- 如果您使用的是您在 AWS Directory Service 中針對組織管理的 AWS Managed Microsoft AD 或 AD Connector 目錄，請參閱《AWS IAM Identity Center 使用者指南》中的[連線至 Microsoft AD 目錄](#)。

步驟 6：讓組織內的群組和使用者使用 AWS Cloud9

根據預設，AWS Organizations 中組織的大多數使用者和群組無法存取任何 AWS 服務，包括 AWS Cloud9。在此步驟中，您將在 AWS Organizations 中使用 IAM Identity Center 來允許整個組織的群組和使用者，在參與帳戶的任意組合內使用 AWS Cloud9。

1. 在 [IAM Identity Center 主控台](#) 中，從服務導覽窗格中選擇 AWS 帳戶。
2. 選擇 Permission sets (許可集合) 標籤。
3. 選擇 Create permission set (建立許可集合) 設定。
4. 選取 Create a custom permission set (建立自訂許可集合)。
5. 輸入此許可集合的 Name (名稱)。建議您針對整個組織的任何 AWS Cloud9 管理員建立至少一個許可集合。然後，重複此程序中的步驟 3 到 10，針對組織中的所有 AWS Cloud9 使用者建立另一個許可集合。您還可以選擇性地重複此程序的步驟 3 到 10，針對整個組織的所有使用者建立第三個許可集合，而與他們共用現有的 AWS Cloud9 開發環境。但是，不允許他們自行建立環境。為了簡單使用，我們建議將這些許可集合分別命名為 AWSCloud9AdministratorsPerms、AWSCloud9UsersPerms 和 AWSCloud9EnvironmentMembersPerms。如需詳細資訊，請參閱 [AWS Cloud9 的 AWS 受管 \(預先定義\) 政策](#)。
6. 輸入許可集合的選用 Description (描述)。
7. 選擇許可集合的 Session duration (工作階段持續時間)，或保留預設的工作階段持續時間 1 hour (1 小時)。
8. 選取 Attach AWS managed policies (連接 AWS 受管政策)。
9. 在政策清單中，從正確的 Policy name (政策名稱) 項目旁選取以下其中一個方塊。(請勿直接選擇政策名稱。如果您在清單中看不到某個政策名稱，請在 Search (搜尋) 方塊中輸入政策名稱讓它顯示。)
 - 針對 AWSCloud9AdministratorsPerms 權限集，選取 AWSCloud9Administrator。
 - 針對 AWSCloud9UsersPerms 權限集，選取 AWSCloud9User。
 - 或者，針對 AWSCloud9EnvironmentMembersPerms 權限集，選取 AWSCloud9EnvironmentMember。


 Note

除了 AWS Cloud9 必需的政策之外，您可以參閱 IAM 使用者指南中的 [受管政策與內嵌政策](#) 和 [了解政策授與的許可](#)，以便了解其他可新增的政策。

10. 選擇 建立。
11. 當您所需的所有許可集合都完成建立時，請在 AWS organization (AWS 組織) 索引標籤上，選擇您要指派 AWS Cloud9 存取許可的 AWS 帳戶。如果看不到 AWS organization (AWS 組織) 索引標籤，請在服務導覽窗格中，選擇 AWS 帳戶。這會顯示 AWS organization (AWS 組織) 索引標籤。
12. 選擇 Assign users (指派使用者)。

13. 在 Groups (群組) 索引標籤中，選取您要指派 AWS Cloud9 存取許可的群組名稱旁的方塊。請勿選擇群組名稱本身。

- 如果您為組織使用 IAM Identity Center 目錄，您可能已為管理員建立名 AWS Cloud9 Administrators 為 AWS Cloud9 管理員的群組。
- 如果您使用的是您在 AWS Directory Service 中針對組織管理的 AWS Managed Microsoft AD 或 AD Connector 目錄，請選擇目錄的 ID。接著，輸入群組的部分或全部名稱，然後選擇 Search connected directory (搜索連線的目錄)。最後，選擇您要指派 AWS Cloud9 存取許可的群組名稱旁的方塊。

 Note

我們建議指派 AWS Cloud9 存取許可給群組，而不是個別使用者。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

14. 選擇 Next: Permission sets (下一步：許可集合)。

15. 選取您要指派給此群組的許可集合名稱旁的方塊 (例如，AWS Cloud9 Administrators Perms 為 AWS Cloud9 管理員選取)。請勿選擇許可集合名稱本身。

16. 選擇 Finish (完成)。

17. 選擇 Proceed to AWS 帳戶 (繼續前往 AWS 帳戶)。

18. 如果您還要為整個組織指派任何其他 AWS Cloud9 存取許可給 AWS 帳戶，請重複此程序的步驟 11 到 17。

步驟 3：開始使用 AWS Cloud9

完成本主題的上述步驟後，您和您的使用者就可以登入 IAM Identity Center 並開始使用 AWS Cloud9。

1. 如果您已登入 AWS 帳戶或 IAM Identity Center，請登出。若要執行此操作，請參閱 AWS 支援網站中的 [如何登出我的 AWS 帳戶](#)，或 AWS IAM Identity Center 使用者指南中的 [如何登出使用者入口網站](#)。
2. 若要登入 IAM Identity Center，請遵循 AWS IAM Identity Center 使用者指南中的 [如何接受邀請加入 IAM Identity Center](#) 進行操作。這包括前往唯一的登入 URL，然後以唯一的登入憑證登入。您的 AWS 帳戶管理員將以電子郵件將此資訊寄給您，或透過其他方式提供給您。

Note

請務必將提供給您的唯一登入 URL 加上書籤。如此，您稍後就可以在輕鬆返回其中。此外，請務必將此 URL 的唯一登入憑證存放在安全位置。

這個由 URL、使用者名稱和密碼形成的組合也可能不同，視 AWS 帳戶管理員提供的不同層級 AWS Cloud9 存取許可而定。例如，您可能使用一個 URL、使用者名稱和密碼，以取得某個帳戶的 AWS Cloud9 管理員存取權。您可能使用不同的 URL、使用者名稱和密碼，只允許 AWS Cloud9 使用者存取不同的帳戶。

3. 在您登入 IAM Identity Center 之後，請選擇 AWS 帳戶 圖磚。
4. 從顯示的下拉式清單中，選擇使用者的顯示名稱。如果出現多個名稱，請選擇您要用來開始使用 AWS Cloud9 的名稱。如果您不確定選擇哪個名稱，請洽詢您的 AWS 帳戶管理員。
5. 選擇使用者的顯示名稱旁的 Management console (管理主控台) 連結。如果出現多個 Management console (管理主控台) 連結，請選擇正確許可集合旁的連結。如果您不確定選擇哪個連結，請洽詢您的 AWS 帳戶管理員。
6. 從 AWS Management Console，執行以下其中一項：
 - 選擇 Cloud9 (如果已顯示)。
 - 展開 All services (所有服務)，然後選擇 Cloud9。
 - 在 Find services (尋找服務) 方塊中，輸入 Cloud9，然後按 Enter。
 - 在 AWS 導覽列中，選擇 Services (服務)，然後選擇 Cloud9。

AWS Cloud9 主控台隨即顯示，而您可以開始使用 AWS Cloud9。

後續步驟

任務	參閱此主題
建立 AWS Cloud9 開發環境，然後使用 AWS Cloud9 IDE 在您的新環境中使用程式碼。	建立環境
了解如何使用 AWS Cloud9 IDE。	入門：基本教學課程 和 使用 IDE
邀請其他人與您即時透過文字聊天支援功能一起使用您的新環境。	使用共用環境

AWS Cloud9 的其他設定選項 (小組和企業)

本主題假設您已完成[小組設定](#)或[企業設定](#)中的設定步驟。

在[小組設定](#)或[企業設定](#)中，您已建立群組，並已將 AWS Cloud9 存取許可直接新增至這些群組。這是為了確保這些群組中的使用者可以存取 AWS Cloud9。在本主題中，您會新增更多存取許可來限制這些群組中的使用者可以建立的環境類型。這麼做有助於控制 AWS 帳戶和組織中 AWS Cloud9 的相關成本。

若要新增這些存取許可，您需要建立自己的政策集合，藉此定義您要強制執行的 AWS 存取許可。我們將其中每個政策稱為客戶受管政策。然後，您會將這些客戶受管政策連接到使用者所屬的群組。在某些情況下，您也必須分離已連接到這些群組的現有 AWS 受管政策。若要進行這項設定，請依照本主題中的程序進行。

Note

下列程序只會說明如何連接及分離 AWS Cloud9 使用者的政策。這些程序假設您已有個別的 AWS Cloud9 使用者群組和 AWS Cloud9 管理員群組。它們還假設您在 AWS Cloud9 管理員群組中只有數量有限的使用者。此 AWS 安全最佳實務可協助您進一步控制、追蹤 AWS 資源存取問題並進行故障診斷。

- [步驟 1：建立客戶管理政策](#)
- [步驟 2：將客戶管理政策新增至群組](#)
- [使用 AWS Cloud9 的小組適用的客戶受管政策範例](#)

步驟 1：建立客戶管理政策

您可以使用 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#) 來建立客戶受管政策。

Note

本步驟只會說明如何建立 IAM 群組的客戶受管政策。若要在 AWS IAM Identity Center 為群組建立自訂許可，請略過此步驟，並遵循《AWS IAM Identity Center 使用者指南》中[建立許可集合](#)中的指示操作。在這個主題中，請依照說明來建立自訂許可集合。如需了解相關自訂許可政策，請參閱本主題稍後的[使用 AWS Cloud9 的小組適用的客戶受管政策範例](#)。

使用主控台建立客戶受管政策

1. 如果您尚未登入 AWS Management Console，請現在登入。

我們建議您在 AWS 帳戶使用管理員使用者的憑證來登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。

2. 開啟 IAM 主控台。方法如下：從主控台的導覽列選擇 Services (服務)，然後選擇 IAM。
3. 在服務的導覽窗格中，選擇 Policies (政策)。
4. 選擇 Create policy (建立政策)。
5. 在 JSON 標籤中，貼上其中一個建議的[客戶受管政策範例](#)。

Note

您也可以建立自己的客戶受管政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策參考](#)，以及 AWS 服務的[文件](#)。

6. 選擇 Review policy (檢閱政策)。
7. 在 Review policy (檢閱政策) 頁面上，輸入政策的 Name (名稱) 和選用的 Description (描述)，然後選擇 Create policy (建立政策)。

針對您要建立的每個額外客戶受管政策重複此步驟。然後，直接跳到[使用主控台將客戶受管政策新增至群組](#)。

使用 AWS CLI 建立客戶受管政策

1. 在您執行 AWS CLI 的電腦上，建立描述政策的檔案 (例如 policy.json)。

如果您使用不同的檔案名稱建立檔案，請在此程序中予以取代。

2. 將我們建議的其中一個[客戶受管政策範例](#)貼到 policy.json 檔案中。

Note

您也可以建立自己的客戶受管政策。如需詳細資訊，請參閱中的 IAM 使用者指南中的 [IAM JSON 政策參考](#) 以及 AWS 服務的[文件](#)。

3. 從終端機或命令提示，切換至包含 policy.json 檔案的目錄。
4. 執行 IAM create-policy 命令，並指定政策的名稱和 policy.json 檔案。

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

在上述命令中，將 MyPolicy 取代為政策的名稱。

直接跳到[使用 AWS CLI 將客戶受管政策新增至群組](#)。

步驟 2：將客戶受管政策新增至群組

您可以使用 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#)，將客戶受管政策新增至群組。

Note

這個步驟只會說明如何在 IAM 群組新增客戶受管政策。若要在 AWS IAM Identity Center 為群組新增自訂許可，請略過此步驟，並改按照 AWS IAM Identity Center 使用者指南中的[指派使用者存取權](#)操作。

使用主控台將客戶受管政策新增至群組

1. 使用上一個程序中開啟的 IAM 主控台，在服務的導覽窗格選擇 Groups (群組)。
2. 選擇群組的名稱。
3. 在 Permissions (許可) 標籤上，針對 Managed Policies (受管政策)，選擇 Attach Policy (連接政策)。
4. 在政策名稱清單中，選擇您要連接到群組的每個客戶受管政策旁的方塊。如果您看不到清單中的特定政策名稱，請在 Filter (篩選條件) 方塊中輸入政策名稱來顯示它。
5. 選擇 Attach Policy (連接政策)。

使用 AWS CLI 將客戶受管政策新增至群組

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

執行 IAM `attach-group-policy` 命令，並指定群組的名稱和政策的 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，將 `MyGroup` 取代為群組的名稱。將 `123456789012` 取代為 AWS 帳戶 ID。將 `MyPolicy` 取代為客戶受管政策的名稱。

使用 AWS Cloud9 的小組適用的客戶受管政策範例

以下幾個政策範例可供您用來限制群組中的使用者能夠在 AWS 帳戶中建立的環境。

- [防止群組中的使用者建立環境](#)
- [防止群組中的使用者建立 EC2 環境](#)
- [允許群組中的使用者建立只含有特定 Amazon EC2 執行個體類型的 EC2 環境](#)
- [允許群組中的使用者在每個 AWS 區域只建立單一 EC2 環境](#)

防止群組中的使用者建立環境

下列客戶受管政策連接到 AWS Cloud9 使用者群組時，會防止這些使用者在 AWS 帳戶中建立環境。如果您想要讓 AWS 帳戶中的管理員使用者管理環境的建立，這個政策會很有用。否則，AWS Cloud9 使用者群組中的使用者會執行此操作。

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

前述客戶受管政策會明確在已連接到 AWS Cloud9 使用者群組的 AWSCloud9User 受管政策中的 "Resource": "*", 將 "Effect": "Allow" 覆寫為 "Action": "cloud9:CreateEnvironmentEC2" 和 "cloud9:CreateEnvironmentSSH"。

防止群組中的使用者建立 EC2 環境

下列客戶受管政策連接到 AWS Cloud9 使用者群組時，會防止這些使用者在 AWS 帳戶中建立 EC2 環境。如果您想要讓 AWS 帳戶中的管理員使用者管理 EC2 環境的建立，這個政策會很有用。否則，AWS Cloud9 使用者群組中的使用者會執行此操作。此政策假設您並未同時連接防止該群組中使用者建立 SSH 環境的政策。否則，這些使用者無法建立環境。

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

前述客戶受管政策會明確在已連接到 AWS Cloud9 使用者群組的 AWSCloud9User 受管政策中的 "Resource": "*", 將 "Effect": "Allow" 覆寫為 "Action": "cloud9:CreateEnvironmentEC2"。

允許群組中的使用者建立只含有特定 Amazon EC2 執行個體類型的 EC2 環境

下列客戶受管政策連接到 AWS Cloud9 使用者群組時，允許使用者群組中的使用者在 AWS 帳戶中建立的 EC2 環境只使用以 t2 開頭的執行個體類型。此政策假設您並未同時連接防止該群組中的使用者建立 EC2 環境的政策。否則，這些使用者無法建立 EC2 環境。

您可以將下列政策中的 "t2.*" 取代為不同的執行個體類別 (例如 "m4.*")。或者，您可以限定其為多個執行個體類別或執行個體類型 (例如 ["t2.*", "m4.*"] 或 ["t2.micro", "m4.large"])。

若是 AWS Cloud9 使用者群組，請從群組分離 AWSCloud9User 受管政策。接著，在其位置新增下列客戶受管政策。如果您未分離 AWSCloud9User 受管政策，下列客戶受管政策會沒有作用。

```
{
  "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"
    }
  }
}
]
}

```

上述客戶受管政策也可讓這些使用者建立 SSH 環境。若要完全防止這些使用者建立 SSH 環境，請從上述客戶受管政策移除 "cloud9:CreateEnvironmentSSH",。

允許群組中的使用者在每個 AWS 區域只建立單一 EC2 環境

下列客戶受管政策連接到 AWS Cloud9 使用者群組時，可讓當中的每個使用者在每個可使用 AWS Cloud9 的 AWS 區域中最多建立一個 EC2 環境。其作法是將環境的名稱限制為該 AWS 區域中的某個特定名稱。在此範例中，環境限制為 my-demo-environment。

Note

AWS Cloud9 不會限制環境不得建立特定的 AWS 區域。AWS Cloud9 也不會限制可以建立的環境總數。唯一的例外是發佈的[服務限制](#)。

針對 AWS Cloud9 使用者群組，請從群組分離 AWSCloud9User 受管政策，然後新增下列客戶受管政策取而代之。如果您未分離 AWSCloud9User 受管政策，下列客戶受管政策沒有作用。

```

{
  "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"
    }
  }
}
```

```
    }  
  ]  
}
```

上述客戶受管政策可讓這些使用者建立 SSH 環境。若要完全防止這些使用者建立 SSH 環境，請從上述客戶受管政策移除 "cloud9:CreateEnvironmentSSH"。

如需更多範例，請參閱 [客戶受管政策範例](#)。

後續步驟

任務	參閱此主題
建立 AWS Cloud9 開發環境，然後使用 AWS Cloud9 IDE 在您的新環境中使用程式碼。	建立環境
了解如何使用 AWS Cloud9 IDE。	入門：基本教學課程 和 使用 IDE
邀請其他人與您即時透過文字聊天支援功能一起使用您的新環境。	使用共享環境

入門：AWS Cloud9 的基本教學課程

剛接觸 AWS Cloud9 嗎？如果您尚未接觸過，請參閱[什麼是 AWS Cloud9](#) 中的 AWS Cloud9。

在以下教學課程中，您會在 AWS Cloud9 中建立環境，然後使用該環境來建立簡易的應用程式。兩個教學課程具有相同的輸入和結果，但一個使用 AWS Cloud9 主控台，另一個則使用 [AWS Command Line Interface \(AWS CLI\)](#)。您可以選擇執行其中之一或兩者。

當您完成這些教學課程時，您可以在 [導覽 AWS Cloud9 IDE](#) 中進一步了解 AWS Cloud9 IDE。

主題

- [教學課程：Hello AWS Cloud9 \(主控台\)](#)
- [教學課程：Hello AWS Cloud9 \(CLI\)](#)

教學課程：Hello AWS Cloud9 (主控台)

本教學課程提供初次了解 AWS Cloud9。其中說明如何使用和導覽 AWS Cloud9 主控台。

在此教學課程中，您會設定 AWS Cloud9 開發環境然後使用 AWS Cloud9 IDE 對第一個應用程式進行編碼、執行和除錯。

此教學課程需約 1 小時方能完成。

Warning

完成此教學可能會向您的 AWS 區域收取費用。收費的項目可能包括 Amazon EC2。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

必要條件

若要成功完成本教學課程，您必須先完成 [設定 AWS Cloud9](#) 中的步驟。

步驟

- [步驟 1：建立環境](#)

- [步驟 2：IDE 基本導覽](#)
- [步驟 3：清理](#)
- [相關資訊](#)

步驟 1：建立環境

([教學課程：Hello AWS Cloud9 \(主控台\)](#) 的第一個步驟)

在此步驟中，您要使用 AWS Cloud9 主控台建立 AWS Cloud9 開發環境，然後將其開啟。

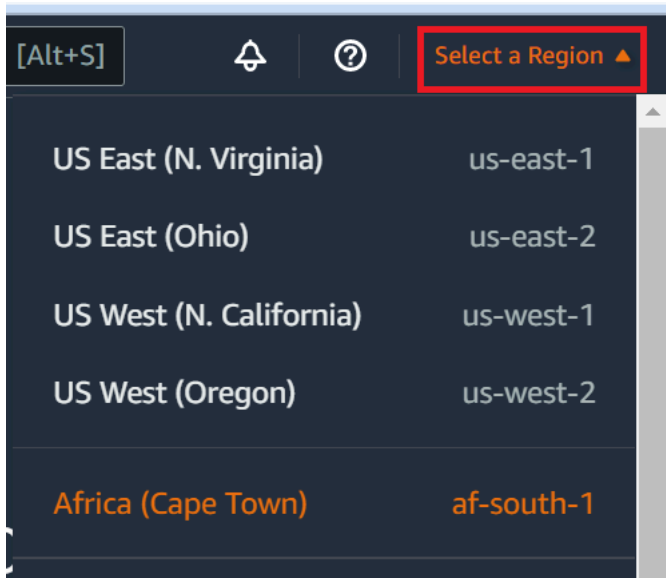
Note

如果您已建立要用於此教學課程的環境，請開啟該環境，並跳至[步驟 2：IDE 基本導覽](#)。

在 AWS Cloud9 中，development environment (開發環境) (或稱 environment (環境)) 是讓您存放開發專案檔案和執行各種工具來開發應用程式的位置。在此教學課程中，您會建立 EC2 環境，然後使用該環境中的檔案和工具。

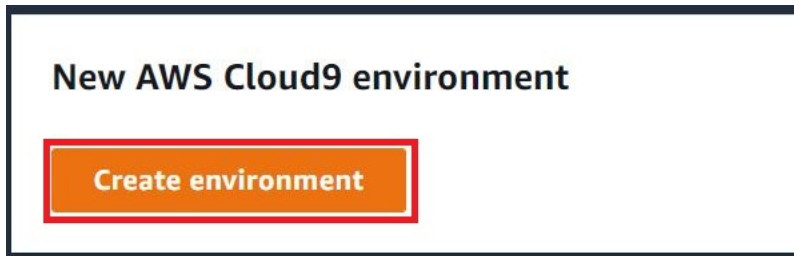
使用主控台建立 EC2 環境

1. 登入 AWS Cloud9 主控台。
 - 如果您是唯一使用 AWS 帳戶的使用者，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶管理員提供登入指示。
 - 如果您是課堂裡的學生，請洽詢講師以取得登入指示。
2. 在頂端導覽列中登入 AWS Cloud9 主控台後，請選擇 AWS 區域來建立環境。如需可用的清單 AWS 區域，請參閱[AWS Cloud9](#)中的 AWS 一般參考。



3. 在其中一個顯示的位置選擇放大的 Create environment (建立環境) 按鈕。

如果您還沒有 AWS Cloud9 環境，按鈕會顯示在歡迎頁面上。



如果您已經有 AWS Cloud9 環境，按鈕會顯示如下。




4. 在 Create environment (建立環境) 頁面的 Name (名稱) 上，輸入環境的名稱。
5. 在 Description (描述) 中，輸入環境的相關資訊。在本教學中，使用 This environment is for the AWS Cloud9 tutorial.
6. 在 Environment type (環境類型) 中，選擇 New EC2 instance (新的 EC2 執行個體)，建立 Amazon EC2 環境：
 - EC2 instance (EC2 執行個體)—啟動 AWS Cloud9 可以直接透過 SSH 進行連線的新 Amazon EC2 執行個體。您可以使用 Systems Manager 與新的 Amazon EC2 執行個體互動，如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

- Existing compute (現有運算) – 啟動需要 SSH 登入詳細資訊的現有 Amazon EC2 執行個體，而 Amazon EC2 執行個體必須具有輸入安全群組規則。
- 如果您選取 Existing compute (現有運算) 選項，則會自動建立服務角色。您也可以設定畫面底部的備註中檢視服務角色名稱。


 Note

使用現有運算的 Amazon EC2 執行個體建立的 AWS Cloud9 環境無法使用自動關機。

 Warning

為您的環境建立 Amazon EC2 執行個體可能會產生 Amazon EC2 的 AWS 帳戶費用。使用 Systems Manager 管理 EC2 執行個體的連線不需要額外費用。

7. 在新 EC2 執行個體面板上的 Instance type (執行個體類型)，保留預設選項。此選項可能具有較少的 RAM 和 vCPU。但是，這個記憶體容量對本教學課程來說已十分充足。

 Warning

選擇內含更多 RAM 和 vCPU 的執行個體類型可能讓 AWS 帳戶產生額外的 Amazon EC2 費用。

8. 對於平台，選擇您想要的 Amazon EC2 實例的類型：Amazon Linux 2023，Amazon Linux 2 或 Ubuntu 22.04 LTS。AWS Cloud9 建立執行個體，然後將環境連接至該執行個體。

 Important

我們建議您選擇適用於 EC2 環境的 Amazon Linux 2023 選項。Amazon Linux 2023 AMI 除了提供安全、穩定和高效能的運行時間環境之外，還包括到 2024 年的長期支持。如需詳細資訊，請參閱 [AL2023 頁面](#)。

9. 選擇 Timeout (逾時) 的時間。此選項會決定自動休眠之前 AWS Cloud9 處於非作用狀態的時間長度。當連接至環境的 IDE 的所有 web 瀏覽器執行個體皆已關閉時，AWS Cloud9 會按照指定的時間長度等候，然後關閉環境的 Amazon EC2 執行個體。

⚠ Warning

選擇更長的一段時間可能會導致您的 AWS 帳戶 產生更多費用。

10. 在 Network settings (網路設定) 面板上，從下列兩個選項中，選擇存取環境的方式：
 - AWS System Manager (SSM) – 此方法存取使用 SSM 的環境，而無需開啟輸入連接埠。
 - Secure Shell (SSH) (安全通訊殼層 (SSH)) – 此方法使用 SSH 存取環境，且需開啟傳入連接埠。
11. 選擇 VPC Settings (VPC 設定) 以顯示您環境的 Amazon Virtual Private Cloud 和子網路。AWS Cloud9 使用 Amazon Virtual Private Cloud (Amazon VPC) 與新建立的 Amazon EC2 執行個體進行通訊。在本教學課程中，我們建議您不要變更預先選取的預設設定。使用預設設定時，AWS Cloud9 會嘗試在與新環境相同的 AWS 帳戶 和區域中自動使用預設 VPC 搭配其單一子網路。

如需 Amazon VPC 選項的詳細資訊，請參閱[使用主控台建立 EC2 環境](#)，以及 [AWS Cloud9 開發環境的 VPC 設定](#)。
12. 藉由為每個標籤提供一個 Key (索引鍵) 和 Value (值)，最多新增 50 個標籤。選取 Add new tag (新增標籤) 來執行此操作。標籤將附加至 AWS Cloud9 環境做為資源標籤，並傳播至以下基礎資源：AWS CloudFormation 堆疊、Amazon EC2 執行個體，以及 Amazon EC2 安全群組。若要進一步了解標籤，請參閱 [IAM 使用者指南](#) 中的 [使用 AWS 資源標籤控制存取](#)，以及本指南中的 [進階資訊](#)。

⚠ Warning

如果您在建立這些標籤之後予以更新，這些變更不會傳播至基礎資源。如需詳細資訊，請參閱[標籤](#)相關進階資訊中的 [將標籤更新傳播至基礎資源](#)。

13. 選擇 Create (建立) 以建立您的環境，然後系統會將您重新導向至首頁。若成功建立帳戶，AWS Cloud9 主控台上方會出現綠色閃爍列。您可以選取新環境，然後選擇 Open in Cloud9 (在 Cloud9 中開啟) 以啟動 IDE。

Delete

View details

Open in Cloud9 

Create environment

若無法建立帳戶，AWS Cloud9 主控台上方會出現紅色閃爍列。您的 Web 瀏覽器、AWS 存取許可、執行個體或相關的網路問題皆可能導致帳戶無法建立。您可以在 [AWS Cloud9 疑難排解區段](#) 找到可行修正方法。

Note

AWS Cloud9 同時支援 IMDSv1 和 IMDSv2。我們建議採用 IMDSv2，因為與 IMDSv1 相比，它提供了更高的安全性層級。如需有關 IMDSv2 優點的詳細資訊，請參閱 [AWS 安全性部落格](#)。如需有關如何從 IMDSv1 轉換到 IMDSv2 的詳細資訊，請參閱《Linux 執行個體的 Amazon EC2 使用者指南》中的〈[轉換為使用執行個體中繼資料服務版本 2](#)〉。

Note

如果您的環境使用代理來存取網際網路，您必須提供代理詳細資訊給 AWS Cloud9 以便安裝相依性。如需詳細資訊，請參閱 [安裝相依性失敗](#)。

下一步驟

[步驟 2：IDE 基本導覽](#)

步驟 2：IDE 基本導覽

(上一個步驟：[步驟 1：建立環境](#))

這個教學課程章節介紹幾種方法，協助您使用 AWS Cloud9 IDE 來建立及測試應用程式。

- 您可以使用 editor (編輯器) 視窗來建立和編輯程式碼。
- 您可以使用 terminal (終端機) 視窗或 Run Configuration (執行組態) 視窗來執行程式碼，無需進行偵錯。
- 您可以使用 Debugger (除錯器) 視窗來除錯您的程式碼。

使用 JavaScript 和 Node.js 引擎執行這三項工作。如需使用其他程式設計語言的說明，請參閱 [AWS Cloud9 教學課程](#)。

主題

- [準備好您的環境](#)
- [撰寫程式碼](#)
- [執行您的程式碼](#)
- [除錯您的程式碼](#)
- [下一步驟](#)

準備好您的環境

您需要運行和調試 JavaScript 代碼的大多數工具已經為您安裝。不過，在本教學中，您需要一個額外的 Node.js 套件。安裝方式如下。

1. 在選單列 (位於 AWS Cloud9 IDE 的頂端) 中選擇 Window (視窗)、New Terminal (新增終端機)，或使用現有的終端機視窗。
2. 在終端機視窗 (IDE 底部的其中一個標籤) 中，輸入以下內容。

```
npm install readline-sync
```

確認結果類似如下：如果同時顯示 npm WARN 訊息，可以忽略。

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

撰寫程式碼

從撰寫一些程式碼開始。

1. 在選單列上，選擇 File (檔案)、New File (新增檔案)。
2. 將以下內容添加 JavaScript 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. 選擇 File (檔案)、Save (儲存)，然後儲存檔案為 `hello-cloud9.js`。

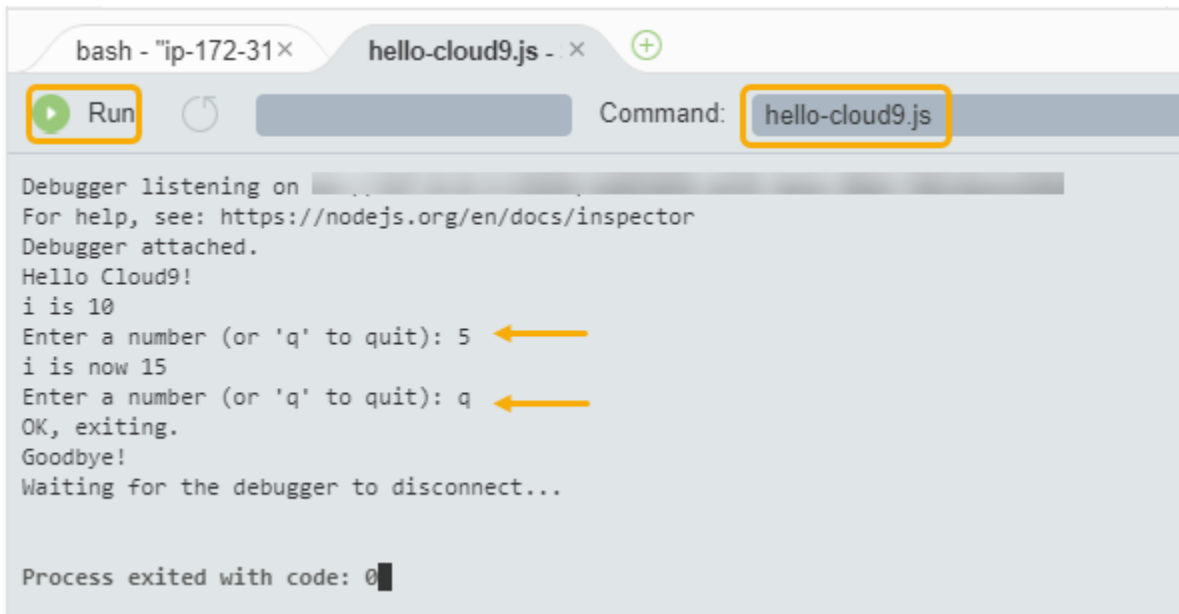
執行您的程式碼

接著，您可以執程式碼。

視您使用的程式設計語言而定，執程式碼的方式可能有很多種。本教程使用 JavaScript，您可以使用終端窗口或運行配置窗口運行。

使用 Run Configuration (執行組態) 視窗執程式碼

1. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在新的 Run Configuration (執行組態) 視窗 (IDE 底部的其中一個標籤)，將 `hello-cloud9.js` 輸入 Command (命令) 欄位，然後選擇 Run (執行)。
3. 請確定 Run Configuration (執行組態) 提示為作用中，然後在提示時輸入數字來與應用程式互動。
4. 在 Run Configuration (執行組態) 視窗中檢視您程式碼的輸出。它應類似以下內容。



```
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
```

使用終端機視窗執行程式碼

1. 移至您先前使用的終端機視窗 (或開啟新的終端視窗)。
2. 在終端機視窗中，於終端機提示下輸入 `ls`，然後確認您的程式碼檔案位於檔案清單中。
3. 在提示字元中輸入 `node hello-cloud9.js` 以啟動應用程式。
4. 在提示字元中輸入數字，以與應用程式互動。
5. 在終端機視窗中檢視程式碼的輸出。它應類似以下內容。

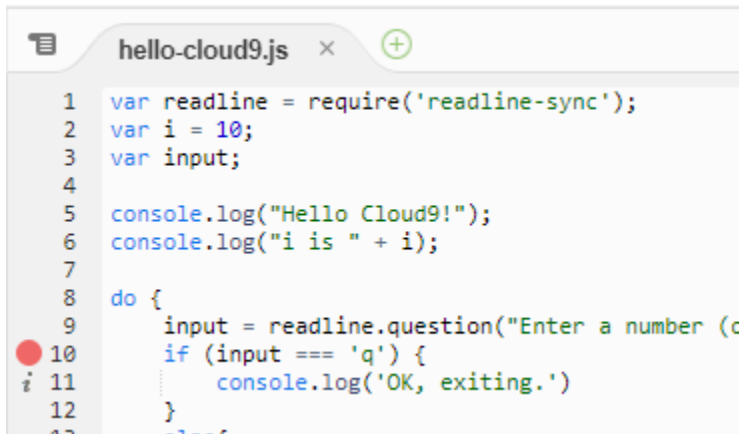


```
node - "ip-172-31" x hello-cloud9.js - ! x +
Admin:~/environment $ node hello-cloud9.js
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Admin:~/environment $
```

除錯您的程式碼

最後，您可以使用 Debugger (除錯器) 視窗偵錯您的程式碼。

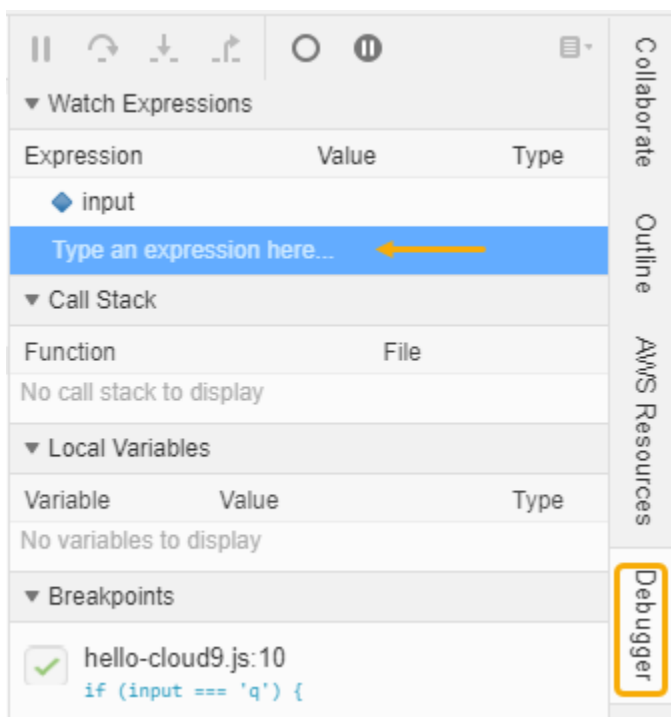
1. 選擇第 10 行旁的邊界，將中斷點新增至第 10 行的程式碼 (if (input === 'q'))。該行號旁會顯示紅色圓圈，如下所示。



```
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 (0
10    if (input === 'q') {
11        console.log('OK, exiting.')
12    }
13
```

2. 選擇 IDE 右側的 Debugger (除錯器) 按鈕，開啟 Debugger (除錯器) 視窗。或者，選擇選單列上的 Window (視窗)、Debugger (除錯器)。

然後在 Debugger (除錯器) 視窗中的 Watch Expressions (監看表達式) 區段選擇 Type an expression here (在此輸入表達式)，以便監看 input 變數。



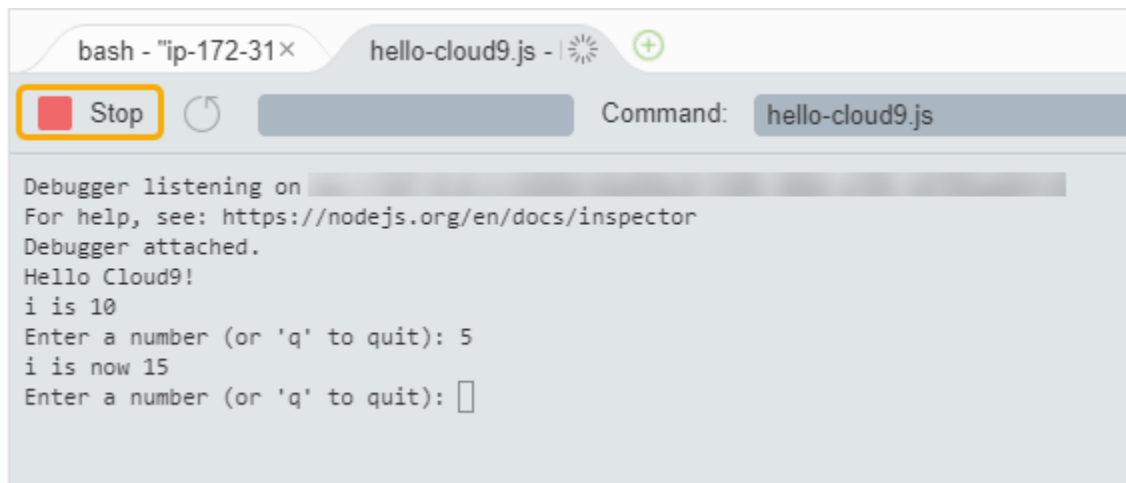
3. 前往您先前用來執行程式碼的 Run Configuration (執行組態) 視窗。選擇執行。

或者，您可以開啟新的 Run Configuration (執行組態) 視窗，並開始執行程式碼。從選單欄選擇 Run (執行)、Run With (執行工具)、Node.js 以執行此動作。

- 在 Run Configuration (執行組態) 提示字元中輸入數字，並查看程式碼在第 10 行暫停。Debugger (除錯器) 視窗會顯示您在 Watch Expressions (監看運算式) 中輸入的值。



- 在 Debugger (除錯器) 視窗中，選擇 Resume (繼續)。這是在上一個螢幕截取畫面中反白顯示的藍色箭頭圖示。
- 選擇 Run Configuration (執行組態) 視窗中的 Stop (停止) 以停止除錯器。



下一步驟

[步驟 3：清理](#)

步驟 3：清理

(上一個步驟：[步驟 2：IDE 基本導覽](#))

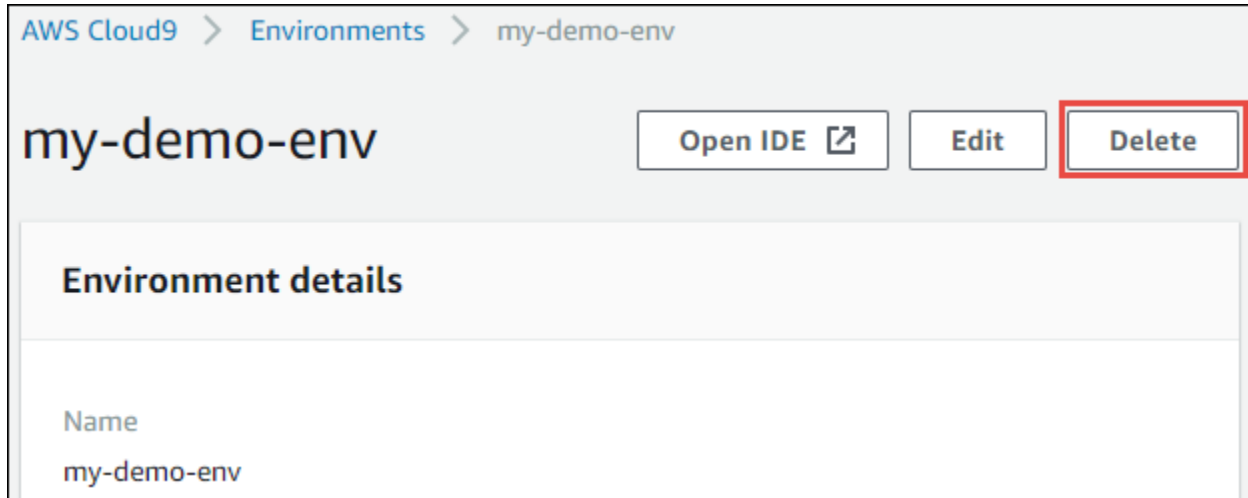
為避免您的 AWS 帳戶 持續產生與本教學相關的費用，請刪除該環境。

Warning

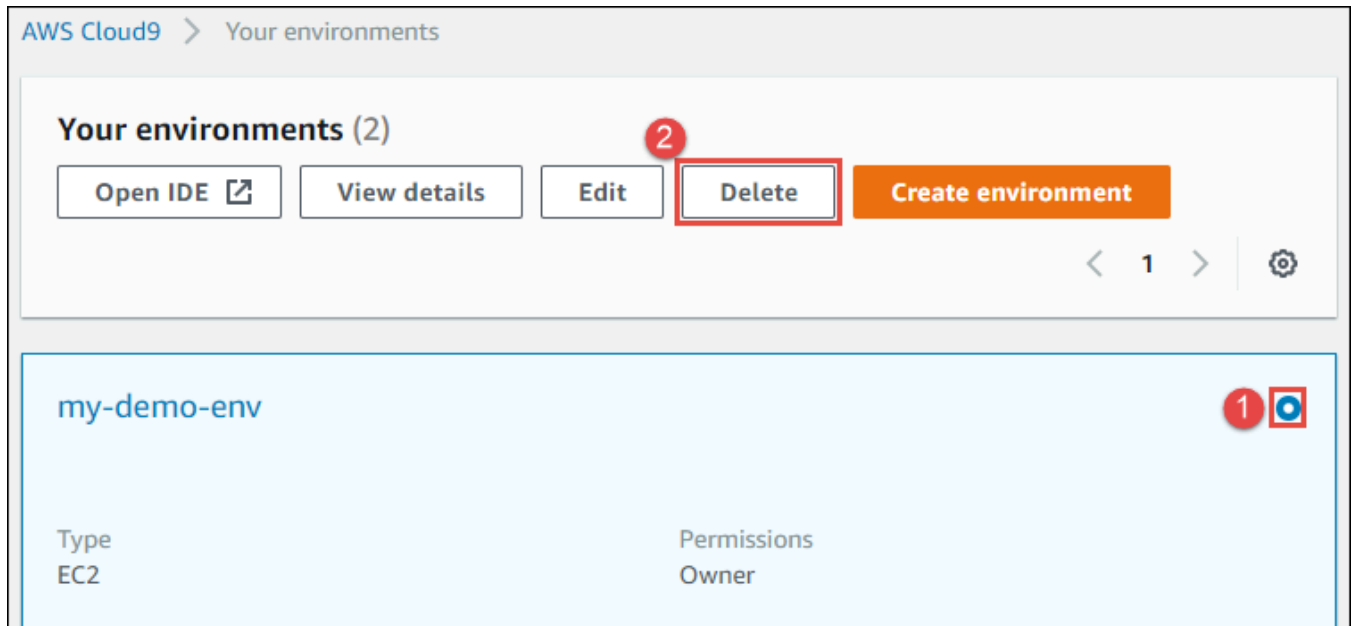
一旦刪除您的環境便無法復原。

使用 AWS Cloud9 主控台刪除環境

1. 若要開啟儀表板，請在 IDE 的選單列上選擇 AWS Cloud9、Go To Your Dashboard (移至儀表板)。
2. 執行以下任意一項：
 - 選擇my-demo-environment卡片內的標題，然後選擇 [刪除]。



- 選取my-demo-environment卡片，然後選擇 [刪除]。



3. 在 Delete (刪除) 對話方塊中輸入 Delete，然後選擇 Delete (刪除)。刪除操作需要幾分鐘的時間。

Note

如果您確實遵循此教學，則環境就會是 EC2 環境，且 AWS Cloud9 也會終止已連線到該環境的 Amazon EC2 執行個體。

不過，如果您使用 SSH 環境，並未遵循此教學，且該環境已連線到 Amazon EC2 執行個體，則 AWS Cloud9 不會終止該執行個體。如果您之後不終止該執行個體，您的 AWS 帳戶可能會持續產生與該執行個體相關的 Amazon EC2 費用。

下一步驟

[相關資訊](#)

相關資訊

以下是 [教學課程：Hello AWS Cloud9 \(主控台\)](#) 的其他資訊。

- 當您建立 EC2 環境時，環境在預設情況下並不包含任何範本程式碼。若要建立含範本程式碼的環境，請參閱下列其中一個主題：
 - [在 AWS Cloud9 整合開發環境 \(IDE\) 使用 Amazon Lightsail 執行個體](#)

- [在AWS Cloud9整合式開發環境 \(IDE\) 中使用AWS CodeStar專案](#)
- 在建立 AWS Cloud9 開發環境的過程中，您指示 AWS Cloud9 建立 Amazon EC2 執行個體。AWS Cloud9 建立執行個體後，再將環境連接至該執行個體。您也可以使用現有的雲端運算執行個體或自己的伺服器，稱為 SSH 環境。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。

選用的後續步驟

請探索下列任何或所有主題以繼續熟悉 AWS Cloud9。

任務	參閱此主題
進一步了解您可以如何利用環境。	使用環境 AWS Cloud9
請嘗試其他電腦語言。	AWS Cloud9 教學課程
進一步了解 AWS Cloud9 IDE。	使用 IDE 中的 <u>導覽 AWS Cloud9 IDE</u>
邀請其他人即時透過文字聊天支援功能一起使用您的新環境。	使用 AWS Cloud9 的共用環境
建立 SSH 環境。這些環境使用您建立的雲端運算執行個體或伺服器，而不是 AWS Cloud9 為您建立的 Amazon EC2 執行個體。	在 AWS Cloud9 中建立環境 和 SSH 環境主機要求
使用 AWS Toolkit 建立、執行與除錯 AWS Lambda 函數中的程式碼及無伺服器應用程式。	透過 AWS Toolkit 使用 AWS Lambda 函式
搭配 AWS Cloud9 使用 Amazon Lightsail。	在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體
搭配 AWS CodeStar 使用 AWS Cloud9。	在AWS Cloud9整合式開發環境 (IDE) 中使用 AWS CodeStar專案
搭配 AWS CodePipeline 使用 AWS Cloud9。	在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline
搭AWS Cloud9配AWS CLI、 <code>、</code> 、AWS Cloud Development Kit (AWSCDK) 或 Amazon DynamoDB GitHub、Node.js、Python 或其他	AWS Cloud9 教學課程

任務	參閱此主題
程式設計語言搭配使用。AWS CloudShell AWS CodeCommit	
使用 AWS RoboMaker.	在開 發人AWS RoboMaker 員指南AWS Cloud9 中進行開發

如需透過社群獲取 AWS Cloud9 的相關協助，請參閱 [AWS Cloud9 開發論壇](#)。(當您進入此論壇時，AWS 可能會要求您登入。)

如需直接從 AWS 獲取 AWS Cloud9 的相關協助，請參閱 [AWS 支援](#) 頁面上的支援選項。

教學課程：Hello AWS Cloud9 (CLI)

本教學課程提供初次了解 AWS Cloud9。它使用 [AWS Command Line Interface \(AWS CLI\)](#)，這可讓您使用命令列 (而非[圖形使用者界面](#)) 來設定和縮減所需的資源。

在此教學課程中，您會設定 AWS Cloud9 開發環境然後使用 AWS Cloud9 IDE 對第一個應用程式進行編碼、執行和除錯。

本教學課程大約需要一小時。

Warning

完成此教學可能會向您的 AWS 帳戶收取費用。收費的項目可能包括 Amazon EC2。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

必要條件

若要成功完成本教學課程，您必須先完成 [設定 AWS Cloud9](#) 中的步驟。

步驟

- [步驟 1：建立環境](#)
- [步驟 2：IDE 基本導覽](#)
- [步驟 3：清理](#)

- [相關資訊](#)

步驟 1：建立環境

([教學課程：Hello AWS Cloud9 \(CLI\)](#) 的第一個步驟)

在此步驟中，您將使用 AWS CLI 建立 AWS Cloud9 開發環境。

在 AWS Cloud9 中，development environment (開發環境) (或稱 environment (環境)) 是讓您存放開發專案檔案和執行各種工具來開發應用程式的位置。在此教學課程中，您會建立 EC2 環境，然後使用該環境中的檔案和工具。

使用 AWS CLI 建立 EC2 環境

1. 如果您尚未安裝並設定 AWS CLI，請先執行這項動作。若要執行此作業，請參閱以下 AWS Command Line Interface 使用者指南中的說明：

- [安裝 AWS Command Line Interface](#)
- [快速組態](#)

您可以使用憑證來為下列其中一項設定 AWS CLI：

- 您在 [AWS Cloud9 的小組設定](#) 中建立的 IAM 使用者。
 - 您 AWS 帳戶中的 IAM 管理員使用者 (如果您將在整個帳戶中為多個使用者定期使用 AWS Cloud9 資源)。若您無法以 IAM 管理員身分設定 AWS CLI，請洽詢您的 AWS 帳戶管理員。如需詳細資訊，請參閱 IAM 使用者指南中的 [建立您的第一個 IAM 管理員使用者和群組](#)。
 - 一名 AWS 帳戶根使用者，但前提是您始終是唯一使用您自己 AWS 帳戶的使用者，並且不需要與他人共用您的環境。我們不建議使用此選項，因為這並非 AWS 安全最佳實務。如需詳細資訊，請參閱中的 [建立、停用和刪除 AWS 帳戶的存取金鑰](#) [Amazon Web Services 一般參考](#)。
 - 至於其他選項，請參閱您的 AWS 帳戶管理員或課堂教師。
2. 在以下 AWS Cloud9 命令中，提供 --region 和 --subnet-id 的值。然後執行命令並記下 "environmentId" 值，以供稍後清除。

```
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` 代表環境的名稱。在此教學中，我們使用 `my-demo-environment` 這個名稱。
- `--description` 代表環境的選用說明。
- `--instance-type` 代表 AWS Cloud9 將啟動並連線到新環境的 Amazon EC2 執行個體類型。本範例指定了 `t2.micro`，具有相對較低的 RAM 和 vCPU，對本教學來說足夠。指定含有更多 RAM 和 vCPU 的執行個體類型可能讓您的 AWS 帳戶承擔額外的 Amazon EC2 費用。如需可用的執行個體類型清單，請參閱 AWS Cloud9 主控台下的建立環境精靈。
- `--image-id` 會指定用於建立 EC2 執行個體的 Amazon Machine Image (AMI) 的識別碼。若要為執行個體選擇 AMI，您必須指定有效的 AMI 別名或有效的 AWS Systems Manager (SSM) 路徑。在上面的例子中指定的是 Amazon Linux 2 AMI 的 SSM 路徑。

若要取得更多資訊，請參閱《AWS CLI 指令參考》中的 [create-environment-ec2](#)。

- `--region` 代表供 AWS Cloud9 在其中建立環境的 AWS 區域 ID。如需可用 AWS 區域的清單，請參閱 [AWS Cloud9](#) 中的 Amazon Web Services 一般參考。
- `--connection-type CONNECT_SSM` 會指定 AWS Cloud9 透過 Systems Manager 連線到 Amazon EC2 執行個體。此選項可確保沒有任何傳入流量允許進入執行個體。如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

Note

當使用此選項時，若尚未建立 `AWSCloud9SSMAccessRole` 服務角色和 `AWSCloud9SSMInstanceProfile`，則您需要建立它們。如需詳細資訊，請參閱 [使用 AWS CLI 管理 Systems Manager 的執行個體描述檔](#)。

- `--subnet-id` 代表您希望 AWS Cloud9 使用的子網路。以 Amazon Virtual Private Cloud (VPC) 子網路的 ID 取代 `subnet-12a3456b`，此 ID 必須與 AWS Cloud9 相容。如需詳細資訊，請參閱 [AWS Cloud9 開發環境的 VPC 設定](#) 中的 [建立 VPC 以及其他 VPC 資源](#)。
- 在為環境連線到 IDE 的所有 Web 瀏覽器執行個體關閉後，AWS Cloud9 會關閉環境的 Amazon EC2 執行個體。若要設定此時間，請新增 `--automatic-stop-time-minutes` 和分鐘數。較短的期間可能導致您的 AWS 帳戶產生較少費用。同樣地，較長的時間可能會產生較多的費用。
- 預設情況下，呼叫此命令的實體擁有該環境。若要變更，請新增 `--owner-id` 以及所屬實體的 Amazon Resource Name (ARN)。

3. 在您成功執行此命令後，請為新建立的環境開啟 AWS Cloud9 IDE。若要執行此操作，請參閱 [在 AWS Cloud9 中開啟環境](#)。然後返回本主題，並繼續進行 [步驟 2：IDE 基本導覽](#)，以了解如何運用 AWS Cloud9 IDE 來使用您的新環境。

如果您嘗試開啟環境，但 AWS Cloud9 經過至少五分鐘後仍未顯示 IDE，問題可能出在 Web 瀏覽器、AWS 存取許可、執行個體，或是關聯 VPC。如需可行修正方法，請參閱 [無法開啟環境](#)。

後續步驟

[步驟 2：IDE 基本導覽](#)

步驟 2：IDE 基本導覽

(上一個步驟：[步驟 1：建立環境](#))

這個教學課程章節介紹幾種方法，協助您使用 AWS Cloud9 IDE 來建立及測試應用程式。

- 您可以使用 editor (編輯器) 視窗來建立和編輯程式碼。
- 您可以使用 terminal (終端機) 視窗或 Run Configuration (執行組態) 視窗來執行程式碼，無需進行偵錯。
- 您可以使用 Debugger (除錯器) 視窗來除錯您的程式碼。

使用 JavaScript 和 Node.js 引擎執行這三項工作。如需使用其他程式設計語言的說明，請參閱 [AWS Cloud9 教學課程](#)。

主題

- [準備好您的環境](#)
- [撰寫程式碼](#)
- [執行您的程式碼](#)
- [除錯您的程式碼](#)
- [後續步驟](#)

準備好您的環境

您需要運行和調試 JavaScript 代碼的大多數工具已經為您安裝。不過，在本教學中，您需要一個額外的 Node.js 套件。安裝方式如下。

1. 在選單列 (位於 AWS Cloud9 IDE 的頂端) 中選擇 Window (視窗)、New Terminal (新增終端機) , 或使用現有的終端機視窗。
2. 在終端機視窗 (IDE 底部的其中一個標籤) 中 , 輸入以下內容。

```
npm install readline-sync
```

確認結果類似如下 : 如果同時顯示 npm WARN 訊息 , 可以忽略。

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

撰寫程式碼

從撰寫一些程式碼開始。

1. 在選單列上 , 選擇 File (檔案)、New File (新增檔案)。
2. 將以下內容添加 JavaScript 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. 選擇 File (檔案)、Save (儲存) , 然後儲存檔案為 hello-cloud9.js。

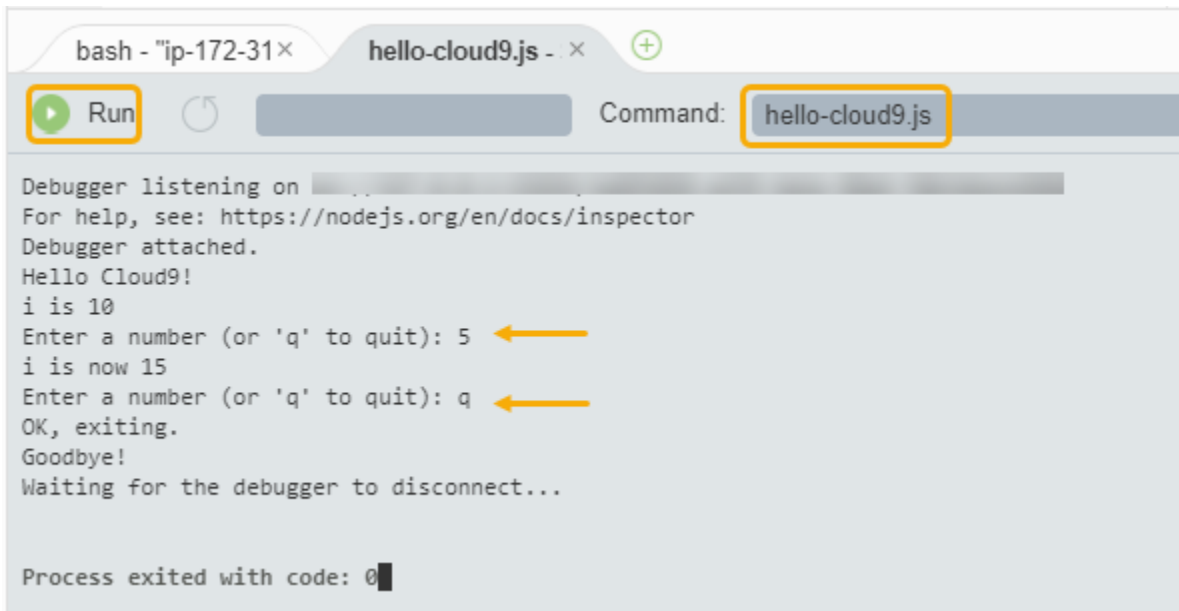
執行您的程式碼

接著，您可以執行程式碼。

視您使用的程式設計語言而定，執行程式碼的方式可能有很多種。本教程使用 JavaScript，您可以使用終端窗口或運行配置窗口運行。

使用 Run Configuration (執行組態) 視窗執行程式碼

1. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在新的 Run Configuration (執行組態) 視窗 (IDE 底部的其中一個標籤)，將 `hello-cloud9.js` 輸入 Command (命令) 欄位，然後選擇 Run (執行)。
3. 請確定 Run Configuration (執行組態) 提示為作用中，然後在提示時輸入數字來與應用程式互動。
4. 在 Run Configuration (執行組態) 視窗中檢視您程式碼的輸出。它應類似以下內容。



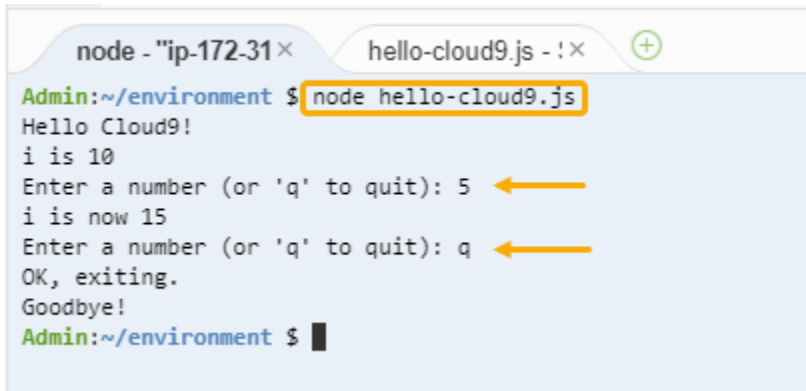
```
bash - "ip-172-31x" hello-cloud9.js - x
Run Command: hello-cloud9.js
Debugger listening on [redacted]
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
Hello Cloud9!
i is 10
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.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

Process exited with code: 0
```

使用終端機視窗執行程式碼

1. 移至您先前使用的終端機視窗 (或開啟新的終端視窗)。
2. 在終端機視窗中，於終端機提示下輸入 `ls`，然後確認您的程式碼檔案位於檔案清單中。
3. 在提示字元中輸入 `node hello-cloud9.js` 以啟動應用程式。
4. 在提示字元中輸入數字，以與應用程式互動。

5. 在終端機視窗中檢視程式碼的輸出。它應類似以下內容。

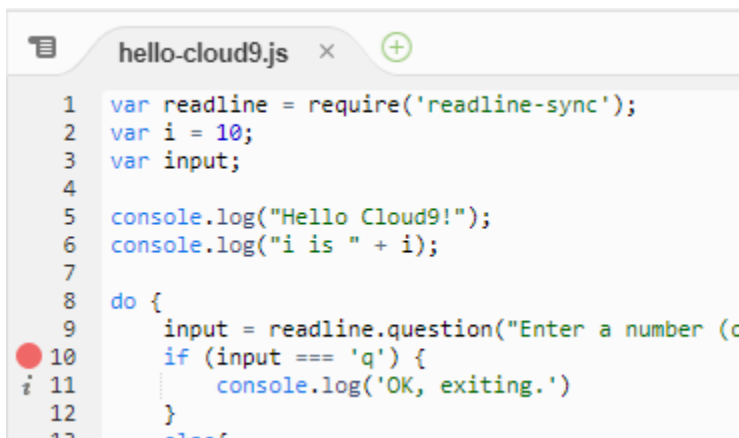


```
Admin:~/environment $ node hello-cloud9.js
Hello Cloud9!
i is 10
Enter a number (or 'q' to quit): 5
i is now 15
Enter a number (or 'q' to quit): q
OK, exiting.
Goodbye!
Admin:~/environment $
```

除錯您的程式碼

最後，您可以使用 Debugger (除錯器) 視窗偵錯您的程式碼。

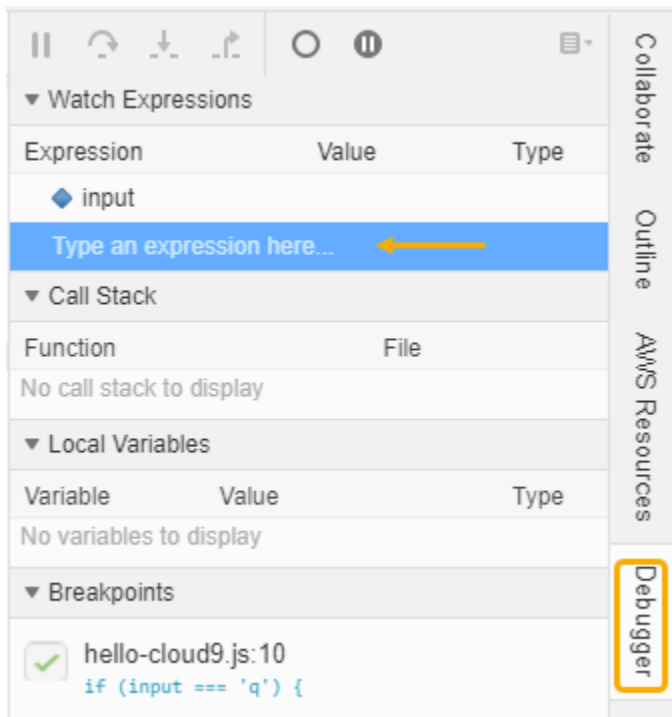
1. 選擇第 10 行旁的邊界，將中斷點新增至第 10 行的程式碼 (if (input === 'q'))。該行號旁會顯示紅色圓圈，如下所示。



```
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. 選擇 IDE 右側的 Debugger (除錯器) 按鈕，開啟 Debugger (除錯器) 視窗。或者，選擇選單列上的 Window (視窗)、Debugger (除錯器)。

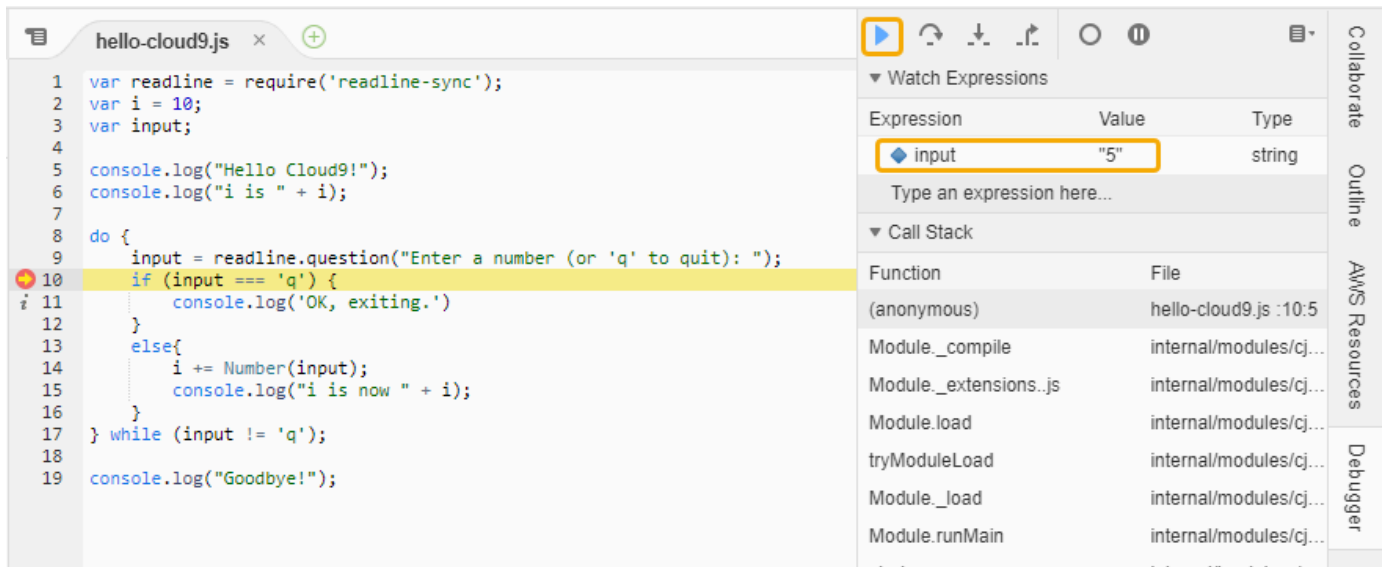
然後在 Debugger (除錯器) 視窗中的 Watch Expressions (監看表達式) 區段選擇 Type an expression here (在此輸入表達式)，以便監看 input 變數。



3. 前往您先前用來執行程式碼的 Run Configuration (執行組態) 視窗。選擇執行。

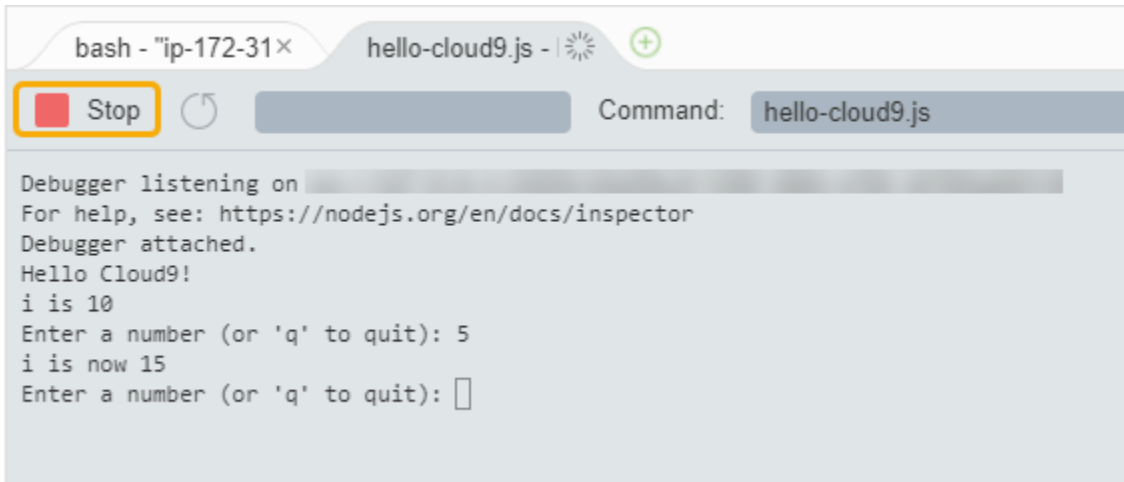
或者，您可以開啟新的 Run Configuration (執行組態) 視窗，並開始執行程式碼。從選單欄選擇 Run (執行)、Run With (執行工具)、Node.js 以執行此動作。

4. 在 Run Configuration (執行組態) 提示字元中輸入數字，並查看程式碼在第 10 行暫停。Debugger (除錯器) 視窗會顯示您在 Watch Expressions (監看運算式) 中輸入的值。



5. 在 Debugger (除錯器) 視窗中，選擇 Resume (繼續)。這是在上一個螢幕截取畫面中反白顯示的藍色箭頭圖示。

- 選擇 Run Configuration (執行組態) 視窗中的 Stop (停止) 以停止除錯器。



後續步驟

[步驟 3：清理](#)

步驟 3：清理

(上一個步驟：[步驟 2：IDE 基本導覽](#))

為避免您的 AWS 帳戶持續產生與本教學相關的費用，您應刪除該環境。

Warning

刪除環境後就無法復原。

以 AWS CLI 刪除環境

- 執行 AWS Cloud9 delete-environment 命令，指定要刪除的環境 ID。

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

在上述命令中，以用於建立環境的 AWS 區域取代 MY-REGION，並以要刪除的環境的 ID 取代 12a34567b8cd9012345ef67abcd890e1。

如果您在建立環境時未儲存 ID，可使用 AWS Cloud9 主控台來找到 ID。在主控台中選取環境名稱，然後尋找 Environment ARN (環境 ARN) 的最後部分。

2. 如果您為此教學課程建立了 Amazon VPC，但不再需要它，請前往 <https://console.aws.amazon.com/vpc> 使用 Amazon VPC 主控台刪除 VPC。

後續步驟

[相關資訊](#)

相關資訊

以下是 [教學課程：Hello AWS Cloud9 \(CLI\)](#) 的其他資訊。

- 當您建立 EC2 環境時，環境在預設情況下並不包含任何範本程式碼。若要建立含範本程式碼的環境，請參閱下列其中一個主題：
 - [在 AWS Cloud9 整合開發環境 \(IDE\) 使用 Amazon Lightsail 執行個體](#)
 - [在 AWS Cloud9 整合式開發環境 \(IDE\) 中使用 AWS CodeStar 專案](#)
- 在建立 AWS Cloud9 開發環境的過程中，您指示 AWS Cloud9 建立 Amazon EC2 執行個體。AWS Cloud9 建立執行個體後，再將環境連接至該執行個體。您也可以使用現有的雲端運算執行個體或自己的伺服器，稱為 SSH 環境。如需詳細資訊，請參閱 [在 AWS Cloud9 中建立環境](#)。

選用後續步驟

請探索下列任何或所有主題以繼續熟悉 AWS Cloud9。

任務	參閱此主題
進一步了解您可以如何利用環境。	使用環境 AWS Cloud9
請嘗試其他電腦語言。	AWS Cloud9 教學課程
進一步了解 AWS Cloud9 IDE。	使用 IDE 中的 導覽 AWS Cloud9 IDE
邀請其他人即時透過文字聊天支援功能一起使用您的新環境。	使用 AWS Cloud9 的共用環境

任務	參閱此主題
建立 SSH 環境。這些環境使用您建立的雲端運算執行個體或伺服器，而不是 AWS Cloud9 為您建立的 Amazon EC2 執行個體。	在 AWS Cloud9 中建立環境 和 SSH 環境主機要求
使用 AWS Toolkit 建立、執行與除錯 AWS Lambda 函數中的程式碼及無伺服器應用程式。	透過 AWS Toolkit 使用 AWS Lambda 函式
搭配 AWS Cloud9 使用 Amazon Lightsail。	在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體
搭配 AWS CodeStar 使用 AWS Cloud9。	在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CodeStar 專案
搭配 AWS CodePipeline 使用 AWS Cloud9。	在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline
搭 AWS Cloud9 配 AWS CLI、 <code>aws</code> 、AWS Cloud Development Kit (AWS CDK) 或 Amazon DynamoDB、GitHub、Node.js、Python 或其他程式設計語言搭配使用。AWS CloudShell、AWS CodeCommit	AWS Cloud9 教學課程
使用中智慧型機器人應用程式的程式碼 AWS RoboMaker。	在 開發人 AWS RoboMaker 員指南 AWS Cloud9 中進行開發

如需透過社群獲取 AWS Cloud9 的相關協助，請參閱 [AWS Cloud9 開發論壇](#)。(當您進入此論壇時，AWS 可能會要求您登入。)

如需直接從 AWS 獲取 AWS Cloud9 的相關協助，請參閱 [AWS 支援](#) 頁面上的支援選項。

使用環境 AWS Cloud9

開發環境是您儲存專案檔案的位置，以及執行工具來開發應用程式的位置。AWS Cloud9

AWS Cloud9 提供兩種類型的開發環境：EC2 環境和 SSH 環境。若要了解這些開發環境之間的主要異同之處，請參閱 [比較 AWS Cloud9 中的 EC2 環境與 SSH 環境](#)。

閱讀下列一或多個主題，了解如何使用中的環境。AWS Cloud9

主題

- [在 AWS Cloud9 中建立環境](#)
- [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)
- [在 AWS Cloud9 中開啟環境](#)
- [從 AWS Cloud9 的環境呼叫 AWS 服務](#)
- [變更 AWS Cloud9 的環境設定](#)
- [使用 AWS Cloud9 的共用環境](#)
- [移動環境並調整 Amazon EBS 磁碟區大小或予以加密](#)
- [刪除 AWS Cloud9 中的環境](#)

在 AWS Cloud9 中建立環境

如要建立 AWS Cloud9 開發環境，請根據您預計使用 AWS Cloud9 的方式，依照本文提供的其中一項程序操作。

如果您不確定如何選擇，我們建議 [建立 EC2 環境](#)。

如需快速設定，請建立 EC2 環境。AWS Cloud9 會自動在您的 AWS 帳戶建立和設定新的 Amazon EC2 執行個體。AWS Cloud9 也會自動為您將該新執行個體連線至環境。

若要了解各種開發環境之間的主要異同之處，請參閱 [比較 AWS Cloud9 中的 EC2 環境與 SSH 環境](#)。

原始程式碼提供者	開發環境主機提供者	相關程序
您	AWS Cloud9	建立 EC2 環境
您	您	建立 SSH 環境
Amazon Lightsail 或您	您 (使用 Lightsail)	在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體
AWS CodeStar 或您	AWS Cloud9 (使用 AWS CodeStar)	在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CodeStar 專案
您 (使用 AWS CodePipeline)	AWS Cloud9 或您	建立 EC2 或 SSH 環境，以及在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline
您 (使用 AWS CodeCommit)	AWS Cloud9 或您	AWS CodeCommit 教學課程 AWS Cloud9
您 (使用 GitHub)	AWS Cloud9 或您	建立 EC2 或 SSH 環境，並使用 Git 面板介面

主題

- [建立 EC2 環境](#)
- [建立 SSH 環境](#)

建立 EC2 環境

在這個程序中，AWS Cloud9 會建立 EC2 環境並建立新的 Amazon EC2 執行個體，然後將環境連線到這個執行個體。AWS Cloud9 會管理此執行個體的生命週期，包括啟動、停止，以及在必要時重新啟動執行個體。如果刪除這個環境，AWS Cloud9 會自動終止這個執行個體。

您可以使用 [AWS Cloud9 主控台](#) 或 [程式碼](#) 建立 AWS Cloud9 EC2 開發環境。

Note

完成此程序可能會向您的 AWS 帳戶收取費用。收費的項目可能包括 Amazon EC2。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

Warning

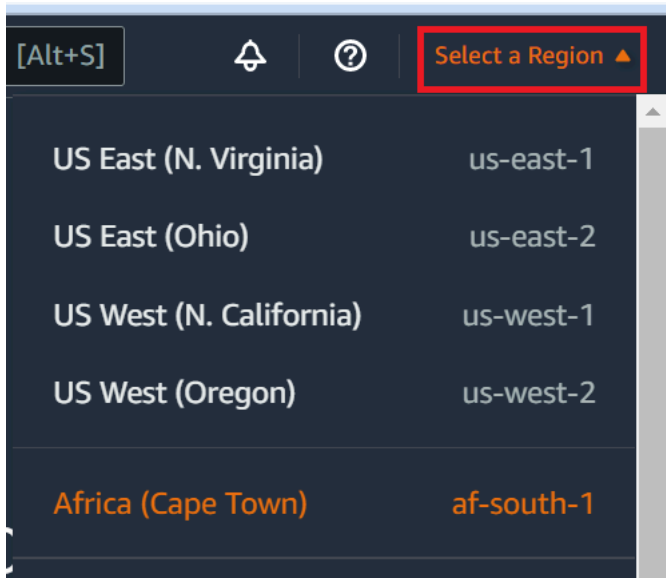
與AWS Cloud9AWS Control Tower主動控制 [CT.EC2.PR. 8](#) 存在相容性問題。如果啟用此控制項，則無法在中建立 EC2 環境AWS Cloud9。如需此問題的詳細資訊，請參閱[疑難排解AWS Cloud9](#)。

必要條件

完成 [設定 AWS Cloud9](#) 中的步驟，您就能登入 AWS Cloud9 主控台並建立環境。

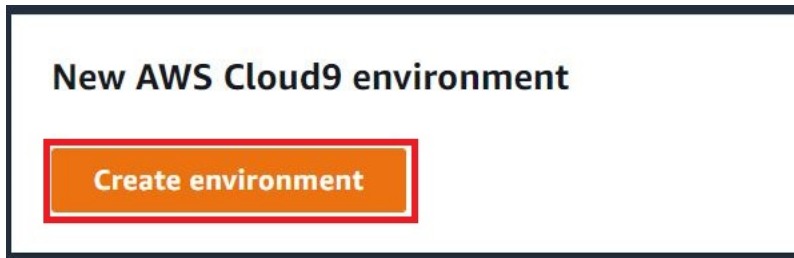
使用主控台建立 EC2 環境

- 登入 AWS Cloud9 主控台。
 - 如果您是唯一使用 AWS 帳戶的使用者，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶管理員提供登入指示。
 - 如果您是課堂裡的學生，請洽詢講師以取得登入指示。
- 在頂端導覽列中登入 AWS Cloud9 主控台後，請選擇 AWS 區域來建立環境。如需可用的清單 AWS 區域，請參閱[AWS Cloud9](#)中的AWS 一般參考。



3. 在其中一個顯示的位置選擇放大的 Create environment (建立環境) 按鈕。

如果您還沒有 AWS Cloud9 環境，按鈕會顯示在歡迎頁面上。



如果您已經有 AWS Cloud9 環境，按鈕會顯示如下。



4. 在 Create environment (建立環境) 頁面的 Name (名稱) 上，輸入環境的名稱。
5. 若要為您的環境新增說明，請輸入至 Description (說明) 欄位。
6. 在 Environment type (環境類型) 中，選擇 New EC2 instance (新的 EC2 執行個體)，建立 Amazon EC2 環境：
 - EC2 instance (EC2 執行個體)—啟動 AWS Cloud9 可以直接透過 SSH 進行連線的新 Amazon EC2 執行個體。您可以使用 Systems Manager 與新的 Amazon EC2 執行個體互動，如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

- Existing compute (現有運算) – 啟動需要 SSH 登入詳細資訊的現有 Amazon EC2 執行個體，而 Amazon EC2 執行個體必須具有輸入安全群組規則。
- 如果您選取 Existing compute (現有運算) 選項，則會自動建立服務角色。您也可以設定畫面底部的備註中檢視服務角色名稱。

Note

使用現有運算的 Amazon EC2 執行個體建立的 AWS Cloud9 環境無法使用自動關機。

Warning

為您的環境建立 Amazon EC2 執行個體可能會產生 Amazon EC2 的 AWS 帳戶費用。使用 Systems Manager 管理 EC2 執行個體的連線不需要額外費用。

7. 對於 Instance type (執行個體類型)，請選擇具有您認為想要執行之任務類型所需 RAM 和 vCPU 數量的執行個體類型。

Warning

選擇內含更多 RAM 和 vCPU 的執行個體類型可能讓 AWS 帳戶產生額外的 Amazon EC2 費用。有關哪種執行個體類型適合您的工作負載的資訊，請參閱 [Amazon EC2 執行個體類型](#) 頁面。

8. 對於平台，選擇您想要的 Amazon EC2 實例的類型：Amazon Linux 2023，Amazon Linux 2 或 Ubuntu 22.04 LTS。AWS Cloud9 建立執行個體，然後將環境連接至該執行個體。

Important

我們建議您選擇適用於 EC2 環境的 Amazon Linux 2023 選項。Amazon Linux 2023 AMI 除了提供安全、穩定和高效能的運行時間環境之外，還包括到 2024 年的長期支持。如需詳細資訊，請參閱 [AL2023](#) 頁面。

9. 選擇 Timeout (逾時) 的時間。此選項會決定自動休眠之前 AWS Cloud9 處於非作用狀態的時間長度。當連接至環境的 IDE 的所有 web 瀏覽器執行個體皆已關閉時，AWS Cloud9 會按照指定的時間長度等候，然後關閉環境的 Amazon EC2 執行個體。

⚠ Warning

選擇更長的一段時間可能會導致您的 AWS 帳戶 產生更多費用。

10. 在 Network settings (網路設定) 面板上，從下列兩個選項中，選擇存取環境的方式：
 - AWS Systems Manager (SSM) – 此方法使用 SSM 存取環境，而不必開啟傳入連接埠。
 - Secure Shell (SSH) (安全通訊殼層 (SSH)) – 此方法使用 SSH 存取環境，且需開啟傳入連接埠。
11. 選擇 VPC Settings (VPC 設定) 以顯示您環境的 Amazon Virtual Private Cloud 和子網路。AWS Cloud9 使用 Amazon Virtual Private Cloud (Amazon VPC) 與新建立的 Amazon EC2 執行個體進行通訊。在本教學課程中，我們建議您不要變更預先選取的預設設定。使用預設設定時，AWS Cloud9 會嘗試在與新環境相同的 AWS 帳戶 和區域中使用預設 VPC 搭配其單一子網路。請根據 Amazon VPC 的設定方式，遵循以下其中一組說明操作。

如果您不確定要選擇何者，我們建立您直接跳到此程序的下一步驟。

如果您略過過去的網路設定 (進階) 並保留預先選取的預設設定，則會 AWS Cloud9 嘗試將預設 VPC 與其單一子網路搭配使用。AWS Cloud9 根據您選取的執行個體類型選擇子網路。這些項目皆位於與新環境相同的 AWS 帳戶和 AWS 區域中。

⚠ Important

若您選取 Existing compute (現有運算) 做為環境類型，您可以將執行個體啟動至公有或私有子網路中。

- 公有子網路：將網際網路閘道連接至子網路，以便允許執行個體 SSM Agent 與 Systems Manager 通訊。
- 私有子網路：建立 NAT 閘道以便讓執行個體與網際網路和其他 AWS 服務 通訊。

目前您還不能使用 [AWS 受管臨時憑證](#) 來允許 EC2 環境代表 AWS 實體 (例如 IAM 使用者) 存取 AWS 服務。

如需設定子網路的詳細資訊，請參閱 [AWS Cloud9 開發環境的 VPC 設定](#)。

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
否	—	—	—	<p>如果沒有 VPC 存在，請建立一個。</p> <p>若要在與新環境相同的 AWS 帳戶 和區域中建立 VPC，請選擇 Create new VPC (建立新的 VPC)，然後依照畫面上的指示進行。如需詳細資訊，請參閱建立 VPC 以及其他 VPC 資源。</p> <p>若要在有別於新環境的不同 AWS 帳戶 中建立 VPC，請參閱《Amazon VPC 使用者指南》中的使用共用 VPC。</p>


AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	是	是	是	<p>直接跳到此程序的下一個步驟。</p> <p>如果您跳過 Network settings (advanced) (網路設定 (進階)) 且並未變更預先選取的預設設定，AWS Cloud9 會嘗試在與新環境相同的帳戶和區域中使用預設 VPC 與其單一子網路搭配。</p>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	是	是	否	<p>如果預設 VPC 有多個子網路，請展開 Network settings (advanced) (網路設定 (進階))。針對 Subnet (子網路)，請選擇您希望 AWS Cloud9 在選取之預設 VPC 中使用的子網路。</p> <p>如果預設 VPC 沒有子網路，請建立一個。若要執行此操作，請選擇 Create new subnet (建立新的子網路)，然後依照螢幕上的指示進行。如需詳細資訊，請參閱 為 AWS Cloud9 建立子網路。</p>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	是	否	是	請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，請選擇您希望 AWS Cloud9 使用的 VPC。

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	是	否	否	<p>請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，請選擇您希望 AWS Cloud9 使用的 VPC。</p> <p>如果選擇的 VPC 有多個子網路，請展開 Network settings (advanced) (網路設定 (進階))。針對 Subnet (子網路)，請選擇您希望 AWS Cloud9 在所選擇之 VPC 中使用的子網路。</p> <p>如果選擇的 VPC 沒有子網路，請建立一個。若要執行此操作，請選擇 Create new subnet (建立新的子網路)，然後依照螢幕上</p>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
				<p>的指示進行。 如需詳細資訊，請參閱 為 AWS Cloud9 建立子網路。</p>
是	否	是	—	<p>AWS 帳戶 若不同於新環境的帳戶，則 AWS Cloud9 無法在其中使用預設 VPC。在這個清單中選擇不同的選項。</p>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶 和區域中？	該 VPC 是否為其 AWS 帳戶 的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	否	否	是	<p>請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，請選擇您希望 AWS Cloud9 使用的 VPC。</p> <div data-bbox="1273 779 1507 1430" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>VPC 必須位於與新環境相同的區域中，即使 VPC 位於不同的帳戶中也一樣。</p> </div>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶和區域中？	該 VPC 是否為其 AWS 帳戶的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
是	否	否	否	<p>請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，請選擇您希望 AWS Cloud9 使用的 VPC。</p> <p>針對 Subnet (子網路)，請選擇您希望 AWS Cloud9 在所選擇之 VPC 中使用的子網路。</p> <p>如果選擇的 VPC 沒有子網路，則若要在有別於新環境的不同 AWS 帳戶中建立 VPC 的子網路，請參閱《Amazon VPC 使用者指南》中的使用共用 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 及子網路</p> </div>

AWS 帳戶 可以存取 Amazon VPC 嗎？	該 VPC 是否位於與新環境相同的 AWS 帳戶和區域中？	該 VPC 是否為其 AWS 帳戶的預設 VPC？	該 VPC 是否包含單一子網路？	請遵循以下說明
				<p>必須位於與新環境相同的區域中，即使 VPC 及子網路位於不同的帳戶中也一樣。</p>

如需這些選擇的詳細資訊，請參閱[AWS Cloud9 開發環境的 VPC 設定](#)。

- 藉由為每個標籤提供一個 Key (索引鍵)和 Value (值)，最多新增 50 個標籤。選取 Add new tag (新增標籤) 來執行此操作。標籤將附加至 AWS Cloud9 環境做為資源標籤，並傳播至以下基礎資源：AWS CloudFormation 堆疊、Amazon EC2 執行個體，以及 Amazon EC2 安全群組。若要進一步了解標籤，請參閱 [IAM 使用者指南](#) 中的 [使用 AWS 資源標籤控制存取](#)，以及本指南中的 [進階資訊](#)。

Warning

如果您在建立這些標籤之後予以更新，這些變更不會傳播至基礎資源。如需詳細資訊，請參閱 [標籤](#) 相關進階資訊中的 [將標籤更新傳播至基礎資源](#)。

- 選擇 Create (建立) 以建立您的環境，然後系統會將您重新導向至首頁。若成功建立帳戶，AWS Cloud9 主控台上方會出現綠色閃爍列。您可以選取新環境，然後選擇 Open in Cloud9 (在 Cloud9 中開啟) 以啟動 IDE。

Delete

View details

 Open in Cloud9 

Create environment

若無法建立帳戶，AWS Cloud9 主控台上方會出現紅色閃爍列。您的 Web 瀏覽器、AWS 存取許可、執行個體或相關的網路問題皆可能導致帳戶無法建立。您可以在 [AWS Cloud9 疑難排解區段](#) 找到可行修正方法。

Note

AWS Cloud9 同時支援 IMDSv1 和 IMDSv2。我們建議採用 IMDSv2，因為與 IMDSv1 相比，它提供了更高的安全性層級。如需有關 IMDSv2 優點的詳細資訊，請參閱 [AWS 安全性部落格](#)。如需有關如何從 IMDSv1 轉換到 IMDSv2 的詳細資訊，請參閱《Linux 執行個體的 Amazon EC2 使用者指南》中的 [〈轉換為使用執行個體中繼資料服務版本 2〉](#)。

Note

如果您的環境使用代理來存取網際網路，您必須提供代理詳細資訊給 AWS Cloud9 以便安裝相依性。如需詳細資訊，請參閱 [安裝相依性失敗](#)。

使用程式碼建立環境

若要使用程式碼在 AWS Cloud9 中建立 EC2 環境，請呼叫 AWS Cloud9 的 建立 EC2 環境操作，如下所示。

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

AWS SDK for .NET	CreateEnvironmentEC2 請求, EC2 回應 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

如果您的環境使用代理來存取網際網路，您必須提供代理詳細資訊給 AWS Cloud9 以便安裝相依性。如需詳細資訊，請參閱[安裝相依性失敗](#)。

建立 SSH 環境

您可以使用 AWS Cloud9 主控台建立 AWS Cloud9 SSH 開發環境。您不能使用 CLI 建立 SSH 環境。

必要條件

- 先確定您已完成 [設定 AWS Cloud9](#) 中的步驟。您就能登入 AWS Cloud9 主控台並建立環境。
- 指定您希望要讓 AWS Cloud9 連線至環境的現有雲端運算執行個體 (例如您 AWS 帳戶 中的 Amazon EC2 執行個體)，或您自己的伺服器。
- 確保現有的執行個體或您自己的伺服器符合所有 [SSH 主機要求](#)。這些條件包括擁有 Python、Node.js 和其他已安裝元件的特定版本；在您想要 AWS Cloud9 於登入之後從中啟動的目錄上設定特定許可；以及設定任何相關聯的 Amazon Virtual Private Cloud。

建立 SSH 環境

1. 請確認您已完成前述必要條件。
2. 如果您尚未連線，請使用 SSH 用戶端連線到您的現有執行個體或自己的伺服器。這能確保您可以將必要的公有 SSH 金鑰值新增至執行個體或伺服器。此程序稍後會再進一步說明。

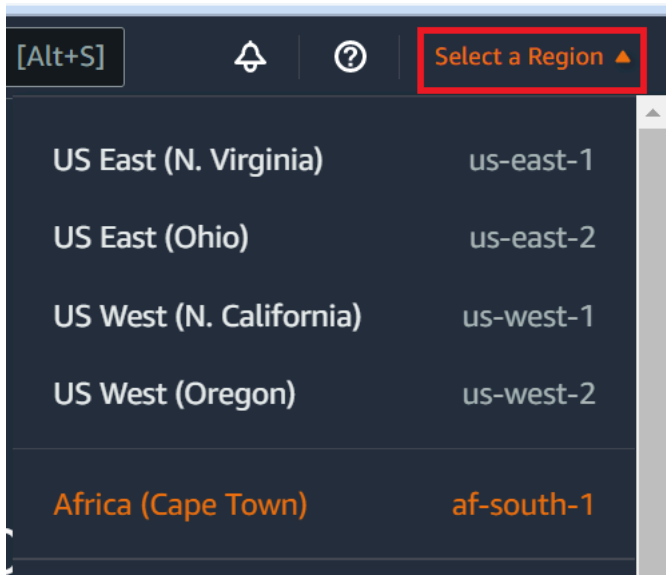
Note

若要連線到現有的 AWS 雲端 運算執行個體，請參閱下列一個或多個資源：

- 如為 Amazon EC2，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的[連線至 Linux 執行個體](#)。
- 針對 Amazon Lightsail，請參閱 [Amazon Lightsail 文件](#) 中的連線至您的 Linux/Unix 型 Lightsail 執行個體。
- 如為 AWS Elastic Beanstalk，請參閱 AWS Elastic Beanstalk 開發人員指南中的[列出並連線到伺服器執行個體](#)。
- 如為 AWS OpsWorks，請參閱 AWS OpsWorks 使用者指南中的[使用 SSH 登入 Linux 執行個體](#)。
- 如為其他 AWS 服務，請參閱特定服務的文件。

若要連線到您自己的伺服器，請使用 SSH。SSH 已安裝在 macOS 和 Linux 作業系統。若要在 Windows 上使用 SSH 連接伺服器，您必須安裝 [PuTTY](#)。

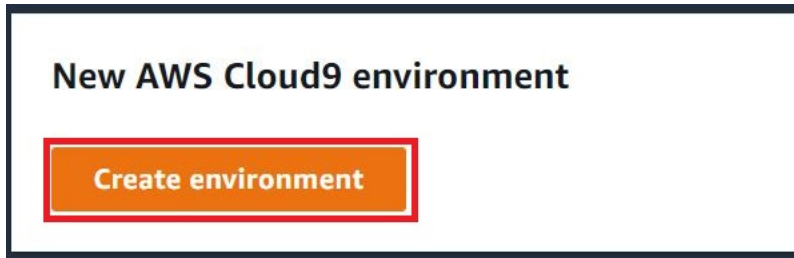
3. 登入 AWS Cloud9 主控台，網址為 <https://console.aws.amazon.com/cloud9/>。
4. 在頂端導覽列中登入 AWS Cloud9 主控台後，請選擇 AWS 區域 來建立環境。如需可用的清單 AWS 區域，請參閱[AWS Cloud9](#)中的AWS 一般參考。



5. 如果這是您第一次建立開發環境，畫面會顯示歡迎頁面。在新的 AWS Cloud9 環境面板中，選擇 Create environment (建立環境)。

如果您先前已建立開發環境，也可以展開畫面左側的窗格。選擇 Your environments (您的環境)，然後選擇 Create environment (建立環境)。

在 welcome (歡迎) 頁面中：



或者在 Your environments (您的環境) 頁面中：



- 在 Create environment (建立環境) 頁面上，輸入環境的名稱。
- 在 Description (描述) 中，輸入環境的相關資訊。在本教學中，使用 This environment is for the AWS Cloud9 tutorial.
- 針對 Environment type (環境類型)，請從下列選項中選擇 Existing Compute (現有運算)：
 - 新的 EC2 執行個體 — 啟動AWS Cloud9可直接透過 SSH 或 SSM 連線到的 Amazon EC2 執行個體。
 - 現有運算 — 啟動需要安全殼層登入詳細資訊以及連接埠 22 才能開啟的現有 Amazon EC2 執行個體。AWS Cloud9透過連線至執行個體[AWS Systems Manager](#)。
 - 如果您選取 Existing compute (現有運算) 選項，則會自動建立服務角色。您可以在介面下方的 [Systems Manager 存取] 區段中檢視服務角色和執行個體設定檔中的服務角色名稱。如需詳細資訊，請參閱[使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

⚠ Warning

為環境建立 EC2 執行個體可能會讓您的 AWS 帳戶產生 Amazon EC2 的費用。使用 Systems Manager 管理 EC2 執行個體的連線不需要額外費用。

⚠ Warning

AWS Cloud9 使用 SSH 公有金鑰安全地連接到您的伺服器。若要建立安全連線，請將公有金鑰新增到您的 `~/.ssh/authorized_keys` 檔案中，並提供以下步驟中的登入憑證。選擇 Copy key to clipboard (複製金鑰至剪貼簿) 以複製 SSH 金鑰，或選擇 View public SSH key (檢視公有 SSH 金鑰) 來顯示。

9. 在 Existing compute (現有運算) 面板上，針對 User (使用者)，輸入您稍早在此程序中，用來連線至執行個體或伺服器的登入名稱。例如，針對 AWS 雲端 運算執行個體，它可能是 `ec2-user`、`ubuntu` 或 `root`。

ℹ Note

建議將登入名稱與執行個體或伺服器上的管理許可或管理員使用者建立關聯。尤其，建議這個登入名稱擁有執行個體或伺服器上的 Node.js 安裝。若要加以確認，請從執行個體或伺服器的終端機中執行命令 `ls -l $(which node)` (如果您使用 `npm`，也可以執行 `ls -l $(npm which node)`)。此命令會顯示 Node.js 安裝的擁有者名稱。同時也會顯示安裝的許可、群組名稱和位置。

10. 針對 Host (主機)，輸入執行個體或伺服器的公有 IP 地址 (偏好) 或主機名稱。
11. 針對 Port (連接埠)，輸入您想要 AWS Cloud9 用來嘗試連線至執行個體或伺服器的連接埠。或者，也可以保留預設連接埠。
12. 選擇 Additional details - optional (其他詳細資訊 - 選用)，顯示環境路徑、node.js 二進位檔路徑和 SSH 跳接主機資訊。
13. 在 Environment path (環境路徑) 中輸入您要 AWS Cloud9 在執行個體或伺服器上啟動的目錄路徑。您會在此程序稍早的先決條件中指定此路徑。如果您將此空白，則 AWS Cloud9 會在登入之後使用您的執行個體或伺服器一般用來啟動的目錄。這通常是家用或預設目錄。
14. 針對 Path to Node.js binary path (node.js 二進位檔路徑)，請輸入路徑資訊，以指定執行個體或伺服器上的 Node.js 二進位檔路徑。若要取得路徑，您可以在執行個體或伺服器上執行命令 `which node` (如果您使用 `npm`，也可以執行 `npm which node`)。例如，路徑可能是 `/usr/bin/node`。如果您保留空白，則 AWS Cloud9 會在嘗試連線 Node.js 二進位檔時嘗試猜測其所在的位置。
15. 針對 SSH jump host (SSH 跳接主機)，請輸入執行個體或伺服器使用的跳接主機相關資訊。使用格式 `USER_NAME@HOSTNAME:PORT_NUMBER` (例如 `ec2-user@ip-192-0-2-0:22`)。

跳接主機必須符合下列要求：

- 它必須可使用 SSH 透過公有網際網路存取。
 - 它必須允許使用任何 IP 地址透過指定的連接埠傳入存取。
 - 複製至現有執行個體或伺服器上 `~/.ssh/authorized_keys` 檔案的公有 SSH 金鑰值，也必須複製至跳接主機上的 `~/.ssh/authorized_keys` 檔案。
 - 必須安裝 Netcat。
16. 藉由為每個標籤提供一個索引鍵和一個值，最多新增 50 個標籤。選取 Add new tag (新增標籤) 來執行此操作。標籤將附加至 AWS Cloud9 環境做為資源標籤，並傳播至以下基礎資源：AWS CloudFormation 堆疊、Amazon EC2 執行個體，以及 Amazon EC2 安全群組。若要進一步了解標籤，請參閱 [IAM 使用者指南](#) 中的 [使用 AWS 資源標籤控制存取](#)，以及本指南中有關標籤的 [進階資訊](#)。

Warning

如果您在建立這些標籤之後予以更新，這些變更不會傳播至基礎資源。如需詳細資訊，請參閱 [標籤相關進階資訊](#) 中的 [將標籤更新傳播至基礎資源](#)。

17. 選擇 Create (建立) 以建立您的環境，然後系統會將您重新導向至首頁。成功建立帳戶後，AWS Cloud9 主控台上方會出現綠色閃爍列。您可以選取新環境，然後選擇 Open in Cloud9 (在 Cloud9 中開啟) 以啟動 IDE。

Delete

View details

Open in Cloud9 

Create environment

若無法建立帳戶，AWS Cloud9 主控台上方會出現紅色閃爍列。您的 Web 瀏覽器、AWS 存取許可、執行個體或相關的網路問題皆可能導致帳戶無法建立。您可以在 [AWS Cloud9 疑難排解區段](#) 找到可行修正方法，解決造成帳戶無法使用的問題。

Note

如果您的環境使用代理來存取網際網路，您必須提供代理詳細資訊給 AWS Cloud9 以便安裝相依性。如需詳細資訊，請參閱 [安裝相依性失敗](#)。

使用 AWS Systems Manager 存取無輸入的 EC2 執行個體

為 EC2 環境建立的「無輸入 EC2 執行個體」可使 AWS Cloud9 與其 Amazon EC2 執行個體連線，無需在該執行個體上開啟任何傳入連接埠。建立 EC2 環境時，您可以使用[主控台](#)、[命令列界面](#)或 [AWS CloudFormation 堆疊](#) 選取無輸入選項。

Important

使用 Systems Manager Session Manager 來管理 EC2 執行個體的連線，不需要額外付費。

在主控台的 Create environment (建立環境) 頁面選取環境類型時，您可以選擇需要傳入連線的新 EC2 執行個體，或是新的無輸入 EC2 執行個體，其不需要下列項目：

- [New EC2 instance](#) (新的 EC2 執行個體) – 使用此設定時，執行個體的安全群組會有允許傳入聯網流量的規則。傳入網路流量受限於[已核准進行 AWS Cloud9 連線的 IP 地址](#)。開放的傳入連接埠可讓 AWS Cloud9 透過 SSH 連線到其執行個體。若您使用 AWS Systems Manager Session Manager，您可以透過 SSM 存取 Amazon EC2 執行個體，而無需開啟傳入連接埠 (無輸入)。此方法僅適用於新的 Amazon EC2 執行個體。如需詳細資訊，請參閱[為 EC2 環境使用 Systems Manager 的好處](#)。
- [Existing compute](#) (現有運算) – 透過此設定，可存取需要 SSH 登入詳細資訊的現有 Amazon EC2 執行個體，而 Amazon EC2 執行個體必須具有輸入安全群組規則。如果您選取此選項，則會自動建立服務角色。您也可以設定畫面底部的備註中檢視服務角色名稱。

如果使用 [AWS CLI](#) 建立環境，您可以設定無輸入 EC2 執行個體，方法是在呼叫 create-environment-ec2 命令時設定 --connection-type CONNECT_SSM 選項。如需詳細了解如何建立必要的服務角色和執行個體描述檔，請參閱[使用 AWS CLI 管理 Systems Manager 的執行個體描述檔](#)。

完成建立使用無輸入 EC2 執行個體的環境之後，請確認下列項目：

- Systems Manager Session Manager 具備代表您對 EC2 執行個體執行動作的許可。如需詳細資訊，請參閱[管理 Systems Manager 許可](#)。
- AWS Cloud9 使用者可以存取由 Session Manager 管理的執行個體。如需詳細資訊，請參閱[讓使用者存取由 Session Manager 管理的執行個體](#)。

為 EC2 環境使用 Systems Manager 的好處

允許 [Session Manager](#) 處理 AWS Cloud9 及其 EC2 執行個體之間的安全連線，可帶來兩項關鍵優勢：

- 不需要為執行個體開啟傳入連接埠
- 提供在公有或私有子網路中啟動執行個體的選項

No open inbound ports

AWS Cloud9 和其 EC2 執行個體之間的安全連線是由 [Session Manager](#) 負責處理。Session Manager 是全受管的 Systems Manager 功能，可讓 AWS Cloud9 無需開啟傳入連接埠即可連線到 EC2 執行個體。

Important

目前只有在建立新的 EC2 環境時，才能使用 Systems Manager 進行無輸入連線的選項。

Session Manager 工作階段啟動後，就會與目標執行個體建立連線。現在連線完成後，環境就可以透過 Systems Manager 服務與執行個體互動。Systems Manager 服務會透過 Systems Manager Agent ([SSM Agent](#)) 與執行個體通訊。



根據預設，SSM Agent 會安裝在 EC2 環境使用的所有執行個體上。

Private/public subnets



在 Network settings (advanced) (網路設定 (進階)) 區塊中為執行個體選取子網路時，如果您是透過 Systems Manager 存取環境的執行個體，則可以選取私有或公有子網路。


▼ **Network settings (advanced)**

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

vpc- [redacted] ▼   **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) ▼   **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.

私有子網路

對於私有子網路，請確保執行個體仍可連線至 SSM 服務。[在公有子網路中設定 NAT 閘道](#)或[為 Systems Manager 設定 VPC 端點](#)即可確保這個要求。

使用 NAT 閘道的優點在於它可以防止網際網路在私有子網路中啟動與執行個體的連線。您環境的執行個體會獲指派私有 IP 地址，而非公用 IP 地址。因此，NAT 閘道會將流量從執行個體轉送至網際網路或其他 AWS 服務，然後將回應送回執行個體。

對於 VPC 選項，請為 Systems Manager 建立至少三個必要的界面端點：com.amazonaws.region.ssm、com.amazonaws.region.ec2messages 和 com.amazonaws.region.ssmessages。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的[為 Systems Manager 建立 VPC 端點](#)。

Important

目前，如果您環境的 EC2 執行個體已在私有子網路中啟動，您就無法使用 [AWS 受管臨時憑證](#)來允許 EC2 環境代表 AWS 實體 (例如 IAM 使用者) 存取 AWS 服務。

公有子網路

如果您的開發環境使用 SSM 存取 EC2 執行個體，請確保該執行個體已由公有子網路指派了可在子網路中啟動的公有 IP 地址。若要這麼做，您可以指定自己的 IP 地址或啟用公有 IP 地址的自動指派功能。如需了解修改自動指派 IP 設定的步驟，請參閱 Amazon VPC 使用者指南中的[設定 VPC 中的 IP 地址](#)。

如需為環境執行個體設定私有和公有子網路的詳細資訊，請參閱 [為 AWS Cloud9 建立子網路](#)。

管理 Systems Manager 許可

在預設情況下，Systems Manager 沒有在您的 EC2 執行個體上執行動作的許可。存取權需透過 AWS Identity and Access Management (IAM) 執行個體描述檔取得。(執行個體描述檔是在啟動時將 IAM 角色資訊傳遞到 EC2 執行個體的容器。)

如果您使用 AWS Cloud9 主控台建立無輸入 EC2 執行個體，服務角色 (AWSCloud9SSMAccessRole) 和 IAM 執行個體描述檔 (AWSCloud9SSMInstanceProfile) 都會自動建立。(您可以在 IAM 管理主控台檢視 AWSCloud9SSMAccessRole。執行個體描述檔不會顯示在 IAM 主控台中。)

Important

如果您第一次使用 AWS CLI 建立無輸入 EC2 環境，您必須明確定義必要的服務角色和執行個體描述檔。如需詳細資訊，請參閱[使用 AWS CLI 管理 Systems Manager 的執行個體描述檔](#)。

Important

如果您正在建立 AWS Cloud9 環境和使用亞馬遜 EC2 系統管理器與 AWSCloud9Administrator 或者 AWSCloud9User 附加政策時，您還必須附加具有特定 IAM 許可的自訂政策，請參閱[建立 SSM 環境的自訂 IAM 政策](#)。這是由於權限問題 AWSCloud9Administrator 和 AWSCloud9User 政策。

為了獲得額外的安全保護，AWS Cloud9 服務連結的角色 AWSServiceRoleforAWSCloud9 會在其 AWSCloud9ServiceRolePolicy 政策中設定 PassRole 限制。當您傳遞 IAM 角色給某個服務時，該角色會允許該服務接受角色，並代表您執行動作。這種情況下，PassRole 許可會確保 AWS Cloud9 只能傳遞 AWSCloud9SSMAccessRole 角色 (及其許可) 給 EC2 執行個體。這麼做會將可在 EC2 執行個體上執行的動作限制為僅限 AWS Cloud9 要求的動作。

Note

如果您不再需要使用 Systems Manager 存取執行個體，您可以刪除 AWSCloud9SSMAccessRole 服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的[刪除角色或執行個體描述檔](#)。

使用 AWS CLI 管理 Systems Manager 的執行個體描述檔

您也可以使用 AWS CLI 建立無輸入的 EC2 環境。當您呼叫 `create-environment-ec2` 時，需在 `CONNECT_SSM` 設定 `--connection-type` 選項。

如果您使用此選項，AWSCloud9SSMAccessRole 服務角色和 AWSCloud9SSMInstanceProfile 不會自動建立。因此，若要建立必要的服務設定檔和執行個體設定檔，請執行下列其中一項操作：

- 在 AWSCloud9SSMAccessRole 服務角色和 AWSCloud9SSMInstanceProfile 自動建立之後使用主控台建立 EC2 環境。環境建立後，服務角色和執行個體描述檔就可以供任何其他透過 AWS CLI 建立的 EC2 環境使用。
- 執行下列 AWS CLI 命令來建立服務角色和執行個體描述檔。

```
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
```

讓使用者存取由 Session Manager 管理的執行個體

若要透過 Systems Manager 開啟連線到 EC2 執行個體的 AWS Cloud9 環境，使用者必須具有 API 操作 `StartSession` 的許可。此操作會啟動與受管 EC2 執行個體的連線，執行 Session Manager 工作階段。為使用者提供存取權時，建議您使用 AWS Cloud9 特定的受管政策，或者您也可以編輯 IAM 政策並新增必要的許可。

方法	描述
使用 AWS Cloud9 特定的受管政策	<p>建議您使用 AWS 受管政策來允許使用者存取由 Systems Manager 管理的 EC2 執行個體。受管政策可提供一組適用於標準 AWS Cloud9 使用案例的許可，並且可以輕鬆連接到 IAM 實體。</p> <p>所有受管政策也都包含執行 StartSession API 操作的許可。以下是 AWS Cloud9 特定的受管政策：</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 1508 1671" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p>⚠ Important</p><p>如果您正在建立AWS Cloud9環境和使用亞馬遜 EC2 系統管理器與AWSCloud9Administrator 或者AWSCloud9User 附加政策時，您還必須附加具有特定 IAM 許可的自訂政策，請參閱建立 SSM 環境的自訂 IAM 政策。這是由於權限問題AWSCloud9 Administrator 和AWSCloud9 User 政策。</p></div> <p>如需詳細資訊，請參閱AWS 受管理的政策 AWS Cloud9。</p>

方法	描述
編輯 IAM 政策並新增必要政策陳述式	<p>若要編輯現有政策，您可以為 StartSession API 新增許可。若要使用 AWS Management Console 或 AWS CLI 編輯政策，請遵循《IAM 使用者指南》中編輯 IAM 政策所提供的指示操作。</p> <p>編輯政策時，請新增 policy statement (請參閱以下內容) 來允許 ssm:startSession API 操作執行。</p>

您可以使用下列許可來執行 StartSession API 操作。ssm:resourceTag 條件索引鍵會指定 Session Manager 工作階段可由任何執行個體 (Resource: arn:aws:ec2:*:*:instance/*) 啟動，但條件是執行個體是 AWS Cloud9 EC2 開發環境 (aws:cloud9:environment)。

Note

下列受管政策也包含這些政策陳述式：AWSCloud9Administrator、AWSCloud9User 及 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 建立無輸入 EC2 環境

當使用 [AWS CloudFormation 範本](#) 定義無輸入的 Amazon EC2 開發環境時，請在建立堆疊之前執行下列動作：

1. 建立 AWSCloud9SSMAccessRole 服務角色和 AWSCloud9SSMInstanceProfile 執行個體描述檔。如需詳細資訊，請參閱[使用 AWS CloudFormation 範本建立服務角色和執行個體描述檔](#)。
2. 為呼叫 AWS CloudFormation 的 IAM 實體更新政策。如此，該實體就能啟動連線到 EC2 執行個體的 Session Manager 工作階段。如需詳細資訊，請參閱[將 Systems Manager 許可新增至 IAM 政策](#)。

使用 AWS CloudFormation 範本建立服務角色和執行個體描述檔

您需要建立服務角色 AWSCloud9SSMAccessRole 和執行個體描述檔 AWSCloud9SSMInstanceProfile，讓 Systems Manager 能夠管理支援開發環境的 EC2 執行個體。

如果您之前已建立無輸入的 EC2 環境 [with the console](#) 或 [執行 AWS CLI 命令](#) 而建立了 AWSCloud9SSMAccessRole 和 AWSCloud9SSMInstanceProfile，該服務角色和執行個體設定檔已可供使用。

Note

假設您嘗試為無輸入 EC2 環境建立 AWS CloudFormation 堆疊，但您沒有先建立必要的服務角色和執行個體設定檔。然後，不會建立堆疊，並顯示下列錯誤訊息：
例項設定檔 AWSCloud9SSMInstanceProfile 帳戶中不存在。

第一次使用 AWS CloudFormation 建立無輸入 EC2 環境時，您可以將 AWSCloud9SSMAccessRole 和 AWSCloud9SSMInstanceProfile 定義為範本中的 IAM 資源。

這項來自範例範本的摘要會顯示如何定義這些資源。AssumeRole 動作會傳回安全憑證，提供 AWS Cloud9 環境及其 EC2 執行個體的存取權。

```
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
```

將 Systems Manager 許可新增至 IAM 政策

在 [AWS CloudFormation 範本](#) 中 [定義服務角色和執行個體設定檔](#) 之後，請確保建立堆疊的 IAM 實體具有啟動 Session Manager 工作階段的許可。工作階段是指使用 Session Manager 建立的 EC2 執行個體連線。

Note

如果您在為無輸入 EC2 環境建立堆疊之前，並未新增啟動 Session Manager 工作階段的許可，系統就會傳回 AccessDeniedException 錯誤。

為呼叫 AWS CloudFormation 的 IAM 實體新增以下許可到政策中。

```
{
    "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/*"
    ]
}
```

將 Amazon S3 的 VPC 端點設為下載相依性

如果您 AWS Cloud9 環境的 EC2 執行個體無法存取網際網路，請為指定的 Amazon S3 儲存貯體建立 VPC 端點。此儲存桶包含保留 IDE 所需的依賴關係 up-to-date。

為 Amazon S3 設定 VPC 端點也涉及自訂存取政策。您想要存取政策，只在受信任的 S3 儲存貯體包含要下載的相依性時，才允許存取該儲存貯體。

Note

您可使用 AWS Management Console、AWS CLI 或 Amazon VPC API 來建立與設定 VPC。下列程序說明如何使用主控台界面來建立 VPC 端點。

為 Amazon S3 建立和設定 VPC 端點

1. 在 AWS Management Console 中，前往 Amazon VPC 的主控台頁面。

2. 在導覽列中選擇 Endpoints (端點)。
3. 在 Endpoints (端點) 頁面，選擇 Create Endpoint (建立端點)。
4. 在 Create Endpoint (建立端點)頁面的搜尋欄位中輸入「s3」，然後按下 Return (傳回) 以列出目前的 AWS 區域中可用的 Amazon S3 的端點。
5. 從傳回的 Amazon S3 端點清單中，選取 Gateway (閘道) 類型。
6. 接下來，選擇包含您環境的 EC2 執行個體的 VPC。
7. 現在選擇 VPC 的路由表。如此，相關聯的子網路可以存取端點。您環境的 EC2 執行個體位於其中一個子網路中。
8. 在 Policy (政策) 中，選擇 Custom (自訂) 選項，並以下列設定取代標準政策。

```
{
  "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/*"
    }
  ]
}
```

針對 Resource 元素，以您的 AWS 區域中可用的儲存貯體實際的名稱取代 {bucket_name}。舉例來說，如果您在歐洲 (愛爾蘭) 區域使用 AWS Cloud9，您可指定下列內容："Resource": "arn:aws:s3:::static-eu-west-1-prod-static-hld3vzaf7c4h/content/dependencies/。

下表列出含有可用 AWS Cloud9 的 AWS 區域中儲存貯體的名稱。

AWS Cloud9 區域中的 Amazon S3 儲存貯體

AWS 區域	儲存貯體名稱
美國東部 (俄亥俄)	static-us-east-2-prod-static-1c3sfcvf9hy4m

AWS 區域	儲存貯體名稱
美國東部 (維吉尼亞北部)	static-us-east-1-prod-static-mft1k1nkc4h1
美國西部 (奧勒岡)	static-us-west-2-prod-static-p21mksqx9zlr
美國西部 (加利佛尼亞北部)	static-us-west-1-prod-static-16d59zrrp01z0
非洲 (開普敦)	static-af-south-1-prod-static-v6v7i5ypdppv
亞太區域 (香港)	static-ap-east-1-prod-static-171xhpfkrorh6
亞太區域 (孟買)	static-ap-south-1-prod-static-ykocre202i9d
亞太區域 (大阪)	static-ap-northeast-3-prod-static-ivmxqzrx2ioi
亞太區域 (首爾)	static-ap-northeast-2-prod-static-1wxyctlhwiajm
亞太區域 (新加坡)	static-ap-southeast-1-prod-static-13ibpyrx4vk6d
亞太區域 (雪梨)	static-ap-southeast-2-prod-static-1cjsl8bx27rfu
亞太區域 (東京)	static-ap-northeast-1-prod-static-4fwvdisquj8
加拿大 (中部)	static-ca-central-1-prod-static-g80lpejy486c
歐洲 (法蘭克福)	static-eu-central-1-prod-static-14lbgls2vrkh

AWS 區域	儲存貯體名稱
歐洲 (愛爾蘭)	static-eu-west-1-prod-static-h1d3vzaf7c4h
歐洲 (倫敦)	static-eu-west-2-prod-static-361bg202837x
歐洲 (米蘭)	static-eu-south-1-prod-static-1379tzkd3ni7d
歐洲 (巴黎)	static-eu-west-3-prod-static-1rwpkf766ke58
歐洲 (斯德哥爾摩)	static-eu-north-1-prod-static-1qzw982y7yu7e
中東 (巴林)	static-me-south-1-prod-static-gmljex38qtqx
南美洲 (聖保羅)	static-sa-east-1-prod-static-1cl8k0y7opidt
以色列 (特拉維夫)	static-il-central-1-prod-static-k02vrnhcesue

9. 選擇 Create Endpoint (建立端點)。

如果您已提供正確的組態資訊，訊息會顯示已建立的端點的 ID。

10. 若要檢查您的 IDE 是否可以存取 Amazon S3 儲存貯體，請在選單列上選擇 Window (視窗)、New Terminal (新增終端機)，以便啟動終端機工作階段。接著執行以下命令，將 {bucket_name} 取代之為您區域中儲存貯體的實際名稱。

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

例如，如果您在美國東部 (維吉尼亞北部) 區域建立 S3 儲存貯體的端點，請執行下列命令。

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

如果 Ping 得到回應，則您可確認 IDE 可存取儲存貯體及其相依性。

如需此功能的詳細資訊，請參閱[亞馬遜 S3 的端點](#)在 AWS PrivateLink 指南。

將 VPC 端點設為使用私有連線

如果您使用 access using Systems Manager (使用 Systems Manager 存取) 選項在子網路中啟動執行個體，該執行個體的安全群組不含允許傳入網路流量的傳入規則。但是，安全群組卻有允許從執行個體傳出流量的傳出規則。若要下載將 AWS Cloud9 IDE 維持在最新狀態的套件和資料庫，這是必要的條件。

為了防止執行個體的傳出和傳入流量，請為 Systems Manager 建立和設定 Amazon VPC 端點。使用界面 VPC 端點 (介面端點)，您可以連線至採用 [AWS PrivateLink](#) 技術的服務。AWS PrivateLink 技術可以用來使用私有 IP 地址，以私密方式存取 Amazon EC2 和 Systems Manager API。若要將 VPC 端點設為使用 Systems Manager，請遵循此[知識中心資源](#)中的指示操作。

Warning

假設您可以設定一個不允許傳入或傳出網路流量的安全群組。然後，支援 AWS Cloud9 IDE 的 EC2 執行個體無法存取網際網路。您需要建立 [VPC 的 Amazon S3 端點](#) 才能允許存取受信任的 S3 儲存貯體中包含的相依性。此外，某些 AWS 服務 (例如 AWS Lambda) 如果沒有網際網路存取權，可能就無法如期運作。

使用 AWS PrivateLink 時，透過 VPC 端點處理的每 GB 資料都會產生處理費用。這與流量的來源或目的地無關。如需詳細資訊，請參閱 [AWS PrivateLink 定價](#)。

在 AWS Cloud9 中開啟環境

此程序說明如何在 AWS Cloud9 中開啟環境。

Note

此程序假設您已建立 AWS Cloud9 開發環境。若要建立環境，請參閱[建立環境](#)。

1. 如下所示登入 AWS Cloud9 主控台：

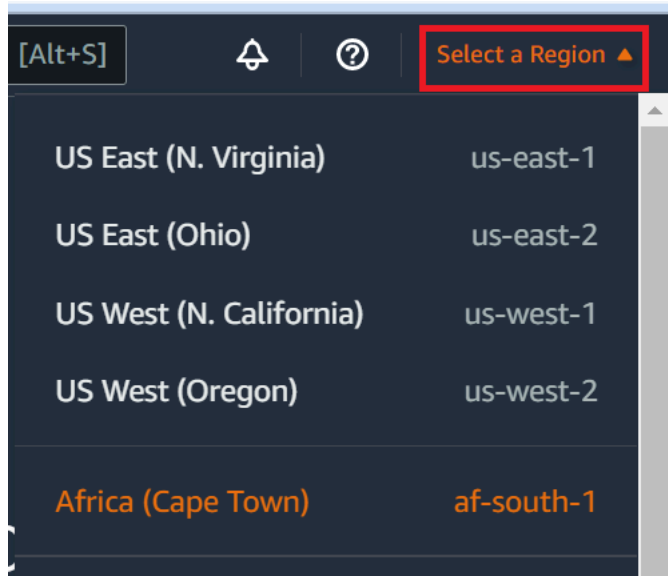
- 如果您是唯一使用 AWS 帳戶的使用者，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。

- 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶 管理員提供登入指示。

⚠ Important

如果您登出 [AWS 帳戶](#) 帳戶，之後 AWS Cloud9 IDE 仍可供存取最多 5 分鐘。當所需的許可到期時，系統會拒絕存取。

2. 從頂端導覽列選擇環境所在的 AWS 區域。



3. 在環境清單中，針對您希望開啟的環境執行下列其中一項動作。
 - 在卡片之內，選擇 Open in Cloud9 (在 Cloud9 中開啟) 連結。
 - 選取卡片，然後選擇 Open in Cloud9 (在 Cloud9 中開啟) 按鈕。



如果您的環境沒有顯示在主控台中，請嘗試執行下列一或多項動作將其顯示。

- 在 Environments (環境) 頁面的下拉式功能表列中，選擇下列一或多個選項。
 - 選擇 My environments (我的環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內擁有的所有環境。
 - 選擇 Shared with me (共用環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內已受邀加入的所有環境。

- 選擇 All account environments (所有帳戶環境) 以顯示您的 AWS 區域 實體在所選 AWS 帳戶及 AWS 內已獲許可顯示的所有環境。
- 若您認為自己是某個環境的成員，但 Shared with you (共用環境) 清單中並未顯示該環境，請向環境擁有者洽詢。
- 在頂端導覽列中，選擇不同的 AWS 區域。

從 AWS Cloud9 的環境呼叫 AWS 服務

您可以從 AWS Cloud9 開發環境呼叫 AWS 服務。例如，您可以執行下列動作：

- 上傳和下載 Amazon Simple Storage Service (Amazon S3) 儲存貯體中的資料。
- 透過 Amazon Simple Notification Service (Amazon SNS) 主題傳送廣播通知。
- 讀取和寫入 Amazon DynamoDB (DynamoDB) 資料庫中的資料。

您可以透過幾種方式從您的環境呼叫 AWS 服務。例如，您可以從終端機工作階段使用 AWS Command Line Interface (AWS CLI) 或 AWS CloudShell 執行命令。您也可以從環境內執行的程式碼呼叫 AWS 服務。您可以使用 JavaScript、Python、Ruby、PHP、Go、和 C++ 等程式設計語言的 AWS SDK 執行此操作。如需詳細資訊，請參閱 [AWS CLI 和 aws-shell 範例](#)、[AWS Command Line Interface 使用者指南](#) 及 [AWS 開發套件](#)。

每次 AWS CLI、AWS CloudShell 或您的程式碼呼叫 AWS 服務，AWS CLI、AWS CloudShell 或您的程式碼都必須隨著呼叫附帶提供一組 AWS 存取憑證。這些登入資料將判斷呼叫端是否具備適當許可，能夠發出該次呼叫。如果憑證當中不包含適當許可，呼叫即會失敗。

為您的環境提供憑證有好幾種方法。下表說明其中一些方法。

環境類型	方法
EC2	<p>使用 AWS 受管臨時憑證</p> <p>我們建議針對 EC2 環境使用此方法。AWS 受管臨時憑證會代您管理 EC2 環境中的 AWS 存取憑證，同時也會遵循 AWS 安全最佳實務。</p> <p>如果您使用的是 EC2 環境，您可以略過此主題其餘內容。這是因為您的環境已經設定 AWS 受管臨時憑證。</p>

環境類型	方法
	<p>如需詳細資訊，請參閱 AWS 受管臨時憑證。</p>
EC2	<p>將 IAM 執行個體描述檔連接至執行個體。</p> <p>只在您由於某些因素無法使用 AWS 受管臨時憑證時，才會使用此方法。與 AWS 受管臨時憑證類似，執行個體描述檔會代您管理 AWS 存取憑證。不過，您必須自行建立和管理執行個體描述檔並將其連接至 Amazon EC2 執行個體。</p> <p>如需相關指示，請參閱「建立及使用執行個體描述檔來管理暫時登入資料」。</p>
EC2 或 SSH	<p>將您的永久 AWS 存取憑證存放在環境中。</p> <p>此方法比起使用臨時 AWS 存取憑證較不安全。不過，這是 SSH 環境唯一支援的方法。</p> <p>如需說明，請參閱「在環境中建立並存放永久存取登入資料」。</p>
EC2 或 SSH	<p>將您的永久 AWS 存取憑證直接插入程式碼。</p> <p>此方法並未遵循 AWS 安全最佳實務，因此不建議採用。</p> <p>由於不建議採用此方法，本主題將不就其做介紹。</p>

建立及使用執行個體描述檔來管理臨時憑證

Note

此程序並不適用於 AWS Cloud9 SSH 開發環境。而應跳至[在環境中建立並存放永久存取登入資料](#)。

建議您使用 AWS 受管臨時憑證，而非執行個體描述檔。若您由於某些因素無法使用 AWS 受管臨時憑證，才應依照以下指示操作。如需詳細資訊，請參閱 [AWS 受管臨時憑證](#)。

此程序會使用 IAM 和 Amazon EC2 建立 IAM 執行個體描述檔，並將其連接至與您的環境連線的 Amazon EC2 執行個體。此執行個體描述檔會代您管理臨時憑證。此程序假設您已在 AWS Cloud9 建立環境。若要建立環境，請參閱 [建立環境](#)。

您可以使用 [IAM 及 Amazon EC2 主控台](#) 或 [AWS 命令列界面 \(AWS CLI\)](#) 完成上述作業。

使用 IAM 主控台建立執行個體描述檔

Note

若您已有包含了執行個體描述檔的 IAM 角色，請跳至 [使用 Amazon EC2 主控台將執行個體描述檔連接至執行個體](#)。

1. 登入 IAM 主控台，網址為 <https://console.aws.amazon.com/iam/>。

對於此步驟，建議您在 AWS 帳戶中使用管理員層級憑證來登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

2. 在導覽列中，選擇 Roles (角色)。

Note

您無法使用 IAM 主控台單獨建立執行個體描述檔，您必須建立一個包含執行個體描述檔的 IAM 角色。

3. 選擇 建立角色。
4. 在已選擇 AWS 服務的 Select type of trusted entity (選取信任的實體類型) 頁面上，針對 Choose the service that will use this role (選擇將使用此角色的服務) 選擇 EC2。
5. 從 Select your use case (選取您的使用案例) 選擇 EC2。
6. 選擇 Next: Permissions (下一步：許可)。
7. 在 [附加權限原則] 頁面的原則清單中，選取旁邊的核取方塊 AdministratorAccess，然後選擇 [下一步：檢閱]。

Note

AdministratorAccess 策略允許無限制存取您的所有 AWS 動作和資源 AWS 帳戶。使用它僅供試驗之用。如需詳細資訊，請參閱 IAM 使用者指南中的 [IAM 政策](#)。

- 在 Review (檢閱) 頁面的 Role Name (角色名稱) 中，輸入角色的名稱 (例如 my-demo-cloud9-instance-profile)。
- 選擇 Create Role (建立角色)。

跳至 [使用 Amazon EC2 主控台將執行個體描述檔連接至執行個體](#)。

使用 AWS CLI 建立執行個體描述檔

Note

若您已有包含了執行個體描述檔的 IAM 角色，請跳至 [使用 AWS CLI 將執行個體描述檔連接至執行個體](#)。

對於本主題，建議您在 AWS 帳戶使用管理員層級憑證來設定 AWS CLI。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

- 在 AWS 中為執行個體描述檔所需的 IAM 角色定義信任關係。請建立含有以下內容的檔案，然後儲存該檔案 (例如，命名為 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. 使用終端機或命令提示字元，切換到您剛儲存該檔案的所在目錄。
3. 為執行個體描述檔建立 IAM 角色。若要這麼做，請執行 IAM `create-role` 命令。當您這樣做時，請指定新 IAM 角色的名稱 (例如，`my-demo-cloud9-instance-profile-role`)，以及您剛儲存的檔案名稱。

```
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. 將 AWS 存取許可連接至執行個體描述檔的 IAM 角色。若要這麼做，請執行 IAM `attach-role-policy` 命令。指定現有 IAM 角色的名稱，以及 AWS 受管政策 (名為 `AdministratorAccess`) 的 Amazon Resource Name (ARN)。

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

Note

政策 `AdministratorAccess` 允許無限制存取您的所有 AWS 動作和資源 AWS 帳戶。使用它僅供試驗之用。如需詳細資訊，請參閱 IAM 使用者指南中的 [IAM 政策](#)。

5. 建立執行個體描述檔。方法如下：執行 IAM `create-instance-profile` 命令，指定新的執行個體描述檔的名稱 (例如 `my-demo-cloud9-instance-profile`)。

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

6. 將 IAM 角色連接至執行個體描述檔。方法如下：執行 IAM `add-role-to-instance-profile`，指定現有 IAM 角色以及執行個體描述檔的名稱。

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

跳至[使用 AWS CLI 建立執行個體描述檔](#)。

使用 Amazon EC2 主控台將執行個體描述檔連接至執行個體

1. 請登入 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。

對於此步驟，建議您在 AWS 帳戶使用管理員層級憑證來登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

2. 在導覽列中，確認區域選擇器所顯示的與您環境的 AWS 區域相符。例如，若您在美國東部 (俄亥俄) 區域建立了環境，請在這裡選擇區域選擇器中的 US East (Ohio) (美國東部 (俄亥俄))。
3. 選擇 Running Instances (運作中的執行個體) 連結，或從導覽窗格中展開 Instances (執行個體)，然後選擇 Instances (執行個體)。
4. 從執行個體清單中，選擇 Name (名稱) 中包含您環境名稱的執行個體。例如，如果您的環境名稱是 my-demo-environment，請選擇名稱包含的執行個體 my-demo-environment。
5. 選擇 Actions (動作)、Security (安全性)、Modify IAM role (修改 IAM 角色)。

Note

本操作雖是將角色連接至執行個體，但該角色包含了執行個體描述檔。

6. 在 Modify IAM Role (修改 IAM 角色) 頁面上，對於 IAM role (IAM 角色)，選擇您在先前程序識別或建立的角色的名稱，然後選擇 Apply (套用)。
7. 回到環境中，使用 AWS CLI 執行 `aws configure` 命令，或者使用 AWS CloudShell 執行 `configure` 命令。請勿對 AWS Access Key ID (AWS 存取金鑰 ID) 或 AWS Secret Access Key (AWS 私密存取金鑰) 指定任何值 (每次出現提示後直接按 Enter)。針對 Default Region name (預設區域名稱)，指定離您最近的 AWS 區域或您的 AWS 資源所在的區域。例如，美國東部 (俄亥俄) 區域請指定 us-east-2。如需「區域」的清單，請參閱中的 [AWS 區域和「端點」](#) Amazon Web Services 一般參考。選擇性指定預設輸出格式的值 (例如 json)。

您現在即可開始從您的環境呼叫 AWS 服務。若要使用 AWS CLI、aws-shell 或兩者呼叫 AWS 服務，請參閱 [AWS CLI 和 aws-shell 範例](#)。若要從程式碼呼叫 AWS 服務，請參閱我們其他的 [教學和範例](#)。

使用 AWS CLI 將執行個體描述檔連接至執行個體

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

1. 執行 Amazon EC2 `associate-iam-instance-profile` 命令。為環境指定執行個體設定檔的名稱，以及 Amazon EC2 執行個體的 ID 和 AWS 區域 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，將 `us-east-2` 取代為執行個體的 AWS 區域 ID，將 `i-12a3b45678cdef9a0` 取代為執行個體的 ID。

若要取得執行個體的 ID，您可以依照本例執行 Amazon EC2 `describe-instances` 命令，指定環境的名稱和 AWS 區域 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，將 `us-east-2` 取代為執行個體的 AWS 區域 ID，將 `my-environment` 取代為環境的名稱。

2. 回到環境中，使用 AWS CLI 執行 `aws configure` 命令，或者使用 `aws-shell` 執行 `configure` 命令。請勿為 AWS Access Key ID (AWS 存取金鑰 ID) 或 AWS Secret Access Key (AWS 私密存取金鑰) 指定任何值。在其中每個提示之後按下 Enter。針對 Default Region name (預設區域名稱)，指定離您最近的 AWS 區域或您的 AWS 資源所在的區域。例如，美國東部 (俄亥俄) 區域請指定 `us-east-2`。如需區域的清單，請參閱中的 [AWS 區域和端點 Amazon Web Services 一般參考](#)。選擇性指定預設輸出格式的值 (例如 `json`)。

您現在即可開始從您的環境呼叫 AWS 服務。若要使用 AWS CLI、`aws-shell` 或兩者呼叫 AWS 服務，請參閱 [AWS CLI 和 `aws-shell` 範例](#)。若要從程式碼呼叫 AWS 服務，請參閱我們其他的 [教學和範例](#)。

在環境中建立並存放永久存取憑證

Note

如果您使用 AWS Cloud9 EC2 開發環境，建議您使用 AWS 受管臨時憑證，而不是 AWS 永久存取憑證。若要使用 AWS 受管臨時憑證，請參閱 [AWS 受管理的臨時憑](#)。

在本節中，您使用 AWS Identity and Access Management (IAM) 產生一組永久憑證。AWS CLI、aws-shell 或您的程式碼可以在呼叫 AWS 服務時使用這組憑證。該組憑證包括了您 AWS 帳戶中的使用者獨有的 AWS 存取金鑰 ID 和 AWS 私密存取金鑰。若您已有 AWS 存取金鑰 ID 和 AWS 私密存取金鑰，請記下這些憑證，然後跳至[在環境中存放永久存取憑證](#)。

您可以使用 [IAM 主控台](#) 或 [AWS CLI](#) 建立一組永久憑證。

授予程式設計存取權

若使用者想要與 AWS Management Console 之外的 AWS 互動，則需要程式設計存取權。授予程式設計存取權的方式取決於存取 AWS 的使用者類型。

若要授予使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	若要	By
人力身分 (IAM Identity Center 中管理的使用者)	使用臨時憑證簽署對 AWS CLI、AWS SDK 或 AWS API 的程式請求。	<p>請依照您要使用的介面所提供的指示操作。</p> <ul style="list-style-type: none"> 關於 AWS CLI，請參閱 AWS Command Line Interface 使用者指南 中的 設定 AWS CLI 來使用 AWS IAM Identity Center。 關於 AWS SDK、工具和 AWS API，請參閱 AWS《SDK 和工具參考指南》 中的 IAM Identity Center 驗證。

哪個使用者需要程式設計存取權？	若要	By
IAM	使用臨時憑證簽署對 AWS CLI、AWS SDK 或 AWS API 的程式請求。	請遵循《IAM 使用者指南》中 使用臨時憑證搭配 AWS 資源 中的指示。
IAM	(不建議使用) 使用長期憑證簽署 AWS CLI 或 AWS API 的程式設計要求 (直接或使用 AWS SDK)。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> • 如需相關資訊 AWS CLI，請參閱《使用者指南》中的使用 IAM 使用者登入 AWS Command Line Interface 資料進行驗證。 • 關於 AWS SDK 和工具，請參閱 AWS《SDK 和工具參考指南》中的使用長期憑證進行驗證。 • 對於 AWS API，請參閱《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

使用 AWS CLI 建立永久存取憑證

Note

對於本節，建議您在 AWS 帳戶使用管理員層級憑證來設定 AWS CLI。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

執行 IAM `create-access-key` 命令，為使用者建立新的 AWS 存取金鑰及對應的 AWS 私密存取金鑰。

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

在上述命令中，將 `MyUser` 取代為使用者的名稱。

將畫面上顯示的 `AccessKeyId` 和 `SecretAccessKey` 值存放到安全的位置。在您執行 IAM `create-access-key` 命令後，就只有這麼一次機會可以使用 AWS 查看使用者的 AWS CLI 私密存取金鑰。若日後需要為使用者產生新的 AWS 私密存取金鑰，請參閱 IAM 使用者指南中的 [建立、修改和檢視存取金鑰 \(API、CLI、PowerShell\)](#)。

在環境中存放永久存取憑證

在本程序中，您會使用 AWS Cloud9 IDE 將永久 AWS 存取憑證存放在您的環境中。本程序假設您已在 AWS Cloud9 建立了環境、開啟了該環境，且您的 Web 瀏覽器已顯示 AWS Cloud9 IDE。如需詳細資訊，請參閱 [建立環境](#) 及 [開啟環境](#)。

Note

以下程序示範如何使用環境變數來存放永久存取登入資料。如果您的環境已安裝 AWS CLI 或 `aws-shell`，您可以改用 AWS CLI 的 `aws configure` 命令或 `aws-shell` 的 `configure` 命令，存放您的永久存取憑證。若要取得指示，請參閱 AWS Command Line Interface 使用者指南中的 [快速組態](#)。

1. 在您的環境已開啟的情況下，從 AWS Cloud9 IDE 啟動新的終端機工作階段 (若尚未啟動)。若要啟動新終端機工作階段，請在選單列上，選擇 `Window` (視窗)、`New Terminal` (新增終端機)。
2. 執行以下命令，分三次各執行一個命令，其設定的本機環境變數代表了您的永久存取登入資料。在這些命令的 `AWS_ACCESS_KEY_ID`: 後面輸入您的 AWS 存取金鑰 ID。在 `AWS_SECRET_ACCESS_KEY` 後面，輸入您的 AWS 私密存取金鑰。在 `AWS_DEFAULT_REGION_ID`

後面，輸入與您最接近之 AWS 區域 相關聯的 AWS 區域 識別符 (或您慣用的 AWS 區域)。如需可用識別碼的清單 [AWS 區域](#)，請參閱 Amazon Web Services 一般參考。例如，若為美國東部 (俄亥俄)，您會使用 us-east-2。

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

3. 請注意，上述環境變數僅對目前的終端機工作階段有效。若要能夠跨終端機工作階段使用這些環境變數，您必須將其加入至 shell 描述檔做為使用者環境變數，如下所示。
 - a. 在 IDE 的 Environment (環境) 視窗中選擇齒輪圖示，然後選擇 Show Home in Favorites (在我的最愛中顯示首頁)。重複此步驟並選擇 Show Hidden Files (顯示隱藏的檔案)。
 - b. 開啟 ~/.bashrc 檔案。
 - c. 在檔案末尾處輸入或貼上以下程式碼。在這些命令的 AWS_ACCESS_KEY_ID: 後面輸入您的 AWS 存取金鑰 ID。在 AWS_SECRET_ACCESS_KEY 後面，輸入您的 AWS 私密存取金鑰。在 AWS_DEFAULT_REGION_ID 後面，輸入與您最接近之 AWS 區域 相關聯的 AWS 區域 識別符 (或您慣用的 AWS 區域)。如需可用識別碼的清單 [AWS 區域](#)，請參閱 Amazon Web Services 一般參考。例如，若為美國東部 (俄亥俄) 區域，您會使用 us-east-2。

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

- d. 儲存檔案。
- e. 援引 ~/.bashrc 檔案以載入上述新的環境變數。

```
. ~/.bashrc
```

您現在即可開始從您的環境呼叫 AWS 服務。若要使用 AWS CLI 或 aws-shell 呼叫 AWS 服務，請參閱 [AWS CLI 和 aws-shell 範例](#)。若要從程式碼呼叫 AWS 服務，請參閱我們其他的 [教學和範例](#)。

變更 AWS Cloud9 的環境設定

您可以變更 AWS Cloud9 開發環境的偏好設定或其他設定。

- [變更環境偏好設定](#)
- [使用主控台變更環境設定](#)

- [使用程式碼變更環境設定](#)

變更環境偏好設定

1. 開啟您想要變更設定的環境。若要開啟環境，請參閱[開啟環境](#)。
2. 在 AWS Cloud9 IDE 的選單列上，選擇 AWS Cloud9、Preferences (偏好設定)。
3. 在 Preferences (偏好設定) 視窗中，選擇 Project Settings (專案設定)。
4. 視需要變更任何可用的專案設定。這些設定包括 Code Editor (Ace) (程式碼編輯器 (Ace)) 及 Find in Files (在檔案中尋找)。

Note

如需詳細資訊，請參閱[您可以變更的專案設定](#)。

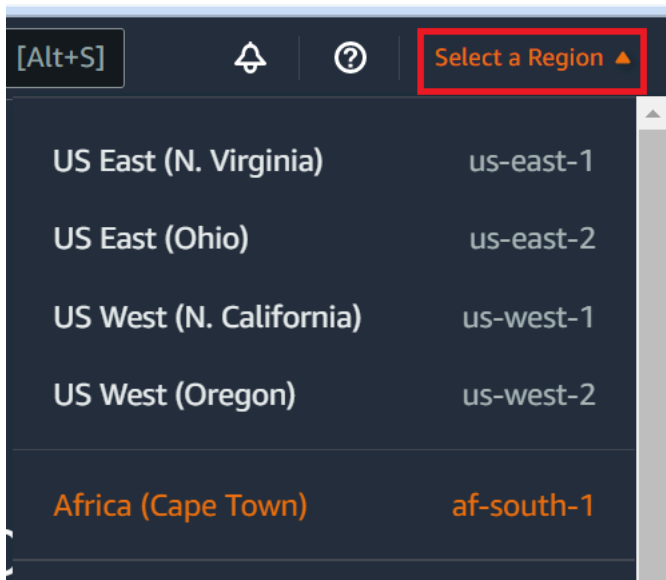
調整 AWS Cloud9 IDE 中環境的逾時

下列步驟概述如何更新 AWS Cloud9 IDE 中 Amazon EC2 環境的逾時期限。這將是環境停止之前的時間量。

1. 開啟您要設定的環境。
2. 在 AWS Cloud9 IDE 的功能表列上，選擇「AWS Cloud9 偏好設定」。
3. 在「偏好設定」視窗中，捲動至 Amazon EC2 執行個體區段。
4. 從可用清單中選取逾時值並進行更新。

使用主控台變更環境設定

1. 如下所示登入 AWS Cloud9 主控台：
 - 如果您是唯一使用 AWS 帳戶的個人，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶管理員提供登入指示。
2. 從頂端導覽列選擇環境所在的 AWS 區域。



3. 在環境清單中，針對您要變更設定的環境，執行下列其中一項操作。
 - 選擇環境的卡片標題。接著在下一頁選擇 View details (檢視詳細資訊)。
 - 選取環境的卡片，然後選擇 View details (檢視詳細資訊) 按鈕。
4. 進行變更，然後選擇 Save changes (儲存變更)。

您可以使用 AWS Cloud9 主控台變更下列設定。

- 如為 EC2 環境，您可變更 Name (名稱) 和 Description (描述)。
- 若是 SSH 環境，可變更：Name (名稱)、Description (描述)、User (使用者)、Host (主機)、Port (連接埠)、Environment path (環境路徑)、Node.js binary path (Node.js 二進位檔路徑) 和 SSH jump host (SSH 跳接主機)。

若要變更其他設定，請執行下列操作：

- 如為 EC2 環境，請執行下列其中一項操作。
 - 您無法變更 Type (類型)、Security groups (安全群組)、VPC、Subnet (子網路)、Environment path (環境路徑) 或 Environment ARN (環境 ARN)。
 - 如需變更 Permissions (許可) 或 Number of members (成員數目)，請參閱[變更環境成員的存取角色](#)、[移除使用者](#)、[邀請 IAM 使用者](#)，以及[移除其他環境成員](#)。
 - 如需變更 EC2 instance type (EC2 執行個體類型)、Memory (記憶體) 或 vCPU，請參閱「[移動環境或調整其規模](#)」。
- 如為 SSH 環境，請執行下列其中一項操作。
 - 您無法變更 Type (類型) 或 Environment ARN (環境 ARN)。

- 如需變更 Permissions (許可) 或 Number of members (成員數目)，請參閱[變更環境成員的存取角色](#)、[移除使用者](#)、[邀請 IAM 使用者](#)以及[移除其他環境成員](#)。

如果您的環境沒有顯示在主控台中，請嘗試執行下列一或多項動作讓環境顯示。

- 在 Environments (環境) 頁面的下拉式功能表列中，選擇下列一或多個選項。
 - 選擇 My environments (我的環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內擁有的所有環境。
 - 選擇 Shared with me (共用環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內已受邀加入的所有環境。
 - 選擇 All account environments (所有帳戶環境) 以顯示您的 AWS 區域 實體在所選 AWS 帳戶及 AWS內已獲許可顯示的所有環境。
- 若您認為自己是某個環境的成員，但 Shared with you (共用環境) 清單中並未顯示該環境，請向環境擁有者洽詢。
- 在頂端導覽列中，選擇不同的 AWS 區域。

使用程式碼變更環境設定

若要使用程式碼變更 AWS Cloud9 中的環境設定，請呼叫 AWS Cloud9 的更新環境操作，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

使用 AWS Cloud9 的共用環境

共用環境是多個使用者受邀參與的 AWS Cloud9 開發環境。本主題提供在 AWS Cloud9 共用環境，以及如何參與共用環境的指示。

若要邀請使用者參與您擁有的環境，請遵循下列其中一組程序進行。根據您要邀請的使用者類型選擇。

- 如果您是與環境相同的使用者AWS 帳戶，則應[邀請使用與環境相同的帳戶中的](#)使用者。
- 如果您是環境相同AWS 帳戶的系統管理AWS Cloud9員，特別是AWS 帳戶根使用者、系統管理員使用者或AWS Cloud9 Administrator附加AWS受管理原則的使用者，則您應該自行邀請管理AWS Cloud9員、[請參閱在相同帳戶中邀請使用者作為環境](#)邀請使用者，或讓系統管理AWS Cloud9員自行邀請 (或其他人AWS 帳戶)，請參閱在相[同帳戶中擁有系統管理AWS Cloud9員作為環境邀請自己或](#)[其他人](#)。

共用環境使用案例

共用環境適用於以下使用案例：

- 結對編程 (也稱為對等編程)：這是兩個用戶在單個環境中共同處理相同代碼的地方。在結對程式設計中，通常由一名使用者撰寫程式碼，另一名使用者觀察所寫的程式碼。觀察者可以立即提供輸入和意見回饋給程式碼撰寫者。在專案期間，這兩種角色會經常切換。在沒有共用環境的情況下，程式設計人員團隊通常是坐在一部機器前。一次只有一名使用者可以撰寫程式碼。使用共用環境，兩個使用者可以坐在自己的機器前。此外，即使他們在不同的實體辦公室，仍可以同時撰寫程式碼。
- 電腦科學課：如果老師或助教想要存取學生的環境，這項功能就很有用。這樣做可以檢閱學生的家庭作業，或即時修正其環境問題。學生也可以在單一環境中即時撰寫程式碼，和同學一起完成共用的家庭作業專案。即使他們可能在不同的位置、使用不同的電腦作業系統和 Web 瀏覽器類型，也可以做到。

- 任何需要多名使用者即時協同合作相同程式碼的其他情況。

關於環境成員的存取角色

在您共用環境或參與 AWS Cloud9 的共用環境之前，您應先了解共用環境的存取許可層級。我們稱這些許可層級為「環境成員存取角色」。

AWS Cloud9 中的共用環境提供三種環境存取角色：擁有者、讀取/寫入和唯讀。

- 擁有者具有環境的完整控制權。每個環境只能有一個擁有者，亦即環境的建立者。擁有者可以執行下列動作。
 - 新增、變更和移除環境成員
 - 開啟、檢視和編輯檔案
 - 執程式碼
 - 變更環境設定
 - 與其他成員聊天
 - 刪除現有的聊天訊息

在 AWS Cloud9 IDE 中，環境擁有者會顯示 Read+Write (讀取+寫入) 存取權。

- 讀取/寫入成員可以執行下列動作。
 - 開啟、檢視和編輯檔案
 - 執程式碼
 - 變更 AWS Cloud9 IDE 內的各種環境設定
 - 與其他成員聊天
 - 刪除現有的聊天訊息

在 AWS Cloud9 IDE 中，讀取/寫入成員會顯示 Read+Write (讀取+寫入) 存取權。

- 唯讀成員可以執行下列動作。
 - 開啟和檢視檔案
 - 與其他成員聊天
 - 刪除現有的聊天訊息

在 AWS Cloud9 IDE 中，唯讀成員會顯示 Read Only (唯讀) 存取權。

使用者必須符合下列其中一個條件，才能成為環境擁有者或成員。

- 使用者為 AWS 帳戶根使用者。
- 使用者為管理員使用者。如需詳細資訊，請參閱 IAM 使用者指南中的[建立您的第一個 IAM 管理員使用者和群組](#)。
- 使用者為隸屬於 IAM 群組的使用者、擔任角色的使用者或擔任角色的聯合身分使用者，而且該群組或角色需連接 AWS 受管政策 `AWSCloud9Administrator` 或 `AWSCloud9User` (若連接 `AWSCloud9EnvironmentMember` 則僅能成為成員)。如需詳細資訊，請參閱 [AWS 受管 \(預先定義\) 政策](#)。
- 如需將上述其中一個受管政策連接至 IAM 群組，則可以使用下列程序所述的 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#)。
- 您可以透過上述其中一個受管政策，在 IAM 中建立使用者或聯合身分使用者可以擔任的角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[建立角色](#)。如需讓使用者或聯合身分使用者擔任角色，請參閱 IAM 使用者指南中[使用 IAM 角色](#)的角色擔任相關說明。

使用主控台將 AWS Cloud9 的 AWS 受管政策連接到群組

下列程序概述如何使用主控台將 AWS 受管理的原則附加 AWS Cloud9 至群組。

1. 如果您尚未登入 AWS Management Console，請現在登入。

對於此步驟，建議您在 AWS 帳戶中使用 IAM 管理員層級憑證來登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。

2. 開啟 IAM 主控台。方法如下：從主控台的導覽列選擇 Services (服務)，然後選擇 IAM。
3. 選擇 Groups (群組)。
4. 選擇群組的名稱。
5. 在 Permissions (許可) 標籤上，針對 Managed Policies (受管政策)，選擇 Attach Policy (連接政策)。
6. 在政策名稱清單中，選擇下列其中一個方塊。
 - `AWSCloud9User` (建議使用) 或 `AWSCloud9Administrator` 可讓群組中的每位使用者成為環境的擁有者
 - `AWSCloud9EnvironmentMember` 可讓群組中的每位使用者成為一般成員

(如果您在清單中看不到這些政策名稱，請在 Search (搜尋) 方塊中輸入政策名稱以顯示。)

7. 選擇 Attach policy (連接政策)。

使用 AWS CLI 將 AWS Cloud9 的 AWS 受管政策連接到群組

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

執行 IAM `attach-group-policy` 命令，將 AWS Cloud9 的 AWS 受管政策連接到群組。指定群組名稱和政策的 Amazon Resource Name (ARN)。

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

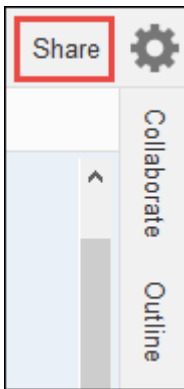
在上述命令中，將 `MyGroup` 取代為群組的名稱。將 `POLICY_NAME` 取代為下列其中一個 AWS 受管政策的名稱。

- `AWSCloud9User` (建議使用) 或 `AWSCloud9Administrator` 可讓群組中的每位使用者成為環境的擁有者
- `AWSCloud9EnvironmentMember` 可讓群組中的每位使用者成為一般成員

邀請所在帳戶與環境相同的使用者

使用本節中的指示，與相同帳戶的使用者共用您在 AWS 帳戶中所擁有的 AWS Cloud9 開發環境。

1. 假設您要邀請的使用者不是下列其中一種類型的使用者。確定您要邀請的使用者已具有對應的環境成員存取角色。如需說明，請參閱 [關於環境成員的存取角色](#) 小節。
 - AWS 帳戶 根使用者。
 - 管理員使用者。
 - 隸屬於 IAM 群組的使用者、擔任角色的使用者或擔任角色的聯合身分使用者，而且該群組或角色已連接 AWS 受管政策 `AWSCloud9Administrator`。
2. 開啟您想邀請使用者加入的環境 (如果尚未開啟)。
3. 在 AWS Cloud9 IDE 的選單列中，執行下列其中一項操作：
 - 選擇 `Window, Share` (視窗、共享)。
 - 選擇 `Share (共享)` (位於 `Preferences (偏好設定)` 齒輪圖示旁)。



4. 在 Share this environment (共享這個環境) 對話方塊的 Invite Members (邀請成員) 中，輸入以下其中一個項目。
 - 若要邀請 IAM 使用者，請輸入使用者名稱。
 - 若要邀請 AWS 帳戶根使用者，請輸入 `arn:aws:iam::123456789012:root`。用您的 AWS 帳戶 ID 取代 123456789012。
 - 若要邀請具有所擔任角色的使用者或具有所擔任角色的聯合身分使用者，請輸入 `arn:aws:sts::123456789012:assumed-role/MyAssumedRole/MyAssumedRoleSession`。將 123456789012 取代為您的 AWS 帳戶 ID，將 MyAssumedRole 取代為所擔任角色的名稱。將 MyAssumedRoleSession 取代為所擔任角色的工作階段名稱。
5. 若要讓此使用者成為唯讀成員，請選擇 R。若要讓此使用者成為讀取/寫入成員，請選擇 RW。
6. 選擇 Invite (邀請)。

Note

如果您讓此使用者成為讀取/寫入成員，畫面會顯示一個對話方塊，其中包含可能會對您的 AWS 安全憑證造成風險的資訊。下列資訊提供更多此問題的背景。

請只與信任的使用者共用環境。

讀取/寫入成員可以在您的環境中使用 AWS CLI、AWS CloudShell 或 AWS SDK，代您在 AWS 中執行動作。此外，如果您將永久 AWS 存取憑證存放在環境中，該成員可能會複製這些憑證，並使用於環境外部。

從您的環境移除永久 AWS 存取憑證，並改用臨時 AWS 存取憑證，並無法完全解決這個問題。這只能降低成員複製這些臨時憑證並用於環境外部的機會 (因為這些臨時憑證只在限定時間內有效)。不過，臨時憑證仍可讓讀取/寫入成員在 AWS 中代您從環境執行動作。

7. 聯絡使用者，讓他們知道可以開啟此環境並開始使用。

讓所在帳戶與環境相同的 AWS Cloud9 管理員邀請自己或其他使用者

Note

若使用 [AWS 受管臨時憑證](#)，則您無法在 AWS Cloud9 IDE 中使用終端機工作階段來執行此章節部分或全部的命令。為了實現 AWS 安全性最佳實務，AWS 受管臨時憑證不會允許執行部分命令。因此，請改從分開安裝的 AWS Command Line Interface (AWS CLI) 來執行這些命令。

下列類型的使用者可以邀請自己 (或相同 AWS 帳戶中的其他使用者) 加入相同帳戶中的任何環境。

- AWS 帳戶根使用者。
- 管理員使用者。
- 隸屬於 IAM 群組的使用者、擔任角色的使用者或擔任角色的聯合身分使用者，而且該群組或角色已連接 AWS 受管政策 `AWSCloud9Administrator`。

假設受邀的使用者不是上述類型的其中一個使用者。確定使用者已具有對應的環境成員存取角色。如需說明，請參閱[關於環境成員的存取角色](#)小節。

如需邀請使用者，請使用 AWS CLI 或 AWS CloudShell 來執行 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，將 `12a34567b8cd9012345ef67abcd890e1` 取代為環境的 ID。將 `PERMISSION_LEVEL` 取代為 `read-write` 或 `read-only`。然後，將 `USER_ARN` 取代為下列其中一項：

- 若要邀請 IAM 使用者，請輸入 `arn:aws:iam::123456789012:user/MyUser`。將 `123456789012` 取代為您的 AWS 帳戶 ID，並將 `MyUser` 取代為使用者的名稱。
- 若要邀請 AWS 帳戶根使用者，請輸入 `arn:aws:iam::123456789012:root`。用您的 AWS 帳戶 ID 取代 `123456789012`。
- 若要邀請具有所擔任角色的使用者或具有所擔任角色的聯合身分使用者，請輸入 `arn:aws:sts::123456789012:assumed-role/MyAssumedRole/MyAssumedRoleSession`。用您的 AWS 帳戶 ID 取代 `123456789012`。將 `MyAssumedRole` 取代為所擔任角色的名稱。然後，將 `MyAssumedRoleSession` 取代為所擔任角色的工作階段名稱。

例如，若要邀請 AWS 帳戶根使用者 (帳戶 ID 為 123456789012) 加入環境成為讀取/寫入成員 (ID 為 12a34567b8cd9012345ef67abcd890e1)，請執行下列命令。

```
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，請省略前述命令的 `aws` 前綴。

開啟共用環境

若要開啟共用環境，請使用 AWS Cloud9 儀表板。使用 AWS Cloud9 IDE 在共用環境中執行動作並完成工作。範例為使用檔案以及與其他團隊成員聊天。

1. 務必將對應的存取政策連接至使用者的群組或角色。如需詳細資訊，請參閱「[關於環境成員存取角色](#)」。
2. 如下所示登入 AWS Cloud9 主控台：
 - 如果您是唯一使用 AWS 帳戶的個人，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的組織使用 IAM Identity Center，請洽詢 AWS 帳戶管理員以取得登入指示。
 - 如果您是課堂裡的學生，請洽詢講師以取得登入指示。
3. 從您的 AWS Cloud9 儀表板開啟共用環境。如需詳細資訊，請參閱[開啟 AWS Cloud9 環境](#)。

您可以使用 Collaborate (協作) 視窗與其他成員互動，如本主題後續內容所述。

Note

如果 Collaborate (協作) 視窗未顯示，請選擇 Collaborate (協作)。如果 Collaborate (協作) 按鈕未顯示，請在選單列上，選擇 Window, Collaborate (視窗、協作)。

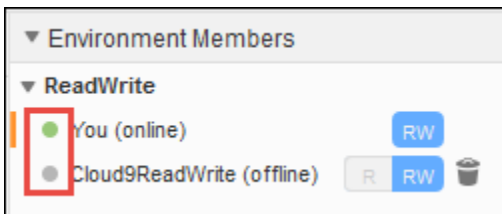


參閱環境成員清單

在開啟的共用環境中，如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。

每個成員旁的圓圈指出其線上狀態，如下所示。

- 活躍的成員有綠色圓圈。
- 離線的成員有灰色圓圈。
- 閒置的成員有橘色圓圈。



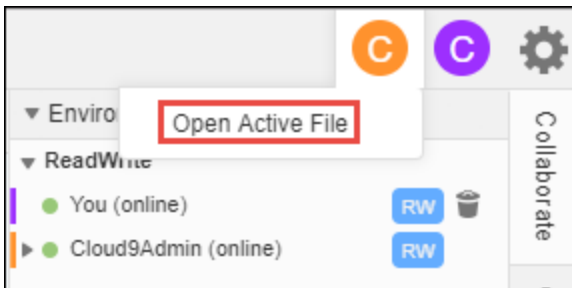
若要使用程式碼來取得環境成員的清單，請呼叫 AWS Cloud9 的 describe memberships (描述成員資格) 操作，如下所示。

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	獲取-C9EnvironmentMembershipList
AWS Cloud9 API	DescribeEnvironmentMemberships

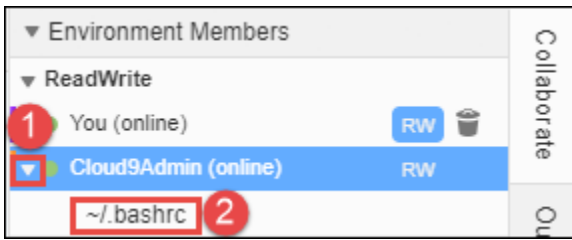
開啟環境成員的作用中檔案

在開啟的共用環境選單列中，選擇成員名稱。然後，選擇 Open Active File (開啟作用中的檔案)。



開啟環境成員的開放檔案

1. 在開啟的共用環境中，如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。
2. 針對您想要在環境中開啟其開放檔案的使用者，展開使用者名稱。
3. 開啟 (按兩下) 您想要開啟的檔案名稱。

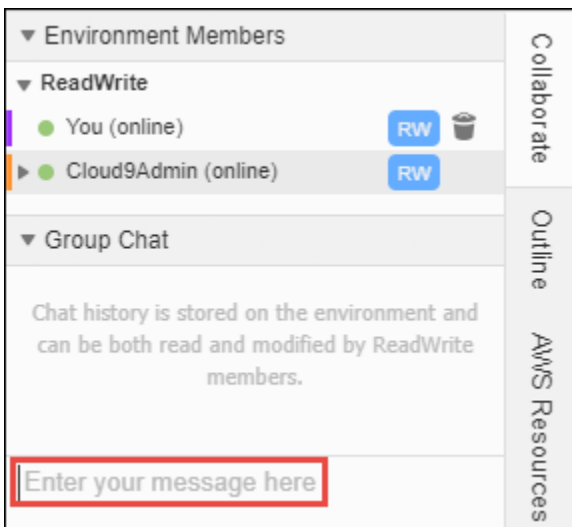


移至環境成員的作用中游標

1. 在開啟的共用環境中，如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。
2. 開啟成員名稱的內容 (右鍵) 功能表，然後選擇 Show Location (顯示位置)。

與其他環境成員聊天

在開啟的共用環境中，在 Collaborate (協作) 視窗底端的 Enter your message here (在此處輸入您的訊息) 中，輸入聊天訊息，然後按 Enter。



檢視共用環境的聊天訊息

在開啟的共用環境中，如果聊天訊息清單未顯示，請在 Collaborate (協作) 視窗中，展開 Group Chat (群組聊天)。

從共用環境刪除聊天訊息

共用環境開啟後，在 Collaborate (協作) 視窗中，為 Group Chat (群組聊天) 的聊天訊息開啟內容 (右鍵) 功能表。然後，選擇 Delete Message (刪除訊息)。

Note

當您刪除聊天訊息時，訊息會從所有成員的環境中刪除。

從共用環境刪除所有聊天訊息

共用環境開啟後，在 Collaborate (協作) 視窗中，於 Group Chat (群組聊天) 的任何位置開啟內容 (右鍵) 功能表。然後，選擇 Clear history (清除歷史記錄)。

Note

當您刪除所有聊天訊息時，訊息會從所有成員的環境中刪除。

變更環境成員的存取角色

1. 開啟您想要為成員變更存取角色的自有環境 (若尚未開啟)。如需詳細資訊，請參閱[開啟 AWS Cloud9 環境](#)。
2. 如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。
3. 執行下列其中一個動作：
 - 在您想要變更存取角色的成員名稱旁，選擇 R 或 RW，分別讓此成為擁有者或讀取/寫入角色。
 - 若要將讀取/寫入成員變更為唯讀成員，請開啟成員名稱的內容 (右鍵) 功能表，然後選擇 Revoke Write Access (撤銷寫入存取權)。
 - 若要將唯讀成員變更為讀取/寫入成員，請開啟成員名稱的內容 (右鍵) 功能表，然後選擇 Grant Read+Write Access (授予讀取/寫入存取權)。

Note

如果您讓此使用者成為讀取/寫入成員，畫面會顯示一個對話方塊，其中包含可能會對您的 AWS 安全憑證造成風險的資訊。除非您信任使用者可代表您在 AWS 中執行動作，否則請不要讓使用者成為讀取/寫入成員。如需詳細資訊，請參閱[邀請所在帳戶與環境相同的使用者](#)小節中的相關附註。

若要使用程式碼變更環境的存取角色，請呼叫 AWS Cloud9 的 update environment membership (更新環境成員資格) 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	更新資料-C9EnvironmentMembership
AWS Cloud9 API	UpdateEnvironmentMembership

從共用環境移除使用者

Note

如果您是環境擁有者，則無法從環境移除您的使用者。
從環境移除您的使用者，並不會一併從 IAM 移除該使用者。

1. 在開啟的共用環境中，如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。
2. 執行下列其中一個動作：

- 選擇在 You (您) 旁邊的垃圾桶圖示。
 - 為 You (您) 開啟內容 (右鍵) 功能表，然後選擇 Leave environment (離開環境)。
3. 出現提示時，請選擇 Leave (離開)。

若要使用程式碼從共用環境移除使用者，請呼叫 AWS Cloud9 的 delete environment membership (刪除環境成員資格) 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	移除-C9EnvironmentMembership
AWS Cloud9 API	DeleteEnvironmentMembership

移除其他環境成員

Note

若要從環境移除您的使用者以外的成員，您必須使用環境擁有者的憑證登入 AWS Cloud9。移除成員並不會從 IAM 移除使用者。

1. 開啟您想要移除成員的環境 (如果尚未開啟)。如需詳細資訊，請參閱[開啟 AWS Cloud9 環境](#)。
2. 如果成員清單未顯示，請在 Collaborate (協作) 視窗中，展開 Environment Members (環境成員)。
3. 執行下列任意一項：
 - 在您想要刪除的成員名稱旁邊選擇垃圾桶圖示。
 - 針對您想要刪除的成員名稱，開啟內容 (右鍵) 功能表，然後選擇 Revoke Access (撤銷存取權)。
4. 出現提示時，請選擇 Remove Member (移除成員)。

若要使用程式碼從環境移除成員，請呼叫 AWS Cloud9 的 delete environment membership (刪除環境成員資格) 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	移除-C9EnvironmentMembership
AWS Cloud9 API	DeleteEnvironmentMembership

共用環境最佳實務

共用環境時，建議您採取以下實務。

- 只邀請您信任的人到您的環境成為讀取/寫入成員。
- 若是 EC2 環境，讀取/寫入成員可以使用環境擁有者的 AWS 存取憑證，從環境呼叫 AWS 服務。此憑證代替他們自己的憑據。為了避免此狀況，環境擁有者可以停用環境的 AWS 受管臨時憑證。不過，這也會讓環境擁有者無法進行呼叫。如需詳細資訊，請參閱 [AWS 受管臨時憑證](#)。
- 開啟 AWS CloudTrail 以追蹤環境中的活動。如需詳細資訊，請參閱 [AWS CloudTrail 使用者指南](#)。
- 不要使用您的 AWS 帳戶根使用者建立和共用環境。請改用帳戶中的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的 [僅限第一次存取：您的根使用者憑證](#) 和 [IAM 使用者](#)。

移動環境並調整 Amazon EBS 磁碟區大小或予以加密

您可以將 AWS Cloud9 開發環境從一個 Amazon EC2 執行個體移至另一個執行個體。舉例來說，您也許想要執行下列其中一個動作：

- 從受損 (或與健康的執行個體相比，以異常的方式執行工作) 的 Amazon EC2 執行個體中移轉環境。
- 將環境從現有執行個體移轉到具有最新系統更新的執行個體。
- 由於現有執行個體中的環境不敷使用或使用率過低，而必須增加或減少執行個體的運算資源。

您可以通過遷移到新的 AWS Cloud9 EC2 環境從一個 AWS Cloud9 受支持的 AMI 升級到另一個，同時保留項目文件。您可能想要升級到另一個版本的 AMI，因為：

- 目前環境的 AMI 已達到 end-of-life 且不再受支援。
- 您需要的軟件包在當前 AMI 中已過時。

您也可以為與環境的 Amazon EC2 執行個體相關聯的 Amazon Elastic Block Store (Amazon EBS) 磁碟區調整大小。舉例來說，您也許想要執行下列其一或全部動作：

- 由於您即將用光執行個體上的儲存空間，而必須增加磁碟區的大小。
- 由於您不想為未使用的額外儲存空間支付費用，而必須減少磁碟區的大小。

在移動環境或調整大小之前，您可以嘗試停用環境中部分正在執行的程序，或將置換檔新增至環境。如需詳細了解如何處理記憶體不足或高 CPU 使用率，請參閱[疑難排解](#)。

Note

本主題僅說明如何在不同 Amazon EC2 執行個體之間轉移環境或調整 Amazon EBS 磁碟區的大小。若要從某部自有伺服器調整環境大小，或為某部自有伺服器變更儲存空間，請參閱您伺服器的文件。

最後，您可以加密 Amazon EBS 資源，以確保執行個體 data-at-rest 及其連接 EBS 儲存 data-in-transit 之間以及之間的安全性。

主題

- [如何移動環境](#)
- [將 AWS Cloud9 EC2 環境移動到不同的 Amazon 機器映像 \(AMI\)](#)
- [為環境使用的 Amazon EBS 磁碟區調整大小](#)
- [加密使 AWS Cloud9 用的 Amazon EBS 磁碟區](#)

如何移動環境

在您開始移動程序之前，請注意以下條件：

- 您無法將環境移至相同類型的 Amazon EC2 執行個體。移動時，您必須為新的執行個體選擇不同的 Amazon EC2 執行個體類型。

Important

如果您將環境移到另一個 Amazon EC2 執行個體類型，目前的執行個體類型也必須受 AWS Cloud9 到支援 AWS 區域。若要檢查每個區域的可用執行個體類型，請移至[使用主控台建立](#)

EC2 環境時顯示的 Configure settings (組態設定) 頁面。您在 [執行個體類型] 區段中 AWS 區域的選擇取決於主控台右上方選取的項目。

- 您必須先停止與環境相關聯的 Amazon EC2 執行個體，方可變更執行個體類型。當執行個體停止時，您和任何成員都不能使用與停止的執行個體相關聯的環境。
- AWS 將執行個體移至新硬體，但執行個體的 ID 不會變更。
- 如果執行個體在 Amazon VPC 中執行，且具有公用 IPv4 地址，請 AWS 釋放該地址並為執行個體提供新的公有 IPv4 地址。執行個體會保留其私有 IPv4 地址，以及任何彈性 IP 地址或 IPv6 地址。
- 請為執行個體停止時的停機時間做好規劃。此程序可能需要幾分鐘時間。

如何移動環境


1. (選用) 若新執行個體類型需要的驅動程式並未安裝於現有的執行個體，請連線至您的執行個體並安裝這些驅動程式。如需詳細資訊，請參閱《適用於 Linux 執行個體的 Amazon EC2 使用者指南》中的「[調整執行個體大小的相容性](#)」。
2. 關閉 Web 瀏覽器中正在顯示環境的所有標籤頁。

Important

如果您未關閉目前顯示環境的所有 Web 瀏覽器索引標籤，AWS Cloud9 可能會干擾完成此程序。具體而言，在此程序期間，AWS Cloud9 可能會在錯誤的時間嘗試重新啟動與環境相關聯的 Amazon EC2 執行個體。此執行個體必須保持停止，直到程序中的最後一個步驟。


3. 如果您尚未登入，請至 <https://console.aws.amazon.com> 登入。AWS Management Console 建議您使用. 中的系統管理員層級認證登入。AWS 帳戶如果您無法這麼做，請洽詢您的 AWS 帳戶系統管理員。
4. 開啟 Amazon EC2 主控台。若要執行此操作，請在 Services (服務) 清單中，選擇 EC2。
5. 在 AWS 導覽列中，選擇包 AWS 區域 含您要移動之環境的環境 (例如，美國東部 (俄亥俄))。
6. 在服務導覽窗格中，展開 Instances (執行個體)，然後選擇 Instances (執行個體)。
7. 在執行個體清單中，選擇與您要移動的環境相關聯的執行個體。如為 EC2 環境，執行個體的名稱會以 aws-cloud9- 開頭，其後接著環境的名稱。舉例來說，如果環境名稱為 my-demo-environment，執行個體名稱開頭即為 aws-cloud9-my-demo-environment。
8. 如果「執行處理狀態」不是「停止」，請選擇「動作」、「執行處理狀態」出現提示時，請選擇 Yes, Stop (是，停止)。停止執行個體可能需要幾分鐘。

9. 在 Instance State (執行個體狀態) 為 stopped (停止) 之後，並且該執行個體仍為已選取的情況下，選擇 Actions (動作)、Instance Settings (執行個體設定)、Change Instance Type (變更執行個體類型)。
10. 在 Change Instance Type (變更執行個體類型) 對話方塊中為您要使用的環境選擇新的 Instance Type (執行個體類型)。

 Note


若您想要的執行個體類型並未出現在清單中，表示該類型與您執行個體的組態不相容。例如，執行個體可能由於虛擬化類型而不相容。

11. (選用) 若您選擇的執行個體類型支援 EBS 最佳化，請選取 EBS-optimized (EBS 最佳化) 以啟用 EBS 最佳化，或取消選取 EBS-optimized (EBS 最佳化) 以停用 EBS 最佳化。

 Note

如果您選擇的執行個體類型預設為 EBS 最佳化，則 EBS-optimized (EBS 最佳化) 便會處於選取狀態，而且您無法取消選取。

12. 選擇 Apply (套用) 以接受新的設定。

 Note

如果您先前未在此程序中，為 Instance Type (執行個體類型) 選擇不同的執行個體類型，在您選擇 Apply (套用) 之後，不會發生任何事情。

13. 重新開啟環境。如需詳細資訊，請參閱 [在 AWS Cloud9 中開啟環境](#)。

如需上述程序的詳細資訊，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的 [變更執行個體類型](#)。

將 AWS Cloud9 EC2 環境移動到不同的 Amazon 機器映像 (AMI)

本主題說明如何將 AWS Cloud9 EC2 環境從一個 Amazon Linux AMI 遷移到另一個 AWS Cloud9 受支援的 AMI。

Note

如果您想要在不更新作業系統版本的情況下將環境移至新執行個體，請參閱[the section called “如何移動環境”](#)。

您可以使用下列其中一個程序在環境之間移轉資料：

透過將歸檔下載至本機電腦來移動環境

1. 在具有不同基礎映像檔的相同可用區域中建立新環境：
 - a. 完成本[the section called “建立 EC2 環境”](#)節中的步驟以建立新環境。

Note

選擇「平台」時，請選取您要移轉環境的目標平台。

- b. 依預設，環境是使用 10 GiB 磁碟區建立的。如果您沒有足夠的空間將存檔上傳或解壓縮到新環境，請完成調整 Amazon EBS 磁碟區大小的[the section called “為環境使用的 Amazon EBS 磁碟區調整大小”](#)程序中的步驟。
2. 開啟您要在 AWS Cloud9 IDE 中移轉的環境。
3. AWS Cloud9 IDE 載入後，從功能表中選取「檔案」>「下載專案」，以下載包含環境專案目錄內容的歸檔。
4. 在新環境中開啟 AWS Cloud9 IDE。
5. 選擇「檔案 > 上傳本機檔案...」。以上傳歸檔。
6. (選擇性) 若要將舊目 .c9 錄備份至環境終端機中 .c9.backup，請執行下列命令：

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

如果您想要稍後還原組態檔案，可能需要這些備份檔案。

7. 若要解壓縮歸檔，請執行下列命令：

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

8. 若要從專案目錄刪除歸檔，請執行下列命令：

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

確定新環境如預期般運作。

9. 您現在可以刪除舊環境。

使用 Amazon EBS 磁碟區移動環境

如果您無法下載存檔，或產生的存檔太大，則可以使用 Amazon EBS 磁碟區進行遷移。此外，此方法可讓您複製位於~/environment目錄外的檔案。

1. 關閉在現有環境中開啟的所有 AWS Cloud9 IDE 索引標籤。
2. 完成下列步驟以停止現有執行處理：
 - a. 在 AWS Cloud9 主控台中，選取要導覽的環境以檢視其詳細資料。
 - b. 在環境詳細資料頁面的 EC2 執行個體索引標籤下，選擇管理 EC2 執行個體。
 - c. 在 EC2 主控台中，選取要導覽至執行個體詳細資料的執行個體。
 - d. 確定「執行處理」狀態已設定為「已停止」。如果沒有，請從執行個體狀態下拉式清單中選取停止執行個體 出現提示時，選擇「停止」。停止執行個體可能需要幾分鐘。
3. 在具有不同基礎映像檔的相同可用區域中建立新環境：
 - a. 完成本[the section called “建立 EC2 環境”](#)節中的步驟以建立新環境。

Note

選擇「平台」時，請選取您要移轉環境的目標平台。

- b. 依預設，環境是使用 10 GiB 磁碟區建立的。如果您沒有足夠的空間將檔案從來源磁碟區移至新環境，請完成調整 Amazon EBS 磁碟區大小的[the section called “為環境使用的 Amazon EBS 磁碟區調整大小”](#)程序中的步驟。
4. 完成下列步驟，將磁碟區與現有執行個體卸離：
 - a. 在 [執行個體摘要] 頁面上，選擇 [儲存] 索引標籤並選取磁碟區。所選磁碟區的裝置名稱必須與根裝置詳細資料區段的根裝置名稱中指定的裝置名稱相同。
 - b. 在磁碟區詳細資訊頁面上，選擇 [動作] > [卸離磁碟區]

- c. 成功卸離磁碟區之後，請選擇 [動作] > [附加磁碟區]，然後從下拉式清單中尋找並選取新環境的執行個體。您選取的 Amazon EC2 執行個體名稱必須包含前置詞的 AWS Cloud9 環境名稱。aws-cloud9
5. 在新環境中開啟 AWS Cloud9 IDE。
6. 環境載入後，若要識別新連接磁碟區的裝置，請在終端機中執行下列指令：

```
lsblk
```

在下面的示例輸出中，根設備nvme0n1的分區nvme0n1p1已經掛載，因此也必須掛載nvme1n1p1分區。其設備的完整路徑是/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

當您在終端機中執行此命令時，輸出會有所不同。

7. 在環境終端機中完成下列步驟，以掛載現有的磁碟區：
 - a. 若要建立要掛載磁碟區分割區的暫存目錄，請執行下列命令：

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

- b. 根據指lsblk令的範例輸出，指定要掛載之裝置的下列路徑：

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

Note

當您在終端機中執行此命令時，輸出會有所不同。

- c. 若要掛接現有的磁碟區，請執行下列命令：

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

- d. 請完成下列步驟，確認現有的磁碟區是否已正確掛載：

- i. 若要確保磁碟區包含在輸出中，請執行下列命令：

```
df -h
```

- ii. 若要驗證磁碟區的內容，請執行下列命令：

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


8. (選擇性) 若要將舊目錄 .c9 錄備份至環境終端機中 .c9.backup，請執行下列命令：

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

如果您想要稍後還原組態檔案，可能需要這些備份檔案。

9. 若要從現有磁碟區複製舊環境，請執行下列命令：

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

 Note

如有必要，您也可以使用上述指令將檔案或目錄複製到環境目錄之外。

確定新環境如預期般運作。

10. 若要卸載先前的裝置，請執行下列兩個指令之一：

```
sudo umount $MOUNT_DEVICE
```

```
sudo umount $MOUNT_POINT
```

11. 從 [動作] 下拉式清單中選擇 [分離磁碟區]，以分離您在步驟 3 中連接的磁碟區。

12. 您現在可以刪除舊環境及其磁碟區。

Note

由於磁碟區不再連接到環境的 Amazon EC2 執行個體，因此您需要手動將其移除。您可以在 [\[磁碟區詳細資訊\]](#) 頁面上選擇刪除來執行此操作。

為環境使用的 Amazon EBS 磁碟區調整大小

1. 針對您要調整大小的 Amazon EBS 磁碟區開啟與 Amazon EC2 執行個體相關聯的環境。
2. 在環境的 AWS Cloud9 IDE 中，建立包含下列內容的檔案，然後以副檔名儲存檔案 .sh (例如，resize.sh)。

注意

此指令碼適用於連接至執行 AL2023、Amazon Linux 2、Amazon Linux 或 Ubuntu 伺服器且已設定為使用 IMDSv2 的 EC2 執行個體的 Amazon EBS 磁碟區。指令碼也會為在 Nitro 型執行個體上經公開為 NVMe 區塊型儲存設備的 Amazon EBS 磁碟區調整大小。如需 Nitro 系統的執行個體清單，請參閱《Linux 執行個體適用的 Amazon EC2 使用者指南》中的 [Nitro 型執行個體](#)。

```
#!/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. 從 IDE 的終端機工作階段中，切換至包含 `resize.sh` 檔案的目錄。然後執行以下其中一個命令，將 20 取代為您要為 Amazon EBS 磁碟區調整的大小 (以 GiB 為單位)：

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

加密使 AWS Cloud9 用的 Amazon EBS 磁碟區

Amazon EBS 加密功能會將下列資料加密：

- 磁碟區內的待用資料
- 所有在磁碟區和執行個體間移動的資料
- 所有從磁碟區建立的快照
- 所有從那些快照建立的磁碟區

您可以為 AWS Cloud9 EC2 開發環境使用的 Amazon EBS 磁碟區採取兩種加密選項：

- **Encryption by default (預設加密)**– 您可以將 AWS 帳戶 設為強制加密您建立的新 EBS 磁碟區和快照複本。依預設，加密是在 AWS 區域的層級啟用。因此，您無法針對該區域中的個別磁碟區或快照啟用加密。此外，Amazon EBS 會加密您啟動執行個體時建立的磁碟區。因此，您必須先啟用此設定，然後才能建立 EC2 環境。如需詳細資訊，請參閱適用於 Linux 執行個體適用的 Amazon EC2 使用者指南中的[預設加密](#)。
- **Encryption of an existing Amazon EBS volume used by an EC2 environment (為 EC2 環境使用的現有 Amazon EBS 磁碟區加密)**– 您可以為針對 EC2 執行個體建立的特定 Amazon EBS 磁碟區進行加

密。此選項涉及使用 AWS Key Management Service (AWS KMS) 來管理加密磁碟區的存取。如需相關程序，請參閱 [加密 AWS Cloud9 使用的現有 Amazon EBS 磁碟區](#)。

Important

如果您的 AWS Cloud9 IDE 使用預設已加密的 Amazon EBS 磁碟區，則的 AWS Identity and Access Management 服務連結角色 AWS Cloud9 需要存取這些 EBS 磁碟區 AWS KMS key 的。如果未提供存取權，AWS Cloud9 IDE 可能無法啟動，並且偵錯可能很困難。

若要提供存取權，請將 AWS Cloud9AWSServiceRoleForAWSCloud9、的服務連結角色新增至 Amazon EBS 磁碟區使用的 KMS 金鑰。如需有關此工作的詳細資訊，請參閱AWS 規範指導模式中[建立使用具有預設加密的 Amazon EBS 磁碟區的 AWS Cloud9 IDE](#)。

加密 AWS Cloud9 使用的現有 Amazon EBS 磁碟區

加密現有的 Amazon EBS 磁碟區需 AWS KMS 要使用建立 KMS 金鑰。建立要取代的磁碟區快照之後，您必須使用 KMS 金鑰來加密快照複本。

接下來，您會建立包含該快照的加密磁碟區。然後，您要將未加密的磁碟區從 EC2 執行個體中分開，並連接加密磁碟區，藉此取代未加密磁碟區。

最後，您必須更新客戶管理金鑰的金鑰政策，才能存取 AWS Cloud9 服務角色。


Note

下列程序主要說明如何使用客戶管理的金鑰來加密磁碟區。您也可以 AWS 服務 在您的帳戶中使用 a. AWS 受管金鑰 Amazon EBS 的別名為 aws/ebs。如果您選擇此預設選項進行加密，請略過建立客戶管理金鑰的步驟 1。此外，請跳過您更新金鑰政策的步驟 8。這是因為您無法變更 AWS 受管金鑰。

如何加密現有 Amazon EBS 磁碟區


1. 在 AWS KMS 主控台中，建立對稱 KMS 金鑰。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的[建立對稱 KMS 金鑰](#)。
2. 在 Amazon EC2 主控台中，停止環境使用的 Amazon EBS 執行個體。您可以[使用主控台或命令列停止執行個體](#)。

3. 在 Amazon EC2 主控台的導覽窗格中，選擇 Snapshots (快照) 為您想要加密的[現有磁碟區建立快照](#)。
4. 在 Amazon EC2 主控台的導覽窗格中，選擇 Snapshots (快照) 來[複製快照](#)。在 Copy snapshot (複製快照) 對話方塊中，執行下列操作以啟用加密：
 - 選擇 Encrypt this snapshot (加密此快照)。
 - 針對 Master Key (主金鑰)，選取您稍早建立的 KMS 金鑰。(如果您使用的是 AWS 受管金鑰，請保留 (默認) aws/ebs 設置。)
5. [從加密快照建立新磁碟區](#)。

 Note

從加密快照建立的新 Amazon EBS 磁碟區會自動經過加密。

6. [將舊的 Amazon EBS 磁碟區從 Amazon EC2 執行個體分開](#)。
7. [將新的加密磁碟區連接到 Amazon EC2 執行個體](#)。
8. [使用 AWS Management Console 預設檢視、原則檢視或 AWS KMS API 更新 KMS 金鑰的金鑰 AWS Management Console 原則](#)。新增下列金鑰原則陳述式 AWSServiceRoleForAWSCloud9，以允許 AWS Cloud9 服務存取 KMS 金鑰。

 Note

如果您使用的是 AWS 受管金鑰，請略過此步驟。

```
{
  "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. 重新啟動 Amazon EC2 執行個體。如需有關重新啟動 Amazon EC2 執行個體的詳細資訊，請參閱[停止並啟動執行個體](#)。

刪除 AWS Cloud9 中的環境

為避免您的 AWS 帳戶 因仍保留已不再使用的 AWS Cloud9 開發環境而持續產生相關費用，請刪除該環境。

- [使用主控台刪除環境](#)
- [使用程式碼刪除環境](#)

使用主控台刪除環境

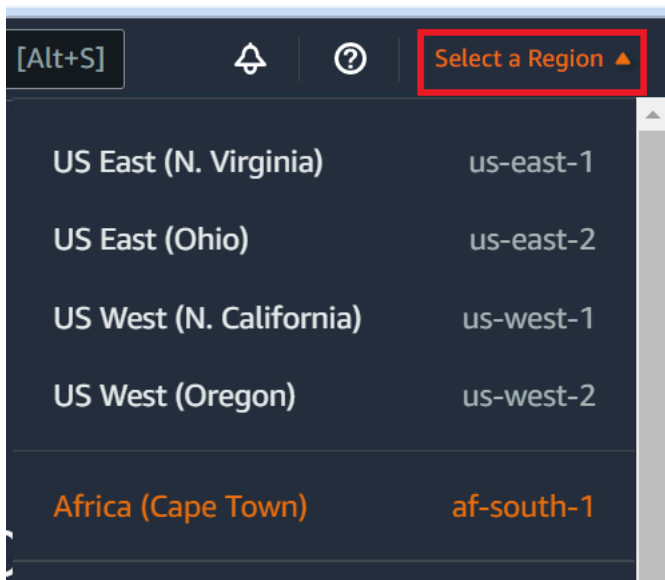
⚠ Warning

刪除環境時，AWS Cloud9 會將環境永久刪除。這包括永久刪除所有相關的設定、使用者資料，與未提交的程式碼。刪除的環境無法復原。

1. 登入 AWS Cloud9 主控台：

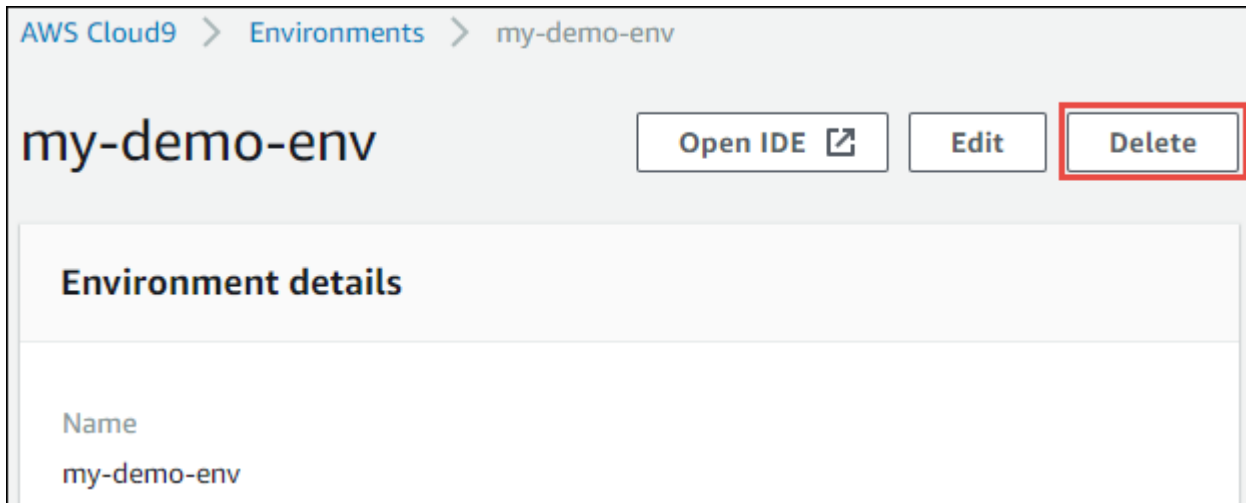
- 如果您是唯一使用 AWS 帳戶的使用者，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
- 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶 管理員提供登入指示。

2. 從頂端導覽列選擇環境所在的 AWS 區域。

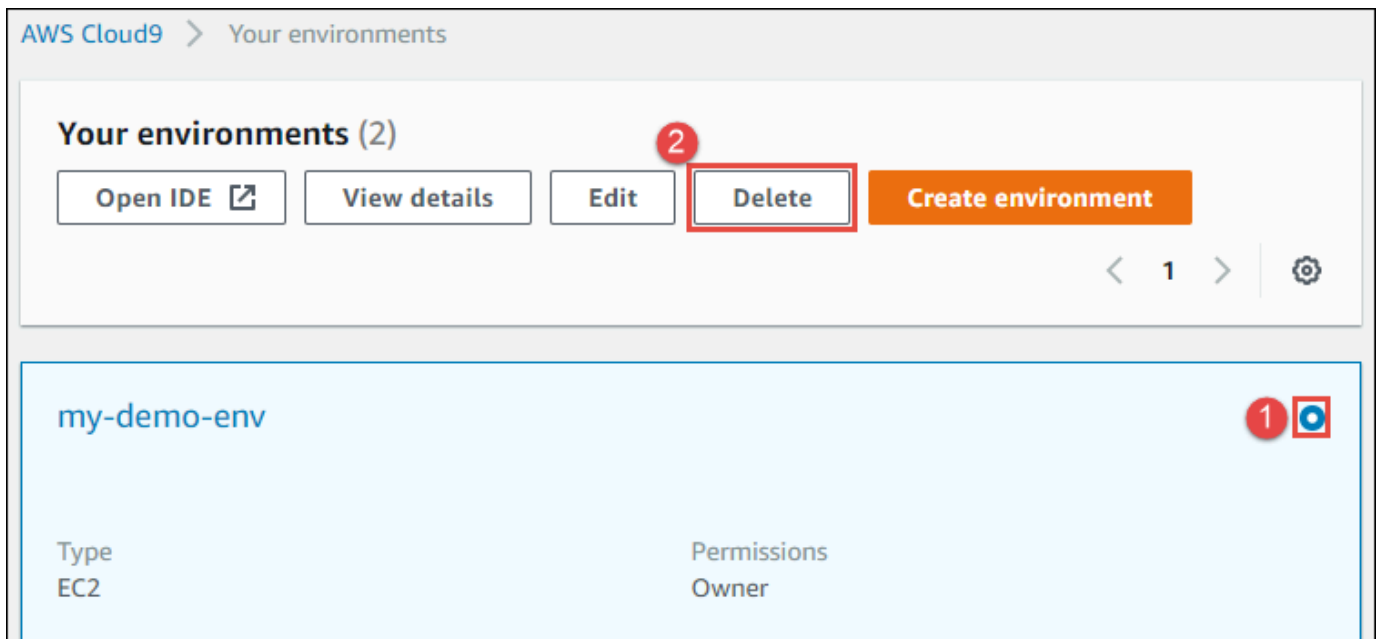


3. 在環境清單中，針對您要刪除的環境執行以下其中一項動作。

- 選擇環境的卡片標題。然後，在下一個頁面上選擇 Delete (刪除)。



- 選取環境的卡片，然後選擇 Delete (刪除) 按鈕。



4. 在 Delete (刪除) 對話方塊中輸入 Delete，然後選擇 Delete (刪除)。

- EC2 環境

AWS Cloud9 也會終止已連線到該環境的 Amazon EC2 執行個體。

Note

如果帳戶刪除失敗，主控台網頁頂端會顯示橫幅。此外，環境的卡片 (如果存在) 會指出環境刪除失敗。

- SSH 環境

如果環境已連線到 Amazon EC2 執行個體，則 AWS Cloud9 不會終止該執行個體。如果您之後不終止該執行個體，您的 AWS 帳戶可能會持續產生與該執行個體相關的 Amazon EC2 費用。

5. 如果是 SSH 環境，AWS Cloud9 會在已連接到該環境的雲端運算執行個體或您自己的伺服器上留存一個隱藏的子目錄。如果想要將其刪除，您現在可以安全地刪除該子目錄。子目錄的名稱是 `.c9`，子目錄位於您建立環境時所指定的 Environment path (環境路徑) 目錄下。

如果您的環境沒有顯示在主控台中，請嘗試執行下列一或多項動作將其顯示。

- 在 Environments (環境) 頁面的下拉式功能表列中，選擇下列一或多個選項。
 - 選擇 My environments (我的環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內擁有的所有環境。
 - 選擇 Shared with me (共用環境) 以顯示您的 AWS 實體在所選 AWS 區域及 AWS 帳戶內已受邀加入的所有環境。
 - 選擇 All account environments (所有帳戶環境) 以顯示您的 AWS 區域 實體在所選 AWS 帳戶及 AWS 內已獲許可顯示的所有環境。
- 若您認為自己是某個環境的成員，但 Shared with you (共用環境) 清單中並未顯示該環境，請向環境擁有人洽詢。
- 在頂端導覽列中，選擇不同的 AWS 區域。

使用程式碼刪除環境

Warning

刪除環境時，AWS Cloud9 會將環境永久刪除。這包括永久刪除所有相關的設定、使用者資料，與未提交的程式碼。刪除的環境無法復原。

若要使用程式碼刪除 AWS Cloud9 中的環境，請呼叫 AWS Cloud9 的刪除環境操作，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 整合開發環境 (IDE)

整合式開發環境 (IDE) 提供一套程式碼編寫生產力工具，例如原始碼編輯器、除錯器及建置工具。

Important

以下是使用 AWS Cloud9 的推薦最佳實務：

- 經常為您的環境使用 source control and backup (來源控制和備份)。AWS Cloud9 不會執行自動備份。
- 在您的環境中定期執行 updates of software (軟體更新)。AWS Cloud9 不會執行自動軟體更新。
- 開啟您AWS CloudTrail的AWS帳戶以追蹤環境中的活動。如需詳細資訊，請參閱 [使用 AWS CloudTrail 記錄 AWS Cloud9 API 呼叫](#)。
- 僅與 trusted users (信任的使用者) 共用您的環境。共用您的環境可能會使您的 AWS 存取憑證面臨風險。如需詳細資訊，請參閱 [使用 AWS Cloud9 的共用環境](#)。

請閱讀以下任一主題或多個主題，了解如何使用 AWS Cloud9 IDE。

主題

- [導覽 AWS Cloud9 IDE](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的語言支援](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 中的增強語言支援](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的選單列命令參考](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中尋找及取代文字](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中預覽檔案](#)
- [預覽 AWS Cloud9 整合開發環境 \(IDE\) 裡執行中的應用程式](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用檔案修訂](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用影像檔案](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用建置器、執行器和除錯器](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用自訂環境變數](#)
- [使用 AWS Cloud9 整合式開發環境 \(IDE\) 中的專案設定](#)

- [使用 AWS Cloud9 IDE 中的使用者設定](#)
- [在 AWS Cloud9 整合式開發環境 \(IDE\) 中使用 AWS 專案和使用者設定](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中處理按鍵繫結關係](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用佈景主題](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中管理初始化指令碼](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 MacOS 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 MacOS Vim 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 MacOS Emacs 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 MacOS Sublime 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 Windows / Linux 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 Windows / Linux Vim 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 Windows / Linux Emacs 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的 Windows / Linux Sublime 預設按鍵繫結關係參考](#)
- [AWS Cloud9 整合開發環境 \(IDE\) 的命令參考](#)

導覽 AWS Cloud9 IDE

本主題提供 AWS Cloud9 整合開發環境 (IDE) 的基本導覽。若要充分利用此導覽，請依序遵循以下顯示的步驟。

主題

- [先決條件](#)
- [步驟 1：選單列](#)
- [步驟 2：儀表板](#)
- [步驟 3：環境視窗](#)
- [步驟 4：編輯器、標籤和窗格](#)
- [步驟 5：主控台](#)
- [步驟 6：開啟檔案區段](#)
- [步驟 7：裝訂邊](#)
- [步驟 8：狀態列](#)

- [步驟 9：大綱視窗](#)
- [步驟 10：執行視窗](#)
- [步驟 11：即時運算標籤](#)
- [步驟 12：程序清單](#)
- [步驟 13：偏好設定](#)
- [步驟 14：終端機](#)
- [步驟 15：除錯器視窗](#)
- [結論](#)

先決條件

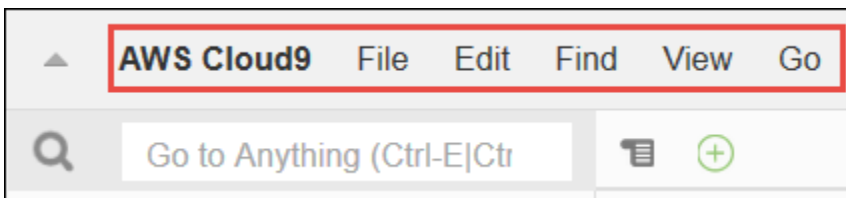
若要繼續此導覽，您必須擁有 AWS 帳戶與開放的 AWS Cloud9 開發環境。若要了解如何執行這些作業，您可以遵循 [入門：AWS Cloud9 的基本教學課程](#) 中的步驟。或者，您可以探索個別的相關主題，例如 [設定 AWS Cloud9](#) 和 [使用環境 AWS Cloud9](#)。

⚠ Warning

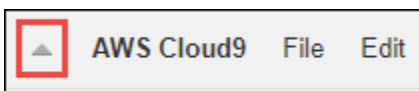
使用 AWS Cloud9 開發環境可能會導致您的 AWS 帳戶產生費用。如果您使用 EC2 環境，則其中包括 Amazon EC2 可能產生的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

步驟 1：選單列

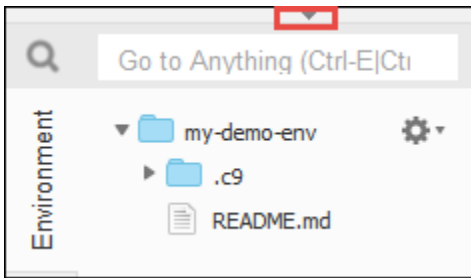
IDE 頂端邊緣的「選單列」包含用於使用檔案和程式碼以及變更 IDE 設定的常見命令。您也可以從選單列預覽和執行程式碼。



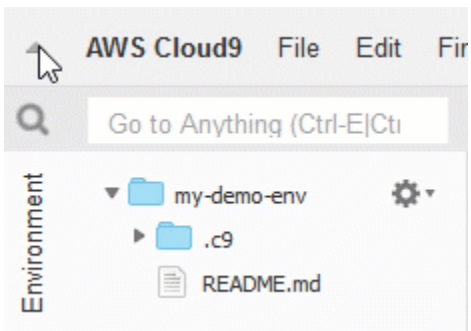
您可以選擇選單列邊緣的箭頭來隱藏選單列，如下所示。



您可以從選單列稍早所在的位置中間選擇箭頭來再次顯示選單列，如下所示。



將您的結果與下列項目比較。



您可以使用 IDE 來使用本教學接下來幾節的一組檔案。若要設定這些檔案，請選擇 File (檔案)、New File (新增檔案)。

接下來，將下列文字複製到 Untitled1 編輯器標籤。

```
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.
```

若要儲存檔案，請選擇 File (檔案)、Save (儲存)。將檔案命名為 fish.txt，然後選擇 Save (儲存)。

重複這些說明，將第二個檔案儲存為 cat.txt，其中包含下列內容。

```
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
```

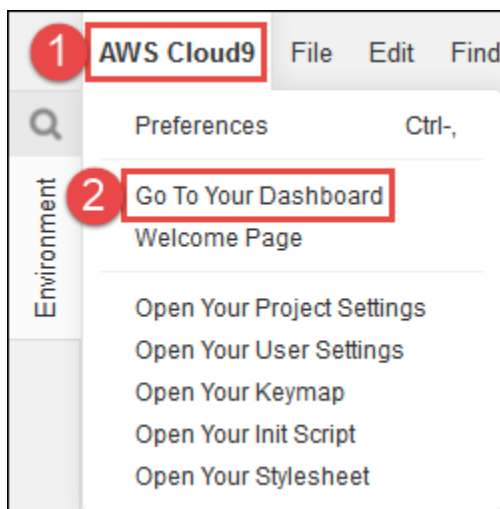
companionship and for their ability to hunt.

通常有幾種方法可以在 IDE 中執行某些動作。例如，若要使用選擇邊緣箭頭以外的方法來隱藏選單列，您可以選擇 View (檢視)、Menu Bar (選單列)。若要使用選擇 File, New File(檔案、新增檔案) 以外的方法來建立新檔案，您可以按下 Alt-N (適用於 Windows/Linux) 或 Control-N (適用於 MacOS)。為縮減此教學的長度，我們只描述一種工作方式來作業。隨著您對 IDE 的使用更加舒適，請隨意試驗並找出最適合您的方式。

步驟 2：儀表板

「儀表板」可讓您快速存取您的每個環境。您可以從儀表板中建立、開啟和變更環境的設定。

若要開啟儀表板，請在選單列上選擇 AWS Cloud9、Go To Your Dashboard (移至儀表板)。



若要檢視您的環境設定，請選擇 my-demo-environment 卡片中的標題。若要返回儀表板，請使用 Web 瀏覽器的返回按鈕或名為 Environments (環境) 的頁面導覽路徑。

若要開啟環境的 IDE，請選擇 my-demo-environment 卡片中的 Open IDE (開啟 IDE)。

Note

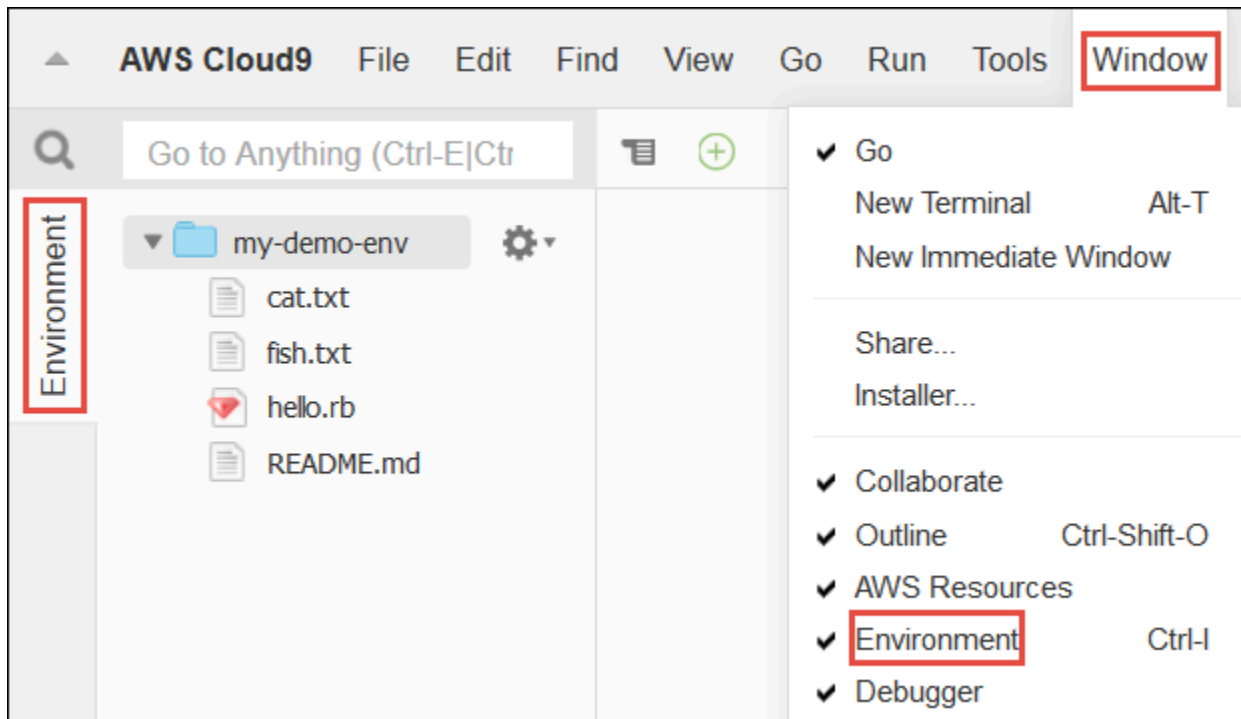
IDE 可能需要一些時間才能再次顯示。

步驟 3：環境視窗

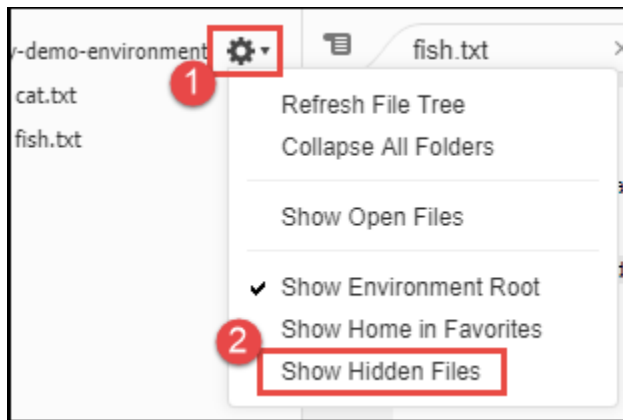
Environment (環境) 視窗顯示您在環境中的資料夾和檔案清單。您也可以顯示不同類型的檔案，例如隱藏檔。

若要顯示或隱藏 Environment (環境) 視窗，請選擇 Environment (環境) 按鈕。

若要顯示或隱藏 Environment (環境) 視窗和 Environment (環境) 按鈕，請選擇選單列上的 Window (視窗)、Environment (環境)。



若要顯示或隱藏檔案，請在 Environment (環境) 視窗中選擇齒輪圖示，然後選擇 Show Hidden Files (顯示隱藏的檔案)。



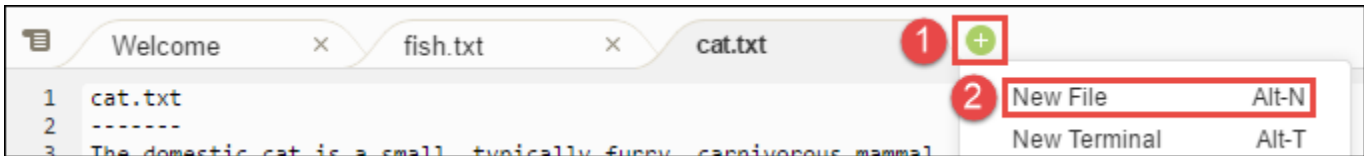
步驟 4：編輯器、標籤和窗格

「編輯器」讓您在其中執行如撰寫程式碼、執行終止工作階段以及變更 IDE 設定等內容。每個執行個體 (開啟檔案、終止工作階段等等) 都由「標籤」表示。標籤可以分組成「窗格」。標籤顯示在其窗格的邊緣。

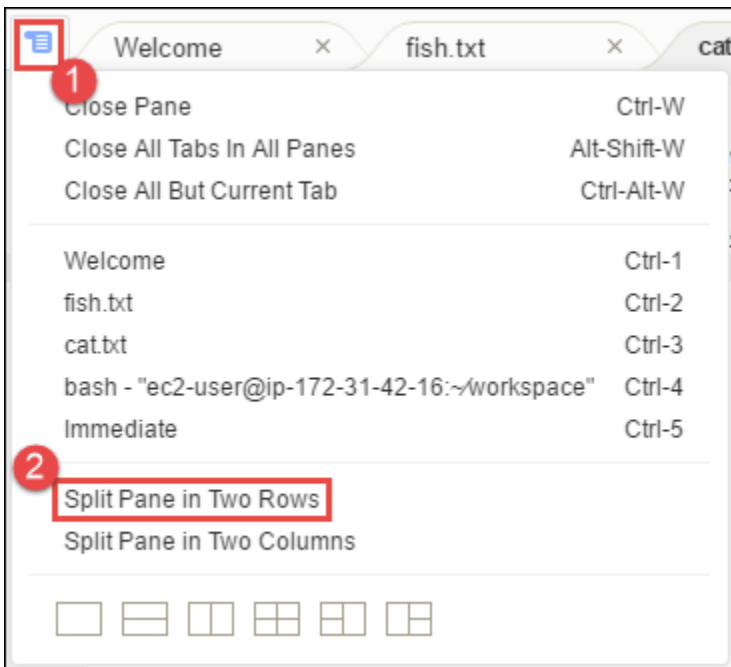


若要顯示或隱藏標籤，請選擇選單列上的 View (檢視)、Tab Buttons (標籤按鈕)。

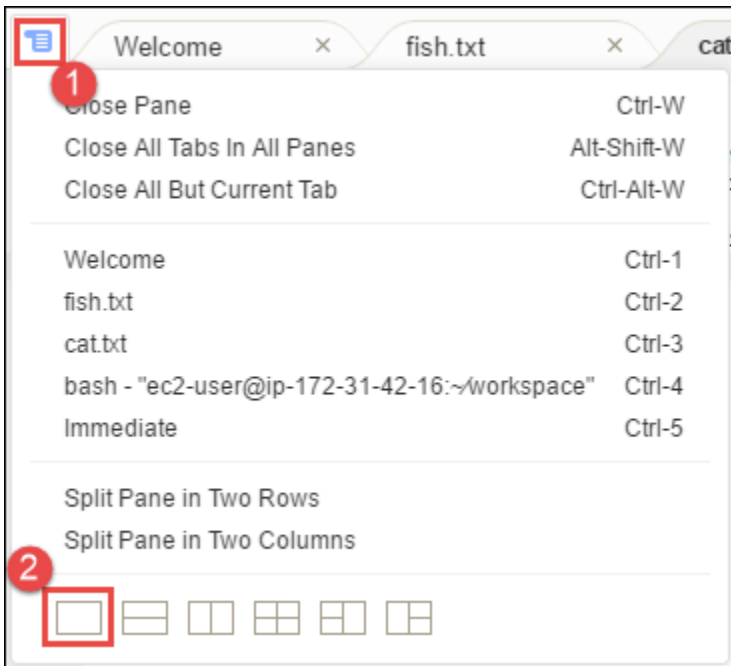
若要開啟新標籤，請選擇標籤列邊緣的 + 圖示。然後選擇一項可用的命令，例如 New File (新增檔案)，如下所示。



若要顯示兩個窗格，請從標籤列邊緣選擇看起來像是下拉式選單的圖示。然後選擇 Split Pane in Two Rows (將窗格分割為兩列)，如下所示。

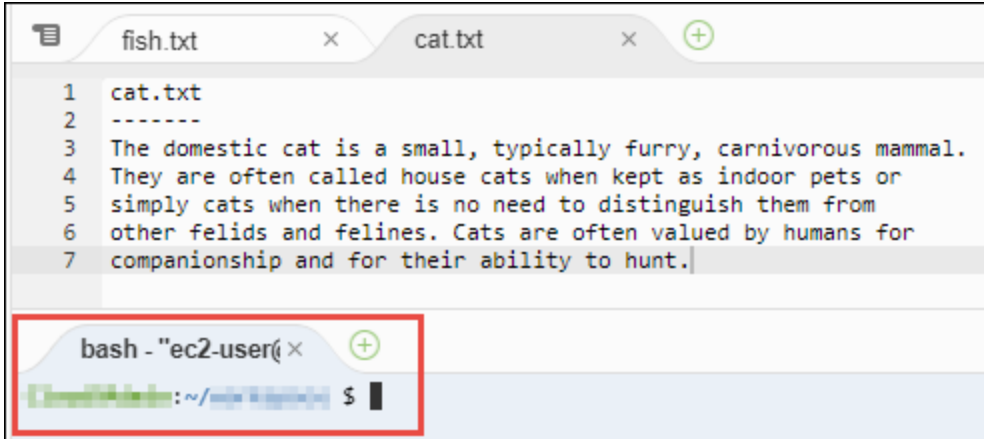


若要返回單一窗格，請再次選擇下拉式選單圖示，然後選擇單一方形圖示，如下所示。



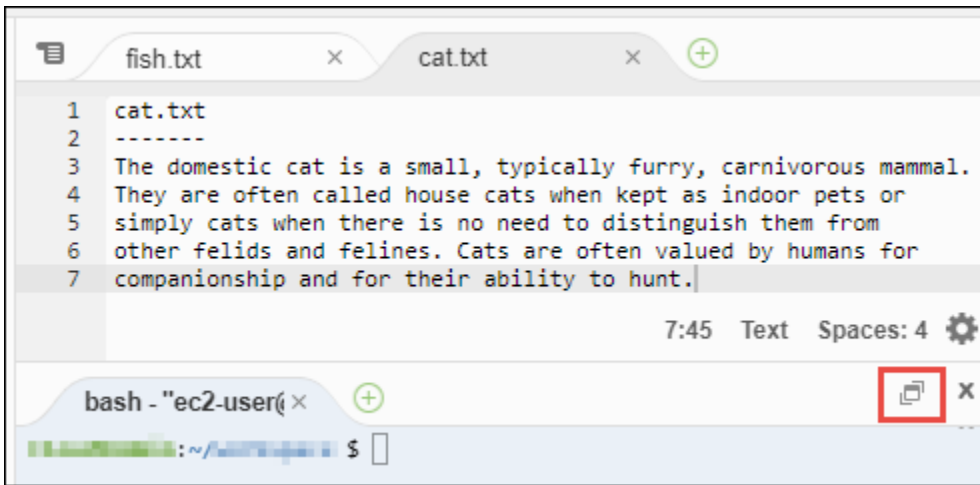
步驟 5：主控台

主控台是建立和管理標籤的替代位置。根據預設，它包含 Terminal (終端機) 標籤，但也可以包含其他類型的標籤。



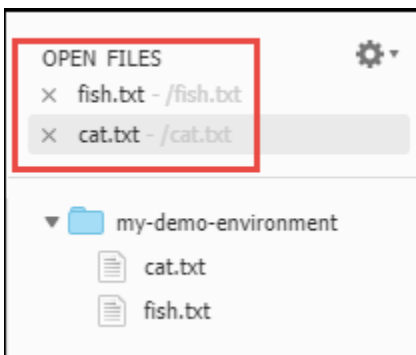
若要顯示或隱藏主控台，請選擇選單列上的 View (檢視)、Console (主控台)。

若要擴展或縮小主控台，請選擇主控台邊緣的調整大小圖示，如下所示。



步驟 6：開啟檔案區段

Open Files (開啟檔案) 區段顯示編輯器中目前已開啟之所有檔案的清單。Open Files (開啟檔案) 是 Environment (環境) 視窗的一部分。

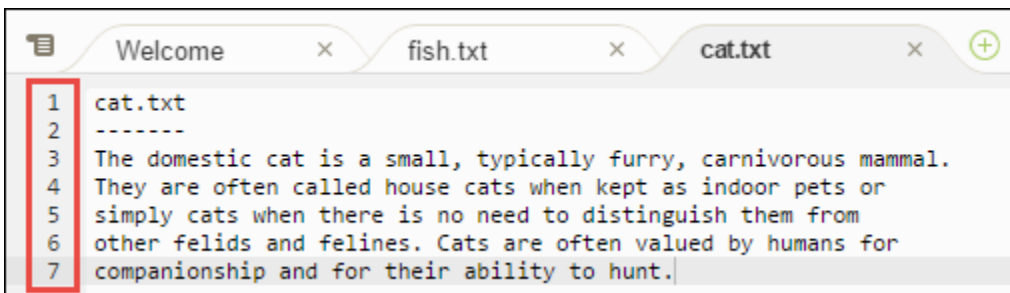


若要開啟或隱藏 Open Files (開啟檔案) 區段，請選擇選單列上的 View (檢視)、Open Files (開啟檔案)。

若要在開啟的檔案之間切換，請從清單中選擇關注的檔案。

步驟 7：裝訂邊

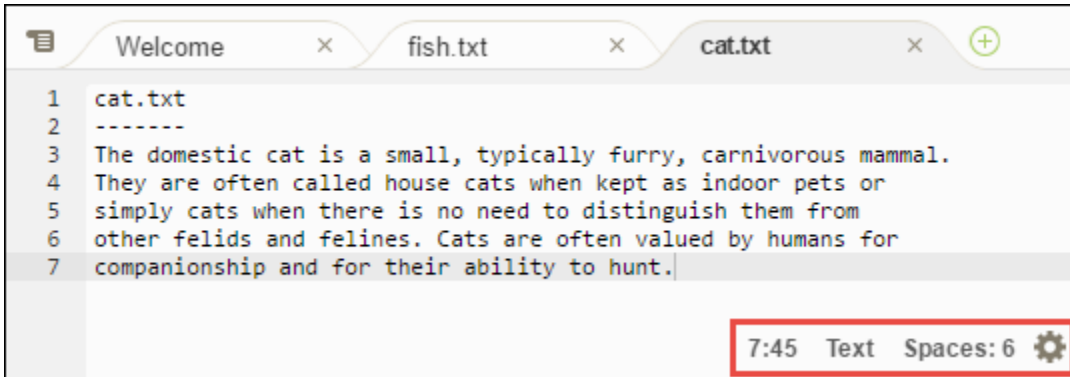
裝訂邊位於編輯器中每個檔案的邊緣，在使用檔案時會顯示行號和關聯的符號等內容。



若要顯示或隱藏裝訂邊，請選擇選單列上的 View (檢視)、Gutter (裝訂邊)。

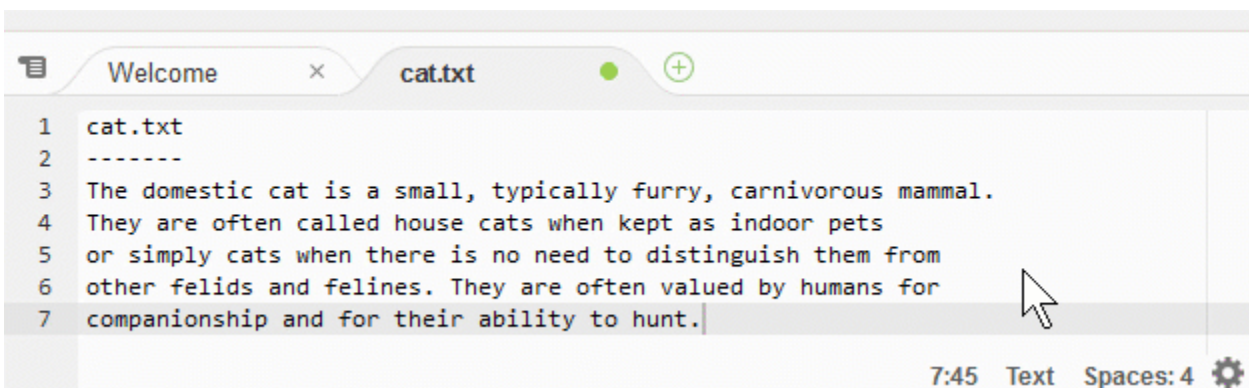
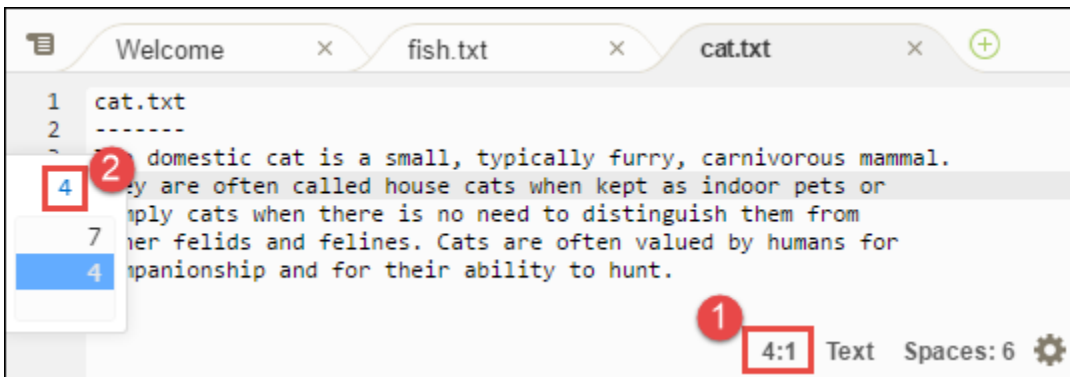
步驟 8：狀態列

狀態列位於編輯器中每個檔案的邊緣，顯示行數和字元數、檔案類型偏好設定、空格和標籤設定以及相關的編輯器設定。

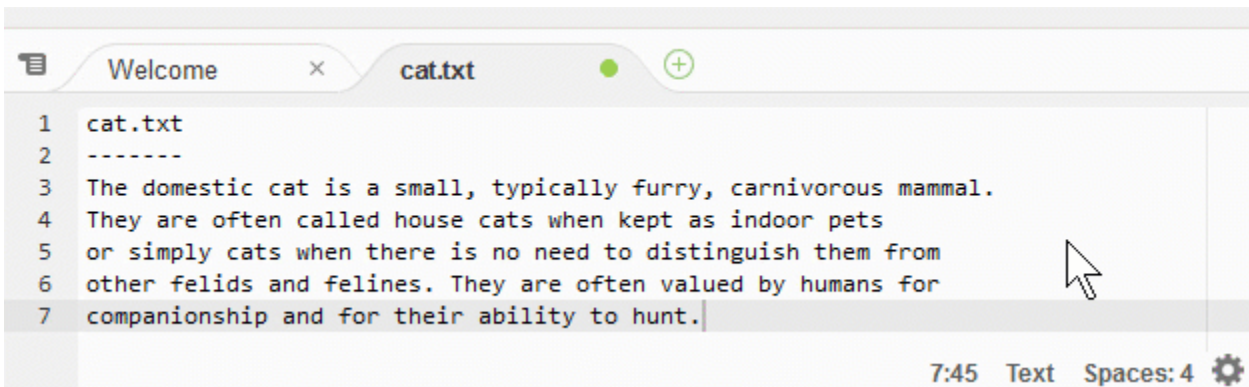
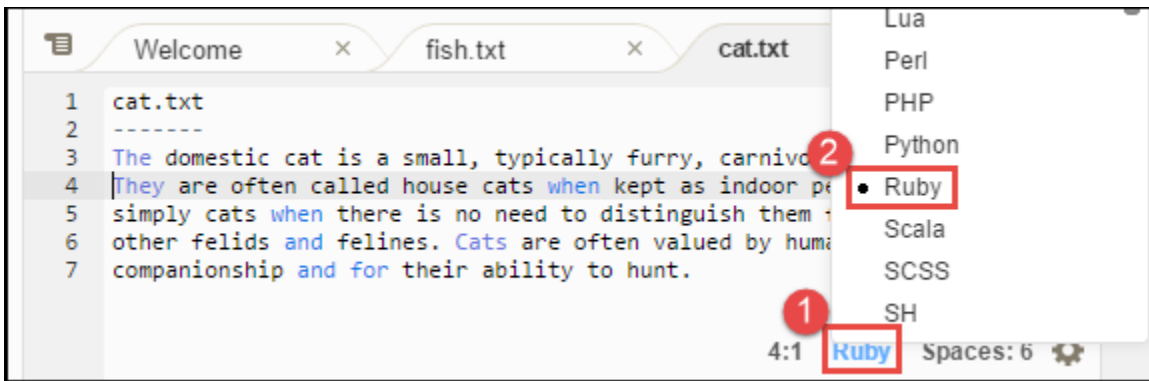


若要顯示或隱藏狀態列，請選擇選單列上的 View (檢視)、Status Bar (狀態列)。

若要移至特定行號，請選擇含有關注檔案的標籤。然後，在狀態列中選擇行號和字元號碼 (看起來像是 7:45)。請輸入行號 (如 4)，然後按下 Enter，如下所示。



若要變更檔案類型偏好設定，請在狀態列中選擇不同的檔案類型。例如，為 `cat.txt` 選擇 Ruby 以查看語法色彩變更。若要返回純文字色彩，請選擇 Plain Text (純文字)，如下所示。



步驟 9：大綱視窗

您可以使用 Outline (大綱) 視窗快速移至特定檔案的位置。

若要顯示或隱藏 Outline (大綱) 視窗和 Outline (大綱) 按鈕，請在選單列上選擇 Window (視窗)、Outline (大綱)。

若要查看 Outline (大綱) 視窗的運作方式，請建立一個名為 `hello.rb` 的檔案。將以下程式碼複製到檔案，然後儲存進行。

```
def say_hello(i)
  puts "Hello!"
  puts "i is #{i}"
end

def say_goodbye(i)
  puts "i is now #{i}"
  puts "Goodbye!"
end
```

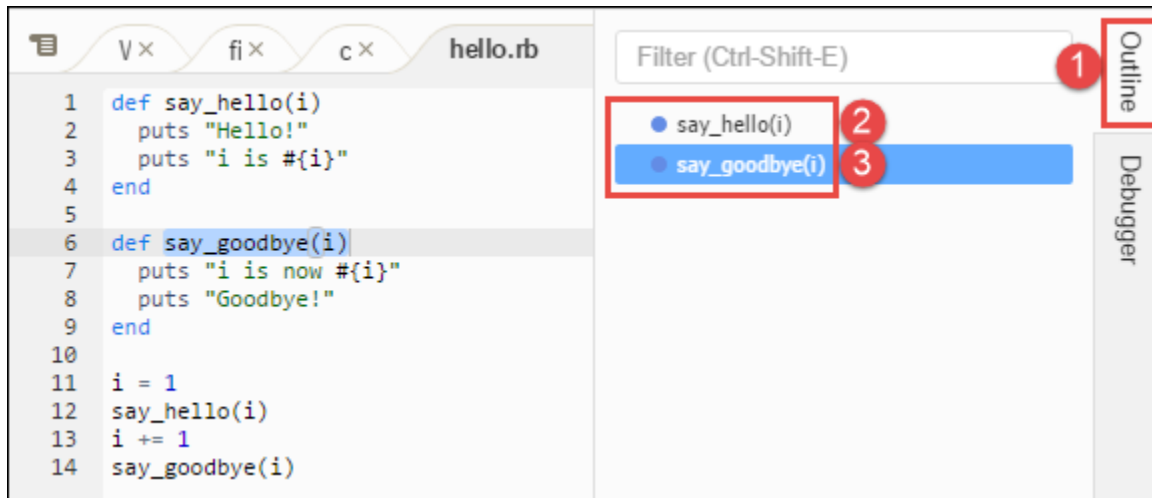
```

i = 1
say_hello(i)
i += 1
say_goodbye(i)

```

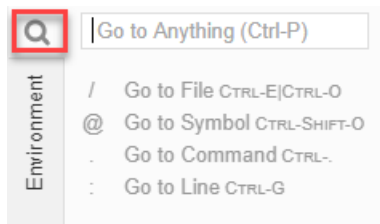
若要顯示或隱藏 Outline (大綱) 視窗的內容，請選擇 Outline (大綱) 按鈕。

在 Outline (大綱) 視窗中選擇 say_hello(i)，然後選擇 say_goodbye(i)，如下所示。



步驟 10：執行視窗

您可以使用 Go (執行) 視窗在編輯器中開啟檔案、移至符號的定義、執行命令，或移至編輯器之作用中檔案中的命令列。



若要顯示 Go (執行) 視窗的內容，請選擇 Go (執行) 按鈕 (放大鏡圖示)。

若要顯示或隱藏 Go (執行) 視窗和 Go (執行) 按鈕，請在選單列上選擇 Window (視窗)、Go (執行)。

在 Go (執行) 視窗開啟時，您可以：

- 輸入正斜線 (/)，隨後跟著部分或所有檔案名稱。在顯示的相符的檔案清單中，選擇要在編輯器中開啟的檔案。例如，輸入 /fish 清單 fish.txt，同時輸入 /.txt 清單 fish.txt 和 cat.txt。

Note

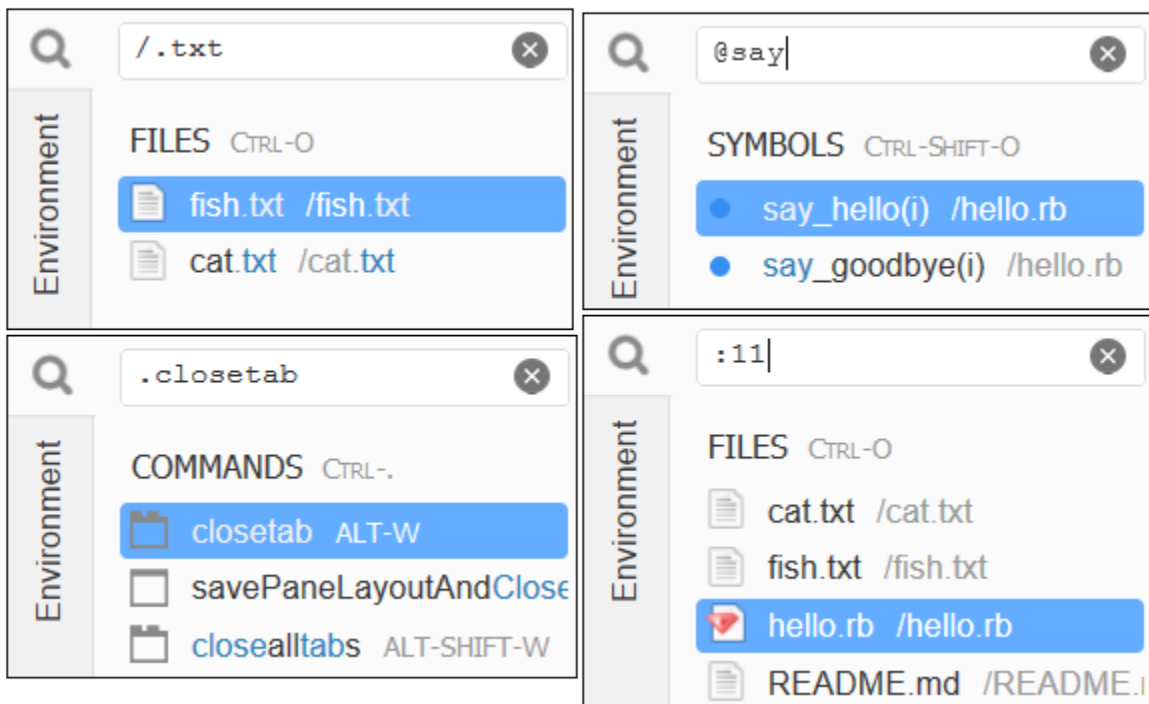
檔案搜尋範圍僅限於 Environment (環境) 視窗中的非隱藏檔案和非隱藏資料夾。

- 輸入 @ 符號 (@)，隨後跟著符號的名稱。在顯示的相符的符號清單中，選擇要在編輯器中前往的符號。例如，在編輯器中的 hello.rb 檔案在開啟和作用中狀態時，輸入 @hello 以列出 say_hello(i)，或輸入 @say 以同時列出 say_hello(i) 和 say_goodbye(i)。

Note

如果編輯器中作用中的檔案屬於受支援語言專案的一部分，那麼符號搜尋範圍僅限於目前的專案。否則，符號搜尋範圍只限於編輯器中的作用中檔案。如需更多詳細資訊，請參閱 [增強的TypeScript支援與功能](#)。

- 輸入點 (.)，隨後跟著命令的名稱。在顯示的命令清單中，選擇命令來執行。例如，輸入 .closetab，然後按 Enter 關閉編輯器目前的標籤。如需可用命令的清單，請參閱 [AWS Cloud9 整合開發環境 \(IDE\) 的命令參考](#)。
- Type 冒號 (:)，隨後跟著要前往之編輯器作用中檔案的行號數字。例如，編輯器之 hello.rb 檔案在開啟和作用中狀態時，輸入 :11 即可前往該檔案的第 11 行。



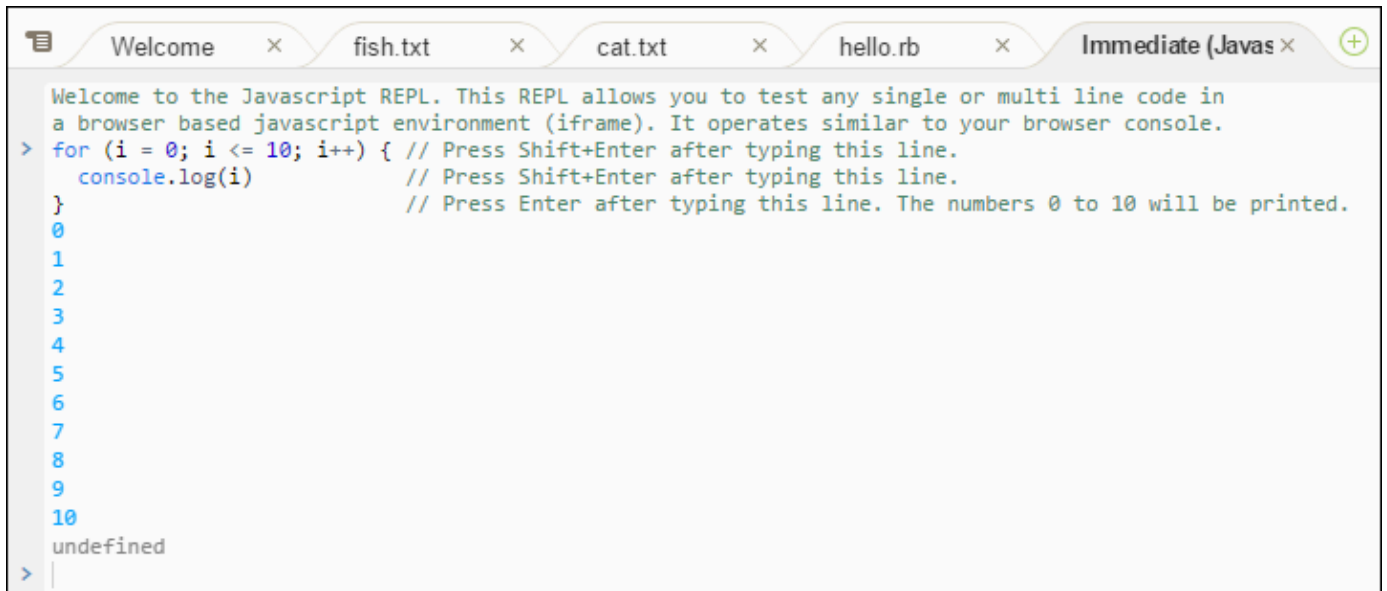
若要根據目前的鍵盤模式和作業系統查看上述個別動作的按鍵繫結關係，請在選單列的 Go (執行) 選單中查看個別可用的 Go To (前往) 命令。

步驟 11：即時運算標籤

Immediate (即時運算) 標籤可讓您測試 JavaScript 程式碼的小片段。若要查看 Immediate (即時運算) 標籤的運作方式，請執行下列動作。

1. 請在狀態列上選擇 Window (視窗)、New Immediate Window (新增即時運算視窗) 以開啟 Immediate (即時運算) 標籤。
2. 在 Immediate (即時運算) 標籤中執行一些程式碼。若要嘗試，請將下列程式碼輸入視窗中，在輸入行 1 之後和行 2 之後各按下 Shift-Enter。在行 3 之後按下 Enter。(如果您在輸入行 1 或行 2 之後按下 Enter 而不是 Shift-Enter，程式碼會比您希望的還更早執行)。

```
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
>
```

步驟 12：程序清單

Process List (程序清單) 顯示所有執行中的程序。您可以停止或甚至強制停止您不想再執行的程序。若要查看 Process List (程序清單) 視窗的運作方式，請執行下列動作。

1. 請在選單列上選擇 Tools (工具)、Process List (程序清單) 來顯示 Process List (程序清單)。
2. 尋找程序。請在 Process List (程序清單) 中輸入程序名稱。
3. 停止或強制停止程序。在程序清單中選擇程序，然後選擇 Kill (終止) 或 Force Kill (強制終止)。

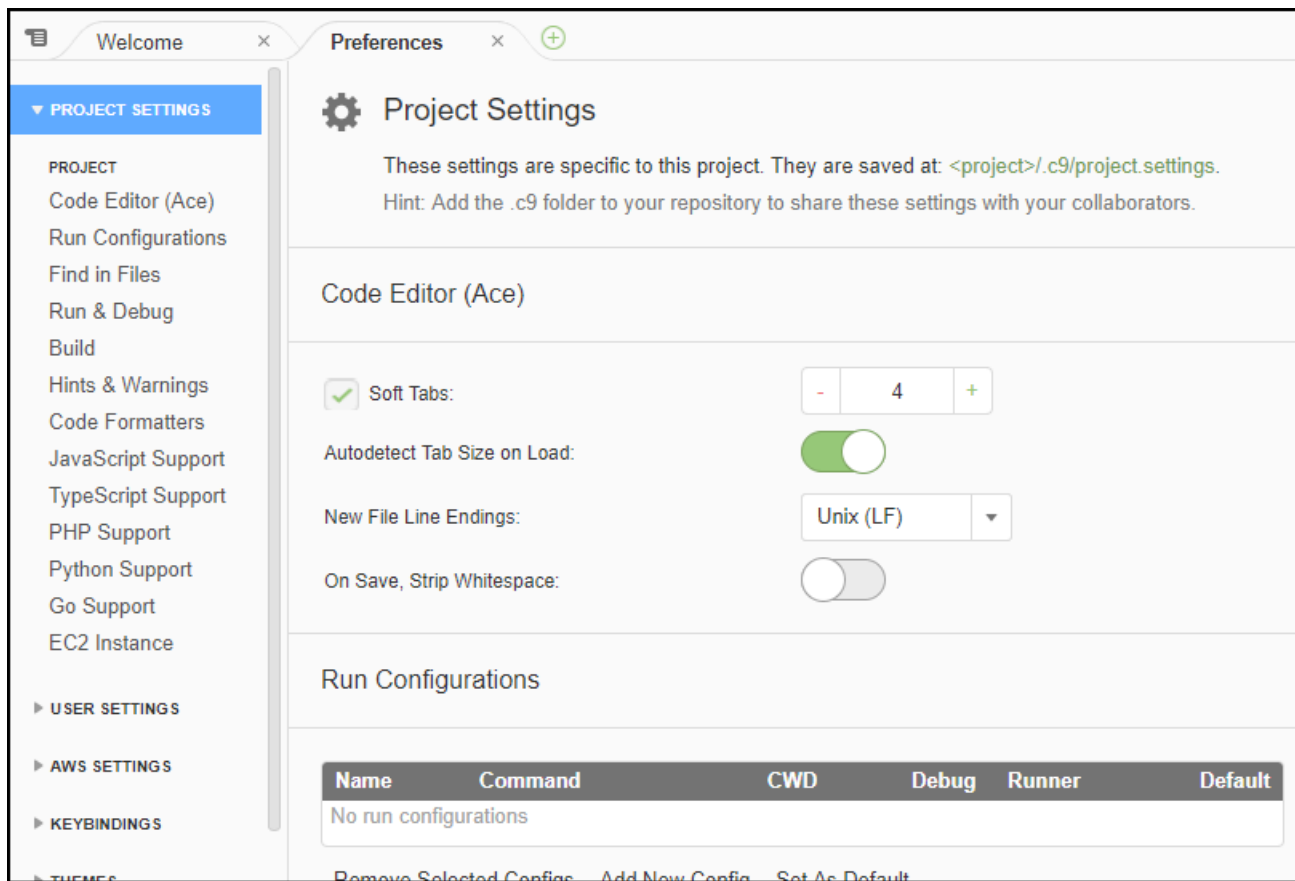
Process Name	CPU	MEM	Process Time	PID	User
kworker/0:1H	0.0%	0.0%	0:00	1491	root
init	0.0%	0.4%	0:00	1	root
ksoftirqd/0	0.0%	0.0%	0:00	3	root
kworker/0:0	0.0%	0.0%	0:00	4	root
kworker/0:0H	0.0%	0.0%	0:00	5	root
rcu_sched	0.0%	0.0%	0:00	7	root
rcu_bh	0.0%	0.0%	0:00	8	root
migration/0	0.0%	0.0%	0:00	9	root
kdevtmpfs	0.0%	0.0%	0:00	10	root
netns	0.0%	0.0%	0:00	11	root
perf	0.0%	0.0%	0:00	12	root
kworker/u30:1	0.0%	0.0%	0:00	13	root
xenwatch	0.0%	0.0%	0:00	15	root
kworker/u30:2	0.0%	0.0%	0:00	17	root

步驟 13：偏好設定

「偏好設定」包含下列設定。

- 僅適用於目前環境的設定，例如是否在編輯器中使用軟標籤、要忽略的檔案類型以及 PHP 和 Python 等語言的程式碼完成行為。
- 您在每個環境中的使用者設定，例如色彩、字型和編輯器行為。
- 您的按鍵繫結關係，例如您希望使用哪些快速鍵組合來使用檔案和編輯器。
- IDE 的整體主題。

若要顯示偏好設定，請在選單列上選擇 AWS Cloud9、Preferences (偏好設定)。會顯示類似下列內容。



步驟 14：終端機

您可以在 IDE 中執行一或多個「終端機」工作階段。若要啟動終端機工作階段，請在選單列上選擇 Window (視窗)、New Terminal (新增終端機)。或者，選擇 Console (主控台) 標籤旁的「plus」圖示，然後選擇 New Terminal (新增終端機)。

您可以嘗試在終端機執行命令。例如，在終端機輸入 `echo $PATH`，然後按下 Enter 來列印 PATH 環境變數的值。

您也可以嘗試執行額外的命令。例如，請嘗試如下命令。

- **pwd** 列印目前目錄的路徑。
- **aws --version** 列印關於 AWS CLI 的版本資訊。
- **ls -l** 列印關於目前目錄的資訊。



```
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
```

(14 Bytes) 6:19 Ruby Spaces: 2

```
bash - "ip-172-31"
Cloud9Admin:~/environment $
```

步驟 15：除錯器視窗

您可以使用 Debugger (除錯器) 視窗來除錯您的程式碼。例如，您可以逐一執行部分程式碼、隨時間觀察變數的值、探索呼叫堆疊。

Note

此程序類似於 [IDE 基本教學課程](#) 中的 [步驟 2：IDE 基本導覽](#)。

若要顯示或隱藏 Debugger (除錯器) 視窗和 Debugger (除錯器) 按鈕，請在選單列上選擇 Window (視窗)、Debugger (除錯器)。

在本教學中，您可以透過執行下列動作，試驗 Debugger (除錯器) 視窗和一些 JavaScript 程式碼。

1. 透過在終端機工作階段執行下列命令，藉此在環境中檢查 Node.js 安裝：**node --version**。如果已安裝 Node.js，將會在輸出中顯示 Node.js 版本號碼，您可以直接跳到此程序的步驟 3 (「撰寫一些 JavaScript 程式碼...」)。

2. 如果您需要安裝 Node.js，請執行下列動作。

- a. 執行以下兩個命令，一次一個，以確定您的環境擁有最新的更新，然後下載 Node Version Manager (nvm)。(nvm 是一種簡單的 Bash shell 指令碼，適用於安裝和管理 Node.js 版本。請參閱 GitHub 上的 [Node Version Manager](#)。)

針對 Amazon Linux：

```
sudo yum -y update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh |
  bash
```

針對 Ubuntu Server：

```
sudo apt update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh |
  bash
```

- b. 使用文字編輯器來更新您的 shell 描述檔檔案 (例如，`~/.bashrc`) 以便能夠載入 nvm。例如，在 IDE 的 Environment (環境) 視窗中選擇齒輪圖示，然後選擇 Show Home in Favorites (在我的最愛中顯示首頁)。重複此步驟並選擇 Show Hidden Files (顯示隱藏的檔案)。
- c. 開啟 `~/.bashrc` 檔案。
- d. 在檔案尾端輸入或貼上下列程式碼，以啟用 nvm 載入。

針對 Amazon Linux：

```
export NVM_DIR="/home/ec2-user/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm.
```

針對 Ubuntu Server：

```
export NVM_DIR="/home/ubuntu/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm.
```

- e. 儲存檔案。
- f. 關閉該終端機工作階段，並開始新的工作階段。然後執行以下命令來安裝最新版的 Node.js。

```
nvm install node
```

3. 撰寫一些 JavaScript 程式碼來除錯。例如，建立一個檔案並將下列程式碼新增至該檔案，然後將其儲存為 `hello.js`。

```
var i;

i = 10;

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

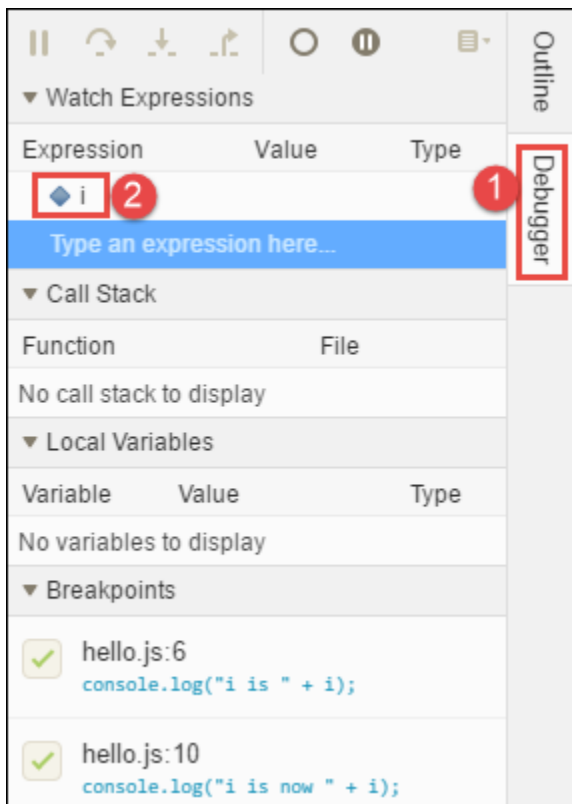
i += 1;

console.log("i is now " + i);
console.log("Goodbye!");
```

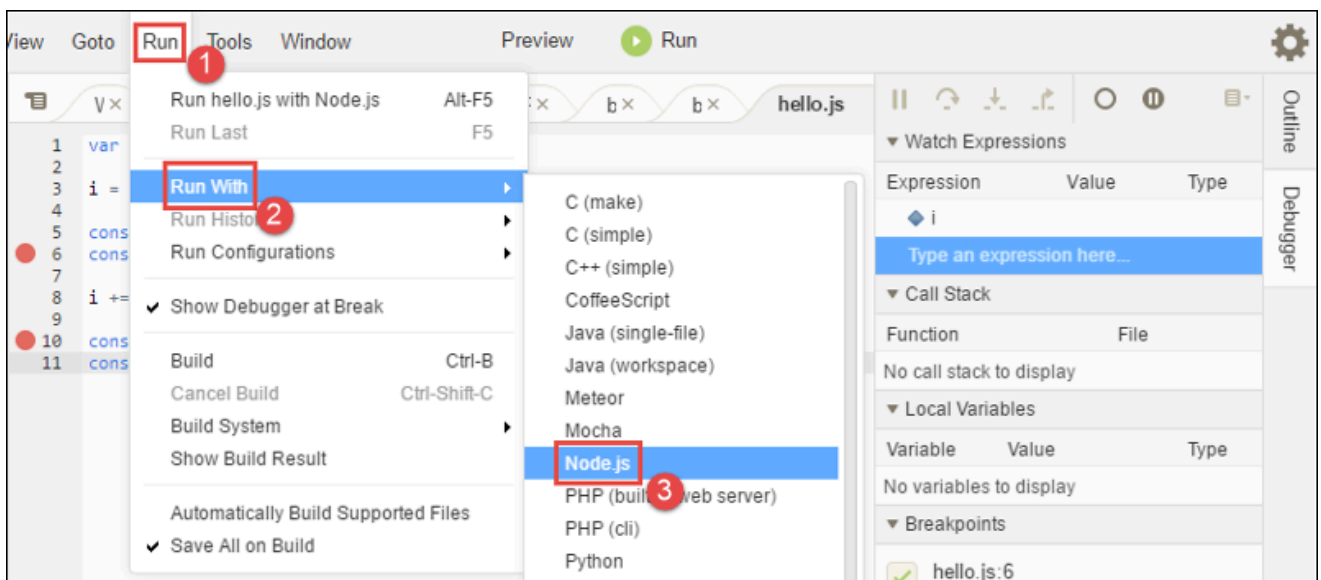
4. 新增一些中斷點至程式碼。例如，在裝訂邊中選擇行 6 和行 10 旁的邊界。在這些行號旁會顯示紅色圓圈，如下所示。



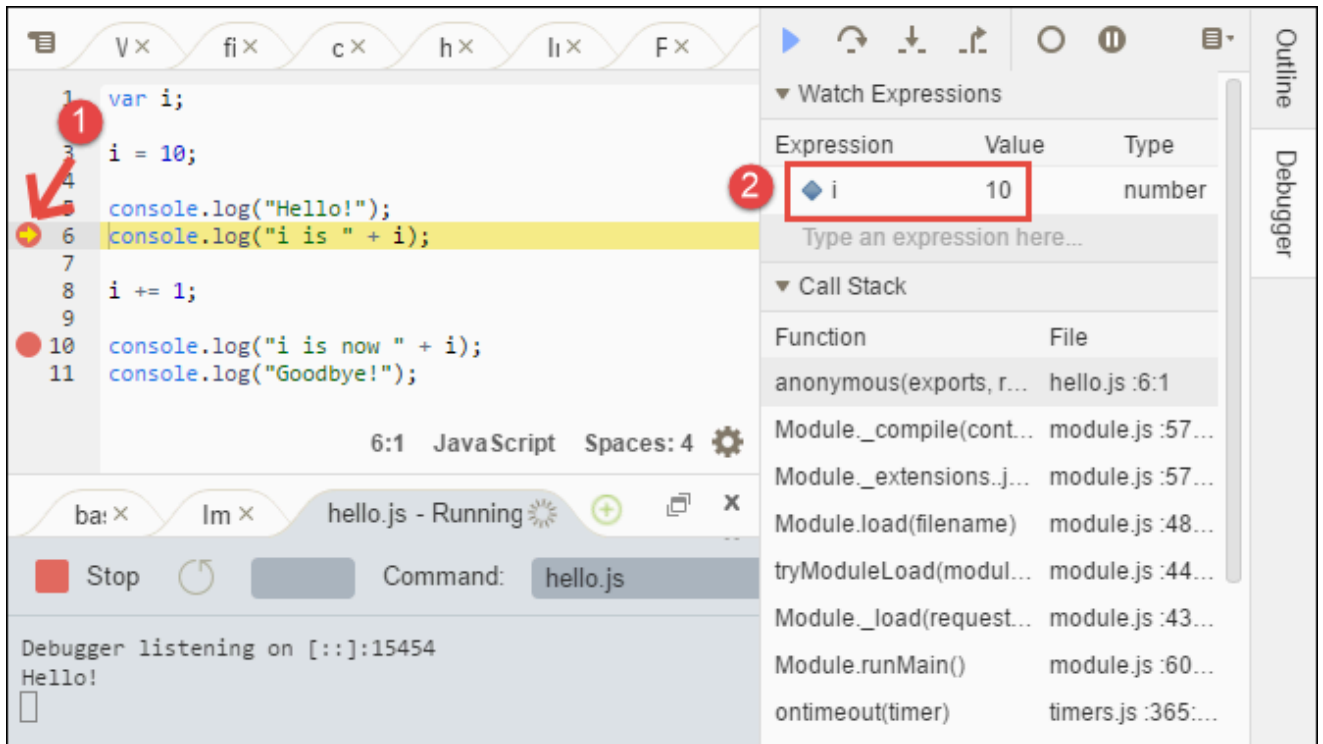
5. 現在您已準備好除錯 JavaScript 程式碼。若要進行這項試驗，請執行下列動作。
 - a. 若要顯示或隱藏 Debugger (除錯器) 視窗的內容，請選擇 Debugger (除錯器) 按鈕，如下一個步驟所示。
 - b. 在執程式碼的同時，監看名為 `i` 之變數的值。在 Debugger (除錯器) 視窗中，為 Watch Expressions (監看運算式) 選擇 Type an expression here (在此處輸入運算式)。輸入字母 `i`，然後按下 Enter，如下所示。



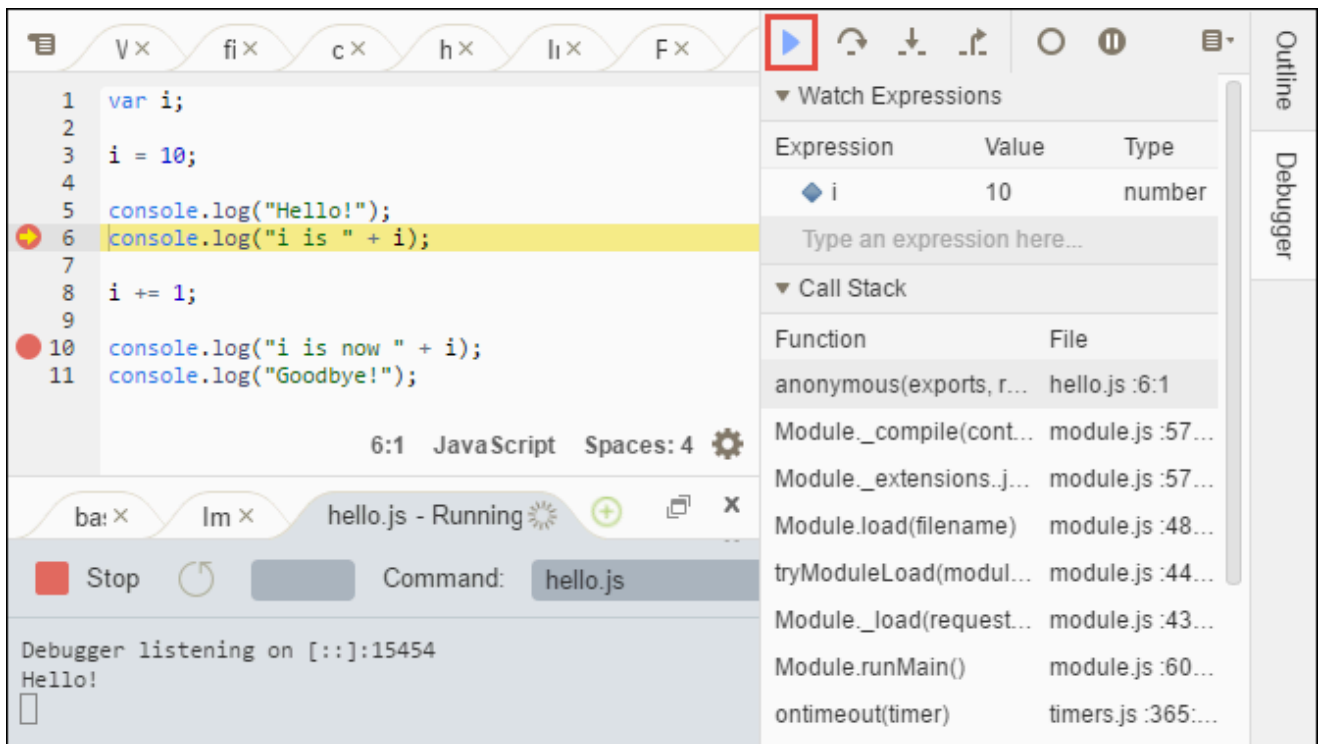
- c. 開始執程式碼。選擇 Run (執行)、Run With (執行工具)、Node.js，如下所示。



- d. 程式碼在行 6 暫停執行。Debugger (除錯器) 視窗會在 Watch Expressions (監看表達式) 中顯示 `i` 的值，此值目前為 10。



- e. 在 Debugger (除錯器) 視窗中，請選擇 Resume (繼續)，即為藍色箭頭圖示，如下所示。



- f. 程式碼在行 10 暫停執行。Debugger (除錯器) 視窗現在顯示 `i` 的新值，目前為 11。
- g. 請再次選擇 Resume (繼續)。程式碼會執行到尾端。輸出將列印到主控台的 `hello.js` 標籤上，如下所示。

The screenshot displays the AWS Cloud9 IDE interface. The editor shows a JavaScript file named `hello.js` with the following code:

```
1 var i;
2 i = 10;
3 console.log("Hello!");
4 console.log("i is " + i);
5 i += 1;
6 console.log("i is now " + i);
7 console.log("Goodbye!");
```

The debugger is paused at line 10. The console output is:

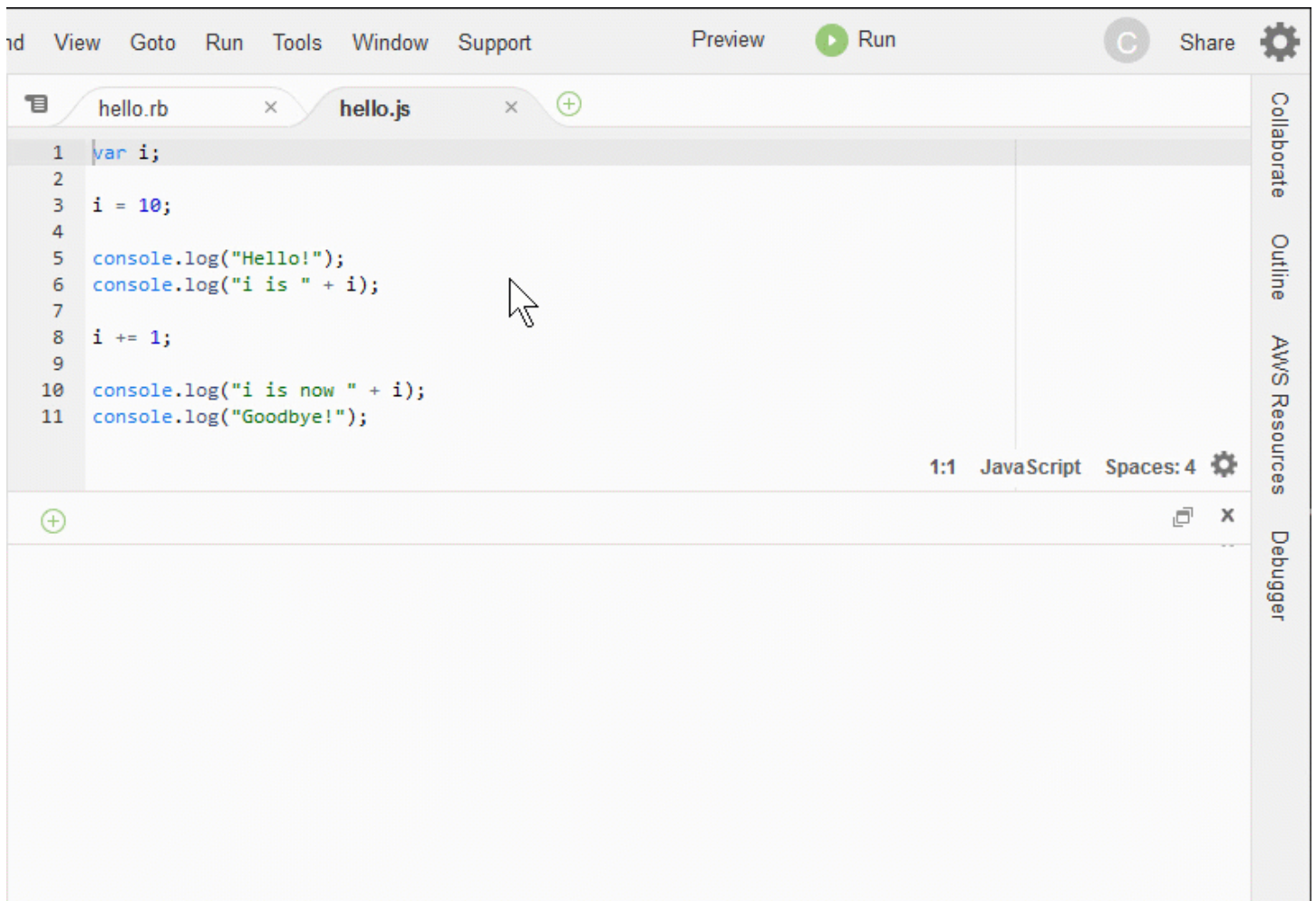
```
Debugger listening on [::]:15454
Hello!
i is 10
i is now 11
Goodbye!
Process exited with code: 0
```

The Watch Expressions panel shows the variable `i` with a value of `11` and a type of `number`.

The Breakpoints panel shows two active breakpoints:

- hello.js:6 `console.log("i is " + i);`
- hello.js:10 `console.log("i is now " + i);`

將您的結果與下列項目比較。



結論

⚠ Warning

請記住，使用 AWS Cloud9 開發環境可能會導致您的 AWS 帳戶產生費用。如果您使用 EC2 環境，則其中包括 Amazon EC2 可能產生的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

父區段 ([使用 IDE](#)) 中有其他主題可供您探索。不過，當您完成 AWS Cloud9 IDE 的概覽，且不再需要環境，請務必將它及其關聯資源刪除，如 [刪除環境](#) 中所述。

AWS Cloud9 整合開發環境 (IDE) 的語言支援

AWS Cloud9 IDE 支援多種程式設計語言。下表列出支援的語言及其支援程度。

語言	語法醒目提示 ¹	執行 UI ²	大綱檢視	程式碼提示與檢查	程式碼完成	偵錯 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 指令碼	✓	✓	✓	✓	✓ ⁵	
TypeScript ⁹	✓	✓	✓	✓	✓	

備註

- ¹ AWS Cloud9 IDE 另為多種其他語言提供語法醒目提示。如需完整清單，請從 IDE 的選單列選擇 View、Syntax (檢視、語法)。
- ² 凡是標註 ✓ 的語言皆可透過按下按鈕執行程式或指令碼，而不需要使用命令列。若所列的語言未標註 ✓ 或未顯示於 IDE 的 Run、Run With (執行、執行工具) 選單列，您則可為該語言建立執行器。如需相關說明，請參閱[建立建置器或執行器](#)。
- ³ 您可以使用 IDE 的內建工具對標註 ✓ 的語言進行程式偵錯或指令碼偵錯。如需相關說明，請參閱[程式碼偵錯](#)。
- ⁴ 此功能對其標註的語言處於實驗性狀態。其仍未完全實作，且無專文介紹或不受支援。
- ⁵ 此功能對其標註的語言僅支援本機函數。
- ⁶ 增強的 Java SE 11 功能支援可在具有 2 GiB 或更多記憶體之 AWS Cloud9 EC2 開發環境中啟用。如需詳細資訊，請參閱[增強對 Java 開發的支援](#)。
- ⁷ 如要指定 AWS Cloud9 用於完成自訂 PHP 程式碼的路徑，請從 AWS Cloud9 IDE 的 Preferences (偏好設定) 開啟 Project、PHP Support、Enable PHP code completion (專案、PHP 支援、啟用 PHP 程式碼完成) 設定，然後於 Project、PHP Support、PHP Completion Include Paths (專案、PHP 支援、PHP 完成 Include 路徑) 設定中加入自訂程式碼的路徑。
- ⁸ 如要指定 AWS Cloud9 用於完成自訂 Python 程式碼的路徑，請從 AWS Cloud9 IDE 的 Preferences (偏好設定) 開啟 Project、Python Support、Enable Python code completion (專案、Python 支援、啟用 Python 程式碼完成) 設定，然後於 Project、Python Support、PYTHONPATH (專案、Python 支援、PYTHONPATH) 設定中加入自訂程式碼的路徑。
- ⁹ AWS Cloud9 IDE 在語言專案的內容中為某些程式設計語言提供額外的支援，例如 TypeScript (AWS Cloud9 IDE 中支援的 3.7.5 版)。如需使用 UI 畫布的詳細資訊，請參閱[使用語言專案](#)。

AWS Cloud9 整合式開發環境 (IDE) 中支援的程式設計語言版本

下表概述 AWS Cloud9 IDE 中特定 AMI 支援哪些版本的程式設計語言。Ubuntu 18 在 2023 年停產，因此無法在中更新編程語言版本。AWS Cloud9

語言	Amazon Linux 2023	Amazon Linux 2	Ubuntu	UBUNTU
Python3	3.9	3.8	3.6	3.10
TypeScript	3.7.5	3.7.5	3.7.5	3.7.5

語言	Amazon Linux 2023	Amazon Linux 2	Ubuntu	UBUNTU
PHP	8.2	8.2	7.2	8.1
Ruby	3.2	3.0	3.0	3.2
Java	11, 17	11	11	11, 17
蟒蛇 2	N/A	2.7	N/A	N/A
C++*	23	17	17	23
Go	1.20	1.20	1.9	1.21
CoffeeScript	2.7	2.7	2.7	2.7

* 您可以運行以下命令，使用您要使用的編程語言版本編譯 C ++ 文件：

```
g++ -std=c++[version-number] "$file" -o "$file.o"
```

AWS Cloud9 整合開發環境 (IDE) 中的增強語言支援

AWS Cloud9 提供增強的支援，以改善您在使用如下語言編碼時的開發體驗。

- Java：延伸模組允許提供諸如程式自動補全、錯誤檢查、內容特定動作和偵錯選項等功能。
- Typescript：語言專案提供了針對 TypeScript 的增強生產力功能的存取權。

主題

- [增強對 Java 開發的支援](#)
- [增強的TypeScript支援與功能](#)

增強對 Java 開發的支援

AWS Cloud9 提供增強的語言支援，以改善您在使用 Java 開發時的體驗。重點的生產力功能包括程式碼自動補全、錯誤檢查、code lenses (程式碼濾鏡) 和偵錯選項 (如中斷點和逐步執行)。

⚠ Important

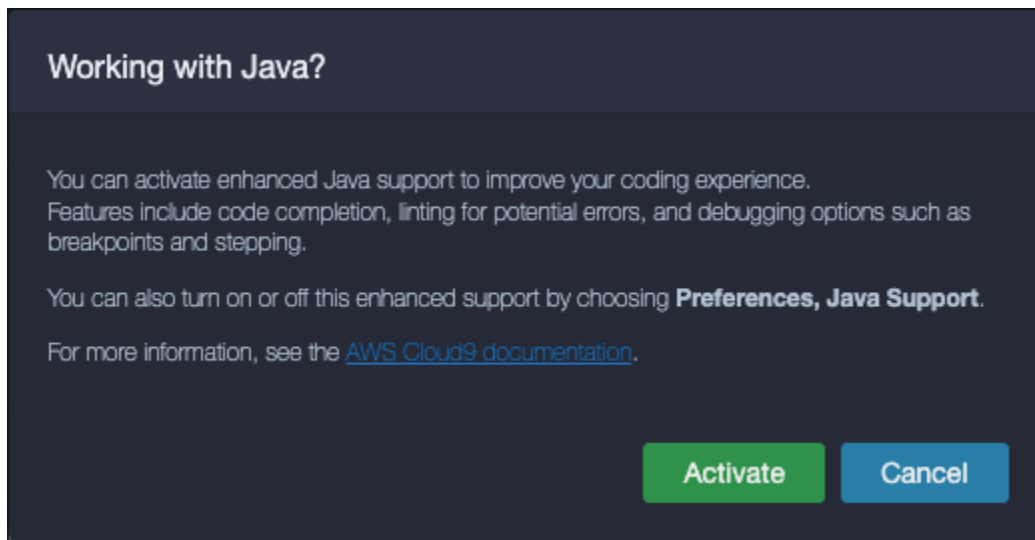
增強的生產力功能僅適用於連接至 Amazon EC2 執行個體的 AWS Cloud9 開發環境。此外，為了確保使用針對 Java 的增強語言支援時得到最佳的 IDE 體驗，為 AWS Cloud9 環境提供支援的 Amazon EC2 運算執行個體需要 2 GB 或更多的記憶體。如果 AWS Cloud9 偵測到您的 EC2 運算執行個體沒有足夠的 RAM，則不會為您提供啟用 Java 增強功能的選項。

啟用和自訂增強的 Java 支援

如果符合下列條件，會自動顯示啟用 Java 增強支援的選項：

- 您的 AWS Cloud9 環境連接到具有 2 GB 或更多記憶體的 Amazon EC2 執行個體。
- 您正在使用與 Java 開發相關聯的檔案。AWS Cloud9 會檢查以下檔名和副檔名：`*.java`、`*.gradle` (與 Gradle 建置工具相關聯) 和 `pom.xml` (與 Apache Maven 建置工具相關聯)。
- 您正在 2020 年 12 月 11 日之後建立的 AWS Cloud9 環境中操作。目前，在此日期之前建立的開發環境中無法使用 Java 生產力功能。

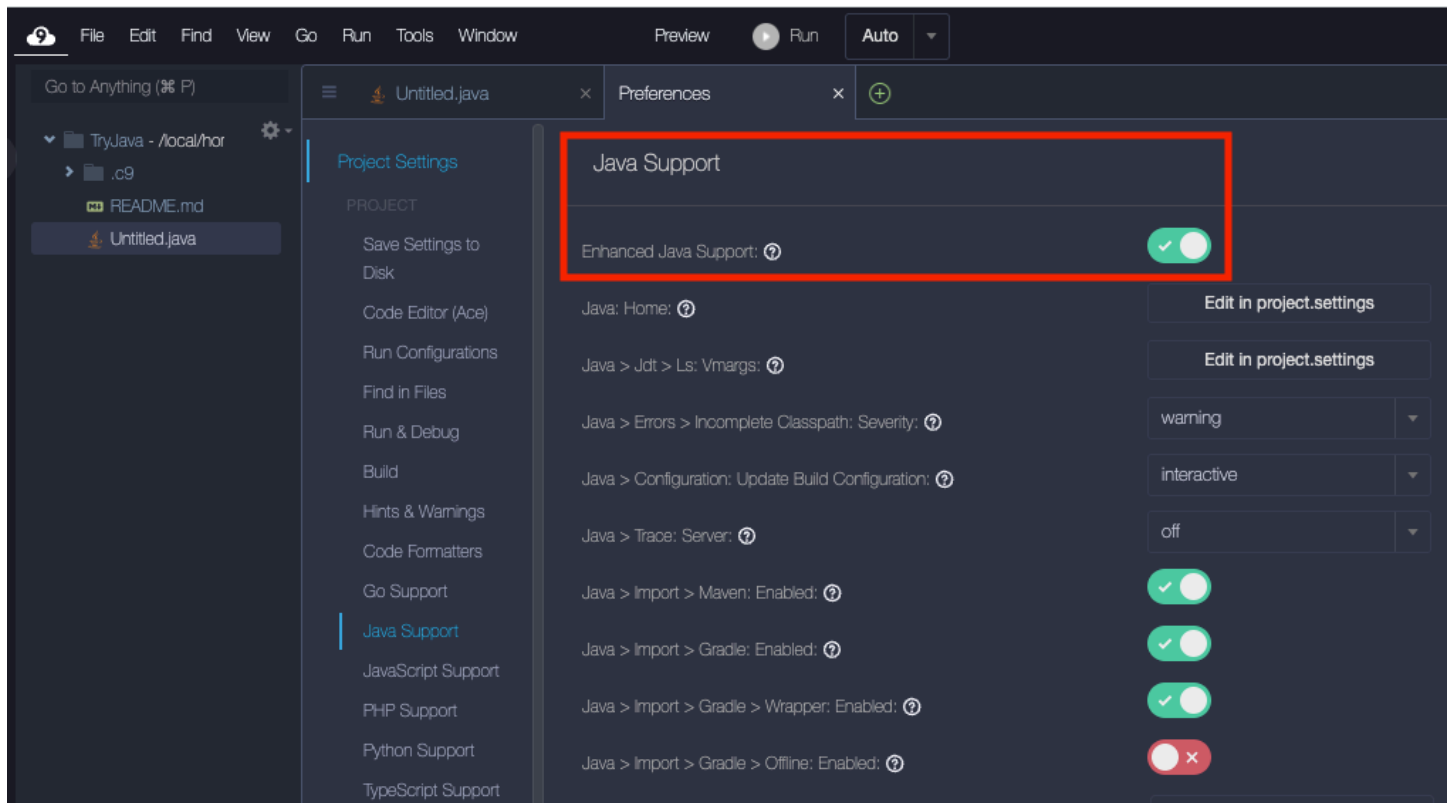
如果滿足這些條件，系統將顯示一個對話方塊，詢問您是否要啟用用於編寫程式碼和除錯 Java 的額外生產力功能。如果選擇 **Activate** (啟用)，則可以開始在 IDE 中使用這些功能。



Note

在您建立 AWS Cloud9 環境時所啟動的 Amazon EC2 執行個體中，已經安裝好 Amazon Coretto 11。Amazon Coretto 是 Open Java Development Kit (OpenJDK) 的免費、多平台的生產就緒分佈。這表示您可以在中開始開發和執行 Java 應用程式AWS Cloud9 out-of-the-box。

您也可以使用 AWS Cloud9 界面手動啟用與停用增強語言和偵錯支援。選擇 Preferences (偏好設定)、Java Support (Java 支援)、Enhanced Java Support (增強型 Java 支援)。



AWS Cloud9 中的增強 Java 開發支援是由兩個 IDE 延伸模組提供：

- Red Hat 提供的 Java(TM) 語言支援
- Java 偵錯器

AWS Cloud9 界面可讓您存取自訂這些延伸模組效能的多種設定。若要變更擴充設定，請選擇 Preferences (偏好設定)、Java Support (Java 支援)。

如需這些設定的詳細資訊，請參閱已安裝的版本。ReadMe擴展程序中的頁面GitHub存儲庫：

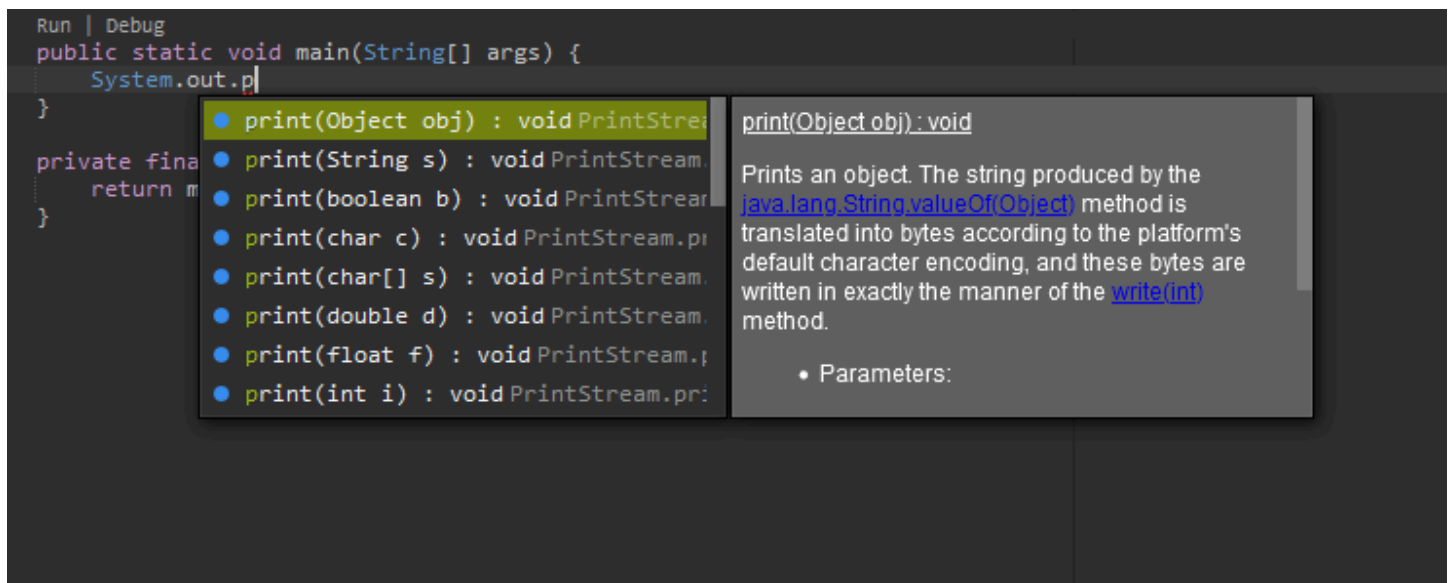
- [Red Hat 提供的 Java\(TM\) 語言支援](#)
- [Java 偵錯器](#)

功能特色

啟用增強的 Java 支援後，您可以使用一系列提高生產力的功能。

程式碼完成

藉助程式碼自動補全，編輯器會根據您鍵入的程式碼提供內容感知的建議。例如，如果在物件名稱後鍵入點 (「.」) 運算子，編輯器將顯示該物件可用的方法或屬性。



Code lenses (程式碼濾鏡)

程式碼濾鏡可讓您直接在原始碼中存取內容特定的動作。對於 Java 開發，程式碼濾鏡可讓您執行和偵錯特定的方法以方便進行單元測試。

```
public class App
{
    private final String message = "Hello World!";

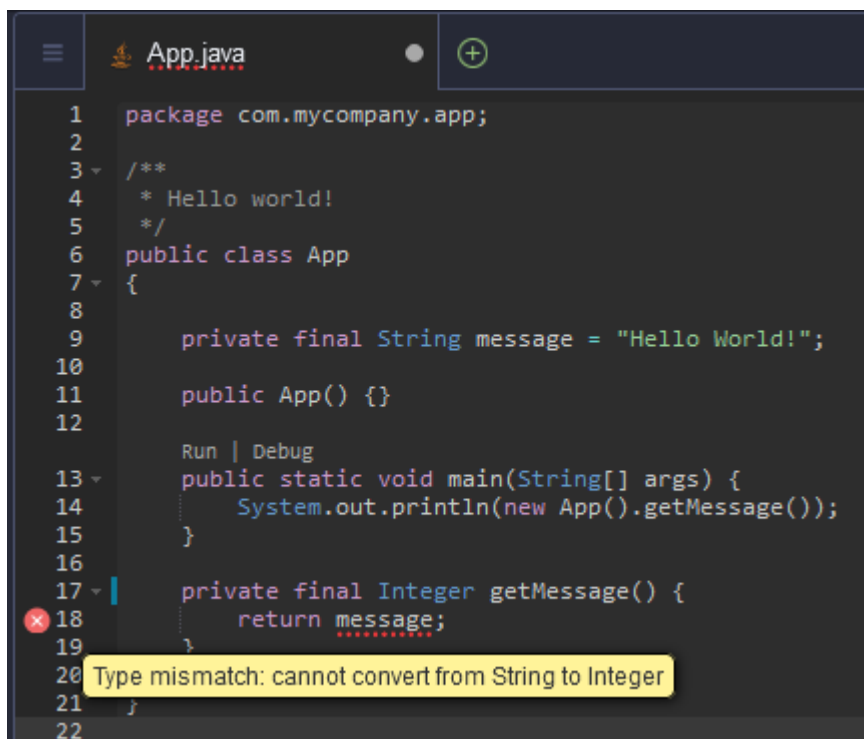
    public App() {}

    Run | Debug
    public static void main(String[] args) {
        System.out.println(new App().getMessage());
    }

    private final String getMessage() {
        return message;
    }
}
```

程式碼檢查

程式碼檢查描述了編輯器如何在建置程式碼之前反白顯示程式碼中的潛在錯誤。例如，如果您嘗試使用未初始化的變數，或嘗試將值指派給需要不同類型的變數，檢查工具就會啟動。

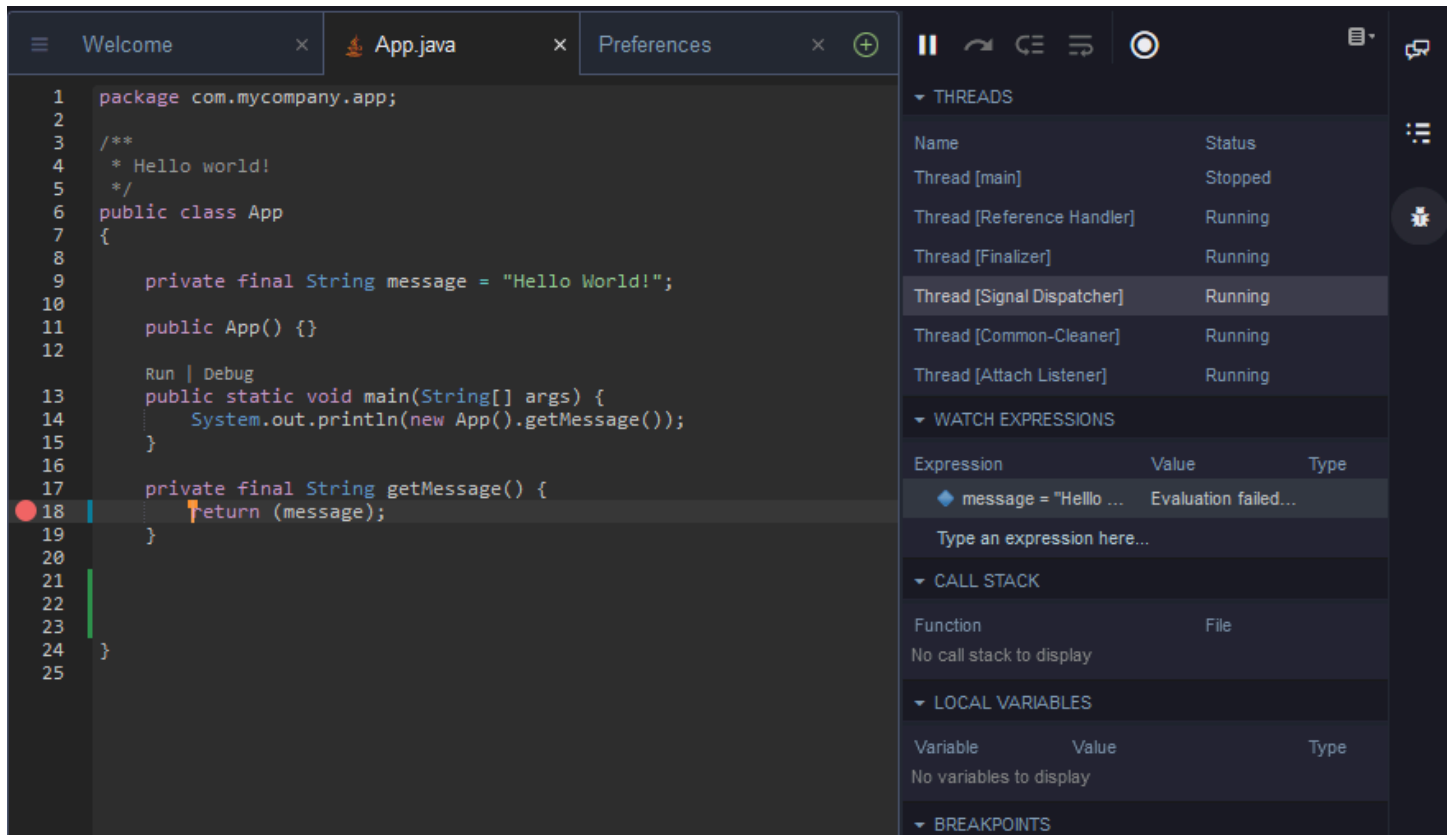


```
App.java
1 package com.mycompany.app;
2
3 /**
4  * Hello world!
5  */
6 public class App
7 {
8
9     private final String message = "Hello World!";
10
11     public App() {}
12
13     Run | Debug
14     public static void main(String[] args) {
15         System.out.println(new App().getMessage());
16     }
17     private final Integer getMessage() {
18         return message;
19     }
20 }
21
22
```

Type mismatch: cannot convert from String to Integer

偵錯選項

您可以實作中斷點和監看表達式。在原始碼中設定中斷點並顯示偵錯器窗格以定義相關條件。

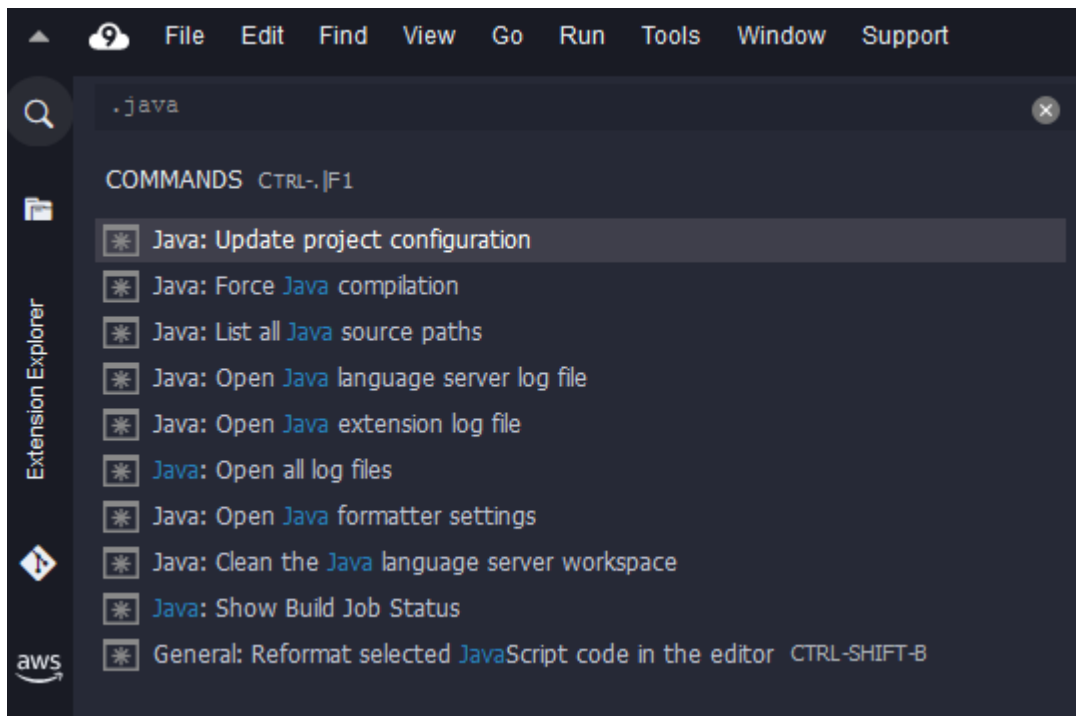


使用組態檔案進行除錯

您還可以使用 AWS Cloud9 透過 `launch.json` 和 `tasks.json` 組態檔案支援的啟動組態和任務來控制的除錯組態。如需啟動組態及其使用方式的範例，請參閱 [Java 除錯組態](#)。

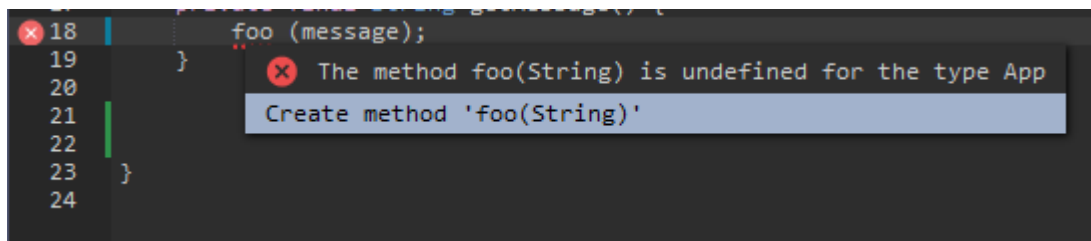
Java 命令

您可以從 AWS Cloud9 命令面板執行命令，方法是按下 `Ctrl+. 或 F1`。然後，透過輸入「java」來篩選相關命令。



快速修復

藉助快速修復，您可以透過為缺失元素建立 stub (虛設常式) 來解決使用未宣告變數或未定義方法導致的錯誤。



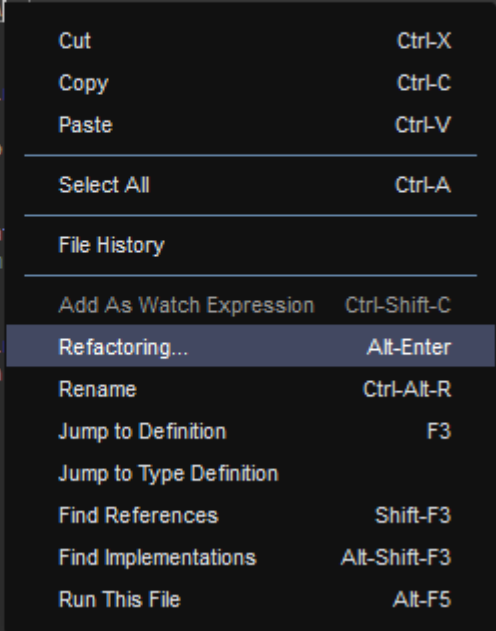
重構

重構可讓您在不變更其行為的情況下重組程式碼的內容。若要存取諸如組織匯入或建立建構函式等選項，請開啟項目的內容 (按一下滑鼠右鍵) 選單，然後選擇 Refactoring (重構)。


```

1  package com.mycompany.app;
2
3  /**
4   * Hello world!
5   */
6  public class App {
7  {
8
9   private final String name = "AWS Cloud9";
10
11  public App() {
12
13   Run | Debug
14   public static void main(String[] args) {
15       System.out.println("Hello World!");
16   }
17
18   private final String name = "AWS Cloud9";
19   return name;
20 }
21
22
23
24 }
25

```



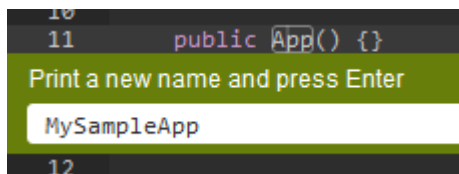
重新命名

重新命名是一種重構功能，可讓您透過單一動作，在程式碼中出現的所有位置輕鬆修改所選變數、函式和類別的名稱。若要變更名稱，請開啟項目的內容 (按一下滑鼠右鍵) 選單，然後選擇 Rename (重新命名)。重新命名會影響程式碼中該名稱的每個實例。

```

10
11  public App() {}

```



```

12

```

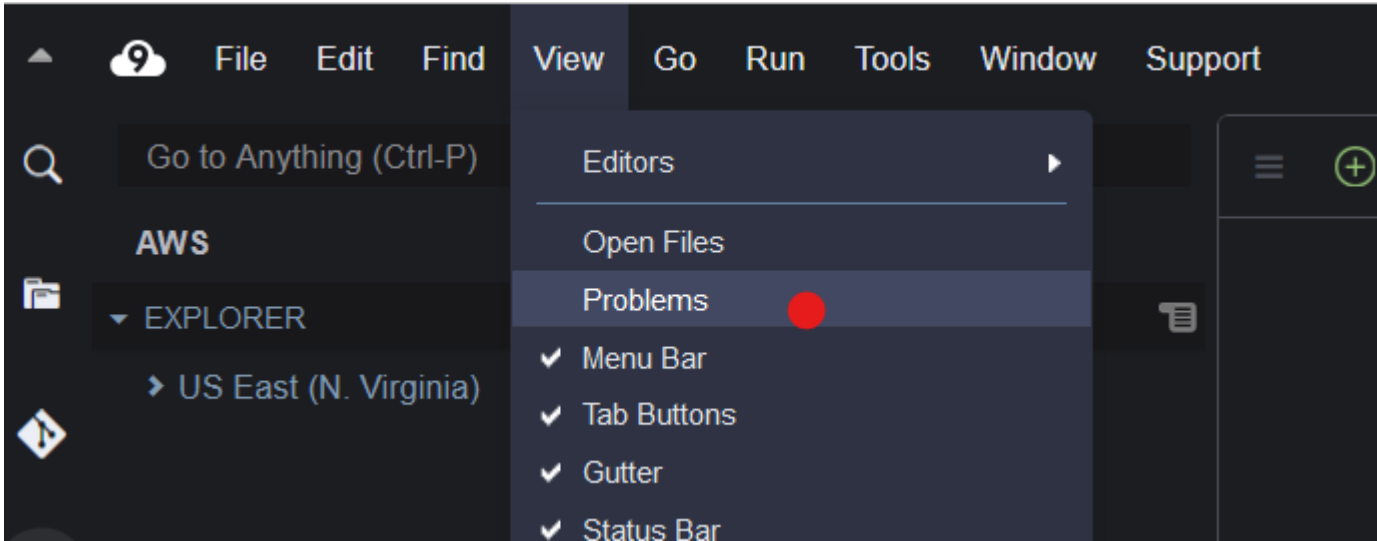
用於 Java 開發的選用工具

提供增強 Java 支援的延伸模組包括可讓您將 Gradle 和 Maven 自動化工具整合至專案開發中的功能。這些工具不會預安裝在 AWS Cloud9 開發環境中。如需安裝和使用這些選用建置工具的詳細資訊，請參閱下列資源：

- Gradle : [Getting started guide](#)
- Maven : [Maven in 5 minutes](#)

Java 擴充功能的 Problems (問題) 索引標籤

您可以在您的 AWS Cloud9 環境中的 AWS Cloud9 IDE Problems (問題) 索引標籤檢閱 java 專案及疑難排解問題。若要顯示 AWS Cloud9 IDE 的 Problems (問題) 索引標籤，請從選單列選取 View (檢視) 後選擇 Problems (問題)。



您也可以選取主控台中的 + 圖示，然後選擇 Open Problems (開啟問題) 來開啟 Problems (問題) 索引標籤。您從索引標籤中選擇問題時，它會打開受影響的檔案並顯示問題詳細資訊。

增強的TypeScript支援與功能

AWS Cloud9IDE 可讓您使用語言專案來存取的增強生產力功能TypeScript。在 AWS Cloud9 開發環境的 IDE 中，語言專案是一組相關檔案、資料夾和設定的集合。

若要使用 IDE 在環境中建立語言專案，請參閱[建立語言專案](#)。

可用的專案生產力功能

AWS Cloud9IDE 提供下列的專案生產力功能TypeScript。

自動完成

隨著您在編輯器輸入於檔案，該內容的插入點處會顯示符號清單，如果有任何符號的話。

若要在插入點插入來自清單的符號，如果尚未選擇符號，請加以選擇，方法是使用向上或向下鍵，然後按下 Tab。

按下 Tab 之前，您可能會看到內含您選擇符號相關資訊的工具提示，如果有提供資訊的話。

若要關閉清單而不插入符號，請按下 Esc。

裝訂邊圖示

圖示可能出現於作用中檔案的裝訂線。這些圖示強調可能的問題，例執行程式碼前出現警告和錯誤訊息。

如需與問題相關的詳細資訊，請將指標暫停在問題圖示上。

快速修復

在編輯器中有效的檔案中，您可以顯示有關編碼錯誤和警告的資訊，以及您可自動套用到該程式碼的可能修復方式。若要顯示錯誤或警告資訊和可能修復的方式，請選擇具有紅色虛線底線 (錯誤) 或灰色虛線底線 (警告) 之程式碼的任何部分。或者，將游標置於具有紅色或灰色虛線底線的程式碼上，按 Option-Enter (適用於 macOS) 或 Alt-Enter (適用於 Linux 或 Windows)。若要將建議修復，選擇清單中的修復，或使用方向鍵來選取修復，然後按 Enter。若要開啟或關閉以滑鼠點選方式選擇快速修復，請選擇 AWS Cloud9、Preferences (偏好設定)、User Settings (使用者設定)、Language (語言)、Hints & Warnings (提示與警告)、Show Available Quick Fixes on Click (按一下時顯示可用的快速修復)。

尋找參考

如果 IDE 可存取插入點符號的所有參考，則您可以在編輯器的作用中檔案顯示這些參考。

若要這樣做，在插入點的任何符號內，執行 **Find References** 命令。例如：

- 在插入點按一下滑鼠右鍵，然後選擇 Find References (尋找參考)。
- 從選單列選擇 Go, Find References (執行、尋找參考)。
- 依預設，針對 macOS、Windows 或 Linux 按下 Shift-F3。

如果有提供參考資料，在該符號旁的作用中檔案頂部會開啟一個窗格。窗格包含符號所參考的檔案的清單。窗格顯示清單中的第一個參考。要顯示不同的參考，請在清單中選擇該參考。

若要關閉窗格中，請在窗格中選擇關閉 (X) 圖示，或按下 Esc。

在以下條件下，**Find References** 命令可能會停用，也可能不會如預期運作：

- 在作用中檔案的專案中，沒有該符號的參考。
- IDE 在作用中檔案的專案中找不到該符號的部分或全部參考。

- IDE 無法在作用中檔案的專案中存取一或多個符號的參考位置。

Go to Definition (移至定義)

如果 IDE 可存取符號定義，則在編輯器的作用中檔案，您可以從某個符號前往該符號定義之處。

若要這樣做，在插入點的任何符號內，執行 **Jump to Definition** 命令。例如：

- 在插入點按一下滑鼠右鍵，然後選擇 Jump to Definition (跳至定義)。
- 在選單列中，選擇 Go, Jump to Definition (執行，跳到定義)。
- 依預設，針對 macOS、Windows 或 Linux 按下 F3。

如果有提供定義，插入點會切換到該定義，即使該定義是在不同的檔案。

在以下條件下，**Jump to Definition** 命令可能會停用，也可能不會如預期運作：

- 該符號是該語言的基本符號。
- IDE 在作用中檔案的專案中找不到定義的位置。
- IDE 在作用中檔案的專案中無法存取定義的位置。

Go to Symbol (移至符號)

您可以在專案中移至特定的符號，如下所示。

1. 在編輯器開啟檔案，使檔案成為專案中其中一個作用中檔案。如果檔案已開啟，請在編輯器中選擇檔案的標籤，使其處於作用中狀態。
2. 執行 **Go to Symbol** 命令。例如：
 - 選擇 Go (執行) 視窗按鈕 (放大鏡圖示)。在 Go to Anything (移至任何內容) 方塊中，輸入 @，然後開始輸入符號。
 - 在選單列上，選擇 Go, Go To Symbol (執行，移至符號)。在 Go (執行) 視窗中，開始開始在 @ 後輸入符號。
 - 依預設針對 macOS 按下 Command-2 或 Command-Shift-0，或依預設按下 Windows 或 Linux 適用的 Ctrl-Shift-0。在 Go (執行) 視窗中，開始開始在 @ 後輸入符號。

舉例來說，若要在名為 toString 的專案中尋找所有符號，請在 @ 後方輸入 @toString (如果 @ 已顯示，則開始在 @ 後輸入 toString)。

3. 如果您看到了您要納入符號清單的符號，請按一下以選擇該符號。或者使用向上或向下鍵加以選取，然後按下 Enter。插入點之後切換到該符號。

如果您想要移往的符號不在作用中檔案的專案中，此程序可能未如預期正常運作。

建立語言專案

請使用下列程序建立語言專案，該專案將使用 AWS Cloud9 IDE 中所支援的專案生產力功能。

Note

我們建議您使用檔案的受支援專案生產力功能，其為語言專案的一部分。雖然您可以使用不屬於專案之檔案上一些支援的檔案專案生產力功能，但這些功能可能會造成意外結果。例如，您可能會使用 IDE 在不屬於專案的環境根層級中搜尋檔案內的參考和定義。然後，IDE 可能會只搜尋在相同根層級的檔案。這可能會導致您找不到參考或定義，即使這些參考或定義實際存在於相同環境的語言專案中。

建立TypeScript語言專案

1. 確保您已經TypeScript安裝在環境中。如需詳細資訊，請參閱 [TypeScript教學課程AWS Cloud9](#) 中的 [步驟 1：安裝必要工具](#)。
2. 在環境的 IDE 中從終端機目錄工作階段切換到您要建立專案的目錄。如果該目錄不存在，請建立它，然後切換到該處。舉例來說，以下命令會在環境的根 (在 ~/environment) 建立名為 my-demo-project 的目錄，然後切換到該目錄。

```
mkdir ~/environment/my-demo-project
cd ~/environment/my-demo-project
```

3. 在您要建立專案的目錄根目錄中，使用 `--init` 選項執行TypeScript編譯器。

```
tsc --init
```

如果此命令成功，TypeScript編譯器會在項目的目錄的根目錄中創建一個 `tsconfig.json` 文件。您可以使用此檔案來定義各種專案設定，例如TypeScript編譯器選項和要包含在專案中或排除的特定檔案。

如需 `tsconfig.json` 檔案的詳細資訊，請參閱以下項目：

- 在網站上[的概述](#)。TypeScript
- json.schemastore.org 網站上的 [tsconfig.json 結構描述](#)。

AWS Cloud9 整合開發環境 (IDE) 的選單列命令參考

下列清單說明 AWS Cloud9 IDE 中的預設選單列命令。如果未顯示選單列，請選擇 IDE 頂端邊緣處的細列以將其顯示。

- [AWS Cloud9 選單](#)
- [File \(檔案\) 選單](#)
- [Edit \(編輯\) 選單](#)
- [Find \(尋找\) 選單](#)
- [View \(檢視\) 選單](#)
- [Go \(執行\) 選單](#)
- [Run \(執行\) 選單](#)
- [工具選單](#)
- [Window \(視窗\) 選單](#)
- [Support \(支援\) 選單](#)
- [Preview \(預覽\) 選單](#)
- [其他選單列命令](#)

AWS Cloud9 選單

命令	描述
Preferences (偏好設定)	執行下列任意一項： <ul style="list-style-type: none">• 如果 Preferences (偏好設定) 標籤未開啟，請將其開啟。• 如果 Preferences (偏好設定) 標籤已開啟但非作用中，請將其設為作用中。• 如果 Preferences (偏好設定) 標籤為作用中，請將其隱藏。

命令	描述
	請參閱「 使用專案設定 」、「 使用使用者設定 」、「 使用按鍵繫結關係 」、「 使用主題 」及「 使用初始化指令碼 」。
Go To Your Dashboard (移至儀表板)	在個別的 Web 瀏覽器標籤中開啟 AWS Cloud9 主控台。請參閱「 建立環境 」、「 開啟環境 」、「 變更環境設定 」和「 刪除環境 」。
Welcome Page (歡迎頁面)	開啟 Welcome (歡迎) 標籤。
Open Your Project Settings (開啟專案設定)	開啟目前環境的 <code>project.settings</code> 檔案。請參閱 使用專案設定 。
Open Your User Settings (開啟使用者設定)	開啟目前使用者的 <code>user.settings</code> 檔案。請參閱 使用使用者設定 。
Open Your Keymap (開啟按鍵對應)	開啟目前使用者的 <code>keybindings.settings</code> 檔案。請參閱 使用按鍵繫結關係 。
Open Your Init Script (開啟初始化指令碼)	開啟目前使用者的 <code>init.js</code> 檔案。請參閱 使用初始化指令碼 。
Open Your Stylesheet (開啟樣式表)	開啟目前使用者的 <code>styles.css</code> 檔案。請參閱 使用主題 。

File (檔案) 選單

命令	描述
New File (新增檔案)	建立新檔案。
New From Template (從範本新增)	根據選擇的檔案範本建立新的檔案。
Open	顯示並移至 Navigate (導覽) 視窗。
Open Recent (開啟最近檔案)	開啟選擇的檔案。

命令	描述
儲存。	儲存目前的檔案。
Save As (另存新檔)	使用不同的檔名、位置或兩者儲存目前的檔案。
Save All (全部儲存)	儲存所有未儲存的檔案。
Revert to Saved (還原為已儲存的版本)	捨棄目前檔案自上次儲存後的變更。
Revert All to Saved (全部還原為已儲存的版本)	捨棄所有未儲存檔案自上次儲存後的變更。
Show File Revision History (顯示檔案修訂歷史記錄)	在編輯器中檢視及管理目前檔案的變更。請參閱 使用檔案修訂 。
Upload Local Files (上傳本機檔案)	顯示 Upload Files (上傳檔案) 對話方塊，這可讓您將本機電腦的檔案拖曳到環境中。
Download Project (下載專案)	將環境中的檔案合併為 .zip 檔，以便下載到本機電腦。
Line Endings (行尾)	使用 Windows (歸位加換行) 或 Unix (僅換行) 的行尾。
Close File (關閉檔案)	關閉目前的檔案。
Close All Files (關閉所有檔案)	關閉所有開啟的檔案。

Edit (編輯) 選單

命令	描述
Undo (復原)	復原上一個動作。
Redo (重做)	重做上一個復原的動作。
Cut (剪下)	將選取項目移動到剪貼簿。
Copy (複製)	將選取項目複製到剪貼簿。

命令	描述
Paste (貼上)	將剪貼簿的內容複製到選取項目點。
Keyboard Mode (鍵盤模式)	要使用的按鍵繫結關係組合，例如 Default、Vim、Emacs 或 Sublime。請參閱 使用按鍵繫結關係 。
Selection, Select All (選取項目、全選)	選取所有可選取的內容。
Selection, Split Into Lines (選取項目、分成數行)	在目前行的行尾新增游標。
Selection, Single Selection (選取項目、單一選取項目)	清除所有之前的選取項目。
Selection, Multiple Selections, Add Cursor Up (選取項目、多個選取項目、在上一行新增游標)	在作用中游標的上一行新增游標。若已新增游標，則在該游標的上一行新增另一個游標。
Selection, Multiple Selections, Add Cursor Down (選取項目、多個選取項目、在下一行新增游標)	在作用中游標的下一行新增游標。若已新增游標，則在該游標的下一行新增另一個游標。
Selection, Multiple Selections, Move Active Cursor Up (選取項目、多個選取項目、將作用中的游標向上移動)	在作用中游標的上一行新增第二個游標。若已新增第二個游標，則將該第二個游標向上移動一行。
Selection, Multiple Selections, Move Active Cursor Down (選取項目、多個選取項目、將作用中的游標向下移動)	在作用中游標的下一行新增第二個游標。若已新增第二個游標，則將該第二個游標向下移動一行。
Selection, Multiple Selections, Add Next Selection Match (選取項目、多個選取項目、新增之後的相符選取項目)	納入選取項目後的更多相符選取項目。
Selection, Multiple Selections, Add Previous Selection Match (選取項目、多個選取項目、新增之前的相符選取項目)	納入選取項目前的更多相符選取項目。

命令	描述
Selection, Multiple Selections, Merge Selection Range (選取項目、多個選取項目、合併選取範圍)	在目前行的行尾新增游標。
Selection, Select Word Right (選取項目、選取右側文字)	將游標右側的下一個文字納入選取項目。
Selection, Select Word Left (選取項目、選取左側文字)	將游標左側的下一個文字納入選取項目。
Selection, Select to Line End (選取項目、選取至行尾)	將游標處至目前行行尾的內容納入選取項目。
Selection, Select to Line Start (選取項目、選取至行首)	將目前行行首到游標處的內容納入選取項目。
Selection, Select to Document End (選取項目、選取至文件結尾)	將游標處以下至目前檔案結尾的內容納入選取項目。
Selection, Select to Document Start (選取項目、選取至文件開頭)	將游標處以上至目前檔案開頭的內容納入選取項目。
Line, Indent (行、縮排)	將選取項目縮排一個定位點。
Line, Outdent (行、凸排)	將選取項目凸排一個定位點。
Line, Move Line Up (行、向上移動一行)	將選取項目向上移動一行。
Line, Move Line Down (行、向下移動一行)	將選取項目向下移動一行。
Line, Copy Lines Up (行、將行複製到上一行)	複製行的內容，並在上一行貼上複製的內容。
Line, Copy Lines Down (行、將行複製到下一行)	複製行的內容，並在下一行貼上複製的內容。
Line, Remove Line (行、移除行)	刪除目前行的內容。
Line, Remove to Line End (行、移除至行尾)	從游標處刪除至目前行的行尾。

命令	描述
Line, Remove to Line Start (行、移除至行首)	從目前行的行首刪除至游標處。
Line, Split Line (行、分行)	將游標的內容移動到行尾，另起一行。
Text, Remove Word Right (文字、移除右側文字)	刪除游標右側的文字。
Text, Remove Word Left (文字、移除左側文字)	刪除游標左側的文字。
Text, Align (文字、對齊)	將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)。
Text, Transpose Letters (文字、調換字母)	調換選取項目。
Text, To Upper Case (文字、轉換為大寫)	將選取項目全部變更為大寫。
Text, To Lower Case (文字、轉換為小寫)	將選取項目全部變更為小寫。
Comment, Toggle Comment (註解、切換註解)	在選取的每一行行首新增行註解字元，若已有註解字元則將其移除。
Code Folding, Toggle Fold (程式碼摺疊、切換摺疊)	摺疊程式碼，若已摺疊則移除程式碼摺疊。
Code Folding, Unfold (程式碼摺疊、展開)	展開選取的程式碼。
Code Folding, Fold Other (程式碼摺疊、摺疊其他項目)	摺疊所有可摺疊的元素，但目前的選取範圍除外。
Code Folding, Fold All (程式碼摺疊、全部摺疊)	摺疊所有可摺疊的元素。
Code Folding, Unfold All (程式碼摺疊、全部展開)	展開整個檔案的程式碼摺疊。
Code Formatting, Apply Code Formatting (程式碼格式化、套用程式碼格式化)	重新格式化選取的 JavaScript 程式碼。

命令	描述
Code Formatting, Open Language & Formatting Preferences (程式碼格式化、開啟語言和格式化偏好設定)	開啟 Preferences (偏好設定) 標籤的 Project Settings (專案設定) 區段，並前往程式設計語言設定。

Find (尋找) 選單

如需詳細資訊，請參閱[尋找和取代文字](#)。

命令	描述
尋找	顯示目前文件的尋找及取代列，並以 Find (尋找) 運算式為焦點。
Find Next (尋找下一個)	移至目前文件中您上次輸入之尋找查詢的下一個符合項目。
Find Previous (尋找上一個)	移至目前文件中您上次輸入之尋找查詢的上一個符合項目。
Replace (取代)	顯示目前文件的尋找及取代列，並以 Replace With (取代為) 運算式為焦點。
Replace Next (取代下一個)	在目前文件的尋找及取代列中，將 Find (尋找) 的下一個符合項目取代為 Replace With (取代為) 的內容。
Replace Previous (取代上一個)	在目前文件的尋找及取代列中，將 Find (尋找) 的上一個符合項目取代為 Replace With (取代為) 的內容。
Replace All (全部取代)	在目前文件的尋找及取代列中，將 Find (尋找) 的所有符合項目取代為 Replace With (取代為) 的內容。
Find in Files (在檔案中尋找)	顯示多個檔案的尋找及取代列。

View (檢視) 選單

命令	描述
Editors (編輯器)	顯示選擇的編輯器。
Open Files (開啟的檔案)	顯示 Environment (環境) 視窗中的 Open Files (開啟的檔案) 清單，若已顯示則隱藏。
問題	在終端的 Problems (問題) 面板中顯示環境的 Java 專案中的任何問題。您可以選擇問題來開啟目標檔案。
Menu Bar (選單列)	顯示選單列，若已顯示則隱藏。
Tab Buttons (標籤按鈕)	顯示標籤，若已顯示則隱藏。
Gutter (裝訂邊)	顯示裝訂邊，若已顯示則隱藏。
Status Bar (狀態列)	顯示狀態列，若已顯示則隱藏。
Console (主控台)	顯示 Console (主控台) 視窗，若已顯示則隱藏。
Layout, Single (配置、單一)	顯示單一窗格。
Layout, Vertical Split (配置、垂直分割)	上下並排顯示兩個窗格。
Layout, Horizontal Split (配置、水平分割)	左右並排顯示兩個窗格。
Layout, Cross Split (配置、十字形分割)	顯示四個相同大小的窗格。
Layout, Split 1:2 (配置、1:2 分割)	在左側顯示一個窗格，並在右側顯示兩個窗格。
Layout, Split 2:1 (配置、2:1 分割)	在左側顯示兩個窗格，並在右側顯示一個窗格。
Font Size, Increase Font Size (字型大小、放大字型)	放大字型。
Font Size, Decrease Font Size (字型大小、縮小字型)	縮小字型。

命令	描述
Syntax (語法)	顯示目前文件的語法類型。
Themes (主題)	顯示 IDE 主題類型。
Wrap Lines (換行)	貼著目前窗格的邊緣將文字換行，若已在換行文字則停止換行。
Wrap To Print Margin (貼著列印邊界)	貼著目前列印邊界的邊緣將文字換行，若已在換行文字則停止換行。

Go (執行) 選單

命令	描述
Go To Anything (移至任何內容)	以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗。
Go To Symbol (移至符號)	以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。
Go To File (移至檔案)	以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。
Go To Command (移至命令)	以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗。
Go To Line (移至命令列)	以 Go to Line (移至命令列) 模式顯示 Go (執行) 視窗。
Next Error (下一個錯誤)	移至下一個錯誤。
Previous Error (上一個錯誤)	移至上一個錯誤。
Word Right (向右移一個文字)	向右移一個文字。
Word Left (向左移一個文字)	向左移一個文字。

命令	描述
Line End (行尾)	移至目前行的行尾。
Line Start (行首)	移至目前行的行首。
Jump to Definition (跳到定義)	移至游標處的變數或函數定義。
Jump to Matching Brace (跳到相配的括號)	移至目前範圍中相配的符號。
Scroll to Selection (捲動至選取項目)	將選取項目捲動至更清楚的檢視位置。

Run (執行) 選單

命令	描述
Run (執行)	執行或除錯目前的應用程式。
Run Last (執行上一個項目)	執行或除錯上次執行的檔案。
Run With (執行工具)	使用選擇的執行器來執行或除錯。請參閱「 使用建置器、執行器和除錯器 」。
Run History (執行歷史記錄)	檢視執行歷史記錄。
Run Configurations (執行組態)	選擇要用來執行或除錯的執行組態，或是建立或管理執行組態。請參閱「 使用建置器、執行器和除錯器 」。
Show Debugger at Break (在中斷點顯示除錯器)	當執行的程式碼到達中斷點時，顯示 Debugger (除錯器) 視窗。
建置	建置目前的檔案。
Cancel Build (取消建置)	停止建置目前的檔案。
Build System (建置系統)	使用選擇的建置系統來建置。
Show Build Result (顯示建置結果)	顯示相關的建置結果。

命令	描述
Automatically Build Supported Files (自動建置支援的檔案)	自動建置支援的檔案。
Save All on Build (在建置時全部儲存)	在建置時儲存所有相關的未儲存檔案。

Tools (工具) 選單

命令	描述
Strip Trailing Space (去除結尾空格)	裁去行尾的空格。
Preview, Preview File (預覽、預覽檔案)	在預覽標籤中預覽目前的文件。
Preview, Preview Running Application (預覽、預覽執行中的應用程式)	在個別的 Web 瀏覽器標籤中，預覽目前的應用程式。
Preview, Configure Preview URL (預覽、設定預覽 URL)	開啟 Preferences (偏好設定) 標籤的 Project Settings (專案設定) 區段，並前往 Run & Debug, Preview URL (執行和除錯、預覽 URL)。
Preview, Show Active Servers (預覽、顯示作用中的伺服器)	在 Process List (程序清單) 對話方塊中顯示可用的作用中伺服器地址清單。
Process List (程序清單)	顯示 Process List (程序清單) 對話方塊。
Show Autocomplete (顯示自動完成)	顯示程式碼完成內容選單。
Rename Variable (重新命名變數)	啟動選取項目的重新命名重構。
Toggle Macro Recording (切換巨集錄製)	開始錄製按鍵輸入，若已在錄製則停止。
Play Macro (播放巨集)	播放先前錄製的按鍵輸入。

Window (視窗) 選單

命令	描述
Go (執行)	顯示 Go (執行) 視窗，若已顯示則隱藏。
New Terminal (新增終端機)	開啟新的 Terminal (終端機) 標籤。
New Immediate Window (新增即時運算視窗)	開啟新的 Immediate (即時運算) 標籤。
Share (分享)	顯示 Share this environment (共享此環境) 對話方塊。
Installer (安裝程式)	顯示 AWS Cloud9 Installer (AWS Cloud9 安裝程式) 對話方塊。
Collaborate (協作)	顯示 Collaborate (協作) 視窗，若已顯示則隱藏。
Outline (大綱)	顯示 Outline (大綱) 視窗，若已顯示則隱藏。
AWS 資源	顯示 AWS Resources (AWS 資源) 視窗，若已顯示則隱藏。
Environment (環境)	顯示 Environment (環境) 視窗，若已顯示則隱藏。
Debugger (除錯器)	顯示 Debugger (除錯器) 視窗，若已顯示則隱藏。
Navigation, Tab to the Right (導覽、向右移一個標籤)	向右移一個標籤。
Navigation, Tab to the Left (導覽、向左移一個標籤)	向左移一個標籤。
Navigation, Next Tab in History (導覽、歷史記錄中的下一個標籤)	移至下一個標籤。

命令	描述
Navigation, Previous Tab in History (導覽、歷史記錄中的上一個標籤)	移至上一個標籤。
Navigation, Move Tab to Right (導覽、將標籤向右移動)	將目前的標籤向右移動。如果標籤已在最右端，則在該處建立分割標籤。
Navigation, Move Tab to Left (導覽、將標籤向左移動)	將目前的標籤向左移動。如果標籤已在最左端，則在該處建立分割標籤。
Navigation, Move Tab to Up (導覽、將標籤向上移動)	將目前的標籤向上移動一個窗格。如果標籤已在最上端，則在該處建立分割標籤。
Navigation, Move Tab to Down (導覽、將標籤向下移動)	將目前的標籤向下移動一個窗格。如果標籤已在最底端，則在該處建立分割標籤。
Navigation, Go to Pane to Right (導覽、向右移一個窗格)	向右移一個窗格。
Navigation, Go to Pane to Left (導覽、向左移一個窗格)	向左移一個窗格。
Navigation, Go to Pane to Up (導覽、向上移一個窗格)	向上移一個窗格。
Navigation, Go to Pane to Down (導覽、向下移一個窗格)	向下移一個窗格。
Navigation, Switch Between Editor and Terminal (導覽、在編輯器及終端機之間切換)	在編輯器及 Terminal (終端機) 標籤之間切換。
Navigation, Next Pane in History (導覽、歷史記錄中的下一個窗格)	移至下一個窗格。
Navigation, Previous Pane in History (導覽、歷史記錄中的上一個窗格)	移至上一個窗格。

命令	描述
Saved Layouts, Save (儲存的配置、儲存)	儲存目前的配置。若稍後要切換到這個配置，請選擇 Saved Layouts, LAYOUT-ID (儲存的配置、LAYOUT-ID)。
Saved Layouts, Save and Close All (儲存的配置、全部儲存並關閉)	儲存目前的配置，然後關閉所有標籤和窗格。
Saved Layouts, Show Saved Layouts in File Tree (儲存的配置、以檔案樹狀目錄顯示儲存的配置)	在 Environment (環境) 視窗中顯示所有儲存的配置。
Tabs, Close Pane (標籤、關閉窗格)	關閉目前的窗格。
Tabs, Close All Tabs In All Panes (標籤、關閉所有窗格中的所有標籤)	關閉所有窗格中所有開啟的標籤。
Tabs, Close All But Current Tab (標籤、關閉目前標籤以外的所有標籤)	關閉目前窗格中所有開啟的標籤，但目前的標籤除外。
Tabs, Split Pane in Two Rows (標籤、將窗格分割為兩列)	將目前的窗格分割為上下兩個窗格。
Tabs, Split Pane in Two Columns (標籤、將窗格分割為兩欄)	將目前的窗格分割為左右兩個窗格。
Presets, Full IDE (預設、完整 IDE)	切換至完整 IDE 模式。
Presets, Minimal Editor (預設、最小編輯器)	切換至最小編輯器模式。
Presets, Sublime Mode (預設、Sublime 模式)	切換至 Sublime 模式。

Support (支援) 選單

命令	描述
Welcome Page (歡迎頁面)	開啟 Welcome (歡迎) 標籤。

命令	描述
Get Help (Community) (取得協助 (社群))	在個別的 Web 瀏覽器標籤中開啟 AWS Cloud9 線上社群網站。
Read Documentation (閱讀文件)	在另一個 Web 瀏覽器標籤中開啟 AWS Cloud9 使用者指南。

Preview (預覽) 選單

命令	描述
Preview File (預覽檔案)	在預覽標籤中預覽目前的文件。
Preview Running Application (預覽執行中的應用程式)	在個別的 Web 瀏覽器標籤中，預覽目前的應用程式。
Configure Preview URL (設定預覽 URL)	開啟 Preferences (偏好設定) 標籤的 Project Settings (專案設定) 區段，並前往 Run & Debug, Preview URL (執行和除錯、預覽 URL)。
Show Active Servers (顯示作用中的伺服器)	在 Process List (程序清單) 對話方塊中顯示可用的作用中伺服器地址清單。

其他選單列命令

命令	描述
Run (執行)	執行或除錯目前的應用程式。
Share (分享)	開啟 Share this environment (共享此環境) 對話方塊。
Preferences (偏好設定) (齒輪圖示)	開啟 Preferences (偏好設定) 標籤。

在 AWS Cloud9 整合開發環境 (IDE) 中尋找及取代文字

您可以在 AWS Cloud9 IDE 使用尋找及取代列，尋找及取代單一檔案或多個檔案中的文字。

- [尋找單一檔案中的文字](#)
- [取代單一檔案中的文字](#)
- [尋找多個檔案中的文字](#)
- [取代多個檔案中的文字](#)
- [尋找及取代選項](#)

尋找單一檔案中的文字

1. 開啟您要在其中尋找文字的檔案。如果該檔案已開啟，請選擇檔案的標籤，使其處於作用中狀態。
2. 從選單列選擇 Find、Find (尋找、尋找)。
3. 在尋找及取代列上，於 Find (尋找) 中輸入您要尋找的文字。
4. 若要指定其他尋找選項，請參閱[尋找及取代選項](#)。
5. 如有任何相符項目，Find (尋找) 方塊中的 0 of 0 會變成非零的數字。如有任何相符項目，編輯器將移至第一個相符項目。若找到多個相符項目後要移至下一個相符項目，請選擇 Find (尋找) 方塊旁的右箭頭，或從選單列選擇 Find、Find Next (尋找、尋找下一個)。如要移至上一個相符項目，請選擇 Find (尋找) 方塊旁的左箭頭，或從選單列選擇 Find、Find Previous (尋找、尋找上一個)。


取代單一檔案中的文字

1. 開啟您要取代其內容文字的檔案。如果該檔案已開啟，請選擇檔案的標籤，使其處於作用中狀態。
2. 從選單列選擇 Find、Replace (尋找、取代)。
3. 在尋找及取代列上，於 Find (尋找) 中輸入您要尋找的文字。
4. 於 Replace With (取代為) 中，輸入您要對 Find (尋找) 中的文字進行取代的文字。
5. 若要指定其他尋找及取代選項，請參閱[尋找及取代選項](#)。
6. 如有任何相符項目，Find (尋找) 方塊中的 0 of 0 會變成非零的數字。如有任何相符項目，編輯器將移至第一個相符項目。若找到多個相符項目後要移至下一個相符項目，請選擇 Find (尋找) 方塊旁的右箭頭，或從選單列選擇 Find、Find Next (尋找、尋找下一個)。如要移至上一個相符項目，請選擇 Find (尋找) 方塊旁的左箭頭，或從選單列選擇 Find、Find Previous (尋找、尋找上一個)。

7. 若要以 Replace With (取代為) 中的文字取代目前相符項目並移至下一個相符項目，請選擇 Replace (取代)。若要以 Replace With (取代為) 中的文字取代所有相符項目，請選擇 Replace All (全部取代)。

尋找多個檔案中的文字

1. 從選單列選擇 Find、Find in Files (尋找、在檔案中尋找)。
2. 在尋找及取代列上，於 Find (尋找) 中輸入您要尋找的文字。
3. 若要指定其他尋找選項，請參閱[尋找及取代選項](#)。
4. 在 Find (尋找) 按鈕右側的方塊 (內有 *.*，-.* 的方塊) 中，輸入您要納入或從尋找目標排除的任何一組檔案。例如：
 - 空白、* 或 *.*：尋找所有檔案。
 - my-file.txt：僅尋找名為 my-file.txt 的檔案。
 - my*：僅尋找檔名以 my 開頭的檔案。
 - my*.txt：僅尋找檔名以 my 開頭且副檔名為 .txt 的檔案。
 - my*.htm*：尋找檔名以 my 開頭且副檔名以 .htm 開頭的所有檔案。
 - my*.htm, my*.html：尋找檔名以 my 開頭且副檔名為 .htm 或 .html 的所有檔案。
 - -my-file.txt：不要搜尋名為 my-file.txt 的檔案。
 - -my*：不要搜尋以 my 開頭的任何檔案。
 - -my*.htm*：不要搜尋檔名以 my 開頭且副檔名以 .htm 開頭的任何檔案。
 - my*.htm*, -my*.html：搜尋檔名以 my 開頭且副檔名以 .htm 開頭的所有檔案，但不要搜尋檔名以 my 開頭且副檔名為 .html 的任何檔案。
5. 從上述方塊旁的下拉式清單中選擇以下任一選項，進一步限制僅在特定的位置尋找：
 - Environment (環境)：僅尋找 Environment (環境) 視窗中的檔案。
 - Project (excludes .gitignore'd) (專案 (排除 .gitignore))：尋找環境中的任何檔案，但環境的 .gitignore 檔案中所列的檔案或檔案類型除外 (如果 .gitignore 檔案存在)。
 - Selection: (選取項目)：僅尋找 Environment (環境) 視窗中目前所選的檔案。

 Note

若要進一步限制僅尋找某個資料夾，請由 Environment (環境) 視窗中選擇該資料夾，然後選擇 Selection (選取項目)。或者，由 Environment (環境) 視窗中對該資料夾按一下滑鼠右鍵，然後從內容選單選擇 Search In This Folder (在此資料夾內搜尋)。

- Favorites (我的最愛)：僅尋找 Environment (環境) 視窗 Favorites (我的最愛) 清單中的檔案。
 - Active File (作用中的檔案)：僅尋找作用中的檔案。
 - Open Files (開啟的檔案)：僅尋找 Environment (環境) 視窗 Open Files (開啟的檔案) 清單中的檔案。
6. 選擇 Find (尋找)。
 7. 若要移至含有相符項目的某個檔案，從 Search Results (搜尋結果) 標籤上按兩下其檔名。若要移至特定的相符項目，從 Search Results (搜尋結果) 標籤上按兩下該相符項目。

取代多個檔案中的文字

1. 從選單列選擇 Find、Find in Files (尋找、在檔案中尋找)。
2. 在尋找及取代列上，於 Find (尋找) 中輸入您要尋找的文字。
3. 若要指定其他尋找選項，請參閱[尋找及取代選項](#)。
4. 在 Find (尋找) 按鈕右側的方塊 (內有 *.* , *.* 的方塊) 中，輸入您要納入或從尋找目標排除的任何一組檔案。例如：
 - 空白、* 或 *.*：所有檔案。
 - my-file.txt：僅限於名為 my-file.txt 的檔案。
 - my*：僅限於檔名以 my 開頭的檔案。
 - my*.txt：僅限於檔名以 my 開頭且副檔名為 .txt 的檔案。
 - my*.htm*：檔名以 my 開頭且副檔名以 .htm 開頭的所有檔案。
 - my*.htm, my*.html：檔名以 my 開頭且副檔名為 .htm 或 .html 的所有檔案。
 - -my-file.txt：不要搜尋名為 my-file.txt 的檔案。
 - -my*：不要搜尋以 my 開頭的任何檔案。
 - -my*.htm*：不要搜尋檔名以 my 開頭且副檔名以 .htm 開頭的任何檔案。
 - my*.htm*, -my*.html：搜尋檔名以 my 開頭且副檔名以 .htm 開頭的所有檔案，但不要搜尋檔名以 my 開頭且副檔名為 .html 的任何檔案。

5. 從上述方塊旁的下拉式清單中選擇以下任一選項，進一步限制僅在特定的位置尋找：
 - Environment (環境)：僅限於 Environment (環境) 視窗中的檔案。
 - Project (excludes .gitignore'd) (專案 (排除 .gitignore))：環境中的任何檔案，但環境的 .gitignore 檔案中所列的檔案或檔案類型除外 (如果 .gitignore 檔案存在)。
 - Selection: / (選取項目)：僅限於目前所選的檔案。
 - Favorites (我的最愛)：僅限於 Environment (環境) 視窗 Favorites (我的最愛) 清單中的檔案。
 - Active File (作用中的檔案)：僅限於作用中的檔案。
 - Open Files (開啟的檔案)：僅限於 Environment (環境) 視窗 Open Files (開啟的檔案) 清單中的檔案。
6. 於 Replace With (取代為) 中，輸入您要對 Find (尋找) 內容進行取代的文字。
7. 選擇 Replace (取代)。

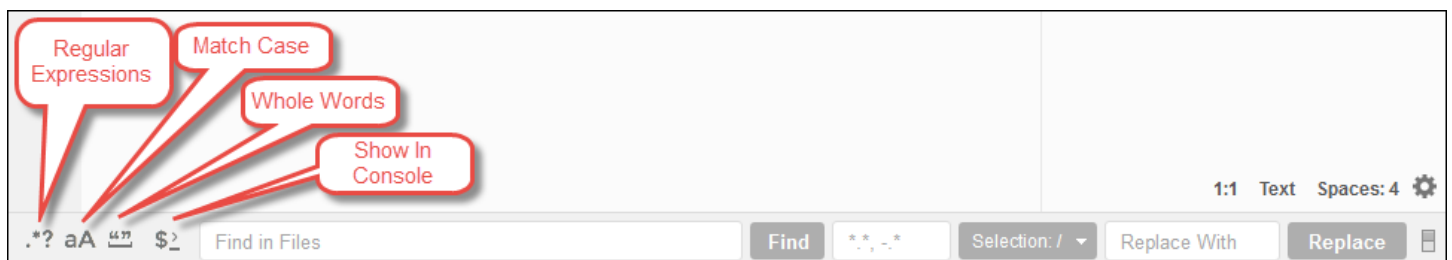
Note

系統將立即對納入範圍內的所有檔案進行取代操作。此操作難以復原。若想要在開始進行取代操作前先了解即將更動哪些內容，請改為選擇 Find (尋找)。

8. 若要移至含有取代項目的某個檔案，從 Search Results (搜尋結果) 標籤上按兩下其檔名。若要移至特定的取代項目，從 Search Results (搜尋結果) 窗格中按兩下該取代項目。

尋找及取代選項

從尋找及取代列選擇以下任何一個按鈕，修改尋找及取代操作方式。



- Regular Expressions (正規運算式)：尋找與 Find (尋找) 或 Find in Files (在檔案中尋找) 所指定的正規運算式相符的文字。請參閱 Mozilla Developer Network 主題「[JavaScript 正規運算式](#)」的撰寫正規表達模式一節。
- Match Case (符合大小寫)：尋找與 Find (尋找) 或 Find in Files (在檔案中尋找) 所指定的字母大小寫相符的文字。
- Whole Words (整個字詞)：使用標準字詞字元規則，尋找 Find (尋找) 或 Find in Files (在檔案中尋找) 所指定的文字。
- Wrap Around (盤繞)：僅限於單一檔案，當移至下一個或上一個相符項目時，若到達檔案末尾或開頭處不要停止。
- Search Selection (搜尋選取項目)：僅限於單一檔案，僅在選取項目中搜尋。
- Show in Console (在主控制台顯示)：適用於多個檔案，在主控制台顯示 Search Results (搜尋結果) 標籤，而不顯示於作用中窗格。
- Preserve Case (維持大小寫)：僅限於單一檔案，在取代文字時維持字母的大小寫 (如適用)。

在 AWS Cloud9 整合開發環境 (IDE) 中預覽檔案

您可以使用 AWS Cloud9 IDE 從 IDE 中預覽 AWS Cloud9 開發環境的檔案。

- [開啟檔案預覽](#)
- [重載載入預覽](#)
- [變更檔案預覽](#)
- [開啟檔案預覽](#)
- [切換至不同的檔案預覽](#)

開啟檔案進行預覽

選擇AWS Cloud9 IDE 中選擇下列其中一項選項目預覽器開啟檔案預覽

- 在環境預覽的檔案預覽的內容選單 (右鍵)，右鍵) 功能表，然後選擇預覽

Note

雖然您可以使用此方式來預覽任何檔案，但是最適合預覽具有下列副檔名的檔案：

- .htm
- .html

- .pdf
- .svg
- .xhtml
- 任何包含 Markdown 格式之內容的檔案。

- 開啟具有下列其中一個副檔名的檔案：
 - .pdf
 - .svg
- 您想要預覽的檔案已開啟並作用中時，請在選單列上，選擇 Preview, Preview File FILE_NAME (預覽、預覽檔案 FILE_NAME)。或者，選擇 Tools, Preview, Preview File FILE_NAME (工具、預覽、預覽檔案 FILE_NAME)，其中 FILE_NAME 是您想要預覽之檔案的名稱。

Note

這些指令僅適用於下列檔案類型：

- .htm
- .html
- .markdown
- .md
- .pdf
- .svg
- .txt：如果檔案內容的格式為 Markdown，則最適合預覽。
- .xhtml：如果檔案包含或參考內容呈現資訊，則最適合預覽。

Note

檔案預覽標籤中的 Preview Settings (預覽設定) 選單目前未運作，而選擇其任何選單命令不會有效果。

重新載入檔案預覽

在檔案預覽標籤上，選擇 Refresh (重新整理) 按鈕 (圓形箭頭)。

變更檔案預覽類型

在檔案預覽標籤上，從預覽類型清單中選擇下列其中一項：

- **Browser (瀏覽器)**：預覽 Web 瀏覽器格式的檔案，僅適用於下列檔案類型：
 - .htm
 - .html
 - .pdf
 - .svg
 - .xhtml：如果檔案包含或參考內容呈現資訊，則最適合預覽。
- **Raw Content (UTF-8) (原始內容 (UTF-8))**：預覽 Unicode 轉換格式 8 位元 (UTF-8) 格式的檔案原始內容。這可能會顯示一些檔案類型的未預期內容。
- **Markdown**：預覽任何包含 Markdown 格式的檔案。嘗試預覽任何其他檔案類型，但可能會顯示未預期的內容。

在個別的 Web 瀏覽器標籤中開啟檔案預覽

在檔案預覽標籤上，選擇 Pop Out Into New Window (彈出到新視窗)。

切換至不同的檔案預覽

在檔案預覽標籤上，於地址列中輸入不同檔案路徑的路徑。網址列的位置在 Refresh (重新整理) 按鈕與預覽類型清單之間。

預覽 AWS Cloud9 整合開發環境 (IDE) 裡執行中的應用程式

您可以使用 AWS Cloud9 IDE 預覽 IDE 內執行中的應用程式。

主題

- [執行應用程式](#)
- [預覽執行中的應用程式](#)
- [重新載入應用程式預覽](#)
- [變更應用程式預覽類型](#)
- [另由其他 Web 瀏覽器標籤開啟應用程式預覽](#)

- [切換到另一個預覽 URL](#)
- [透過網際網路共用執行中的應用程式](#)

執行應用程式

在您可以從 IDE 中預覽應用程式之前，您的應用程式必須在 AWS Cloud9 開發環境中執行。它必須透過下列連接埠使用 HTTP：

- 8080
- 8081
- 8082

上述所有連接埠都必須使用 `127.0.0.1localhost`、或的 IP 位址 `0.0.0.0`。

Note

執行應用程式無需透過連接埠 8080、8081 或 8082 使用 HTTP 並搭配 IP 地址 `127.0.0.1`、`localhost` 或 `0.0.0.0`。然而，如果您不這樣做，將無法從 IDE 內預覽執行中的應用程式。

Note

預覽應用程式會在 IDE 中執行，並載入 `iframe` 元素中。根據預設，某些應用程式伺服器可能會封鎖來自 `iframe` 元素的要求，例如 `X-Frame-Options` 標頭。如果您的預覽應用程式未顯示在預覽標籤中，請確定您的應用程式伺服器未禁止在 `iframe` 中顯示內容。

若要編寫程式碼於特定的連接埠和 IP 地址執行應用程式，請參閱應用程式的說明文件。

若要執行應用程式，請參閱[執行您的程式碼](#)。

若要測試此行為，請將下列 JavaScript 程式碼新增至在您環境根目錄 `server.js` 中命名的檔案中。此程式碼使用名為 `Node.js` 的檔案來執行伺服器。

Note

在下列範例中，text/html 是傳回內容的 Content-Type。若要以不同格式傳回內容，請指定不同的 Content-Type。例如，您可以為 CSS 黨案格式指定 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'.
```

在環境的根目錄內，您可將以下 Python 程式碼加入名為 server.py 的檔案。在下面的範例中，伺服器是使用 Python 執行。

```
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()
```

在環境的根目錄內，將以下 HTML 程式碼加入名為 `index.html` 的檔案。

```
<html>
  <head>
    <title>Hello Home Page</title>
  </head>
  <body>
    <p style="font-family:Arial;color:blue">Hello, World!</p>
  </body>
</html>
```

若要由應用程式預覽標籤上查看此檔案的 HTML 輸出，可使用 Node.js 執行 `server.js` 或使用 Python 執行 `server.py` 檔案。接著，請執行下一節中的步驟來預覽。在應用程式預覽標籤上，於 URL 末尾處加入 `/index.html`，然後按 Enter。

預覽執行中的應用程式

預覽您的應用程式之前，請先確認以下項目：

- 您的應用程式透過連接埠 8080、8081 或 8082 使用 HTTP 通訊協定來執行。
- 您的應用程式在環境中的 IP 地址是 127.0.0.1、localhost 或 0.0.0.0。
- 您的應用程式程式碼檔案已在 AWS Cloud9 IDE 中開啟且處於作用中狀態。

確認所有這些詳細資料後，請從選單列選擇下列其中一個選項：

- Preview、Preview Running Application (預覽、預覽執行中的應用程式)
- Tools、Preview、Preview Running Application (工具、預覽、預覽執行中的應用程式)

這些選項中的任何一個會在環境中開啟應用程式預覽標籤，然後於該標籤上顯示應用程式的輸出。

Note

如果應用程式預覽標籤顯示錯誤或空白，請在 [應用程式預覽標籤顯示錯誤或一片空白](#) 中嘗試下列故障診斷步驟。如果您嘗試預覽應用程式或檔案時，您會收到下列通知「預覽功能已停用，因為您的瀏覽器已停用第三方 Cookie」，請依照中的疑難排解步驟進行 [應用程式預覽或檔案預覽通知：「Third-party cookies disabled」\(已停用第三方 Cookie\)](#)。

Note

如果應用程式並未執行，應用程式預覽標籤會顯示錯誤。若要解決此問題，請再次執行或重新啟動應用程式，然後從選單列選擇命令。

例如，假設您的應用程式無法在提及的任何連接埠或 IP 上執行。或者，您的應用程式必須同時在一個以上的連接埠上執行。例如，您的應用程式必須同時在連接埠 8080 和 3000 上執行。如果是這種情況，則應用程式預覽標籤可能會顯示錯誤或可能空白。這是因為環境中的應用程式預覽標籤只能搭配上上述連接埠和 IP 運作。此外，應用程式一次只能使用一個連接埠。

建議您不要向其他人透露應用程式預覽標籤上的 URL (URL 的格式如

下：`https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/`。在這種格式中，12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 指派給環境的 ID。us-east-2 是 AWS 區域 指派給環境的 ID。) 此 URL 僅限於環境的 IDE 已開啟且應用程式由同一個 Web 瀏覽器執行時才有效。

如果您嘗試使用 IDE 中的應用程式預覽標籤或 IDE 外部的不同 Web 瀏覽器標籤來造訪 IP 127.0.0.1、localhost 或 0.0.0.0，AWS Cloud9 IDE 預設會嘗試移至您的本機電腦，而不會嘗試移至連接到環境的執行個體或您自己的伺服器。

如需如何在 IDE 之外向其他人提供執行中應用程式預覽的指示，請參閱[透過網際網路共用執行中的應用程式](#)。

重新載入應用程式預覽

在應用程式預覽標籤上，選擇 Refresh (重新整理) 按鈕 (圓形箭頭)。

Note

此命令不會重新啟動伺服器。而只是重新整理應用程式預覽標籤的內容。

變更應用程式預覽類型

在應用程式預覽索引標籤上，從預覽類型清單中選擇下列其中一項：

- Browser (瀏覽器)：預覽 Web 瀏覽器格式的輸出。
- Raw Content (UTF-8) (原始內容 (UTF-8))：嘗試預覽 Unicode 轉換格式 8 位元 (UTF-8) 格式的輸出 (如適用)。
- Markdown：嘗試預覽 Markdown 格式的輸出 (如適用)。

另由其他 Web 瀏覽器標籤開啟應用程式預覽

在應用程式預覽標籤上，選擇 Pop Out Into New Window (彈出到新視窗)。

Note

AWS Cloud9IDE 也必須在同一個 Web 瀏覽器中至少一個其他索引標籤中執行。否則，應用程式預覽不會顯示在單獨的 Web 瀏覽器標籤中。

AWS Cloud9 IDE 也必須在同一個 Web 瀏覽器的至少一個其他標籤中執行。否則，應用程式預覽不會顯示在單獨的 Web 瀏覽器標籤中。如果應用程式預覽標籤顯示錯誤或空白，請在 [應用程式預覽或檔案預覽通知：「Third-party cookies disabled」\(已停用第三方 Cookie\)](#) 中嘗試下列故障診斷步驟。

切換到另一個預覽 URL

在應用程式預覽標籤上，於網址列輸入另一個 URL 的路徑。網址列的位置在 Refresh (重新整理) 按鈕與預覽類型清單之間。

透過網際網路共用執行中的應用程式

預覽執行中的應用程式之後，您可以透過網際網路供予其他人存取該應用程式。

若已有 Amazon EC2 執行個體連接到您的環境，請執行以下步驟。否則，請參閱您的伺服器所附說明文件。

主題

- [步驟 1：取得執行個體的 ID 及 IP 地址](#)
- [步驟 2：為執行個體設定安全群組](#)
- [步驟 3：為執行個體設定子網路](#)
- [步驟 4：共用執行中應用程式的 URL](#)

步驟 1：取得執行個體的 ID 及 IP 地址

在此步驟中，請針對已連接到環境的 Amazon EC2 執行個體記下執行個體 ID 及公有 IP 地址。後續步驟將需要憑此執行個體 ID 以允許傳入應用程式請求。接著，將公有 IP 地址給予其他人，使對方能夠存取執行中的應用程式。

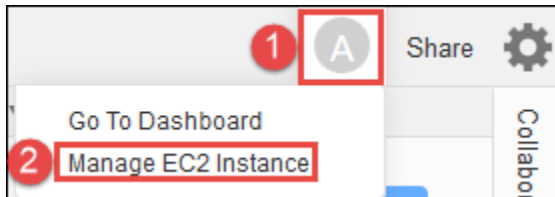
1. 取得 Amazon EC2 執行個體的 ID。為此，請執行以下其中一項操作：

- 於環境的 AWS Cloud9 IDE 終端機工作階段執行以下命令，取得 Amazon EC2 執行個體的 ID。

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

執行個體 ID 的格式如下：i-12a3b456c789d0123。記下該執行個體 ID。

- 在環境的 IDE 中，從選單列選擇您的使用者圖示，然後選擇 Manage EC2 Instance (管理 EC2 執行個體)。



Amazon EC2 主控台出現後，記下 Instance ID (執行個體 ID) 欄內所顯示的執行個體 ID。執行個體 ID 的格式如下：i-12a3b456c789d0123。

2. 取得 Amazon EC2 執行個體的公有 IP 地址。為此，請執行以下其中一項操作：

- 在環境的 IDE 中，從選單列選擇 Share (共用)。在 Share this environment (共享此環境) 對話方塊中，記下 Application (應用程式) 方塊所顯示的公有 IP 地址。公有 IP 地址的格式如下：192.0.2.0。
- 於環境的 IDE 終端機工作階段執行以下命令，取得 Amazon EC2 執行個體的公有 IP 地址。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

公有 IP 地址的格式如下：192.0.2.0。記下該公有 IP 地址。

- 在環境的 IDE 中，從選單列選擇您的使用者圖示，然後選擇 Manage EC2 Instance (管理 EC2 執行個體)。Amazon EC2 主控台出現後，記下 Description (說明) 標籤上 IPv4 Public IP (IPv4 公有 IP) 欄位內的公有 IP 地址。公有 IP 地址的格式如下：192.0.2.0。

Note

應用程式的公有 IP 地址可能會在應用程式的執行個體重新啟動時變更。若要防止 IP 地址變更，請配置彈性 IP 地址。然後，將該地址指派給執行中的執行個體。如需相關說明，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的[配置彈性 IP 地址](#)及[將彈性 IP](#)

[地址與運作中的執行個體建立關聯](#)。配置彈性 IP 地址可能會導致您的 AWS 帳戶產生費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

步驟 2：為執行個體設定安全群組

在此步驟中，您將使用 Amazon EC2 主控台為連接到環境的執行個體設定 Amazon EC2 安全群組。將其設定為允許透過連接埠 8080、8081 或 8082 傳入的 HTTP 要求。

Note

您不需要透過連接埠 8080、8081 或 8082 使用 HTTP 來執行。如果您不這樣做，將無法從 IDE 內預覽執行中的應用程式。如需詳細資訊，請參閱[預覽執行中的應用程式](#)。否則，如果您是使用其他通訊協定或連接埠執行，請在此步驟中替換它。

如要獲得多一層的安全保障，請為 VPC 中可供執行個體使用的子網路設定一份網路存取控制清單 (ACL)。如需安全群組與網路 ACL 的詳細資訊，請參閱以下內容：

- [步驟 3：為執行個體設定子網路](#)
- Amazon VPC 使用者指南中的[安全性](#)
- Amazon VPC 使用者指南中的[VPC 的安全群組](#)
- Amazon VPC 使用者指南中的[網路 ACL](#)

1. 在環境的 IDE 中，從選單列選擇您的使用者圖示，然後選擇 Manage EC2 Instance (管理 EC2 執行個體)。接著跳至本程序的步驟 3。
2. 如果選擇 Manage EC2 Instance (管理 EC2 執行個體) 或進行本程序的其他步驟時傳回錯誤，請使用 AWS 帳戶中管理員的憑證登入 Amazon EC2 主控台。然後完成下列指示。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。
 - a. 若您尚未登入 AWS Management Console，請前往 <https://console.aws.amazon.com/> 登入。
 - b. 開啟 Amazon EC2 主控台。方法如下：從導覽列選擇 Services (服務)。然後選擇 EC2。
 - c. 從導覽列選擇環境所在的 AWS 區域。
 - d. 出現 EC2 Dashboard (EC2 儀表板) 時，選擇 Running Instances (運作中的執行個體)。否則，若服務導覽窗格中仍未展開 Instances (執行個體)，請將其展開後再選擇 Instances (執行個體)。

- e. 從執行個體清單中，選取 Instance ID (執行個體 ID) 與您稍早記下的執行個體 ID 相符的執行個體。
3. 在執行個體的 Description (說明) 標籤上，選擇 Security groups (安全群組) 旁的安全群組連結。
4. 安全群組顯示後，查看 Inbound (傳入) 標籤。若當中已有規則，其 Type (類型) 設為 Custom TCP Rule (自訂 TCP 規則) 且 Port Range (連接埠範圍) 設為 8080、8081 或 8082，請選擇 Cancel (取消)，然後跳至 [步驟 3：為執行個體設定子網路](#)。否則選擇 Edit (編輯)。
5. 在 Edit inbound rules (編輯傳入規則) 對話方塊中，選擇 Add Rule (新增規則)。
6. 針對 Type (類型)，選擇 Custom TCP Rule (自訂 TCP 規則)。
7. 針對 Port Range (連接埠範圍)，輸入 8080、8081 或 8082。
8. 針對 Source (來源)，選擇 Anywhere (隨處)。

Note

Source (來源) 選擇 Anywhere (隨處) 時，將允許來自任何 IP 地址的傳入請求。若要將此限制為特定 IP 地址，請選擇 Custom (自訂)，然後輸入 IP 地址範圍。或者，選擇 My IP (我的 IP)，將請求限制為僅來自您的 IP 地址。

9. 選擇 儲存。

步驟 3：為執行個體設定子網路

使用 Amazon EC2 和 Amazon VPC 主控台，為連接環境的 Amazon EC2 執行個體設定子網路。接著，允許透過連接埠 8080、8081 或 8082 傳入的 HTTP 要求。

Note

您不需要透過連接埠 8080、8081 或 8082 使用 HTTP 來執行。然而，如果您不這樣做，將無法從 IDE 內預覽執行中的應用程式。如需詳細資訊，請參閱[預覽執行中的應用程式](#)。否則，如果您是使用其他通訊協定或連接埠執行，請在此步驟中替換它。

此步驟說明如何為 Amazon VPC 中可供執行個體使用的子網路設定網路 ACL。此步驟非必要，但仍建議執行。設定網路 ACL 可增加一層額外的安全性。如需網路 ACL 的詳細資訊，請參閱以下內容：

- Amazon VPC 使用者指南中的[安全性](#)
- Amazon VPC 使用者指南中的[網路 ACL](#)

1. 在 Amazon EC2 主控台上，若服務導覽窗格中仍未展開 Instances (執行個體)，請將其展開後再選擇 Instances (執行個體)。
2. 從執行個體清單中，選取 Instance ID (執行個體 ID) 與您稍早記下的執行個體 ID 相符的執行個體。
3. 在執行個體的 Description (說明) 標籤上，記下 Subnet ID (子網路 ID) 的值。子網路 ID 的格式如下：subnet-1fab8aEX。
4. 開啟 Amazon VPC 主控台。若要執行此作業，請在 AWS 導覽列上，選擇 Services (服務)，然後選擇 VPC。

針對此步驟，建議您使用 AWS 帳戶中管理員的憑證登入 Amazon VPC 主控台。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。

5. 出現 VPC Dashboard (VPC 儀表板) 時，選擇 Subnets (子網路)。否則，從服務導覽窗格中選擇 Subnets (子網路)。
6. 從子網路清單中，選取 Subnet ID (子網路 ID) 與您稍早記下的值相符的子網路。
7. 在 Summary (摘要) 標籤上，選擇 Network ACL (網路 ACL) 旁的網路 ACL 連結。
8. 從網路 ACL 清單中選取網路 ACL (只會有一份網路 ACL)。
9. 查看網路 ACL 的 Inbound Rules (傳入規則) 標籤。若當中有一項規則，其 Type (類型) 設為 HTTP* (8080)、HTTP* (8081) 或 HTTP* (8082)，請直接跳到 [步驟 4：共用執行中應用程式的 URL](#)。否則選擇 Edit (編輯)。
10. 選擇 Add another rule (新增其他規則)。
11. 在 Rule # (規則 #) 中輸入規則的編號 (例如 200)。
12. 針對 Type (類型)，選擇 Custom TCP Rule (自訂 TCP 規則)。
13. 在 Port Range (連接埠範圍) 中輸入 8080、8081 或 8082。
14. 對於 Source (來源)，輸入要允許傳入請求的 IP 地址範圍。例如，輸入 0.0.0.0/0 將允許來自任何 IP 地址的傳入請求。
15. 在 Allow / Deny (允許/拒絕) 設為 ALLOW (允許) 的情況下，選擇 Save (儲存)。

步驟 4：共用執行中應用程式的 URL

應用程式執行後，您可以透過提供應用程式的 URL，來與其他人共用您的應用程式。為此，您需要先記下的公有 IP 地址。若要撰寫應用程式的完整 URL，請務必使用正確的通訊協定來啟動應用程式的公用 IP 地址。接著，如果您的應用程式連接埠不是所使用通訊協定的預設連接埠，請新增連接埠號碼資訊。下列是應用程式 URL 範例：`http://192.0.2.0:8080/index.html` 透過連接埠 8080 使用 HTTP。

如果產生的 web 瀏覽器標籤顯示錯誤或空白，請按照 [無法從 IDE 外部顯示執行中的應用程式](#) 中的故障診斷步驟操作。

Note

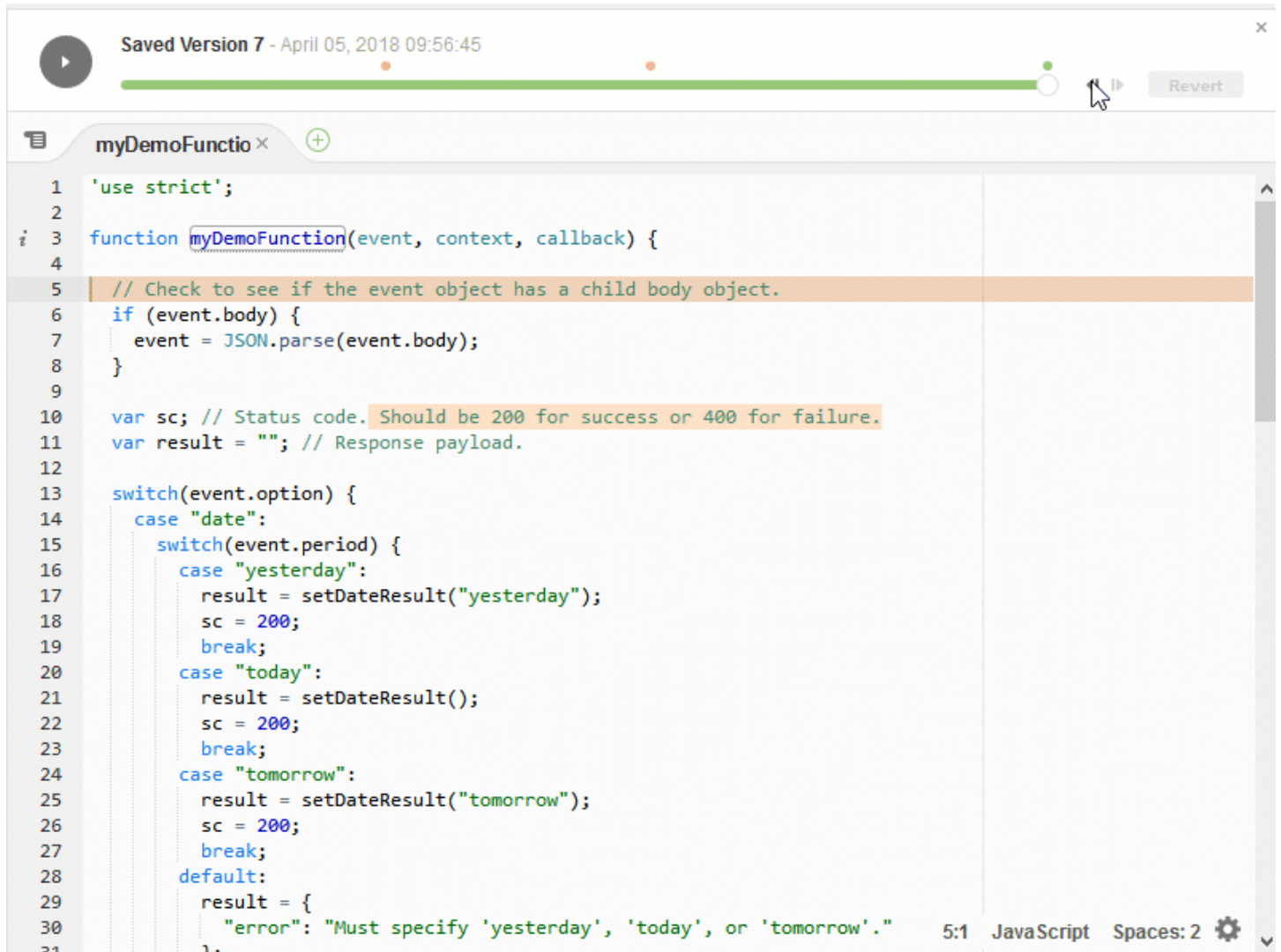
應用程式的公有 IP 地址可能會在應用程式的執行個體重新啟動時變更。為避免您的 IP 地址發生變更，請配置彈性 IP 地址，然後將該地址指派給運作中的執行個體。如需相關說明，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的 [配置彈性 IP 地址](#) 及 [將彈性 IP 地址與運作中的執行個體建立關聯](#)。配置彈性 IP 地址可能會導致您的 AWS 帳戶產生費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

您不需要透過連接埠 8080、8081 或 8082 使用 HTTP 來執行應用程式。然而，如果您不這樣做，將無法從 IDE 內預覽執行中的應用程式。如需詳細資訊，請參閱 [預覽執行中的應用程式](#)。例如，假設來自 VPN 的要求會封鎖所要求通訊協定或連接埠上的流量。然後，那些存取應用程式 URL 的請求可能會失敗。必須換成允許透過請求的通訊協定及連接埠傳送流量的其他網路。如需詳細資訊，請洽詢您的網路管理員。

建議您不要向其他人透露 IDE 應用程式預覽標籤上的 URL。(此 URL 的格式如下：`https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/`。在這種格式中，12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 指派給環境的 ID。us-east-2 是 AWS 區域 指派給環境的 ID。) 此 URL 僅限於環境的 IDE 已開啟且應用程式由同一個 Web 瀏覽器執行時才有效。

在 AWS Cloud9 整合開發環境 (IDE) 中使用檔案修訂

您可以使用 AWS Cloud9 IDE 中的 File Revision History (檔案修訂歷史記錄) 窗格，檢視和管理 AWS Cloud9 EC2 環境中檔案的變更。File Revision History (檔案修訂歷史記錄) 窗格不適用於 AWS Cloud9 SSH 開發環境中的檔案。



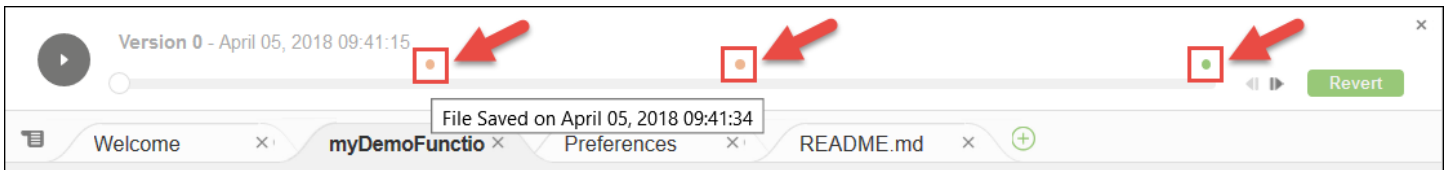
```
1 'use strict';
2
3 function myDemoFunction(event, context, callback) {
4
5 // Check to see if the event object has a child body object.
6 if (event.body) {
7     event = JSON.parse(event.body);
8 }
9
10 var sc; // Status code. Should be 200 for success or 400 for failure.
11 var result = ""; // 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                 }
```

若要顯示檔案的 File Revision History (檔案修訂歷史記錄) 窗格，請在編輯器中開啟檔案。然後，在選單列上，選擇 File, Show File Revision History (檔案、顯示檔案修訂歷史記錄)。

File Revision History (檔案修訂歷史記錄) 窗格會在您第一次於環境中的編輯器開啟檔案之後，開始追蹤 IDE 中檔案的修訂歷史記錄，但僅針對該環境。File Revision History (檔案修訂歷史記錄) 窗格只會追蹤來自編輯器本身的檔案修訂。它不會追蹤以任何其他方式進行的檔案修訂 (例如，透過終端機、Git 或其他檔案修訂工具)。

顯示 File Revision History (檔案修訂歷史記錄) 窗格時，即無法編輯檔案。若要隱藏窗格，請再次選擇 File, Show Revision History (檔案、顯示修訂歷史記錄)，或選擇該窗格角落的 X (Close timeslider (關閉時間滑桿))。

若要跳到與檔案儲存動作相關聯的檔案版本，請選擇修訂滑桿上方的 File Saved on (檔案儲存於) 點。



若要將修訂滑桿上目前選取的檔案版本前進或後退一個版本，請選擇其中一個步驟箭頭 (Step revision forward (前進修訂) 或 Step revision backward (後退修訂))。



若要從修訂歷史記錄的開始到結束一次自動前進檔案的一個版本，請選擇播放按鈕 (Playback file history (播放檔案歷史記錄))。

若要讓檔案的目前選取版本成為修訂歷史記錄中的最新版本，請選擇 Revert (還原)。

在 AWS Cloud9 整合開發環境 (IDE) 中使用影像檔案

您可以使用 AWS Cloud9 IDE 檢視及編輯影像檔案。

- [檢視或編輯影像](#)
- [調整影像大小](#)
- [裁剪影像](#)
- [旋轉影像](#)
- [翻轉影像](#)
- [縮放影像](#)
- [平滑影像](#)

檢視或編輯影像

在 AWS Cloud9 IDE 中，開啟您要檢視或編輯的影像檔案。支援的影像檔案類型包括：

- .bmp
- .gif (僅限檢視)
- .ico (僅限檢視)
- .jpeg

- .jpg
- .png
- .tiff

調整影像大小

1. 在 IDE 中開啟影像檔案。
2. 在影像編輯列上，選擇 Resize (調整大小)。
3. 若要變更影像寬度，請以像素為單位輸入新的 Width (寬度)。或者，選擇 Width (寬度) 旁的 "-" 或 "+", 一次調整一個像素來變更目前寬度。
4. 若要變更影像高度，請以像素為單位輸入新的 Height (高度)。或者，選擇 Height (高度) 旁的 "-" 或 "+", 一次調整一個像素來變更目前高度。
5. 若要維持影像的寬度和高度比，請保留核取 Maintain Aspect Ratio (維持外觀比例)。
6. 若要確認影像的新的尺寸，請在影像編輯列上，查看寬度 (W) 和高度 (H) 度量 (以像素為單位)。
7. 選擇 Resize (調整大小)。
8. 若要捨棄調整大小，請在選單列上，選擇 Edit (編輯)、Undo (復原)。若要保留新的尺寸，請選擇 File (檔案)、Save (儲存)。

裁剪影像

1. 在 IDE 中開啟影像檔案。
2. 將指標拖曳到您要保留的影像部分上方。
3. 若要確認選取範圍的維度，請在影像編輯列上，查看 Selection (選取範圍) 維度，如下所示：
 - 從原始影像左邊緣到選取範圍左邊緣的距離 (L) (以像素為單位)
 - 從原始影像上邊緣到選取範圍上邊緣的距離 (T) (以像素為單位)
 - 選取範圍的寬度 (W) (以像素為單位)
 - 選取範圍的高度 (H) (以像素為單位)
4. 在影像編輯列上，選擇 Crop (裁剪)。
5. 若要捨棄裁剪，請在選單列上，選擇 Edit (編輯)、Undo (復原)。若要保留新的裁剪影像，請選擇 File (檔案)、Save (儲存)。

旋轉影像

1. 在 IDE 中開啟影像檔案。
2. 若要逆時針旋轉影像，請在影像編輯列上，選擇 Rotate 90 Degrees Left (向左旋轉 90 度)。
3. 若要順時針旋轉影像，請在影像編輯列上，選擇 Rotate 90 Degrees Right (向右旋轉 90 度)。
4. 若要捨棄旋轉，請在選單列上，選擇 Edit (編輯)、Undo (復原)。若要保留新的旋轉影像，請選擇 File (檔案)、Save (儲存)。

翻轉影像

1. 在 IDE 中開啟影像檔案。
2. 若要水平翻轉影像，請在影像編輯列上，選擇 FlipH (水平翻轉)。
3. 若要垂直翻轉影像，請在影像編輯列上，選擇 FlipV (垂直翻轉)。
4. 若要捨棄翻轉，請在選單列上，選擇 Edit (編輯)、Undo (復原)。若要保留新的翻轉影像，請選擇 File (檔案)、Save (儲存)。

縮放影像

1. 在 IDE 中開啟影像檔案。
2. 在影像編輯列上，選擇其中一個可用的縮放係數 (例如 75%、100% 或 200%)。

平滑影像

1. 在 IDE 中開啟影像檔案。
2. 在影像編輯列上，選取 Smooth (平滑) 以降低影像中的像素化數量。若要捨棄平滑，請取消選取 Smooth (平滑)。
3. 在選單列上，選擇 File (檔案)、Save (儲存)。

在 AWS Cloud9 整合開發環境 (IDE) 中使用建置器、執行器和除錯器

建置器會指示 AWS Cloud9 IDE 如何建置專案檔案。執行器會指示 AWS Cloud9 IDE 如何執行特定的檔案類型。執行器可以使用「除錯器」來協助尋找檔案中來源碼的任何問題。

您可以採取下列方式使用 AWS Cloud9 IDE 來建置、執行和除錯您的程式碼：

- 使用建置器來建置專案檔案。請參閱「[建置專案檔案](#)」。
- 使用執行器來執行 (和選擇性除錯) 您的程式碼。請參閱「[內建組建、執行和除錯支援](#)」和「[執行您的程式碼](#)」。
- 以不同於原始定義的方式變更內建執行器來執行 (和選擇性除錯) 您的程式碼。請參閱「[變更內建執行器](#)」。
- 以自訂檔案名稱組合、命令列選項、除錯模式、目前工作目錄和環境變數使用執行器來執行 (和選擇性除錯) 您的程式碼。請參閱「[建立執行組態](#)」。
- 建立您自己的建置器或執行器。請參閱「[建立建置器或執行器](#)」。

內建組建、執行和除錯支援

AWS Cloud9 IDE 為程式碼的建置、執行和除錯提供多種語言的內建支援。如需完整清單，請參閱「[語言支援](#)」。

選單列以 Run (執行)、Build System (建置系統) 和 Run (執行)、Build (建置) 選單命令提供內建組建支援。若要為未列出的程式設計語言或工具新增支援，請參閱「[建立建置器或執行器](#)」。

內建執行支援包括 Run (執行) 按鈕，以及選單列的 Run (執行)、Run With (執行工具) 和 Run (執行)、Run Configurations (執行組態) 選單命令。若要為未列出的程式設計語言或工具新增支援，請參閱「[建立建置器或執行器](#)」和「[建立執行組態](#)」。

內建除錯支援可透過 Debugger (除錯器) 視窗獲得。若要顯示 Debugger (除錯器) 視窗，請選擇 Debugger (除錯器) 按鈕。如果 Debugger (除錯器) 按鈕不可見，請在選單列上選擇 Window (視窗)、Debugger (除錯器)。

建置專案檔案

1. 開啟對應至您希望建置之程式碼的檔案。
2. 在選單列上，選擇 Run, Build System (執行、建置系統)，然後選擇要使用的建置器名稱 (若尚未選擇)。如果您希望使用的建置器未列出，請停止此程序，先完成「[建立建置器或執行器](#)」中的步驟，再返回此程序。
3. 選擇 Run, Build (執行、建置)。

執行您的程式碼

1. 開啟對應至您希望執行的程式碼檔案 (若尚未開啟及選取該檔案)。
2. 在選單列上，選擇下列其中一項動作：
 - 若要以最接近的相符內建執行器來執行程式碼，請選擇 Run, Run (執行、執行)。如果 AWS Cloud9 找不到，則此命令已停用。
 - 若要以 AWS Cloud9 上一次使用的執行組態來執行程式碼，請選擇 Run, Run Last (執行、執行上一個項目)。
 - 若要以特定執行器來執行程式碼，請選擇 Run, Run With (執行、執行工具)，然後選擇執行器的名稱。如果您希望使用的執行器未列出，請停止此程序，先完成「[建立建置器或執行器](#)」中的步驟，再返回此程序。
 - 若要以特定執行器與自訂檔案名稱組合、命令列選項、除錯模式，目前工作目錄和環境變數來執行程式碼，請選擇 Run, Run Configurations (執行、執行組態)，然後選擇執行組態的名稱。在顯示的執行組態標籤中，選擇 Runner : Auto (執行器：自動)，選擇您希望使用的執行器，然後選擇 Run (執行)。如果您希望使用的執行器未列出，請停止此程序，先完成「[建立建置器或執行器](#)」中的步驟，再返回此程序。

除錯您的程式碼

1. 在程式碼的執行組態標籤上選擇 Run in Debug Mode (以除錯模式執行)。錯誤圖示會在白色背景中轉換為綠色。如需詳細資訊，請參閱「[執行您的程式碼](#)」和「[建立執行組態](#)」。
2. 在您的程式碼中設定執行期間您想要暫停的中斷點，如下所示：
 - a. 開啟每個您想要設定中斷點的檔案。
 - b. 在您想要設定中斷點之檔案的每個點上，選擇行號左側裝訂邊中的空白區域。即會顯示紅色圓圈。

若要移除中斷點，請在裝訂邊中選擇現有的中斷點。

若要停用中斷點，而非將其移除，請在 Debugger (除錯器) 視窗、Breakpoints (中斷點) 中，清除對應至您希望停用之中斷點的方塊。若要再次啟用中斷點，請選取您清除的方塊。

若要一次停用所有中斷點，請在 Debugger (除錯器) 視窗中選擇 Deactivate All Breakpoints (停用所有中斷點)。若要再次啟用所有中斷點，請選擇 Activate All Breakpoints (啟用所有中斷點)。

如果 Debugger (除錯器) 視窗不可見，請選擇 Debugger (除錯器) 按鈕。如果 Debugger (除錯器) 按鈕不可見，請在選單列上選擇 Window (視窗)、Debugger (除錯器)。

3. 在您希望取得值的暫停點上設定任一監看運算式，如下所示：
 - a. 在 Debugger (除錯器) 視窗的 Watch Expressions (監看運算式) 中，選擇 Type an expression here (在此處輸入運算式)。
 - b. 輸入您希望監看的運算式，然後按下 Enter。

若要變更現有的監看運算式，請用滑鼠右鍵按一下監看運算式，然後選擇 Edit Watch Expression (編輯監看運算式)。輸入變更，然後按下 Enter。

若要移除現有的監看運算式，請用滑鼠右鍵按一下監看運算式，然後選擇 Remove Watch Expression (移除監看運算式)。

4. 如「[執行您的程式碼](#)」中所述來執行您的程式碼。

每當執行暫停，您可以將指標停留在顯示的任何程式碼片段 (例如變數) 上，以便在工具提示中顯示任何可用的相關資訊。

變更內建執行器

1. 在選單列上，選擇 Run, Run With (執行、執行工具)，然後選擇您希望變更的內建執行器。
2. 選擇執行組態標籤上顯示的 Stop (停止) 使執行器停止嘗試執行您的程式碼。
3. 選擇 Runner: My Runner (執行器：我的執行器)，其中 My Runner (我的執行器) 即為您希望變更的執行器名稱，然後選擇 Edit Runner (編輯執行器)。
4. 在顯示的 My Runner.run (My Runner.run) 索引標籤上變更執行器目前的定義。請參閱「[定義建置器或執行器](#)」。
5. 選擇 File, Save As (檔案、另存新檔)。將檔案以相同名稱 (My Runner.run) 儲存在 my-environment/.c9/runners 目錄中，其中 my-environment 即為您的 AWS Cloud9 開發環境名稱。

Note

您對內建執行器所做的任何變更僅適用於您進行變更的環境。若要將變更套用至個別的環境，請開啟其他環境，然後遵循上述步驟開啟、編輯和儲存相同的變更至該內建執行器。

建立執行組態

在選單列上，選擇 Run, Run Configurations, New Run Configuration (執行、執行組態、新增執行組態)。請在顯示的執行組態標籤上執行下列動作：

1. 在 Run (執行) 和 Restart (重新啟動) 旁的方塊中，輸入將會針對此執行組態顯示在 Run, Run Configurations (執行、執行組態) 選單上的名稱。
2. 在 Command (命令) 方塊中輸入您希望使用的任何自訂命令列選項。
3. 若要讓此執行組態使用執行器預先定義的除錯設定，請選擇 Run in Debug Mode (以除錯模式執行)。錯誤圖示將會在白色背景中轉換為綠色。
4. 若要讓此執行組態使用特定的工作目錄，請選擇 CWD、選擇要使用的目錄，然後選擇 Select (選取)。
5. 若要讓此執行組態使用特定的環境變數，請選擇 ENV，然後輸入每個環境變數的名稱和值。

若要使用此執行組態，請開啟對應至您希望執行的程式碼檔案。請在選單列上，選擇 Run, Run Configurations (執行、執行組態)，然後選擇此執行組態名稱。在顯示的執行組態標籤中，選擇 Runner : Auto (執行器：自動)，選擇您希望使用的執行器，然後選擇 Run (執行)。

Note

任何您建立的執行組態僅適用於您建立該執行組態的環境。若要將該執行組態新增至不同的環境，請開啟其他環境，然後遵循上述步驟在該環境中建立相同的執行組態。

建立建置器或執行器

1. 若要建立建置器，請在選單列上，選擇 Run, Build System, New Build System (執行、建置系統、新增建置系統)。若要建立執行器，請在選單列上，選擇 Run, Run With, New Runner (執行、執行工具、新增執行器)。
2. 在顯示的建置器標籤 (標記為 My Builder.build) 或執行器標籤 (標記為 My Runner.run (我的 Runner.run)) 上定義建置器或執行器。請參閱「[定義建置器或執行器](#)」。
3. 在您定義建置器或執行器後，請選擇 File, Save As (檔案、另存新檔)。針對建置器，請使用 .build 副檔名將檔案儲存在 my-environment/.c9/builders 目錄中，其中 my-environment 即為您的環境名稱。針對執行器，請使用 .run 檔案副檔名將檔案儲存在 my-environment/.c9/runners 目錄中，其中 my-environment 即為您環境的名稱。您指定的檔

案名稱將會顯示在 Run, Build System (執行、建置系統) 選單 (或建置器) 上，或 Run, Run With (執行、執行工具) 選單 (或執行器) 上。因此，除非您指定不同的檔案名稱，預設情況下顯示名稱將會是 My Builder (我的建置器) (針對建置器) 或 My Runner (我的執行器) (針對執行器)。

若要使用此建置器或執行器，請參閱「[建置專案檔案](#)」或「[執行您的程式碼](#)」。

Note

任何您建立的建置器或執行器僅適用於您建立建置器或執行器的環境。若要將該建置器或執行器新增至不同的環境，請開啟其他環境，然後遵循上述步驟在該環境中建立相同的建置器或執行器。

定義建置器或執行器

此程序假設您已透過選擇 Run, Build System, New Build System (執行、建置系統、新增建置系統) (針對建置器) 或 Run, Run With, New Runner (執行、執行工具、新增執行器) (針對執行器) 來開始建立建置器或執行器。

在顯示的建置器或執行器標籤上，使用 JSON 來定義執行器或建置器。開始使用下列程式碼做為範本。

針對建置器，請使用此程式碼。

```
{
  "cmd": [],
  "info": "",
  "env": {},
  "selector": ""
}
```

針對執行器，請使用此程式碼。

```
{
  "cmd": [],
  "script": "",
  "working_dir": "",
  "info": "",
  "env": {},
}
```

```
"selector": "",
"debugger": "",
"debugport": ""
}
```

在上述程式碼中：

- `cmd`：代表 AWS Cloud9 的逗號分隔字串清單，用於執行單一命令。

當 AWS Cloud9 執行此命令時，清單中的每個字串會以單一空格分隔。例如，AWS Cloud9 將以 `ls $file $args` 來執行 `"cmd": ["ls", "$file", "$args"]`，其中 AWS Cloud9 會將 `$file` 取代為目前檔案的完整路徑，以及將 `$args` 取代為在檔案名稱後輸入的任何引數。如需詳細資訊，請參閱本節後文所述之支援的變數清單。

- `script`：代表終端機中執行器執行的 `bash` 指令碼 (如需可讀性，也可以指定為行陣列)。
- `working_dir`：代表執行器將在其中執行的目錄。
- `info`：代表您希望在執行開始時向使用者顯示的任何文字字串。此字串可以包含變數，例如 `Running $project_path$file_name...`，其中 AWS Cloud9 會將 `$project_path` 取代為目前檔案的目錄路徑，以及將 `$file_name` 取代為目前檔案的名稱部分。請參閱本節後文所述之支援的變數清單。
- `env`：代表 AWS Cloud9 使用的任何命令列引數陣列，例如：

```
"env": {
  "LANG": "en_US.UTF-8",
  "SHLVL": "1"
}
```

- `selector`：代表您希望 AWS Cloud9 使用的任何規則運算式，用於識別適用於本執行器的檔案名稱。例如，您可以指定 `source.py` 用於 Python 檔案。
- `debugger`：代表您想要 AWS Cloud9 使用之與此執行器相容的任何可用除錯器名稱。例如，您可以指定 `v8` 用於 V8 除錯器。
- `debugport`：代表您希望 AWS Cloud9 在除錯期間使用的連接埠號碼。例如，您可以指定 `15454` 用於連接埠號碼。

下列資料表顯示您可以使用的變數。

變數	Description (描述)
<code>\$file_path</code>	目前檔案的目錄，例如 <code>/home/ec2-user/environment</code> 或 <code>/home/ubuntu/environment</code> 。
<code>\$file</code>	目前檔案的完整路徑，例如 <code>/home/ec2-user/environment/hello.py</code> 或 <code>/home/ubuntu/environment/hello.py</code> 。
<code>\$args</code>	在檔案名稱後輸入的任何引數，例如 "5" "9"。
<code>\$file_name</code>	目前檔案的名稱部分，例如 <code>hello.py</code> 。
<code>\$file_extension</code>	目前檔案的副檔名，例如 <code>py</code> 。
<code>\$file_base_name</code>	不含檔案副檔名的目前檔案名稱，例如 <code>hello</code> 。
<code>\$packages</code>	套件資料夾的完整路徑。
<code>\$project</code>	目前專案資料夾的完整路徑。
<code>\$project_path</code>	目前專案檔案的目錄，例如 <code>/home/ec2-user/environment/</code> 或 <code>/home/ubuntu/environment/</code> 。
<code>\$project_name</code>	不含檔案副檔名的目前專案檔案名稱，例如 <code>my-demo-environment</code> 。
<code>\$project_extension</code>	目前專案檔案的副檔名。
<code>\$project_base_name</code>	不含副檔名的目前專案檔案名稱。
<code>\$hostname</code>	環境的主機名稱，例如 <code>192.0.2.0</code> 。
<code>\$hostname_path</code>	含專案檔案相關路徑的環境主機名稱，例如 <code>https://192.0.2.0/hello.js</code> 。

變數	Description (描述)
\$url	存取環境的完整 URL，例如 https://192.0.2.0。
\$port	指派給環境的連接埠，例如 8080。
\$ip	對環境執行程序的 IP 地址，例如 0.0.0.0。

舉例來說，下列名為 `G++.build` 的建置器檔案會針對使用 `-o` 選項執行 `g++` 命令的 GCC 來定義建置器，以將目前檔案 (例如 `hello.cpp`) 編譯至物件模組。然後會將物件模組以與目前檔案相同的名稱 (例如 `hello`) 連結至程式。在此處，相同的命令為 `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"
}
```

再舉另一個例子，下列名為 `Python.run` 的執行器檔案會定義一個執行器，該執行器會使用所提供的任何引數來使用 Python 執行目前檔案。例如，如果目前檔案名為 `hello.py`，且已提供 5 和 9 引數，則相同的命令為 `python hello.py 5 9`。

```
{
  "cmd": [ "python", "$file_name", "$args" ],
  "info": "Running $file_name...",
  "selector": "source.py"
}
```

最後，下列名為 `Print Run Variables.run` 的執行器檔案會定義一個執行器，該執行器在輸出每個可用變數的值之後即停止。

```
{
  "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 整合開發環境 (IDE) 中使用自訂環境變數

AWS Cloud9 IDE 支援設定自訂環境變數。您可以藉由下列方式在 AWS Cloud9 IDE 中設定自訂環境變數。

- [設定命令層級自訂環境變數](#)
- [在 ~/.bash_profile 中設定自訂使用者環境變數](#)
- [設定本機自訂環境變數](#)
- [在 ~/.bashrc 中設定自訂使用者環境變數](#)
- [在 ENV 清單中設定自訂環境變數](#)

設定命令層級自訂環境變數

當您在 AWS Cloud9 開發環境中執行命令時，您可以設定命令層級的自訂環境變數。若要測試此行為，請以下列程式碼建立名為 `script.sh` 的檔案：

```
#!/bin/bash  
  
echo $MY_ENV_VAR
```

如果您執行下列命令，終端機會顯示 Terminal session：

```
MY_ENV_VAR='Terminal session' sh ./script.sh
```

如果您使用本主題所述的多個方法來設定自訂環境變數，那麼當您嘗試取得自訂環境變數的值，此設定的優先順序將高於所有其他設定。

在 ~/.bash_profile 中設定自訂使用者環境變數

您可以在環境的 `~/.bash_profile` 檔案中設定自訂使用者環境變數。若要測試此行為，請將下列程式碼新增至您環境中的 `~/.bash_profile` 檔案：

```
export MY_ENV_VAR='.bash_profile file'
```

如果您接下來從命令列執行 `sh ./script.sh`，終端機會顯示 `.bash_profile file`。(假設您已建立 `script.sh` 檔案，如稍早所述。)

設定本機自訂環境變數

您可以藉由執行 `export` 命令，在終端機工作階段設定本機自訂環境變數。若要測試此行為，請在終端機工作階段執行下列命令：

```
export MY_ENV_VAR='Command line export'
```

如果您接下來從命令列執行 `sh ./script.sh`，終端機會顯示 `Command line export`。(假設您已建立 `script.sh` 檔案，如稍早所述。)

如果您在 `~/.bash_profile` 檔案中以 `export` 命令設定相同的自訂環境變數，則當您嘗試取得自訂環境變數的值時，`export` 命令設定將具有最高優先順序。

在 `~/.bashrc` 中設定自訂使用者環境變數

您可以在環境的 `~/.bashrc` 檔案中設定自訂使用者環境變數。若要測試此行為，請將下列程式碼新增至您環境中的 `~/.bashrc` 檔案：

```
export MY_ENV_VAR='.bashrc file'
```

如果您接下來從命令列執行 `sh ./script.sh`，終端機會顯示 `.bashrc file`。(假設您已建立 `script.sh` 檔案，如稍早所述。)

如果您在 `~/.bashrc` 檔案中以 `export` 命令設定相同的自訂環境變數，則當您嘗試取得自訂環境變數的值時，`export` 命令設定將具有最高優先順序。

在 ENV 清單中設定自訂環境變數

您可以在 Run (執行) 標籤上的 ENV 清單中設定自訂環境變數。

若要測試此行為，請執行下列動作：

1. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Shell script (Shell 指令碼)。

3. 選擇 ENV，然後為 MY_ENV_VARName (名稱) 輸入，並為 ENV listValue (值) 輸入。
4. 在 Command (命令) 中輸入 `./script.sh`。
5. 選擇 Run (執行) 按鈕。執行器標籤顯示 ENV list。(假設您已建立 `script.sh` 檔案，如稍早所述。)

如果您在 `~/.bash_profile` 檔案 (透過 `export` 命令)、`~/.bashrc` 檔案以及 ENV 清單中設定相同的自訂環境變數，則當您嘗試取得自訂環境變數的值時，`~/.bash_profile` 檔案設定將具有最高優先順序，緊接著為 `export` 命令設定、`~/.bashrc` 檔案設定以及 ENV 清單設定。

Note

ENV 清單是藉由使用程式碼取得和設定自訂環境變數的唯一方法，與 shell 指令碼不同。

使用 AWS Cloud9 整合式開發環境 (IDE) 中的專案設定

專案設定僅會套用於目前的 AWS Cloud9 開發環境，其中包含下列設定類型：

- 程式碼編輯器設定，例如是否使用軟標籤和新檔案行尾
- 要忽略的檔案類型
- 要顯示或隱藏的提示和警告類型
- JavaScript、PHP、Python 和 Go 等程式設計語言的程式碼和格式設定
- 在執行和建置程式碼時使用的組態類型

雖然專案設定只套用至單一環境，但是您可以將一個環境的專案設定套用至任何其他環境。

- [檢視或變更專案設定](#)
- [將一個環境的目前專案設定套用至另一個環境](#)
- [您所做的專案設定變更](#)

檢視或變更專案設定

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 若要檢視目前環境的專案設定，請在 Preferences (偏好設定) 標籤的側邊導覽窗格中，選擇 Project Settings (專案設定)。

3. 若要變更環境的目前專案設定，請在 Project Settings (專案設定) 窗格中變更您想要的設定。

請參閱「[您可以做的專案設定變更](#)」。

將一個環境的目前專案設定套用至另一個環境

1. 在來源和目標環境兩處，從 AWS Cloud9 IDE 的選單列選擇 AWS Cloud9, Open Your Project Settings (開啟專案設定)。
2. 在來源環境中，複製顯示的 project.settings 標籤內容。
3. 在目標環境中，以從來源環境複製的內容覆寫 project.settings 標籤的內容。
4. 在目標環境中，儲存 project.settings 標籤。

您可以變更的專案設定

這些章節說明 Preferences (偏好設定) 標籤之 Project Settings (專案設定) 窗格上您可變更的專案設定類型。

- [EC2 執行個體](#)
- [程式碼編輯器 \(Ace\)](#)
- [在檔案中尋找](#)
- [提示與警告](#)
- [JavaScript 支援](#)
- [建置](#)
- [執行與偵錯](#)
- [執行組態](#)
- [程式碼格式器](#)
- [TypeScript 支援](#)
- [PHP 支援](#)
- [Python 支援](#)
- [Go 支援](#)

EC2 instance (EC2 執行個體)

Stop my environment (停止我的環境)

選擇在您關閉連線到該環境 IDE 的所有 Web 瀏覽器執行個體後，自動停止環境的 Amazon EC2 執行個體 (如有使用) 的時機。您可以選擇從一週到 30 分鐘的時間範圍。您也可以選擇在離開 AWS Cloud9 IDE 後一律不要自動停止 Amazon EC2 執行個體。

如果您想在結束使用 IDE 後的 30 分鐘之內停止執行個體，可以[使用主控台界面手動停止](#)。

程式碼編輯器 (Ace)

Soft tabs (軟標籤)

如果選取，則每次按 Tab 都會插入指定數目的空格，而不是 Tab 字元。

Autodetect tab size on load (載入時自動偵測標籤大小)

如果選取，則 AWS Cloud9 會嘗試猜測標籤大小。

New file line endings (新檔案行尾)

用於新檔案的行尾類型。

有效選項包括下列：

- Windows (CRLF) 使用歸位然後使用換行結束行。
- Unix (LF) 只使用換行結束行。

On save, strip whitespace (儲存時、去除空格)

如果選取，則每次儲存檔案時，AWS Cloud9 都會嘗試從檔案移除其認為不必要的空格和標籤。

在檔案中尋找

Ignore these Files (忽略這些檔案)

在檔案中尋找時，AWS Cloud9 忽略的類型檔案。

Maximum number of files to search (in 1000) (要搜尋的檔案數目上限 (單位為 1000))

在檔案中尋找時，AWS Cloud9 在目前範圍中尋找的檔案數目上限 (單位為 1,000 的倍數)。

提示與警告

警告層級

要啟用的訊息層級下限。

有效值包括以下項目：

- Info (資訊) 以啟用資訊、警告和錯誤訊息。
- Warning (警告) 僅啟用警告和錯誤訊息。
- Error (錯誤) 僅啟用錯誤訊息。

Mark Missing Optional Semicolons (標記遺失選用分號)

如果啟用，則 AWS Cloud9 每次發現可在程式碼中使用但未使用的分號時，都會在檔案中予以標記。

Mark Undeclared Variables (標記未宣告變數)

如果啟用，則 AWS Cloud9 每次發現程式碼中有未宣告變數時，都會在檔案中予以標記。

Mark Unused Function Arguments (標記未使用的函數引數)

如果啟用，則 AWS Cloud9 每次發現函數中有未使用引數時，都會在檔案中予以標記。

Ignore Messages Matching Regex (忽略符合 Regex 的訊息)

AWS Cloud9 將不會顯示任何符合指定規則運算式的訊息。如需詳細資訊，請參閱 Mozilla Developer Network 上 JavaScript 正規運算式主題中的[撰寫正規運算式模式](#)。

JavaScript 支援

使用 .eslintrc 自訂 JavaScript 警告

如果啟用，則 AWS Cloud9 會使用 .eslintrc 檔案，判斷要啟用或停用哪些 JavaScript 警告。如需詳細資訊，請參閱 ESLint 網站上的[組態檔格式](#)。

JavaScript 程式庫程式碼完成

AWS Cloud9 用來嘗試建議或執行自動程式碼完成的 JavaScript 程式庫。

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 每次儲存 JavaScript 檔案時都會嘗試格式化該檔案中的程式碼。

Use builtin JSBeautify as code formatter (使用內建 JSBeautify 做為程式碼格式器)

如果啟用，則 AWS Cloud9 會使用其內部 JSBeautify 實作，嘗試提高檔案中程式碼的可讀性。

Custom code formatter (自訂程式碼格式器)

在格式化 JavaScript 檔案中的程式碼時供 AWS Cloud9 嘗試執行的命令。

組建

Builder path in environment (環境中的建置器路徑)

任何自訂建置組態的路徑。

執行與除錯

Runner path in environment (環境中的執行器路徑)

任何自訂執行組態的路徑。

Preview URL (預覽 URL)

用來預覽環境的應用程式的 URL。

執行組態

此環境的自訂執行組態。

Remove selected configs (移除選取的組態)

刪除選取的執行組態。

Add new config (新增組態)

建立新執行組態。

Set as default (設定為預設)

將選取的執行組態設定為預設執行組態。

程式碼格式器

JSBeautify 設定

提高檔案中程式碼可讀性的設定。

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 只要儲存程式碼檔案就會嘗試套用 JSBeautify 設定。

針對 JavaScript 使用 JSBeautify

如果啟用，則 AWS Cloud9 只要儲存 JavaScript 檔案就會嘗試套用 JSBeautify 設定。

Preserve empty lines (保留空白行)

如果啟用，則 AWS Cloud9 不會移除程式碼檔案中的空白行。

Keep array indentation (保持陣列縮排)

如果啟用，則 AWS Cloud9 會保留程式碼檔案的陣列中之元素宣告的縮排。

JSLint strict whitespace (JSLint 嚴格空格)

如果啟用，則 AWS Cloud9 會嘗試在程式碼檔案中套用 JSLint 空格規則。如需詳細資訊，請參閱 [JSLint 協助](#) 中的「空格」。

Braces (大括號)

指定程式碼中大括號的對齊方式。

有效值包括以下項目：

- Braces with control statement (具有控制陳述式的大括號) 移動每個左右大括號以符合其相關控制陳述式 (需要時)。

例如，此程式碼的格式如下：

```
for (var i = 0; i < 10; i++) { if (i == 5) { console.log("Halfway done.") } }
```

儲存檔案時，會變成此程式碼：

```
for (var i = 0; i < 10; i++) {  
    if (i == 5) {  
        console.log("Halfway done.")  
    }  
}
```

- Braces on own line (大括號單獨一行) 視需要將每個大括號另起一行。

例如，此程式碼的格式如下：

```
for (var i = 0; i < 10; i++) { if (i == 5) { console.log("Halfway done.") }}
```

儲存檔案時，會變成此程式碼：

```
for (var i = 0; i < 10; i++) {if (i == 5)
{
    console.log("Halfway done.")
}
}
```

- End braces on own line (右大括號單獨一行) 視需要將每個大右括號另起一行。

例如，此程式碼的格式如下：

```
for (var i = 0; i < 10; i++) {
    if (i == 5) { console.log("Halfway done.") }
}
```

儲存檔案時，會變成此程式碼：

```
for (var i = 0; i < 10; i++) {
    if (i == 5) {
        console.log("Halfway done.")
    }
}
```

Preserve inline blocks (保留內嵌區塊)

若已啟用，AWS Cloud9 不會嘗試移動內嵌區塊的左右括號來分隔行 (如果那些括號位於同一行的話)。

Space before conditionals (條件式前面的空格)

如果啟用，則 AWS Cloud9 會視需要在每個條件式宣告前新增一個空格。

Unescape strings (未逸出字串)

如果啟用，則 AWS Cloud9 會將逸出字串轉換為其非逸出字串。例如，將 `\n` 轉換為換行字元，並將 `\r` 轉換為歸位字元。

Indent inner HTML (縮排內部 HTML)

如果啟用，則 AWS Cloud9 會縮排 HTML 程式碼中的 `<head>` 和 `<body>` 區段。

TypeScript 支援

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 只要儲存 TypeScript 檔案就會嘗試格式化 TypeScript 程式碼。

Custom code formatter (自訂程式碼格式器)

TypeScript 程式碼之任何自訂程式碼格式組態的路徑。

PHP 支援

Enable PHP code completion (啟用 PHP 程式碼完成)

如果啟用，則 AWS Cloud9 會嘗試完成 PHP 程式碼。

PHP completion include paths (PHP 完成 Include 路徑)

AWS Cloud9 用來嘗試協助完成 PHP 程式碼的位置。例如，如果您有想要 AWS Cloud9 用於完成的自訂 PHP 檔案，而且這些檔案位在 `~/environment` 目錄中的某個位置，則請將 `~/environment` 新增至此路徑。

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 只要儲存 PHP 檔案就會嘗試格式化 PHP 程式碼。

Custom code formatter (自訂程式碼格式器)

PHP 程式碼之任何自訂程式碼格式組態的路徑。

Python 支援

Enable Python code completion (啟用 Python 程式碼完成)

如果啟用，則 AWS Cloud9 會嘗試完成 Python 程式碼。若要設定 AWS Cloud9 用於完成 Python 程式碼的路徑，請使用 `PYTHONPATH` 設定。

Python version (Python 版本)

指定要使用的 Python 版本。

Pylint command line options (Pylint 命令列選項)

AWS Cloud9 用於 Pylint 與 Python 程式碼搭配的選項。如需詳細資訊，請參閱 Pylint 網站上的 [Pylint 使用者手冊](#)。

PYTHONPATH

供 AWS Cloud9 使用之 Python 程式庫和套件的路徑。例如，如果您在 `~/environment` 目錄中有自訂 Python 程式庫和套件，則請將 `~/environment` 新增至此路徑。

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 只要儲存 Python 檔案就會嘗試格式化 Python 程式碼。

Custom code formatter (自訂程式碼格式器)

Python 程式碼之任何自訂程式碼格式組態的路徑。

Go 支援

Enable Go code completion (啟用 PHP 程式碼完成)

如果啟用，則 AWS Cloud9 會嘗試完成 Go 程式碼。

Format Code on Save (儲存時格式化程式碼)

如果啟用，則 AWS Cloud9 只要儲存 Go 檔案就會嘗試格式化 Go 程式碼。

Custom code formatter (自訂程式碼格式器)

Go 程式碼之任何自訂程式碼格式組態的路徑。

手動停止環境的 EC2 執行個體

[EC2 執行個體](#) 設定可讓您在關閉所有連線到 IDE 的 Web 瀏覽器執行個體的 30 分鐘後，立即自動停止環境的 Amazon EC2 執行個體。

您也可以使用主控台立即手動停止執行個體。

如何手動停止環境的 EC2 執行個體

1. 關閉了所有連線到 IDE 的 Web 瀏覽器執行個體後，請在 AWS Cloud9 主控台選擇 Your environments (您的環境)。

2. 在顯示所用環境詳細資訊的窗格選擇右上角的按鈕，然後選擇 View details (檢視詳細資訊)。
3. 在 Environment details (環境詳細資訊) 的 EC2 Instance (EC2 執行個體) 中，選擇 Go To Instance (前往執行個體)。
4. 在 Amazon EC2 主控台的 Instance state (執行個體狀態) 下，選擇核取方塊來選取您環境的執行個體。Instance state (執行個體狀態) 可能表示執行個體仍在執行中。
5. 選擇 Instance state (執行個體狀態)，然後選取 Stop instance (停止執行個體)。
6. 出現確認提示時，請選擇 Stop (停止)。停止執行個體可能需要幾分鐘。

使用 AWS Cloud9 IDE 中的使用者設定

使用者設定是套用於與 AWS Identity and Access Management (IAM 使用者) 相關聯的各個 AWS Cloud9 開發環境的設定。它們包括下列設定：

- 一般使用者界面行為，例如啟用動畫和標記變更的標籤
- 檔案系統導覽設定
- 檔案尋找和搜尋設定
- 終端機工作階段和輸出的色彩配置
- 其他程式碼編輯器設定，例如程式碼摺疊、選取完整行、捲動動畫和字型大小

如果您變更使用者設定，AWS Cloud9 會將變更推送到雲端，並與您的 IAM 使用者建立關聯。AWS Cloud9 也會持續掃描雲端中與您的 IAM 使用者相關聯的使用者設定是否有變更，並將變更套用到目前的環境。無論您在何種 AWS Cloud9 環境中工作，都可以使用此操作，體驗相同的外觀和風格。

Note

若要存放和擷取您的 IDE 設定，AWS Cloud9 會使用內部 API GetUserSettings 和 UpdateUserSettings。

您可以與其他使用者共享您的使用者設定，如下所示：

- [檢視或變更您的使用者設定](#)
- [與其他使用者共享您的使用者設定](#)
- [您所做的使用者設定變更](#)

檢視或變更您的使用者設定

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 若要檢視各個環境的使用者設定，請在 Preferences (偏好設定) 標籤的側邊導覽窗格中，選擇 User Settings (使用者設定)。
3. 在 User Settings (使用者設定) 窗格中，變更各個環境的使用者設定。
4. 若要將您的變更套用到任何其他環境，開啟該環境即可。如果該環境已開啟，請重新整理該環境的 Web 瀏覽器標籤。

如需詳細資訊，請參閱「[您可以變更的使用者設定](#)」。

與其他使用者共享您的使用者設定

1. 在來源和目標環境兩處，從 AWS Cloud9 IDE 的選單列選擇 AWS Cloud9, Open Your User Settings (開啟使用者設定)。
2. 在來源環境中，複製顯示的 user.settings 標籤內容。
3. 在目標環境中，以從來源環境複製的內容覆寫 user.settings 標籤的內容。
4. 在目標環境中，儲存 user.settings 標籤。

您所做的使用者設定變更

這些章節說明您在 Preferences (偏好設定) 標籤的 User Settings (使用者設定) 窗格中可以變更的使用者設定類型：


- [一般](#)
- [使用者界面](#)
- [協作](#)
- [Tree 和 Go 面板](#)
- [在檔案中尋找](#)
- [中繼資料](#)
- [監看器](#)
- [終端機](#)
- [輸出](#)
- [程式碼編輯器 \(Ace\)](#)

- [輸入](#)
- [提示與警告](#)
- [執行與偵錯](#)
- [預覽版](#)
- [建置](#)

一般

Reset to Factory Settings (重設為原廠設定)

如果您選擇 Reset to Default (重設為預設) 按鈕，AWS Cloud9 會將您所有的使用者設定重設為 AWS Cloud9 預設使用者設定。若要確認，請選擇 Reset settings (重設設定)。

 Warning
您無法復原此動作。

在離開前警告

每當您嘗試關閉 IDE 時，AWS Cloud9 會要求您確認是否要結束。

使用者界面

Enable UI Animations (啟用 UI 動畫)

AWS Cloud9 會在 IDE 中使用動畫。

Use an Asterisk (*) to Mark Changed Tabs (使用星號 (*) 標記已變更的標籤)

AWS Cloud9 會將星號 (*) 新增至已變更但尚未儲存內容的標籤。

Display Title of Active Tab as Browser Title (將作用中標籤的標題顯示為瀏覽器標題)

AWS Cloud9 會將相關聯之 Web 瀏覽器標籤的標題變更為作用中標籤的標題 (例如，Untitled1、hello.js、Terminal (終端機)、Preference (偏好設定))。

Automatically Close Empty Panes (自動關閉空白窗格)

當您重新載入環境時，AWS Cloud9 會自動關閉環境認為空白的任何窗格。

Environment Files Icon and Selection Style (環境檔案圖示和選取項目樣式)

為環境檔案使用的圖示 AWS Cloud9，及 AWS Cloud9 使用的檔案選取行為。

有效值包含：

- Default (預設) - AWS Cloud9 會使用預設圖示和預設檔案選取行為。
- Alternative (其他) - AWS Cloud9 會使用其他圖示和其他檔案選取行為。

協作

Disable collaboration security warning (停用協同合作安全警告)

當讀取/寫入成員新增至環境時，AWS Cloud9 不會顯示安全警告對話方塊。

Show Authorship Info (顯示作者來源資訊)

AWS Cloud9 會將其他環境成員輸入的文字加底線，並在裝訂邊中顯示相關的重點提示。

Tree 和 Go 面板

Scope Go to Anything to Favorites (將移至任何項目的範圍設為我的最愛)

Go (移至) 視窗中的 Go to File (移至檔案) 僅會顯示 Environment (環境) 視窗中 Favorites (我的最愛) 範圍內的結果。

Enable Preview on Tree Selection (在樹狀目錄選取項目上啟用預覽)

只要按一下滑鼠 (不需按兩下)，AWS Cloud9 就會顯示選擇的檔案。

Hidden File Pattern (隱藏的檔案模式)

要讓 AWS Cloud9 視為隱藏的檔案類型。

Reveal Active File in Project Tree (在專案樹狀目錄中顯示作用中檔案)

AWS Cloud9 會在 Environment (環境) 視窗中反白作用中檔案。

Download Files As (將檔案下載為)

AWS Cloud9 在下載檔案時使用的行為。

有效值包括以下項目：

- auto (自動) - AWS Cloud9 無須修改即可下載檔案。
- tar.gz - AWS Cloud9 可將檔案下載為壓縮的 TAR 檔。

- zip - AWS Cloud9 可將檔案下載為 .zip 檔。

Find in Files (在檔案中尋找)

Search In This Path When 'Project' Is Selected (當選取 'Project' (專案) 時，在此路徑搜尋)

在 find in files (在檔案中尋找) 列上，當選取 Project (專案) 做為搜尋範圍時，要在其中尋找的路徑。

Show Full Path in Results (在結果中顯示完整路徑)

會在 Search Results (搜尋結果) 標籤中顯示每個相符檔案的完整路徑。

Clear Results Before Each Search (在每次搜尋前先清除結果)

會在目前的搜尋開始前，先清除任何先前搜尋結果的 Search Results (搜尋結果) 標籤。

Scroll Down as Search Results Come In (在找到搜尋結果時向下捲動)

會在找出搜尋結果時，將 Search Results (搜尋結果) 標籤捲動到結果清單底部。

Open Files when Navigating Results with (Up and Down) (在使用 (向上和向下鍵) 導覽結果時開啟檔案)

會在結果清單中的 Search Results (搜尋結果) 標籤中按向上和向下鍵時，開啟每個相符的檔案。

中繼資料

Maximum of Undo Stack Items in Meta Data (中繼資料內復原堆疊項目的上限)

AWS Cloud9 在其可復原的動作清單中保留的項目數上限。

監看器

Auto-Merge Files When a Conflict Occurs (在發生衝突時自動合併檔案)

當發生合併衝突時，AWS Cloud9 會嘗試自動合併檔案。

終端機

文字顏色

Terminal (終端機) 標籤中的文字色彩。

Background Color (背景色彩)

Terminal (終端機) 標籤中的背景色彩。

Selection Color (選取項目色彩)

Terminal (終端機) 標籤中所選文字的色彩。

Font Family (字型系列)

Terminal (終端機) 標籤中的字型樣式。

字型大小

Terminal (終端機) 標籤中的文字大小。

Antialiased Fonts (去鋸齒化字型)

AWS Cloud9 會嘗試使 Terminal (終端機) 標籤中的文字顯示變得平滑。

Blinking Cursor (閃爍游標)

AWS Cloud9 會持續閃爍 Terminal (終端機) 標籤中的游標。

Scrollback (向後捲動)

您可以在 Terminal (終端機) 標籤中向上或向後捲動的行數。

Use AWS Cloud9 as the Default Editor (使用 AWS Cloud9 做為預設編輯器)

使用 AWS Cloud9 做為預設文字編輯器。

輸出

文字顏色

顯示輸出之標籤中的文字色彩。

Background Color (背景色彩)

顯示輸出之標籤中文字的背景色彩。

Selection Color (選取項目色彩)

顯示輸出之標籤中所選文字的色彩。

Warn Before Closing Unnamed Configuration (在關閉未命名組態前警告)

AWS Cloud9 會在組態標籤關閉前，提示您儲存任何未儲存的組態標籤。

Preserve log between runs (保留執行之間的日誌)

AWS Cloud9 會保留所有嘗試執行的日誌。

程式碼編輯器 (Ace)

Auto-pair Brackets, Quotes, etc. (自動配對括號、引號等)

AWS Cloud9 會嘗試在編輯器標籤中為輸入的每個相關起始字元 (例如括號、引號和大括號)，新增相配的結尾字元。

Wrap Selection with Brackets, Quote, etc. (用括號、引號等括住選取項目)

AWS Cloud9 會在選取文字並輸入相關起始字元 (例如括號、引號和大括號) 後，在編輯器標籤中於文字結尾插入相配的結尾字元。

Code Folding (程式碼摺疊)

AWS Cloud9 會根據相關的程式碼語法規則，嘗試在編輯器標籤中顯示、展開、隱藏或摺疊程式碼區段。

Fade Fold Widgets (淡化摺疊 Widget)

當您在編輯器標籤中將滑鼠游標暫停於這些控制項上時，AWS Cloud9 就會在裝訂邊中顯示程式碼摺疊控制項。

以空白選取範圍複製

AWS Cloud9 可讓您複製和/或剪下文字，此選項決定是否將空白文字複製到剪貼簿。

Full Line Selection (選取完整行)

AWS Cloud9 會在編輯器標籤中選取按三下的完整行。

Highlight Active Line (醒目提示作用中的行)

AWS Cloud9 會在編輯器標籤中反白整個作用中的行。

Highlight Gutter Line (醒目提示裝訂邊行)

AWS Cloud9 會在編輯器標籤內反白作用中行旁的裝訂邊位置。

Show Invisible Characters (顯示隱藏字元)

AWS Cloud9 會在編輯器標籤中顯示其認為隱藏的字元，例如歸位字元與換行字元、空格和定位字元。

Show Gutter (顯示裝訂邊)

AWS Cloud9 顯示裝訂邊。

Show Line Numbers (顯示行號)

在裝訂邊中顯示行號的行為。

有效值包括以下項目：

- Normal (一般) - 顯示行號。
- Relative (相關) - 顯示與作用中的行相關的行號。
- None (無) - 隱藏行號。

Show Indent Guides (顯示縮排指引)

AWS Cloud9 將顯示指引，讓您可在編輯器標籤中輕鬆呈現縮排的文字。

Highlight Selected Word (醒目提示選取的文字)

AWS Cloud9 會在編輯器標籤中選取按兩下的完整文字。

Scroll Past the End of the Document (一直捲動至文件結尾)

允許使用者在編輯器標籤中一直捲動至目前檔案結尾的行為。

有效值包括以下項目：

- Off (關閉) - 不允許一直捲動至目前檔案的結尾。
- Half Editor Height (編輯器高度的一半) - 允許一直捲動至目前檔案的結尾，直至編輯器螢幕高度的一半。
- Full Editor Height (完整的編輯器高度) - 允許一直捲動至目前檔案的結尾，直至編輯器的完整螢幕高度。

Animate Scrolling (套用捲動的動畫)

AWS Cloud9 會在編輯器標籤中於捲動動作過程中套用動畫行為。

Font Family (字型系列)

要在編輯器標籤中使用的字型樣式。

字型大小

要在編輯器標籤中使用的字型大小。

Antialiased Fonts (去鋸齒化字型)

AWS Cloud9 會嘗試使編輯器標籤中的文字顯示變得平滑。

Show Print Margin (顯示列印邊界)

在編輯器標籤中於指定字元位置後方顯示垂直線。

Mouse Scroll Speed (滑鼠捲動速度)

滑鼠在編輯器標籤中捲動的相對速度。值愈大會捲動得愈快。

Cursor Style (游標樣式)

編輯器標籤中指標的樣式和行為。

有效值包含：

- Ace (略寬) - 將指標顯示為比 Slim (細) 寬的垂直線。
- Slim (細) - 將指標顯示為相對細的垂直線。
- Smooth (連續) - 將指標顯示為比 Slim (細) 寬的垂直線，且比 Slim (細) 的閃爍更連續。
- Smooth and Slim (連續和細) - 將指標顯示為比 Slim (細) 的閃爍更連續的較細垂直線。
- Wide (寬) - 將指標顯示為較寬的垂直線。

Merge Undo Deltas (合併復原差異)

- Always (一律) - 允許回復合併衝突。
- Never (永不) - 一律不允許回復合併衝突。
- Timed (限時) - 允許在指定期間後回復合併衝突。

Enable Wrapping For New Documents (為新文件啟用換行)

AWS Cloud9 會在新的檔案中將程式碼換行。

Input

Complete As You Type (在您輸入時完成)

AWS Cloud9 會在您輸入時，嘗試顯示可能的完整文字。

Complete On Enter (在按 Enter 鍵後完成)

AWS Cloud9 會在您按下 Enter 鍵後，嘗試顯示可能的完整文字。

Highlight Variable Under Cursor (在游標下醒目提示變數)

AWS Cloud9 會反白程式碼中所選變數的所有參考。

Use Cmd-Click for Jump to Definition (使用按住 Cmd 鍵並按一下跳到定義)

AWS Cloud9 會前往您按住 Mac 的 Command 鍵或 Windows 的 Ctrl 鍵時選取之程式碼的任何原始定義。

提示與警告

Enable Hints and Warnings (啟用提示與警告)

AWS Cloud9 會顯示適用的提示與警告訊息。

點擊顯示可用的快速修復

AWS Cloud9 當您按一下程式碼中的關鍵字時，會顯示包含重構建議的工具提示。

Ignore Messages Matching Regex (忽略符合 Regex 的訊息)

AWS Cloud9 不會顯示任何符合指定規則運算式的訊息。如需詳細資訊，請參閱 Mozilla Developer Network 上 JavaScript 正規運算式主題中的[撰寫正規運算式模式](#)。

執行與除錯

Save All Unsaved Tabs Before Running (在執行前先儲存所有未儲存的標籤)

在執行相關程式碼前，AWS Cloud9 會先嘗試儲存開啟標籤的所有未儲存檔案。

預覽版

Preview Running Apps (預覽正在執行的應用程式)

只要選擇 Preview (預覽) 按鈕後，AWS Cloud9 會嘗試在作用中的標籤顯示程式碼輸出預覽。

Default Previewer (預設預覽器)

AWS Cloud9 用於預覽程式碼輸出的格式。

有效值包含：

- Raw (原始) - 嘗試以一般格式顯示程式碼輸出。
- Browser (瀏覽器) - 嘗試以 Web 瀏覽器慣用的格式顯示程式碼輸出。

When Saving Reload Previewer (在儲存時重新載入預覽器)

AWS Cloud9 用於在儲存檔案時預覽程式碼輸出的行為。

有效值包括以下項目：

- Only on Ctrl-Enter (僅在按下 Ctrl-Enter 鍵時) - 在目前程式碼標籤中按下 Ctrl+Enter 時，嘗試預覽程式碼輸出。
- Always (一律) - 在儲存程式碼檔案時，嘗試預覽程式碼輸出。

組建

Automatically Build Supported Files (自動建置支援的檔案)

如果啟動建置動作，而且程式碼採用支援的格式時，AWS Cloud9 會嘗試自動建置目前的程式碼。

在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS 專案和使用者設定

AWS 服務設定位於 Preferences (偏好設定) 標籤中的 AWS Settings (設定) 窗格，包含下列設定：

- 要為 AWS Resources (AWS 資源) 視窗使用哪個 AWS 區域
- 是否使用 AWS 受管臨時憑證
- 是以純文字還是視覺化模式顯示 AWS Serverless Application Model (AWS SAM) 範本編輯器

若要檢視或變更這些設定，請在環境的 IDE 選單列中，選擇 AWS Cloud9, Preferences (AWS Cloud9、偏好設定)。

在下列清單中，專案層級設定僅會套用至目前的 AWS Cloud9 開發環境。反之，使用者層級設定會套用至與您 IAM 使用者相關聯的每個環境。如需詳細資訊，請參閱「[將某個環境的目前專案設定套用到另一個環境](#)」和「[與其他使用者共享您的使用者設定](#)」。

- [專案層級設定](#)
- [使用者層級設定](#)

專案層級設定

AWS 區域

要為 AWS Resources (資源) 視窗的 Lambda 區塊使用哪個 AWS 區域。

AWS 的受管臨時憑證

如果開啟，則在環境中從 AWS CLI、AWS CloudShell，或 AWS SDK 程式碼呼叫 AWS 服務服務時，會使用 AWS 受管臨時憑證。如需詳細資訊，請參閱 [AWS 受管臨時憑證](#)。

使用者層級設定

使用 AWS SAM 視覺化編輯器

如果開啟，則會在使用 AWS Resources (AWS 資源) 視窗的 Lambda 區塊時，於視覺化模式中顯示 AWS Serverless Application Model (AWS SAM) 範本編輯器。如果關閉，則會以文字模式顯示編輯器。

在 AWS Cloud9 整合開發環境 (IDE) 中處理按鍵繫結關係

按鍵繫結關係定義了您的快速鍵組合。按鍵繫結關係會套用於與您的 IAM 使用者相關聯的每個 AWS Cloud9 開發環境。當您變更按鍵繫結關係，AWS Cloud9 會將變更推送到雲端，並與您的 IAM 使用者建立關聯。AWS Cloud9 也會持續掃描雲端中與您的 IAM 使用者相關聯的按鍵繫結關係是否有變更，並將這些變更套用到目前的環境。

您可以與其他使用者共享您的按鍵繫結關係。

- [檢視或變更您的按鍵繫結關係](#)
- [與另一名使用者共享您的按鍵繫結關係](#)
- [變更您的鍵盤模式](#)
- [變更您的作業系統按鍵繫結關係](#)
- [變更特定的按鍵繫結關係](#)
- [移除您的所有自訂按鍵繫結關係](#)

檢視或變更您的按鍵繫結關係

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 若要檢視您每個環境的按鍵繫結關係，請從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Keybindings (按鍵繫結關係)。
3. 若要變更您每個環境的按鍵繫結關係，請在 Keybindings (按鍵繫結關係) 窗格中變更您需要的設定。

4. 若要將您的變更套用到任何環境，直接開啟該環境即可。如果該環境已開啟，請重新整理該環境的 Web 瀏覽器標籤。

如需詳細資訊，請參閱下列內容：

- [MacOS 預設按鍵繫結關係參考](#)
- [MacOS Vim 按鍵繫結關係參考](#)
- [MacOS Emacs 按鍵繫結關係參考](#)
- [MacOS Sublime 按鍵繫結關係參考](#)
- [Windows / Linux 預設按鍵繫結關係參考](#)
- [Windows / Linux Vim 按鍵繫結關係參考](#)
- [Windows / Linux Emacs 按鍵繫結關係參考](#)
- [Windows / Linux Sublime 按鍵繫結關係參考](#)

與另一名使用者共享您的按鍵繫結關係

1. 在來源和目標環境兩處，從 AWS Cloud9 IDE 的選單列選擇 AWS Cloud9, Open Your Keymap (開啟按鍵映射)。
2. 在來源環境中，複製隨後顯示的 `keybindings.settings` 標籤的內容。
3. 在目標環境中，以從來源環境複製的內容覆寫 `keybindings.settings` 標籤的內容。
4. 在目標環境中，儲存 `keybindings.settings` 標籤。

變更您的鍵盤模式

您可以變更 AWS Cloud9 IDE 在與您的 IAM 使用者相關聯的各個環境中與編輯器內的文字進行互動時使用的鍵盤模式。

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇以下其中一種鍵盤模式：
 - Default (預設) - 使用一組預設按鍵繫結關係。
 - Vim - 使用 Vim 模式。如需詳細資訊，請參閱 [Vim 說明檔案](#) 網站。
 - Emacs - 使用 Emacs 模式。如需詳細資訊，請參閱 GNU 作業系統網站上的 [Emacs 編輯器](#)。

- Sublime - 使用 Sublime 模式。如需詳細資訊，請參閱 [Sublime Text 說明文件](#) 網站。

變更您的作業系統按鍵繫結關係

您可以變更 AWS Cloud9 IDE 在與您的 IAM 使用者相關聯的各個環境中所能辨識的作業系統按鍵繫結關係設定。

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Keybindings (按鍵繫結關係)。
3. 針對 Operating System (作業系統)，選擇以下其中一種作業系統：
 - Auto (自動) 會使 AWS Cloud9 IDE 嘗試偵測該使用哪一組作業系統按鍵繫結關係。
 - MacOS 會使 AWS Cloud9 IDE 使用以 macOS 格式列出的按鍵繫結關係。
 - Windows / Linux 會使 AWS Cloud9 IDE 使用以 Windows 及 Linux 格式列出的按鍵繫結關係。

變更特定的按鍵繫結關係

您可以變更與您的 IAM 使用者相關聯的各個環境所使用的個別按鍵繫結關係。

同時變更一項按鍵繫結關係

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Keybindings (按鍵繫結關係)。
3. 在按鍵繫結關係清單的 Keystroke (按鍵輸入) 欄，開啟 (按兩下) 您要變更的按鍵繫結關係。
4. 使用鍵盤指定替代按鍵組合，然後按 Enter。

Note

若要完全移除目前的按鍵組合，Windows 或 Linux 請按 Backspace，macOS 請按 Delete。

同時變更多項按鍵繫結關係

1. 從選單列選擇 AWS Cloud9、Open Your Keymap (開啟按鍵對應)。
2. 在 `keybindings.settings` 檔案中，定義所要變更的每一項按鍵繫結關係。以下為範例語法。

```
[
  {
    "command": "addfavorite",
    "keys": {
      "win": ["Ctrl-Alt-F"],
      "mac": ["Ctrl-Option-F"]
    }
  },
  {
    "command": "copyFilePath",
    "keys": {
      "win": ["Ctrl-Shift-F"],
      "mac": ["Alt-Shift-F"]
    }
  }
]
```

在此範例中，`addFavorite` 和 `copyFilePath` 是按鍵繫結關係的名稱，其值取自於 Preferences (偏好設定) 標籤的 Keybindings (按鍵繫結關係) 窗格，如 Keystroke (按鍵輸入) 欄所示。您要用於 Windows 或 Linux 以及 macOS 的按鍵繫結關係分別為 `win` 和 `mac`。

若要套用您的變更，請儲存 `keybindings.settings` 檔案。您所做的變更會在短暫延遲後顯示於 Keybindings (按鍵繫結關係) 窗格。

移除您的所有自訂按鍵繫結關係

您可以移除所有自訂按鍵繫結關係，並將所有按鍵繫結關係回復為與您的 IAM 使用者相關聯的各個環境的預設值。

Warning

您無法復原此動作。

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Keybindings (按鍵繫結關係)。
3. 選擇 Reset to Defaults (重設為預設)。

在 AWS Cloud9 整合開發環境 (IDE) 中使用佈景主題

佈景主題定義了您的 IDE 整體的顏色。這個設定會套用於與您的 IAM 使用者相關聯的每個 AWS Cloud9 開發環境。如果您變更佈景主題，AWS Cloud9 會將這些變更推送到雲端，並將它們與您的 IAM 使用者建立關聯。AWS Cloud9 也會持續掃描雲端中與您的 IAM 使用者相關聯的佈景主題是否有變更。AWS Cloud9 會將這些變更套用到您目前的環境。

- [檢視或變更您的佈景主題](#)
- [您可以變更的整體佈景主題設定](#)
- [佈景主題覆寫](#)

檢視或變更您的佈景主題

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 若要檢視您各個環境的佈景主題，請從 Preferences (偏好設定) 標籤的側邊導覽窗格中選擇 Themes (佈景主題)。
3. 若要變更您各個環境的佈景主題，請在 Themes (佈景主題) 窗格中視需要變更各項設定。若要使用程式碼變更佈景主題的若干部分，請選擇 your stylesheet (樣式表) 連結。
4. 若要將您的變更套用到任何環境，請開啟該環境。如果該環境已開啟，請重新整理該環境的 Web 瀏覽器標籤。

您可以變更的整體佈景主題設定

您可以從 Themes (佈景主題) 窗格的 Preferences (偏好設定) 標籤變更以下類型的整體佈景主題設定。

平面佈景主題

在 AWS Cloud9 IDE 中套用內建的平面佈景主題。

傳統佈景主題

在 AWS Cloud9 IDE 中套用所選的內建傳統佈景主題。

語法佈景主題

在 AWS Cloud9 IDE 中將所選佈景主題套用到程式碼檔案。

佈景主題覆寫

Important

AWS Cloud9 不再支援允許使用者藉由更新 `styles.css` 檔案來覆寫 IDE 佈景主題的功能。使用者可以繼續使用編輯器檢視、編輯和儲存 `styles.css` 檔案。但是，當 AWS Cloud9 IDE 載入時，不會套用任何佈景主題覆寫。

如果 AWS Cloud9 偵測到 `styles.css` 檔案遭到修改，IDE 會顯示下列訊息：

Support for theme overrides has been discontinued.(佈景主題覆寫已不再受支援。) The contents of this `styles.css` file will no longer be applied on loading the AWS Cloud9 IDE. (此 `styles.css` 檔案已無法再於 AWS Cloud9 IDE 載入時套用。)

如果您需要使用樣式表來定義 IDE 的佈景主題，請直接[聯絡我們](#)。

在 AWS Cloud9 整合開發環境 (IDE) 中管理初始化指令碼

Important

AWS Cloud9 不再支援允許使用者自訂初始化指令碼的實驗性功能。此指令碼在 IDE 中自動執行。使用者可以繼續使用編輯器檢視、編輯和儲存 `init.js` 檔案。但是，自訂初始化指令碼已無法再執行，也無法修改 IDE 的行為。

如果 AWS Cloud9 偵測到 `init.js` 檔案遭到修改，IDE 會顯示下列訊息：

Support for initialization scripts has been discontinued.(初始化指令碼已不再受支援。) The contents of this `init.js` file will no longer be executed on loading the AWS Cloud9 IDE. (此 `init.js` 檔案已無法再於 AWS Cloud9 IDE 載入時執行。)

如果您需要執行 IDE 的自訂初始化指令碼，請[聯絡我們](#)。

初始化指令碼定義了所有外掛程式載入後 IDE 會執行的自訂初始化程式碼。這個設定適用於每個與 IAM 使用者相關聯的 AWS Cloud9 開發環境。AWS Cloud9 也會持續掃描初始化指令碼的變更，並在發生修改時警示使用者。

開啟初始化指令碼

若要開啟您的初始化指令碼，請在選單列上選擇 AWS Cloud9、Open Your Init Script (開啟初始化指令碼)。

⚠ Important

您可以使用編輯器編輯和儲存 `init.js` 檔案，但您的自訂指令碼無法在 IDE 中執行。

AWS Cloud9 整合開發環境 (IDE) 的 MacOS 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 MacOS 作業系統的預設鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Default (預設)。
4. 針對 Operating System (作業系統)，選擇 MacOS。

另請參閱 [使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Command-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut

描述	按鍵繫結關係	命令
顯示程式碼完成內容選單	Control-Space Option-Space	complete
程式碼完成，然後覆寫	Control-Shift-Space Option-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Command-C	copy
將選取項目剪下到剪貼簿	Command-X	cut
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Command-F	find
在目前文件中選取所有符合尋找的項目	Control-Option-G	findAll
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Command-G	findnext
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Command-Shift-G	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Command-Shift-B	formatcode
顯示「移至行」方塊	Command-L	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace

描述	按鍵繫結關係	命令
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Command-Shift-U	lambdaUploadFunction
建立新檔案	Control-N	newfile
顯示 Preferences (偏好設定) 標籤	Command-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Command-Option-L	opentermhere
在游標處貼上剪貼簿目前的內容	Command-V	paste
針對修復錯誤顯示建議	Command-F3	quickfix
重做上一個動作	Command-Shift-Z Command-Y	redo
重新整理預覽窗格	Command-Enter	reloadpreview
啟動選取項目的重新命名重構	Option-Command-R	renameVar
顯示目前文件的尋找及取代之列，並以取代之為運算式為焦點	Option-Command-F	replace
重新執行初始化指令碼	Command-Enter	rerunInitScript
重新啟動環境	Command-R	restartc9
將目前的檔案重設為上次儲存的版本	Control-Shift-Q	reverttosaved

描述	按鍵繫結關係	命令
將每個開啟的檔案重設為其已儲存的版本	Option-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Command-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Command-Shift-S	saveas
顯示多個檔案的尋找及取代列	Shift-Command-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Command-Option-P	showprocesslist
復原上一個動作	Command-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Option-Control-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Option-Shift-W	closealltabs
關閉目前的窗格	Command-Control-W	closepane
關閉目前的標籤	Option-W	closetab
向下移一個窗格	Control-Command-Down	gotopanedown
向左移一個窗格	Control-Command-Left	gotopaneleft
向右移一個窗格	Control-Command-Right	gotopaneright
向上移一個窗格	Control-Command-Up	gottopaneup

描述	按鍵繫結關係	命令
向左移一個標籤	Command-[gototableft
向右移一個標籤	Command-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Command-Option-Shift-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Command-Option-Shift-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Command-Option-Shift-Right	movetabright
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Command-Option-Shift-Up	movetabup
移至下一個窗格	Option-Esc	nextpane
移至下一個標籤	Option-Tab	nexttab
移至上一個窗格	Option-Shift-Esc	previouspane
移至上一個標籤	Option-Shift-Tab	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Option-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Command-Shift-L	revealtab
移至第十個標籤	Command-0	tab0
移至第一個標籤	Command-1	tab1

描述	按鍵繫結關係	命令
移至第二個標籤	Command-2	tab2
移至第三個標籤	Command-3	tab3
移至第四個標籤	Command-4	tab4
移至第五個標籤	Command-5	tab5
移至第六個標籤	Command-6	tab6
移至第七個標籤	Command-7	tab7
移至第八個標籤	Command-8	tab8
移至第九個標籤	Command	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Command-E Command-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Command-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Command-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Command-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Command-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	Control-Esc	toggleconsole

描述	按鍵繫結關係	命令
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Command-U	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Control-Option-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Control-Option-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Control-Option-A	alignCursors
倒退一個空格	Control-Backspace Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Control-]	blockindent

描述	按鍵繫結關係	命令
將選取項目凸排一個定位點	Control-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce
將選取項目置中	Control-L	centerselection
複製行的內容，並在下一行貼上複製的內容	Command-Option-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Command-Option-Up	copylinesup
刪除一個空格	Delete Control-Delete Shift-Delete	del
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Command-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Command-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Control-Shift-M	expandToMatching
摺疊選取的程式碼，若選取的單位已摺疊，則予以展開	Command-Option-L Command-F1	fold
摺疊所有可摺疊的元素	Control-Command-Option-0	foldall
摺疊所有可摺疊的元素，但目前的選取範圍除外	Command-Option-0	fold0ther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup

描述	按鍵繫結關係	命令
移至檔案尾端	Command-End Command-Down	gotoend
向左移一個空格	Left Control-B	gotoleft
移至目前行的行尾	Command-Right End Control-E	gotolineend
移至目前行的行首	Command-Left Home Control-A	gotolinestart
移至下一個錯誤	F4	goToNextError
向下移一頁	Page Down Control-V	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Shift-F4	goToPreviousError
向右移一個空格	Right Control-F	gotoright
移至檔案開頭	Command-Home Command-Up	gotostart
向左移一個文字	Option-Left	gotowordleft
向右移一個文字	Option-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent
移至目前範圍中的相符符號	Control-P	jumptomatching
放大字型	Command+= Command-=	largerfont
將游標左側的數字減 1 (若其為數字)	Option-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Option-Shift-Up	modifyNumberUp

描述	按鍵繫結關係	命令
將選取項目向下移動一行	Option-Down	movelinesdown
將選取項目向上移動一行	Option-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若開啟則關閉	Insert	overwrite
向下移一頁	Option-Page Down	pagedown
向上移一頁	Option-Page Up	pageup
移除目前行	Command-D	removeline
從游標處刪除至目前行的行尾	Control-K	removetolineend
從目前行的行首刪除至游標處	Command-Backspace	removetolinestart
刪除游標左側的文字	Option-Backspace Control-Option-Backspace	removewordleft
刪除游標右側的文字	Option-Delete	removewordright
重新播放先前記錄的按鍵輸入	Command-Shift-E	replaymacro
選取所有可選取的內容	Command-A	selectall
將下一行以下的行納入選取項目	Shift-Down Control-Shift-N	selectdown
將左側的下一個空格納入選取項目	Shift-Left Control-Shift-B	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart

描述	按鍵繫結關係	命令
納入選取項目後的更多相符選取項目	Control-Option-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Control-Option-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Control-Option-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Control-Option-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Control-G	selectOrFindNext
選取或尋找上一個相符選取項目	Control-Shift-G	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright
將游標處以下至目前檔案結尾的內容納入選取項目	Command-Shift-End Command-Shift-Down	selectttoend
將游標處至目前行行尾的內容納入選取項目	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
將目前行行首到游標處的內容納入選取項目	Command-Shift-Left Control-Shift-A	selecttolinestart

描述	按鍵繫結關係	命令
將游標處至下一個相符符號的內容納入目前的範圍	Control-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Command-Shift-Home Command-Shift-Up	selecttostart
將下一行以上的內容納入選取項目	Shift-Up Control-Shift-Up	selectup
將游標左側的下一個文字納入選取項目	Option-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Option-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Command-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Command--	smallerfont
如果選取多行，則將其重新排序	Command-Option-S	sortlines
在目前行的行尾新增游標	Control-Option-L	splitIntoLines
將游標的內容移動到行尾，另起一行	Control-0	splitline
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Command-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Command-/	togglecomment

描述	按鍵繫結關係	命令
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Option-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Command-Option-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Control-W	toggleWordWrap
將選取項目全部變更為小寫	Control-Shift-U	toLowerCase
將選取項目全部變更為大寫	Control-U	toUpperCase
調換選取項目	Control-T	transposeLetters
展開選取的程式碼	Command-Option-Shift-L Command-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Command-Option-Shift-0	unfoldAll

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Command-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Control-Option-E	emmet_expand_abbreviation

描述	按鍵繫結關係	命令
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Command-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Command-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Control-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Option-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Option-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Command-B	build
恢復目前暫停的程序	F8 Command-\	resume
執行或除錯目前的應用程式	Option-F5	run
執行或除錯上次執行的檔案	F5	runlast

描述	按鍵繫結關係	命令
進入堆疊上的下一個函數	F11 Command-;	stepinto
離開目前的函數範圍	Shift-F11 Command-Shift-'	stepout
略過堆疊上目前的運算式	F10 Command-'	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Control-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 MacOS Vim 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 MacOS 作業系統的 Vim 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Vim。
4. 針對 Operating System (作業系統)，選擇 MacOS。

另請參閱[使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Command-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Control-Space Option-Space	complete
程式碼完成，然後覆寫	Control-Shift-Space Option-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Command-C	copy
將選取項目剪下到剪貼簿	Command-X	cut
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Command-F	find
在目前文件中選取所有符合尋找的項目	Control-Option-G	findAll
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Command-G	findnext
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Command-Shift-G	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree

描述	按鍵繫結關係	命令
重新格式化選取的 JavaScript 程式碼	Command-Shift-B	formatcode
顯示「移至行」方塊	Command-L	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Command-Shift-U	lambdaUploadFunction
建立新檔案	Control-N	newfile
顯示 Preferences (偏好設定) 標籤	Command-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Command-Option-L	opentermhere
在游標處貼上剪貼簿目前的內容	Command-V	paste
針對修復錯誤顯示建議	Command-F3	quickfix
重做上一個動作	Command-Shift-Z Command-Y	redo
重新整理預覽窗格	Command-Enter	reloadpreview
啟動選取項目的重新命名重構	Option-Command-R	renameVar

描述	按鍵繫結關係	命令
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Option-Command-F	replace
重新執行初始化指令碼	Command-Enter	rerunInitScript
重新啟動環境	Command-R	restartc9
將目前的檔案重設為上次儲存的版本	Control-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Option-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Command-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Command-Shift-S	saveas
顯示多個檔案的尋找及取代列	Shift-Command-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Command-Option-P	showprocesslist
復原上一個動作	Command-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Option-Control-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Option-Shift-W	closealltabs
關閉目前的窗格	Command-Control-W	closepane

描述	按鍵繫結關係	命令
關閉目前的標籤	Option-W	closetab
向下移一個窗格	Control-Command-Down	gotopanedown
向左移一個窗格	Control-Command-Left	gotopaneleft
向右移一個窗格	Control-Command-Right	gotopaneright
向上移一個窗格	Control-Command-Up	gottopaneup
向左移一個標籤	Command-[gototableft
向右移一個標籤	Command-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Command-Option-Shift-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Command-Option-Shift-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Command-Option-Shift-Right	movetabright
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Command-Option-Shift-Up	movetabup
移至下一個窗格	Option-Esc	nextpane
移至下一個標籤	Option-Tab	nexttab
移至上一個窗格	Option-Shift-Esc	previouspane
移至上一個標籤	Option-Shift-Tab	previoustab

描述	按鍵繫結關係	命令
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Option-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Command-Shift-L	revealtab
移至第十個標籤	Command-0	tab0
移至第一個標籤	Command-1	tab1
移至第二個標籤	Command-2	tab2
移至第三個標籤	Command-3	tab3
移至第四個標籤	Command-4	tab4
移至第五個標籤	Command-5	tab5
移至第六個標籤	Command-6	tab6
移至第七個標籤	Command-7	tab7
移至第八個標籤	Command-8	tab8
移至第九個標籤	Command	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Command-E Command-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Command-. F1	gotocommand

描述	按鍵繫結關係	命令
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Command-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Command-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Command-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	Control-Esc	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Command-U	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Control-Option-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Control-Option-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent

描述	按鍵繫結關係	命令
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Control-Option-A	alignCursors
倒退一個空格	Control-Backspace Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Control-]	blockindent
將選取項目凸排一個定位點	Control-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce
將選取項目置中	Control-L	centerselection
複製行的內容，並在下一行貼上複製的內容	Command-Option-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Command-Option-Up	copylinesup
刪除一個空格	Delete Control-Delete Shift-Delete	del
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Command-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Command-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Control-Shift-M	expandToMatching
摺疊選取的程式碼，若選取的單位已摺疊，則予以展開	Command-Option-L Command-F1	fold

描述	按鍵繫結關係	命令
摺疊所有可摺疊的元素	Control-Command-Option-0	foldall
摺疊所有可摺疊的元素，但目前的選取範圍除外	Command-Option-0	foldOther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
移至檔案尾端	Command-End Command-Down	gotoend
向左移一個空格	Left Control-B	gotoleft
移至目前行的行尾	Command-Right End Control-E	gotolineend
移至目前行的行首	Command-Left Home Control-A	gotolinestart
移至下一個錯誤	F4	goToNextError
向下移一頁	Page Down Control-V	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Shift-F4	goToPreviousError
向右移一個空格	Right Control-F	gotoright
移至檔案開頭	Command-Home Command-Up	gotostart
向左移一個文字	Option-Left	gotowordleft
向右移一個文字	Option-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent

描述	按鍵繫結關係	命令
移至目前範圍中的相符符號	Control-P	jumptomatching
放大字型	Command-+ Command-=	largerfont
將游標左側的數字減 1 (若其為數字)	Option-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Option-Shift-Up	modifyNumberUp
將選取項目向下移動一行	Option-Down	movelinesdown
將選取項目向上移動一行	Option-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若開啟則關閉	Insert	overwrite
向下移一頁	Option-Page Down	pagedown
向上移一頁	Option-Page Up	pageup
移除目前行	Command-D	removeline
從游標處刪除至目前行的行尾	Control-K	removetolineend
從目前行的行首刪除至游標處	Command-Backspace	removetolinestart
刪除游標左側的文字	Option-Backspace Control-Option-Backspace	removewordleft
刪除游標右側的文字	Option-Delete	removewordright
重新播放先前記錄的按鍵輸入	Command-Shift-E	replaymacro
選取所有可選取的內容	Command-A	selectall

描述	按鍵繫結關係	命令
將下一行以下的行納入選取項目	Shift-Down Control-Shift-N	selectdown
將左側的下一個空格納入選取項目	Shift-Left Control-Shift-B	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Control-Option-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Control-Option-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Control-Option-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Control-Option-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Control-G	selectOrFindNext
選取或尋找上一個相符選取項目	Control-Shift-G	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright

描述	按鍵繫結關係	命令
將游標處以下至目前檔案結尾的內容納入選取項目	Command-Shift-End Command-Shift-Down	selecttoend
將游標處至目前行行尾的內容納入選取項目	Command-Shift-Right Shift-End Control- Shift-E	selecttolineend
將目前行行首到游標處的內容納入選取項目	Command-Shift-Left Control-Shift-A	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Control-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Command-Shift-Home Command-Shift-Up	selecttostart
將下一行以上的內容納入選取項目	Shift-Up Control-S hift-P	selectup
將游標左側的下一個文字納入選取項目	Option-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Option-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Command-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Command--	smallerfont
如果選取多行，則將其重新排序	Command-Option-S	sortlines
在目前行的行尾新增游標	Control-Option-L	splitIntoLines

描述	按鍵繫結關係	命令
將游標的內容移動到行尾，另起一行	Control-0	splitline
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Command-Shift-/ Command-Shift-0	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Command-/ Command-0	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Option-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Command-Option-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Control-W	toggleWordWrap
將選取項目全部變更為小寫	Control-Shift-U	tolowercase
將選取項目全部變更為大寫	Control-U	touppercase
調換選取項目	Control-T	transposeletters
展開選取的程式碼	Command-Option-Shift-L Command-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Command-Option-Shift-0	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Command-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Control-Option-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Command-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Command-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Control-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Option-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Option-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Command-B	build
恢復目前暫停的程序	F8 Command-\	resume
執行或除錯目前的應用程式	Option-F5	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11 Command-;	stepinto
離開目前的函數範圍	Shift-F11 Command-Shift-'	stepout
略過堆疊上目前的運算式	F10 Command-'	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Control-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 MacOS Emacs 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 MacOS 作業系統的 Emacs 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Emacs。
4. 針對 Operating System (作業系統)，選擇 MacOS。

另請參閱 [使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Command-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Control-Space Option-Space	complete
完成程式碼，然後覆寫	Control-Shift-Space Option-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Command-C	copy
將選取項目剪下到剪貼簿	Command-X	cut
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Command-F	find
在目前文件中選取所有符合尋找的項目	Control-Option-G	findAll
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Command-G	findnext

描述	按鍵繫結關係	命令
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Command-Shift-G	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Command-Shift-B	formatcode
顯示「移至行」方塊	Command-L	gotoline
尋找及取代列已顯示時將其隱藏	Esc	hidesearchreplace
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Command-Shift-U	lambdaUploadFunction
建立新檔案	Control-N	newfile
顯示 Preferences (偏好設定) 標籤	Command-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Command-Option-L	opentermhere
在游標處貼上剪貼簿目前的內容	Command-V	paste

描述	按鍵繫結關係	命令
針對修復錯誤顯示建議	Command-F3	quickfix
重做上一個動作	Command-Shift-Z Command-Y	redo
重新整理預覽窗格	Command-Enter	reloadpreview
啟動選取項目的重新命名重構	Option-Command-R	renameVar
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Option-Command-F	replace
重新執行初始化指令碼	Command-Enter	rerunInitScript
重新啟動環境	Command-R	restartc9
將目前的檔案重設為上次儲存的版本	Control-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Option-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Command-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Command-Shift-S	saveas
顯示多個檔案的尋找及取代列	Shift-Command-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Command-Option-P	showprocesslist
復原上一個動作	Command-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Option-Control-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Option-Shift-W	closealltabs
關閉目前的窗格	Command-Control-W	closepane
關閉目前的標籤	Option-W	closetab
向下移一個窗格	Control-Command-Down	gotopanedown
向左移一個窗格	Control-Command-Left	gotopaneleft
向右移一個窗格	Control-Command-Right	gotopaneright
向上移一個窗格	Control-Command-Up	gottopaneup
向左移一個標籤	Command-[gototableft
向右移一個標籤	Command-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Command-Option-Shift-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Command-Option-Shift-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Command-Option-Shift-Right	movetabright

描述	按鍵繫結關係	命令
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Command-Option-Shift-Up	movetabup
移至下一個窗格	Option-Esc	nextpane
移至下一個標籤	Option-Tab	nexttab
移至上一個窗格	Option-Shift-Esc	previouspane
移至上一個標籤	Option-Shift-Tab	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Option-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Command-Shift-L	revealtab
移至第十個標籤	Command-0	tab0
移至第一個標籤	Command-1	tab1
移至第二個標籤	Command-2	tab2
移至第三個標籤	Command-3	tab3
移至第四個標籤	Command-4	tab4
移至第五個標籤	Command-5	tab5
移至第六個標籤	Command-6	tab6
移至第七個標籤	Command-7	tab7
移至第八個標籤	Command-8	tab8
移至第九個標籤	Command	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Command-E Command-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Command-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Command-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Command-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Command-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	Control-Esc	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Command-U	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Control-Option-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent

描述	按鍵繫結關係	命令
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Control-Option-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Control-Option-A	alignCursors
倒退一個空格	Control-Backspace Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Control-]	blockindent
將選取項目凸排一個定位點	Control-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce
將選取項目置中	Control-L	centerselection
複製行的內容，並在下一行貼上複製的內容	Command-Option-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Command-Option-Up	copylinesup
刪除一個空格	Delete Control-Delete Shift-Delete	del

描述	按鍵繫結關係	命令
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Command-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Command-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Control-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Command-Option-L Command-F1	fold
摺疊所有可摺疊的元素	Control-Command-Option-0	foldall
摺疊所有可摺疊的元素，但目前的選取範圍除外	Command-Option-0	fold0ther
向下移一行	Down Control-N	golinedown
向上移一行	Up Control-P	golineup
移至檔案尾端	Command-End Command-Down	gotoend
向左移一個空格	Left Control-B	gotoleft
移至目前行的行尾	Command-Right End Control-E	gotolineend
移至目前行的行首	Command-Left Home Control-A	gotolinestart
移至下一個錯誤	F4	goToNextError
向下移一頁	Page Down Control-V	gotopagedown
向上移一頁	Page Up	gotopageup

描述	按鍵繫結關係	命令
移至上一個錯誤	Shift-F4	goToPreviousError
向右移一個空格	Right Control-F	gotoright
移至檔案開頭	Command-Home Command-Up	gotostart
向左移一個文字	Option-Left	gotowordleft
向右移一個文字	Option-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent
移至目前範圍中的相符符號	Control-P	jumptomatching
放大字型	Command-+ Command-=	largerfont
將游標左側的數字減 1 (若其為數字)	Option-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Option-Shift-Up	modifyNumberUp
將選取項目向下移動一行	Option-Down	movelinesdown
將選取項目向上移動一行	Option-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
向下移一頁	Option-Page Down	pagedown
向上移一頁	Option-Page Up	pageup
移除目前行	Command-D	removeline
從游標處刪除至目前行的行尾	Control-K	removetolineend

描述	按鍵繫結關係	命令
從目前行的行首刪除至游標處	Command-Backspace	removetolinestart
刪除游標左側的文字	Option-Backspace Control-Option-Backspace	removewordleft
刪除游標右側的文字	Option-Delete	removewordright
重新播放先前記錄的按鍵輸入	Command-Shift-E	replaymacro
選取所有可選取的內容	Command-A	selectall
將下一行以下的行納入選取項目	Shift-Down Control-Shift-N	selectdown
將左側的下一個空格納入選取項目	Shift-Left Control-Shift-B	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Control-Option-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Control-Option-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Control-Option-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Control-Option-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Control-G	selectOrFindNext

描述	按鍵繫結關係	命令
選取或尋找上一個相符選取項目	Control-Shift-G	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright
將游標處以下至目前檔案結尾的內容納入選取項目	Command-Shift-End Command-Shift-Down	selecttoend
將游標處至目前行行尾的內容納入選取項目	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
將目前行行首到游標處的內容納入選取項目	Command-Shift-Left Control-Shift-A	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Control-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Command-Shift-Home Command-Shift-Up	selecttostart
將下一行以上的內容納入選取項目	Shift-Up Control-Shift-Up	selectup
將游標左側的下一個文字納入選取項目	Option-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Option-Shift-Right	selectwordright

描述	按鍵繫結關係	命令
顯示 Preferences (偏好設定) 標籤	Command-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Command--	smallerfont
如果選取多行，則將其重新排序	Command-Option-S	sortlines
在目前行的行尾新增游標	Control-Option-L	splitIntoLines
將游標的內容移動到行尾，另起一行	Control-0	splitline
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Command-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Command-/	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Option-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Command-Option-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Control-W	toggleWordWrap
將選取範圍內的字母變更為全部小寫	Control-Shift-U	tolowercase

描述	按鍵繫結關係	命令
將選取範圍內的字母變更為全部大寫	Control-U	touppercase
將選取範圍內的字母大小寫對調	Control-T	transposeletters
展開選取的程式碼	Command-Option-Shift-L Command-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Command-Option-Shift-0	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Command-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Control-Option-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Command-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Command-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Control-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Option-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Option-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Command-B	build
恢復目前暫停的程序	F8 Command-\	resume
執行或除錯目前的應用程式	Option-F5	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11 Command-;	stepinto
離開目前的函數範圍	Shift-F11 Command-Shift-'	stepout
略過堆疊上目前的運算式	F10 Command-'	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Control-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 MacOS Sublime 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 MacOS 作業系統的 Sublime 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，請選擇 Sublime。
4. 針對 Operating System (作業系統)，選擇 MacOS。

另請參閱 [使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Command-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Control-Space Option-Space	complete

描述	按鍵繫結關係	命令
程式碼完成，然後覆寫	Control-Shift-Space Option-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Command-C	copy
將選取項目剪下到剪貼簿	Command-X	cut
從游標處刪除至所在該行的開頭	Command-K Command-B ackspace Command-B ackspace	delete_to_hard_bol
從游標處刪除至所在該行的結尾	Command-K Command-K Command-Delete Control-K	delete_to_hard_eol
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Command-F	find
反白所有符合的選項	Control-Command-G	find_all_under
反白下一個符合的選項	Option-Command-G	find_under
反白游標周圍區域以及所有符合反白區域的項目	Command-D	find_under_expand
反白游標周圍區域並為所有符合反白區域的項目加上邊框	Command-K Command-D	find_under_expand_skip
反白前一項符合選取範圍的內容	Shift-Option-Command-G	find_under_previous
在目前文件中選取所有符合尋找的項目	Control-Option-G	findAll
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Command-G	findnext

描述	按鍵繫結關係	命令
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Shift-Command-G	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Control-Option-F	formatcode
顯示「移至行」方塊	Control-G	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F12 Command-Option-Down	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Command-Shift-U	lambdaUploadFunction
移至目前字詞的結尾。	Option-Right	moveToWordEndRight
移至目前字詞的開頭。	Option-Left	moveToWordStartLeft
建立新檔案	Control-N	newfile
顯示 Preferences (偏好設定) 標籤	Command-,	openpreferences

描述	按鍵繫結關係	命令
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Command-Option-L	opentermhere
在游標處貼上剪貼簿目前的內容	Command-V	paste
針對修復錯誤顯示建議	Command-F3	quickfix
重做上一個動作	Command-Shift-Z Command-Y	redo
重新整理預覽窗格	Command-Enter	reloadpreview
啟動選取項目的重新命名重構	Option-Command-R	renameVar
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Command-Option-F	replace
使用尋找及取代列中的「取代為」運算式來取代所有尋找運算式符合項目	Control-Option-Enter	replaceall
使用尋找及取代列中的「取代為」運算式來取代下一個尋找運算式符合項目	Command-Option-E	replacenext
重新執行初始化指令碼	Command-Enter	rerunInitScript
重新啟動環境	Command-R	restartc9
將目前的檔案重設為上次儲存的版本	Control-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Option-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Command-S	save

描述	按鍵繫結關係	命令
使用其他檔案名稱將目前的檔案儲存至磁碟	Command-Shift-S	saveas
顯示多個檔案的尋找及取代列	Command-Shift-F	searchinfiles
將從游標開始至目前字詞結尾的內容納入選取項目	Option-Shift-Right	selectToWordEndRight
將從游標開始至目前字詞開頭的內容納入選取項目	Option-Shift-Left	selectToWordStartLeft
顯示 Process List (處理序清單) 對話方塊	Command-Option-P	showprocesslist
復原上一個動作	Command-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Option-Control-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Option-Shift-W	closealltabs
關閉目前的窗格	Command-Control-W	closepane
關閉目前的標籤	Option-W	closetab
向下移一個窗格	Control-Command-Down	gotopanedown
向左移一個窗格	Control-Command-Left	gotopaneleft
向右移一個窗格	Control-Command-Right	gotopaneright

描述	按鍵繫結關係	命令
向上移一個窗格	Control-Command-Up	gottopaneup
向左移一個標籤	Command-Shift-[Command-Option-Left	gototableft
向右移一個標籤	Command-Shift-] Command-Option-Right	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Command-Option-Shift-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Command-Option-Shift-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Command-Option-Shift-Right	movetabright
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Command-Option-Shift-Up	movetabup
移至下一個標籤	Control-Tab	nexttab
移至上一個窗格	Option-Shift-Esc	previouspane
移至上一個標籤	Control-Shift-Tab	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Command-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Command-E	revealtab
移至第十個標籤	Command-0	tab0

描述	按鍵繫結關係	命令
移至第一個標籤	Command-1	tab1
移至第二個標籤	Command-2	tab2
移至第三個標籤	Command-3	tab3
移至第四個標籤	Command-4	tab4
移至第五個標籤	Command-5	tab5
移至第六個標籤	Command-6	tab6
移至第七個標籤	Command-7	tab7
移至第八個標籤	Command-8	tab8
移至第九個標籤	Command	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Command-E Command-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Command-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Command-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Command-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Command-Shift-R	outline

描述	按鍵繫結關係	命令
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	Control-`	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Command-K Command-B	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Control-Shift-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Control-Option-Shift-Up	addCursorAboveSkipCurrent
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Control-Shift-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Control-Option-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Control-Option-A	alignCursors
倒退一個空格	Control-Backspace Shift-Backspace Backspace	backspace

描述	按鍵繫結關係	命令
將選取項目縮排一個定位點	Control-]	blockindent
將選取項目凸排一個定位點	Control-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Command-Z Command-Shift-Z Command-S Command-Y	cancelBrowserUndoInAce
將選取項目置中	Command-K Command-C Control-L	centerselection
複製行的內容，並在下一行貼上複製的內容	Command-Option-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Command-Option-Up	copylinesup
刪除一個空格	Delete Control-Delete Shift-Delete	del
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Command-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Command-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Control-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Command-Option-L Command-F1	fold
摺疊所有可摺疊的元素	Control-Command-Option-0	foldall
摺疊所有可摺疊的元素，但目前的選取範圍除外	Command-K Command-1	foldOther
向下移一行	Down Control-N	golinedown

描述	按鍵繫結關係	命令
向上移一行	Up Control-P	golineup
移至檔案尾端	Command-End Command-Down	gotoend
向左移一個空格	Left Control-B	gotoleft
移至目前行的行尾	Command-Right End Control-E	gotolineend
移至目前行的行首	Command-Left Home Control-A	gotolinestart
移至下一個錯誤	Control-F6	goToNextError
向下移一頁	Page Down Control-V	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Control-Shift-F6	goToPreviousError
向右移一個空格	Right Control-F	gotoright
移至檔案開頭	Command-Home Command-Up	gotostart
向左移一個文字	Option-Left	gotowordleft
向右移一個文字	Option-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent
將所選各行合併為一行	Command-J	joinlines
移至目前範圍中的相符符號	Control-M	jumptomatching
放大字型	Command-= Command-+	largerfont

描述	按鍵繫結關係	命令
將游標左側的數字減 1 (若其為數字)	Option-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Option-Up	modifyNumberUp
將選取項目向下移動一行	Control-Command-Down	movelinesdown
將選取項目向上移動一行	Control-Command-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
向下移一頁	Option-Page Down	pagedown
向上移一頁	Option-Page Up	pageup
刪除目前行的內容	Control-Shift-K	removeline
從游標處刪除至目前行的行尾	Control-K	removetolineend
從目前行的行首刪除至游標處	Command-Backspace	removetolinestart
刪除游標左側的文字	Option-Backspace Control-Option-Backspace	removewordleft
刪除游標右側的文字	Option-Delete	removewordright
重新播放先前記錄的按鍵輸入	Control-Shift-Q	replaymacro
選取所有可選取的內容	Command-A	selectall
將下一行以下的行納入選取項目	Shift-Down Control-Shift-N	selectdown

描述	按鍵繫結關係	命令
將左側的下一個空格納入選取項目	Shift-Left Control-Shift-B	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
納入從目前該行的開頭至游標處的內容做為選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Control-Option-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Control-Option-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Control-Option-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Control-Option-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Control-G	selectOrFindNext
選取或尋找上一個相符選取項目	Control-Shift-G	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright
將游標處以下至目前檔案結尾的內容納入選取項目	Command-Shift-End Command-Shift-Down	selecttoend

描述	按鍵繫結關係	命令
將游標處至目前行行尾的內容納入選取項目	Command-Shift-Right Shift-End Control-Shift-E	selecttolineend
將目前行行首到游標處的內容納入選取項目	Command-Shift-Left Control-Shift-A	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Control-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Command-Shift-Home Command-Shift-Up	selecttostart
將下一行以上的內容納入選取項目	Shift-Up Control-Shift-P	selectup
將游標左側的下一個文字納入選取項目	Option-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Option-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Command-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Command--	smallerfont
如果選取多行，則將其重新排序	F5	sortlines
在目前行的行尾新增游標	Command-Shift-L	splitIntoLines
將游標的內容移動到行尾，另起一行	Control-0	splitline

描述	按鍵繫結關係	命令
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Command-Option- /	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Command- /	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	Command-Option- [toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Option-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Control-Q	toggleRecording
將文字換行，若已在將文字換行則停止換行	Control-W	toggleWordWrap
將選取項目全部變更為小寫	Command-K Command-L	tolowercase
將選取項目全部變更為大寫	Command-K Command-U	touppercase
調換選取項目	Control-T	transposeletters
展開選取的程式碼	Command-Option-]	unfold
展開整個檔案的程式碼摺疊	Command-K Command-O Command-K Command-J	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Command-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Control-Option-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Command-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Command-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Control-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Option-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Option-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	F7 Command-B	build
恢復目前暫停的程序	F8 Command-\	resume
執行或除錯目前的應用程式	Command-Shift-B	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11 Command-;	stepinto
離開目前的函數範圍	Shift-F11 Command-Shift-'	stepout
略過堆疊上目前的運算式	F10 Command-'	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Control-Break	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 Windows / Linux 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 Windows / Linux 作業系統的預設鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Default (預設)。
4. 針對 Operating System (作業系統)，請選擇 Windows / Linux。

另請參閱 [使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Ctrl-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Ctrl-Space Alt-Space	complete
程式碼完成，然後覆寫	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Ctrl-C	copy
將選取項目剪下到剪貼簿	Ctrl-X	cut
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Ctrl-F	find
在目前文件中選取所有符合尋找的項目	Ctrl-Alt-K	findall
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Ctrl-K	findnext

描述	按鍵繫結關係	命令
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Ctrl-Shift-K	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Ctrl-Shift-B	formatcode
顯示「移至行」方塊	Ctrl-G	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Ctrl-Shift-U	lambdaUploadFunction
建立新檔案	Alt-N	newfile
顯示 Preferences (偏好設定) 標籤	Ctrl-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Alt-L	opentermhere
在游標處貼上剪貼簿目前的內容	Ctrl-V	paste

描述	按鍵繫結關係	命令
針對修復錯誤顯示建議	Ctrl-F3	quickfix
重做上一個動作	Ctrl-Shift-Z Ctrl-Y	redo
重新整理預覽窗格	Ctrl-Enter	reloadpreview
啟動選取項目的重新命名重構	Ctrl-Alt-R	renameVar
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Alt-Shift-F Ctrl-H	replace
重新執行初始化指令碼	Ctrl-Enter	rerunInitScript
重新啟動環境	Ctrl-R	restartc9
將目前的檔案重設為上次儲存的版本	Ctrl-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Alt-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Ctrl-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Ctrl-Shift-S	saveas
顯示多個檔案的尋找及取代列	Ctrl-Shift-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Ctrl-Alt-P	showprocesslist
復原上一個動作	Ctrl-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Ctrl-Alt-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Alt-Shift-W	closealltabs
關閉目前的窗格	Ctrl-W	closepane
關閉目前的標籤	Alt-W	closetab
向下移一個窗格	Ctrl-Meta-Down	gotopanedown
向左移一個窗格	Ctrl-Meta-Left	gotopaneleft
向右移一個窗格	Ctrl-Meta-Right	gotopaneright
向上移一個窗格	Ctrl-Meta-Up	gottopaneup
向左移一個標籤	Ctrl-[gototableft
向右移一個標籤	Ctrl-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Ctrl-Meta-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Ctrl-Meta-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Ctrl-Meta-Right	movetabright

描述	按鍵繫結關係	命令
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Ctrl-Meta-Up	movetabup
移至下一個窗格	Ctrl-`	nextpane
移至下一個標籤	Ctrl-Tab Alt-`	nexttab
移至上一個窗格	Ctrl-Shift-`	previouspane
移至上一個標籤	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Alt-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Ctrl-Shift-L	revealtab
移至第十個標籤	Ctrl-0	tab0
移至第一個標籤	Ctrl-1	tab1
移至第二個標籤	Ctrl-2	tab2
移至第三個標籤	Ctrl-3	tab3
移至第四個標籤	Ctrl-4	tab4
移至第五個標籤	Ctrl-5	tab5
移至第六個標籤	Ctrl-6	tab6
移至第七個標籤	Ctrl-7	tab7
移至第八個標籤	Ctrl-8	tab8
移至第九個標籤	Ctrl-9	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Ctrl-E Ctrl-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Ctrl-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Ctrl-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Ctrl-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Ctrl-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	F6	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Ctrl-I	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Ctrl-Alt-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	按鍵繫結關係	命令
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Ctrl-Alt-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Ctrl-Alt-A	alignCursors
倒退一個空格	Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Ctrl-]	blockindent
將選取項目凸排一個定位點	Ctrl-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
將選取項目置中	Ctrl-L	centerselection
複製行的內容，並在下一行貼上複製的內容	Alt-Shift-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Alt-Shift-Up	copylinesup
剪下選取項目，或如果沒有選取項目，請刪除一個空格	Shift-Delete	cut_or_delete
刪除一個空格	Delete	del

描述	按鍵繫結關係	命令
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Ctrl-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Ctrl-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Ctrl-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Alt-L Ctrl-F1	fold
摺疊所有可摺疊的元素	Ctrl-Command-Option-0	foldall
摺疊所有可摺疊的元素，但目前的選取範圍除外	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
移至檔案尾端	Ctrl-End	gotoend
向左移一個空格	Left	gotoleft
移至目前行的行尾	Alt-Right End	gotolineend
移至目前行的行首	Alt-Left Home	gotolinestart
移至下一個錯誤	Alt-E	goToNextError
向下移一頁	Page Down	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Alt-Shift-E	goToPreviousError
向右移一個空格	Right	gotoright

描述	按鍵繫結關係	命令
移至檔案開頭	Ctrl-Home	gotostart
向左移一個文字	Ctrl-Left	gotowordleft
向右移一個文字	Ctrl-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent
移至目前範圍中的相符符號	Ctrl-P	jumptomatching
放大字型	Ctrl-+ Ctrl-=	largerfont
將游標左側的數字減 1 (若其為數字)	Ctrl-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Ctrl-Shift-Up	modifyNumberUp
將選取項目向下移動一行	Alt-Down	movelinesdown
將選取項目向上移動一行	Alt-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
向下移一頁	Option-Page Down	pagedown
向上移一頁	Option-Page Up	pageup
刪除目前行的內容	Ctrl-D	removeline
從游標處刪除至目前行的行尾	Alt-Delete	removetolineend
從目前行的行首刪除至游標處	Alt-Backspace	removetolinestart
刪除游標左側的文字	Ctrl-Backspace	removewordleft

描述	按鍵繫結關係	命令
刪除游標右側的文字	Ctrl-Delete	removewordright
重新播放先前記錄的按鍵輸入	Ctrl-Shift-E	replaymacro
將目前檔案向下捲動一行	Ctrl-Down	scrolldown
將目前檔案向上捲動一行	Ctrl-Up	scrollup
選取所有可選取的內容	Ctrl-A	selectall
將下一行以下的行納入選取項目	Shift-Down	selectdown
將左側的下一個空格納入選取項目	Shift-Left	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Ctrl-Alt-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Ctrl-Alt-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Ctrl-Alt-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Ctrl-Alt-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Alt-K	selectOrFindNext

描述	按鍵繫結關係	命令
選取或尋找上一個相符選取項目	Alt-Shift-K	<code>selectOrFindPrevious</code>
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	<code>selectpagedown</code>
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	<code>selectpageup</code>
將游標右側的下一個空格納入選取項目	Shift-Right	<code>selectright</code>
將游標處以下至目前檔案結尾的內容納入選取項目	Ctrl-Shift-End	<code>selecttoend</code>
將游標處至目前行行尾的內容納入選取項目	Alt-Shift-Right	<code>selecttolineend</code>
將目前行行首到游標處的內容納入選取項目	Alt-Shift-Left	<code>selecttolinestart</code>
將游標處至下一個相符符號的內容納入目前的範圍	Ctrl-Shift-P	<code>selecttomatching</code>
將游標處以上至目前檔案開頭的內容納入選取項目	Ctrl-Shift-Home	<code>selecttostart</code>
將下一行以上的內容納入選取項目	Shift-Up	<code>selectup</code>
將游標左側的下一個文字納入選取項目	Ctrl-Shift-Left	<code>selectwordleft</code>
將游標右側的下一個文字納入選取項目	Ctrl-Shift-Right	<code>selectwordright</code>
顯示 Preferences (偏好設定) 標籤	Ctrl-,	<code>showSettingsMenu</code>

描述	按鍵繫結關係	命令
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Ctrl--	smallerfont
如果選取多行，則將其重新排序	Ctrl-Alt-S	sortlines
在目前行的行尾新增游標	Ctrl-Alt-L	splitIntoLines
將游標的內容移動到行尾，另起一行	Ctrl-0	splitline
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Ctrl-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Ctrl-/	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Alt-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Ctrl-Alt-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Ctrl-Q	toggleWordWrap
將選取項目全部變更為小寫	Ctrl-Shift-U	tolowercase
將選取項目全部變更為大寫	Ctrl-U	touppercase
調換選取項目	Alt-X	transposeletters

描述	按鍵繫結關係	命令
展開選取的程式碼	Alt-Shift-L Ctrl-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Alt-Shift-0	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Ctrl-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Ctrl-Alt-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Ctrl-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Ctrl-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Ctrl-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Alt-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Alt-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Ctrl-B	build
恢復目前暫停的程序	F8	resume
執行或除錯目前的應用程式	Alt-F5	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11	stepinto
離開目前的函數範圍	Shift-F11	stepout
略過堆疊上目前的運算式	F10	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Ctrl-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 Windows / Linux Vim 預設按鍵繫結關係參考

下列是 AWS Cloud9 IDE 中 Windows/Linux 作業系統的 Vim 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Vim。
4. 針對 Operating System (作業系統)，請選擇 Windows / Linux。

另請參閱[使用按鍵繫結關係](#)

- [一般](#)
- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Ctrl-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Ctrl-Space Alt-Space	complete
程式碼完成，然後覆寫	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Ctrl-C	copy
將選取項目剪下到剪貼簿	Ctrl-X	cut
展開程式碼，如適用	Tab	expandSnippet

描述	按鍵繫結關係	命令
顯示目前文件的尋找及取代列	Ctrl-F	find
在目前文件中選取所有符合尋找的項目	Ctrl-Alt-K	findall
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Ctrl-K	findnext
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Ctrl-Shift-K	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Ctrl-Shift-B	formatcode
顯示「移至行」方塊	Ctrl-G	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Ctrl-Shift-U	lambdaUploadFunction
建立新檔案	Alt-N	newfile
顯示 Preferences (偏好設定) 標籤	Ctrl-,	openpreferences

描述	按鍵繫結關係	命令
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Alt-L	opentermhere
在游標處貼上剪貼簿目前的內容	Ctrl-V	paste
針對修復錯誤顯示建議	Ctrl-F3	quickfix
重做上一個動作	Ctrl-Shift-Z Ctrl-Y	redo
重新整理預覽窗格	Ctrl-Enter	reloadpreview
啟動選取項目的重新命名重構	Ctrl-Alt-R	renameVar
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Alt-Shift-F Ctrl-H	replace
重新執行初始化指令碼	Ctrl-Enter	rerunInitScript
重新啟動環境	Ctrl-R	restartc9
將目前的檔案重設為上次儲存的版本	Ctrl-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Alt-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Ctrl-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Ctrl-Shift-S	saveas
顯示多個檔案的尋找及取代列	Ctrl-Shift-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Ctrl-Alt-P	showprocesslist
復原上一個動作	Ctrl-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Ctrl-Alt-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Alt-Shift-W	closealltabs
關閉目前的窗格	Ctrl-W	closepane
關閉目前的標籤	Alt-W	closetab
向下移一個窗格	Ctrl-Meta-Down	gotopanedown
向左移一個窗格	Ctrl-Meta-Left	gotopaneleft
向右移一個窗格	Ctrl-Meta-Right	gotopaneright
向上移一個窗格	Ctrl-Meta-Up	gottopaneup
向左移一個標籤	Ctrl-[gototableft
向右移一個標籤	Ctrl-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Ctrl-Meta-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Ctrl-Meta-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Ctrl-Meta-Right	movetabright

描述	按鍵繫結關係	命令
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Ctrl-Meta-Up	movetabup
移至下一個窗格	Ctrl-`	nextpane
移至下一個標籤	Ctrl-Tab Alt-`	nexttab
移至上一個窗格	Ctrl-Shift-`	previouspane
移至上一個標籤	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Alt-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Ctrl-Shift-L	revealtab
移至第十個標籤	Ctrl-0	tab0
移至第一個標籤	Ctrl-1	tab1
移至第二個標籤	Ctrl-2	tab2
移至第三個標籤	Ctrl-3	tab3
移至第四個標籤	Ctrl-4	tab4
移至第五個標籤	Ctrl-5	tab5
移至第六個標籤	Ctrl-6	tab6
移至第七個標籤	Ctrl-7	tab7
移至第八個標籤	Ctrl-8	tab8
移至第九個標籤	Ctrl-9	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Ctrl-E Ctrl-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Ctrl-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Ctrl-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Ctrl-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Ctrl-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	F6	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Ctrl-I	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Ctrl-Alt-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	按鍵繫結關係	命令
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Ctrl-Alt-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Ctrl-Alt-A	alignCursors
倒退一個空格	Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Ctrl-]	blockindent
將選取項目凸排一個定位點	Ctrl-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
複製行的內容，並在下一行貼上複製的內容	Alt-Shift-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Alt-Shift-Up	copylinesup
剪下選取項目。如果沒有選取項目，請刪除一個空格	Shift-Delete	cut_or_delete
刪除一個空格	Delete	del
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Ctrl-Shift-D	duplicateSelection

描述	按鍵繫結關係	命令
將目前行的內容納入選取項目	Ctrl-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Ctrl-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Alt-L Ctrl-F1	fold
摺疊所有可摺疊的元素，但目前的選取範圍除外	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
移至檔案尾端	Ctrl-End	gotoend
向左移一個空格	Left	gotoleft
移至目前行的行尾	Alt-Right End	gotolineend
移至目前行的行首	Alt-Left Home	gotolinestart
移至下一個錯誤	Alt-E	goToNextError
向下移一頁	Page Down	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Alt-Shift-E	goToPreviousError
向右移一個空格	Right	gotoright
移至檔案開頭	Ctrl-Home	gotostart
向左移一個文字	Ctrl-Left	gotowordleft
向右移一個文字	Ctrl-Right	gotowordright

描述	按鍵繫結關係	命令
將選取項目縮排一個定位點	Tab	indent
移至目前範圍中的相符符號	Ctrl-P	jumptomatching
放大字型	Ctrl-+ Ctrl-=	largerfont
將游標左側的數字減 1 (若其為數字)	Ctrl-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Ctrl-Shift-Up	modifyNumberUp
將選取項目向下移動一行	Alt-Down	movelinesdown
將選取項目向上移動一行	Alt-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
刪除目前行的內容	Ctrl-D	removeline
從游標處刪除至目前行的行尾	Alt-Delete	removetolineend
從目前行的行首刪除至游標處	Alt-Backspace	removetolinestart
刪除游標左側的文字	Ctrl-Backspace	removewordleft
刪除游標右側的文字	Ctrl-Delete	removewordright
重新播放先前記錄的按鍵輸入	Ctrl-Shift-E	replaymacro
將目前檔案向下捲動一行	Ctrl-Down	scrolldown
將目前檔案向上捲動一行	Ctrl-Up	scrollup
選取所有可選取的內容	Ctrl-A	selectall

描述	按鍵繫結關係	命令
將下一行以下的行納入選取項目	Shift-Down	selectdown
將左側的下一個空格納入選取項目	Shift-Left	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Ctrl-Alt-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Ctrl-Alt-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Ctrl-Alt-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Ctrl-Alt-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Alt-K	selectOrFindNext
選取或尋找上一個相符選取項目	Alt-Shift-K	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright

描述	按鍵繫結關係	命令
將游標處以下至目前檔案結尾的內容納入選取項目	Ctrl-Shift-End	selecttoend
將游標處至目前行行尾的內容納入選取項目	Alt-Shift-Right	selecttolineend
將目前行行首到游標處的內容納入選取項目	Alt-Shift-Left	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Ctrl-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Ctrl-Shift-Home	selecttostart
將下一行以上的內容納入選取項目	Shift-Up	selectup
將游標左側的下一個文字納入選取項目	Ctrl-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Ctrl-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Ctrl-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Ctrl--	smallerfont
如果選取多行，則將其重新排序	Ctrl-Alt-S	sortlines
在目前行的行尾新增游標	Ctrl-Alt-L	splitIntoLines

描述	按鍵繫結關係	命令
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Ctrl-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Ctrl-/	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Alt-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Ctrl-Alt-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Ctrl-Q	toggleWordWrap
將選取項目全部變更為小寫	Ctrl-Shift-U	tolowercase
將選取項目全部變更為大寫	Ctrl-U	touppercase
調換選取項目	Alt-X	transposeletters
展開選取的程式碼	Alt-Shift-L Ctrl-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Alt-Shift-0	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Ctrl-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Ctrl-Alt-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Ctrl-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Ctrl-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Ctrl-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Alt-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Alt-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Ctrl-B	build
恢復目前暫停的程序	F8	resume
執行或除錯目前的應用程式	Alt-F5	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11	stepinto
離開目前的函數範圍	Shift-F11	stepout
略過堆疊上目前的運算式	F10	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Ctrl-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 Windows / Linux Emacs 預設按鍵繫結關係參考

下列是 AWS Cloud9 IDE 中 Windows / Linux 作業系統的 Emacs 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，選擇 Emacs。
4. 針對 Operating System (作業系統)，請選擇 Windows / Linux。

另請參閱 [使用按鍵繫結關係](#)

- [一般](#)

- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Ctrl-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Ctrl-Space Alt-Space	complete
程式碼完成，然後覆寫	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Ctrl-C	copy
將選取項目剪下到剪貼簿	Ctrl-X	cut
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Ctrl-F	find
在目前文件中選取所有符合尋找的項目	Ctrl-Alt-K	findall
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	Ctrl-K	findnext
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Ctrl-Shift-K	findprevious

描述	按鍵繫結關係	命令
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Ctrl-Shift-B	formatcode
顯示「移至行」方塊	Ctrl-G	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F3	jumptodef
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Ctrl-Shift-U	lambdaUploadFunction
建立新檔案	Alt-N	newfile
顯示 Preferences (偏好設定) 標籤	Ctrl-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Alt-L	opentermhere
在游標處貼上剪貼簿目前的內容	Ctrl-V	paste
針對修復錯誤顯示建議	Ctrl-F3	quickfix
重做上一個動作	Ctrl-Shift-Z Ctrl-Y	redo

描述	按鍵繫結關係	命令
重新整理預覽窗格	Ctrl-Enter	reloadpreview
啟動選取項目的重新命名重構	Ctrl-Alt-R	renameVar
顯示目前文件的尋找及取代列，並以取代為運算式為焦點	Alt-Shift-F Ctrl-H	replace
重新執行初始化指令碼	Ctrl-Enter	rerunInitScript
重新啟動環境	Ctrl-R	restartc9
將目前的檔案重設為上次儲存的版本	Ctrl-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Alt-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Ctrl-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Ctrl-Shift-S	saveas
顯示多個檔案的尋找及取代列	Ctrl-Shift-F	searchinfiles
顯示 Process List (處理序清單) 對話方塊	Ctrl-Alt-P	showprocesslist
復原上一個動作	Ctrl-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Ctrl-Alt-W	closeallbutme

描述	按鍵繫結關係	命令
關閉所有窗格中所有開啟的標籤	Alt-Shift-W	closealltabs
關閉目前的窗格	Ctrl-W	closepane
關閉目前的標籤	Alt-W	closetab
向下移一個窗格	Ctrl-Meta-Down	gotopanedown
向左移一個窗格	Ctrl-Meta-Left	gotopaneleft
向右移一個窗格	Ctrl-Meta-Right	gotopaneright
向上移一個窗格	Ctrl-Meta-Up	gottopaneup
向左移一個標籤	Ctrl-[gototableft
向右移一個標籤	Ctrl-]	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Ctrl-Meta-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Ctrl-Meta-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Ctrl-Meta-Right	movetabright
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Ctrl-Meta-Up	movetabup
移至下一個窗格	Ctrl-`	nextpane
移至下一個標籤	Ctrl-Tab Alt-`	nexttab

描述	按鍵繫結關係	命令
移至上一個窗格	Ctrl-Shift-`	previouspane
移至上一個標籤	Ctrl-Shift-Tab Alt-Shift-`	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Alt-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Ctrl-Shift-L	revealtab
移至第十個標籤	Ctrl-0	tab0
移至第一個標籤	Ctrl-1	tab1
移至第二個標籤	Ctrl-2	tab2
移至第三個標籤	Ctrl-3	tab3
移至第四個標籤	Ctrl-4	tab4
移至第五個標籤	Ctrl-5	tab5
移至第六個標籤	Ctrl-6	tab6
移至第七個標籤	Ctrl-7	tab7
移至第八個標籤	Ctrl-8	tab8
移至第九個標籤	Ctrl-9	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Ctrl-E Ctrl-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Ctrl-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Ctrl-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Ctrl-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Ctrl-Shift-E	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	F6	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Ctrl-I	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Ctrl-Alt-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Ctrl-Alt-Shift-Up	addCursorAboveSkipCurrent

描述	按鍵繫結關係	命令
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Ctrl-Alt-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Ctrl-Alt-A	alignCursors
倒退一個空格	Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Ctrl-]	blockindent
將選取項目凸排一個定位點	Ctrl-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
複製行的內容，並在下一行貼上複製的內容	Alt-Shift-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Alt-Shift-Up	copylinesup
剪下選取項目，或如果沒有選取項目，請刪除一個空格	Shift-Delete	cut_or_delete
刪除一個空格	Delete	del
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Ctrl-Shift-D	duplicateSelection

描述	按鍵繫結關係	命令
將目前行的內容納入選取項目	Ctrl-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Ctrl-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Alt-L Ctrl-F1	fold
摺疊所有可摺疊的元素，但目前的選取範圍除外	Alt-0	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
移至檔案尾端	Ctrl-End	gotoend
向左移一個空格	Left	gotoleft
移至目前行的行尾	Alt-Right End	gotolineend
移至目前行的行首	Alt-Left Home	gotolinestart
移至下一個錯誤	Alt-E	goToNextError
向下移一頁	Page Down	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Alt-Shift-E	goToPreviousError
向右移一個空格	Right	gotoright
移至檔案開頭	Ctrl-Home	gotostart
向左移一個文字	Ctrl-Left	gotowordleft
向右移一個文字	Ctrl-Right	gotowordright

描述	按鍵繫結關係	命令
將選取項目縮排一個定位點	Tab	indent
移至目前範圍中的相符符號	Ctrl-P	jumptomatching
放大字型	Ctrl-+ Ctrl-=	largerfont
將游標左側的數字減 1 (若其為數字)	Ctrl-Shift-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Ctrl-Shift-Up	modifyNumberUp
將選取項目向下移動一行	Alt-Down	movelinesdown
將選取項目向上移動一行	Alt-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
刪除目前行的內容	Ctrl-D	removeline
從游標處刪除至目前行的行尾	Alt-Delete	removetolineend
從目前行的行首刪除至游標處	Alt-Backspace	removetolinestart
刪除游標左側的文字	Ctrl-Backspace	removewordleft
刪除游標右側的文字	Ctrl-Delete	removewordright
重新播放先前記錄的按鍵輸入	Ctrl-Shift-E	replaymacro
將目前檔案向下捲動一行	Ctrl-Down	scrolldown
將目前檔案向上捲動一行	Ctrl-Up	scrollup
選取所有可選取的內容	Ctrl-A	selectall

描述	按鍵繫結關係	命令
將下一行以下的行納入選取項目	Shift-Down	selectdown
將左側的下一個空格納入選取項目	Shift-Left	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Ctrl-Alt-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Ctrl-Alt-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Ctrl-Alt-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Ctrl-Alt-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Alt-K	selectOrFindNext
選取或尋找上一個相符選取項目	Alt-Shift-K	selectOrFindPrevious
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright

描述	按鍵繫結關係	命令
將游標處以下至目前檔案結尾的內容納入選取項目	Ctrl-Shift-End	selecttoend
將游標處至目前行行尾的內容納入選取項目	Alt-Shift-Right	selecttolineend
將目前行行首到游標處的內容納入選取項目	Alt-Shift-Left	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Ctrl-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Ctrl-Shift-Home	selecttostart
將下一行以上的內容納入選取項目	Shift-Up	selectup
將游標左側的下一個文字納入選取項目	Ctrl-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Ctrl-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Ctrl-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection
縮小字型	Ctrl--	smallerfont
如果選取多行，則將其重新排序	Ctrl-Alt-S	sortlines
在目前行的行尾新增游標	Ctrl-Alt-L	splitIntoLines
將游標的內容移動到行尾，另起一行	Ctrl-0	splitline

描述	按鍵繫結關係	命令
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Ctrl-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Ctrl-/	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	F2	toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Alt-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Ctrl-Alt-E	toggleRecording
將文字換行，若已在將文字換行則停止換行	Ctrl-Q	toggleWordWrap
將選取項目全部變更為小寫	Ctrl-Shift-U	tolowercase
將選取項目全部變更為大寫	Ctrl-U	touppercase
調換選取項目	Alt-X	transposeletters
展開選取的程式碼	Alt-Shift-L Ctrl-Shift-F1	unfold
展開整個檔案的程式碼摺疊	Alt-Shift-0	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Ctrl-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Ctrl-Alt-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Ctrl-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Ctrl-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Ctrl-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Alt-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Alt-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	Ctrl-B	build
恢復目前暫停的程序	F8	resume
執行或除錯目前的應用程式	Alt-F5	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11	stepinto
離開目前的函數範圍	Shift-F11	stepout
略過堆疊上目前的運算式	F10	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Ctrl-Shift-C	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的 Windows / Linux Sublime 預設按鍵繫結關係參考

以下是 AWS Cloud9 IDE 中 Windows / Linux 作業系統的 Sublime 鍵盤模式按鍵繫結關係清單。

如需詳細資訊，請前往 AWS Cloud9 IDE：

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤上，選擇 Keybindings (按鍵繫結關係)。
3. 針對 Keyboard Mode (鍵盤模式)，請選擇 Sublime。
4. 針對 Operating System (作業系統)，請選擇 Windows / Linux。

另請參閱[使用按鍵繫結關係](#)

- [一般](#)

- [標籤](#)
- [面板](#)
- [程式碼編輯器](#)
- [emmet](#)
- [終端機](#)
- [執行與除錯](#)

一般

描述	按鍵繫結關係	命令
將選取項目新增為監看運算式	Ctrl-Shift-C	addwatchfromselection
從剪貼簿移除剪下選取項目	Esc	clearcut
顯示程式碼完成內容選單	Ctrl-Space	complete
程式碼完成，然後覆寫	Ctrl-Shift-Space Alt-Shift-Space	completeoverwrite
將選取項目複製到剪貼簿	Ctrl-C	copy
將選取項目剪下到剪貼簿	Ctrl-X	cut
從游標處刪除至該行的開頭	Ctrl-Shift-Backspace Ctrl-K Ctrl-Backspace	delete_to_hard_bol
從游標處刪除至該行的結尾	Ctrl-Shift-Delete Ctrl-K Ctrl-K	delete_to_hard_eol
展開程式碼，如適用	Tab	expandSnippet
顯示目前文件的尋找及取代列	Ctrl-F	find
反白所有符合的選項	Alt-F3	find_all_under

描述	按鍵繫結關係	命令
反白下一個符合的選項	Ctrl-F3	find_under
反白游標周圍區域以及所有符合反白區域的項目	Ctrl-D	find_under_expand
反白游標周圍區域並為所有符合反白區域的項目加上輪廓線條	Ctrl-K Ctrl-D	find_under_expand_skip
反白上一個符合的選項	Ctrl-Shift-F3	find_under_prev
在目前文件中選取所有符合尋找的項目	Ctrl-Alt-K	findAll
移至目前文件中您上次輸入之尋找查詢的下一個符合項目	F3	findnext
移至目前文件中您上次輸入之尋找查詢的上一個符合項目	Shift-F3	findprevious
在編輯器的作用中檔案的插入點符號處顯示所有已知參考	Shift-F3	findReferences
開啟 Environment (環境) 視窗，然後使檔案清單處於作用中狀態	Shift-Esc	focusTree
重新格式化選取的 JavaScript 程式碼	Ctrl-Alt-F	formatcode
顯示「移至行」方塊	Ctrl-G	gotoline
隱藏尋找和取代列 (若其目前顯示)	Esc	hidesearchreplace
移至游標處的變數或函數定義	F12	jumptodef

描述	按鍵繫結關係	命令
如果已在 AWS Resources (AWS 資源) 視窗的 Lambda 的區塊中選取本機 Lambda 函數，請嘗試將函數上傳到 Lambda 作為遠端函數	Ctrl-Shift-U	lambdaUploadFunction
移至目前字詞的結尾。	Ctrl-Right	moveToWordEndRight
移至目前字詞的開頭。	Ctrl-Left	moveToWordStartLeft
建立新檔案	Alt-N	newfile
顯示 Preferences (偏好設定) 標籤	Ctrl-,	openpreferences
開啟 Terminal (終端機) 標籤，然後切換到檔案清單中所選檔案的上層資料夾	Alt-L	opentermhere
在游標處貼上剪貼簿目前的內容	Ctrl-V	paste
針對修復錯誤顯示建議	Ctrl-F3	quickfix
重做上一個動作	Ctrl-Shift-Z Ctrl-Y	redo
重新整理預覽窗格	Ctrl-Enter	reloadpreview
啟動選取項目的重新命名重構	Ctrl-Alt-R	renameVar
顯示目前文件的尋找及取代之列，並以取代之為運算式為焦點	Ctrl-H	replace
使用尋找及取代之列中的「取代之為」運算式來取代所有尋找運算式符合項目	Ctrl-Alt-Enter	replaceall

描述	按鍵繫結關係	命令
使用尋找及取代列中的「取代之為」運算式來取代下一個尋找運算式符合項目	Ctrl-Shift-H	replacenext
重新執行初始化指令碼	Ctrl-Enter	rerunInitScript
重新啟動環境	Ctrl-R	restartc9
將目前的檔案重設為上次儲存的版本	Ctrl-Shift-Q	reverttosaved
將每個開啟的檔案重設為其已儲存的版本	Alt-Shift-Q	reverttosavedall
將目前的檔案儲存至磁碟	Ctrl-S	save
使用其他檔案名稱將目前的檔案儲存至磁碟	Ctrl-Shift-S	saveas
顯示多個檔案的尋找及取代列	Ctrl-Shift-F	searchinfiles
將從游標開始至目前字詞結尾的內容納入選取項目	Ctrl-Shift-Right	selectToWordEndRight
將從游標開始至目前字詞開頭的內容納入選取項目	Ctrl-Shift-Left	selectToWordStartLeft
顯示 Process List (處理序清單) 對話方塊	Ctrl-Alt-P	showprocesslist
復原上一個動作	Ctrl-Z	undo

標籤

描述	按鍵繫結關係	命令
關閉目前窗格中所有開啟的標籤，但目前的標籤除外	Ctrl-Alt-W	closeallbutme
關閉所有窗格中所有開啟的標籤	Alt-Shift-W	closealltabs
關閉目前的窗格	Ctrl-W	closepane
關閉目前的標籤	Alt-W	closetab
向下移一個窗格	Ctrl-Meta-Down	gotopanedown
向左移一個窗格	Ctrl-Meta-Left	gotopaneleft
向右移一個窗格	Ctrl-Meta-Right	gotopaneright
向上移一個窗格	Ctrl-Meta-Up	gottopaneup
向左移一個標籤	Ctrl-Page Up	gototableft
向右移一個標籤	Ctrl-Page Down	gototabright
將目前的標籤向下移動一個窗格，如果標籤已在最底端，則在該處建立分割標籤	Ctrl-Meta-Down	movetabdown
向左移動目前的標籤，如果標籤已在最左端，則在該處建立分割標籤	Ctrl-Meta-Left	movetableft
向右移動目前的標籤，如果標籤已在最右端，則在該處建立分割標籤	Ctrl-Meta-Right	movetabright

描述	按鍵繫結關係	命令
將目前的標籤向上移動一個窗格，如果標籤已在最頂端，則在該處建立分割標籤	Ctrl-Meta-Up	movetabup
移至下一個標籤	Ctrl-Tab	nexttab
移至上一個窗格	Ctrl-Shift-`	previouspane
移至上一個標籤	Ctrl-Shift-Tab	previoustab
返回最後一個標籤	Esc	refocusTab
再次開啟最後一個標籤	Ctrl-Shift-T	reopenLastTab
在檔案樹狀目錄中顯示目前的標籤	Ctrl-E	revealtab
移至第十個標籤	Ctrl-0	tab0
移至第一個標籤	Ctrl-1	tab1
移至第二個標籤	Ctrl-2	tab2
移至第三個標籤	Ctrl-3	tab3
移至第四個標籤	Ctrl-4	tab4
移至第五個標籤	Ctrl-5	tab5
移至第六個標籤	Ctrl-6	tab6
移至第七個標籤	Ctrl-7	tab7
移至第八個標籤	Ctrl-8	tab8
移至第九個標籤	Ctrl-9	tab9

面板

描述	按鍵繫結關係	命令
以 Go to Anything (移至任何內容) 模式顯示 Go (執行) 視窗	Ctrl-E Ctrl-P	gotoanything
以 Go to Command (移至命令) 模式顯示 Go (執行) 視窗	Ctrl-. F1	gotocommand
以 Go to File (移至檔案) 模式顯示 Go (執行) 視窗。	Ctrl-0	gotofile
以 Go to Symbol (移至符號) 模式顯示 Go (執行) 視窗。	Ctrl-Shift-0	gotosymbol
顯示 Outline (大綱) 視窗	Ctrl-R Ctrl-Shift-R	outline
顯示原本隱藏的 Console (主控台) 視窗，若已顯示則隱藏	Ctrl-`	toggleconsole
顯示原本隱藏的 Environment (環境) 視窗，若已顯示則隱藏	Ctrl-K Ctrl-B	toggletree

程式碼編輯器

描述	按鍵繫結關係	命令
在作用中游標的上一行新增游標，若已新增游標，則在該游標的上一行新增另一個游標	Ctrl-Alt-Up	addCursorAbove
在作用中游標的上一行新增第二個游標，若已新增第二個游標，則將第二個游標向上移動一行	Ctrl-Alt-Shift-Up	addCursorAboveSkip Current

描述	按鍵繫結關係	命令
在作用中游標的下一行新增游標，若已新增游標，則在該游標的下一行新增另一個游標	Ctrl-Alt-Down	addCursorBelow
在作用中游標的下一行新增第二個游標，若已新增第二個游標，則將第二個游標向下移動一行	Ctrl-Alt-Shift-Down	addCursorBelowSkipCurrent
將所有游標移動到與其各行作用中游標相同空格的位置 (若它們未對齊)	Ctrl-Alt-A	alignCursors
倒退一個空格	Shift-Backspace Backspace	backspace
將選取項目縮排一個定位點	Ctrl-]	blockindent
將選取項目凸排一個定位點	Ctrl-[blockoutdent
控制是否能將焦點從編輯器切換至 IDE 中的其他位置	Ctrl-Z Ctrl-Shift-Z Ctrl-Y	cancelBrowserUndoInAce
將選取項目置中	Ctrl-K Ctrl-C	centerselection
複製行的內容，並在下一行貼上複製的內容	Alt-Shift-Down	copylinesdown
複製行的內容，並在上一行貼上複製的內容	Alt-Shift-Up	copylinesup
剪下選取項目，或如果沒有選取項目，請刪除一個空格	Shift-Delete	cut_or_delete
刪除一個空格	Delete	del

描述	按鍵繫結關係	命令
複製選取項目的內容，並立即在選取項目的後方貼上複製的內容	Ctrl-Shift-D	duplicateSelection
將目前行的內容納入選取項目	Ctrl-Shift-L	expandtoline
將直到下一個相符符號的內容納入選取項目	Ctrl-Shift-M	expandToMatching
折疊選取的程式碼；若選取的單位已摺疊，則予以展開	Alt-L Ctrl-F1	fold
摺疊所有可摺疊的元素，但目前的選取範圍除外	Ctrl-K Ctrl-1	foldOther
向下移一行	Down	golinedown
向上移一行	Up	golineup
移至檔案尾端	Ctrl-End	gotoend
向左移一個空格	Left	gotoleft
移至目前行的行尾	Alt-Right End	gotolineend
移至目前行的行首	Alt-Left Home	gotolinestart
移至下一個錯誤	Ctrl-F6	goToNextError
向下移一頁	Page Down	gotopagedown
向上移一頁	Page Up	gotopageup
移至上一個錯誤	Ctrl-Shift-F6	goToPreviousError
向右移一個空格	Right	gotoright
移至檔案開頭	Ctrl-Home	gotostart

描述	按鍵繫結關係	命令
向左移一個文字	Ctrl-Left	gotowordleft
向右移一個文字	Ctrl-Right	gotowordright
將選取項目縮排一個定位點	Tab	indent
將從游標開始至目前字詞開頭的內容納入選取項目	Ctrl-J	joinlines
移至目前範圍中的相符符號	Ctrl-M	jumptomatching
放大字型	Ctrl-- Ctrl-= Ctrl-+	largerfont
將游標左側的數字減 1 (若其為數字)	Alt-Down	modifyNumberDown
將游標左側的數字加 1 (若其為數字)	Alt-Up	modifyNumberUp
將選取項目向下移動一行	Ctrl-Shift-Down	movelinesdown
將選取項目向上移動一行	Ctrl-Shift-Up	movelinesup
將選取項目凸排一個定位點	Shift-Tab	outdent
開啟覆寫模式，若已開啟則予以關閉	Insert	overwrite
刪除目前行的內容	Ctrl-Shift-K	removeline
從游標處刪除至目前行的行尾	Alt-Delete	removetolineend
從目前行的行首刪除至游標處	Alt-Backspace	removetolinestart
刪除游標左側的文字	Ctrl-Backspace	removewordleft
刪除游標右側的文字	Ctrl-Delete	removewordright

描述	按鍵繫結關係	命令
重新播放先前記錄的按鍵輸入	Ctrl-Shift-Q	replaymacro
將目前檔案向下捲動一行	Ctrl-Down	scrolldown
將目前檔案向上捲動一行	Ctrl-Up	scrollup
選取所有可選取的內容	Ctrl-A	selectall
將下一行以下的行納入選取項目	Shift-Down	selectdown
將左側的下一個空格納入選取項目	Shift-Left	selectleft
從游標處開始，將目前行的其餘部分納入選取項目	Shift-End	selectlineend
將目前行行首至游標處的內容納入選取範圍	Shift-Home	selectlinestart
納入選取項目後的更多相符選取項目	Ctrl-Alt-Right	selectMoreAfter
納入選取項目前的更多相符選取項目	Ctrl-Alt-Left	selectMoreBefore
納入選取項目後的下一個相符選取項目	Ctrl-Alt-Shift-Right	selectNextAfter
納入選取項目前的下一個相符選取項目	Ctrl-Alt-Shift-Left	selectNextBefore
選取或尋找下一個相符選取項目	Alt-K	selectOrFindNext
選取或尋找上一個相符選取項目	Alt-Shift-K	selectOrFindPrevious

描述	按鍵繫結關係	命令
將游標處以下至目前頁面頁尾的內容納入選取項目	Shift-Page Down	selectpagedown
將游標處以上至目前頁面頁首的內容納入選取項目	Shift-Page Up	selectpageup
將游標右側的下一個空格納入選取項目	Shift-Right	selectright
將游標處以下至目前檔案結尾的內容納入選取項目	Ctrl-Shift-End	selecttoend
將游標處至目前行行尾的內容納入選取項目	Alt-Shift-Right	selecttolineend
將目前行行首到游標處的內容納入選取項目	Alt-Shift-Left	selecttolinestart
將游標處至下一個相符符號的內容納入目前的範圍	Ctrl-Shift-P	selecttomatching
將游標處以上至目前檔案開頭的內容納入選取項目	Ctrl-Shift-Home	selecttostart
將下一行以上的內容納入選取項目	Shift-Up	selectup
將游標左側的下一個文字納入選取項目	Ctrl-Shift-Left	selectwordleft
將游標右側的下一個文字納入選取項目	Ctrl-Shift-Right	selectwordright
顯示 Preferences (偏好設定) 標籤	Ctrl-,	showSettingsMenu
清除所有之前的選取項目	Esc	singleSelection

描述	按鍵繫結關係	命令
縮小字型	Ctrl-- Ctrl-Shift-= Ctrl-Shift-+	smallerfont
如果選取多行，則將其重新排序	F9	sortlines
在目前行的行尾新增游標	Ctrl-Shift-L	splitIntoLines
以區段註解字元環繞選取項目，若已有註解字元則將其移除	Ctrl-Shift-/	toggleBlockComment
在選取的每一行行首新增行註解字元，若已有註解字元則將其移除	Ctrl-/	togglecomment
摺疊程式碼，若已摺疊則移除程式碼摺疊	Ctrl-Shift-[toggleFoldWidget
摺疊父程式碼，如已摺疊則移除摺疊	Alt-F2	toggleParentFoldWidget
開始錄製按鍵輸入，若已開始錄製則將其停止	Ctrl-Q	toggleRecording
將文字換行，若已在將文字換行則停止換行	Ctrl-Q	toggleWordWrap
將選取項目全部變更為小寫	Ctrl-K Ctrl-L	tolowercase
將選取項目全部變更為大寫	Ctrl-K Ctrl-U	touppercase
調換選取項目	Alt-X	transposeletters
展開選取的程式碼	Ctrl-Shift-]	unfold
展開整個檔案的程式碼摺疊	Ctrl-K Ctrl-0 Ctrl-K Ctrl-J	unfoldall

emmet

描述	按鍵繫結關係	命令
評估單純的數學運算式 (例如 $2*4$ 或 $10/2$) 並輸出其結果	Shift-Ctrl-Y	emmet_evaluate_math_expression
根據目前檔案的語法，將類似 CSS 的縮寫展開為 HTML、XML 或 CSS 程式碼	Ctrl-Alt-E	emmet_expand_abbreviation
透過定位停駐點，周遊展開之類似 CSS 的縮寫	Tab	emmet_expand_abbreviation_with_tab
移至下一個可編輯的程式碼部分	Shift-Ctrl-.	emmet_select_next_item
移至上一個可編輯的程式碼部分	Shift-Ctrl-,	emmet_select_previous_item
展開縮寫，然後將目前的選取項目放入所產生程式碼片段的最後一個元素	Shift-Ctrl-A	emmet_wrap_with_abbreviation

終端機

描述	按鍵繫結關係	命令
開啟新的 Terminal (終端機) 標籤	Alt-T	openterminal
在編輯器及 Terminal (終端機) 標籤之間切換	Alt-S	switchterminal

執行與除錯

描述	按鍵繫結關係	命令
建置目前的檔案	F7 Ctrl-B	build
恢復目前暫停的程序	F8	resume
執行或除錯目前的應用程式	Ctrl-Shift-B	run
執行或除錯上次執行的檔案	F5	runlast
進入堆疊上的下一個函數	F11	stepinto
離開目前的函數範圍	Shift-F11	stepout
略過堆疊上目前的運算式	F10	stepover
停止執行或除錯目前的應用程式	Shift-F5	stop
停止建置目前的檔案	Ctrl-Break	stopbuild

AWS Cloud9 整合開發環境 (IDE) 的命令參考

如何在 AWS Cloud9 IDE 執行命令：

1. 選擇 Go (執行) 按鈕 (放大鏡) 顯示 Go (執行) 視窗。如果 Go (執行) 按鈕未顯現，請從選單列選擇 Window, Go (視窗、執行)。
2. 在 Go To Anything (移至任何內容) 方塊中，開始輸入 command group (命令群組) 的名稱，例如 Code Editor (程式碼編輯器)。群組中包含針對共同主題或 IDE 功能而整理的多個命令。
3. 在 Commands (命令) 標題下方，從群組中選擇要執行的特定命令。

可用的命令群組

命令群組	描述
AWS	AWS 工具組 的命令

命令群組	描述
Clipboard	用於複製與貼上內容的命令
Code Editor	用於導覽程式碼編輯器界面並與編輯器內容互動的命令
Emmet	用於處理 HTML 和 CSS 內容專用的 Emmet 工具組 的命令
General	用於管理 IDE 組態和專案檔案的其他命令
Panels	用於管理 IDE 界面中的面板顯示內容的命令
Run & Debug	用於在 AWS Cloud9 執行和除錯專案的命令
Tabs	用於管理 IDE 界面中索引標籤的顯示與導覽的命令
Terminal	用於管理命令列終端機的命令
Window	用於管理 IDE 視窗的窗格配置的命令

使用其他 AWS 服務

使用時AWS Cloud9，您可以與 Amazon Lightsail 密切合作AWS CodeStar, 和AWS CodePipeline. 本節的主題說明如何執行上述操作。

Important

AWS 工具組功能提供便利的視覺界面，方便您使用關鍵 AWS 服務，例如 AWS Lambda、AWS Serverless Application Model 和 Amazon S3。如需詳細資訊，請參閱 [AWS Toolkit](#)。

主題

- [在 AWS Cloud9 整合開發環境 \(IDE\) 使用 Amazon Lightsail 執行個體](#)
- [在AWS Cloud9整合式開發環境 \(IDE\) 中使用AWS CodeStar專案](#)
- [與亞馬遜合作CodeWhisperer通過使用AWS Cloud9](#)
- [在 AWS Cloud9 整合開發環境 \(IDE\) 中使用 AWS CodePipeline](#)
- [與 Amazon 合作 CodeCatalyst](#)
- [在 AWS Cloud9 整合式開發環境 \(IDE\) 中使用 AWS CDK](#)

在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體

您可以使用 AWS Cloud9 IDE，處理已使用熱門應用程式和架構預先設定的 Amazon Lightsail 執行個體上的程式碼。它們包括 WordPress、LAMP (Linux、Apache、MySQL 和 PHP)、Node.js、NGINX、Drupal 和 Joomla。也包括 Linux 發行版，例如 Amazon Linux、Ubuntu、Debian、FreeBSD 和 openSUSE。

Lightsail 提供方便、快速設定的虛擬私有伺服器解決方案。Lightsail 提供運算、儲存及聯網容量，以及可在雲端部署及管理網站和 Web 應用程式的能力。您可以使用 Lightsail 以低廉、可預測的每月價格快速啟動專案。如需詳細資訊，請參閱 [Amazon Lightsail 功能](#)。

在本主題中，您將建立和設定與 AWS Cloud9 相容的 Linux 型 Lightsail 執行個體，接著還要建立 AWS Cloud9 SSH 開發環境並將其連接至 Lightsail 執行個體。

Note

完成這些程序可能會向您的 AWS 帳戶收取費用。收費的項目可能包括 Lightsail 等服務。如需詳細資訊，請參閱 [Amazon Lightsail 定價](#)。

若要建立並設定更進階的解決方案，其中包括具有 AWS Cloud9 IDE 的工具鏈、原始檔控制、建置、部署、虛擬伺服器或無伺服器資源等，請參閱 [使用 AWS CodeStar 專案](#)。

若要使用 AWS Cloud9 IDE 來處理執行 Amazon Linux 或 Ubuntu Server 且未包含任何範本程式碼的 Amazon EC2 執行個體，請參閱 [入門：基本教學課程](#)。

- [步驟 1：建立 Linux 型 Lightsail 執行個體](#)
- [步驟 2：設定執行個體以搭配 AWS Cloud9 使用](#)
- [步驟 3：建立並連接至 AWS Cloud9 SSH 開發環境](#)
- [步驟 4：使用 AWS Cloud9 IDE 變更執行個體上的程式碼](#)

步驟 1：建立 Linux 型 Lightsail 執行個體

在此步驟中，您要使用 Lightsail 主控台建立 Amazon EC2 執行個體來執行 Linux 發行版本的應用程式。此執行個體自動包含下列項目：

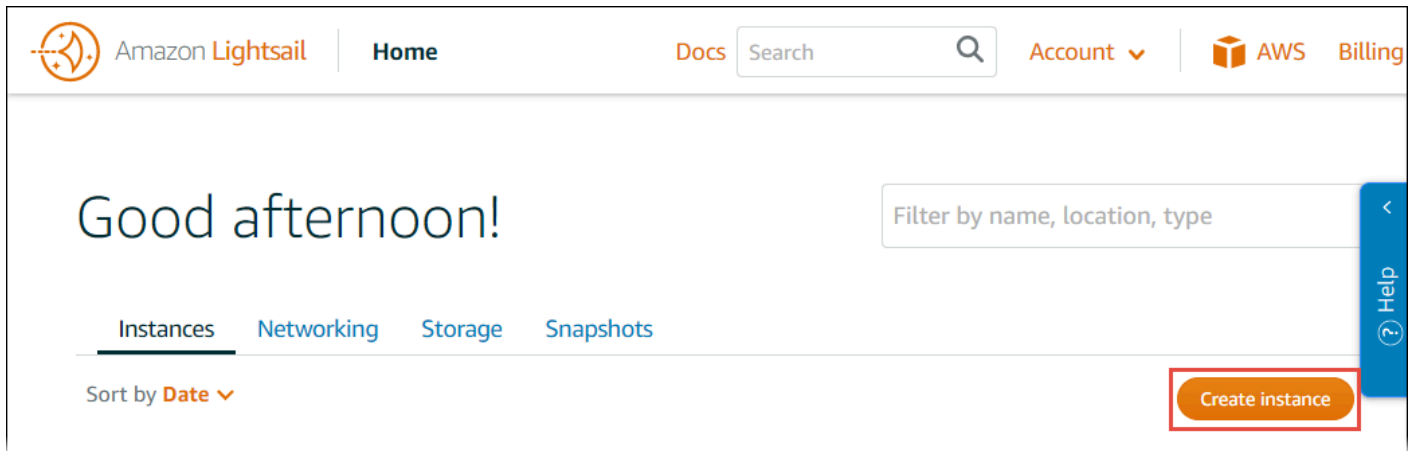
- 公有及私有 IP 地址 (您可於稍後建立靜態公有 IP)。
- 使用 SSH 透過連接埠 22、HTTP 透過連接埠 80 及 HTTPS 透過連接埠 443 存取執行個體 (您可以變更這些設定。)
- 區塊儲存磁碟 (您可於稍後連接額外的磁碟)。
- 內建系統報告。

在 Lightsail 主控台上，您稍後可以備份、重新啟動、停止或刪除執行個體。

1. 前往 <https://lightsail.aws.amazon.com> 開啟 Lightsail 主控台，然後登入。

我們建議您在 AWS 帳戶使用 IAM 管理員使用者的憑證來登入。若您無法以 IAM 管理員使用者身分登入，請洽詢您的 AWS 帳戶管理員。


2. 出現提示時，選擇要在主控台使用的語言，然後選擇 Save (儲存)。
3. 出現提示時，選擇 Let's get started (讓我們開始吧)。
4. 在首頁已選取 Instances (執行個體) 標籤的情況下，選擇 Create instance (建立執行個體)。




5. 在 Instance location (執行個體位置) 中，確定位置為您要在其中建立執行個體的 AWS 區域 AWS Cloud9。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [AWS Cloud9](#)。若要變更 AWS 區域和/或可用區域，請選擇 Change AWS Region and Availability Zone (變更 AWS 區域與可用區域)，然後依照螢幕上的指示進行。
6. 在 Pick your instance image (挑選執行個體映像) 中的 Select a platform (選取平台) 已選擇 Linux/Unix 且 Select a blueprint (選取藍圖) 已選擇 Apps + OS (應用程式與作業系統) 的情況下，選擇一份藍圖。

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

如果想要建立不含應用程式的執行個體，應選擇 OS Only (僅作業系統) 而非 Apps + OS (應用程式與作業系統)，然後選擇任一發行版本。

若要了解各種可用選項，請參閱 Lightsail 網站上的 [選擇 Amazon Lightsail 執行個體映像](#)。

7. 從 Choose your instance plan (選擇執行個體方案) 中選擇任一方案，或者保留所選的預設方案。
8. 在 Name your instance (為執行個體命名) 中輸入執行個體的名稱，或者保留建議的預設名稱。
9. 針對執行個體數目，輸入您想要建立的執行個體數目，或者保留預設為單一執行個體 (x 1)。
10. 選擇建立。

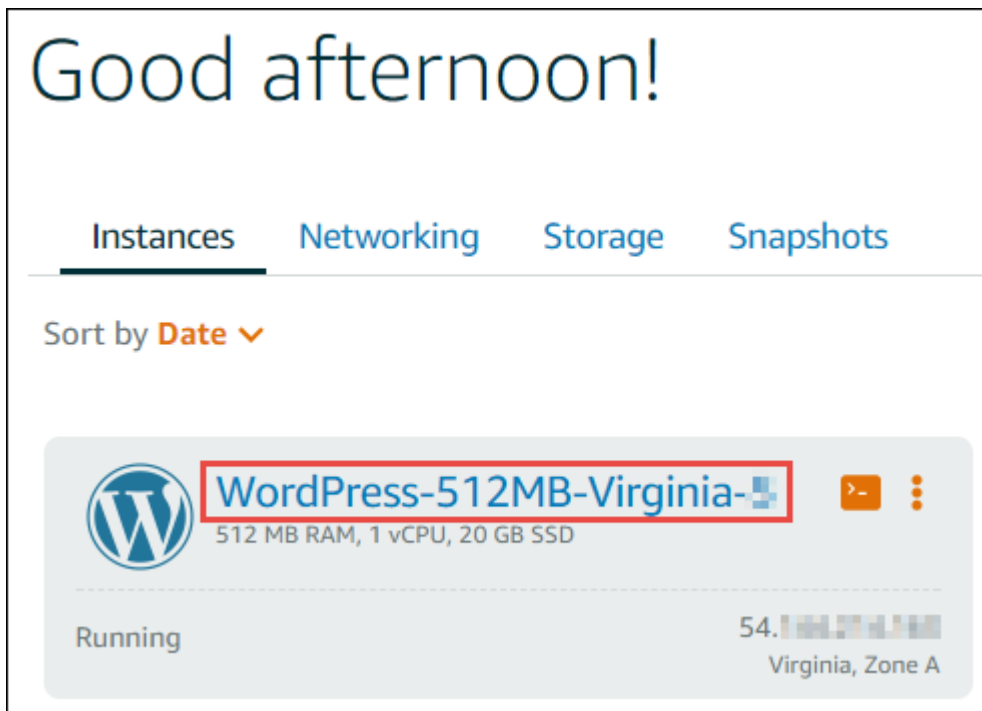
步驟 2：設定執行個體以搭配 AWS Cloud9 使用

在此步驟中，您要連接到運作中的執行個體，然後對其進行設定以供 AWS Cloud9 稍後使用。

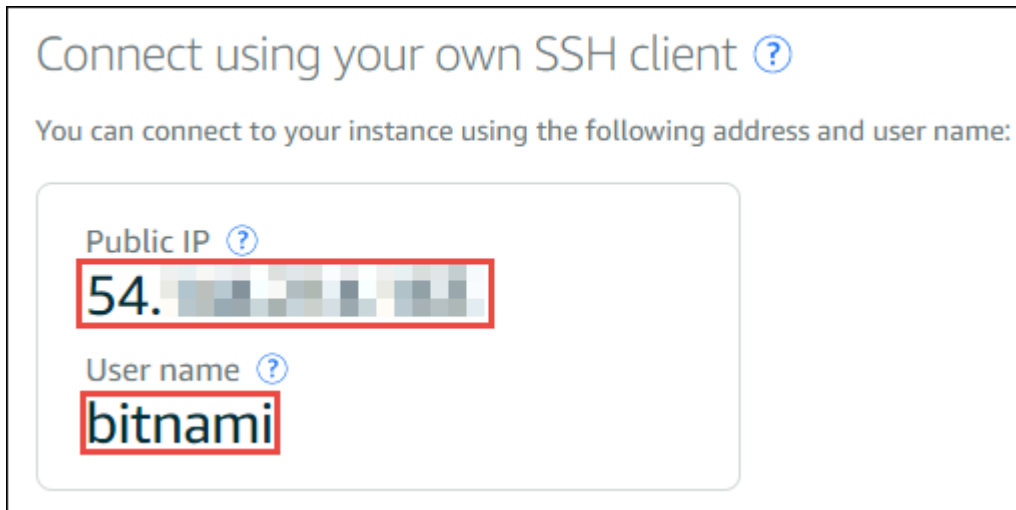
Note

以下說明假設您由前一步驟選擇了 Apps + OS (應用程式與作業系統)。如果您是選擇 OS Only (僅作業系統) 和 Ubuntu 以外的發行版本，可能就需要視情況調整以下說明的操作方式。

1. 在前一步驟的 Lightsail 主控台仍開啟的情況下，在 Instances (執行個體) 標籤上的執行個體卡片選擇執行個體的名稱。



2. 在 Connect (連接) 標籤上，於 Connect using your own SSH client (使用您自己的 SSH 用戶端連接) 中，記下 Public IP (公有 IP) 和 User name (使用者名稱) 的值，因為稍後需要用到。



3. 選擇 Connect using SSH (使用 SSH 連接)。
4. 確認執行個體已有最新的系統更新。為此，請從畫面上顯示的終端機工作階段，執行 **sudo apt update** 命令。
5. 檢查是否已安裝 Python，如已安裝，務必確定為 2.7 版。若要檢查版本，請執行 **python --version** 命令並記下其顯示的版本編號。如果未出現版本編號或版本不是 2.7，則執行 **sudo apt install -y python-minimal** 命令，於執行個體上安裝 Python 2.7。
6. 檢查是否已安裝 Node.js，如已安裝，再確認是否為 0.6.16 或更新版本。若要檢查版本，請執行 **node --version** 命令並記下其顯示的版本編號。如果未出現版本編號或版本不是 0.6.16 或更新版本，建議您使用 Node Version Manager (nvm)，在執行個體上安裝 Node.js。

為此，請依序逐一執行以下命令，其用意是更新執行個體、在執行個體上安裝 Node Version Manager (nvm)、由執行個體啟用 nvm，然後於執行個體上安裝最新版 Node.js。

```
sudo apt update
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
. ~/.bashrc
nvm install node
```

7. 執行 **which node** 命令並記下其顯示的值。以供稍後使用。

Note

如果 **which node** 命令的輸出類似於 `/usr/sbin/node`，表示 AWS Cloud9 未能在該路徑下找到 Node.js。請改用 nvm，依本程序的前一步驟所述來安裝 Node.js。然後，再次執行 **which node** 命令並記下其出現的新值。

8. 在執行個體上[下載並執行 AWS Cloud9 安裝程式](#)。

步驟 3：建立並連接至 AWS Cloud9 SSH 開發環境

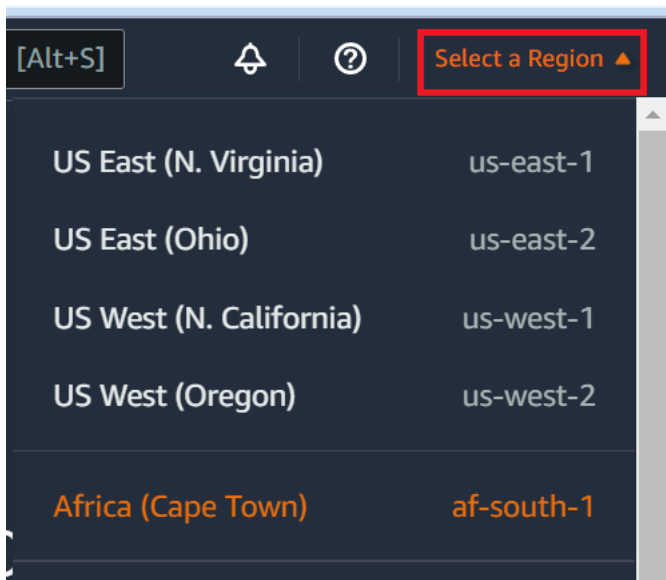
在此步驟中，您要使用 AWS Cloud9 主控台和執行個體的終端機建立 SSH 環境，然後將該環境連接至運作中的執行個體。

1. 由前一步驟仍開啟終端機工作階段的情況下，登入 AWS Cloud9 主控台，如下所示：
 - 如果您是唯一使用 AWS 帳戶的個人，或單一 AWS 帳戶中的 IAM 使用者，請前往 <https://console.aws.amazon.com/cloud9/>。
 - 如果您的組織使用 AWS IAM Identity Center，請要求 AWS 帳戶管理員提供登入指示。

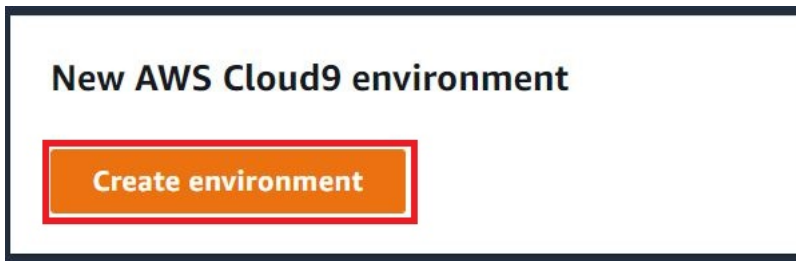
Note

此步驟將同時使用兩種不同的 AWS 服務。現在，假設您以 IAM 管理員使用者的身分登入 Lightsail 主控台，但是想要不同的實體擁有新的 SSH 環境。對於這種情況，建議開啟不同的 Web 瀏覽器並以該實體的身分登入 AWS Cloud9 主控台。

2. 在 AWS Cloud9 主控台中，選擇您已在其中建立執行個體的 AWS 區域。



3. 若顯示歡迎頁面，請針對 New AWS Cloud9 environment (新增 AWS Cloud9 環境)，選擇 Create environment (建立環境)。否則，請選擇 Create environment (建立環境)。



或者：



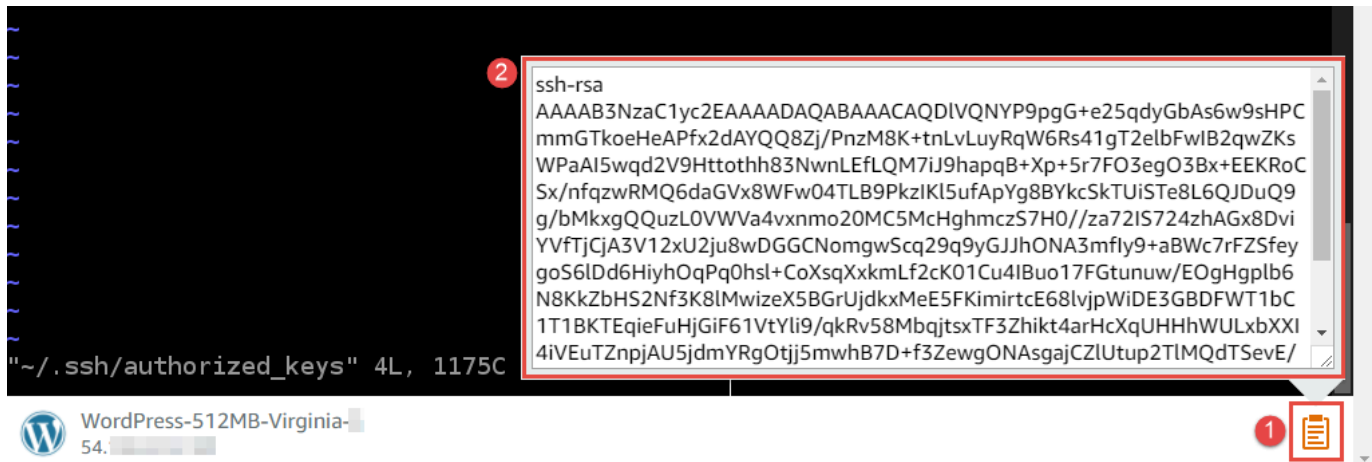
4. 在 Name environment (為環境命名) 頁面的 Name (名稱) 上，輸入環境的名稱。
5. 在「描述」欄位中將描述新增至您的環境。
6. 對於環境類型，選擇現有計算。這一點很重要，因為您需要選取此選項才能顯示 [使用者] 和 [主機] 選項。
7. 對於 User (使用者)，輸入您稍早記下的 User name (使用者名稱) 值。
8. 對於 Host (主機)，輸入您稍早記下的 Public IP (公有 IP) 值。
9. 將 Port (連接埠) 保留預設值 22。
10. 展開其他詳細資料。
11. 對於 Environment path (環境路徑)，輸入登入後 AWS Cloud9 開始的路徑，也就是 ~/。這是使用者主目錄的根目錄。
12. 對於 Node.js binary path (Node.js 二進位檔路徑)，輸入您稍早記下的 **which node** 值。
13. 將 SSH jump host (SSH 跳接主機) 保留空白。
14. 將 AWS Cloud9 為此環境建立的公有 SSH 金鑰存入系統剪貼簿。為此，選擇 Copy key to clipboard (複製金鑰至剪貼簿)。

Note

若要查看複製的公有 SSH 金鑰值，請展開 View public SSH key (檢視公有 SSH 金鑰)。

15. 將您剛才複製的公有 SSH 金鑰值存放於執行個體。若要這樣做，請使用已安裝在執行個體的 vi (熱門文字編輯器)：
 - a. 從執行個體的終端機工作階段，執行 **vi ~/.ssh/authorized_keys** 命令。
 - b. 在出現的 vi 編輯器中，移至檔案尾端並切換到插入模式。方法如下：先按 I 再按 A。(-- INSERT -- 出現在 vi 編輯器底部。)

- c. 按兩次 Enter 於檔案末尾添加兩個換行符號。
- d. 將系統剪貼簿的內容 (當中包含您剛才複製的公有 SSH 金鑰值) 貼入終端機工作階段剪貼簿。方法如下：於終端機工作階段視窗的右下角選擇剪貼簿按鈕，然後將系統剪貼簿的內容貼入方塊中。



- e. 將終端機工作階段剪貼簿的內容貼入 vi 編輯器。方法如下：於 vi 編輯器的插入點位置，按 Ctrl + Shift + V。
- f. 儲存檔案。方法如下：按 Esc 進入命令模式。(vi 編輯器底部的 -- INSERT -- 將消失)。輸入 :wq (意思是 write 檔案，然後 quit vi 編輯器)，然後按 Enter。

16 返回 AWS Cloud9 主控台，然後選擇 Next step (下一個步驟)。

17 在 Review choices (檢閱選擇) 頁面上，選擇 Create environment (建立環境)。等待 AWS Cloud9 建立您的環境，隨後便會顯示環境的 AWS Cloud9 IDE。這可能需要幾分鐘的時間。

AWS Cloud9 在建立環境之後，會顯示環境的 AWS Cloud9 IDE。

若 AWS Cloud9 經過至少五分鐘後仍未顯示 IDE，問題可能出在您的 Web 瀏覽器、您的 AWS 存取許可、執行個體或是關聯 Virtual Private Cloud (VPC)。如需可行的修正方法，請參閱「[故障診斷](#)」一文的無法開啟環境。

步驟 4：使用 AWS Cloud9 IDE 變更執行個體上的程式碼

新環境的 IDE 出現後，即可使用 IDE 的終端機工作階段，而不必再使用 Lightsail 終端機工作階段。IDE 提供豐富的程式碼編輯體驗，可支援多種程式設計語言和執行時間除錯器。IDE 還包括彩色佈景主題、快速鍵按鍵繫結關係、程式設計語言特定語法著色和程式碼格式編排。

若要了解如何使用 IDE，請參閱 [導覽 AWS Cloud9 IDE](#)。

若要了解如何變更執行個體上的程式碼，建議您參閱以下資源：

- All Lightsail 網站上 [取得由 Bitnami 提供的 Lightsail 映像的應用程式密碼](#)
- Drupal : Bitnami 網站上的 [BitnamiDrupal For AWS 雲端](#) , 以及 Drupal 網站上的 [教學課程與網站使用技巧](#)
- GitLab CE : Bitnami 網站上的 [BitnamiGitLab CE for AWS 雲端](#) , 以及 GitLab 網站上的 [GitLab 文件](#)
- Joomla! : Bitnami 網站上的 [BitnamiJoomla! For AWS 雲端](#) , 以及 Joomla! 網站上的 [開始使用 Joomla!](#)
- LAMP 堆疊 : Bitnami 網站上的 [BitnamiLAMP for AWS 雲端](#)
- Magento : Bitnami 網站上的 [BitnamiMagento For AWS 雲端](#) , 以及 Magento 網站上的 [Magento 使用者手冊](#)
- MEAN : Bitnami 網站上的 [BitnamiMEAN For AWS 雲端](#)
- NGINX : Bitnami 網站上的 [BitnamiNGINX For AWS 雲端](#) , 以及 NGINX 網站上的 [NGINX Wiki](#)
- Node.js : Bitnami 網站上的 [BitnamiNode.Js For AWS 雲端](#) , 以及 Node.js 網站上的 [入門指南](#)
- Ubuntu 上的 Plesk 託管堆疊 : [在 Amazon Lightsail 上設定和配置 Plesk。](#)
- Redmine : Bitnami 網站上的 [Bitnami Redmine For AWS 雲端](#) , 以及 Redmine 網站上的 [入門](#)
- WordPress : Lightsail 網站上的 [從 Amazon Lightsail 執行個體開始使用 WordPress](#) , 以及 Bitnami 網站上的 [Bitnami WordPress For AWS 雲端](#)

在AWS Cloud9整合式開發環境 (IDE) 中使用AWS CodeStar專案

您可以使用 AWS Cloud9 IDE 在專案中處理AWS CodeStar程式碼。

AWS CodeStar 是一種雲端服務，用於建立、管理和處理 AWS 上的軟體開發專案。您可以使用 AWS CodeStar 專案在 AWS 上迅速開發、建置及部署應用程式。AWS CodeStar專案會AWS 服務為您的專案開發工具鏈建立並整合。視您選擇的 AWS CodeStar 專案範本而定，此工具鏈可能包括原始碼控制、建置、部署、虛擬伺服器或無伺服器資源等。如需詳細資訊，請參閱 [AWS CodeStar 使用者指南](#)。

Note

完成這些程序可能會向您的 AWS 帳戶收取費用。這些服務包括 Amazon EC2 等服務的可能費用 AWS CodeStarAWS 服務，並且受支援AWS CodeStar。如需詳細資訊，請參閱 [Amazon EC2 定價](#)、[AWS CodeStar定價](#)和[雲端服務定價](#)。

若要使用 AWS Cloud9 IDE，搭配新啟動的 Amazon EC2 執行個體運作，而此執行個體是使用熱門應用程式或架構 (例如 WordPress、MySQL、PHP、Node.js、NGINX、Drupal 或 Joomla，或 Ubuntu、Debian、FreeBSD 或 openSUSE 等 Linux 發行版) 預先設定的，您可

以使用 Amazon Lightsail 搭配 AWS Cloud9。為此，請跳過本主題的其餘部分，直接參閱[使用 Amazon Lightsail 執行個體](#)。

若要使用 AWS Cloud9 IDE 處理執行 Amazon Linux 且未包含任何範本程式碼的新啟動 Amazon EC2 執行個體，請跳過本主題的其餘部分，改為直接參閱[入門：基本教學課程](#)。

- [步驟 1：準備使用AWS CodeStar專案](#)
- [步驟 2：在中建立專案 AWS CodeStar](#)
- [步驟 3：建立 AWS Cloud9 開發環境並將其連接至專案](#)

步驟 1：準備使用AWS CodeStar專案

在此步驟中，您會建立AWS CodeStar服務角色和 Amazon EC2 key pair，以便開始建立和使用AWS CodeStar專案。

如果您AWS CodeStar之前使用過，請跳至[中的步驟 2：建立專案AWS CodeStar](#)。

對於此步驟，請遵循「AWS CodeStar使用者指南」中的「[設定](#)」AWS CodeStar 中的指示。切勿依照上述指示建立新的 AWS 帳戶、IAM 使用者或 IAM 群組。請使用您在進行 [AWS Cloud9 的小組設定](#)時建立或識別的項目。依照指示完成操作後，返回本主題。

步驟 2：在中建立專案 AWS CodeStar

此步驟將在 AWS CodeStar 中建立專案。

如果您已經有要使用的AWS CodeStar專案，請跳到[步驟 3：建立AWS Cloud9開發環境並將其 Connect 到專案](#)。

對於此步驟，請遵循《使用指南》中的[「建立專案」](#) AWS CodeStar 中的AWS CodeStar指示。在 AWS CodeStar 建立專案精靈中，當來到 Set up tools (設定工具) 頁面或 Connect to your source repository (連接至來源儲存庫) 頁面時，選擇 Skip (略過)，然後返回本主題。

步驟 3：建立 AWS Cloud9 開發環境並將其連接至專案

在此步驟中，您會在AWS CodeStar或AWS Cloud9主控台中建立AWS Cloud9開發環境。然後將新環境連接到AWS CodeStar專案。

對於此步驟，請遵循下列其中一組指示操作。根據AWS Cloud9開發環境類型，您想要使用和AWS CodeStar項目存儲其代碼的存儲庫類型。

環境類型	儲存庫類型	指示
EC2 環境	CodeCommit	《AWS CodeStar使用者指南》中的 專案建立AWS Cloud9環境
SSH 環境	CodeCommit	AWS CodeCommit 範例
EC2 或 SSH 環境	GitHub	GitHub 與使用AWS CodeStar 者指南AWS Cloud9中的搭配使用

與亞馬遜合作CodeWhisperer通過使用AWS Cloud9

什麼是 CodeWhisperer ?

亞馬遜CodeWhisperer是一項由機器學習驅動的服務，可根據開發人員在 IDE 中使用自然語言的註解及其程式碼產生程式碼建議，藉此提升開發人員的生產力。CodeWhisperer可用於Java,JavaScript,Python,C#和TypeScript編程語言，還支持代碼生成Ruby,Go,PHP,C++,C,Shell,Scala,Rust,Kotlin和SQL。該服務與多個整合式開發環境 (IDE) 整合，包括 JetBrains (IntelliJ、PyCharm、WebStorm 和 Rider)、Visual Studio Code、AWS Cloud9 和 AWS Lambda 主控台。有關如何操作的例子CodeWhisperer整合AWS Cloud9並顯示程式碼建議AWS Cloud9IDE 中，請參閱[程式碼範例](#)在亞馬遜CodeWhisperer使用者指南。

有關使用的更多信息CodeWhisperer與AWS Cloud9，請參閱[亞馬遜CodeWhisperer使用者指南](#)。

AWS Cloud9 的 AWS Identity and Access Management 權限

對於CodeWhisperer在中提供建議AWS Cloud9主控台時，您必須為 IAM 使用者或角色啟用正確的 IAM 許可。您必須新增 `codewhisperer:GenerateRecommendations` 許可，如以下 IAM 政策範例所述：

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

```
"Action": ["codewhisperer:GenerateRecommendations"],
"Resource": "*"
}
]
}
```

最佳實務是使用 IAM 政策將更嚴苛許可授予 IAM 主體。

在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline

您可以使用 AWS Cloud9 IDE 處理與 AWS CodePipeline 相容的儲存庫來源碼。

CodePipeline 是一種持續交付服務，您可以用來建模、視覺化和自動化發行軟體所需的步驟，以及您對軟體所做的持續變更。您可以使用 CodePipeline 快速模型化和設定軟體發行程序的不同階段。如需詳細資訊，請參閱 [AWS CodePipeline 使用者指南](#)。

Note

完成這些程序可能會向您的 AWS 帳戶收取費用。這些服務包括可能的費用，例如 Amazon EC2 CodePipeline，Amazon S3 和支持的 AWS 服務 CodePipeline。如需詳細資訊，請參閱 [Amazon EC2 定價](#)、[AWS CodePipeline 定價](#)、[Amazon S3 定價](#) 和 [Cloud Services 定價](#)。AWS CodeStar 提供其他功能和管道 (例如專案範本、儀表板和團隊)。若要使用 AWS CodeStar 而不是 CodePipeline，請略過本主題的其餘部分，並改為參閱 [使用 AWS CodeCommit 專案](#)。

- [步驟 1：建立或識別來源碼儲存庫](#)
- [步驟 2：建立 AWS Cloud9 開發環境、將其連線至程式碼儲存庫，並上傳程式碼](#)
- [步驟 3：準備使用 AWS CodePipeline](#)
- [步驟 4：在 AWS CodePipeline 中建立管道](#)

步驟 1：建立或識別來源碼儲存庫

在此步驟中，您將建立或識別與 CodePipeline 相容的來源碼儲存庫。

稍後在本主題中，您上傳軟體的原始碼至該儲存庫。CodePipeline 將建置、測試和部署在該儲存庫上傳的原始碼，方法是使用您同時建立的相關管道。

您的來源碼儲存庫必須是 CodePipeline 所支援的下列其中一種儲存庫類型：

- [AWS CodeCommit](#)。如果您已經有要使用 CodeCommit 的儲存庫，請跳至[步驟 2：建立 AWS Cloud9 開發環境，將其 Connect 至程式碼儲存庫，然後上傳您的程式碼](#)。否則，若要使用 CodeCommit，請依此順序遵循 AWS CodeCommit 範例中的這些指示，然後返回本主題：
 - [步驟 1：以必要的存取許可設定 IAM 群組](#)
 - [步驟 2：在 AWS CodeCommit 中建立儲存庫](#)
- [Amazon S3](#) 如果您在 Amazon S3 中已經有想要使用的儲存貯體，請直接跳到[步驟 2：建立 AWS Cloud9 開發環境，將其連線至程式碼儲存庫，並上傳程式碼](#)。否則，若要使用 Amazon S3，請依此順序遵循《Amazon Simple Storage Service 使用者指南》中的這些說明，然後返回本主題：
 - [註冊 Amazon S3](#)
 - [建立儲存貯體](#)
- [GitHub](#)。如果您已經在中已有儲存庫 GitHub，您可以使用 [Git 面板](#) 介面在開發環境中複製並建立本機副本。如果您尚未在上設定帳戶或儲存庫 GitHub，請參閱[相關文件](#)以取得指示。

步驟 2：建立 AWS Cloud9 開發環境、將其連線至程式碼儲存庫，並上傳程式碼

在此步驟中，您將在 AWS Cloud9 主控台中建立 AWS Cloud9 開發環境。然後，您可以將環境連接到 CodePipeline 將使用的存放庫。最後，您將為環境使用 AWS Cloud9 IDE，將來源碼上傳至儲存庫。

若要建立環境，請遵循[建立環境](#)中的說明，然後返回本主題 (如果您已經有環境，則可以直接使用，不需要建立新的環境。)

若要將環境連線至儲存庫，然後將來源碼上傳至儲存庫 (若還沒上傳)，請按照下列其中一組說明操作。您選擇的設定取決於可存放來源碼的儲存庫類型。

儲存庫類型	指示
CodeCommit	<p>請遵循以下 AWS CodeCommit 範例中的說明：</p> <ul style="list-style-type: none">• 步驟 3：將環境連線至遠端儲存庫• 步驟 4：將遠端儲存庫複製到您的環境。• 步驟 5：將檔案新增至儲存庫，並替換此步驟中您自己的來源碼

儲存庫類型	指示
Amazon S3	<ul style="list-style-type: none"> • 按照 AWS CLI 和 AWS CloudShell 範例 所述，在環境中安裝和設定 AWS CLI 或 AWS CloudShell。 • 若要將來源碼上傳至儲存貯體，請在環境中使用 AWS CLI 或 AWS CloudShell 來執行 aws s3 cp 命令。(針對 AWS CloudShell，您可以從命令中移除 aws。)
GitHub	您可以使用 「Git」面板 介面複製託管的儲存庫 GitHub 並與之互動。

將環境連線至儲存庫後，每當您將原始程式碼變更從 AWS Cloud9 IDE 推送至儲存庫時，都會透過相關管道 CodePipeline 自動傳送這些變更，以進行建置、測試和部署。您會在本主題稍後建立相關管道。

步驟 3：準備使用 AWS CodePipeline

在此步驟中，您要將特定 AWS 受管政策連接至您在 [團隊設定](#) 中建立或識別的 IAM 群組。這可讓群組的使用者在 CodePipeline 中開始建立和處理管道。

如果您 CodePipeline 之前曾使用過，請跳至 [中的步驟 4：建立管道 AWS CodePipeline](#)。

操作此步驟時，請遵循 AWS CodePipeline 使用者指南中的 [步驟 3：使用 IAM 受管政策將 AWS CodePipeline 許可指派給 IAM 使用者](#) 相關說明，然後返回本主題。

步驟 4：在 AWS CodePipeline 中建立管道

在此步驟中，您會在中建立一個管道，CodePipeline 該管道使用您在本主題稍早建立或識別的存放庫。

操作此步驟時，請遵循 AWS CodePipeline 使用者指南中的 [在 AWS CodePipeline 中建立管道](#) 相關說明。

建立管道之後，CodePipeline 會透過要建置、測試和部署的管道傳送儲存庫中目前版本的原始程式碼。然後，每當您將原始程式碼變更從 AWS Cloud9 IDE 推送至儲存庫時，都 CodePipeline 會自動透過管線傳送這些變更，以進行建置、測試和部署。

若要檢視管道，請遵循 AWS CodePipeline 使用者指南中的[在 AWS CodePipeline 中檢視管道詳細資訊和歷史記錄](#)相關說明。

與 Amazon 合作 CodeCatalyst

Amazon CodeCatalyst 是軟體開發團隊的雲端協作空間。CodeCatalyst 是使用持續整合/交付 (CI/CD) 工具來工作、協同編寫程式碼以及建置、測試和部署應用程式的整合場所。您可以透過將您的空間連接 AWS 帳戶到您的 CodeCatalyst 空間，將 AWS 資源與專案連接起來。您還可以使用 CodeCatalyst 以快速自信的方式交付軟件。如需詳細資訊 CodeCatalyst，請參閱[什麼是 Amazon CodeCatalyst？](#) 在 Amazon CodeCatalyst 指南。

開發環境是基於雲的開發環境，您可以使用它來處理存儲在項目源存儲庫中的代碼。CodeCatalyst 您可以在 CodeCatalyst。然後，您可以在受支援的整合式開發環境 (IDE) 下，在專案特定 CodeCatalyst 的程式碼中處理程式碼。或者，建立空白的 Dev 環境，從協力廠商存放庫複製程式碼，以使用支援的 IDE。

用於在 CodeCatalyst 控制台中訪問您的開發環境的 AWS Cloud9 IDE 與運行的 AWS Cloud9 IDE 不同 AWS。在 CodeCatalyst AWS Cloud9 IDE 中，您會自動登入，CodeCatalyst 並且可以使用 IDE 中的 aws- Explorer 選項存取服務。如需有關「AWS 工具組」的詳細資訊，請參閱 AWS Cloud9 指南 AWS Cloud9 中的[「AWS 工具組」](#)。

主題

- [開始使用](#)
- [在 Amazon 中複製 AWS Cloud9 代碼資源 CodeCatalyst](#)
- [使用複製工具](#)
- [有關複製程序的常見問題](#)
- [Amazon 的開發環境 CodeCatalyst](#)

開始使用

本節提供如何開始使用的概觀 CodeCatalyst。本節中的主題涵蓋如何 AWS Cloud9 在 Amazon 中使用以 CodeCatalyst 及如何在中複製 AWS Cloud9 環境 CodeCatalyst。稍後的主題也會詳細說明如何建立 CodeCatalyst 開發環境，以及如何使用 AWS Cloud9 IDE 存取您的開發環境。

AWS 工具組是 IDE 專用軟體開發套件 (SDK)，可讓您快速存取 AWS 雲端帳戶、服務和資源。透過 AWS 工具組中的 CodeCatalyst 帳戶，您可以在方便的介面中檢視、編輯和管理您的 CodeCatalyst

開發環境、空間和專案。若要深入瞭解可透過 AWS Toolkit 取得的 AWS 雲端 服務和功能，請參閱[什麼是？ AWS Toolkit for Visual Studio Code](#)，[AWS 工具包 AWS Cloud9](#)，[什麼是 AWS Toolkit for JetBrains？](#)。[什麼是指 AWS Toolkit for JetBrains](#)。

若要 CodeCatalyst 搭配 AWS Cloud9 IDE 使用，您必須擁有在 CodeCatalyst 主控台中建立的現有 Space、專案和開發環境。

Note

請勿在 AWS Cloud9 IDE 的檔案系統內的相同名稱的資料夾中建立名為 CodeCatalyst project 的子資料夾。如果這樣做，您將無法訪問此目錄中的任何文件。此問題會影響檔案路徑/專案/專案。/test/專案和/專案 /測試/ 專案之類的檔案路徑不受這個問題影響。這是已知問題，只會影響 AWS Cloud9 IDE 檔案總管。

Note

目前無法使用 AWS Cloud9 IDE 的檔案系統，在相同名稱的資料夾中建立名為 CodeCatalyst project 的子資料夾。您將無法從 AWS Cloud9 IDE 檔案總管存取此目錄中的任何檔案，但是您可以使用命令列存取這些檔案。請使用其他資料夾名稱。此問題只會影響檔案路徑 /projects/projects，其他檔案路徑如 /test/projects 和 /projects/test/projects 應能正常執行。這是已知問題，只會影響 AWS Cloud9 IDE 檔案總管。

在 Amazon 中複製 AWS Cloud9 代碼資源 CodeCatalyst

AWS Cloud9 in CodeCatalyst 提供了與之互動的完全受控體驗 AWS Cloud9。您可以在 Amazon 中手動複製目前的 AWS Cloud9 程式碼資源 CodeCatalyst。該過程將在以下各節中詳細說明。要移動代碼資源並複製它們，請在其中創建一個 Space CodeCatalyst。空間代表您的公司、部門或群組。您需要建立空間來新增專案、成員和您在其中建立的關聯雲端資源 CodeCatalyst。當使用者接受專案的邀請時，CodeCatalyst 會自動將其新增至空間。具有 Space 管理員角色的使用者可以管理空間。

在此空間中，您可以建立專案並新增來源儲存庫。專案是支援開發團隊和工作 CodeCatalyst 的協同合作空間。建立專案後，您可以新增、更新或移除資源。您也可以自訂專案儀表板，並監控團隊工作進度。您可以在一個空間內有多個專案。您新增的來源儲存庫數量取決於您在 AWS Cloud9 環境中已經使用的儲存庫數目。建立此專案並新增適用的來源儲存庫之後，您可能需要返回 AWS Cloud9 環境，並將環境資料複製到中的這些新儲存庫 CodeCatalyst。你做什麼取決於你所擁有的源儲存庫的類型 AWS Cloud9。

建立空間、專案和來源儲存庫之後，您可以在與開發環境 AWS Cloud9 搭配 CodeCatalyst 使用中啟動環境。開發環境是基於雲的開發環境。您可以在 CodeCatalyst 使用 Dev 環境來處理儲存在專案原始碼儲存庫中的程式碼。您也可以在中建立開發環境，CodeCatalyst 以便在具有支援的整合式開發環境 (IDE) 的專案特定開發環境中處理程式碼。

您也可以將目前的 AWS Cloud9 程式碼資源複製到 CodeCatalyst 使用複製工具。這是您在 AWS Cloud9 環境中下載並執行的工具。如果您已經註冊並建立了空間，該工具會自動在 CodeCatalyst 此空間內建立專案，並將您的程式碼資源複製到 CodeCatalyst 類似於手動複製程序。這取決於您所擁有的源儲存庫的類型 AWS Cloud9。例如，如果您有 GitHub 存放庫，您仍然需要使用 CodeCatalyst 控制台中的 GitHub 擴展程序複製這些儲存庫。

- [步驟 1. 註冊 Amazon CodeCatalyst 並創建一個空間](#)
- [步驟 2. 在您的 Space 中建立專案](#)
- [步驟 3. 在您的項目中創建源儲存庫](#)
- [步驟 4. 將您的 AWS Cloud9 代碼資源複製到源儲存庫 CodeCatalyst](#)
- [步驟 5. 在 CodeCatalyst 使用中建立開發環境 AWS Cloud9](#)

步驟 1. 註冊 Amazon CodeCatalyst 並創建一個空間

您可以註冊 Amazon，CodeCatalyst 而無需邀請到現有空間或項目。當您註冊時，您會建立空間和專案。您可以輸入用於的現有 AWS 帳戶 ID AWS Cloud9。這同樣 AWS 帳戶 可以用於計費目的。如需如何尋找 AWS 帳戶 ID 的詳細資訊，請參閱[您的 AWS 帳戶 ID 及其別名](#)。請按照此程序註冊您的 Amazon 個 CodeCatalyst 人檔案、建立空間，並為您的空間新增帳戶。

註冊為新使用者

1. 開啟 [CodeCatalyst 主控台](#)。
2. 在歡迎頁面上，選擇 [註冊]。

[建立您的 AWS 產生器 ID] 頁面隨即顯示。您的 AWS 產生器 ID 是您建立用來登入的身分。此 ID 與 AWS 帳戶 ID 不同。要了解有關 AWS 生成器 ID 的更多信息，請參閱 AWS 登錄用戶指南中的[AWS 生成器 ID 和其他 AWS 憑據](#)。

3. 在 [您的電子郵件地址] 中，輸入您要與之關聯的電子郵件地址 CodeCatalyst。然後選擇下一步。
4. 在 [您的名稱] 中，輸入您要在使用 AWS Builder ID 的應用程式中顯示的名字和姓氏。

這個名稱是您的 AWS 產生器 ID 設定檔名稱。如果需要，您可以稍後更改名稱。

選擇下一步。電子郵件驗證頁面隨即出現。驗證碼會傳送至您指定的電子郵件。

5. 針對 [驗證碼]，輸入您收到的驗證碼，然後選擇 [驗證]。

如果您在 5 分鐘後仍未收到驗證碼，但在垃圾郵件或垃圾郵件資料夾中找不到驗證碼，請選擇「重新傳送驗證碼」。

6. 驗證碼後，輸入密碼並選擇 [確認密碼]。

勾選確認您已閱讀並確認「AWS 客戶協議」與「AWS 服務條款」的核取方塊，然後選擇「建立我的設定檔」。

7. 在 [建立別名] 頁面上，輸入要用於的別名 CodeCatalyst。其他使用 CodeCatalyst 者會在註解和提取要求中使用此別名來 @mention 您。您的個 CodeCatalyst 人資料將包含您的 AWS Builder ID 中的全名和 CodeCatalyst 別名。您無法變更 CodeCatalyst 別名。

您的全名和別名將顯示在中的不同區域 CodeCatalyst。例如，您的設定檔名稱會顯示在活動摘要中，但專案成員會使用您的別名來 @mention 您。

選擇 [建立別名]。頁面會更新以顯示 [建立您的空間] 區段。

8. 針對空間名稱，輸入您的空間名稱，然後選擇下一步。

您無法變更此名稱。

9. 對於 AWS 帳戶 ID，請連結您要連線到空間的帳戶的十二位數 ID。

在 AWS 帳戶 驗證令牌中，複製生成的令牌 ID。系統會自動為您複製權杖。但是，您可能希望在批准 AWS 連接請求時將其存儲。

10. 選擇驗證於 AWS。

11. 驗證 Amazon CodeCatalyst 空間頁面會在中開啟 AWS Management Console。

這是 Amazon CodeCatalyst 空間頁面。您可能需要登錄才能訪問該頁面。

若要存取頁面，請登入中的 Amazon CodeCatalyst 空間 [AWS Management Console](#)。

中的驗證權杖欄 AWS Management Console 位會自動填入中產生的權杖 CodeCatalyst。

12. 選擇驗證空間。

帳戶驗證成功訊息會顯示，顯示帳戶已新增至空間。

依預設，您將使用 CodeCatalyst 免費方案。如果要變更，請選擇若要啟用標準層或為此空間新增 IAM 角色，請檢視空間詳細資料。

如需有關 CodeCatalyst 定價層的詳細資訊，請參閱 [Amazon CodeCatalyst -定價](#)。

CodeCatalyst 空間詳細資訊頁面會在中開啟 AWS Management Console。這是 Amazon CodeCatalyst 空間頁面。您可能需要登錄才能訪問該頁面。

13. 選擇去 [Amazon CodeCatalyst](#)。
14. 在中的建立頁面上 CodeCatalyst，選擇建立空間。

建立您的空間時，會顯示狀態訊息。建立空間後，會 CodeCatalyst 開啟您空間的頁面。檢視預設為 [專案] 索引標籤。

Note

如果顯示權限錯誤或橫幅，請重新整理頁面並嘗試再次檢視頁面。

註冊 CodeCatalyst 並建立空間後，複寫程序的下一個步驟是在此空間內建立專案。

步驟 2. 在您的 Space 中建立專案

下列步驟概述如何在上一個步驟中建立的空間內建立空白專案。透過此專案，您可以稍後手動新增所需的資源。在建立專案之前，您必須具有 Space 管理員角色，並且必須加入要在其中建立專案的空間。當您建立空間時，CodeCatalyst 會自動為您指派 Space 管理員角色。Space 管理員角色是最強大的角色 CodeCatalyst。如需有關此角色及其權限的詳細資訊，請參閱 [Space 系統管理員角色](#)。

建立空專案的步驟

1. 導覽至您要在其中建立專案的空間。
2. 在空間儀表板上，選擇建立專案。
3. 選擇「從頭開始」。
4. 在「為您的專案命名」下，輸入您要指派給專案的名稱。名稱在您的空間中必須是唯一的。
5. 選擇建立專案。

建立專案之後，複製程序的下一個步驟是建立一或多個來源儲存庫。

步驟 3. 在您的項目中創建源存儲庫

在剛剛創建的項目中，您需要創建一個源存儲庫。此儲存庫包含一個 README.md 檔案，您可以隨時編輯或刪除該檔案。根據您在建立來源存放庫時所做的選擇，它也可能包含 .gitignore 檔案。

若要建立來源儲存庫

1. 開啟 [CodeCatalyst 主控台](#)。
2. 導航到您的項目。
3. 在瀏覽窗格中，選擇 [程式碼]，然後選擇 [原始碼儲存庫]。
4. 選擇新增儲存庫，然後選擇建立儲存庫。
5. 在存放庫名稱中，提供存放庫的名稱。

存放庫名稱在專案中必須是唯一的。如需有關儲存 [庫名稱需求的詳細資訊](#)，請參閱 CodeCatalyst。

6. (選擇性) 在說明中，新增存放庫的說明，以協助專案中的其他使用者瞭解存放庫的用途。
7. (選擇性) 針對您計劃推送的程式碼類型新增 .gitignore 檔案。
8. 選擇建立。

Note

CodeCatalyst 創建 README.md 文件時將文件添加到存儲庫中。CodeCatalyst 還會在名為 main 的默認分支中為存儲庫創建初始提交。您可以編輯或刪除 README.md 檔案，但無法變更或刪除預設分支。

9. 若要取得來源儲存庫複製 URL 和 PAT，請選擇複製儲存庫。
10. 若要複製每個 HTTPS 複製網址和 PAT，請選擇「複製」。然後，將克隆 URL 存儲在可以檢索它的位置。

複製 URL 和 PAT 將在步驟 4 中使用，並參照
為 CODECATALYST_SOURCE_REPO_CLONE_URL 和 CODECATALYST_PAT。

在專案中建立來源儲存庫之後，請將資 AWS Cloud9 料複製到此來源儲存庫。

步驟 4. 將您的 AWS Cloud9 代碼資源複製到源存儲庫 CodeCatalyst

您 AWS Cloud9 環境中所擁有的來源儲存庫類型會決定您遵循的複寫方法，以便將程式碼資源放入您建立的 CodeCatalyst 來源儲存庫。選項如下：

- [使用 GitHub 儲存庫 AWS Cloud9](#)
- [使用非 GitHub，例如 GitLab 或比特桶，存儲庫 AWS Cloud9](#)
- 在中 [使用空的儲存庫 AWS Cloud9](#)。此選項表示您不會在中使用任何來源儲存庫 AWS Cloud9。

使用GitHub儲存庫 CodeCatalyst

使用GitHub儲存庫擴充功能，您可以 AWS Cloud9 在 Amazon CodeCatalyst 專案中使用連結的GitHub儲存庫。下列步驟概述如何從 CodeCatalyst 目錄安裝GitHub擴充功能。這些步驟還顯示了如何將現有GitHub帳戶連接到您的 CodeCatalyst 空間，並將GitHub儲存庫鏈接到 CodeCatalyst 項目。

此方法的第一個步驟是從 CodeCatalyst 目錄安裝GitHub儲存庫擴充功能。若要安裝擴充功能，請執行下列步驟。

Important

作為安裝和配置Github儲存庫擴展的一部分，您必須在GitHub帳戶中安裝擴展程序。若要這麼做，您必須是GitHub帳戶管理員和 CodeCatalyst空間管理員。

步驟 1. 若要從 CodeCatalyst 目錄安裝擴充功能

1. 開啟 [CodeCatalyst 主控台](#)。
2. 導覽至您的空間。

Tip

如果您屬於多個空間，則可以選擇要在頂部導覽列中檢視的空間。

3. 選擇搜尋列旁頂端功能表列中的「目錄」圖示，以導覽至目錄。CodeCatalyst 您可以搜索GitHub儲存庫或根據類別過濾擴展名。
4. (選擇性) 若要查看有關擴充功能的詳細資訊，例如與其相關聯的權限，請選擇GitHub儲存庫擴充功能名稱。
5. 選擇 Install (安裝)。檢閱擴充功能所需的權限，如果您要繼續，請再次選擇 [安裝]。

安裝GitHub儲存庫擴充功能之後，您會前往GitHub儲存庫擴充功能詳細資料頁面，您可以在其中檢視和管理連線的GitHub帳戶和連結的GitHub儲存庫。

安裝GitHub儲存庫擴充功能後，請將您的GitHub帳戶連線至您的 CodeCatalyst 空間。要連接您的GitHub帳戶，請執行以下步驟。

步驟 2. 將您的GitHub帳戶連接到 CodeCatalyst

1. 在 [已連線的Github帳戶] 索引標籤中，選擇 [Connect GitHub 帳戶] 以移至其外部網站GitHub。

2. 使用您的GitHub的GitHub憑據登錄到您的帳戶，然後選擇要安裝 Amazon 的帳戶 CodeCatalyst。
3. 選擇是否允許存 CodeCatalyst 取所有目前和 future 的儲存庫。或者，您也可以選擇要在其中使用的特定GitHub存放庫 CodeCatalyst。預設選項是GitHub空間中的所有GitHub儲存庫。
4. 檢閱授予的權限 CodeCatalyst，然後選擇 [安裝]。

將GitHub帳戶連接到後 CodeCatalyst，您可以在GitHub儲存庫擴展詳細信息頁面的GitHub帳戶選項卡中查看已連接的帳戶。

在中使用儲存GitHub庫的最後一步 CodeCatalyst 是將存放庫連結到您要使用它的 CodeCatalyst 專案。若要將存GitHub放庫連結至 CodeCatalyst 專案，請執行整體程序步驟 3 所述的下列步驟：

步驟 3。若要從儲GitHub存庫擴充功能詳細資訊頁面將存放GitHub庫連結至 CodeCatalyst專案

1. 在「連結的GitHub儲存庫」頁籤中，選擇「連結儲GitHub存庫」。
2. 對於GitHub帳戶，選取包含您要連結之存放庫的GitHub帳戶。
3. 對於GitHub存放庫，請選取您要連結至 CodeCatalyst 專案的存放庫。
4. 對於CodeCatalyst 專案，請選取您要連結GitHub存放庫的 CodeCatalyst 專案。
5. 選擇 Link (連結)。

您的 CodeCatalyst 儲存庫現在應該具有更新的文件和剛剛推送的提交。您現在可以從此分支建立開發環境，並使用 AWS Cloud9。如需有關開發環境的詳細資訊，請參閱 [CodeCatalyst](#)。

您現在可以從此分支建立開發環境，並使用 AWS Cloud9。執行此操作的步驟在 [步驟 5：使用 in 創建開發環境 AWS Cloud9 中概述 CodeCatalyst](#)

在中使用非GitHub儲存庫 CodeCatalyst

您需要在 Amazon 中建立個人存取權杖 (PAT)，CodeCatalyst然後再 AWS Cloud9 使用非GitHub儲存庫複製環境。以下部分概述了如何創建此令牌。

在 Amazon 中創建個人訪問令牌 CodeCatalyst

您可以在具有Git用戶端或整合式開發環境 (IDE) 的本機電腦上存取您在專案中建立的來源存放庫。若要這麼做，您必須輸入應用程式特定的密碼。您可以建立個人存取權杖 (PAT) 以用於此目的。您建立的個人存取權杖 (PAT) 會在中的所有空間和專案中 CodeCatalyst與您的使用者身分相關聯。您可以查看您創建的 PAT 的名稱和到期日，也可以刪除不再需要的 PAT。您只能在建立 PAT 密碼時複製它。

若要建立個人存取權杖 (PAT)

1. [請在以下位置開啟 CodeCatalyst 主控台。](https://codecatalyst.aws/) <https://codecatalyst.aws/>
2. 在頂端選單列中，選擇您的設定檔徽章，然後選擇 [我的設定]。

Tip

您也可以找到您的使用者設定檔。若要這麼做，請在專案或空間的成員頁面上，從成員清單中選擇您的名稱。

3. 在「個人存取權杖」下，選擇「建立」
4. 在 PAT 名稱中，輸入個人存取權杖 (PAT) 的描述性名稱。
5. 在到期日中，保留預設日期，或選擇行事曆圖示以選取自訂日期。到期日預設為從目前日期算起 1 年。
6. 選擇建立。

Tip

當您為來源儲存庫選擇複製儲存庫時，也可以建立此權杖。

7. 若要複製 PAT 密碼，請選擇「複製」。將 PAT 秘密存儲在您可以檢索它的位置。

Important

PAT 秘密僅顯示一次。關閉窗口後無法檢索它。如果您未將 PAT 密碼儲存在安全的位置，您可以建立另一個密碼。

建立來源儲存庫的 PAT 之後，CodeCatalyst 透過在 AWS Cloud9 環境中新增遠端存放庫並將資料推送至此儲存庫，將資料從您的 AWS Cloud9 環境複製到中，如下節所述。

在您的 AWS Cloud9 環境中新增遠端儲存庫

假設您正在運行不是儲存庫的 GitHub 儲存庫。您可以在 AWS Cloud9 環境中新增遠端儲存庫，並將資料推送至中的來源儲存庫 CodeCatalyst。若要完成此程序，請執行下列命令。

在 AWS Cloud9 IDE 中，新增指向您在複寫程序步驟 3 中建立的來源存放庫的遠端存放庫 CodeCatalyst。將命令 `CODECATALYST_SOURCE_REPO_CLONE_URL` 中的複製 URL 取代為您在步驟 [3](#) 的步驟 10 中儲存的複製 URL。在您的項目中創建源儲存庫。

```
git remote add codecatalyst CODECATALYST_SOURCE_REPO_CLONE_URL
```

使用以下命令將新分支推送到源存儲庫。當系統提示您輸入密碼時，請CODECATALYST_PAT使用您儲存在步驟 3 步驟 10 中的密碼。在您的項目中創建源存儲庫：

```
git checkout -b replication && git push codecatalyst replication
```

以下是預期命令運行輸出的示例：

```
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
```

您在中建立的來源儲存庫中可以使用此分支 CodeCatalyst。您可以從此分支創建開發環境並使用 AWS Cloud9。如需有關開發環境的詳細資訊，請參閱 [CodeCatalyst](#)。

您現在可以從此分支建立開發環境，並使用 AWS Cloud9。執行此操作的步驟在 [步驟 5：使用 in 創建開發環境 AWS Cloud9 中](#) 概述 CodeCatalyst

在中使用空存儲庫 AWS Cloud9

首先在 Amazon 中創建個人訪問令牌 (PAT)，CodeCatalyst 然後才能 AWS Cloud9 使用空存儲庫複製環境。以下部分概述了如何創建此令牌。

在 Amazon 中創建個人訪問令牌 CodeCatalyst

您可以在具有 Git 用戶端或整合式開發環境 (IDE) 的本機電腦上存取您在專案中建立的來源存放庫。若要這麼做，您必須輸入應用程式特定的密碼。您可以建立個人存取權杖 (PAT) 以用於此目的。您建立的個人存取權杖 (PAT) 會在中的所有空間和專案中 CodeCatalyst 與您的使用者身分相關聯。您可以查看您創建的 PAT 的名稱和到期日，也可以刪除不再需要的 PAT。您只能在建立 PAT 密碼時複製它。

若要建立個人存取權杖 (PAT)

1. 開啟主 CodeCatalyst 控制台，[網址為 https://codecatalyst.aws/](https://codecatalyst.aws/)。
2. 在頂端選單列中，選擇您的設定檔徽章，然後選擇 [我的設定]。

Tip

您也可以找到您的使用者設定檔。若要這麼做，請在專案或空間的成員頁面上，從成員清單中選擇您的名稱。

3. 在「個人存取權杖」下，選擇「建立」。
4. 在 PAT 名稱中，輸入個人存取權杖 (PAT) 的描述性名稱。
5. 在到期日中，保留預設日期，或選擇行事曆圖示以選取自訂日期。到期日預設為從目前日期算起 1 年。
6. 選擇建立。

Tip

當您為來源儲存庫選擇複製儲存庫時，也可以建立此權杖。

7. 若要複製 PAT 密碼，請選擇「複製」。將 PAT 秘密存儲在您可以檢索它的位置。

Important

PAT 秘密僅顯示一次。關閉窗口後無法檢索它。如果您未將 PAT 密碼儲存在安全的位置，您可以建立另一個密碼。

建立來源儲存庫的 PAT 之後，請在您的 AWS Cloud9 環境中啟動空白儲存庫，並指向您在中建立的來源儲存庫 (如下節所述) CodeCatalyst，將您的資料從您的 AWS Cloud9 環境複製到。CodeCatalyst

在中啟動一個空存儲庫 AWS Cloud9

如果您沒有在中設定任何來源儲存庫 AWS Cloud9，請在中啟動一個空的儲存庫 AWS Cloud9。此外，請指向您在其中建立的來源存放庫 CodeCatalyst，然後新增並推送您要複製的檔案 Git。執行下列步驟並執行下列命令以將檔 AWS Cloud9 案複製到 CodeCatalyst。

1. 在您的 AWS Cloud9 環境中，執行下列命令來啟動空的儲存庫：

```
git init -b main
```

然後，您會看到類似的輸出，如下所示：

```
Initialized empty Git repository in /home/ec2-user/environment/.git/
```

2. 從複製來源儲存庫 URL CodeCatalyst。瀏覽至您在 CodeCatalyst 主控台中建立的 CodeCatalyst 專案，然後在導覽窗格中選擇 [程式碼]，然後選擇 [原始碼儲存庫]。
3. 從您想要的來源儲存區域清單中選擇儲存區域，然後選擇複製儲存區域來複製 URL。
4. 使用您克隆的 URL 添加 CodeCatalyst 儲存庫，並將已在空儲存庫中的內容推送到：CodeCatalyst

```
git remote add origin [...]  
git push origin --force
```

5. 新增您要複製的檔案。如果要複製環境目錄中的所有檔案，請執行 `git add -A`：

```
git add -A .  
git commit -m "replicate"
```

6. 合併兩個不相關的歷史記錄。解決合併衝突（如果發生）：

```
git merge origin/main --allow-unrelated-histories
```

7. 執行下列命令，將變更推回中 CodeCatalyst 的來源儲存庫。當系統提示您輸入密碼時，請輸入您在 [步驟 3 的步驟 10](#) 中產生的個人存取權杖 (CODECATALYST_PAT)。在您的項目中創建源儲存庫：

```
Admin:~/environment (main) $ git push origin main  
Password for 'https://222581768915@git.us-west-2.codecatalyst.aws/v1/  
MySpace222581768915/Replication/Replication':
```

完成此程序後，您的 CodeCatalyst 儲存庫會有您剛才推送的更新檔案和提交。您現在可以從此分支建立開發環境，並使用 AWS Cloud9。以下部分概述了執行此操作的步驟。

步驟 5：使用 AWS Cloud9 in 建立開發環境 CodeCatalyst

下列程序概述如何在 CodeCatalyst 使用中建立開發環境，以 AWS Cloud9 及您剛才複製的資料。

若要使用建立開發環境 AWS Cloud9

1. 開啟主 CodeCatalyst 控制台，網址為 <https://codecatalyst.aws/>。
2. 導航到要創建開發環境的項目。
3. 在功能窗格中，選擇 [概觀]，然後瀏覽至 [我的開發環境] 區段。
4. 選擇 [建立開發環境]。
5. AWS Cloud9 從下拉式功能表中選擇。
6. 選擇複製儲存庫。

Note

目前 CodeCatalyst 不支援複製第三方儲存庫，但您可以建立開發環境，並從您選擇的 IDE 將第三方存放庫複製到其中。

7. 執行以下任意一項：
 - a. 選擇要複製的存放庫，選擇在現有分支中工作，然後從現有分支下拉式功能表中選擇分支。
 - b. 選擇要複製的存放庫，選擇在新分支中工作，在「分支名稱」欄位中輸入分支名稱，然後從「建立分支來源」下拉式功能表中選擇要從中建立新分支的分支。
8. 選擇性地新增開發環境的別名。
9. 您也可以選擇 [開發環境] 設定編輯按鈕，以編輯開發環境的運算、儲存或逾時設定。
10. 選擇建立。建立您的開發環境時，開發環境狀態欄會顯示 [開始]，而且建立開發環境後，狀態欄會顯示 [執行中]。

使用複製工具

AWS Cloud9 in CodeCatalyst 提供了與之互動的完全受控體驗 AWS Cloud9。為了讓客戶能夠嘗試使用 AWS Cloud9 in CodeCatalyst，我們建立了複製工具。在您的 AWS Cloud9 環境中複製並執行指令碼之後，請依照提示執行指令碼，並將程式碼資源從複製 AWS Cloud9 到 CodeCatalyst。如需複製工具和程序的詳細資訊，請參閱下面概述的[複製程序常見問題集](#)。

Note

此複製程序不會對您現有的 AWS Cloud9 環境造成任何影響。複製程序完成後，您可以刪除開發環境、來源儲存庫、專案和空間，而且不會影響您的 AWS Cloud9 環境。此工具只會將您

的代碼資源複製到 AWS Cloud9 中 CodeCatalyst，不會刪除或配置您現有的 AWS Cloud9 環境。此複製工具已發行至初始選取的 AWS 帳戶群組。因此，它可能不會出現在某些 AWS 帳戶中。

Note

建議您在下載工具之前註冊 Amazon CodeCatalyst 並創建一個空間。如需註冊的相關資訊 CodeCatalyst，請參閱[註冊 Amazon CodeCatalyst 和建立空間](#)。

AWS Cloud9 在 Amazon 上使用的好處 CodeCatalyst

下節概述了 AWS Cloud9 在上使用時將體驗到的一些效能優勢和增強功能 CodeCatalyst：

- CodeCatalyst 提供整合式體驗，可讓您使用完全受控的開發環境，從單一位置管理整個軟體開發生命週期。
- 在推出時增強了 Amazon EBS 磁碟區大小選項。
- Support 暫時性環境，以及隨需擴充開發環境運算的能力。
- 自定義 AMI 支持，可通過自定義圖像的規格獲得。
- Devfile 支持，使您能夠將配置描述為代碼。

CodeCatalyst 使用複製工具複製您的 AWS Cloud9 程式碼資源

下列程序詳細說明如何複製和執行複製工具以完成複製程序。

1. 複製下面的腳本，並確保您在 AWS Cloud9 環境中運行它：

```
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. [選擇性] 複製工具會使用您的 AWS 帳戶 ID 進行遙測。這樣做的目的是幫助我們更好地識別您在使用該工具時可能遇到的任何問題。我們發出 tool starts、tool failstool completes successfully 和 tool creates a Dev Environment for the user 的遙測事件。tool is cancelled by user 如果您想要使用 [複製工具停用遙測](#)，請參閱下面的 [停用複製工具的遙測](#)。

3. 在您的 AWS Cloud9 環境中複製並執行複製工具之後，您必須導覽至瀏覽器中 AWS 帳戶 的存取 URL，並在 10 分鐘內按一下 [允許]，以連結您的 AWS 產生器 ID。請確保您只打開一次鏈接，如果您多次打開鏈接，則會導致錯誤，您需要重新開始。如需有關 [AWS Builder ID 的詳細資訊](#)，請參閱 [登入使用者指南中的使用 AWS Builder ID AWS](#) 登入。這將授予複製工具對您的代碼資源的訪問權限，以便將其複製到中 CodeCatalyst。
4. 選擇您要使用的空間。如果您只有一個空間，則會選取該空間。如需有關空間的詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南 CodeCatalyst 中的 [空間](#)。
5. 選擇是否要在中複寫程式碼，CodeCatalyst 或使用新的開發環境試用程式碼。我們建議您直接在中 CodeCatalyst 複製您的程式碼。如需有關開發環境的詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南 CodeCatalyst 中的 [開發環境](#)。
6. 輸入專案名稱，或按 Enter 以使用提供的預設名稱。
7. 出現提示時，選取您要如何將檔案複製到中的新來源儲存庫 CodeCatalyst。您可以選擇將根資料夾推送至單一儲 CodeCatalyst 存庫，或將子資料夾推送至不同的儲存 CodeCatalyst 庫。
8. 工具完成後，透過終端機訊息中提供的 URL 導覽至 CodeCatalyst 主控台內的專案，以存取中的程式碼資源 CodeCatalyst。

完成此程序後，您的 CodeCatalyst 儲存庫會有您剛才推送的更新檔案和提交。您現在可以從此分支建立開發環境，並使用 AWS Cloud9。

停用複寫工具的遙測

下列步驟概述如何設定環境變數以停用複寫工具的遙測。

1. 在您的 AWS Cloud9 環境中開啟終端機
2. 執行下列其中一個命令：

```
export CLOUD9_REPLICATION_TOOL_TELEMETRY=off
```

或

```
export CLOUD9_REPLICATION_TOOL_TELEMETRY=0
```

3. 一旦您執行上述其中一個命令，就會設定環境變數，並停用複寫工具的遙測。停用遙測之後，您必須再次複製並重新執行複製工具指令碼，才能開始此程序。

複製工具回饋

如果您遇到任何問題，或想要對使用複製工具的經驗提供意見反應，請建立並提交支援案例。如需建立支援案例的相關資訊，請參閱[建立支援案例和案例管理](#)。

AWS Cloud9 和 Amazon 之間的差異 CodeCatalyst

下表概述了 AWS Cloud9 和 (AWS Cloud9 在上) 之間的一些差異 CodeCatalyst。

AWS Cloud9	AWS Cloud9 在 Amazon CodeCatalyst
私人 VPC 非常適合 AWS Cloud9。	目前不支援 AWS Cloud9 在 CodeCatalyst 上使用私有 VPC。
AWS Cloud9 支援預先設定的 AWS 受管理認證。	您必須手動將身份證明配置為 AWS Cloud9 「打開 CodeCatalyst」。
有可能有 30 分鐘到 7 天的間隔，並禁用關機。 AWS Cloud9	有可能有 15 分鐘到 20 小時的 AWS Cloud9 間隔 CodeCatalyst，你不能禁用關機。
AWS Cloud9 支持 Ubuntu 和 AL2 操作系統平台。	AWS Cloud9 上 CodeCatalyst 支持 MDE 通用圖像和自定義圖像，其中可以包括 Ubuntu 和 AL2。如需詳細資訊，請參閱 Amazon CodeCatalyst 使用者指南中的通用 開發檔映像 。
支持上傳和下載 AWS Cloud9	目前不支援 AWS Cloud9 在上傳和下載 CodeCatalyst。使用者必須使用 Amazon S3 儲存貯體上傳和下載。
協同合作可用於 AWS Cloud9	「合作」目前不適用 AWS Cloud9 於 (詳見) CodeCatalyst。

有關複寫程序的常見問題

以下部分旨在回答與複製工具和複製過程相關的一些常見問題解答。

問題：如果我在上複製 AWS Cloud9 環境 CodeCatalyst，我的 AWS Cloud9 環境是否會受到影響？

答：否，您環境的複寫只會複製您的程式碼資源，以 AWS Cloud9 CodeCatalyst 便您繼續工作。您的程式碼資源和環境不 AWS Cloud9 會受到任何影響。

問題：如果我想要復原，我的 AWS Cloud9 環境會受到影響嗎？

答：否，您可以刪除 CodeCatalyst 開發環境、原始碼儲存庫、專案和空間，而且不會影響您的 AWS Cloud9 環境。

問題：新地點是否符合 HIPAA、SOC 等標準？

答：開發環境自 CodeCatalyst 前不符合這些標準。這些標準的遵守在路線圖上。

問題：我的程式碼資源會去哪裡？

回答：您的程式碼資源會複製到中的專案中的原始碼儲存庫 CodeCatalyst。

問題：我的使用會受到限制嗎？

答：在複寫程序中，您將建立免費方案內有 16 GB 的開發環境。這表示您最多可以擁有 4 個開發環境。如需有關定價、儲存和可用不同層級的詳細資訊，請參閱 [Amazon CodeCatalyst -定價](#)。

問題：我的計算會去哪裡？

答：您現有的運算不會有任何變更。它將保持原樣。

問題：我可以在 CodeCatalyst 中使用現有的 AWS 帳戶憑證嗎？它們會自動轉移嗎？

回答：您可以在中手動設定您的 AWS 帳戶憑證 CodeCatalyst。它們不會自動傳輸。

問題：這需要多少錢？

答：您可以免費開始使用 CodeCatalyst。如需定價和可用不同層級的詳細資訊，請參閱 [Amazon CodeCatalyst -定價](#)。

問題：資料複製程序和資料儲存是否 CodeCatalyst 安全？

答：是的，我們將使用帶有 https 的 git push 來複製代碼資源並將數據 CodeCatalyst 安全地存儲在服務中。所有數據都在傳輸和靜態中進行加密。如需中資料保護的詳細資訊 CodeCatalyst，請參閱 [Amazon CodeCatalyst 使用者指南 CodeCatalyst 中的 Amazon 中的資料保護](#)。

問題：我應該選擇哪種複寫方法？

答：複寫工具提供兩種方法；您可以將程式碼資源從複製 AWS Cloud9 到單一 CodeCatalyst CodeCatalyst 來源儲存庫，或者每個子資料夾都會轉換成不同的 CodeCatalyst 來源儲存庫。我們建

議使用第一種方法，因為它不需要諸如源儲存庫之類的 CodeCatalyst 概念的先驗知識。這種方法是一個很好的起點，可以探索中的 AWS Cloud9 體驗 CodeCatalyst，同時使用您習慣的類似設置 AWS Cloud9。

當您獨立使用根 AWS Cloud9 環境資料夾下的子資料夾時，最好選擇第二個選項。使用此方法，將不會複寫根資料夾下的任何檔案。如需有關中來源儲存庫的詳細資訊 CodeCatalyst，請參閱 Amazon CodeCatalyst 使用者指南 CodeCatalyst [中的來源儲存庫](#)。

問題：在複製過程中生成的個人訪問令牌是什麼？為什麼我需要它？如果我失去了它，我可以再次生成它嗎？

答：個人存取權杖與中的使用者身分相關聯 CodeCatalyst。當您使用 git 將本地更改推送到 CodeCatalyst 源儲存庫時，它是必需的密碼。有關令牌以及如何生成令牌的更多信息，請參閱 Amazon CodeCatalyst 用戶指南中的 [在 Amazon CodeCatalyst 中管理個人訪問令牌](#)。

問題：如果在複寫程序期間發生錯誤，會發生什麼情況？

回答：如果使用複製工具時發生錯誤，您應該先嘗試再次使用此工具。如果錯誤與來源儲存庫有關，您可以在複寫程式碼資源後，手動將程式碼資 CodeCatalyst 源推送至來源儲存庫。這應該工作，因為本地儲存庫已經配置為與 CodeCatalyst 上游一起工作。如果問題仍然存在，請創建並提交支持案例。如需建立支援案例的相關資訊，請參閱 [建立支援案例和案例管理](#)。

問題：為什麼我需要使用 AWS BuilderID 對複製工具進行身份驗證並授予權限？

答：在複寫過程中，複製工具需要代表使用者讀取和寫入多個資源 (專案、開發環境、來源儲存庫)，CodeCatalyst 並複製本機內容，因此需要您的許可才能執行此操作。

問題：如果我搬到，延遲會有變化 CodeCatalyst 嗎？

答：根據您正在執行的操作，您可能會看到延遲減少。這是因為 CodeCatalyst 服務器被託管在 PDX 區域中。

問題：我安裝的所有軟體都會轉移嗎？

答：不會，只會傳輸您的程式碼資源。二進制文件，配置和已安裝的軟件將不會被傳輸。

Amazon 的開發環境 CodeCatalyst

以下各節概述了如何 CodeCatalyst 使用 AWS Cloud9 IDE 來創建和管理您的開發環境。

- [建立開發環境](#)

- [開啟開發環境設定](#)
- [恢復開發環境](#)
- [刪除開發環境](#)
- [編輯開發環境的儲存庫 devfile](#)
- [複製儲存庫](#)
- [疑難排解開發環境](#)

建立開發環境

您可以透過多種方式建立開發環境：

- CodeCatalyst 使用摘要、開發環境或 CodeCatalyst 來源存放庫頁面中的來源儲存庫建立開發環境。
- 從開發環境中創建一個未連接到源儲存庫的空開 CodeCatalyst 發環境。
- 在您選擇的 IDE 中建立開發環境，並將 CodeCatalyst 來源儲存庫複製到開發環境。

您可以為每個分支和儲存庫建立一個開發環境。一個專案可以有多個儲存庫。您的開發環境僅與您的 CodeCatalyst 帳戶相關聯，而且只能由您的 CodeCatalyst 帳戶管理。您可以開啟開發環境，並與任何支援的 IDE 搭配使用。選擇特定的 IDE 之後，您只能使用選擇的 IDE 開啟該開發環境。如果您想要使用不同的 IDE，您可以選取導覽列中的 [開發環境] 並選擇 [編輯]，或建立新的 [開發環境] 來變更 IDE。根據預設，開發環境是使用 2 核心處理器、4 GB RAM 和 16 GB 的持續性儲存裝置來建立。

有關如何在中建立開發環境的詳細資訊 CodeCatalyst，請參閱 Amazon CodeCatalyst 指南中的[建立開發環境](#)。

如需在中建立開發環境的相關資訊和步驟 CodeCatalyst，請參閱 Amazon CodeCatalyst 使用者指南中的[建立開發環境](#)。

Note

您現在可以使用第三方來源儲存庫建立開發環境。如需將第三方來源儲存庫連結至其中專案的相關資訊 CodeCatalyst，請參閱 Amazon CodeCatalyst 使用者指南中的[連結來源儲存庫](#)。

開啟開發環境設定

在 CodeCatalyst 主控台中建立開發環境之後，您可以檢視特定的開發環境設定：

1. 在主 CodeCatalyst 控台中，透過 AWS Cloud9 IDE 導覽至您的開發環境。
2. 從 AWS Cloud9 側邊欄選擇 aws-explorer。
3. 在 [開發人員工具] 導覽窗格中，展開 CodeCatalyst 並選擇 [開啟設定] 以開啟 [開發人員環境設定] 檢視
4. 從 Dev Environment Settings (開發環境設定) 檢視中，下列各章節包含適用於您開發環境的選項：
 - 別名：檢視和變更指派給開發環境的別名。
 - 狀態：查看您當前的開發環境狀態，它被分配到的項目，並停止您的開發環境。
 - Devfile：檢視開發環境的 Devfile 名稱和位置。Devfile 通過選擇在編輯器中打開您的。
 - Compute Settings: (運算設定：) 變更開發環境的 Timeout Length (逾時長度) 長短和預設值。

Note

您無法在建立開發環境之後變更指派給開發環境的儲存空間量。

Note

CodeCatalyst AWS CLI 從終端使用 Amazon 時，您必須確保在運行任何命令之前設置 `AWS_PROFILE = 代碼催化劑`。CodeCatalyst

恢復開發環境

開發環境 `$HOME` 目錄中的所有內容皆會永久儲存。如有必要，您可以停止在開發環境中工作，並在稍後繼續在開發環境中工作。假設開發環境處於閒置狀態超過建立開發環境時在「逾時」欄位中選取的時間量。在此情況下，工作階段會自動停止。

您只能從恢復開發環境 CodeCatalyst。如需有關如何恢復開發環境的詳細資訊，請參閱 Amazon CodeCatalyst 指南中的 [繼續開發環境](#)。

Note

恢復開發環境可能需要幾分鐘的時間。

刪除開發環境

當您完成處理儲存在開發環境中的內容時，您可以刪除該內容。在刪除開發環境之前，請確保您提交並將代碼更改推送到原始源儲存庫。刪除開發環境之後，開發環境的計算和儲存體計費就會結束。

您只能從中的 [開發環境] 頁面刪除開發環境 CodeCatalyst。如需有關如何刪除開發環境的詳細資訊，請參閱 Amazon CodeCatalyst 指南中的 [刪除開發環境](#)。

編輯開發環境devfile的存放庫

若要變更開發環境的組態，請編輯devfile。您可以用devfiles來標準化整個團隊的開發開發環境。您可以在devfile中編輯來源儲存庫的根目錄 CodeCatalyst。或者，您可以在支援devfile的 IDE 中編輯。如果您在支援的 IDE devfile 中編輯，請認可並將變更推送至來源儲存庫，或建立提取要求。如此一來，團隊成員就可以檢閱並核准編devfile輯內容。

Note

您只能將公用容器映像檔包含在devfile。

Note

如果缺少相依性，某些 AWS Cloud9 IDE 功能可能無法在自訂中運作devfile。可能需要額外的努力才能使它們在 Linux x64 以外的其他平台上運行。

若要編輯開發環境的存放庫 devfileAWS Cloud9

1. 在主 CodeCatalyst 控台中，透過 AWS Cloud9 IDE 導覽至您的開發環境。
2. 從 AWS Cloud9 側邊欄中，選擇 AWS 資源管理器。
3. 在開發人員工具導航窗格中，選擇工 CodeCatalyst 具包菜單。
4. 選擇 Open Devfile (開啟 Devfile)。
5. 編輯 devfile，然後儲存檔案。
6. 選擇「原始檔控制」，這是功能表側邊欄中的Git擴充功能。
7. 暫存變更前，在 Message (訊息) 文字欄位中輸入訊息。
8. 若要準備提交，請選擇 Stage All Changes (+) (暫存所有變更 (+)) 圖示。
9. 若要檢視Git指令，請選擇存放庫名稱旁邊的功能表圖示。

10. 選擇 Commit (提交) 和 Push (推送)。
11. 從功能表中選擇「更新開發環境 AWS 工具組」。

選擇 Commit (提交) 和 Push (推送)。更新devfile已儲存，且變更已提交並推送。

Note

假設您要使用自定義啟動的開發環境devfile不起作用。這可能devfile是因為與 AWS Cloud9。若要疑難排解，請檢閱devfile。如果問題仍然存在，請將其刪除並嘗試創建一個新問題。

您也可以透過devfile編輯開發環境的 CodeCatalyst。如需詳細資訊，請參閱 [Amazon CodeCatalyst 指南中的設定您的開發環境](#)。

複製儲存庫

若要有效地處理來源儲存庫中的多個檔案、分支和認可，您可以將來源儲存庫複製到本機電腦。然後，使用用Git戶端或 IDE 進行變更。從中 CodeCatalyst，您可以使用與任何其他Git主機提供者相同的方式使用 AWS Cloud9 IDE Git 延伸模組，也可以使用命令列。若要瞭解如何複製第三方存放庫，請參閱 [初始化或複製Git存放庫](#)。

如需有關從來源儲存庫建立開發環境並使用進行複製的詳細資訊 CodeCatalyst，請參閱 Amazon CodeCatalyst 指南中的 [來源儲存庫概念](#)。

疑難排解開發環境

如果您在使用開發環境時遇到問題，請參閱 [Amazon CodeCatalyst 指南中的開發環境疑難排解問題](#)。

Note

CodeCatalyst AWS CLI 從終端使用 Amazon 時，您必須確保在運行任何命令之前設置 `AWS_PROFILE = 代碼催化劑`。CodeCatalyst

如果您在使用開發環境時遇到問題，請參閱 [Amazon CodeCatalyst 指南中的開發環境疑難排解問題](#)。

在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CDK

AWS CDK Explorer 可讓您操作 [AWS Cloud Development Kit \(AWS CDK\) 應用程式](#) 或應用程式。您可以在 [AWS Cloud Development Kit \(AWS CDK\) 開發人員指南](#) [AWS CDK](#) 中找到有關的詳細資訊。

AWS CDK 應用程式是由稱為 [建構模組](#) 的建構區塊所組成。這些建構區塊包括 AWS CloudFormation 堆疊的定義及其中的 AWS 資源。使用 AWS CDK Explorer，您可以看到在 AWS CDK 樹狀檢視中定義的 [堆疊](#) 和 [資源](#)。您可以在 AWS Cloud9 編輯器的「開發人員工具」窗格中存取此檢視。

本節提供有關如何在 AWS Cloud9 編輯器中存取和使用 AWS CDK。

使用 AWS CDK 應用程式

使用 AWS Cloud9 整合式開發環境 (IDE) 中的 AWS CDK Explorer，以視覺化方式呈現並使用 AWS CDK 應用程式。

先決條件

安裝 AWS CDK 命令列界面。如需說明，請參閱 [AWS Cloud Development Kit \(AWS CDK\) 開發人員指南](#) 中的 [AWS CDK 入門](#)。

Important

您安裝的 AWS CDK 版本必須為 1.17.0 或更新版本。您可以使用 `cdk --version` 命令檢查您正在運行的版本。

視覺化 AWS CDK 應用程式。

使用 AWS Cloud9 IDE 的 AWS CDK Explorer，您可以管理儲存在應用程式 CDK 建構模組中的 [堆疊](#) 和 [資源](#)。AWS CDK Explorer 會使用 `tree.json` 檔案中定義的資訊，在樹狀檢視中顯示您的資源。呼叫您執行 `cdk synth` 命令時建立的檔案。根據預設，`tree.json` 檔案位於應用程式的 `cdk.out` 目錄。

若要開始使用 Toolkit AWS CDK Explorer，請建立 CDK 應用程式。

1. 完成 [AWS CDK 開發人員指南](#) 的 [Hello World 教學課程](#) 的前幾個步驟。

Important

當您到達部署堆疊步驟時，請停止並返回本指南。

Note

您可以在作業系統命令列界面或 VS Code 編輯器內的 Terminal (終端機) 視窗中執行教學課程中提供的命令，例如 `mkdir` 和 `cdk init`。

2. 完成 CDK 教學課程的必要步驟後，請開啟您在 AWS Cloud9 IDE 編輯器中建立的 CDK 內容。
3. 在 AWS 導覽窗格中，展開 CDK 標題。您的 CDK 應用程式及其相關資源現在會顯示在 CDK 檔案總管樹狀檢視中。您也可以終端機中的 AWS Cloud9 內執行下列指令，以確認 CDK 功能是否正常運作：

```
mkdir mycdkapp
cd mycdkapp
cdk init app --language=typescript
cdk synth
cdk bootstrap
```

重要說明

- 當您將 CDK 應用程式載入 AWS Cloud9 編輯器時，可以一次載入多個資料夾。每個資料夾可以包含多個 CDK 應用程式，如上圖所示。AWS CDK Explorer 會在專案根目錄及其直接子目錄中尋找應用程式。
- 當您執行教學課程的開頭幾個步驟時，您可能注意到您執行的最後一個命令是 `cdk synth`。此命令將您的 AWS CDK 應用程式轉換為 CFN，以合成 CloudFormation 範本。作為副產品，它也會產生 `tree.json` 檔案。如果您對 CDK 應用程式進行變更，請再次執行 `cdk synth` 命令以查看樹狀檢視中反映的變更。一個範例變更是將更多資源新增至應用程式。

在 AWS CDK 應用程式上執行其他操作

您可以使用 AWS Cloud9 編輯器在 CDK 應用程式上執行其他作業，方法與使用命令列界面相同。例如，您可以使用 AWS Cloud9 Terminal (終端機) 視窗更新編輯器中的程式碼檔案，並部署應用程式。

若要嘗試這些類型的動作，請使用 AWS Cloud9 編輯器繼續 AWS CDK 開發人員指南中的 [Hello World 教學課程](#)。確保執行最後一步，銷毀應用程式的資源。否則，您的 AWS 帳戶可能會產生非預期的費用。

使用 Git 面板的視覺化來源控制

AWS Cloud9 的 Git 面板提供了便利的視覺化界面，可使用基本的 Git 功能。

使用 Git 面板界面的選項，您就可以管理完整的來源控制生命週期：初始化儲存庫或複製遠端儲存庫、新增檔案至臨時區域、將暫存檔案遞交至工作目錄，然後將變更推送至上游儲存庫。

只要在 Git 面板界面點按幾下，您即可快速實作 Git 的核心協作和專案管理功能，例如建立和合併分支。此外，透過 IDE 的編輯器視窗，您也可以識別並解決合併衝突。

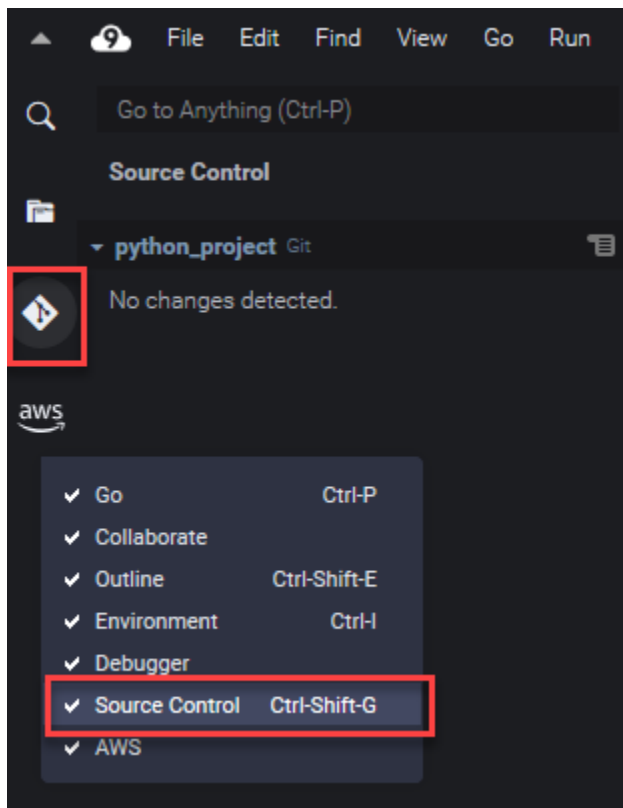
Important

Git 面板僅適用於以 Amazon EC2 執行個體建立的 AWS Cloud9 環境。如果您使用的是 [SSH 開發環境](#) 而非 EC2 環境，則不可存取此功能。

此外，在預設情況下，只有 2020 年 12 月 11 日之後建立的新 AWS Cloud9 環境可使用 Git 面板。我們正在努力為在此日期之前建立的開發環境啟用 Git 面板。

若要存取界面並與之互動，請選擇 Window (視窗)、Source Control (來源控制)。或者，您可以在 IDE 側邊面板的任何位置按一下右鍵，並選擇 Source Control (來源控制) 來前往來源控制。然後，選擇顯示在 IDE 界面中的 Git 圖示。

Ctrl-Shift-G 按鍵組合也可以用來切換 Git 面板的顯示。



Note

Git 面板文件的螢幕擷取畫面顯示的 AWS Cloud9 IDE 套用了 Jett Dark 佈景主題。如果您以不同佈景主題來使用 IDE，某些界面元素會以不同的方式顯示。若要開啟 Git 面板，請選擇標籤 Source Control (來源控制) 的連結，而不是 Git 圖示。

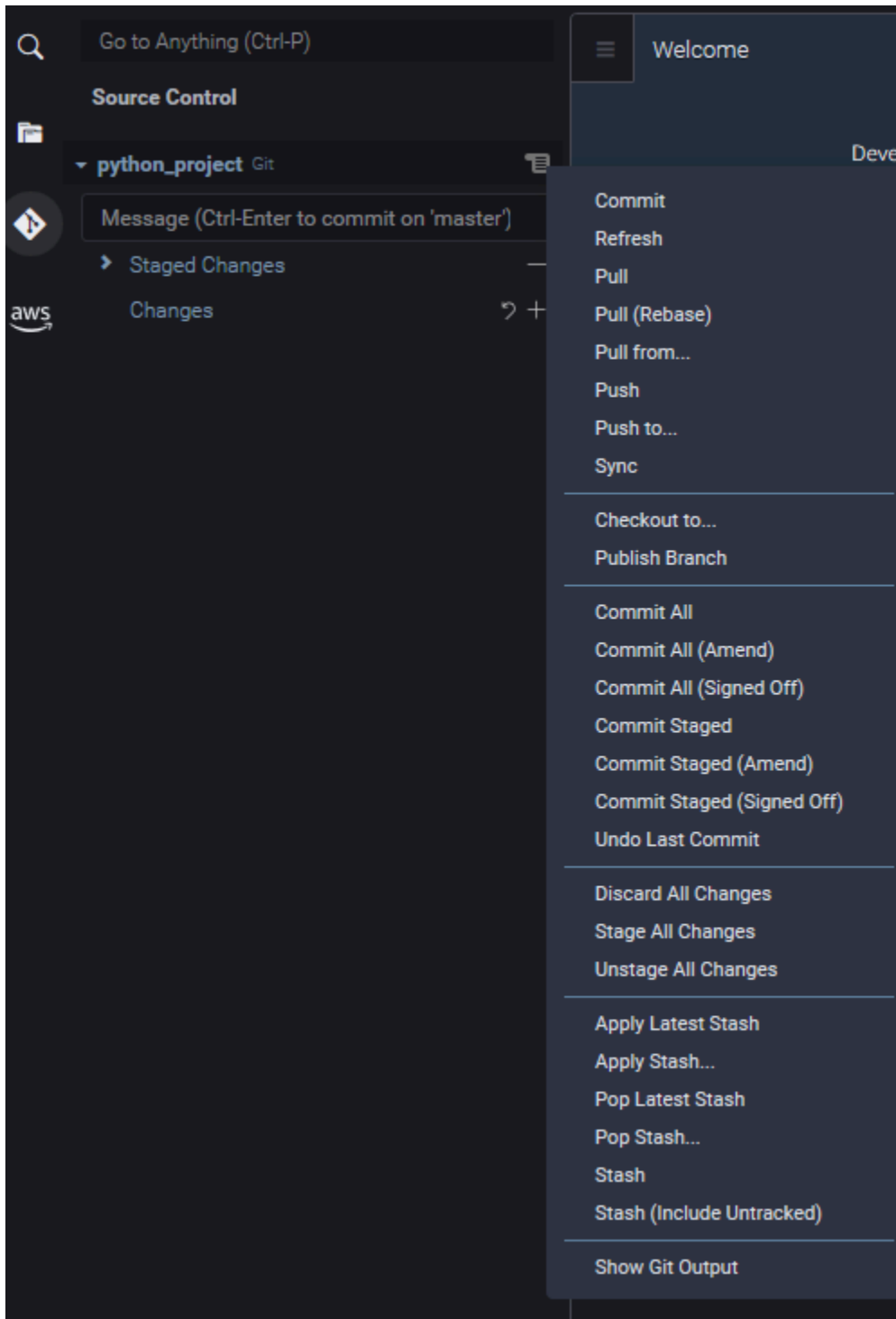
主題

- [使用 Git 面板管理來源控制](#)
- [參考：Git 面板中提供的 Git 命令](#)

使用 Git 面板管理來源控制

AWS Cloud9 的 Git 面板擴充功能提供了便利的使用者界面，可用於存取核心與進階的 Git 命令。

本節將示範如何存取主要 Git 功能以管理來源控制。這些程序的重點在於使用 Git 面板選單，針對您的儲存庫及其內容執行 Git 命令。



您也可以在 Git 面板搜尋方塊中輸入名稱，藉此存取任何支援的 Git 命令：



您也可以檢視您在與 Git 面板界面互動時實際執行的 Git 命令。若要檢視命令列活動，請移至 Git panel (Git 面板) 選單，然後選擇 Show Git Output (顯示 Git 輸出)。

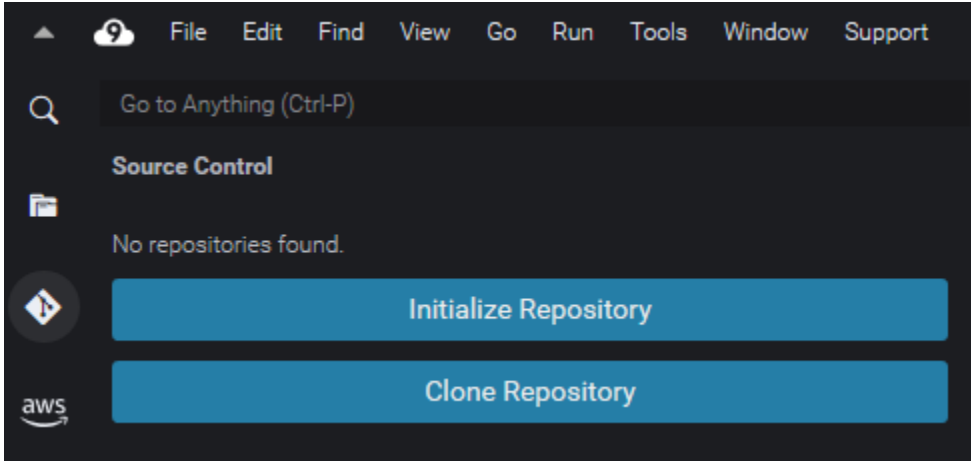
A screenshot of the Git command palette in AWS Cloud9. The palette is dark-themed and shows a search bar at the top with the text 'git'. Below the search bar, there are three sections: 'TABS CTRL-I' with 'No tabs found.', 'FILES CTRL-E|CTRL-O' with 'No files found.', and 'COMMANDS CTRL-.' with a list of Git commands. The first command, 'Git: Merge Branch...', is highlighted. Other commands include 'Git: Commit', 'Git: Add Remote', 'Git: Commit All', 'Git: Commit All (Signed Off)', 'Git: Fetch From All Remotes', 'Git: Commit All (Amend)', 'Git: Pull from...', 'Git: Create Branch From...', and 'Git: Remove Remote'. A 'More...' link is at the bottom.

初始化或複製 Git 儲存庫

Git 儲存庫包含專案從頭到尾的完整歷史記錄。儲存庫由專案內容的所有快照組成，這些快照是您每次將暫存檔案遞交該儲存庫時經由系統所擷取。

Git 面板支援兩種取得 Git 儲存庫的方式：

- 將現有目錄初始化為 Git 儲存庫。
- 將現有的儲存庫複製到本機目錄。



Note

如果您尚未將 Git 儲存庫新增至環境中的工作空間資料夾，才能使用初始化或複製儲存庫的界面選項。如果您已經有儲存庫的工作目錄，Git 面板視窗會顯示工作目錄和臨時區域的狀態。Git panel (Git 面板) 選單也可用來存取針對儲存庫執行的 Git 命令。

如何初始化或複製儲存庫

1. 如果 Git 面板還未啟用，請選擇 Window (視窗)、Source Control (來源控制)，然後選擇 Git 圖示，即可存取面板。

Note

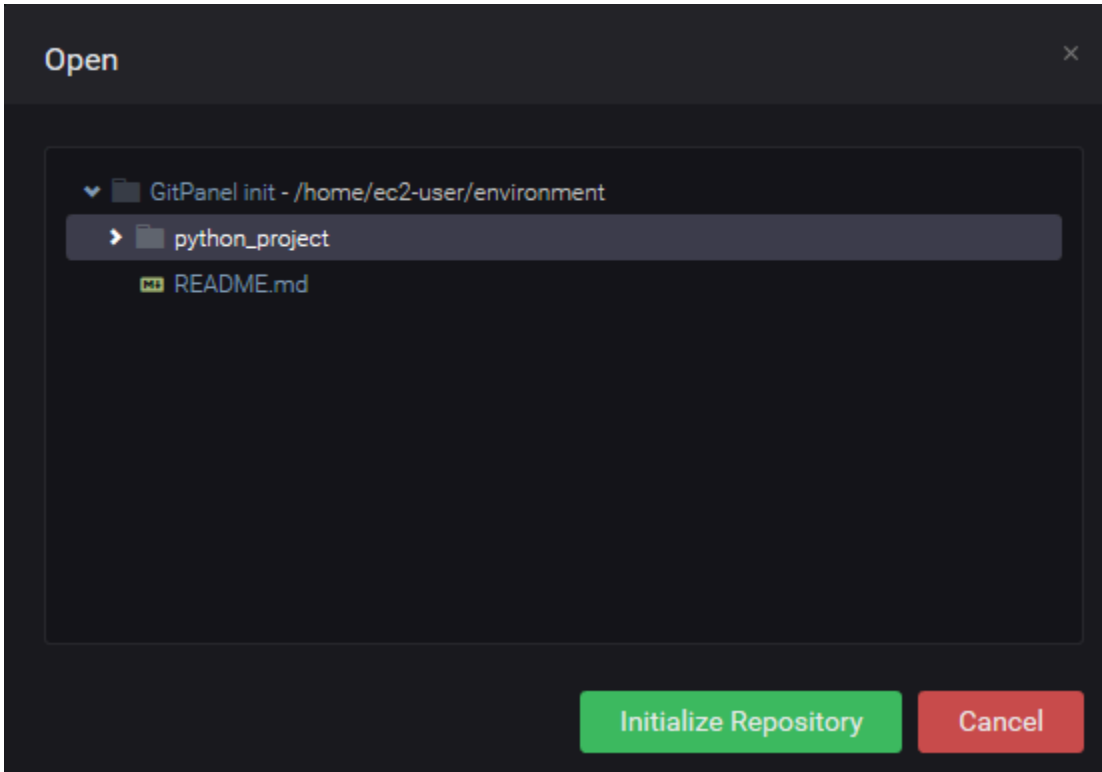
您也可以使用鍵盤快速鍵 Ctrl + Shift + G 開啟 Git 面板。

2. 選擇要初始化新的儲存庫還是複製現有的儲存庫。

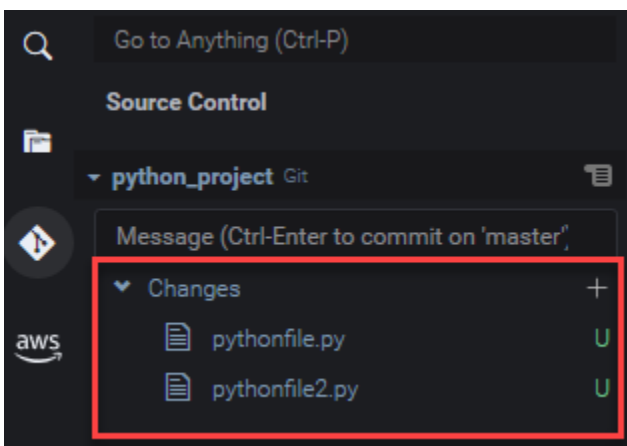
Initialize a repository

- 在 Git 面板中，選擇 Initialize Repository (初始化儲存庫)。

- 接下來，選擇要將 Git 儲存庫初始化的工作空間資料夾。您可以輸入資料夾的路徑、選擇路徑，或在對話方塊中選擇資料夾。
- 如果您使用對話方塊，請選取目的地資料夾並選擇 Initialize Repository (初始化儲存庫)。



在您初始化所選資料夾的 Git 儲存庫之後，Git 面板會顯示該資料夾中尚未追蹤但隨時可新增至 Git 臨時區域的現有檔案。



Clone a repository

- 在 Git 面板視窗中，選擇 Clone Repository (複製儲存庫)。

- 接下來，請輸入要複製的遠端儲存庫 URL (例如 `https://github.com/my-own-repo/my-repo-project-name.git`，代表複製在 GitHub 上託管的儲存庫)，然後按 Return (傳回)。
- 在顯示的對話方塊中，為複製的儲存庫選取工作空間資料夾，然後選擇 Select Repository Location (選取儲存庫位置)。

Note

如果您存取的是在外部網站 (例如 GitHub) 託管的儲存庫，您也必須輸入網站的登入憑證，才能完成此程序。

複製所選資料夾的遠端儲存庫後，您可以執行 `git pull` 命令使本機儲存庫與遠端儲存庫中的最新更新進行同步。如需更多詳細資訊，請參閱 [使用遠端儲存庫](#)。

暫存和遞交檔案

取得 Git 儲存庫之後，您就可以使用兩步驟程序來填入內容：

1. 將未追蹤和最近修改的內容新增至臨時區域。
2. 將臨時區域中的檔案遞交至工作目錄。

Important

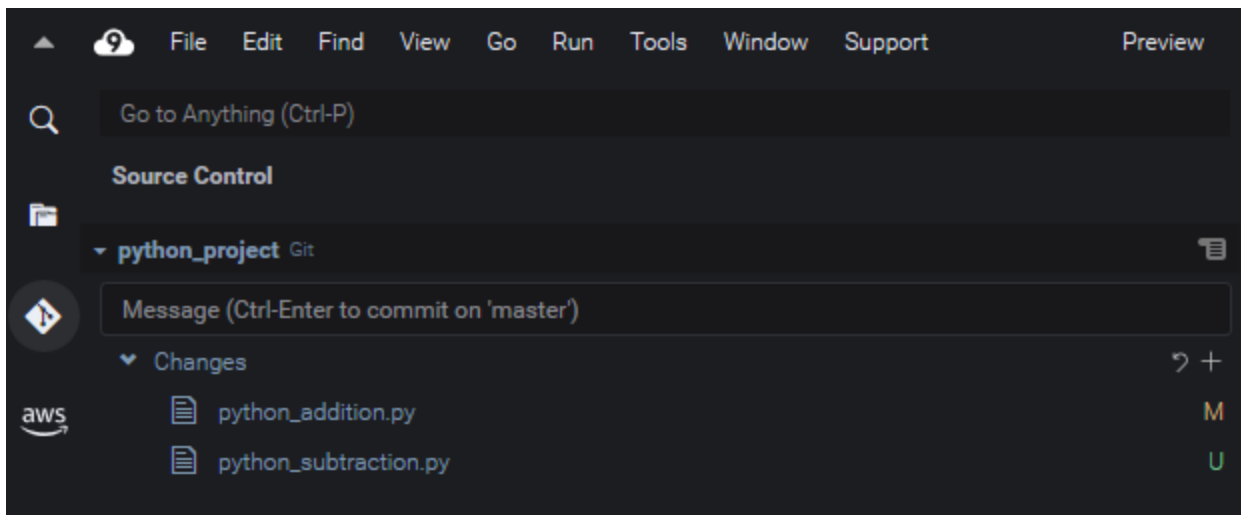
您未必會將工作目錄中的每個檔案都遞交到儲存庫。舉例來說，您也許不太可能會將執行時間產生的檔案新增到專案的儲存庫中。透過 Git 面板，您就可以將要忽略的檔案新增至 `.gitignore` 檔案中的清單，以便標記這些檔案。

若要更新 `.gitignore` 的清單，請以滑鼠右鍵按一下尚未新增至臨時區域的檔案，然後選取 Add File to `.gitignore` (將檔案新增到 `.gitignore`)。IDE 會開啟 `.gitignore` 檔案，而所選檔案的名稱會新增至忽略檔案清單。

如需了解如何在 `.gitignore` 使用模式比對來排除檔案類型，請參閱 git-scm.com 網站中的 [相關參考資料](#)。

Stage files

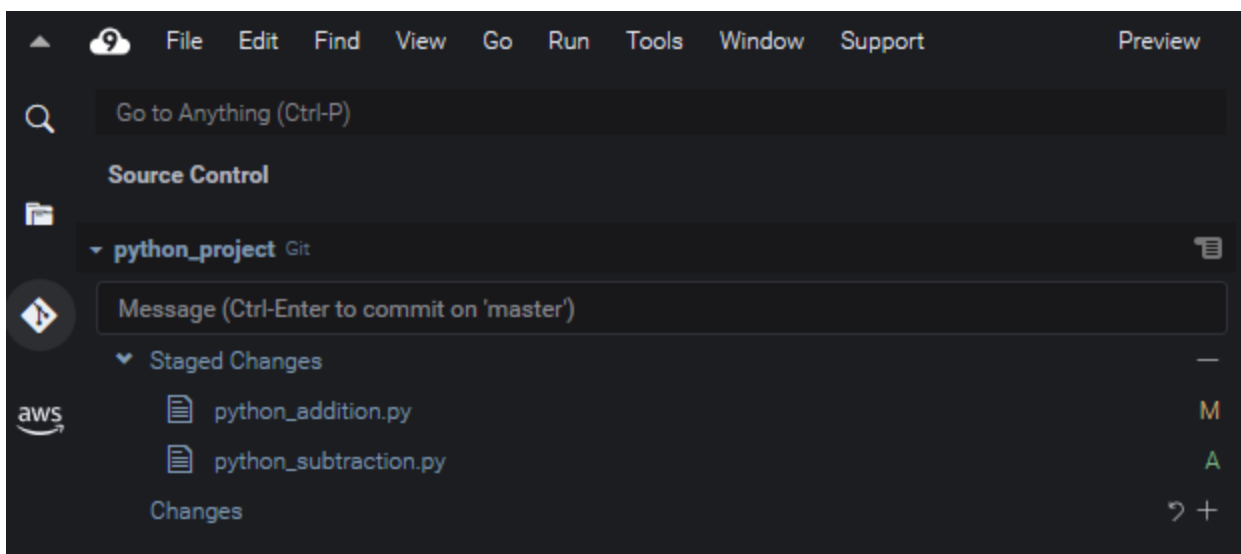
未追蹤的檔案 (標記為 "U") 和未新增至暫存區域的已修改檔案 (標示為 "M") 會列在 Git 面板窗格的 Changes (變更) 下。



使用 Git 面板界面，您可以將特定檔案或所有未追蹤與修改的檔案新增至臨時區域：

- 特定檔案：暫停在檔案上，然後選擇 + 以便新增到臨時區域。或者，在檔案上按一下滑鼠右鍵，然後選擇 Stage Changes (暫存變更)。
- 所有檔案：前往 Git panel (Git 面板) 選單，然後選擇 Stage All Changes (暫存所有變更)。

新增至儲存庫索引的檔案會列在 Staged Changes (已暫存的變更) 中。之前未追蹤的檔案會標記為 "A"，表示檔案已暫存。



Note

您也可以取消暫存特定變更或所有變更。如果只有一個檔案，請暫停在該檔案上，然後選擇 `-`。或者，按一下滑鼠右鍵，然後選擇 `Unstage Changes` (取消暫存變更)。若要取消暫存所有變更，請前往 `Git panel` (Git 面板) 選單，然後選擇 `Unstage All Changes` (取消暫存所有變更)。

Commit files

您可以使用 Git 的 `commit` 命令在儲存庫中將暫存檔案擷取為永久快照。使用 Git 面板界面，您可以選擇要遞交的檔案：

- 遞交臨時區域中的檔案：前往 `Git panel` (Git 面板) 選單，然後選擇 `Commit` (遞交) 或 `Commit Staged` (遞交暫存檔案)。
- 遞交工作目錄中的所有檔案：前往 `Git panel` (Git 面板) 選單，然後選擇 `Commit All` (全部遞交)。(此選項會使用 `git add` 將檔案新增至臨時區域，然後再呼叫 `git commit`。)

Note

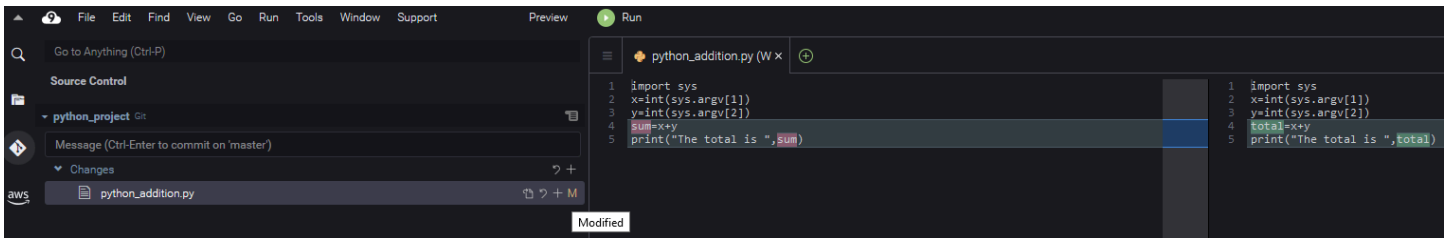
您也可以在透過 Git 面板遞交檔案時使用 `amend` 和 `signed-off` 選項。`amend` 選項會修改最近遞交檔案的遞交訊息。`sign-off` 選項可以在 Git 日誌中識別執行遞交的人員。您也可以前往 `Git panel` (Git 面板) 選單並選擇 `Undo Last Commit` (復原上次的遞交)，藉此還原遞交檔案。

檢視不同的檔案版本

您可以比較檔案在經過暫存或遞交後的修改版本。

- 列於 `Changes` (變更) 之下的檔案：選擇「M」以檢視工作目錄中的版本與上次暫存或遞交至儲存庫的版本差異。
- 列於 `Staged Changes` (暫存的變更) 之下的檔案：選擇「M」以檢視臨時區域中的版本與上次遞交至儲存庫的版本差異。

選擇「M」之後，IDE 視窗會顯示兩個檔案版本之間的差異。一側會顯示儲存庫中目前追蹤的版本。另一側則會顯示尚未遞交的修改版本。



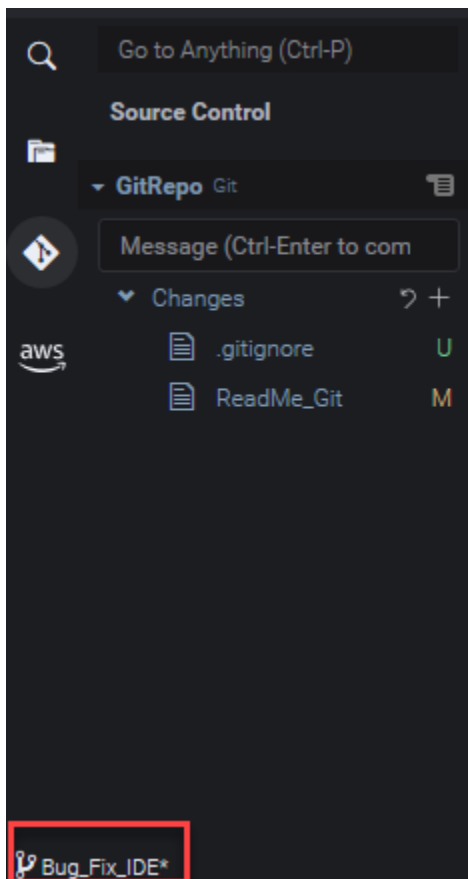
使用分支

Git 允許您在有別於儲存庫主分支的分支中使用新功能，因此大幅改進了工作流程的管理。您可以在多個分支之間流暢切換，同時確保您在主分支中一律擁有隨時可建置的原始程式碼。

建立分支

建立分支時必須為分支命名並選取其起點。

1. 在 Git panel (Git 面板) 選單中，選擇 Checkout to (簽出至)。或者，您也可以可以在 Git 面板底部選擇目前顯示的分支名稱。



2. 選擇建立新分支的選項：

- Create new branch (建立新分支)：從目前的分支上次遞交的位置建立新分支。
 - Create new branch from (選擇建立新分支的位置)：從您在下個畫面選取的分支上次遞交的位置建立新分支。
3. 輸入新分支的名稱。
 4. 如果您要將特定分支指定為分支起點，請從清單中選取分支。

切換到新分支之後，查看 Git 面板底部即可看到目前分支的名稱。

Note

如果您使用遠端儲存庫，[publish the new branch](#) (發佈新分支) 到上游遠端儲存庫，即可允許其他人存取您的內容。

切換分支

使用 Git 管理來源控制的其中一個主要優點，就是切換分支時，也可以在不同的專案之間切換。

Important

如果目前的分支還有檔案尚未遞交到儲存庫，就無法切換分支。您必須先清除工作目錄，方法是[遞交](#)或[隱藏](#)工作。

1. 在 Git 面板底部選擇目前分支的名稱。或者，前往 Git panel (Git 面板)，然後選擇 Checkout to (簽出至)。
2. 從顯示的清單中選擇分支。

切換之後，儲存庫的工作目錄會更新為最近遞交至所選分支的檔案版本。

合併分支

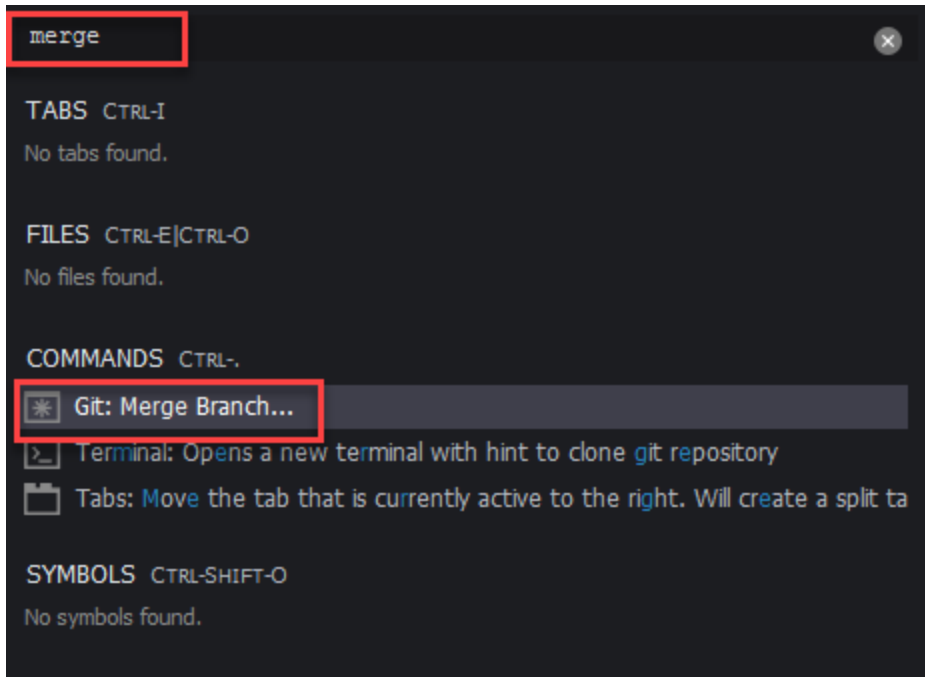
在您使用完特定分支中的功能之後，通常會想要將變更整合到主要專案中。透過 Git，您就可將將某個分支 (例如功能分支) 合併到其他分支 (通常是儲存庫的主分支或預設分支)，藉此促成上述整合。

1. 若要選取想要合併到其他分支的分支，請前往 Git panel (Git 面板) 選單，然後選擇 Checkout to (簽出至)。

或者，在 Git 面板底部選擇目前分支的名稱。

2. 從顯示的清單中，選擇要當作切換目標的分支。
3. 在 Git 面板的 Search (搜尋) 方塊中，輸入「merge」(合併)。

選擇顯示在 Commands (命令) 清單底下的 Git: Merge Branch。

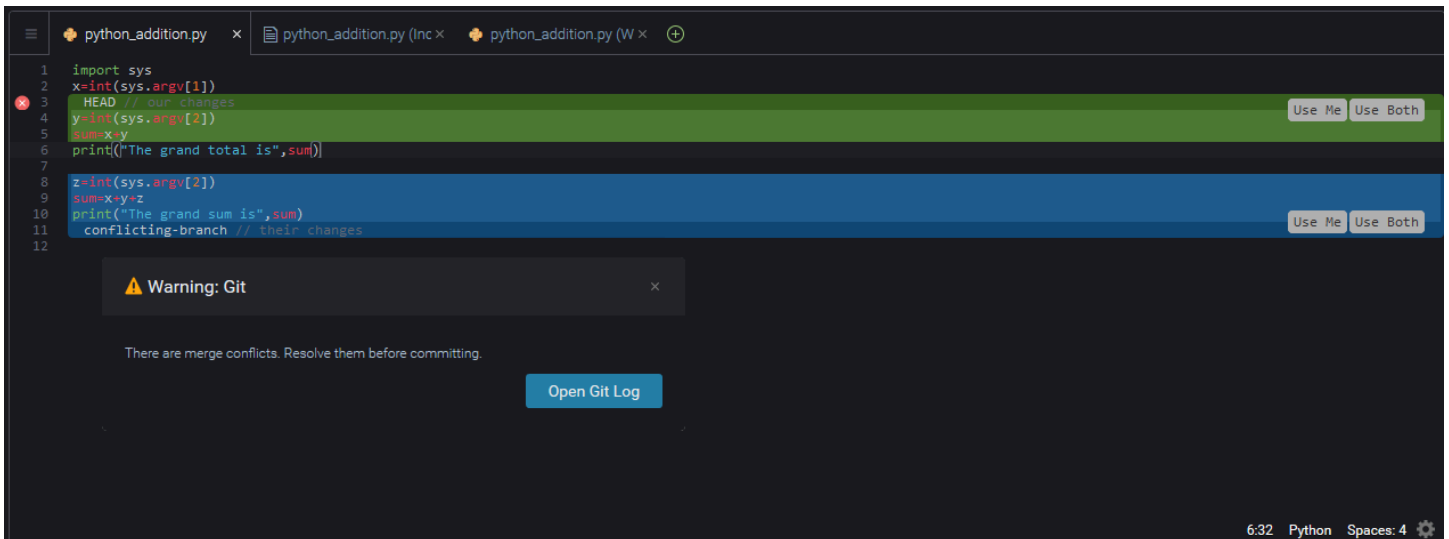


4. 從顯示的清單中，選擇要合併到目標分支中的分支。

如果合併順利完成，沒有發生衝突，Git 面板界面會重新整理，顯示含有已合併變更的目標分支。

合併分支時，您可能會遇到對相同內容做出不相容變更造成的合併衝突。如果發生這種情況，系統會警告您必須解決衝突才能遞交變更。

您可以使用 IDE 的程式碼編輯器視窗來識別兩個分支中的衝突內容，然後做出變更來解決差異問題。



```

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

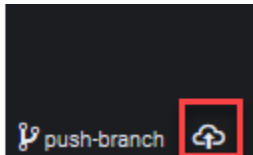
使用遠端儲存庫

託管在網際網路或網路上的遠端儲存庫可讓團隊成員分享他們針對本機職責所做的變更，藉此促進協作。透過上傳和下載資料的 Git 命令，您可以確保「下游」(本機) 儲存庫的內容能與「上游」(遠端) 儲存庫的內容同步。

將分支發佈到遠端儲存庫

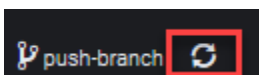
為本機儲存庫建立分支後，它會成為您的私有儲存庫，且除非您推送到「上游」的遠端儲存庫，否則協作者無法存取。

- 若要發布目前的分支，請前往 Git panel (Git 面板) 選單，然後選擇 Publish Branch (發佈分支)。或者，在位於 Git 面板底部按一下分支名稱旁的雲朵符號。



- 如有必要，請輸入您的登入憑證，以存取遠端儲存庫。

如果分支成功發佈至遠端儲存庫，Git 面板底部的分支名稱旁會顯示同步處理符號。選擇此符號可同步本機和遠端儲存庫的內容。



在本機和遠端儲存庫之間推送和提取內容

使用 Git 協作共用專案時，通常會先從遠端儲存庫將其他團隊成員最近的變更提取到您的本機儲存庫中。在您將變更遞交到本機儲存庫後，您會將變更推送到遠端儲存庫，以便讓團隊其他人存取。這些動作是由 `git pull` 和 `git push` 命令來執行。

Note

將變更推送至大多數託管儲存庫 (例如 GitHub 上的儲存庫)，以及從中提取變更時，您需要輸入登入憑證。

Pull changes from remote

透過 Git 面板界面使用 `git pull` 命令，您就可以將本機儲存庫更新為遞交至遠端儲存庫分支的最新變更。

1. 在 Git panel (Git 面板) 選單中，選擇 Checkout to (簽出至)。
2. 在分支清單中，選擇您要將變更提取至當中的本機分支。
3. 接著，前往 Git panel (Git 面板) 選單並選擇 Pull from (提取來源)。
4. 選擇遠端儲存庫，然後從中選擇要提取變更的分支。

完成提取後，您就可以存取從工作目錄的遠端儲存庫中擷取的檔案。修改檔案後，您接著就可以將變更推送至遠端分支。

Push changes to remote

透過 Git 面板界面使用 `git push` 命令，您就可以將遠端儲存庫更新為在本機儲存庫指定分支中所做的最新變更。

1. 在 Git panel (Git 面板) 選單中，選擇 Checkout to (簽出至)。
2. 在分支清單中，選擇您要當作推送變更來源的本機分支。
3. 接著，前往 Git panel (Git 面板) 選單並選擇 Push to (推送到)。
4. 選擇遠端儲存庫，然後從中選擇要將變更推送到其中的分支。

執行推送後，其他團隊成員就可以將變更提取到自己的本機儲存庫副本來存取變更。

隱藏和擷取檔案

使用 Git 的隱藏功能，您不必先遞交暫存或修改檔案就可以切換分支。隱藏功能會擷取工作目錄和臨時區域的目前狀態，然後儲存以供日後使用。如果您在處理的內容尚未完成，且需要立即切換分支，此功能就非常實用。

隱藏工作

- 若要隱藏工作目錄的目前狀態，請前往 Git panel (Git 面板) 選單並選擇以下其中一個選項：
 - Stash (隱藏)：工作目錄中所有已修改或暫存的檔案會新增至隱藏項目。這個動作不會新增未追蹤的檔案。
 - Stash (include Untracked) (隱藏 (包括未追蹤))：工作目錄中的所有檔案 (包括未追蹤的檔案) 都會新增至隱藏項目。
- 按需要而輸入訊息，有助您識別隱藏項目，以便日後擷取。

隱藏後，Git 面板界面會重新整理，顯示已清除的工作目錄。

擷取儲存庫

- 若要擷取隱藏項目並套用到工作目錄，請前往 Git panel (Git 面板) 選單並選擇以下其中一個選項：
 - Apply Stash (套用隱藏項目)：將所選隱藏項目套用到工作目錄，並保留隱藏項目供日後使用。
 - Pop Stash (快顯隱藏項目)：將所選隱藏項目套用到工作目錄，並從隱藏堆疊中刪除隱藏項目。

Note

您也可以選擇套用或快顯最近新增至隱藏堆疊的隱藏項目。

- 選取要套用到工作目錄的隱藏項目。

Git 面板界面會重新整理，顯示已套用隱藏項目的工作目錄。

參考：Git 面板中提供的 Git 命令

AWS Cloud9 的 Git 面板選單提供了便利的使用者界面，可用於存取核心與進階的 Git 命令。

某些 Git 命令 (例如用來合併和刪除分支的命令) 只能透過 Git 面板搜尋欄位使用。

您也可以自訂 Git 面板執行命令以及與儲存庫互動的方式。若要修改預設設定，請先選擇 AWS Cloud9、Preferences (偏好設定)。接下來，在 Preferences (偏好設定) 視窗的 Project Settings (專案設定) 中，選擇 Git。

暫停在資訊圖示上，閱讀設定的簡短說明。

The screenshot shows the 'Project Settings' window in AWS Cloud9, with the 'Git' section selected. The 'Git Enabled' toggle is turned on and highlighted with a yellow box, with the text 'Whether git is enabled.' displayed next to it. Other settings include:

- Aws: Log Level: ?
- Aws: Telemetry: ? (toggle on)
- Save Settings to Disk
- Code Editor (Ace)
- Run Configurations
- Find in Files
- Run & Debug
- Build
- Hints & Warnings
- Code Formatters
- JavaScript Support
- TypeScript Support
- PHP Support
- Python Support
- Go Support
- EC2 Instance
- EXTENSIONS
- AWS Configuration
- Git (selected)
- User Settings
- AWS Settings
- Keybindings
- Themes
- Experimental

Git settings include:

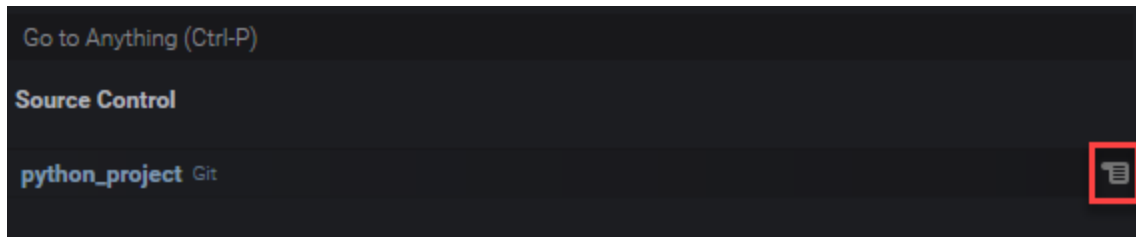
- Git Path: ?
- Git: Auto Repository Detection: ? (Scan for both subfolders of t)
- Git: Autorefresh: ? (toggle on)
- Git: Autofetch: ? (toggle off)
- Git: Autofetch Period: ? (180)
- Git: Branch Validation Regex: ?
- Git: Branch Whitespace Char: ? (-)
- Git: Confirm Sync: ? (toggle on)
- Git: Count Badge: ? (Count all changes.)
- Git: Checkout Type: ? (Show all references.)
- Git: Ignore Legacy Warning: ? (toggle off)
- Git: Ignore Missing Git Warning: ? (toggle off)
- Git: Ignore Limit Warning: ? (toggle off)
- Git: Default Clone Directory: ?

Note

您可以從官方 Git 網站存取列出的 Git 命令的詳細文件：<https://git-scm.com/doc>。

Git 面板選單中提供的 Git 命令參考資料

您可以選擇儲存庫名稱對面的符號來存取 Git panel (Git 面板) 選單。



Git 面板選單

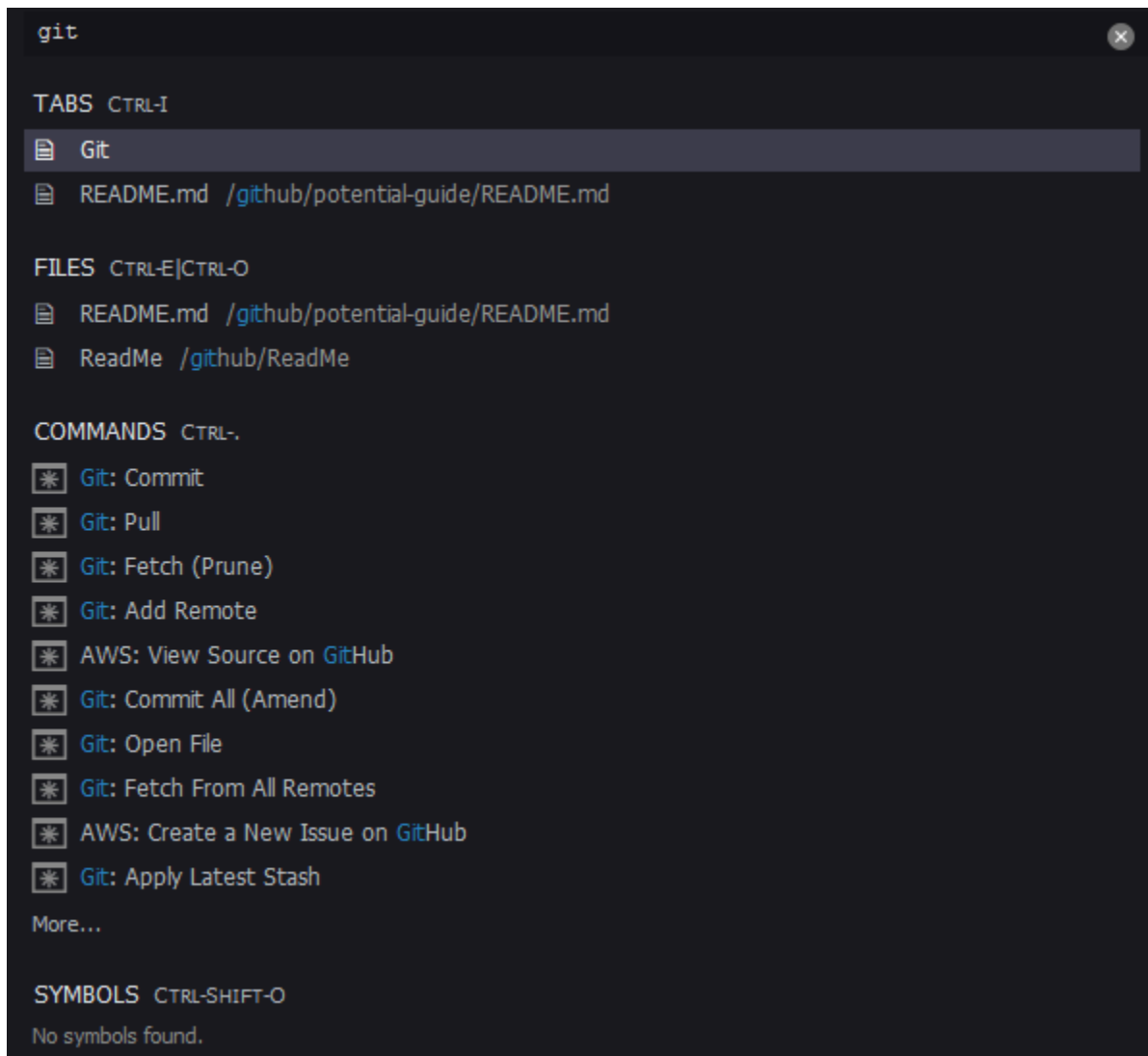
選單選項	描述
遞交	將新增了臨時區域的內容遞交到儲存庫的工作目錄。新增遞交訊息。
重新整理	重新整理 GitPanel 界面，以顯示工作目錄和臨時區域的狀態。
提取	將遠端儲存庫的最新變更提取到本機儲存庫。
提取 (重訂基底)	將本機變更重新套用到從遠端分支提取的遠程變更。
推送來源...	將遞交到本機儲存庫的分支中的變更推送到遠端儲存庫中的分支。
推送	將遞交至本機儲存庫的變更推送到遠端儲存庫。
推送到...	將遞交到本機儲存庫的分支中的變更推送到遠端儲存庫中的分支。
Sync	先後執行 <code>git pull</code> 命令和 <code>git push</code> 命令，藉此同步本機和遠端儲存庫的內容。
簽出至...	切換到現有分支，或建立分支並切換至該分支。

選單選項	描述
發佈分支	發佈建立於本機儲存庫的私有分支，並將它設為可在遠端儲存庫上使用。
全部遞交	將已暫存和未暫存的檔案都遞交至儲存庫。(系統會在執行 <code>git commit</code> 命令之前先執行 <code>git add -A</code> 命令，以便將檔案新增到臨時區域。)
全部遞交 (修改)	修改上次遞交的訊息。(執行 <code>git commit</code> 命令時增加了 <code>-amend</code> 選項。)
全部遞交 (登出)	識別在 Git 日誌中執行遞交的身分。(執行 <code>git commit</code> 命令時增加了 <code>-signed-off</code> 選項。)
遞交暫存檔案	只將暫存的檔案遞交至儲存庫。
遞交暫存檔案 (修改)	修改上次遞交的訊息。(執行 <code>git commit</code> 命令時增加了 <code>-amend</code> 選項。)
遞交暫存檔案 (登出)	識別在 Git 日誌中執行遞交的身分。(執行 <code>git commit</code> 命令時增加了 <code>-signed-off</code> 選項。)
復原上次的遞交	復原上一次遞交的檔案。檔案會移回臨時區域。
捨棄所有變更	從儲存庫的臨時區域中刪除所有檔案和資料夾。
暫存所有變更	將未追蹤和已修改的內容新增至臨時區域。
取消暫存所有變更	將所有檔案移出臨時區域。取消暫存的檔案無法遞交至儲存庫。
套用最新隱藏項目	將新增至堆疊隱藏項目的最後一個隱藏項目套用至工作目錄。隱藏項目仍然存在堆疊中。
套用隱藏項目...	將從隱藏堆疊中選取的隱藏項目套用至工作目錄。隱藏項目仍然存在堆疊中。
快顯最新隱藏項目	將新增至堆疊隱藏項目的最後一個隱藏項目套用至工作目錄。該隱藏項目接著會從堆疊中遭到刪除。

選單選項	描述
快顯隱藏項目...	將選取的隱藏項目套用到工作目錄。該隱藏項目接著會從堆疊中遭到刪除。
隱藏	將工作目錄中已修改且暫存的檔案新增到已命名的隱藏項目。
隱藏 (包括未追蹤)	將工作目錄中的所有檔案 (包括未追蹤的檔案) 新增到已命名的隱藏項目。
顯示 Git 輸出	顯示的視窗中會顯示您與 Git 面板界面互動時執行的 Git 命令。

Git 面板搜尋欄位中可用的 Git 命令

有些受支援的 Git 命令並未在 Git 面板選單中提供，但您可以在搜尋方塊中輸入「git」，以便存取這些命令：



可以透過上述方式存取的特定 Git 命令相關說明如下表所示。

特定 Git 命令

選單選項	描述
Git : 新增遠端	將遠端儲存庫的連線新增到您的 Git 組態檔
Git : 刪除分支	刪除指定的分支。
Git : 擷取	從遠端儲存庫的分支下載內容。與 <code>git pull</code> 不同的是，遠端的變更不會合併到本機儲存庫中。

選單選項	描述
Git : 合併分支	將某個分支中的變更整合到其他分支中。如需詳細資訊，請參閱 合併分支程序 。

AWS Toolkit

為什麼要使用 AWS Toolkit ?

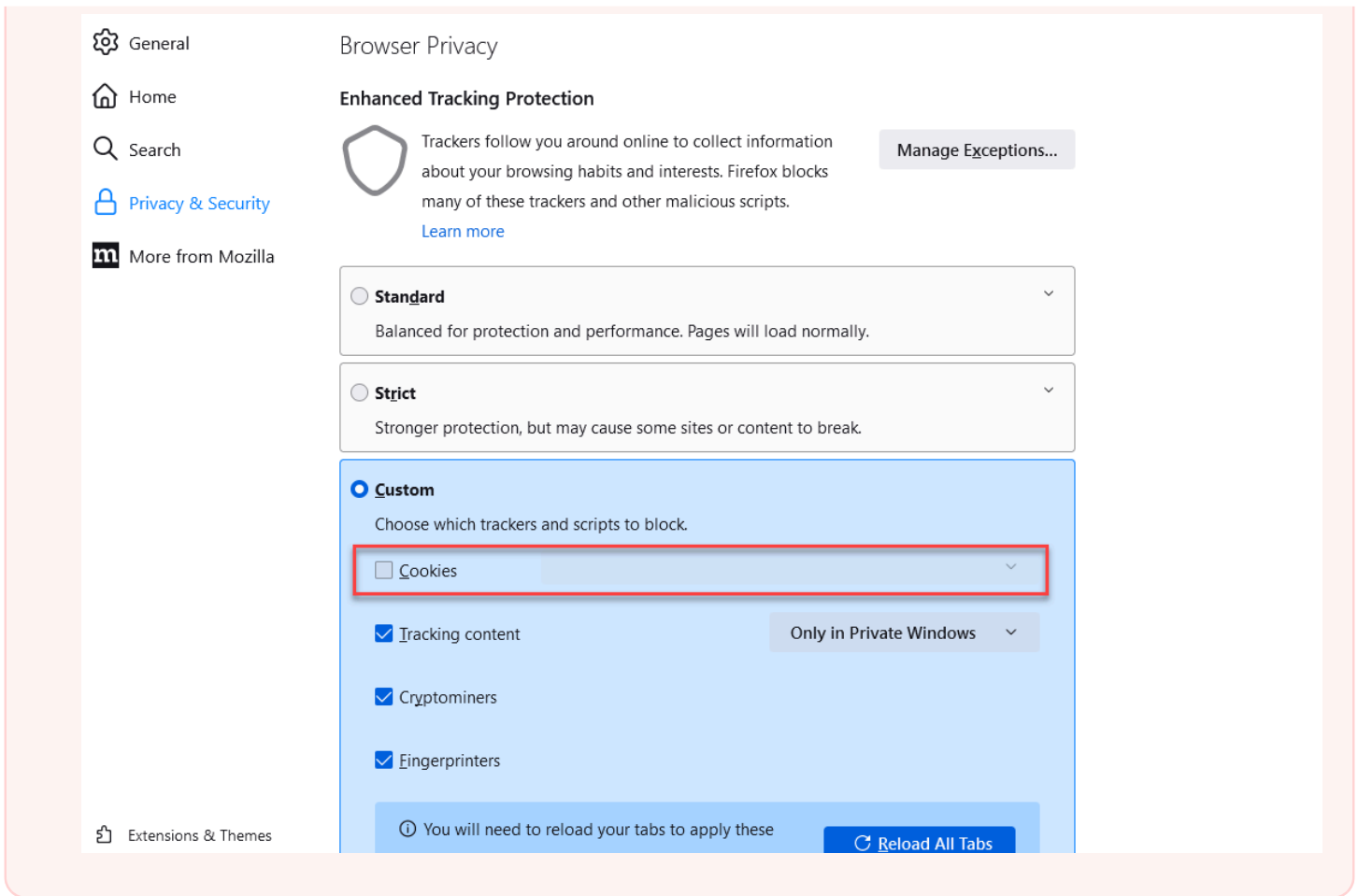
AWS Toolkit 是 AWS Cloud9 整合式開發環境 (IDE) 的延伸模組。您可以藉由此延伸模組來存取和使用多種 AWS 服務。此 AWS Toolkit 取代了 AWS Cloud9 的 Lambda 外掛程式所提供的功能。如需詳細資訊，請參閱[停用 AWS Toolkit](#)。

Important

AWS Toolkit 支援是一項 AWS Cloud9 的整合功能。您目前無法使用第三方延伸模組自訂 AWS Cloud9 IDE。

Warning

如果您使用 Mozilla 火狐瀏覽器作為您首選的瀏覽器與 AWS Cloud9 IDE，有一個第三方 cookie 設置，可以防止AWS Cloud9網絡視圖和工AWS具包在瀏覽器中正常工作。為了解決此問題，您必須確保您未在瀏覽器設定的「隱私權與安全性」區段中封鎖 Cookie，如下圖所示。



目前 AWS Toolkit 延伸模組可提供以下 AWS 服務 與資源：

- [AWS App Runner](#)
- [API Gateway](#)
- [AWS CloudFormation 堆疊](#)
- [CloudWatch 日誌](#)
- [AWS Lambda](#)
- [資源](#)
- [Amazon S3 儲存貯體和物件](#)
- [AWS Serverless Application Model 應用程式](#)
- [Step Functions 和狀態機器](#)
- [Systems Manager 自動化文件](#)
- [在 AWS Cloud9 IDE 中使用 Amazon ECR](#)

- [AWS IoT](#)
- [???](#)
- [Amazon EventBridge](#)
- [與 Amazon 合作 CodeWhisperer](#)
- [使用 AWS Cloud Development Kit \(AWS CDK\)](#)

啟用 AWS Toolkit

如果 AWS Toolkit 在您的環境中無法使用，您可以在 Preferences (偏好設定) 索引標籤。

啟用 AWS Toolkit

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 索引標籤的側邊導覽窗格中，選擇 AWS Settings (AWS 設定)。
3. 在 AWS Resources (AWS 資源) 窗格中，啟用 AWS Toolkit，這樣它就會在綠色背景上顯示核取記號。

當您啟用 AWS Toolkit 時，整合式開發環境 (IDE) 會重新整理，以顯示更新的 Enable AWS Toolkit (啟用 AWS Toolkit) 設定。在 Environment (環境) 選項下方的 IDE 側邊也會顯示 AWS Toolkit 選項。

Important

如果您 AWS Cloud9 環境的 EC2 執行個體無法存取網際網路 (不允許傳出流量)，則在您開啟 AWS Toolkit 並重新啟動 IDE 後可能會顯示相關訊息。此訊息會指出 AWS Toolkit 所需的相依性無法下載。若遇到此情況，您也無法使用 AWS Toolkit。

若要修正此問題，請為 Amazon S3 建立 VPC 端點。這麼一來您就能存取 AWS 區域中的 Amazon S3 儲存貯體，其中包含將 IDE 維持在最新狀態所需的相依性。

如需詳細資訊，請參閱[將 Amazon S3 的 VPC 端點設為下載相依性](#)。

管理 AWS Toolkit 的存取憑證

AWS Toolkit 會與多個 AWS 服務 互動。若要管理存取控制，請確認 AWS Toolkit 服務的 IAM 實體具有此服務範圍的必要許可。若要快速入門，請使用 [AWS 管理的臨時憑證](#) 取得必要許可。這些受管憑證的運作方式是代表 AWS 實體 (例如 IAM 使用者)，授予 EC2 環境存取 AWS 服務。

但是，如果您已在私有子網路中啟動開發環境的 EC2 執行個體，AWS 受管臨時憑證就無法使用。如需替代方案，您可以手動建立一組自有憑證，藉此允許 AWS Toolkit 存取 AWS 服務。這組憑證稱為設定檔。設定檔具有稱為存取金鑰的長期憑證。您可以從 IAM 主控台取得這些存取金鑰。

建立設定檔為 AWS Toolkit 提供存取憑證

1. 若要取得存取金鑰 (包含存取金鑰 ID 和私密存取金鑰)，請前往 IAM 主控台，網址為 <https://console.aws.amazon.com/iam>。
2. 從導覽窗格選擇 Users (使用者)，然後選擇您的 AWS 使用者名稱 (而不是核取方塊)。
3. 選擇 Security credentials (安全憑證) 標籤，然後選擇 Create access key (建立存取金鑰)。

Note

如果您已經擁有存取金鑰，但無法存取私密金鑰，請將舊金鑰停用，然後建立新的金鑰。

4. 在顯示存取金鑰 ID 和私密存取金鑰的對話方塊中，選擇 Download .csv file (下載 .csv 檔案)，將此資訊存放在安全位置。
5. 下載存取金鑰後，請啟動 AWS Cloud9 環境，並選擇 Window (視窗)、New Terminal (新增終端機) 來啟動終端機工作階段。
6. 在終端機視窗中，執行下列命令。

```
aws configure --profile toolkituser
```

`toolkituser` 是本例使用的設定檔名稱，但您可以選擇自己的設定檔。

7. 在命令列輸入您先前從 IAM 主控台下載的 AWS Access Key ID 和 AWS Secret Access Key。
 - 針對 Default region name，請指定 AWS 區域 (例如 `us-east-1`)。
 - 針對 Default output format，請指定檔案格式 (例如 `json`)。

Note

如需了解設定檔的設定選項，請參閱《AWS Command Line Interface 使用者指南》中的[組態基本概念](#)。

8. 建立設定檔後，請啟動 AWS Toolkit、前往 [AWS Toolkit menu \(AWS Toolkit 功能表\)](#)，然後選擇 Connect to AWS (連線到 AWS)。
9. 在 Select an AWS credential profile (選取 AWS 憑證設定檔) 欄位中，選擇您剛剛在終端機中建立的設定檔 (例如，profile:toolkituser)。

如果選取的設定檔包含有效的存取憑證，AWS Explorer 窗格會重新整理，顯示您現在可以存取的 AWS 服務。

使用 IAM 角色為 EC2 執行個體上的應用程式授予許可

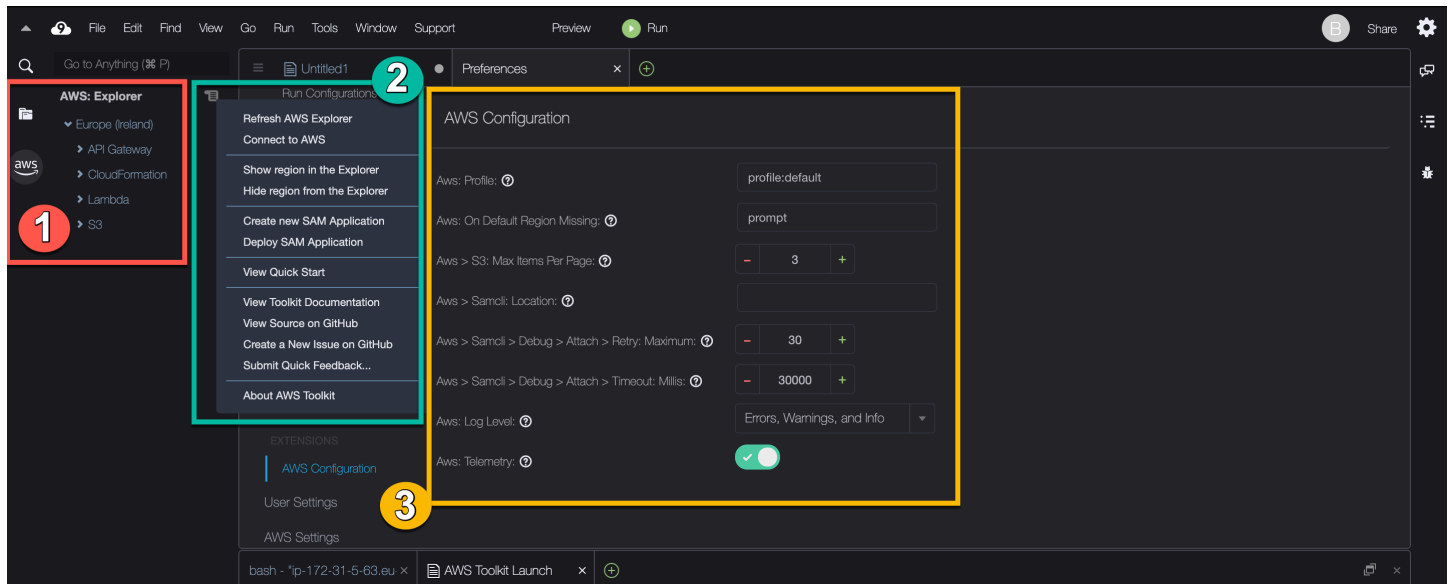
您也可以使用 IAM 角色為在 EC2 執行個體上執行的應用程式管理臨時憑證。該角色提供應用程式在呼叫其他 AWS 資源時可以使用的臨時許可。啟動 EC2 執行個體時，指定要與執行個體建立關聯的 IAM 角色。然後，在執行個體上執行的應用程式就可以使用角色提供的臨時憑證來回應 AWS 服務的 API 請求。

建立角色之後，請建立執行個體設定檔，藉此將此角色及其相關許可指派給執行個體。執行個體描述檔會連接到執行個體，並可將角色的臨時憑證提供給在執行個體上執行的應用程式。

如需詳細資訊，請參閱 IAM 使用者指南中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

識別 AWS Toolkit 元件

以下螢幕擷取畫面顯示 AWS Toolkit 的三種主要 UI 元件。



1. AWS Explorer 視窗：可讓您與可透過 Toolkit 存取的 AWS 服務 互動。您可以使用位於整合式開發環境 (IDE) 左側的 AWS 選項來切換 AWS Explorer 的顯示和隱藏。如需詳細了解如何使用此界面元件和存取其他 AWS 區域的 AWS 服務，請參閱 [透過 AWS Explorer 來使用多個區域的服務和資源](#)。
2. 工具組功能表：用來管理連線AWS、自訂 AWSExplorer 視窗的顯示、建立和部署無伺服器應用程式、使用 GitHub 儲存庫，以及存取文件。如需詳細資訊，請參閱[存取和使用 AWS 工具組選單](#)。
3. AWS Configuration (AWS 組態) 窗格：用來自訂您使用 Toolkit 與其進行互動之 AWS 服務的行為。如需詳細資訊，請參閱[使用 AWS 組態窗格修改 AWS 工具組設定](#)。

停用 AWS Toolkit

您可以在 Preferences (偏好設定) 索引標籤停用 AWS Toolkit。

如何停用 AWS Toolkit

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 索引標籤的側邊導覽窗格中，選擇 AWS Settings (AWS 設定)。
3. 在 AWS Resources (資源) 窗格中，關閉 AWS AWS Toolkit。

停用 AWS Toolkit 後，整合式開發環境 (IDE) 會重新整理，在 IDE 側邊移除 Environment (環境) 選項下方的 AWS 工具組選項。

AWS Toolkit 主題

- [導覽和設定 AWS 工具組](#)
- [使用 AWS App Runner 和 AWS Toolkit](#)
- [透過 AWS 工具組使用 API Gateway](#)
- [透過 AWS 工具組使用 AWS CloudFormation 堆疊](#)
- [透過 AWS Toolkit 使用 AWS Lambda 函式](#)
- [使用 資源](#)
- [透過 AWS 工具組使用 Amazon S3](#)
- [透過 AWS Toolkit 來使用 AWS 無伺服器應用程式](#)
- [與 Amazon 合作 CodeCatalyst](#)
- [在 AWS Cloud9 IDE 中使用 Amazon ECR](#)

導覽和設定 AWS 工具組

您可以透過下列 AWS 工具組界面元素存取資源並修改設定：

- [AWS Explorer 視窗](#)：從不同 AWS 區域 存取 AWS 服務。
- [AWS Toolkit \(工具組\) 選單](#)：建立和部署無伺服器應用程式、顯示或隱藏 AWS 區域、存取使用者協助功能，以及與 Git 儲存庫互動。
- [AWS Configuration \(AWS 組態\) 窗格](#)：修改設定，變更您在 AWS 工具組中與 AWS 服務 互動的方式。

透過 AWS Explorer 來使用多個區域的服務和資源

AWS Explorer 視窗可讓您選取 AWS 服務，並使用與該服務相關聯的特定資源。在 AWS Explorer 中，選擇服務名稱節點 (例如，API Gateway 或 Lambda)。然後，選擇與該服務相關的特定資源 (例如，REST API 或 Lambda 函數)。選擇特定資源時，選單會顯示可用的互動選項，例如上傳或下載、叫用或複製。

請考量下列範例。如果您的 AWS 帳戶 憑證可以存取 Lambda 函數，請展開針對 AWS 區域 列出的 Lambda 函數，然後選取可以叫用或以程式碼形式上傳至 AWS Cloud9 IDE 的特定 Lambda 函數。您也可以按節點的標題上按一下滑鼠右鍵，開啟內容選單，開始建立使用 AWS Serverless Application Model 的應用程式。

Note

如果您看不到整合式開發環境 (IDE) 中檢視 AWS Explorer 視窗的選項，請確認您已啟用 AWS 工具組。確認啟用後，請再試一次。如需更多詳細資訊，請參閱 [啟用 AWS Toolkit](#)。

AWS Explorer 視窗也可以顯示託管在多個 AWS 區域的服務。

如何存取所選區域中的 AWS 服務

1. 在 AWS Explorer 視窗中，選擇 Toolkit (工具組) 選單、Show region in the Explorer (在 Explorer 中顯示區域)。
2. 從 Select a region to show in the AWS Explorer (選取要顯示在 AWS Explorer 的區域) 清單中，選擇 AWS 區域。

所選的區域會新增至 AWS Explorer 視窗。若要存取可用的服務和資源，請選擇地區名稱前的箭頭 (>)。

Note

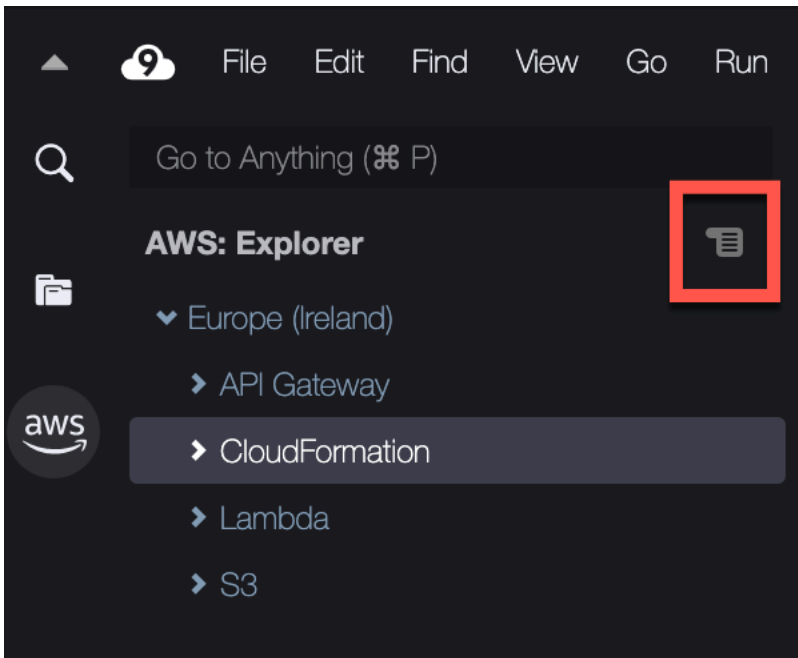
您也可以使用下列選項在 AWS Explorer 視窗隱藏所選 AWS 區域：

- 開啟 (按一下右鍵) 區域的內容選單，然後選擇 Hide region from the Explorer (從 Explorer 隱藏區域)。
- 在 AWS 工具組選單中，選擇 Hide region from the Explorer (在 Explorer 中隱藏區域)，然後選取要隱藏的區域。

存取和使用 AWS 工具組選單

AWS 工具組可供存取建立和部署[無伺服器應用程式](#)的選項。您可以使用該選單管理連線、更新 AWS: Explorer 視窗、存取文件，以及與 GitHub 儲存庫互動。

若要存取 Toolkit (工具組) 選單，請選擇 AWS Explorer 視窗中 AWS: Explorer 標題對面的捲動圖示。



下表提供 Toolkit (工具組) 選單中可用選項的概觀。

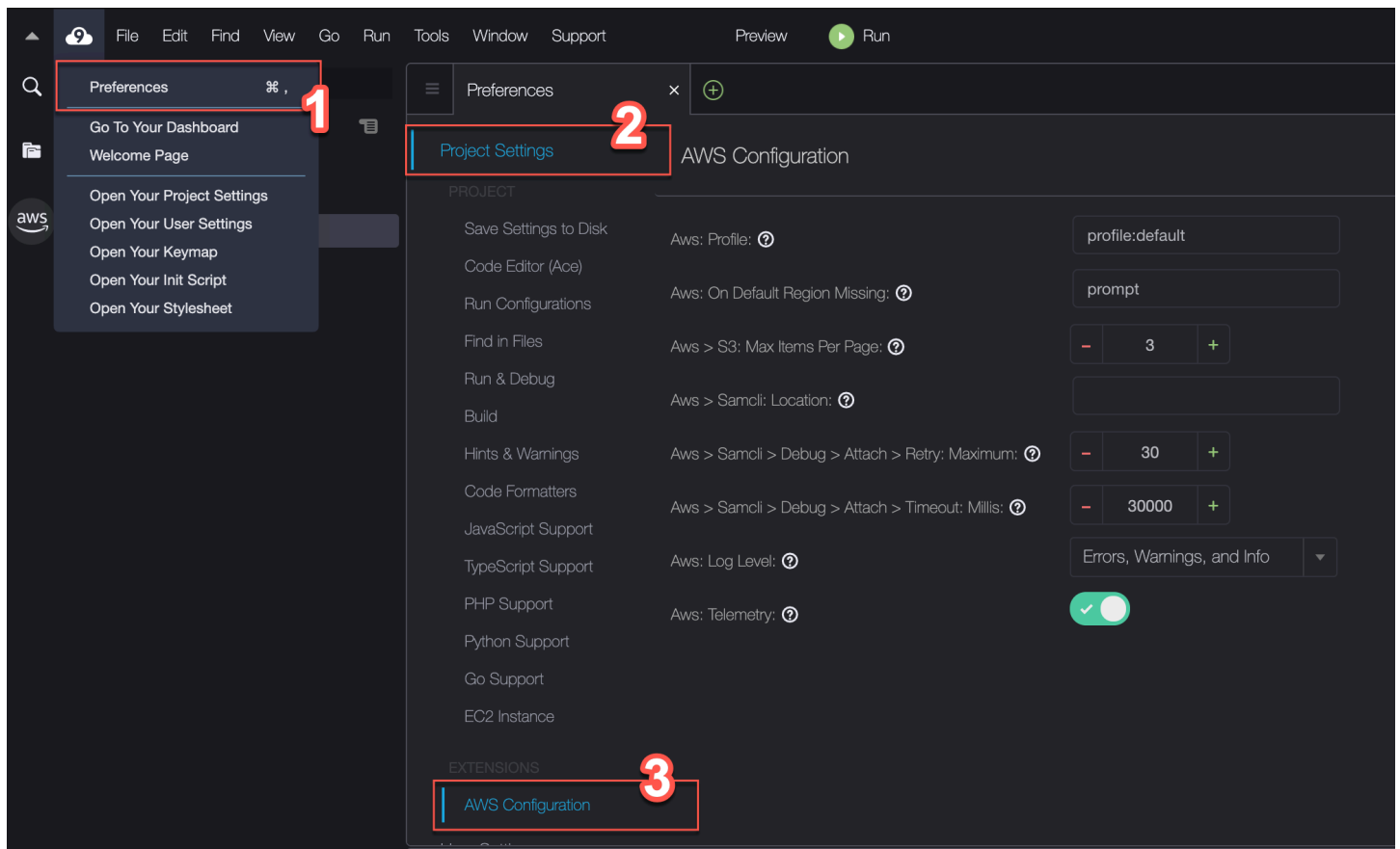
Toolkit (工具組) 選單選項

選單選項	描述
重新整理 AWS Explorer	選擇此選項會重新整理 AWS Explorer，以便顯示您在上次開啟視窗以來修改過的任何 AWS 服務。
連線到 AWS	將 AWS 工具組連線到使用存放在設定檔中的憑證的 AWS 帳戶。如需更多詳細資訊，請參閱 管理 AWS Toolkit 的存取憑證 。
在 Explorer 中顯示區域	在 AWS Explorer 視窗顯示 AWS 區域。如需更多詳細資訊，請參閱 透過 AWS Explorer 來使用多個區域的服務和資源 。
在 Explorer 中隱藏區域	在 AWS Explorer 視窗隱藏 AWS 區域。如需詳細資訊，請參閱 透過 AWS Explorer 來使用多個區域的服務和資源 。
建立新的 SAM 應用程式	產生一組程式碼檔案供新的 AWS 無伺服器應用程式使用。如需建立和部署 SAM 應用程式的詳

選單選項	描述
	細資訊，請參閱 透過 AWS Toolkit 來使用 AWS 無伺服器應用程式 。
部署 SAM 應用程式	在 AWS 中部署無伺服器應用程式。如需建立和部署 SAM 應用程式的詳細資訊，請參閱 透過 AWS Toolkit 來使用 AWS 無伺服器應用程式 。
檢視快速入門	開啟 Quick Start 指南。
檢視工具組文件	開啟 AWS 工具組使用者指南。
在 GitHub 上檢視來源	開啟 AWS 工具組的 GitHub 儲存庫。
在 GitHub 上建立新議題	在 Github 上開啟 AWS 工具組的「新議題」頁面
提交快速意見回饋	提交單向的私人意見回饋至 AWS 工具組開發團隊。如有需要討論或錯誤修正的問題，請在 Github 中選取 Create a New Issue on Github (在 Github 上建立新議題) 選單選項來提交問題。
關於 AWS 工具組	顯示執行中的工具組版本資訊以及搭配設定的 Amazon 作業系統相關資訊。

使用 AWS 組態窗格修改 AWS 工具組設定

若要存取 AWS Configuration (AWS 組態) 窗格中，請選擇 AWS Cloud9、Preferences (偏好設定)。接下來，在 Preferences (偏好設定) 視窗的 Project Settings (專案設定) 中，選擇 AWSConfiguration (組態)。



下表提供 AWS Configuration (AWS 組態) 窗格中可用選項的概觀。

選單選項	描述
AWS : 設定檔	針對可從中取得憑證的憑證設定檔設定名稱。
AWS: On Default Region Missing (AWS : 缺少預設區域)	<p>在所選憑證設定檔的預設 AWS 區域 並未顯示在 AWS Explorer 視窗中的情況下，指出應採取的行動。您可以選擇以下三個選項：</p> <ul style="list-style-type: none"> prompt (提示)(預設)：系統會詢問您想要做什麼。 add (新增)：該區域會顯示在 AWS Explorer 視窗中。 ignore (忽略)：不採取任何動作。
AWS > S3: Max Items Per Page (AWS > S3 : 每頁項目上限)	指定 AWSExplorer 視窗中一次顯示多少 Amazon S3 物件或資料夾。顯示最大數量時，

選單選項	描述
	<p>您可以選擇 Load More (載入更多) 以顯示下一個批次。</p> <p>此欄位接受的值範圍介於 3 到 1000 之間。此設定僅適用於一次顯示的物件或資料夾數量。您建立的所有儲存貯體都會一次顯示。每個 AWS 帳戶預設最多可以建立 100 個儲存貯體。</p>
AWS > Samcli: Location (AWS > Samcli : 位置)	指出 SAM CLI 用來建立、建置、封裝及部署 無伺服器應用程式 的位置。
AWS > Samcli > Debug > Attach > Retry: Maximum: (AWS > Samcli > 除錯 > 連接 > 重試 : 上限 :)	<p>指定工具組會嘗試連接 SAM CLI 除錯器幾次再放棄。預設配額為嘗試 30 次。</p> <p>如果您在 AWS SAMCL 中以除錯模式透過本機方式叫用 Lambda 函數時，您可以將除錯器連接到函數。</p>
AWS > Samcli > Debug > Attach > Timeout: Millis: (AWS > Samcli > 除錯 > 連接 > 逾時 : 毫秒 :)	<p>指定工具組會嘗試連接 SAM CLI 除錯器多久再放棄。預設逾時時間為 30,000 毫秒 (30 秒)。</p> <p>如果您在 AWS SAMCL 中以除錯模式透過本機方式叫用 Lambda 函數時，您可以將除錯器連接到函數。</p>
AWS: Log Level: (AWS : 日誌層級 :)	<p>設定要記錄的工作流程事件類別。以下為可用的層級：</p> <ul style="list-style-type: none"> • Errors Only (僅限錯誤) • Errors and Warnings (錯誤和警告) • Errors, Warnings, and Info (錯誤、警告和資訊) (預設選項) • Errors, Warnings, and Info, Verbose, and Debug (錯誤、警告和資訊、詳細資訊和除錯)
AWS: Telemetry (AWS : 遙測)	啟用或停用將使用資料傳送至 AWS 的功能。預設為啟用

透過 AWS 工具組使用 API Gateway

API Gateway 可讓您建立 RESTful API 和 WebSocket API，藉此啟用即時雙向通訊應用程式。如需詳細了解如何使用 API Gateway 建立和管理 API，請參閱 [API Gateway 開發人員指南](#)。

透過 AWS 工具組，您可以指定 REST 資源、方法類型和傳入作為輸入的資料，藉此設定 REST API 的呼叫。

在 API Gateway 中叫用 REST API

Important

使用 AWS 工具組叫用 API 方法可能會對資源造成無法復原的變更。舉例來說，如果您呼叫 POST 方法，只要呼叫成功，API 的資源就會更新。

您可以從 AWS 工具組針對 AWS 叫用 API Gateway。

叫用 REST API

1. 在 AWS Explorer 視窗中，選擇 API Gateway 節點以檢視目前 AWS 區域中的可用 REST API 清單。
2. 以滑鼠右鍵按一下 REST API，然後選擇 Invoke on AWS (在 AWS 叫用)。

Note

您可以使用內容選單，複製 REST API 的 URL、名稱和 Amazon Resource Name (ARN)。

Invoke methods (叫用方法) 視窗會顯示。您可以設定對 API 的呼叫。

3. 針對 Select a resource (選取資源)，請選擇您要互動的 REST 資源。
4. 針對 Select a method (選取方法)，請選擇下列其中一個方法類型：
 - GET：從透過 API 存取的後端服務取得資源。
 - OPTIONS：針對 API Gateway 所支援的方法和作業請求相關資訊。
 - POST：在透過 API 存取的後端服務上建立新資源。
5. 若要提供 API 方法呼叫的輸入，您可以使用查詢字串或 JSON 格式的酬載：

- Query string (查詢字串)：使用格式 `parameter1=value1¶meter2=value2` 輸入查詢字串。(在使用查詢字串之前，請先建立[對應範本](#)，以便在將傳入的 Web 請求傳送到整合後端之前先完成轉換。)
- JSON 格式：您可以在 Invoke methods (呼叫方法) 視窗中的大型文字欄位中定義 JSON 格式的酬載。

舉例來說，您可以使用包含下列酬載的 POST 方法新增資源：

```
{"type": "soda", "price" : 3.99}
```

6. 選擇 Invoke (叫用) 按鈕呼叫 REST API 資源。

REST API 回應會顯示在 AWS Remote Invocations (AWS 遠端叫用) 索引標籤中。回應的內文包含 JSON 格式的資源資料。

使用 AWS App Runner 和 AWS Toolkit

[AWS App Runner](#) 提供快速且符合成本效益的方式，可在 AWS 雲端中直接將原始碼或容器映像部署到可擴展的安全 Web 應用程式上。使用時，您不需要學習新技術、決定要使用哪種運算服務，也無需了解如何佈建和設定 AWS 資源。

您可以使用 AWS App Runner 來建立和管理以來源映像或原始碼為基礎的服務。如果您使用來源映像，則可以選擇存放在映像儲存庫中的公有或私有容器映像。App Runner 支援下列映像儲存庫提供者：

- Amazon Elastic Container Registry (Amazon ECR)：將私有映像存放在您的 AWS 帳戶中。
- Amazon Elastic Container Registry Public (Amazon ECR Public)：存放公有的可讀取映像。

如果您選擇原始碼選項，則可以從由支援的儲存庫提供者所維護的原始碼儲存庫進行部署。目前，App Runner 支持[GitHub](#)作為源代碼存儲庫提供程序。

必要條件

若要使用 AWS Toolkit 與 App Runner 互動，則需要下列項目：

- 一個 AWS 帳戶
- 具備 AWS App Runner 功能的 AWS Toolkit 版本

除了這些核心需求之外，請確定所有相關的 IAM 使用者都具有與 App Runner 服務互動的許可。也請務必取得有關服務來源的特定資訊，例如容器映像 URI 和 GitHub 存放庫的連線。當您建立 App Runner 服務時需要此資訊。

設定 App Runner 的 IAM 許可

若要快速授與 App Runner 所需的許可，請將現有的 AWS 受管政策連接到相關的 AWS Identity and Access Management (IAM) 實體。具體而言，您可以將政策連接到使用者或群組。App Runner 提供兩個您可以連接至 IAM 使用者的受管政策：

- `AWSAppRunnerFullAccess`：允許使用者執行所有 App Runner 動作。
- `AWSAppRunnerReadOnlyAccess`：允許使用者列出和檢視 App Runner 資源的詳細資訊。

如果您從 Amazon Elastic Container Registry (Amazon ECR) 選擇私有儲存庫做為服務來源，則必須為 App Runner 服務建立下列存取角色：

- `AWSAppRunnerServicePolicyForECRAccess`：允許 App Runner 存取您帳戶中的 Amazon Elastic Container Registry (Amazon ECR) 映像。

使用 AWS Toolkit 命令窗格設定服務執行個體時，您可以自動建立此角色。

Note

`AWSServiceRoleForAppRunner` 服務連結角色可 AWS App Runner 讓您完成下列工作：

- 將日誌推送到 Amazon CloudWatch 日誌日誌群組。
- 創建 Amazon CloudWatch 活動規則以訂閱 Amazon Elastic Container Registry (Amazon ECR) 映像推送。

您不需要手動建立 服務連結角色。當您在 AWS Management Console 中建立 AWS App Runner 或使用 AWS Toolkit 所呼叫的 API 作業時，AWS App Runner 會為您建立此服務連結角色。

如需詳細資訊，請參閱《AWS App Runner 開發人員指南》中的「[App Runner 的 Identity and Access Management](#)」。

取得 App Runner 的服務來源

您可以使用 AWS App Runner 從來源映像或原始碼來部署服務。

Source image

若您從來源映像進行部署，請從私有或公有 AWS 映像登錄來為該映像取得到儲存庫的連結。

- Amazon ECR 私有登錄：複製使用 Amazon ECR 主控台之私有儲存庫的 URI，網址為 <https://console.aws.amazon.com/ecr/repositories>。
- Amazon ECR 公有登錄：複製使用 Amazon ECR Public Gallery (Amazon ECR 公有映像庫) 的公有儲存庫的 URI，網址為 <https://gallery.ecr.aws/>。

Note

您也可以直接從 AWS Toolkit 中的 AWS Explorer 來取得私有 Amazon ECR 儲存庫的 URI：

- 開啟 AWS Explorer，然後展開 ECR 節點，以檢視該 AWS 區域的儲存庫清單。
- 開啟 (按一下右鍵) 儲存庫的內容選單，然後選擇 Copy Repository URI (複製儲存庫 URI) 將連結複製到剪貼簿。

您可以在使用 AWS Toolkit 命令窗格來設定服務執行個體時，指定映像儲存庫的 URI。

如需詳細資訊，請參閱 AWS App Runner 開發人員指南中的「[以來源映像為基礎的 App Runner 服務](#)」。

Source code

為了將您的原始碼部署到 AWS App Runner 服務，該原始碼必須存放在 Git 儲存庫中。此 Git 儲存庫必須由受支援的儲存庫提供者進行維護。應用程式運行器支持一個源代碼庫提供商：[GitHub](#)。

如需有關設定 GitHub 存放庫的資訊，請參閱上的[入門文件](#) GitHub。

要從 GitHub 儲存庫將源代碼部署到應用程式運行器服務，應用程式運行器建立了一個連接到 GitHub。如果您的儲存庫是私有的 (也就是說，它在上不可公開訪問 GitHub)，則必須向 App Runner 提供連接詳細信息。

Important

若要建立 GitHub 連線，您必須使用應用程式執行器主控台 (<https://console.aws.amazon.com/apprunner>) 建立連結的 GitHub 連線AWS。使用 AWS Toolkit 的命令窗格設定服務執行個體時，您可以選取GitHub連線頁面上可用的連線。

如需詳細資訊，請參閱 AWS App Runner 開發人員指南中的「[管理 App Runner 連線](#)」。

App Runner 服務執行個體提供受管的執行時間，允許您的程式碼建置和執行。AWS App Runner 目前支援下列執行時間：

- Python 受管的執行時間
- Node.js 受管的執行時間

做為服務設定的一部分，您會提供 App Runner 服務如何建置和開始服務的相關資訊。您可以使用 Command Palette (命令選擇區) 或指定一個 YAML 格式化的 [App Runner 組態檔案](#) 來輸入此資訊。此檔案中的值會指示 App Runner 如何建置和開始您的服務，並提供執行時間內容。這包括相關的網路設定和環境變數。組態檔案命名為 `apprunner.yaml`。它會自動新增到您應用程式儲存庫的根目錄中。

定價

您需支付應用程式使用的運算和記憶體資源的費用。此外，如果您將部署自動化，您也要針對每個涵蓋當月所有自動化部署的應用程式來支付固定月費。如果您選擇從原始碼進行部署，您需要根據 App Runner 從原始碼建置容器所需的時間來支付建置費用。

如需詳細資訊，請參閱 [AWS App Runner 定價](#)。

主題

- [建立 App Runner 服務](#)
- [管理 App Runner 服務](#)

建立 App Runner 服務

您可以使用AWS Explorer 在 AWS Toolkit 中建立 App Runner 服務。在您選擇在特定 AWS 區域 建立服務後，AWS Toolkit 命令窗格會引導您設定應用程式執行之服務執行個體。

在建立 App Runner 服務之前，請務必先完成[先決條件](#)。這包括提供相關的 IAM 許可，以及確認您要部署的特定來源儲存庫。

建立 App Runner 服務

1. 開啟 AWS Explorer (如尚未開啟的話)。
2. 在 App Runner 節點上按一下滑鼠右鍵並選擇 Create Service (建立服務)。

AWS Toolkit 命令窗格隨即顯示。

3. 對於 Select a source code location type (選取原始碼位置類型)，請選擇 ECR 或 Repository (儲存庫)。

如果選擇 ECR，您可以在由 Amazon Elastic Container Registry (Amazon ECR) 維護的儲存庫中指定容器映像。如果選擇 Repository (儲存庫)，您可以指定由支援的儲存庫提供者維護的原始碼儲存庫。目前，App Runner 支持[GitHub](#)作為源代碼儲存庫提供程序。

從 ECR 部署

1. 若為 Select or enter an image repository (選取或輸入映像儲存庫)，請選擇或輸入由您的 Amazon ECR 私有登錄或 Amazon ECR 公有映像庫維護的映像儲存庫 URL。

Note

若您從 Amazon ECR Public Gallery (Amazon ECR 公有映像庫) 指定儲存庫，請確認自動部署已關閉。App Runner 不支援 ECR 公有映像庫中映像的自動部署。

自動部署預設為關閉狀態。當命令窗格標頭上的圖示出現斜跨圖示的對角線時即表示此狀態。如果您選擇開啟自動部署，則會顯示一則訊息，通知您此選項可能產生額外費用。

2. 如果命令窗格中的步驟報告 No tags found (找不到標籤)，請返回上一個步驟來選取包含標記容器映像的儲存庫。
3. 對於 Port (連接埠)，請輸入服務所使用的 IP 連接埠 (例如：連接埠 8000)。
4. (可選) 對於 Configure environment variables (設定環境變數)，請指定包含用來自訂服務執行個體中行為的環境變數的檔案。
5. 如果您使用的是 Amazon ECR 私有登錄，則需要 AppRunnerECR AccessRole ECR 存取角色。此角色允許 App Runner 存取您帳戶中的 Amazon Elastic Container Registry (Amazon ECR) 映像。選擇命令窗格標頭上的「+」圖示，即可建立此角色。如果您的映像存放在 Amazon ECR 公有 (其映像可公開存取) 中，則不需要存取角色。

6. 對於 Name your service (為您的服務命名)，請輸入唯一的名稱，然後按下 Enter 鍵。名稱不可含有空格。
7. 對於 Select instance configuration (選取執行個體組態)，請為您的服務執行個體選擇 CPU 單位和記憶體組合 (皆以 GB 為單位)。

建立您的服務時，其狀態會從建立中變更至執行中。

8. 服務開始執行後，在該服務上按一下滑鼠右鍵，開啟內容選單，然後選擇 Copy Service URL (複製服務 URL)。
9. 若要存取您部署的應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

從遠端儲存庫部署

1. 對於選取連線，請選擇連結的 GitHub 連線 AWS。可供選取的連線會列在 App Runner 主控台的 GitHub 連線頁面上。
2. 針對選取遠端 GitHub 存放庫，選擇或輸入遠端存放庫的 URL。

有已設定 AWS Cloud9 來源控制管理的遠端儲存庫可供選擇。如果沒有列出儲存庫的連結，您也可以貼上該儲存庫的連結。

3. 對於 Select a branch (選取分支)，請選擇您要部署之原始碼中的 Git 分支。
4. 對於 Choose configuration source (選擇組態來源)，請指定您要如何定義執行時間組態。

如果選擇 Use configuration file (使用組態檔案)，則您的服務執行個體是由 `apprunner.yaml` 組態檔案所定義之設定來設定。此檔案位於應用程式儲存庫的根目錄中。

如果選擇 Configure all settings here (在此處調整所有設定)，請使用命令窗格來指定下列項目：

- Runtime (執行時間)：選擇 Python 3 或 Nodejs 12。
 - Build command (建置命令)：輸入命令以在服務執行個體的執行階段環境中建置應用程式。
 - Start command (開始命令)：輸入命令以在服務執行個體的執行階段環境中啟動應用程式。
5. 對於 Port (連接埠)，請輸入服務所使用的 IP 連接埠 (例如：連接埠 8000)。
 6. 對於 Configure environment variables (設定環境變數)，您可以指定包含用來自訂服務執行個體中行為的環境變數的檔案。
 7. 對於 Name your service (為您的服務命名)，請輸入唯一的名稱，然後按下 Enter 鍵。名稱不可含有空格。
 8. 對於 Select instance configuration (選取執行個體組態)，請為您的服務執行個體選擇 CPU 單位和以 GB 為單位的記憶體的組合。

建立您的服務時，其狀態會從建立中變更至執行中。

9. 服務開始執行後，在該服務上按一下滑鼠右鍵，然後選擇 Copy Service URL (複製服務 URL)。
10. 若要存取您部署的應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

Note

如果您嘗試建立 App Runner 服務失敗，服務會在 AWS Explorer 中顯示建立失敗狀態。如需疑難排解詳細資訊，請參閱《App Runner 開發人員指南》中的[建立服務失敗時](#)。

管理 App Runner 服務

建立 App Runner 服務之後，您可以使用 AWS Explorer 窗格來執行下列活動以管理服務：

- [暫停和繼續 App Runner 服務](#)
- [部署 App Runner 服務](#)
- [檢視 App Runner 的日誌串流](#)
- [刪除 App Runner 服務](#)

暫停和繼續 App Runner 服務

如果您需要暫時停用 Web 應用程式並停止執行程式碼，您可以暫停 AWS App Runner 服務。App Runner 會將服務的運算容量縮減為零。當您準備好再次執行應用程式時，請繼續您的 App Runner 服務。App Runner 會佈建新的運算容量、將您的應用程式部署到該容量，並執行應用程式。

Important

只有在執行 App Runner 時，才會向您收取費用。因此，您可以視需要暫停及繼續應用程式以管理成本。這對開發和測試情境特別有幫助。

暫停您的 App Runner 服務

1. 開啟 AWS Explorer (如尚未開啟的話)。
2. 展開 App Runner 以檢視服務清單。

3. 在服務上按一下滑鼠右鍵並選擇 Pause (暫停)。
4. 在顯示的對話方塊中，選擇 Confirm (確認)。

服務暫停時，服務狀態會從 Running (執行中) 變更至 Pausing (暫停中)，然後到 Paused (已暫停)。

繼續您的 App Runner 服務

1. 開啟 AWS Explorer (如尚未開啟的話)。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Resume (繼續)。

服務繼續時，服務狀態會從 Resuming (繼續中) 變更為 Running (執行中)。

部署 App Runner 服務

如果您為服務選擇手動部署選項，則需要明確起始每個服務的部署。

1. 開啟 AWS Explorer (如尚未開啟的話)。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Start Deployment (開始部署)。
4. 部署應用程式時，服務狀態會從 Deploying (部署中) 變更為 Running (執行中)。
5. 若要確認您的應用程式已成功部署，請在相同的服務上按一下滑鼠右鍵並選擇 Copy Service URL (複製服務 URL)。
6. 若要存取您部署的 Web 應用程式，請將複製的 URL 貼上至您 Web 瀏覽器的網址列中。

檢視 App Runner 的日誌串流

使用 CloudWatch 日誌來監視，存儲和訪問您的日誌流服務，例如應用程序運行器。日誌串流是共享相同來源的一系列日誌事件。

1. 展開 App Runner 以檢視服務執行個體清單。
2. 展開特定服務執行個體以檢視日誌群組清單。(日誌群組是共享相同保留、監控和存取控制設定的日誌串流群組。)
3. 在日誌群組上按一下滑鼠右鍵並選擇 View Log Streams (檢視日誌串流)。

4. 從命令窗格中，選擇群組中的日誌串流。

此 AWS Cloud9 IDE 會顯示組成串流的日誌事件清單。您可以選擇將較舊或較新的事件載入編輯器中。

刪除 App Runner 服務

Important

如果您刪除 App Runner 服務，系統會永久移除該服務，並刪除存放的資料。當您需要重新建立服務時，若來源為程式碼儲存庫，則 App Runner 需要再次擷取您的來源並建置它。您的 Web 應用程式 取得一個新的 App Runner 網域。

1. 開啟 AWS Explorer (如尚未開啟的話)。
2. 展開 App Runner 以檢視服務清單。
3. 在服務上按一下滑鼠右鍵並選擇 Delete Service (刪除服務)。
4. 在 AWS Toolkit 命令窗格中，輸入 delete 然後按下 Enter 鍵以確認。

刪除的服務會顯示 Deleting (刪除中) 狀態，然後服務就會從清單中消失。

透過 AWS 工具組使用 AWS CloudFormation 堆疊

AWS 工具組支援使用 [AWS CloudFormation](#) 堆疊。透過 AWS 工具組，您可以刪除 AWS CloudFormation 堆疊。

刪除 AWS CloudFormation 堆疊

您可以使用 AWS 工具組來檢視和刪除 AWS CloudFormation 堆疊。

先決條件

- 確認您在 AWS Cloud9 環境中使用的憑證包括 AWS CloudFormation 服務的適當讀取/寫入存取權。如果在 AWS Explorer 的 CloudFormation 下方，您看到類似「Error loading CloudFormation resources」(載入 CloudFormation 資源時發生錯誤) 的訊息，請檢查與這些憑證連接的許可。您對這些許可所做的變更需要幾分鐘的時間才會對 AWS Explorer 生效。

刪除 AWS CloudFormation 堆疊

1. 在 AWS Explorer 中，開啟 (按一下右鍵)您要刪除之 AWS CloudFormation 堆疊的內容選單。
2. 選擇 Delete CloudFormation Stack (刪除 CloudFormation 堆疊)。
3. 在出現的訊息中，選擇 Yes (是)，以確認刪除。

刪除堆疊後，該堆疊不會再列於 AWS Explorer 之中。

透過 AWS 工具組使用 CloudWatch Logs

Amazon CloudWatch Logs 可將您使用的所有系統、應用程式和 AWS 服務的日誌，集中在可高度擴展的單一服務中。然後，您可以輕鬆檢視日誌、在日誌中搜尋特定的錯誤碼或模式、根據特定欄位篩選日誌，或安全封存日誌以供日後分析。如需更多詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的 [什麼是 Amazon CloudWatch Logs ?](#)。

下列主題說明如何使用 AWS 工具組來使用 AWS 帳戶中的 CloudWatch Logs。

主題

- [使用 AWS 工具組檢視 CloudWatch 日誌群組和日誌串流](#)
- [透過 AWS 工具組使用日誌串流中的 log](#)

使用 AWS 工具組檢視 CloudWatch 日誌群組和日誌串流

日誌串流是一系列共用相同來源的日誌事件。每個單獨日誌串流是由 CloudWatch Logs 中的單獨日誌來源所組成。

日誌群組是共用相同保留、監控和存取控制設定的日誌串流群組。您可以定義日誌群組，並指定放入每個群組的串流。可以屬於一個日誌群組的日誌串流數量並沒有限制。

如需詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的 [使用日誌群組和日誌串流](#)。

主題

- [以 CloudWatch Logs 節點檢視日誌群組和日誌串流](#)

以 CloudWatch Logs 節點檢視日誌群組和日誌串流

1. 開啟 AWS Explorer (如尚未開啟的話)。

2. 按一下 CloudWatch Logs 節點展開日誌群組清單。

目前 AWS 區域 的日誌群組會顯示在 CloudWatch Logs 節點下方。

3. 若要檢視特定日誌群組中的日誌串流，請在日誌群組名稱上按一下滑鼠右鍵，然後選擇 View Log Streams (檢視日誌串流)。
4. 日誌群組的內容會顯示在 Select a log stream (選取日誌串流) 標題底下。

您可以從清單中選擇特定的串流，或在欄位中輸入文字來篩選串流。

選擇資料流之後，該資料流中的事件會顯示在 IDE 的 Log Streams (日誌串流) 視窗。如需了解如何與每個串流中的日誌事件互動，請參閱 [使用CloudWatch log 事件](#)。

透過工AWS具CloudWatch組使用日誌串流中的 log

開啟 [記錄串流] 視窗之後，您可以存取每個串流中的記錄事件。日誌事件是由應用程式或正受到系統監控的資源所記錄的活動記錄。

主題

- [檢視並複製日誌串流資訊](#)
- [將日誌串流編輯器的內容儲存至本機檔案](#)

檢視並複製日誌串流資訊

開啟日誌串流時，Log Stream (日誌串流)視窗會顯示該串流の日誌事件序列。

1. 若要找到要檢視的日誌串流，請開啟 Log Stream (日誌串流) 視窗。如需詳細資訊，請參閱[檢視 CloudWatch 日誌群組和日誌串流](#)。

列出事件的每一行都會加上時間戳記，藉此顯示事件的記錄時間。

2. 您可以使用下列選項檢視和複製串流事件的相關資訊：
 - 依時間檢視事件：選擇 Load newer events (載入較新的事件) 或 Load older events (載入較舊的事件)，可顯示最新與較舊的日誌事件。

Note

Log Stream (日誌串流)編輯器一開始會批次載入最近 10,000 行的日誌事件或 1 MB 的日誌資料，以較小者為優先。如果選擇 Load newer events (載入較新的事件)，編輯器

會顯示上一批次載入後記錄的事件。如果選擇 Load older events (載入較舊的事件)，編輯器會顯示目前顯示的事件之前發生的批次事件。

- 複製日誌事件：選取要複製的事件，然後按一下右鍵，從內容選單中選取 Copy (複製)。
- 複製日誌串流的名稱：以滑鼠右鍵按一下 Log Stream (日誌串流) 視窗的內容選單，然後選擇 Copy Log Stream Name (複製日誌串流名稱)。

將日誌串流編輯器的內容儲存至本機檔案

您可以將 CloudWatch log 串流編輯器的內容下載到本機中的 log 檔案。

Note

此選項可讓您僅將目前顯示在日誌串流編輯器中的日誌事件儲存至檔案。舉例來說，假設日誌串流的總大小為 5MB，而只有 2MB 載入編輯器。您儲存的檔案也只會包含 2MB 的日誌資料。若要顯示更多要儲存的資料，請在編輯器中選擇 Load newer events (載入較新的事件) 或 Load older events (載入較舊的事件)。

1. 若要找到要複製的日誌串流，請開啟 Log Stream (日誌串流) 視窗 (請參閱[檢視 CloudWatch 日誌群組和日誌串流](#))。
2. 按一下右鍵，開啟 Log Stream (日誌串流) 視窗索引標籤的內容選單，然後選擇 Save Current Log Content to File (將目前的日誌內容儲存至檔案)
3. 透過對話方塊選取或建立日誌檔案的下載資料夾，然後選擇 Save (儲存)。

透過 AWS Toolkit 使用 AWS Lambda 函式

AWS Toolkit 支援 [AWS Lambda](#) 函數。AWS Toolkit 取代了先前由 AWS Cloud9 的 Lambda 外掛程式所提供的功能。透過 AWS Toolkit，您可以為[無伺服器應用程式](#)中的 Lambda 函式編寫程式碼。此外，您可以在本機或 AWS 上叫用 Lambda 函式。

Lambda 是一種完整受管的運算服務，可執行程式碼回應自訂程式碼所產生的事件，或執行來自各種 AWS 服務的程式碼。其中包括 Amazon Simple Storage Service (Amazon S3)、Amazon DynamoDB、Amazon Kinesis、Amazon Simple Notification Service (Amazon SNS) 和 Amazon Cognito。

Important

如果您想要建置 Lambda 應用程式來使用無伺服器應用模型 (SAM) 所提供的資源，請參閱 [透過 AWS Toolkit 來使用 AWS 無伺服器應用程式](#)。

主題

- [叫用遠端 Lambda 函式](#)
- [下載、上傳並刪除 Lambda 函式](#)

叫用遠端 Lambda 函式

您可以使用 AWS Toolkit 以各種方式與 [AWS Lambda](#) 函數互動。

如需有關 Lambda 的詳細資訊，請參閱 [AWS Lambda 開發人員指南](#)。

Note

假設您已使用 AWS Management Console 或其他方式建立 Lambda 函數。您可以從 AWS Toolkit 叫用這些函數。若要使用 AWS Toolkit 建立您可以部署至 AWS Lambda 的新函式，您必須先 [建立無伺服器應用程式](#)。

先決條件

- 請確認您在其中設定的登入資料包括 AWS Lambda 服務的合適讀取/寫入存取權。如果在 AWS Explorer 的 Lambda 下方，您看到類似「Error loading Lambda resources」(載入 Lambda 資源時發生錯誤) 的訊息，請檢查與這些憑證連接的許可。您對這些許可所做的變更需要幾分鐘的時間才會在 AWS Toolkit 中對 AWS Explorer 生效。

叫用 Lambda 函式

Important

使用 AWS Toolkit 叫用 API 方法可能會對資源造成無法復原的變更。舉例來說，如果您叫用 POST 方法，只要呼叫成功，API 的資源就會更新。

您可以使用 AWS Toolkit 在 AWS 叫用 Lambda 函式。

1. 在 AWS Explorer 中，選擇您想要叫用的 Lambda 函式名稱，然後開啟其內容選單。
2. 選擇 Invoke on AWS (在 AWS 叫用)。
3. 在開啟的 Invoke function (叫用函式) 視窗中，針對 Lambda 函式所需的酬載選擇選項。(酬載是您想要提供給 Lambda 函式作為輸入的 JSON。) 您可以選擇 Browse (瀏覽) 來選取要當作酬載的檔案，或使用下拉式欄位來挑選酬載的範本。在本例中，Lambda 函數可能會以字串方式輸入，如文字方塊中所示。

選擇 Invoke (叫用) 來叫用 Lambda 並傳入酬載。

您會在 AWS Lambda 索引標籤看到 Lambda 函數的輸出。

下載、上傳並刪除 Lambda 函式

AWS Toolkit 提供匯入和上傳 Lambda 函式到 AWS Cloud9 IDE 的選項。

下載 Lambda 函式

下載 Lambda 函數，您就可以從 AWS 雲端下載描述函式的專案檔案，並在 AWS Cloud9 IDE 中使用。

如何下載 Lambda 函式

1. 在 AWSExplorer 的 Lambda 節點下，開啟 (按一下右鍵) 函數的內容選單，然後選擇 Download (下載)。
2. 如果系統要求 Select a workspace folder for your new project (為新專案選取工作空間資料夾)，您可以執行以下其中一項操作：
 - 選擇建議的資料夾來建立與 Lambda 專案相同名稱的子資料夾。
 - 選擇 Select a different folder (選取其他資料夾) 開啟對話方塊，為您的專案子資料夾瀏覽並選取其他上層資料夾。

IDE 會開啟新的編輯器視窗。

設定已下載的 Lambda 函式，以便對其執行和除錯

您必須在 `launch.json` 檔案中定義啟動組態，才能執行做為無伺服器應用程式的 Lambda 函數並對其偵錯。在 AWS Management Console 中建立的 Lambda 函數可能未包含於啟動組態中。因此，您可能需要手動新增。

若要將 Lambda 函式新增至啟動組態

1. 下載 Lambda 函數後，請開啟 Environment (環境) 視窗以檢視其資料夾和檔案。
 2. 然後，請檢查您的 Lambda 函式是否包含在 `/home/ec2-user/.c9/launch.json` 檔案內。如果它不存在，請執行以下操作以添加 CodeLens 鏈接到您的函數的代碼：
 1. 開啟定義 Lambda 函數的原始程式碼檔案 (例如 `.js` 或 `.py` 檔案)。然後，檢查是否有 CodeLens 鏈接，你可以用來將你的 lambda 函數添加到 `launch.json` 文件。一個 CodeLens 出現在函數上方，並包括 Add Debug Config 鏈接。
 2. 請選擇 IDE 左側的 Go (前往)(放大鏡圖示)，然後輸入「sam hint」以顯示 AWS: Toggle SAM hints in source files 命令。選擇該命令以執行它。
 3. 關閉 Lambda 原始碼檔案，然後重新開啟它。
 4. 如果 CodeLens 在您重新打開文件後，在源代碼中可用，選擇 Add Debug Config 以新增啟動組態。
 3. 如果您無法新增 CodeLens 即使在切換 SAM 提示選項之後，請執行以下操作以添加啟動配置：
 1. 請選擇 IDE 左側的 Go (前往)(放大鏡圖示)，然後輸入「config」以顯示 AWS: SAM Debug Configuration Editor 命令。選擇該命令以執行它。
 2. SAM Debug Configuration Editor (SAM 除錯組態編輯工具) 會顯示。您可以使用此編輯器來定義啟動組態屬性。如需詳細資訊，請參閱 [使用 SAM 範本執行並除錯無伺服器應用程式中 configuring launch properties](#) 的步驟。
-  Note
- 如果 Lambda 函數沒有 SAM 應用程式所需的 `template.yaml`，您就必須加以新增。如需詳細資訊，請參閱「[建立您的 AWS SAM 範本](#)」。
3. 在編輯器中輸入所需的組態資訊後，啟動組態就會新增至 `launch.json` 檔案。

定義了 Lambda 函數的啟動組態後，可以藉由以下操作來執行它：

1. 選擇 IDE 頂部 Auto (自動) 旁邊的箭頭，然後選取相關的啟動組態。
2. 然後選擇 Run (執行)。

上傳 Lambda 函式

您可以使用本機程式碼更新現有的 Lambda 函式。以這種方式更新程式碼時，並不會使用 AWS Serverless Application Model CLI 進行部署，也不會建立 AWS CloudFormation 堆疊。如此一來，您可以使用 Lambda 支援的任何執行時間來上傳 Lambda 函數。

有幾個界面選項可使用 AWS Toolkit 上傳 Lambda 函式。

從 Environment (環境) 視窗或 Command (命令) 窗格上傳

1. 在您專案檔案的 Environment (環境) 視窗中，在您要上傳的 Lambda 應用程式的 `template.yaml` 上按一下右鍵開啟內容選單，然後選擇 Upload Lambda (上傳 Lambda)。

或是，按下 Ctrl+P 開啟 Go to Anything (移至任何項目) 窗格，然後輸入「lambda」以存取 AWS Upload Lambda (上傳 Lambda) 命令。然後選取它以開始上傳程序。

2. 接下來選擇您要上傳至哪個 AWS 區域。
3. 現在選擇一個選項來上傳 Lambda 函式：

上傳 .zip 存檔

1. 從選單中選擇 ZIP Archive (ZIP 存檔)。
2. 從 AWS Cloud9 檔案系統選擇 .zip 檔案，然後選擇 Open (開啟)。

直接上傳目錄

1. 從選單中選擇 Directory (目錄)。
2. 從 AWS Cloud9 檔案系統選擇目錄，然後選擇 Open (開啟)。
4. 指定用於處理事件的 Lambda 函式處理常式。當有人叫用您的函數時，Lambda 會執行此處理常式方法。

Note

選擇 Lambda 函式時，您可以從顯示的列表中進行選擇。如果您不知道要選擇哪個函數，可以輸入 Toolkit 中提供的 Lambda 函數的 Amazon Resource Number (ARN)。

出現一個對話框，詢問您是否要將此代碼作為 Lambda 函式的最新版本發佈。選擇 Yes (是) 以確認發佈。

Note

您也可以透過上層資料夾的內容功能表上傳 Lambda 應用程式，方法是在資料夾上按一下滑鼠右鍵開啟，然後選取 Upload Lambda (上傳 Lambda)。系統會自動選取上層資料夾進行上傳。

從 AWS Explorer 上傳

1. 在 AWSExplorer 中，以滑鼠右鍵按一下您要匯入的 Lambda 函數名稱，開啟內容選單。
2. 選擇 Upload Lambda (上傳 Lambda)。
3. 從三個選項中選擇要用於上傳 Lambda 函式的方式。

上傳預先製作的 .zip 存檔

1. 從選單中選擇 ZIP Archive (ZIP 存檔)。
2. 從 AWS Cloud9 檔案系統選擇 .zip 檔案，然後選擇 Open (開啟)。
3. 透過強制回應對話方塊確認上傳。這個動作會直接上傳 .zip 文件，並在部署後立即更新 Lambda。

直接上傳目錄

1. 從選單中選擇 Directory (目錄)。
2. 從 AWS Cloud9 檔案系統選擇目錄，然後選擇 Open (開啟)。
3. 在系統提示建置目錄時選擇 No (否)。
4. 透過強制回應對話方塊確認上傳。這個動作會直接上傳目錄，並在部署後立即更新 Lambda。

建置並上傳目錄

1. 從選單中選擇 Directory (目錄)。
2. 從 AWS Cloud9 檔案系統選擇目錄，然後選擇 Open (開啟)。
3. 在系統提示建置目錄時選擇 Yes (是)。

4. 透過強制回應對話方塊確認上傳。這個動作會使用 AWS SAM CLI 的 `sam build` 命令在目錄中建置程式碼，並在部署後立即更新 Lambda。

部署用於遠端存取的 Lambda 函式

您可將本機功能部署為無伺服器 SAM 應用程式，使其可供遠端使用。

若要將 Lambda 函式部署為 SAM 應用程式

1. 在 AWS Explorer 視窗中，以滑鼠右鍵按一下 Lambda 節點開啟內容選單，並選取 Deploy SAM Application (部署 SAM 應用程式)。
2. 在命令窗格中，選取 [YAML template](#) (YAML 範本)，其可將您的功能定義為無伺服器應用程式。
3. 接下來，選取要用於 Lambda 部署的 Amazon S3 儲存貯體。您也可以選擇建立用於部署的儲存貯體。
4. 現在輸入您部署目標位置的 AWS CloudFormation 堆疊名稱。如果指定現有堆疊，則命令會更新該堆疊。如果指定新的堆疊，則命令會加以建立。

輸入堆疊名稱後，Lambda 函式會開始部署為 SAM 應用程式。成功部署後，即可從遠端使用 SAM Lambda 應用程式。如此一來，您就可以從其他 AWS Cloud9 開發環境下載或叫用它。

如需從新開始建立 Lambda 函式，我們建議遵循 [使用 AWS Toolkit 建立無伺服器應用程式](#) 的步驟。

刪除 Lambda 函式

您也可以使用同一個內容選單 (按一下右鍵) 刪除 Lambda 函數。

Warning

請勿使用此程序刪除與 [AWS CloudFormation](#) 相關聯的 Lambda 函數。例如，在本指南前述步驟中 [建立無伺服器應用程式時](#) 建立的 Lambda 函數。這些函式必須透過 AWS CloudFormation 堆疊刪除。

1. 在 AWS Explorer 中，選擇您要刪除的 Lambda 函數名稱，然後開啟 (按一下右鍵) 其內容選單。
2. 選擇 Delete (刪除)。
3. 在出現的訊息中，選擇 Yes (是)，以確認刪除。

刪除函式後，該函數不會再列於 AWS Explorer 檢視中。

使用 資源

除了存取預設會列在 AWS Explorer 的 AWS 服務，您也可以前往 Resources (資源)，並從數百種資源中選擇要新增至介面的資源。在 AWS 中，資源是您可以使用的實體。一些新增的資源包括 AmazonAppFlow Kinesis Data Streams、AWS IAM 角色、Amazon VPC 和 AmazonCloudFront 分佈。

若要檢視可用資源，請前往 Resources (資源)，然後展開資源類型以列出該類型的可用資源。例如：若您選取 `AWS::Lambda::Function` 資源類型，您可以存取定義不同功能、其屬性及其屬性的資源。

將資源類型新增到 Resources (資源) 後，您便可使用下列方式來與資源互動：

- 檢視此資源類型目前的 AWS 區域 中可用的現有資源清單。
- 檢視描述資源的 JSON 檔案的唯讀版本。
- 複製資源的資源識別符。
- 檢視 AWS 文件，它解釋了用於建構資源模型的資源類型和結構描述 (JSON 和 YAML 格式) 的目的。

存取資源的 IAM 許可

您需要特定的 AWS Identity and Access Management 許可來存取與 AWS 服務 相關聯的資源。例如：IAM 實體 (諸如使用者或角色) 需要 `Lambda` 許可才能存取 `AWS::Lambda::Function` 資源。

除了服務資源的許可外，IAM 實體還需要許可才能允許 AWS Toolkit 呼叫 AWS 雲端控制 API 作業。雲端控制 API 作業允許 IAM 使用者或角色存取和更新遠端資源。

快速授與許可的方法是將受 AWS 管政策 `PowerUserAccess`，亦即受管政策連接至使用 Toolkit 介面呼叫這些 API 作業的 IAM 實體。這個受管政策會授與一系列許可，以執行應用程式開發任務，包括呼叫 API 作業。

如需定義遠端資源上允許 API 作業的特定許可，請參閱 [《AWS 雲端控制 API 使用者指南》](#)。

與現有資源互動

1. 在 AWS Explorer 中，選擇 Resources (資源)。

資源類型清單會顯示在 Resources (資源) 節點下方。

2. 有文件描述定義資源類型範本的語法。若要此文件，請在該資源類型上按一下滑鼠右鍵開啟，然後選擇 View Documentation (檢視文件)。

Note

系統可能會要求您關閉瀏覽器的快顯封鎖程式，以便您可以存取文件頁面。

3. 若要檢視資源類型已存在的資源，請展開該類型的項目。

可用資源清單會顯示在資源類型下方。

4. 若要與特定資源互動，請在其名稱上按一下滑鼠右鍵開啟，然後選擇以下其中一個選項：
 - Copy Identifier (複製識別符)：將特定資源的識別符複製到剪貼簿。例如，AWS::DynamoDB::Table 資源可以使用 TableName 屬性進行識別。
 - Preview (預覽)：檢視描述資源的 JSON 格式範本的唯一讀版本。

透過 AWS 工具組使用 Amazon S3

下列主題說明如何透過 AWS 工具組來使用 AWS 帳戶中的 [Amazon S3](#) 儲存貯體和物件。

主題

- [使用 Amazon S3 儲存貯體](#)
- [使用 Amazon S3 物件](#)

使用 Amazon S3 儲存貯體

您存放在 Amazon S3 的每個物件都位在儲存貯體內。您可以使用儲存貯體將相關物件分組，其方式與使用目錄將檔案系統中的檔案分組相同。

主題

- [建立 Amazon S3 儲存貯體](#)
- [在 Amazon S3 儲存貯體中新增資料夾](#)
- [刪除 Amazon S3 儲存貯體](#)
- [設定 Amazon S3 項目的顯示方式](#)

建立 Amazon S3 儲存貯體

1. 在 AWSExplorer，以滑鼠右鍵按一下 S3 節點開啟內容選單，然後選擇 Create Bucket (建立儲存貯體)。
2. 在 Bucket Name (儲存貯體名稱) 欄位中，輸入有效的儲存貯體名稱。按下 Enter 確認輸入。

新的儲存貯體會顯示在 S3 節點下方。

注意

因為 S3 可讓您的儲存貯體做為可公開存取的 URL 使用，所以您選擇的儲存貯體名稱必須是全域唯一的。如果有其他帳戶已使用您選擇的名稱建立儲存貯體，您則必須使用其他名稱。

如果您無法建立儲存貯體，您可以檢查 Output (輸出) 索引標籤中的 AWS Toolkit Logs (AWS 工具組日誌)。例如，如果您使用的儲存貯體名稱已在使用中，就會發生 BucketAlreadyExists 錯誤。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的[儲存貯體限制與局限](#)。

儲存貯體建立後，您可以將其名稱和 Amazon 資源名稱 (ARN) 複製到剪貼簿。按一下滑鼠右鍵開啟儲存貯體項目的內容選單，然後從選單中選取相關選項。

在 Amazon S3 儲存貯體中新增資料夾

您可以將資料夾中的物件分組，藉此整理儲存貯體的內容。您也可以在其他資料夾中建立資料夾。

1. 在 AWSExplorer 中選擇 S3 節點，檢視儲存貯體清單。
2. 按一下滑鼠右鍵開啟儲存貯體或資料夾的內容選單，然後選擇 Create Folder (建立資料夾)。
3. 輸入 Folder Name (資料夾名稱)，然後按下 Enter 鍵。

新資料夾現在會顯示在 AWS Explorer 視窗中的指定儲存貯體和資料夾下方。

刪除 Amazon S3 儲存貯體

刪除儲存貯體時，您也會刪除其中的資料夾和物件。刪除儲存貯體之前，系統會先詢問您是否確認要執行此動作。

Note

若只要刪除資料夾，而不是整個儲存貯體，請使用 AWS Management Console。

1. 在 AWSExplorer 中選擇 S3 節點，展開儲存貯體清單。
2. 針對要刪除的儲存貯體開啟內容選單，然後選擇 Delete (刪除)。
3. 輸入儲存貯體的名稱以確認刪除，然後按下 Enter (確認)。

Note

如果儲存貯體包含物件，則會在刪除前清空儲存貯體。如果需要刪除上千個物件的每個版本，可能需要一些時間才能完成。刪除程序完成後，會顯示通知。

設定 Amazon S3 項目的顯示方式

如果您要處理大量 Amazon S3 物件或資料夾，指定一次顯示的數量是很實用的方法。顯示最大數量時，您可以選擇 Load More (載入更多) 以顯示下一個批次。

1. 從選單列選擇 AWS Cloud9、Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 視窗中展開 Project Settings (專案設定)，然後前往 EXTENSIONS (擴充功能) 區塊選擇 AWS Configuration (AWS 組態)。
3. 在 AWS Configuration (AWS 組態) 窗格中，前往 AWS > S3: Max Items Per Page (每頁最多項目) 設定。
4. 將預設值變更為您要顯示的 S3 項目數量，然後再選擇載入更多。

Note

系統接受的值範圍介於 3 到 1000 之間。此設定僅適用於一次顯示的物件或資料夾數量。您建立的所有儲存貯體都會一次顯示。每個 AWS 帳戶 預設最多可以建立 100 個儲存貯體。

使用 Amazon S3 物件

物件是存放在 Amazon S3 中的基本實體。物件是由物件資料與中繼資料構成。

主題

- [上傳檔案至 Amazon S3 儲存貯體](#)
- [下載 Amazon S3 物件](#)
- [刪除 Amazon S3 物件](#)
- [為 Amazon S3 物件產生預先簽章的 URL](#)

上傳檔案至 Amazon S3 儲存貯體

您可以使用工具組界面或命令將檔案上傳到儲存貯體

這兩種方法都允許您從使用者的環境上傳檔案，並在 AWS 雲端儲存為 S3 物件。您可以將檔案上傳到儲存貯體，或用於整理儲存貯體內容的資料夾。

使用界面上傳檔案至 S3 儲存貯體

1. 在 AWSExplorer 中選擇 S3 節點，檢視儲存貯體清單。
2. 按一下滑鼠右鍵開啟儲存貯體或該儲存貯體中資料夾的內容選單，然後選擇 Upload File (上傳檔案)。

注意

如果您以滑鼠右鍵按一下 S3 物件開啟內容選單，可以選擇 Upload to Parent (上傳至父項)。這個動作可讓您將檔案新增至資料夾或包含所選檔案的儲存貯體。

3. 使用環境的檔案管理員選取檔案，然後選擇 Upload (上傳)。

選取的檔案會以 S3 物件的形式上傳至儲存貯體或資料夾。每個物件的項目會描述儲存物件的大小，以及上傳物件的時間。您可以暫留在物件的資訊清單上檢視路徑、大小和上次修改時間。

使用命令將目前的檔案上傳至 S3 儲存貯體

1. 若要選取要上傳的檔案，請選擇檔案的索引標籤。
2. 按下 Ctrl+P 以顯示 Commands (命令) 窗格。
3. 在 Go To Anything (移至任何內容) 中，開始輸入 upload file 片語以顯示 AWS: Upload File 命令。命令顯示時請選擇該命令。
4. 在 Step 1: Select a file to upload (步驟 1：選取要上傳的檔案) 中，您可以選擇已選取的檔案或瀏覽其他檔案。

5. 在 Step 2: Select an S3 bucket to upload to (步驟 2：選取要上傳檔案的 S3 儲存貯體) 中，請從清單中選擇儲存貯體。

選取的檔案會以 S3 物件的形式上傳至儲存貯體或資料夾。每個物件的項目會描述儲存物件的大小，以及上傳物件的時間。您可以暫留在物件的資訊清單上檢視路徑、大小和上次修改時間。

下載 Amazon S3 物件

您可以從 AWS 雲端將 Amazon S3 儲存貯體中的物件下載到 AWS Cloud9 環境。

1. 在 AWSExplorer 中選擇 S3 節點，檢視儲存貯體清單。
2. 在儲存貯體或儲存貯體的資料夾中，以右鍵按一下物件開啟內容選單，然後選擇 Download As (下載為)。
3. 使用環境的檔案管理員選取目的地資料夾，輸入檔案名稱，然後選擇 Download (下載)。

下載檔案後，您可以在 AWS Cloud9 開啟。

刪除 Amazon S3 物件

如果物件位於非版本控制的儲存貯體，您就可以永久刪除物件。但如果是啟用版本控制的儲存貯體，刪除請求並不會永久刪除該物件。相反地，Amazon S3 會在儲存貯體中插入刪除標記。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的「[刪除物件版本](#)」。

1. 在 AWSExplorer 中選擇 S3 節點，檢視儲存貯體清單。
2. 在儲存貯體或儲存貯體的資料夾中，以右鍵按一下物件開啟內容選單，然後選擇 Delete (刪除)。
3. 選擇 Delete (刪除)，即可確認刪除。

為 Amazon S3 物件產生預先簽章的 URL

使用預先簽章的 URL 時，物件擁有者可以與他人共享私有的 Amazon S3 物件，方法是授與他人時間限定許可以供下載物件。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的「[使用預先簽章的 URL 來共享物件](#)」。

1. 在 AWSExplorer 中選擇 S3 節點，檢視儲存貯體清單。
2. 於儲存貯體或儲存貯體中的資料夾內，在物件上按一下滑鼠右鍵，然後選擇 Generate Presigned URL (產生預先簽章的 URL)。

3. 在 AWS 工具組命令窗格中，輸入 URL 存取物件時可使用的分鐘數。按下 Enter 鍵 確認輸入。

IDE 底部的狀態可確認物件的預先簽署 URL 已複製到剪貼簿。

透過 AWS Toolkit 來使用 AWS 無伺服器應用程式

AWS Toolkit 支援使用[無伺服器應用程式](#)。使用 AWS Toolkit，您就能建立包含 [AWS Lambda](#) 函數的無伺服器應用程式，然後將該應用程式部署到 AWS CloudFormation 堆疊。

主題

- [建立無伺服器應用程式](#)
- [執行與除錯無伺服器應用程式](#)
- [同步無伺服器應用程式](#)
- [啟用 AWS Toolkit 程式碼濾鏡](#)
- [從 AWS 雲端刪除無伺服器應用程式](#)
- [除錯無伺服器應用程式的組態選項](#)

建立無伺服器應用程式

本範例示範如何使用 AWS Toolkit 建立無伺服器應用程式。如需了解如何執行與除錯無伺服器應用程式，請參閱 [執行與除錯無伺服器應用程式](#)。

建立無伺服器應用程式的必要先決條件包含 AWS SAM CLI 與 AWS CLI。這些都包含在 AWS Cloud9 當中。如果未安裝 AWS SAM CLI 或已過期，則可能需要執行安裝或升級。如需有關如何安裝 AWS SAM CLI 的指示，請參閱[安裝 AWS SAM CLI](#)。如需如何升級 AWS SAM CLI 的指示，請參閱[升級 AWS SAM CLI](#)。

使用 AWS Toolkit 建立無伺服器應用程式

本範例示範如何透過 AWS Toolkit，使用 [AWS Serverless Application Model \(AWS SAM\)](#) 建立無伺服器應用程式。

1. 在 AWS Explorer 中，開啟 (按一下右鍵) Lambda 節點的內容選單，然後選擇 Create Lambda SAM Application (建立 Lambda SAM 應用程式)。

Note

或者，您也可以在此標題中選取選單圖示，然後選擇 Create Lambda SAM Application (建立 Lambda SAM 應用程式)。

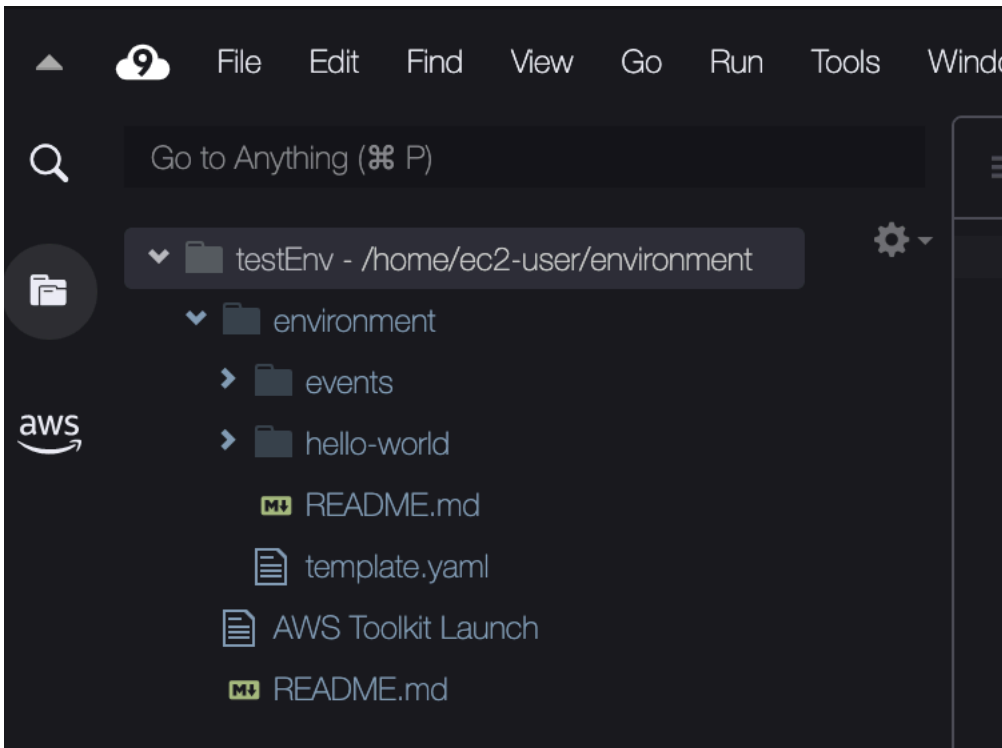
- 選擇您 SAM 應用程式的執行時間。在此範例中，請選擇 nodejs12.x。

Note

如果您選取了包含「(Image)」的執行時間，您的應用程式就是 Image 套件類型。如果您選取了不包含「(Image)」的執行時間，您的應用程式就是 Zip 類型。如需詳細了解 Image 和 Zip 套件類型的差異，請參閱 AWS Lambda 開發人員指南中的 [Lambda 部署套件](#)。

- 為您的無伺服器應用程式選擇下列其中一個範本：
 - AWS SAM Hello World：包含 Lambda 函式的基本範本，會傳回常見的「Hello World」訊息。
 - AWS Step Functions 範例應用程式：用於執行股票交易工作流程的範例應用程式。Step Functions 會協調 Lambda 函數有參與的互動。
- 您可以選擇新專案的位置。若已存在，您可以選取現有的工作空間資料夾。否則，請瀏覽其他資料夾。如果選擇 Select a different folder (選取其他資料夾)，畫面會顯示一個對話方塊，您可以從中選取資料夾位置。
- 輸入新應用程式的名稱。在此範例中，使用 my-sam-app-nodejs。按下 Enter 後，AWS Toolkit 需要一些時間才能建立專案。

建立專案時，您可以在「Environment」(環境)視窗中檢視應用程式的檔案。列於 Explorer 視窗中。



執行與除錯無伺服器應用程式

您可以使用 AWS Toolkit 來設定如何對無伺服器應用程式進行除錯，並在本機開發環境中執行應用程式。您可以對 AWS Serverless Application Model (AWS SAM) 範本所定義的無伺服器應用程式進行除錯。此範本使用簡單的 YAML 語法來描述函數、API、資料庫和事件來源對應等組成無伺服器應用程式的資源。

若要詳細了解 AWS SAM 範本，請參閱 AWS Serverless Application Model 開發人員指南中的 [AWS SAM 範本剖析](#)。

或者，您可以對尚未經認可使用於 SAM 範本的無伺服器應用程式進行快速除錯。

如要開始設定除錯行為，請使用內嵌動作來識別符合資格的 AWS Lambda 函數。若要使用 SAM 範本所定義的基礎架構，請在使用相關 YAML 格式檔案中的內嵌動作。若要在不使用範本的情況下直接測試函數，請使用應用程式檔案中 Lambda 處理常式的內容感知連結。

Note

在此範例中，我們正在偵錯使用 JavaScript。但是，您可以透過以下語言和執行時間使用 AWS Toolkit 中的除錯功能：

- JavaScript — Node.js 10. 十、十二 十、十四 x

- Python — 3.7、3.8、3.9、3.10 (Python 2.7 和 3.6 無伺服器應用程式可以執行，但無法由工具組進行除錯。) AWS

您的語言選擇也會在內容感知連結指出符合資格的 Lambda 處理常式時產生影響。如需詳細資訊，請參閱[直接從程式碼執行和除錯無伺服器函數](#)。

使用 SAM 範本執行並除錯無伺服器應用程式

應用程式透過 SAM 範本來執行和除錯時，它的行為和所使用的資源會由 YAML 格式的檔案所描述。如果您使用 AWS Toolkit 建立無伺服器應用程式，系統會自動為您的專案產生名為 `template.yaml` 的檔案。

在此程序中，請使用建立於 [建立無伺服器應用程式](#) 的範例應用程式。

如何使用 SAM 範本執行並除錯無伺服器應用程式

1. 若要檢視無伺服器應用程式內含的應用程式檔案，請前往 Environment (環境) 視窗。
2. 從應用程式資料夾 (例如，`my-sample-app`) 開啟 `template.yaml` 檔案。
3. 針對 `template.yaml`，選取 Edit Launch Configuration (編輯啟動組態)。

新的編輯器會顯示 `launch.json` 檔案，該檔案提供的除錯組態採用預設屬性。

4. 編輯或確認下列組態屬性的值：
 - "name" – 輸入方便識讀的名稱，讓它顯示在 Run (執行) 檢視的 Configuration (組態) 下拉式欄位中。
 - "target" – 確保該值是 "template"。如此一來，SAM 範本是除錯工作階段的進入點。
 - "templatePath" – 輸入 `template.yaml` 檔案的相對或絕對路徑。
 - "logicalId" – 確認名稱與 SAM 範本 Resources (資源) 區段中指定的名稱相符。在此例中，此名為 `AWS::Serverless::Function` 類型的 `HelloWorldFunction`。

如需詳細了解 `launch.json` 檔案中的上述與其他項目，請參閱 [除錯無伺服器應用程式的組態選項](#)。

5. 如果您認為除錯組態沒有問題，請儲存 `launch.json`。接下來，選擇 RUN (執行) 旁邊的綠色「播放」按鈕，即可開始除錯。

Note

如果您的 SAM 應用程式無法執行，請前往 Output (輸出) 視窗查看錯誤是否因未建置 Docker 映像所致。您可能需要釋放環境中的磁碟空間。

如需詳細資訊，請參閱[在本機 AWS Toolkit 執行 SAM 應用程式時發生錯誤，因為 AWS Cloud9 環境沒有足夠的磁碟空間](#)。

當除錯工作階段啟動時，DEBUG CONSOLE (除錯主控台) 面板會顯示除錯輸出，並顯示 Lambda 函數傳回的任何值。除錯 SAM 應用程式時，Output (輸出) 面板會將 AWS Toolkit 選為輸出管道。

Note

如果 Windows 使用者在此程序中遇到 Docker 掛載問題，您可能需要重新整理自己的共用磁碟機的憑證 (在 Docker 設定中)。Docker 掛載錯誤類似如下。

```
Fetching lambci/lambdajava: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 ...
```

直接從程式碼執行和除錯無伺服器函數

測試 AWS SAM 應用程式時，您可以選擇只執行與除錯 Lambda 函數。請排除 SAM 範本所定義的其他資源。這種方法必須使用內嵌動作，在可以直接叫用的來源程式碼中識別 Lambda 函數處理常式。

內容感知連結偵測到的 Lambda 處理常式取決於您用於應用程式的語言和執行時間。

語言/執行時間	內容感知連結識別 Lambda 函數的條件
JavaScript (Node.js 十、十二及十四倍)	函式具備以下特性： <ul style="list-style-type: none"> 它是經過匯出的函式，最多有三個參數。

語言/執行時間	內容感知連結識別 Lambda 函數的條件
Python (三、三、三、三)	<p>函式具備以下特性：</p> <ul style="list-style-type: none"> • 它在工作空間資料夾中的父項資料夾有一個 <code>package.json</code> 檔案。 • 它是最上層函數。 • 它在工作空間資料夾中的父項資料夾有一個 <code>requirements.txt</code> 檔案。

如何從應用程式程式碼直接執行與除錯無伺服器應用程式

1. 若要檢視無伺服器應用程式檔案，請選擇編輯器旁邊的資料夾圖示，導覽至應用程式資料夾。
2. 從應用程式資料夾 (例如 `my-sample-app`)，展開函數資料夾 (在此範例中為 `hello-world`)，然後 `app.js` 開啟檔案。
3. 在用於識別符合資格的 Lambda 處理常式函數的內嵌動作中，選擇 `Add Debug Configuration`。如未顯示新增除錯組態選項，您必須啟用程式碼濾鏡。若要啟用程式碼濾鏡，請參閱 [the section called “啟用 AWS Toolkit 程式碼濾鏡”](#)。
4. 選取執行 SAM 應用程式的執行時間。
5. 在 `launch.json` 檔案的編輯器中編輯或確認下列組態屬性的值：
 - `"name"` – 輸入方便識讀的名稱。
 - `"target"` – 確認值為 `"code"`，以便直接叫用 Lambda 函式處理常式。
 - `"lambdaHandler"` – 輸入程式碼中所用方法的名稱，Lambda 會呼叫此程式碼來叫用您的函數。例如，對於中的應用程式 JavaScript，預設值為 `app.lambdaHandler`。
 - `"projectRoot"` – 為包含 Lambda 函式的應用程式輸入檔案路徑。
 - `"runtime"` – 輸入或確認 Lambda 執行環境的有效執行時間 (例如 `"nodejs.12x"`)。
 - `"payload"` – 選擇下列其中一個選項來定義您想要提供給 Lambda 函式作為輸入的事件酬載：
 - `"json"`：定義事件酬載的 JSON 格式金鑰數值組。
 - `"path"`：作為事件酬載的檔案的路徑。
6. 如果您認為除錯組態沒有問題，請選擇 `RUN (執行)` 旁的綠色播放箭頭開始除錯。

當除錯工作階段啟動時，DEBUG CONSOLE (除錯主控台) 面板會顯示除錯輸出，並顯示 Lambda 函數傳回的任何值。除錯 SAM 應用程式時，Output (輸出) 面板會將 AWS Toolkit 選為輸出管道。

Note

如果錯誤訊息提及 Docker，請參閱本[注意事項](#)。

執行並除錯本機 Amazon API Gateway 資源

您可以執行或除錯 `template.yaml` 中指定的 AWS SAM API Gateway 本機資源。請使用 `invokeTarget.target=api` 執行 `type=aws-sam` 的 AWS Cloud9 啟動組態，以執行此操作。

Note

API Gateway 支援兩種類型的 API。分別為 REST 和 HTTP API。不過，AWS Toolkit 的 API Gateway 功能僅支援 REST API。有時 HTTP API 可稱為「API Gateway V2 API」。

如何執行並除錯本機 API Gateway 資源

1. 選擇以下其中一種方法來建立 AWS SAM API Gateway 資源的啟動組態：

- 選項 1：前往您 AWS SAM 的處理常式來源程式碼 (尤其是 `.js`、`.cs` 或 `.py` 檔案) 專案，將游標懸浮在 Lambda 處理常式上，然後選擇 Add Debug Configuration (新增除錯組態)。如果未顯示新增除錯組態選項，請啟用程式碼濾鏡。若要啟用程式碼濾鏡，請參閱 [the section called “啟用 AWS Toolkit 程式碼濾鏡”](#)。然後，在選單中選擇標示為 API Event (API 事件) 的項目。
- 選項 2：編輯 `launch.json`，並使用以下語法建立新的啟動組態。

```
{
  "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 (執行) 按鈕旁邊的下拉式選單中，選擇啟動組態 (在上面的範例中名稱為 myConfig)。
3. (選用) 在 Lambda 專案程式碼中新增中斷點。
4. 選擇綠色「播放」按鈕旁邊的 Run (執行) 按鈕。
5. 在輸出窗格中檢視結果。

組態

如果您使用 `invokeTarget.target` 屬性值 `api`，Toolkit 會變更啟動組態驗證和行為，以便支援 `api` 欄位。

```
{
  "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": {}
}
```

將範例中的值更換為如下所示：

`invokeTarget.logicalId`

API 資源。

路徑

啟動組態請求的 API 路徑 (例如 `"path": "/hello"`)。

必須為經由 `invokeTarget.templatePath` 指定，從 `template.yaml` 解析的有效 API 路徑。

`httpMethod`

使用以下其中一個動詞：「delete」、「get」、「head」、「options」、「patch」、「post」、「put」。

`payload`

要在請求中傳送的 JSON 酬載 (HTTP 主體)，結構和規則與 `lambda.payload` 欄位相同。

`payload.path` 指向包含 JSON 酬載的檔案。

`payload.json` 指定 JSON 酬載內嵌。

標頭

選用的名稱/值組對應。使用此選項可以指定要包含在請求中的 HTTP 標頭。

```
"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

(選用) 使用此字串來設定請求中的 `querystring` (例如 `"querystring": "abc=def&ghi=jkl"`)。

aws

AWS 連線資訊的提供方式。如需詳細資訊，請參閱 [除錯無伺服器應用程式的組態選項](#) 中的 AWS 連線 (`aws`) 屬性 資料表。

sam

AWS SAM CLI 建置應用程式的方式。如需詳細資訊，請參閱 [除錯無伺服器應用程式的組態選項](#) 中的 AWS SAM CLI (「`sam`」) 屬性。

同步無伺服器應用程式

此範例顯示如何將上一個主題 ([建立無伺服器應用程式](#)) 中建立的無伺服器應用程式與AWS使用AWS Toolkit for Visual Studio Code。

必要條件

- 請務必選擇全域唯一的 Amazon S3 儲存貯體名稱。
- 確認您在當中設定的憑證包括下列服務的合適讀取/寫入存取權：Amazon S3、AWS CloudFormation、AWS Lambda 和 Amazon API Gateway。
- 如果是部署類型為 Image 的應用程式，請確保您可在部署中使用全域唯一的 Amazon S3 儲存貯體名稱和 Amazon ECR 儲存庫 URI。

同步無伺服器應用程式

1. 在AWS檔案總管視窗中，開啟 Lambda 節點的內容 (按一下滑鼠右鍵) 功能表，然後選取同步 SAM 應用程式。
2. 選擇AWS 區域要部署到的目的地。
3. 選擇要用於部署的 `template.yaml` 檔案。
4. 輸入此部署可以使用的 Amazon S3 儲存貯體名稱。儲存貯體必須位於您要部署檔案的 區域。

⚠ Warning

在現有的所有 Amazon S3 儲存貯體名稱中，該 Amazon S3 儲存貯體名稱必須為全域唯一。將唯一識別符新增至在以下範例中指定的名稱，或選擇不同的名稱。

5. 如果您的無伺服器應用程式含有套件類型為 Image 的函式，請輸入此部署可以使用的 Amazon ECR 儲存庫名稱。儲存庫必須位於您要部署檔案的區域。
6. 輸入部署之堆疊的名稱，即新堆疊的名稱或現有堆疊的名稱。
7. 在 Console (主控台) 的 AWS Toolkit 索引標籤上確認部署成功。

如果發生錯誤，右下角會顯示彈出式訊息。

如果發生這種情況，請查看 AWS Toolkit 索引標籤中的文字了解詳情。以下是錯誤詳細資訊的範例。

```
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.
```

在此範例中，由於 Amazon S3 儲存貯體不存在，因此發生錯誤。

部署完成時，您將看到應用程式在 AWS Explorer 中列出。若要了解如何叫用 Lambda 函式 (該函數是做為應用程式的一部分而建立)，請參閱 [叫用遠端 Lambda 函式](#)。

啟用 AWS Toolkit 程式碼濾鏡

1. 從選單列選擇 AWS Cloud9、然後選擇 Preferences (偏好設定)。
2. 在 Preferences (偏好設定) 標籤的側邊欄中，選擇 AWS Toolkit。
3. 若要啟用程式碼濾鏡，請選擇 Enable Code Lenses (啟用程式碼濾鏡)。

從 AWS 雲端刪除無伺服器應用程式

刪除無伺服器應用程式時，您會刪除之前部署至 AWS 雲端的 AWS CloudFormation 堆疊。請注意，此程序不會從本機主機刪除應用程式目錄。

1. 開啟 AWS Explorer。
2. 在 AWS Explorer 視窗中，展開您要刪除的已部署應用程式所在的區域，然後展開 AWS CloudFormation。
3. 找到與您要刪除的無伺服器應用程式對應的 AWS CloudFormation 堆疊，以滑鼠右鍵按一下後開啟內容選單。然後，選擇「刪除 CloudFormation 堆疊」。
4. 如要確認刪除所選堆疊，請選擇 Delete (刪除)。

如果堆疊刪除成功，AWS Toolkit 會從 AWS Explorer 的 AWS CloudFormation 清單中移除堆疊名稱。

除錯無伺服器應用程式的組態選項

透過內嵌動作，您可以輕鬆地以直接方式或使用 SAM 範本來尋找並定義叫用 Lambda 函數的屬性。您也可以定義以下屬性："lambda" (函數的執行方式)、"sam" (AWS SAM CLI 建置應用程式的方式)，以及"aws" (AWS 連線資訊的提供方式)。

AWS SAM：直接的 Lambda 處理常式叫用/以範本為基礎的 Lambda 叫用

屬性	描述
type	指定用於管理啟動組態的擴充功能。一律設為 <code>aws-sam</code> ，以便使用 AWS SAM CLI 進行本機建置與除錯。
name	指定方便識讀的名稱，讓它顯示在 Debug launch configuration (除錯啟動組態清單) 中。
request	指定要經由指定擴充功能執行的組態類型 (<code>aws-sam</code>)。一律設為 <code>direct-invoke</code> 來啟動 Lambda 函數。
invokeTarget	指定叫用資源的進入點。 如要直接叫用 Lambda 函數，請設定以下 <code>invokeTarget</code> 欄位的值： <ul style="list-style-type: none"> • <code>target</code> – 設為 <code>code</code>。

屬性	描述
	<ul style="list-style-type: none"> • <code>lambdaHandler</code> – 要叫用的 Lambda 函數處理常式名稱。 • <code>projectRoot</code> – 包含 Lambda 處理常式的應用程式檔案路徑。 <p>如要使用 SAM 範本叫用 Lambda 資源，請設定以下 <code>invokeTarget</code> 欄位的值：</p> <ul style="list-style-type: none"> • <code>target</code> – 設為 <code>template</code>。 • <code>templatePath</code> – SAM 範本檔案的路徑。 • <code>logicalId</code> – 要叫用的 <code>AWS::Lambda::Function</code> 或 <code>AWS::Serverless::Function</code> 資源名稱。您可以在 YAML 格式的 SAM 範本中找到資源名稱。

Lambda ("lambda") 屬性

屬性	描述
<code>environmentVariables</code>	傳送操作參數至函數中。例如，如果您正在寫入到 Amazon S3 儲存貯體，請將儲存貯體名稱設定為環境變數。請勿將您要寫入的儲存貯體名稱寫死在程式碼中。
<code>payload</code>	針對您提供給 Lambda 函數作為輸入的事件酬載提供兩個選項。 <ul style="list-style-type: none"> • <code>"json"</code>：定義事件酬載的 JSON 格式金鑰數值組。 • <code>"path"</code>：作為事件酬載的檔案的路徑。
<code>memoryMB</code>	指定執行一個叫用的 Lambda 函數時要提供的記憶體 MB 值。
<code>runtime</code>	指定 Lambda 函數使用的執行時間。如需詳細資訊，請參閱 AWS Lambda 執行時間 。
<code>timeoutSec</code>	設定除錯工作階段逾時之前的可用時間 (以秒為單位)。

AWS 工具組擴充功能會使用 AWS SAM CLI 在本機建置並除錯無伺服器應用程式。您可以使用 `launch.json` 檔案中的 "sam" 組態屬性設定 AWS SAM CLI 命令的行為。

AWS SAM CLI ("sam") 屬性

屬性	描述	預設值
buildArguments	設定 sam build 命令建置 Lambda 來源程式碼的方式。若要檢視建置選項，請參閱 AWS Serverless Application Model 開發人員指南中的 SAM 建置 。	空字串
containerBuild	指出是否在 AWS Lambda 式的 Docker 容器中建置函數。	false
dockerNetwork	指定 Lambda Docker 容器應連線的現有 Docker 網路名稱或 ID，以及預設橋接網路。如果沒有指定，則 Lambda 容器只會連線到預設的橋接 Docker 網路。	空字串
localArguments	其他本機叫用引數。	空字串
skipNewImageCheck	指定命令是否應略過將 Lambda 執行階段的最新 Docker 映像提取下來的動作。	false
template	使用參數輸入客戶值，藉此自訂 SAM 範本。如需更多詳細資訊，請參閱 https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html 使用者指南中的 AWS CloudFormation 參數部分。	"parameters": {}

AWS 連線 ("aws") 屬性

屬性	描述	預設值
credentials	從憑證檔案選取特定設定檔 (例如 profile:default)，藉此取得 AWS 憑證。	現有共用 AWS 組態檔案或共用 AWS 憑證檔案提供的 AWS 憑證。
Region	設定服務的 AWS 區域 (例如 us-east-1)。	與作用中的憑證設定檔相關聯的預設 AWS 區域。

透過 AWS 工具組使用 AWS Step Functions

AWS 工具組支援 [AWS Step Functions](#)。Step Functions 允許您建立狀態機器來定義 AWS Lambda 函式和其他 AWS 服務的工作流程，這些服務支援商業關鍵應用程式。

您可以使用 AWS 工具組，藉助 Step Functions 執行以下操作：

- 建立和發佈狀態機器，該狀態機器是由單個步驟組成的工作流程。
- 下載定義狀態機器工作流程的檔案。
- 使用已輸入或選擇的輸入運行狀態機器工作流程。

主題

- [先決條件](#)
- [建立和發佈狀態機器](#)
- [在 AWS 工具組中運行狀態機器](#)
- [下載狀態機器定義檔案並視覺化其工作流程](#)

先決條件

Step Functions 可以運程式碼並存取 AWS 資源 (例如叫用 Lambda 函式)。為了維護安全性，您必須使用 IAM 角色將這些資源的存取授予 Step Functions。

藉助 AWS 工具組，您可利用自動產生的 IAM 角色，該角色可用於您建立狀態機器的 AWS 區域。若要為狀態機器建立自己的 IAM 角色，請參閱 AWS Step Functions 開發人員指南中的 [AWS Step Functions 如何與 IAM 一起運作](#)。

建立和發佈狀態機器

使用 AWS 工具組建立狀態機器時，您可以選擇定義商業案例工作流程的入門範本。接著，您可以編輯或取代該範本以滿足特定需求。有關在表示其結構的檔案中定義狀態機器的詳細資訊，請參閱 AWS Step Functions 開發人員指南中的 [Amazon States Language](#)。

1. 在 AWS Explorer 窗格中，開啟 Step Functions 的內容 (按一下滑鼠右鍵) 選單，然後選擇 Create a new Step Function state machine (建立新的 Step Functions 狀態機器)。
2. 在命令面板中，為狀態機器的工作流程選擇入門範本。
3. 接著，選擇用於定義狀態機器的 Amazon States Language (ASL) 檔案的格式。

此時將打開一個編輯器，其中顯示定義狀態機器工作流程的 ASL 檔案。

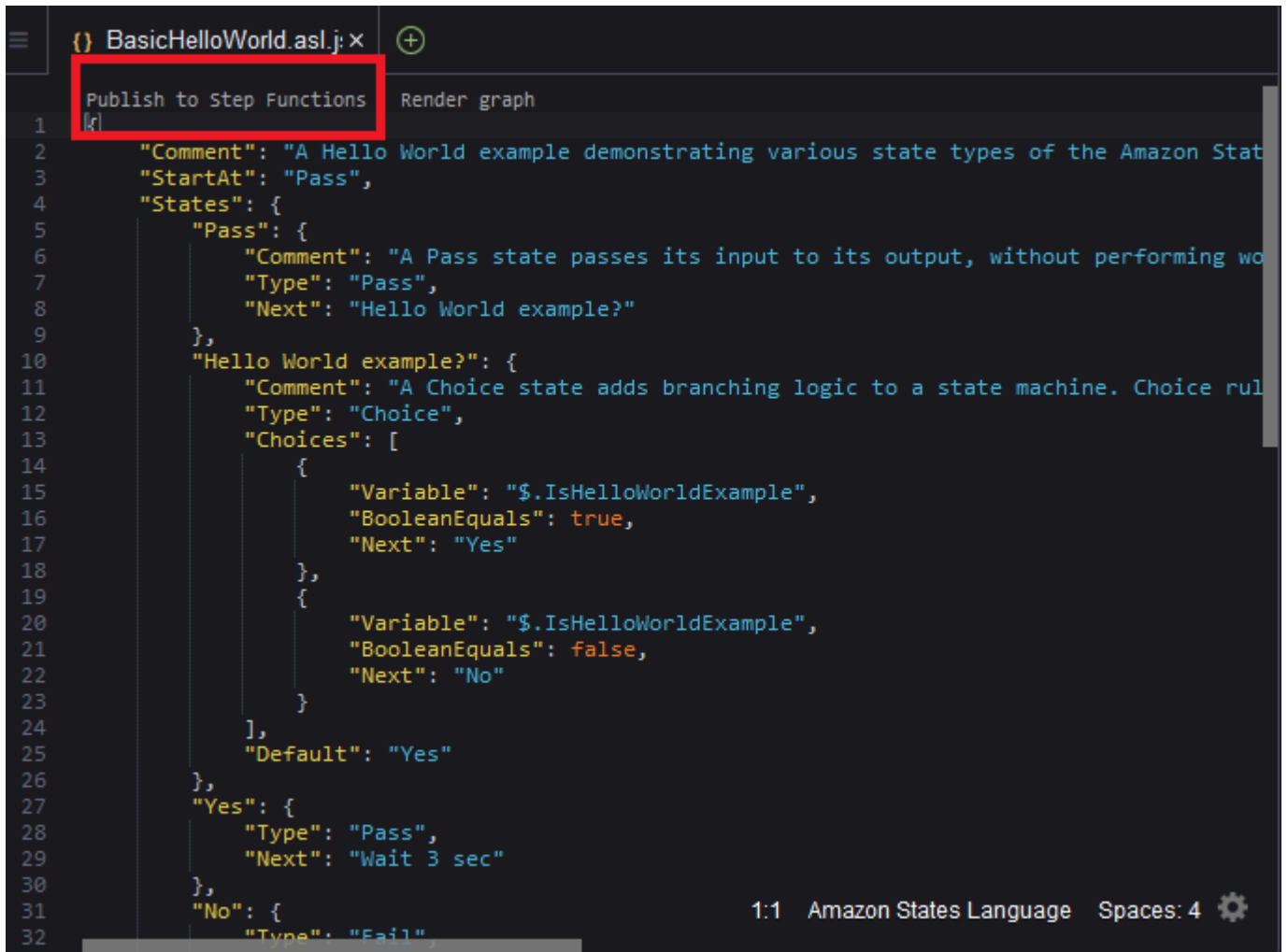
Note

有關編輯 ASL 檔案以自訂工作流程的資訊，請參閱[狀態機器結構](#)。

4. 在 ASL 檔案中，選擇 Publish to Step Functions (發佈至 Step Functions) 以將狀態機器新增至 AWS Cloud。

Note

您也可以 ASL 檔案中選擇 Render graph (轉譯圖形)，以顯示狀態機器工作流程的視覺化呈現。



```
{
  "Comment": "A Hello World example demonstrating various state types of the Amazon States Language",
  "StartAt": "Pass",
  "States": {
    "Pass": {
      "Comment": "A Pass state passes its input to its output, without performing any actions",
      "Type": "Pass",
      "Next": "Hello World example?"
    },
    "Hello World example?": {
      "Comment": "A Choice state adds branching logic to a state machine. Choice rules evaluate a list of conditions and execute the state associated with the first condition that evaluates to true.",
      "Type": "Choice",
      "Choices": [
        {
          "Variable": "$.IsHelloWorldExample",
          "BooleanEquals": true,
          "Next": "Yes"
        },
        {
          "Variable": "$.IsHelloWorldExample",
          "BooleanEquals": false,
          "Next": "No"
        }
      ],
      "Default": "Yes"
    },
    "Yes": {
      "Type": "Pass",
      "Next": "Wait 3 sec"
    },
    "No": {
      "Type": "Fail"
    }
  }
}
```

5. 在命令面板中，選擇用於託管 Step Functions 的 AWS 區域。
6. 接著，您可以選擇建立新的 Step Functions 或更新現有的 Step Functions。

Quick Create

此選項可讓您使用 [step-functions/latest/dg/concepts-standard-vs-express.html](https://docs.aws.amazon.com/step-functions/latest/dg/concepts-standard-vs-express.html)，從 ASL 檔案建立新的 Step Functions。系統會要求您指定以下內容：

- 一個 IAM 角色，可讓您的 Step Functions 運程式碼並存取 AWS 資源。(您可以選擇自動產生的 IAM 角色，該角色對在其中建立狀態機器的 AWS 區域有效。)
- 為新的函式輸入一個名稱。

您可以在 AWS 工具組輸出索引標籤中檢查狀態機器是否已成功建立，以及取得其 ARN。

Quick Update

如果狀態機器已存在於 AWS 區域中，則可以選擇一個狀態機器以使用目前的 ASL 檔案進行更新。

您可以在 AWS 工具組輸出索引標籤中檢查狀態機器是否已成功更新，以及取得其 ARN。

建立狀態機器後，它會顯示在 AWS Explorer 窗格的 Step Functions 之下。如果此狀態機器沒有立即出現，請選擇 Toolkit (工具組) 選單中的 Refresh Explorer (重新整理 Explorer)。

在 AWS 工具組中運行狀態機器

您可以使用 AWS 工具組運行遠端狀態機器。運行中的狀態機器會收到 JSON 文字做為輸入，並將該輸入傳遞給工作流程中的第一個狀態。個別狀態會收到 JSON 做為輸入，而且通常會將 JSON 當作輸出傳遞至下一個狀態。如需詳細資訊，請參閱 [Step Functions 中的輸入和輸出處理](#)。

1. 在 AWS Explorer 窗格中，選擇 Step Functions。接著開啟特定狀態機器的內容 (按一下滑鼠右鍵) 選單，然後選擇 Start Execution (開始執行)。
2. 在 Start Execution (開始執行) 窗格中，透過直接在下面的欄位中輸入文字或從本機裝置上傳檔案來新增狀態機器工作流程的 JSON 格式輸入。
3. 選擇 Execute (執行)。

AWS 工具組輸出索引標籤顯示工作流程已啟動的確認資訊和程序 ID 的 ARN。您可以使用該程序 ID，在 AWS Step Functions 主控台中檢查工作流程是否成功運行。您還可以檢視工作流程啟動和結束時間的時間戳記。

下載狀態機器定義檔案並視覺化其工作流程

下載狀態機器意即下載一個包含表示該狀態機器結構的 JSON 文字的檔案。接著，您可以編輯此檔案以建立新的狀態機器或更新現有的狀態機器。如需詳細資訊，請參閱 AWS Step Functions 開發人員指南中的 [Amazon States Language](#)。

1. 在 AWS Explorer 窗格中，選擇 Step Functions。接著開啟特定狀態機器的內容 (按一下滑鼠右鍵) 選單，然後選擇 Download Definition (下載定義)。

Note

該內容選單還提供了 Copy Name (複製名稱) 和 Copy ARN (複製 ARN) 的選項。

2. 在 Save (儲存) 對話方塊中，在您的環境中選擇存放狀態機器下載檔案的資料夾，然後選擇 Save (儲存)。

定義狀態機器工作流程的 JSON 格式檔案顯示在編輯器中。

3. 若要顯示工作流程的視覺化呈現，請選擇 Render graph (轉譯圖形)。

視窗將顯示一個流程圖，其中顯示狀態機器工作流程中的狀態順序。

使用 Systems Manager 自動化文件

AWS Systems Manager 可讓您檢視及控制 AWS 上的基礎設施。Systems Manager 提供統一的使用者界面，以便您檢視來自多項 AWS 服務的操作資料，以及自動執行跨 AWS 資源的操作任務。

[Systems Manager 文件](#) 定義 Systems Manager 在受管執行個體上執行的動作。自動化文件是一種 Systems Manager 文件類型，使用其執行常見的維護和部署任務。包括建立或更新 Amazon Machine Image (AMI)。本主題概述如何使用 AWS 工具組建立、編輯、發佈和刪除自動化文件。

主題

- [假設和先決條件](#)
- [Systems Manager 自動化文件的 IAM 許可](#)
- [建立新的 Systems Manager 自動化文件](#)
- [發佈 Systems Manager 自動化文件](#)
- [編輯現有的 Systems Manager 自動化文件](#)
- [處理版本](#)
- [刪除 Systems Manager 自動化文件](#)
- [運行 Systems Manager 自動化文件](#)
- [疑難排解 AWS 工具組中的 Systems Manager 自動化文件](#)

假設和先決條件

在開始之前，請確保您符合以下條件：

- 您熟悉 Systems Manager。如需詳細資訊，請參閱《[AWS Systems Manager 使用者指南](#)》。
- 您熟悉 Systems Manager 自動化使用案例。如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 [AWS Systems Manager 自動化](#)。

Systems Manager 自動化文件的 IAM 許可

您必須具有一個憑證描述檔，其中包含所需的 AWS Identity and Access Management (IAM) 許可，以建立、編輯、發佈和刪除 Systems Manager 自動化文件。以下政策文件定義了可在主體政策中使用的必要 IAM 許可。

```
{
  "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": "*"
    }
  ]
}
```

如需如何更新 IAM 政策的詳細資訊，請參閱《IAM 使用者指南》中的 [建立 IAM 政策](#)。

建立新的 Systems Manager 自動化文件

您可以使用 AWS 工具組，在 JSON 或 YAML 中建立自動化文件。建立自動化文件之後，它會顯示在未命名檔案中。您可以命名並儲存您的檔案。但您要發佈該檔案，才會上傳至 AWS。

要新建自動化文件

1. 在左側導覽窗格中選擇搜尋圖示，或按下 Ctrl+P 開啟 Search (搜尋) 窗格。
2. 在 Search (搜尋) 窗格中，開始輸入「systems manager」一詞，然後選擇顯示的 AWS: Create a new Systems Manager Document Locally (AWS：在本機建立新的 Systems Manager 文件) 命令。
3. 為 Hello World 範例選擇其中一個入門範本。
4. 選擇 JSON 或 YAML 作為文件的格式。

編輯器將顯示新的自動化文件。

Note

首次建立本機自動化文件之後，它不會自動顯示在 AWS 中。您必須將其發佈到 AWS，才能執行它。

發佈 Systems Manager 自動化文件

在 AWS 工具組中建立或編輯自動化文件後，您即可將其發佈至 AWS。

發佈自動化文件

1. 使用 [編輯現有的 Systems Manager 自動化文件](#) 中概述的程序，開啟要發佈的自動化文件。
2. 在左側導覽窗格中選擇搜尋圖示，或按下 Ctrl+P 開啟 Search (搜尋) 窗格。
3. 在 Search (搜尋) 窗格中，開始輸入「systems manager」一詞，然後選擇顯示的 AWS: Publish a new Systems Manager Document (AWS：發佈新的 Systems Manager 文件) 命令。
4. 對於步驟 1 (共 3 步)，選擇您要在其中發佈文件的 AWS 區域。
5. 對於步驟 2 (共 3 步)，選擇 Quick Create (快速建立) 以建立自動化文件。或者，選擇 Quick Update (快速更新) 以更新區域中現有的自動化文件。

Note

只能更新自己擁有的自動化文件。如果選擇 Quick Update (快速更新)，而您在該區域中並未擁有任何文件，則系統會顯示一則訊息，通知您在更新之前先發佈文件。

- 對於步驟 3 (共 3 步)，根據您在上一步中的選擇，輸入新自動化文件的名稱或選擇要更新的現有文件。

Note

當您將更新發佈到 AWS 中現有的自動化文件時，則會將新版本新增至文件中。如果文件具有多個版本，則可以設定[預設的版本](#)。

編輯現有的 Systems Manager 自動化文件

使用 AWS Explorer 搜尋現有的 Systems Manager 自動化文件。開啟現有文件時，該文件將作為未命名檔案顯示在 AWS Cloud9 編輯器中。您可以下載 3 種類型的自動化文件：

- Owned by Amazon (Amazon 擁有)：可透過在執行時間指定參數來使用的預先設定 SSM 文件。
- Owned by me (我擁有)：我建立並發佈到 AWS 的文件。
- Shared with me (與我共享)：擁有者已與您共享的文件 (基於您的 AWS 帳戶 ID)。

您可以在 AWS 上更新的唯一文件類型是 owned by me (我擁有)。您也可以下載由 Amazon 共享或擁有的自動化文件，然後在 AWS Cloud9 中編輯它們。但是，當您發佈到 AWS 時，則必須使用建立新文件或更新您擁有的現有文件。您無法建立具有其他擁有者或由 Amazon 擁有的文件的新版本。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的 [AWS Systems Manager 文件](#)。

- 在 AWS Explorer 中，對於 Systems Manager，選擇要下載的 SSM 文件的類別：Owned by Amazon (Amazon 擁有)、Owned by me (我擁有) 或 Shared with me (與我共享)。
- 對於特定文件，開啟內容 (按一下滑鼠右鍵) 選單，然後選擇 Download as YAML (以 YAML 形式下載) 或 Download as JSON (以 JSON 形式下載)。

格式化的 SSM 文件將顯示在新的編輯器索引標籤中。

完成編輯後，您可以使用 AWS: Publish a new Systems Manager Document (AWS：發佈新的 Systems Manager 文件) 命令，在 AWS Cloud 中建立新的文件或更新您擁有的現有文件。

處理版本

Systems Manager 自動化文件使用版本進行變更管理。藉助 AWS 工具組，您可以設定文件的預設版本，這是運行文件時使用的版本。

設定預設版本

- 在 AWS Explorer 中，導覽至您要設定預設版本的文件，開啟文件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Set default version (設定預設版本)。

Note

如果選定文件只有一個版本，則將無法變更預設版本。

刪除 Systems Manager 自動化文件

您可以在 AWS 工具組中刪除擁有的自動化文件。刪除自動化文件會同時刪除該文件及其所有版本。

Important

- 刪除是一種破壞性動作，而且無法復原。
- 刪除已啟動的自動化文件不會刪除在運行時建立或修改的 AWS 資源。
- 僅當您擁有此文件時，才允許執行刪除。

刪除您的自動化文件

- 在 AWS Explorer 窗格中，對於 Systems Manager，展開 Owned by Me (我擁有) 以列出您的文件。
- 開啟您要刪除文件的內容 (按一下滑鼠右鍵) 選單，然後選擇 Delete document (刪除文件)。
- 在顯示的警告對話方塊中，選擇 Delete (刪除) 以確認。

運行 Systems Manager 自動化文件

將自動化文件發佈至 AWS 之後，您就可以運行它以代替您在 AWS 帳戶中執行任務。要運行自動化文件，請使用 AWS Management Console、Systems Manager API、AWS CLI 或 AWS Tools for PowerShell。如需如何運行自動化文件的詳細指示，請參閱《AWS Systems Manager 使用者指南》中的[執行簡易自動化](#)。

或者，如果您想搭配使用一個 AWS 開發套件與 Systems Manager API 執行自動化文件，請參閱 [AWS 開發套件參考](#)。

Important

執行自動化文件可能會在 AWS 中建立新的資源並可能產生帳單費用。我們強烈建議您在運行之前了解自動化文件將在您帳戶中建立的資源。

疑難排解 AWS 工具組中的 Systems Manager 自動化文件

我將自動化文件儲存在 AWS 工具組中，但在 AWS Management Console 中未看到此文件。

將自動化文件儲存在 AWS 工具組中不會將其發佈到 AWS。如需發佈自動化文件的詳細資訊，請參閱 [發佈 Systems Manager 自動化文件](#)。

發佈我的自動化文件失敗，出現許可錯誤。

請確認您的 AWS 憑證描述檔具有發佈自動化文件所需的許可。如需許可政策範例，請參閱 [Systems Manager 自動化文件的 IAM 許可](#)。

我將自己的自動化文件發佈到 AWS，但沒有在 AWS Explorer 窗格中看到它。

請確保已將文件發佈到在 AWS Explorer 窗格中瀏覽的同一 AWS 區域。

我已經刪除自己的自動化文件，但仍然需要為它建立的資源支付費用。

刪除自動化文件不會同時刪除其建立或修改的資源。您可以從 [AWS 計費管理主控台](#) 識別您所建立的 AWS 資源，瞭解您的費用，然後選擇要從中刪除的資源。

在 AWS Cloud9 IDE 中使用 Amazon ECR

Amazon Elastic Container Registry (Amazon ECR) 是 AWS 受管的容器登錄服務，具安全性和可擴展性。您可以從 AWS 工具組 Explorer 存取數個 Amazon ECR 服務功能：

- 建立儲存庫。
- 為儲存庫或標記影像建立 AWS App Runner 服務。
- 存取影像標籤和儲存庫 URI 或 ARN。
- 刪除映像檔標籤和儲存庫。

您也可以安裝 AWS CLI 和其他平台，透過 AWS Cloud9 主控台存取完整的 Amazon ECR 功能。

如需有關 Amazon ECR 的詳細資訊，請參閱 Amazon Elastic Container Registry 使用者指南中的[什麼是 Amazon ECR？](#)。

先決條件

下列項目已預先安裝在適用於 AWS Cloud9 Amazon EC2 環境的 AWS Cloud9 IDE 中。他們需要從 AWS Cloud9 IDE 存取 Amazon ECR 服務。

IAM 登入資料

您在主控台中建立並用於身分驗證的 AWS IAM 角色。如需 IAM 的詳細資訊，請參閱《[AWS Identity and Access Management 使用者指南](#)》。

Docker 組態

Docker 已預先安裝在適用於 AWS Cloud9 Amazon EC2 環境的 AWS Cloud9 IDE 中。如需詳細資訊，請參閱[安裝 Docker 引擎](#)。

AWS CLI 第 2 版組態

AWS CLI 第 2 版已預先安裝在適用於 AWS Cloud9 Amazon EC2 環境的 AWS Cloud9 IDE 中。如需有關 AWS CLI 第 2 版的資訊，請參閱[安裝、更新和解除安裝 AWS CLI 第 2 版](#)。

主題

- [使用 Amazon 彈性容器註冊表服務 AWS Cloud9](#)

使用 Amazon 彈性容器註冊表服務 AWS Cloud9

您可以直接從 AWS Cloud9 IDE 中的 AWS 資源管理器訪問 Amazon Elastic Container Registry (Amazon ECR) 服務。您可以使用 Amazon ECR 將程式映像檔推送至 Amazon ECR 儲存庫。若要開始使用，請依照下列步驟執行：

1. 建立一個 Dockerfile，其中包含建置映像檔所需的資訊。
2. 從該 Dockerfile 建置映像檔並標記映像檔以進行處理。
3. 建立位於 Amazon ECR 執行個體內的儲存庫。
4. 將映像檔推送到您的儲存庫。

章節

- [必要條件](#)

- [1. 建立 Dockerfile。](#)
- [2. 從 Dockerfile 建置映像檔](#)
- [3. 建立新的儲存庫](#)
- [4. 推送、提取和刪除映像檔](#)

必要條件

在您可以使用 AWS 工具組的 Amazon ECR 功能之前 AWS Cloud9，請先確定您符合這些[先決條件](#)。這些先決條件已預先安裝在適用於 AWS Cloud9 Amazon EC2 環境的 AWS Cloud9 IDE 中，且必須存取 Amazon ECR。

1. 建立 Dockerfile。

Docker 使用稱為 Dockerfile 的檔案來定義可以推送和儲存於遠端儲存庫上的映像檔。在您可以將映像檔上傳到 ECR 儲存庫之前，請先建立一個 Dockerfile，然後從該 Dockerfile 建置映像檔。

建立 Dockerfile。

1. 若要導覽到要儲存 Dockerfile 的目錄，請在 AWS Cloud9 IDE 中的左側導覽列中選擇 Toggle Tree (切換樹狀檢視) 選項。
2. 建立名為 Dockerfile 的新檔案。

Note

AWS Cloud9 IDE 可能會提示您選取檔案類型或檔案副檔名。如果發生這種情況，請選取純文字。AWS Cloud9 IDE 有一個「碼頭文件」擴展名。不過，我們建議您不要使用它。這是因為副檔名可能會導致與特定版本的 Docker 或其他相關應用程式發生衝突。

使用 IDE 編輯碼頭文件 AWS Cloud9

如果 Dockerfile 有副檔名，請開啟該檔案的內容 (按一下滑鼠右鍵) 選單，然後移除副檔名。具有副檔名的 Dockerfile 可能會導致與 Docker 或其他相關應用程式的某些版本發生衝突。

從 Dockerfile 中刪除副檔名後：

1. 直接在 AWS Cloud9 IDE 中打開空的碼頭文件。
2. 將以下範例的內容複製到您的 Dockerfile 中。

Example Dockerfile 映像檔範本

```
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
```

這是一個使用 Ubuntu 22.04 圖像的碼頭文件。RUN 指令會更新套件快取。安裝 Web 伺服器的軟體套件服務，然後寫入「Hello World!」內容至 Web 伺服器的文件根目錄。EXPOSE 指令會公開容器上的連接埠 80，而 CMD 指令會啟動 Web 伺服器。

3. 儲存 Dockerfile。

2. 從 Dockerfile 建置映像檔

您建立的 Dockerfile 包含建置映像檔的必要資訊。將該映像檔推送到 Amazon ECR 執行個體之前，請先建立映像檔。

從 Dockerfile 建置映像檔

1. 若要瀏覽至包含 Dockerfile 的目錄，請使用 Docker CLI 或與 Docker 執行個體整合的 CLI。
2. 若要建置 Dockerfile 中定義的映像檔，請從與 Dockerfile 相同的目錄中運行 Docker build 命令。

```
docker build -t hello-world .
```

- 若要驗證映像檔已正確建立，請執行 Docker images 命令。

```
docker images --filter reference=hello-world
```

Example

輸出如下。

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 241MB	latest	e9ffedc8c286	4 minutes ago

- 要運行基於 Ubuntu 22.04 的新構建的映像，請使用迴聲命令。

Note

建立或推送映像檔不需要此步驟。但是，您可以看到程式映像檔在運行時如何工作。

```
FROM ubuntu:22.04  
CMD ["echo", "Hello from Docker in Cloud9"]
```

然後，運行並建置 Dockerfile。您必須從 Dockerfile 所在的目錄執行此命令。

```
docker build -t hello-world .  
docker run --rm hello-world
```

Example

輸出如下。

```
Hello from Docker in Cloud9
```

如需 Docker run 命令的詳細資訊，請參閱 Docker 網站上的 [Docker run 參考資料](#)。

3. 建立新的儲存庫

若要將映像檔上傳到 Amazon ECR 執行個體，請建立可存放它的新儲存庫。

建立新的 Amazon ECR 儲存庫

1. 從 AWS Cloud9 IDE 導覽列中，選擇「AWS 工具組」圖示。
2. 展開 AWS Explorer 選單。
3. 找出與您關聯的預設值 AWS 區域 AWS 帳戶。然後，選擇它以查看通過 AWS Cloud9 IDE 的服務列表。
4. 開啟 ECR 選項的內容選單 (按一下滑鼠右鍵)，以啟動建立新儲存庫程序。然後，選取 Create Repository (建立儲存庫)。
5. 要完成該過程，請按照提示進行操作。
6. 程序完成後，您可以從「檔案總管」(AWS Explorer) 功能表的 ECR 區段存取新的存放庫。

4. 推送、提取和刪除映像檔

從 Dockerfile 建立映像檔並建立儲存庫之後，您可以將映像檔推送到 Amazon ECR 儲存庫。此外，將 AWS 資源管理器與 Docker 和 AWS CLI 一起使用，您可以執行以下操作：

- 從儲存庫中提取映像檔。
- 刪除儲存庫中儲存的映像檔。
- 刪除您的儲存庫。

使用您的預設登錄檔驗證 Docker

在 Amazon ECR 和 Docker 執行個體之間交換資料時，需要進行身分驗證。若要向您的登錄檔驗證 Docker：

1. 在 AWS Cloud9 IDE 中開啟終端機。

2. 使用此方get-login-password法向您的私人 ECR 登錄進行驗證，並輸入您的地區和 AWS 帳戶 ID。

```
aws ecr get-login-password \  
  --region <region> \  
| docker login \  
  --username AWS \  
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Important

在上述命令中，將 **region** 和 **AWS_account_id** 取代為您 AWS 帳戶的特定資訊。有效的 **region** 值是 us-east-1。

標記並推送映像檔到您的儲存庫

使用的執行個體驗證 Docker 之後 AWS，請將映像檔推送至您的存放庫。

1. 使用 docker image 指令檢視儲存在本機的影像，並識別您要標記的影像。

```
docker images
```

Example

輸出如下。

REPOSITORY	TAG	IMAGE ID	CREATED
hello-world	latest	e9ffedc8c286	4 minutes ago
SIZE			
241MB			

2. 使用 Docker tag 命令標記 Docker 映像檔。

```
docker tag hello-world:latest AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

3. 使用 Docker push 命令將標記的映像檔推送到儲存庫。

⚠ Important

請確定本機儲存庫的名稱與 AWS Amazon EC2 儲存庫的名稱相同。在此範例中，兩個儲存庫都必須稱為 hello-world。有關使用 Docker 推送映像檔的詳細資訊，請參閱[推送 Docker 映像檔](#)。

```
docker push AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example

輸出如下。

```
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
```

將標記的影像成功上傳至儲存庫之後，請從檔案總管索引標籤中選擇重新整理總管來重新整理 AWS 工具組。AWS 然後在 AWS Cloud9 IDE 的 AWS 資源管理器菜單中可見。

從 Amazon ECR 提取映像檔

- 您可以將映像檔提取到 Docker tag 命令的本機執行個體。

```
docker pull AWS_account_id.dkr.ecr.region.amazonaws.com/hello-world:latest
```

Example

輸出如下。

```
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
```

從 Amazon ECR 儲存庫刪除映像檔

有兩種從 AWS Cloud9 IDE 刪除映像的方法。第一種方法是使用 AWS 資源管理器。

1. 從 AWS 檔案總管中，展開 ECR 功能表。
2. 展開您要從中刪除映像檔的儲存庫。
3. 開啟與您要刪除的映像檔關聯的映像檔標籤的內容選單 (按一下滑鼠右鍵)。
4. 若要刪除與該標記關聯的所有已儲存映像檔，請選擇 Delete Tag... (刪除標記...)。

使用 AWS CLI 刪除映像

- 您也可以使用 AWS `ecr batch-delete-image` 指令從儲存庫中刪除影像。

```
aws ecr batch-delete-image \
  --repository-name hello-world \
  --image-ids imageTag=latest
```

Example

輸出如下。

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "latest",
      "imageDigest":
"sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
    }
  ]
}
```

從您的 Amazon ECR 執行個體刪除儲存庫

有兩種從 AWS Cloud9 IDE 刪除儲存庫的方法。第一種方法是使用 AWS 資源管理器：

1. 從 AWS 檔案總管中，展開 ECR 功能表。
2. 開啟您要刪除之儲存庫的內容選單 (按一下滑鼠右鍵)。
3. 選擇 Delete Repository... (刪除儲存庫...)。

從 CLI 刪除 Amazon ECR 儲存庫 AWS

- 您可以使用 `AWS ecr delete-repository` 命令刪除儲存庫。

Note

如果不先刪除儲存庫中包含的映像檔，通常無法刪除儲存庫。然而，如果您新增 `—force` 旗標，則可在一個步驟內刪除儲存庫及其所有映像檔。

```
aws ecr delete-repository \  
--repository-name hello-world \  
--force
```

Example

輸出如下。

```
--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

若 AWS IoT 在 AWS Cloud9 IDE 中，您可以與 AWS IoT 服務互動，同時將 AWS Cloud9 工作流程的中斷情形降到最低。本指南說明如何開始使用 AWS Cloud9 IDE 的 AWS IoT 服務功能。如需詳細資訊，請參閱《AWS IoT 開發人員指南》中的[什麼是 AWS IoT ?](#) 章節。

AWS IoT 先決條件

若要開始使用 AWS Cloud9 IDE 的 AWS IoT，請先確認您的 AWS 帳戶和 AWS Cloud9 設定符合所有要求。如需詳細了解 AWS IoT 服務的特定 AWS 帳戶需求和 AWS 使用者權限，請參閱《AWS IoT 開發人員指南》中的[開始使用 AWS IoT 核心](#)。

AWS IoT 物件

AWS IoT 將裝置連接到 AWS 服務和 AWS 資源。您可以使用稱為物件的物件，將裝置連接到 AWS IoT。物件是特定裝置或邏輯實體的代表。它可以是實體裝置或感應器 (例如燈泡或牆上的開關)。如需有關 AWS IoT 物件的詳細資訊，請參閱《AWS IoT 開發人員指南 AWS IoT》中的[使用 AWS IoT 管理裝置](#)。

管理 AWS IoT 物件

AWS Cloud9 IDE 具有多種功能，可以協助您更有效率地進行管理。若要管理您的 AWS IoT 物件，請遵循下列步驟：

- [Create a thing](#)
- [Attach a certificate to a thing](#)
- [Detach a certificate from a thing](#)
- [Delete a thing](#)

如何建立物件

1. 從 AWS Explorer 中，展開 IoT 服務區段。

2. 開啟 (按一下右鍵) 該 thing (物件) 的內容選單，然後選擇 Create Thing (建立物件)。
3. 在 Thing Name (物件名稱) 欄位中輸入 thing (物件) 的名稱，然後遵照提示操作。
4. 完成此步驟後，Thing (物件) 區段會顯示一個 thing icon (物件圖示)，後面有您指定的名稱。

將憑證連接到物件

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 在 Things (物件) 子區段下，找出您要連接憑證的 thing (物件)。
3. 開啟 (按一下右鍵) thing (物件) 的內容選單，接著從中選擇 Attach Certificate (連接憑證)，開啟包含憑證清單的輸入選擇器。
4. 從清單中，選擇與您要連接至物件的憑證對應的 certificate ID (憑證 ID)。
5. 完成此步驟後，您的憑證就能做為您所連接之物件項目，在 AWS Explorer 存取。

從物件中分離憑證

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 在 Things (物件) 子區段中，找出您要分離憑證的 thing (物件)。
3. 開啟 (按一下右鍵) 該 thing (物件) 的內容選單，然後選擇 Attach Certificate (連接憑證)。
4. 完成此步驟後，已分離的憑證便不再顯示於 AWS Explorer 中的物件下方。但是，您仍然可以從 Certificates (憑證) 子區段存取該憑證。

刪除物件

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 在 Things (物件) 子區段中，找出您要刪除的 thing (物件)。
3. 開啟 (按一下右鍵) 該 thing (物件) 的內容選單，然後選擇 Delete Thing (刪除物件)。
4. 完成此步驟後，已刪除的物件便無法從物件子區段存取。

Note

您只能刪除沒有連接憑證的物件。

AWS IoT 憑證

憑證是在 AWS IoT 服務與裝置之間建立安全連線的常用方法。X.509 憑證為數位憑證，其依據 X.509 公有金鑰基礎設施標準，將公有金鑰與憑證內含的身分建立關聯。如需 AWS IoT 憑證的資訊，請參閱《AWS IoT 開發人員指南》中的[身分驗證 \(IoT\)](#)。

管理憑證

此 AWS 工具組提供多種方式，讓您可以直接從 AWS Explorer 管理您的 AWS IoT 憑證。下列步驟提供進一步的詳細說明：

- [Create a certificate](#)
- [Change a certificate status](#)
- [Attach a policy to a certificate](#)
- [Delete a certificate](#)

建立 AWS IoT 憑證

X.509 憑證可用來連接您的 AWS IoT 執行個體。

1. 從 AWS Explorer 展開 IoT 服務區段，然後開啟 (按一下右鍵) Certificates (憑證)。
2. 若要開啟對話方塊，請從內容選單中選擇 Create Certificate (建立憑證)。
3. 若要儲存 RSA 金鑰對和 X.509 憑證，請從本機檔案系統中選取一個目錄。

Note

- 預設檔案名稱會包含憑證 ID 做為字首。
- 只有 X.509 憑證會透過 AWS IoT 服務與您的 AWS 帳戶一起儲存。
- 您的 RSA 金鑰對只能核發一次，請在出現提示時將其儲存到檔案系統的安全位置。
- 若憑證或金鑰對無法儲存到您的檔案系統，AWS 工具組會從您的 AWS 帳戶刪除憑證。

修改憑證狀態

個別憑證的狀態會顯示在 AWS Explorer 的憑證 ID 旁邊，且能設定為作用中、非作用中或已撤銷。

Note

- 憑證要處於作用中狀態，您才能使用該憑證將裝置連線至 AWS IoT 服務。
- 您可以重新啟動非作用中的憑證，無論其先前是否遭到停用或預設為非作用。
- 已撤銷的憑證則無法重新啟動。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
 2. 在 Certificates (憑證) 子區段中，找到您要修改的憑證。
 3. 開啟 (按一下右鍵) 憑證內容選單，該選單會顯示憑證可用的狀態變更選項。
- 若憑證狀態為非作用中，請選擇 activate (啟動)，將狀態變更為作用中。
 - 若憑證狀態為作用中，請選擇 deactivate (停用)，將狀態變更為非作用中。
 - 若憑證狀態為作用中或非作用中，請選擇 revoke (撤銷)，將狀態變更為已撤銷。

Note

若您選取的憑證顯示於物件子區段時有連接物件，您便可以使用這些狀態變更動作。

將 IoT 政策連接至憑證

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 在 Certificates (憑證) 子區段中，找到您要修改的憑證。
3. 開啟 (按一下右鍵) 憑證的內容選單，選擇 Attach Policy (連接政策)，開啟包含可用政策清單的輸入選擇器。
4. 選擇您要附加至憑證的政策。
5. 完成此步驟後，所選政策會新增至憑證，做為子選單項目。

將 IoT 政策從憑證分離

1. 從 AWS Explorer 中，展開 IoT 服務區段。

2. 在 Certificates (憑證) 子區段中，找到您要修改的憑證。
3. 展開憑證並找出您要分離的政策。
4. 開啟 (按一下右鍵) 政策內容選單，並選擇 Detach (分離)。
5. 完成此步驟後，便無法再從憑證存取該政策，但可以從 Policy (政策) 子區段存取。

刪除憑證

1. 從 AWS Explorer 中，展開 IoT 服務標題。
2. 在 Certificates (憑證) 子區段中，找到您要刪除的憑證。
3. 開啟 (按一下右鍵) 憑證的內容選單，從中選擇 Delete Certificate (刪除憑證)。

Note

若憑證已連接至物件或處於作用中狀態，則無法刪除。您可以刪除已連接政策的憑證。

AWS IoT 政策：

AWS IoT 核心政策是透過 JSON 文件定義。每項必須至少包含一個政策陳述式。政策會定義 AWS IoT、AWS，以及裝置的互動方式。如需詳細了解如何建立政策文件，請參閱《AWS IoT 開發人員指南》中的 [IoT 政策](#)。

Note

已命名政策皆受到版本控制，所以您可以復原。在 AWS Explorer 中，您的 IoT 政策會列在 AWS IoT 服務的 Policies (政策) 子區段下。您可以展開政策來檢視政策版本。預設版本以星號 (*) 表示。

管理政策

AWS Cloud9 IDE 提供多種管理 AWS IoT 服務政策的方法。以下提供多種方式，說明如何直接從 VS Code 的 AWS Explorer 管理或修改政策：

- [Create a policy](#)
- [Upload a new policy version](#)

- [Edit a policy version](#)
- [Change the policy version default](#)
- [Change the policy version default](#)

如要建立 AWS IoT 政策

Note

您可以從 AWS Explorer 建立新政策。不過，定義政策的 JSON 文件必須先存在於您的檔案系統中。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 開啟 (按一下右鍵) Policies (政策) 子區段的内容選單，並開啟 Policy Name (政策名稱) 輸入欄位，選擇 Create Policy from Document (從文件建立政策)。
3. 輸入名稱並依照提示開啟對話方塊，從檔案系統中選取 JSON 文件。
4. 選擇包含政策定義的 JSON 文件，完成此步驟後，該政策便可在 AWS explorer 中使用。

上傳新的 AWS IoT 政策版本

您可以將 JSON 文件上傳至政策，以建立新版本的政策。

Note

新的 JSON 文件必須在您的檔案系統中，才能使用 AWS Explorer 建立新版本。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，檢視您的 AWS IoT 政策。
3. 開啟 (按一下右鍵) 您要更新的政策之内容選單，並選擇 Create new version from Document (從文件建立新版本)。
4. 對話方塊開啟時，請選擇包含政策定義更新的 JSON 檔案。

您可以從 AWS Explorer 中的政策存取新版本。

編輯 AWS IoT 政策版本

您可以使用 AWS Cloud9 開啟和編輯政策文件。完成編輯後，請儲存至您的檔案系統。接著，從 AWS Explorer 將其上傳到您的 AWS IoT 服務。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，找出您要更新的政策。
3. 若要開啟 Policy Name (政策名稱)，請從 Document (文件) 中選擇 Create Policy (建立政策)。
4. 展開您要更新的政策，接著開啟 (按一下右鍵) 您要編輯之政策版本的內容選單。
5. 若要在 AWS Cloud9 中開啟政策版本，請從內容選單選擇 View (檢視)，以開啟政策版本。
6. 當政策文件開啟時，請編輯並儲存變更。

Note

此時，您對政策所做的變更只會儲存到本機檔案系統。若要更新版本並以 AWS Explorer 追蹤，請重複 [Upload a new policy version](#) 中的步驟。

選取新的政策版本預設值

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，找出您要更新的政策。
3. 展開您要更新的政策，接著開啟 (按一下右鍵) 您要設定之政策版本的內容選單，選擇 Set as Default (設為預設)。

完成此步驟後，您選取的新預設版本旁邊會有一個星號。

刪除 政策

Note

刪除政策或政策版本之前，請確認已符合下列條件：

- 若政策已連接至憑證，則無法刪除該政策。
- 若政策有任何非預設版本，則無法刪除該政策。
- 您只能在選取新的預設版本或刪除整個政策時，才能刪除政策的預設版本。

- 刪除整個政策之前，您須先刪除該政策的所有非預設版本。

1. 從 AWS Explorer 中，展開 IoT 服務區段。
2. 展開 Policies (政策) 子區段，找出您要更新的政策。
3. 展開您要更新的政策，開啟 (按一下右鍵) 您要刪除之政策版本的內容選單，選擇 Delete (刪除)。
4. 刪除版本後，該版本不會再顯示於 AWS Explorer。
5. 若只剩下政策的預設版本，請開啟父政策的內容選單 (按一下右鍵)，然後選擇 Delete (刪除)。

使用 Amazon Elastic Container Service

AWS Cloud9 IDE 針對 [Amazon Elastic Container Service \(Amazon ECS\)](#) 提供了一些支援。您可以使用 AWS Cloud9 IDE 來管理 Amazon ECS 資源。例如，您可以建立任務定義。

主題

- [AWS 工具包中適用於 AWS Cloud9 的 Amazon Elastic Container Service Exec](#)

AWS 工具包中適用於 AWS Cloud9 的 Amazon Elastic Container Service Exec

您可以使用適用於 AWS Cloud9 的 AWS 工具組，在 Amazon Elastic Container Service (Amazon ECS) 容器中發出單一命令。您可以使用 Amazon ECS Exec 功能進行此操作。

Important

啟用和停用 Amazon ECS Exec 會變更您 AWS 帳戶中的 ECS 資源狀態。變更包括停止和重新啟動服務。此外，在啟用 Amazon ECS Exec 時變更資源狀態，可能會導致無法預期的結果。如需詳細資訊，請參閱 Amazon ECS 開發人員指南中的[使用 Amazon ECS Exec 除錯](#)。

Amazon ECS Exec 的先決條件

您必須符合某些先決條件，才能使用 Amazon ECS Exec 功能。

Amazon ECS 要求

ECS Exec 具有版本要求，取決於您的任務是否託管於 Amazon EC2 或 AWS Fargate (Fargate)：

- 如果您使用的是 Amazon EC2，則必須使用在 2021 年 1 月 20 日之後發行的 Amazon ECS 最佳化 AMI，且代理程式版本為 1.50.2 或更高版本。如需詳細資訊，請參閱 Amazon ECS 開發人員指南中的 [Amazon ECS 最佳化 AMI](#)。
- 如果您使用的是 AWS Fargate，則必須使用平台版本 1.4.0 或更新版本。如需詳細資訊，請參閱 Amazon ECS 開發人員指南中的 [AWS Fargate 平台版本](#)。

AWS 帳戶組態和 IAM 許可

若要使用 Amazon ECS Exec 功能，您必須擁有與您的 AWS 帳戶 關聯的 Amazon ECS 叢集。Amazon ECS Exec 使用 Systems Manager，與叢集中的容器建立連線。Amazon ECS 需要特定的任務 IAM 角色許可才能與 SSM 服務通訊。

如需 Amazon ECS Exec 專屬的 IAM 角色和政策的相關資訊，請參閱 Amazon ECS 開發人員指南中的 [ECS Exec 需要的 IAM 許可](#)。

使用 Amazon ECS Exec

您可以直接從 AWS Cloud9 的 AWS 工具組中的 AWS Explorer 啟用或停用 Amazon ECS Exec。啟用 Amazon ECS Exec 後，請從 Amazon ECS 選單中選擇容器，然後對它們執行命令。

啟用 Amazon ECS Exec

1. 在 AWS Explorer 中，找出並展開 Amazon ECS 選單。
2. 使用您要修改的服務展開叢集。
3. 開啟服務的內容功能表 (按一下滑鼠右鍵)，然後選擇 Enable Command Execution (啟用命令執行)。

Important

此步驟會啟動服務的新部署，可能需要幾分鐘的時間。如需詳細資訊，請參閱本節開頭的附註。

停用 Amazon ECS Exec

1. 在 AWS Explorer 中，找出並展開 Amazon ECS 選單。
2. 展開包含所需服務的叢集。

3. 開啟服務的內容功能表 (按一下滑鼠右鍵)，然後選擇 **Disable Command Execution** (停用命令執行)。

Important

此步驟會啟動服務的新部署，可能需要幾分鐘的時間。如需詳細資訊，請參閱本節開頭的附註。

針對容器執行命令

若要使用 AWS Explorer 對容器執行命令，必須啟用 Amazon ECS Exec。如果未啟用，請參閱本節中的 [啟用 Amazon ECS Exec](#) 程序。

1. 在 AWS Explorer 中，找出並展開 Amazon ECS 選單。
2. 展開您想要的服務的叢集。
3. 展開服務以列出關聯的容器。
4. 開啟 (按一下滑鼠右鍵) 容器的內容選單，然後選擇 **Run Command in Container** (在容器中執行命令)。
5. 隨即開啟提示，其中包含執行中的任務清單。選擇您想要的任務 ARN。

Note

如果只有一項任務正在執行，則不會開啟提示。而是自動選取任務。

6. 出現提示時，輸入您要執行的命令，然後按 **Enter** 繼續。

使用 Amazon EventBridge

適用於 AWS Cloud9 的 AWS 工具組支援 [Amazon EventBridge](#)。適用於 AWS Cloud9 的 AWS 工具組可讓您使用 EventBridge 的部份功能，例如結構描述。

主題

- [使用 Amazon EventBridge 結構描述](#)

使用 Amazon EventBridge 結構描述

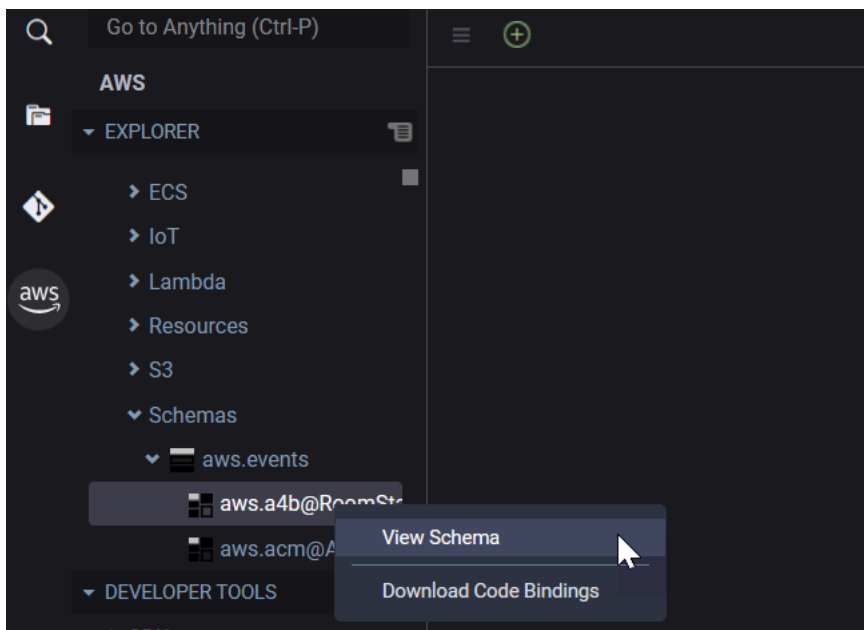
您可以使用 AWS Cloud9 的 AWS 工具組，對 [Amazon EventBridge 結構描述](#) 執行各種操作。

先決條件

您要使用的 EventBridge 結構描述必須位於您的 AWS 帳戶中。如果無法使用，請建立或上傳結構描述。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》<https://docs.aws.amazon.com/eventbridge/latest/userguide/> 中的 [Amazon EventBridge 結構描述](#)。

檢視可用的結構描述

1. 在 AWS Explorer 中，展開 Schemas (結構描述)。
2. 展開登錄的名稱，內含您要檢視的結構描述。例如，AWS 提供的許多結構描述都在 aws.events 登錄中。
3. 若要在編輯器中檢視結構描述，請開啟 (按一下右鍵) 結構描述的內容選單，然後選擇 View Schema (檢視結構描述)。



尋找可用的結構描述

在 AWS Explorer 中，執行下列一或多個動作：

- 開始輸入您要尋找的結構描述名稱。AWS Explorer 會反白包含相符項目的結構標題。(您必須展開登錄才能看到反白顯示的標題。)

- 開啟 (按一下右鍵) Schemas (結構描述) 的內容選單，然後選擇 Search Schemas (搜尋結構描述)。或者，展開 Schemas (結構描述)，在您要尋找的結構描述所在的登錄上，開啟 (按一下右鍵) 內容選單，然後選擇 Search Schemas in Registry (在登錄中搜尋結構描述)。在 EventBridge Schemas Search (EventBridge 結構描述搜尋) 對話方塊中，開始輸入您要尋找的結構描述名稱。此對話方塊會顯示包含相符項目的結構描述標題。

若要在對話方塊中顯示結構描述，請選取結構描述的標題。

為可用的結構描述產生程式碼

1. 在 AWS Explorer 中，展開 Schemas (結構描述)。
2. 展開登錄的名稱，內含您想產生程式碼的結構描述。
3. 在結構描述的標題上開啟 (按一下右鍵) 內容選單，然後選擇 Download code bindings (下載程式碼繫結)。
4. 在出現的精靈頁面中，選擇下列項目：
 - 結構描述的 Version (版本)
 - 程式碼繫結語言
 - 工作空間文件夾，供您在本機開發機器上存放產生的程式碼

AWS Cloud9 教學課程

剛接觸 AWS Cloud9 嗎？在 [入門：基本教學課程](#) 中進行 IDE 的導覽。

試用這些教學課程和範例程式碼，使用AWS Cloud9各種程式設計語言和AWS服務來增加您的知識和信心。

主題

- [AWS Command Line Interface和 aws-shell 教程 AWS Cloud9](#)
- [AWS CodeCommit教學課程 AWS Cloud9](#)
- [亞馬遜動態 B 教學 AWS Cloud9](#)
- [AWS CDK教學課程AWS Cloud9](#)
- [燈泡教程 AWS Cloud9](#)
- [WordPress教學課程AWS Cloud9](#)
- [爪哇教學 AWS Cloud9](#)
- [C ++ 教程 AWS Cloud9](#)
- [AWS Cloud9 的 Python 教學課程](#)
- [.NET 教學課程AWS Cloud9](#)
- [Node.js 教學課程 AWS Cloud9](#)
- [對於 PHP 的教程 AWS Cloud9](#)
- [紅寶石AWS Cloud9](#)
- [前往的教學課程 AWS Cloud9](#)
- [TypeScript教學課程AWS Cloud9](#)
- [碼頭教程 AWS Cloud9](#)
- [相關教學](#)

AWS Command Line Interface和 aws-shell 教程 AWS Cloud9

下面的教程使您可以在AWS Cloud9開發環境中設置AWS Command Line Interface (AWS CLI)，aws-shell 或兩者。AWS CLI 和 aws-shell 是統一的工具，可提供一致的界面，讓您與 AWS 的所有部分互動。您可以使用AWS CLI而不是快速執行AWS Management Console要與之互動的命令AWS，而且其中一些命令可以使用AWS CLI又或者使用AWS CloudShell。

如需 AWS CLI 的詳細資訊，請參閱 [AWS Command Line Interface 使用者指南](#)。如需了解 aws-shell，請參閱下列資源：

- 在網站上的 [aws-shell](#) GitHub
- pip 網站上的 [aws-shell](#)

如需可透過 AWS CLI 執行以便與 AWS 互動的命令清單，請參閱 [AWS CLI 命令參考](#)。除了啟動不含aws前置詞的指令外AWS CloudShell，您可以使用相同的指令。

建立此範例可能會向您的 AWS 帳戶收取費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

主題

- [必要條件](#)
- [步驟 1：在環境中安裝 AWS CLI、aws-shell 或同時安裝](#)
- [步驟 2：在環境中設定憑證管理](#)
- [步驟 3：在環境中使用 AWS CLI 或 aws-shell 執行基本命令](#)
- [步驟 4：清理](#)

必要條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱在 [AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱在 [AWS Cloud9 中開啟環境](#)。

步驟 1：在環境中安裝 AWS CLI、aws-shell 或同時安裝

在此步驟中，您會使用 AWS Cloud9 IDE 在環境中安裝 AWS CLI、aws-shell 或同時安裝兩者，如此就能執行命令來與 AWS 互動。

如果您使用 AWS Cloud9 EC2 開發環境，而且只想要使用 AWS CLI，您可以直接跳到 [步驟 3：在環境中使用 AWS CLI 或 aws-shell 執行基本命令](#)。這是因為 AWS CLI 已安裝在 EC2 環境中，且環境中已設定一組 AWS 存取憑證。如需詳細資訊，請參閱 [AWS 受管理的臨時憑](#)。

如果您並非使用 EC2 環境，請執行下列動作安裝 AWS CLI：

1. 在環境開啟的情況下，在 IDE 中檢查 AWS CLI 是否已安裝。在終端機中執行 `aws --version` 命令。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果已安裝 AWS CLI，則版本號碼會顯示，並附帶 Python 版本號碼和 Amazon EC2 執行個體或自有伺服器作業系統的版本號碼等資訊。如果已安裝 AWS CLI，請直接跳到 [步驟 2：在環境中設定憑證管理](#)。
2. 若要安裝 AWS CLI，請參閱 AWS Command Line Interface 使用者指南中的 [安裝 AWS Command Line Interface](#)。例如，若是執行 Amazon Linux 的 EC2 環境，請在終端機中逐一執行這三個命令以安裝 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.
```

如為執行 Ubuntu Server 的 EC2 環境，請改為在終端機中逐一執行這三個命令以安裝 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.
```

如果您想安裝 aws-shell，請執行下列動作：

1. 在環境開啟的情況下，在 IDE 中檢查 aws-shell 是否已安裝。在終端機中執行 `aws-shell` 命令。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果已安裝 aws-shell，會顯示 `aws>` 提示。如果已安裝 aws-shell，請直接跳到 [步驟 2：在環境中設定憑證管理](#)。
2. 若要安裝 aws-shell，您要使用 pip。若要使用 pip，您必須安裝 Python。

若要檢查是否已安裝 Python (並視需要加以安裝)，請遵循 Python 範例中 [步驟 1：安裝 Python](#) 的說明，然後返回本主題。

若要檢查是否已安裝 pip，請在終端機中執行 `pip --version` 命令。如果已安裝 pip，會顯示版本號碼。如果 pip 並未安裝，請在終端機中逐一執行下列三個命令加以安裝。

```
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.
```

- 若要使用 pip 安裝 aws-shell，請執行下列命令。

```
sudo pip install aws-shell
```

步驟 2：在環境中設定憑證管理

每次使用 AWS CLI 或 aws-shell 來呼叫 AWS 服務時，您都必須隨著呼叫提供一組登入資料。這些登入資料可判斷 AWS CLI 或 aws-shell 是否具備發出該呼叫的適當許可。如果登入資料未涵蓋適當許可，呼叫即會失敗。

如果您使用 AWS Cloud9 EC2 開發環境，您可以直接跳到 [步驟 3：在環境中使用 AWS CLI 或 aws-shell 執行基本命令](#)。這是因為 EC2 環境中已設定憑證。如需詳細資訊，請參閱 [AWS 受管理的臨時憑](#)。

如果您並非使用 EC2 環境，您必須手動將憑證存放到環境內。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

步驟 3：在環境中使用 AWS CLI 或 aws-shell 執行基本命令

在此步驟中，您會在環境中使用 AWS CLI 或 aws-shell，以在 Amazon S3 建立儲存貯體、列出可用的儲存貯體，然後刪除儲存貯體。

- 如果您想要使用 aws-shell 但尚未將其啟動，請執行 aws-shell 命令來啟動 aws-shell。系統會顯示 aws> 提示。
- 建立儲存貯體。使用 AWS CLI 執行 **aws s3 mb** 命令，或使用 aws-shell 執行 **s3 mb** 命令，並提供要建立的儲存貯體名稱。在此範例中，我們使用名為 cloud9-123456789012-bucket 的儲存貯體，其中 123456789012 是您的 AWS 帳戶 ID。如果您使用其他名稱，請在此步驟中全部取代為該名稱。

```
aws s3 mb s3://cloud9-123456789012-bucket # For the AWS CLI.
```

```
s3 mb s3://cloud9-123456789012-bucket # For the aws-shell.
```

Note

儲存貯體名稱在所有 AWS 中都必須是唯一的，而不只是在您的 AWS 帳戶中。上述建議的儲存貯體名稱可協助您想出唯一的儲存貯體名稱。如果您收到含有 `BucketAlreadyExists` 錯誤的訊息，您必須使用其他儲存貯體名稱重新執行命令。

- 列出可用的儲存貯體。使用 AWS CLI 執行 `aws s3 ls` 命令，或使用 `aws-shell` 執行 `s3 ls` 命令。隨即顯示可用的儲存貯體清單。
- 刪除儲存貯體。使用 AWS CLI 執行 `aws s3 rb` 命令，或使用 `aws-shell` 執行 `s3 rb` 命令，並提供要刪除的儲存貯體名稱。

```
aws s3 rb s3://cloud9-123456789012-bucket # For the AWS CLI.
s3 rb s3://cloud9-123456789012-bucket # For the aws-shell.
```

若要確認是否已刪除儲存貯體，請再次使用 AWS CLI 執行 `aws s3 ls` 命令，或使用 `aws-shell` 執行 `s3 ls` 命令。清單中應該不會再顯示已刪除的儲存貯體名稱。

Note

如果您想要繼續使用儲存貯體，就不需要刪除它。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的 [新增物件至儲存貯體](#)。另請參閱 AWS CLI 命令參考中的 [S3 命令](#)。(請記住，如果您不刪除儲存貯體，它可能導致您的 AWS 帳戶持續產生費用。)

若要繼續實驗 AWS CLI，請參閱 AWS Command Line Interface 使用者指南中的 [使用 Amazon Web Services](#) 及 [AWS CLI 命令參考](#)。若要繼續實驗 `aws-shell`，請參閱 [AWS CLI 命令參考](#)；請注意，命令開頭不需要 `aws` 字首。

步驟 4：清理

如果您使用 `aws-shell`，您可以執行 `.exit` 或 `.quit` 命令來停止使用。

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

AWS CodeCommit教學課程 AWS Cloud9

您可以使用AWS CodeCommit教學課程來設定與中的遠端程式碼儲存庫互動的AWS Cloud9開發環境CodeCommit。CodeCommit是一項原始程式碼控制服務，您可以用來在AWS 雲端. Git 如需有關CodeCommit 的詳細資訊，請參閱《[使用者指南](#)》[AWS CodeCommit](#)。

遵循本教學課程並建立此範例可能會導致您的AWS 帳戶. 其中包括 Amazon EC2 和CodeCommit. 等服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [AWS CodeCommit 定價](#)。

- [先決條件](#)
- [步驟 1：設定具有必要存取許可的 IAM 群組](#)
- [步驟 2：在 AWS CodeCommit 中建立儲存庫](#)
- [步驟 3：將環境連線至遠端儲存庫](#)
- [步驟 4：將遠端儲存庫複製到您的環境](#)
- [步驟 5：將檔案新增至儲存庫](#)
- [步驟 6：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：設定具有必要存取許可的 IAM 群組

假設您的AWS認證與您中的管理員使用者相關聯AWS 帳戶，而您想要使用該使用者來使用CodeCommit Then，請跳至[步驟 2：在中建立存放庫AWS CodeCommit](#)。

您可以使用 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#) 完成此步驟。

使用主控台設定具有必要存取許可的 IAM 群組

1. 如果您尚未登入 AWS Management Console，請現在登入。

針對此步驟，建議您在 AWS 帳戶使用管理員使用者的憑證來登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。

2. 開啟 IAM 主控台。方法如下：從主控台的導覽列選擇 Services (服務)，然後，選擇 IAM。

3. 選擇 Groups (群組)。

4. 選擇群組的名稱。

5. 在 Permissions (許可) 標籤上，針對 Managed Policies (受管政策)，選擇 Attach Policy (連接政策)。

6. 在政策名稱清單中，選取下列其中一個方塊：

- 選取 `AWSCodeCommitPowerUser` 取此選項可存取與儲存庫相關資源 CodeCommit 的所有功能。不過，這不允許您刪除 CodeCommit 儲存庫，或建立或刪除其他儲存庫相關資源 AWS 服務，例如 Amazon Events。CloudWatch
- 選取 `AWSCodeCommitFullAccess` 以完全控制中的 CodeCommit 儲存庫和相關資源 AWS 帳戶。這包括刪除儲存庫的能力。

如果您在清單中未看到上述任一政策名稱，請在 Filter (篩選條件) 方塊中輸入政策名稱來顯示它們。

7. 選擇 Attach Policy (連接政策)。

若要查看這些 AWS 受管政策給予群組的存取許可清單，請參閱 AWS CodeCommit 使用者指南中的 [AWS CodeCommit 的 AWS 受管 \(預先定義\) 政策](#)。

請直接跳到 [步驟 2：在 AWS CodeCommit 中建立儲存庫](#)。

使用 AWS CLI 設定具備必要存取許可的 IAM 群組

執行 IAM `attach-group-policy` 命令，並指定群組名稱以及描述必要存取許可的 AWS 受管政策的 Amazon Resource Name (ARN)。語法如下。

```
aws iam attach-group-policy --group-name MyGroup --policy-arn POLICY_ARN
```

在上述命令中，將 `MyGroup` 取代為群組的名稱。將 `POLICY_ARN` 取代為 AWS 受管政策的 ARN：

- `arn:aws:iam::aws:policy/AWSCodeCommitPowerUser` 以存取與儲存庫相關資源 CodeCommit 的所有功能。不過，它不允許您刪除 CodeCommit 儲存庫，或建立或刪除其他儲存庫相關資源 AWS 服務，例如 Amazon Events。CloudWatch
- `arn:aws:iam::aws:policy/AWSCodeCommitFullAccess` 以完 CodeCommit 全控制 AWS 帳戶。這包括刪除儲存庫的能力。

若要查看這些 AWS 受管政策給予群組的存取許可清單，請參閱 AWS CodeCommit 使用者指南中的 [AWS CodeCommit 的 AWS 受管 \(預先定義\) 政策](#)。

步驟 2：在中建立儲存庫 CodeCommit

在此步驟中，您將使用 CodeCommit 主控台在 CodeCommit 中建立遠端程式碼儲存庫。

如果您已經有 CodeCommit 儲存庫，請直接跳到 [步驟 3：將環境連線至遠端儲存庫](#)。

您可以使用 [AWS Management Console](#) 或 [AWS 命令列界面 \(AWS CLI\)](#) 完成此步驟。

CodeCommit 使用主控台建立存放庫

1. 假設您以先前步驟的管理員使用者身分登入 AWS Management Console，而且不想要使用管理員使用者來建立儲存庫。然後，登出 AWS Management Console。
2. 開啟 CodeCommit 主控台，網址為 <https://console.aws.amazon.com/codecommit>。
3. 在主控台的導覽列中，使用區域選擇器來選擇您想要建立儲存庫的 AWS 區域，例如 US East (Ohio) (美國東部 (俄亥俄))。
4. 若顯示歡迎頁面，請選擇 Get started (開始使用)。否則，請選擇 Create repository (建立儲存庫)。
5. 在 Create repository (建立儲存庫) 頁面的 Repository name (儲存庫名稱) 中，輸入新儲存庫的名稱 (例如 MyDemoCloud9Repo)。如果您選擇其他名稱，請將它在此範例中皆予以取代。
6. (選用) 針對 Description (描述)，輸入關於儲存庫的描述。例如，您可以輸入：This is a demonstration repository for the AWS Cloud9 sample.
7. 選擇 Create repository (建立儲存庫)。即會顯示 Connect to your repository (連線至儲存庫) 窗格。選擇 Close (關閉)，因為您將以本主題稍後的不同方式連線至儲存庫。

請直接跳到「[步驟 3：將環境連線至遠端儲存庫](#)」。

使用中建立CodeCommit存放庫 AWS CLI

執行 AWS CodeCommit `create-repository` 命令。指定儲存庫名稱、選用描述，以及要在其中建立儲存庫的 AWS 區域。

```
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，將 `us-east-2` 取代為要建立儲存庫的 AWS 區域 ID。如需支援的區域清單，請參閱 [AWS CodeCommit](#) 中的 Amazon Web Services 一般參考。

如果您選擇使用其他儲存庫名稱，請將它在此範例中皆予以取代。

步驟 3：將環境連線至遠端儲存庫

在此步驟中，您可以使用 AWS Cloud9 IDE 連線至您在上一個步驟中建立或識別的 CodeCommit 存放庫。

Note

如果您偏好透過視覺化界面使用 Git，則可以複製遠端儲存庫。然後，您可以使用 IDE 中提供的 [Git 面板](#) 功能來新增檔案。

根據您的 AWS Cloud9 開發環境類型，完成下列其中一組程序。

環境類型	遵循下列程序
EC2 環境	<ol style="list-style-type: none">從 IDE 的終端機工作階段，執行下列兩個命令： <pre>git config --global credentia l.helper '!aws codecommit credentia l-helper \$@' git config --global credentia l.UseHttpPath true</pre> <p>如需詳細資訊，請在 AWS CodeCommit 使用者指南的整合 AWS Cloud9 與 AWS</p>

環境類型	遵循下列程序
	<p>CodeCommit 參閱步驟 2：在 AWS Cloud9 EC2 開發環境上設定 AWS CLI 憑證協助程式。</p> <ol style="list-style-type: none">請直接跳到本主題稍後的步驟 4：將遠端儲存庫複製到您的環境。
SSH 環境	<ol style="list-style-type: none">如果 Git 尚未安裝在環境中，請在 IDE 中使用終端機工作階段安裝。如需詳細資訊，請在《AWS CodeCommit 使用者指南》的透過 SSH 連線與 Linux、macOS 或 Unix 的 AWS CodeCommit 儲存庫連線的設定步驟參閱步驟 2：安裝 Git。完成 AWS CodeCommit 使用者指南中的透過 SSH 連線與 Linux、macOS 或 Unix 的 AWS CodeCommit 儲存庫連線的設定步驟的步驟 3：在 Linux、macOS 或 Unix 上設定憑證。 <p>系統指示您登入 AWS Management Console 並開啟 IAM 主控台時，建議您使用 AWS 帳戶帳戶中的 IAM 管理員使用者憑證登入。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。</p> <ol style="list-style-type: none">請直接跳到本主題稍後的步驟 4：將遠端儲存庫複製到您的環境。

步驟 4：將遠端儲存庫複製到您的環境

在此步驟中，您可以使用 AWS Cloud9 IDE 將中的遠端存放庫複製 CodeCommit 到您的環境中。

若要複製儲存庫，請執行 `git clone` 命令。將 `CLONE_URL` 取代為儲存庫的複製 URL。

```
git clone CLONE_URL
```

如果是 EC2 環境，您必須提供以 `https://` 開頭的 HTTPS 複製 URL。如果是 SSH 環境，您必須提供以 `ssh://` 開頭的 SSH 複製 URL。

若要取得儲存庫的完整複製 URL，請參閱 AWS CodeCommit 使用者指南中的[使用 AWS CodeCommit 主控台檢視儲存庫詳細資訊](#)。

如果您的儲存庫未含任何檔案，則會顯示警告訊息，例如 `You appear to have cloned an empty repository.`。這是預期的情況。您稍後會解決。

步驟 5：將檔案新增至儲存庫

在此步驟中，您將在 AWS Cloud9 環境的複製儲存庫中建立三個簡單的檔案。接著，您可以將檔案新增至所複製儲存庫中的 Git 預備區域。最後，您提交暫存文件並將提交推送到中的遠程存儲庫 CodeCommit。

如果複製的儲存庫中已經有檔案，表示您已完成，而且可以略過此範例的其餘部分。

將檔案新增至儲存庫

1. 建立新檔案。在選單列上，選擇 File (檔案)、New File (新增檔案)。
2. 將下列內容輸入至檔案中，然後選擇 File (檔案)、Save (儲存) 以在 AWS Cloud9 環境的 MyDemoCloud9Repo 目錄中將檔案儲存為 `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

若要確認您將此檔案儲存在正確的目錄，請在 Save As (另存新檔) 對話方塊中，選擇 MyDemoCloud9Repo 資料夾。然後，確保 Folder (資料夾) 顯示 / MyDemoCloud9Repo。

3. 使用以下內容建立兩個名為 `insect.txt` 和 `reptile.txt` 的檔案。將檔案儲存在相同的 MyDemoCloud9Repo 目錄中。

```
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. 在終端機中，執行 **cd** 命令以切換至 MyDemoCloud9Repo 目錄。

```
cd MyDemoCloud9Repo
```

5. 執行 **git status** 命令以確認檔案已成功儲存至 MyDemoCloud9Repo 目錄中。這三個檔案都會列為未追蹤檔案。

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    bird.txt
    insect.txt
    reptile.txt
```

6. 執行 **git add** 命令將檔案新增至 Git 預備區域。

```
git add --all
```

7. 再次執行 **git status** 命令以確認檔案已成功新增至 Git 預備區域。這三個檔案現在已列為將遞交的變更。

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   bird.txt
    new file:   insect.txt
    new file:   reptile.txt
```

8. 執行 **git commit** 命令以遞交預備檔案。

```
git commit -m "Added information about birds, insects, and reptiles."
```

9. CodeCommit 通過運行 **git push** 命令將提交推送到您的遠程存儲庫。

```
git push -u origin master
```

10. 確認是否已成功推送檔案。開啟 CodeCommit 主控台 (若尚未開啟)，網址為 <https://console.aws.amazon.com/codecommit>。
11. 在頂端導覽列中，接近右邊緣處，選擇您已建立儲存庫的 AWS 區域，例如 US East (Ohio) (美國東部 (俄亥俄))。
12. 在「儀表板」頁面上，選擇「MyDemoCloud9Repo」。即會顯示三個檔案。

若要繼續試驗 CodeCommit 儲存庫，請參閱《AWS CodeCommit 使用指南》中的「[瀏覽存放庫內容](#)」。

如果您是新手，Git 並且不想弄亂您的 CodeCommit 儲存庫，[請嘗試在 Try Git](#) 網站上的示例 Git 儲存庫。

步驟 6：清除

若要避免在使用此範例 AWS 帳戶後向您收取持續費用，請刪除 CodeCommit 存放庫。如需說明，請參閱 AWS CodeCommit 使用者指南中的[刪除 AWS CodeCommit 儲存庫](#)。

確定也一併刪除環境。如需相關說明，請參閱[刪除環境](#)。

亞馬遜動態 B 教學 AWS Cloud9

本教學可讓您設定 AWS Cloud9 開發環境以與 Amazon DynamoDB 搭配使用。

DynamoDB 是全受管的 NoSQL 資料庫服務。您可以使用 DynamoDB 建立資料庫資料表，藉此存放和擷取任意數量的資料，並為任何層級的請求流量提供服務。DynamoDB 會自動將資料表的資料和流量分散到足夠數量的伺服器，以便處理指定的請求容量和存放的資料量，同時保持一致且快速的效能。如需詳細資訊，請參閱 AWS 網站上的 [Amazon DynamoDB](#)。

建立此範例可能會向您的 AWS 帳戶收取費用。其中包括 Amazon EC2 和 DynamoDB 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#) 和 [Amazon DynamoDB 定價](#)。

如需其他 AWS 資料庫產品的相關資訊，請參閱 AWS 網站上的 [亞馬遜關聯式資料庫服務 \(RDS\)](#)、ElastiCache、[亞馬遜和亞馬遜紅移](#)。另請參閱 AWS 網站上的 [AWS Database Migration Service](#)。

- [先決條件](#)
- [步驟 1：在環境中安裝和設定 AWS CLI、AWS CloudShell，或兩者兼具](#)

- [步驟 2：建立資料表](#)
- [步驟 3：新增項目到資料表](#)
- [步驟 4：新增多個項目到資料表](#)
- [步驟 5：建立全域輔助索引](#)
- [步驟 6：取得資料表中的項目](#)
- [步驟 7：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：在環境中安裝和設定 AWS CLI、AWS CloudShell，或兩者兼具

在此步驟中，您會使用 AWS Cloud9 IDE 在環境中安裝和設定 AWS CLI、AWS CloudShell，或兩者兼具，以便執行命令來和 DynamoDB 互動。然後，使用 AWS CLI 執行基本 DynamoDB 命令來測試您的安裝和組態。

1. 若要設定 AWS CLI 或 AWS CloudShell 的憑證管理，以及在環境中安裝 AWS CLI、AWS CloudShell 或兩者皆安裝，請遵循 [AWS CLI 和 AWS CloudShell 範例](#) 的步驟 1 和 2，然後返回本主題。如果您已經在環境中安裝和設定 AWS CLI、AWS CloudShell，或兩者皆有，則不需要再次執行。
2. 在環境的終端機工作階段執行 **list-tables** 命令，以列出現有 DynamoDB 資料表 (如果有的話)，藉此測試 AWS CLI、aws-shell 或兩者的安裝和組態。若要啟動新終端機工作階段，請在選單列上，選擇 Windows (視窗)、New Terminal (新增終端機)。

```
aws dynamodb list-tables # For the AWS CLI.
dynamodb list-tables     # For the aws-shell.
```

Note

在此範例中，如果您使用 `aws-shell`，則會省略開頭為 `aws` 的每個命令中的 `aws`。若要開始 `aws-shell`，請執行 `aws-shell` 命令。若要停止使用 `aws-shell`，請執行 `.exit` 或 `.quit` 命令。

如果此命令成功，則輸出的 `TableNames` 陣列會包含您可能已有的現有 DynamoDB 資料表清單。如果您還沒有 DynamoDB 資料表，則 `TableNames` 陣列會是空白。

```
{
  "TableNames": []
}
```

如果您有任何 DynamoDB 資料表，則 `TableNames` 陣列會包含資料表名稱清單。

步驟 2：建立資料表

在此步驟中，您會在 DynamoDB 中建立資料表，並指定資料表名稱、配置、簡單主索引鍵和資料輸送量設定。

這個名為 `Weather` 的範例資料表包含有關美國幾個城市的氣象預測資訊。此資料表包含下列類型的資訊 (在 DynamoDB 中，每個資訊片段都稱為屬性)：

- 必要的唯一城市 ID (`CityID`)
- 必要的預測日期 (`Date`)
- 城市名稱 (`City`)
- 州名稱 (`State`)
- 預測氣象條件 (`Conditions`)
- 預測溫度 (`Temperatures`)
 - 預測華氏高溫 (`HighF`)
 - 預測華氏低溫 (`LowF`)

若要建立資料表，請在 AWS Cloud9 IDE 的終端機工作階段中執行 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` 代表資料表名稱 (在此範例中為 `Weather`)。在 AWS 帳戶的每個 AWS 區域內，資料表名稱不可重複。
- `--attribute-definitions` 代表用來唯一識別資料表項目的屬性。此資料表的每個項目都是以數字 ID 屬性與 `Date` 屬性 (以 ISO-8601 格式字串呈現) 的組合予以唯一識別。
- `--key-schema` 代表資料表的索引鍵結構描述。此資料表具有 `CityID` 和 `Date` 的複合主索引鍵。這表示每個資料表項目都必須有 `CityID` 屬性值和 `Date` 屬性值，但資料表中不能有相同 `CityID` 屬性值和 `Date` 屬性值的兩個項目。
- `--provisioned-throughput` 代表資料表的讀寫容量。DynamoDB 允許最大達 4 KB 的項目每秒最多 5 次強式一致性讀取，或最大達 4 KB 的項目每秒最多 5 次最終一致讀取。DynamoDB 也允許最大達 1 KB 的項目每秒最多 5 次寫入。

Note

設定較高的佈建輸送量可能會使您的 AWS 帳戶產生額外的費用。
如需上述操作和其他 DynamoDB 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 [dynamodb](#)。

如果此命令成功，則會顯示所建立之新資料表的摘要資訊。若要確認已成功建立資料表，請執行 DynamoDB **describe-table** 命令，指定資料表名稱 (`--table-name`)。

```
aws dynamodb describe-table --table-name Weather
```

成功建立資料表時，`TableStatus` 值會從 `CREATING` 變更為 `ACTIVE`。除非成功建立資料表，否則請不要跳過此步驟。

步驟 3：新增項目到資料表

在此步驟中，您會新增項目到剛建立的資料表。

1. 建立名為 `weather-item.json` 且具有下列內容的檔案。若要建立新檔案，請在選單列上，選擇 `File (檔案)`、`New File (新增檔案)`。若要儲存檔案，請選擇 `File (檔案)`、`Save (儲存)`。

```
{
  "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" }
  }
}
```

在此程式碼中，N 代表 number 屬性值。S 是 string 屬性值。M 是 map 屬性，其為一對屬性值組。只要您使用項目，就必須指定屬性的資料類型。如需了解其他可用的屬性資料類型，請參閱 Amazon DynamoDB 開發人員指南中的 [資料類型](#)。

2. 執行 DynamoDB `put-item` 命令，並指定資料表名稱 (`--table-name`) 以及 JSON 格式的項目 (`--item`) 路徑。

```
aws dynamodb put-item \
--table-name Weather \
--item file://weather-item.json
```

如果命令成功，則執行時未發生錯誤，而且未顯示任何確認訊息。

3. 若要確認資料表的目前內容，請執行 DynamoDB `scan` 命令，並指定資料表名稱 (`--table-name`)。

```
aws dynamodb scan --table-name Weather
```

如果命令成功，則會顯示資料表和您剛新增之項目的摘要。

步驟 4：新增多個項目到資料表

在此步驟中，您會新增數個項目到 Weather 資料表。

1. 建立名為 `more-weather-items.json` 且具有下列內容的檔案。

```
{
  "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" }
      }
    }
  }
}
```

```
    }  
  }  
}  
]  
}
```

在此程式碼中，8 個 Item 物件定義新增到資料表的 8 個項目，這與先前步驟中定義的單一項目類似。不過，當您在後續步驟中執行 DynamoDB **batch-write-item** 命令時，必須提供 JSON 格式的物件，其中包括包含 PutRequest 物件的每個 Item 物件。然後，您必須將這些 PutRequest 物件包含在名稱與資料表相同的父陣列中。

2. 執行 DynamoDB **batch-write-item** 命令，指定要新增的 JSON 格式項目 (`--request-items`) 的路徑。

```
aws dynamodb batch-write-item \  
--request-items file://more-weather-items.json
```

如果命令成功，則會顯示下列訊息，確認已成功新增項目。

```
{  
  "UnprocessedItems": {}  
}
```

3. 若要確認資料表的目前內容，請再次執行 DynamoDB **scan** 命令。

```
aws dynamodb scan --table-name Weather
```

如果命令成功，則現在會顯示 9 個項目。

步驟 5：建立全域輔助索引

執行 DynamoDB **scan** 命令來取得項目的相關資訊可能會很慢，尤其在資料表的大小增加或您要取得的資訊類型較複雜時更是如此。您可以建立一或多個輔助索引，以加速進行，並讓取得資訊更容易。在此步驟中，您會了解執行上述作業時，DynamoDB 支援的兩種輔助索引類型。這些稱為「本機輔助索引」和「全域輔助索引」。然後，您會建立全域輔助索引。

若要了解這些輔助索引類型，您需要先了解可識別資料表中唯一項目的主索引鍵。DynamoDB 支援簡單主索引鍵或複合主索引鍵。簡單主索引鍵具有單一屬性，而資料表中每個項目的屬性值都必須是唯

一。此屬性也稱為分割區索引鍵 (或雜湊屬性)，DynamoDB 可使用此屬性來分割項目，以便加速存取。資料表也可以有複合主索引鍵，其中包含兩個屬性。第一個屬性是分割區索引鍵，而第二個屬性是「排序索引鍵」(也稱為「範圍屬性」)。在具有複合主索引鍵的資料表中，任何兩個項目都可以有相同的分割區索引鍵值，但不能同時也有相同的排序索引鍵值。Weather 資料表具有複合主索引鍵。

本機輔助索引具有與資料表本身相同的分割區索引鍵，但此索引類型可以有不同的排序索引鍵。全域輔助索引可以有與資料表本身不同的分割區索引鍵和排序索引鍵。

例如，您可能已使用主索引鍵，以 Weather 存取 CityID 項目。若要以 Weather 存取 State 項目，您可以建立具有分割區索引鍵 CityID (必須與資料表本身相同) 和排序索引鍵 State 的本機輔助索引。若要以 Weather 存取 City 項目，您可以建立具有分割區索引鍵 City 和排序索引鍵 Date 的全域輔助索引。

您只有在建立資料表時才能建立本機輔助索引。因為 Weather 資料表已存在，所以您無法新增任何本機輔助索引到資料表。不過，您可以新增全域輔助索引。請現在練習新增一個全域輔助索引。

Note

建立輔助索引可能會使您的 AWS 帳戶產生額外的費用。

1. 建立名為 weather-global-index.json 且具有下列內容的檔案。

```
[
  {
    "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
}
}
}
]

```

在此程式碼中：

- 全域輔助索引的名稱為 `weather-global-index`。
- `City` 屬性是分割區索引鍵 (雜湊屬性)，而 `Date` 屬性是排序索引鍵 (範圍屬性)。
- 如果每個項目都符合使用此索引的資料表搜尋，則 `Projection` 預設會定義要擷取的屬性 (除了擷取雜湊屬性和任何範圍屬性)。在此範例中，會擷取每個相符項目的 `State`、`Conditions`、`HighF` (`Temperatures` 的一部分) 和 `LowF` (也是 `Temperatures` 的一部分) 屬性 (以及 `City` 和 `Date` 屬性)。
- 與資料表類似，全域輔助索引必須定義其佈建輸送量設定。
- `IndexName`、`KeySchema`、`Projection` 和 `ProvisionedThroughput` 設定都必須包含在 `Create` 物件中，而此物件定義您要在後續步驟中執行 `DynamoDB update-table` 命令時建立的全域輔助索引。

2. 執行 `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` 是要更新的資料表名稱。
- `--attribute-definitions` 是要包含在索引中的屬性。分割區索引鍵一律會先列出，而任何排序索引鍵一律會後列出。
- `--global-secondary-index-updates` 是定義全域輔助索引之檔案的路徑。

如果此命令成功，則會顯示所建立之新全域輔助索引的摘要資訊。若要確認已成功建立全域輔助索引，請執行 DynamoDB **describe-table** 命令，並指定資料表名稱 (--table-name)。

```
aws dynamodb describe-table --table-name Weather
```

成功建立全域輔助索引時，TableStatus 值會從 UPDATING 變更為 ACTIVE，而 IndexStatus 值會從 CREATING 變更為 ACTIVE。除非成功建立全域輔助索引，否則請不要跳過此步驟。這可能需要幾分鐘的時間。

步驟 6：取得資料表中的項目

有多種方法可以取得資料表中的項目。在此步驟中，您會使用資料表的主索引鍵、資料表的其他屬性和全域輔助索引來取得項目。

根據項目的主索引鍵值來取得資料表的單一項目

如果您知道項目的主索引鍵值，則可以執行 DynamoDB 命令 **get-item**、**scan** 或 **query** 來取得相符項目。以下是這些命令的主要差異：

- **get-item** 會為具有指定主索引鍵的項目傳回一組屬性。
- **scan** 會存取資料表中的每個項目或輔助索引，以傳回一或多個項目和項目屬性。
- **query** 會根據主索引鍵值尋找項目。您可以查詢任何具有複合主索引鍵 (一個分割區索引鍵及一個排序索引鍵) 的資料表或輔助索引。

在此範例中，以下說明如何使用所有這些命令來取得項目，其包含 CityID 屬性值 1 和 Date 屬性值 2017-04-12。

1. 若要執行 DynamoDB **get-item** 命令，請指定資料表名稱 (--table-name)、主索引鍵值 (--key) 以及要顯示的項目屬性值 (--projection-expression)。由於 Date 是 DynamoDB 保留的關鍵字，您也必須為 Date 屬性值 (--expression-attribute-names) 提供別名。(State 也是保留的關鍵字，您會在後續步驟看到專為它提供的別名)。

```
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" }'
```

在這個和其他命令中，若要顯示項目的所有屬性，請不要包含 `--projection-expression`。在此範例中，因為您未包含 `--projection-expression`，所以也不需要包含 `--expression-attribute-names`。

```
aws dynamodb get-item \  
--table-name Weather \  
--key '{ "CityID": { "N": "1" }, "Date": { "S": "2017-04-12" } }'
```

2. 若要執行 DynamoDB **scan** 命令，請指定：

- 資料表名稱 (`--table-name`)。
- 要執行的搜尋 (`--filter-expression`)。
- 要使用的搜尋條件 (`--expression-attribute-values`)。
- 要為相符項目顯示的屬性類型 (`--select`)。
- 要顯示之項目的屬性值 (`--projection-expression`)。
- 如果您有任何屬性正在 DynamoDB 中使用保留的關鍵字，請為這些屬性建立別名 (`--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. 若要執行 DynamoDB **query** 命令，請指定：

- 資料表名稱 (`--table-name`)。
- 要執行的搜尋 (`--key-condition-expression`)。
- 要用於搜尋的屬性值 (`--expression-attribute-values`)。
- 要為相符項目顯示的屬性類型 (`--select`)。
- 要顯示之項目的屬性值 (`--projection-expression`)。
- 如果您有任何屬性正在 DynamoDB 中使用保留的關鍵字，請為這些屬性建立別名 (`--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" } }' \  
--select SPECIFIC_ATTRIBUTES \  
--projection-expression \  
  "City, #D, Conditions, Temperatures.HighF, Temperatures.LowF" \  
--expression-attribute-names '{ "#D": "Date" }'
```

請注意，**scan** 命令需要掃描全部 9 個項目才能取得結果，而 **query** 命令只需要掃描 1 個項目。

根據項目的主索引鍵值來取得資料表的多個項目

如果您知道項目的主索引鍵值，則可以執行 DynamoDB **batch-get-item** 命令來取得相符項目。在此範例中，以下說明如何取得項目，其包含 CityID 屬性值 3 和 Date 屬性值 2017-04-13 或 2017-04-14。

執行 DynamoDB **batch-get-item** 命令，並指定檔案路徑來描述要取得的項目 (**--request-items**)。

```
aws dynamodb batch-get-item --request-items file://batch-get-item.json
```

在此範例中，batch-get-item.json 檔案中的程式碼指定搜尋 Weather 資料表中 CityID 為 3 且 Date 為 2017-04-13 或 2017-04-14 的項目。對於找到的每個項目，會顯示 City、State、Date 和 HighF (Temperatures 的一部分) 的屬性值 (如果存在的話)。

```
{  
  "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" }
}
}
```

從資料表取得所有相符項目

如果您知道資料表中屬性值的某個資訊，則可以執行 DynamoDB **scan** 命令來取得相符項目。在此範例中，以下說明如何取得 Conditions 屬性值包含 Sunny 而 HighF 屬性值 (Temperatures 的一部分) 大於 53 的日期。

執行 DynamoDB **scan** 命令，並指定：

- 資料表名稱 (--table-name)。
- 要執行的搜尋 (--filter-expression)。
- 要使用的搜尋條件 (--expression-attribute-values)。
- 要為相符項目顯示的屬性類型 (--select)。
- 要顯示之項目的屬性值 (--projection-expression)。
- 如果您有任何屬性正在 DynamoDB 中使用保留的關鍵字，請為這些屬性建立別名 (--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" }'
```

從全域輔助索引取得所有相符項目

若要使用全域輔助索引搜尋，請使用 DynamoDB **query** 命令。在此範例中，以下說明如何使用 weather-global-index 輔助索引以取得名為 Portland 的城市在日期 2017-04-13 和 2017-04-14 的預測條件。

執行 DynamoDB **query** 命令，並指定：

- 資料表名稱 (--table-name)。
- 全域輔助索引的名稱 (--index-name)。
- 要執行的搜尋 (--key-condition-expression)。
- 要用於搜尋的屬性值 (--expression-attribute-values)。
- 要為相符項目顯示的屬性類型 (--select)。
- 如果您有任何屬性正在 DynamoDB 中使用保留的關鍵字，請為這些屬性建立別名 (--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" }'
```

步驟 7：清除

為避免結束使用此範例後您的 AWS 帳戶仍持續產生費用，您應刪除資料表。刪除資料表也會刪除全域輔助索引。您也應該刪除環境。

若要刪除資料表，請執行 DynamoDB **delete-table** 命令，並指定資料表名稱 (--table-name)。

```
aws dynamodb delete-table --table-name Weather
```

如果命令成功，則會顯示資料表的相關資訊 (包含 TableStatus 值 DELETING)。

若要確認已成功刪除資料表，請執行 DynamoDB **describe-table** 命令，並指定資料表名稱 (--table-name)。

```
aws dynamodb describe-table --table-name Weather
```

如果已成功刪除資料表，則會顯示包含 Requested resource not found 字詞的訊息。

若要刪除環境，請參閱[刪除環境](#)。

AWS CDK教學課程AWS Cloud9

本自學課程將向您展示如何使用AWS Cloud Development Kit (AWS CDK)在一個AWS Cloud9開發環境。AWS CDK 是一組軟體工具和程式庫，開發人員可使用此套件以程式碼形式為 AWS 基礎設施元件建造模型。

AWS CDK 包含 AWS 建構程式庫，可用來快速解決許多 AWS 上的任務。例如，您可以使用 Fleet 建構，完全且安全地部署程式碼到主機群。您可以建立自己的結構，為您的架構的各種元素建立模型，或與他人共享，或將它們發佈到社群。如需詳細資訊，請參閱 [AWS Cloud Development Kit 開發人員指南](#)。

遵循本教學課程並建立此範例可能會向您收取費用AWS帳戶。其中包括 Amazon EC2 和 Amazon SNS 和 Amazon SQS 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)、[Amazon SNS 定價](#)及 [Amazon SQS 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：清理](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將安裝環境中的所有工具AWS CDK需要運行一個寫在TypeScript編程語言。

1. [Node Version Manager](#) 或 **nvm** (用於稍後安裝 Node.js)。
2. [Node.js](#)，這是示例所要求的，並包含節點包管理器，或 **npm**，您用來安裝 TypeScript 和 AWS CDK 後來。
3. [TypeScript](#)，這是本示例所需的。(AWS CDK 還提供多種其他程式設計語言的支援。)
4. [AWS CDK](#)。

步驟 1.1：安裝 Node Version Manager (nvm)

1. 在 AWS Cloud9 IDE 終端機工作階段，確認最新安全性更新和錯誤修正皆已安裝。若要這麼做，請執行 **yum update** 命令 (適用於 Amazon Linux) 或 **apt update** 命令 (適用於 Ubuntu Server)。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server：

```
sudo apt update
```

2. 確認是否已安裝 **nvm**。若要這樣做，請執行 **nvm** 命令與 **--version** 選項。

```
nvm --version
```

如果成功，輸出內容會包含 **nvm** 版本編號，而您可以直接跳到 [步驟 1.2：安裝 Node.js](#)。

3. 下載並安裝 **nvm**。若要執行此操作，請執行安裝指令碼。這個範例已安裝 v0.33.0，但您可以[在此](#)確認最新版本的 **nvm**。

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

4. 開始使用 **nvm**。您可以先關閉終端機工作階段，然後再重新啟動，或是援引 `~/.bashrc` 檔案，其中包含載入 **nvm** 的命令。

```
. ~/.bashrc
```

步驟 1.2 : 安裝 Node.js

1. 確認您是否已安裝 Node.js，若已安裝，請確認安裝的版本為 16.17.0 或更高版本。此範例已經用 Node.js 16.17.0 測試過。若要確認終端機工作階段仍在 IDE 中開啟，請以 **--version** 選項執行 **node** 命令。

```
node --version
```

若已安裝 Node.js，輸出包含版本編號。如果版本編號為 v16.17.0，請直接跳到 [步驟 1.3 : 安裝 TypeScript](#)。

2. 透過執行下列指令來安裝 Node.js 16 **nvm** 使用指令 **install** 動作。

Note

您也可以執行 **nvm install node** 來安裝 Node.js 的 LTS (長期支援) 版本。AWS Cloud9 支援會追蹤 Node.js 的 LTS 版本。

```
nvm install v16
```

3. 開始使用。Node.js 若要執行此操作，請執行 **nvm** 命令與 **alias** 動作、對應別名的版本編號，以及用於該別名的版本，如下所示。

```
nvm alias default 16
```

Note

上述命令會將 Node.js 16 設定為 Node.js 的預設版本。或者，您可以執行 **nvm** 命令與 **use** 動作，而不是 **alias** 動作 (例如，**nvm use 16.17.0**)。不過，**use** 動作會使該 Node.js 版本僅在目前終端工作階段執行時執行。

4. 若要確認您使用的是 Node.js 16，請執行 **node --version** 再次命令。如果安裝了正確的版本，則輸出將包含版本 v16。

步驟 1.3：安裝 TypeScript

1. 確認您是否已經擁有 TypeScript 已安裝。要做到這一點，在終端會話仍然在 IDE 中打開的情況下，運行命令行 TypeScript 編譯器與 **--version** 選項。

```
tsc --version
```

如果你有 TypeScript 已安裝，輸出包含 TypeScript 版本號碼。如果 TypeScript 已安裝，請跳至 [步驟 1.4：安裝 AWS CDK](#)。

2. 安裝 TypeScript。要做到這一點，運行 **npm** 使用指令 **install** 行動，**-g** 選項，以及 TypeScript 包裝。這會安裝 TypeScript 作為環境中的全局包。

```
npm install -g typescript
```

3. 確認 TypeScript 已安裝。要做到這一點，運行命令行 TypeScript 編譯器與 **--version** 選項。

```
tsc --version
```

如果 TypeScript 已安裝，輸出包含 TypeScript 版本號碼。

步驟 1.4：安裝 AWS CDK

1. 確認您是否已安裝 AWS CDK。若要執行此操作，請於終端機工作階段在 IDE 中仍開啟的狀態，使用 **--version** 選項執行 **cdk** 命令。

```
cdk --version
```

若已安裝 AWS CDK，輸出包含 AWS CDK 版本和組建編號。直接跳到 [步驟 2：新增程式碼](#)。

2. 安裝 AWS CDK 的方法是搭配 **npm** 動作、要安裝的 **install** 套件名稱，以及在環境全域安裝套件的 AWS CDK 選項來執行 **-g** 命令。

```
npm install -g aws-cdk
```

3. 確認 AWS CDK 已安裝，且已正確地參考。若要這樣做，請執行 **cdk** 命令與 **--version** 選項。

```
cdk --version
```


如果成功，AWS CDK 版本和組建編號會顯示。

步驟 2：新增程式碼

在此步驟中，您將建立範例TypeScript包含您需要的所有源代碼的項目AWS CDK以編程方式部署AWS CloudFormation堆疊。這個堆疊會在您的 AWS 帳戶中建立 Amazon SNS 主題和 Amazon SQS 佇列，然後訂閱主題的佇列。

1. 在 IDE 仍開啟終端機工作階段的情況下，建立要存放專案原始碼的目錄，例如環境中的 `~/environment/hello-cdk` 目錄。然後，切換到該目錄。

```
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. 將目錄設置為TypeScript語言專案AWS CDK。若要執行此操作，請以 **init** 動作、**sample-app** 範本與 **--language** 選項搭配程式設計語言的名稱來執行 **cdk** 命令。

```
cdk init sample-app --language typescript
```

此在目錄中建立下列檔案和子目錄。

- 隱藏的 `.git` 子目錄和隱藏的 `.gitignore` 檔案，這讓專案相容於來源控制工具如 Git。
- 包含 `hello-cdk-stack.ts` 檔案的 `lib` 子目錄。這個檔案包含 AWS CDK 堆疊的程式碼。此程式碼描述於此程序的後續步驟。
- 包含 `hello-cdk.ts` 檔案的 `bin` 子目錄。這個檔案包含 AWS CDK 應用程式的進入點。
- `node_modules` 子目錄，其中包含支援的程式碼套件，其可供應用程式和堆疊視需來使用。
- 隱藏的 `.npmignore` 檔案，其中列出當建置程式碼時 **npm** 不需要的子目錄和檔案類型。
- `cdk.json` 檔案，其中包含可讓 **cdk** 命令更輕鬆地執行的資訊。
- `package-lock.json` 檔案，其中包含 **npm** 可用於降低可能的建置和執行錯誤的相關資訊。
- `package.json` 檔案，其中包含可讓 **npm** 命令更輕鬆地執行，且減少建置和執行錯誤的相關資訊。
- `README.md` 檔案，其中列出可以和 **npm** 及 AWS CDK 搭配執行的實用命令。
- `tsconfig.json` 檔案，其中包含可讓 **tsc** 命令更輕鬆地執行，且減少建置和執行錯誤的相關資訊。

3. 在 Environment (環境) 視窗中，開啟 lib/hello-cdk-stack.ts 檔案，並瀏覽該檔案中的以下程式碼。

```
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 類別，各自代表 AWS CloudFormation 堆疊及其屬性、可執行程式、Amazon SQS 佇列和 Amazon SNS 主題。
 - HelloCdkStack 類別代表此應用程式的 AWS CloudFormation 堆疊。此堆疊包含適用於此應用程式的新 Amazon SQS 佇列和 Amazon SNS 主題。
4. 在 Environment (環境) 視窗中，開啟 bin/hello-cdk.ts 檔案，並瀏覽該檔案中的以下程式碼。

```
#!/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();
```

此程式碼會載入 lib/hello-cdk-stack.ts 檔案中的 HelloCdkStack 類別、建立其執行個體，然後執行它。

5. 使用 **npm** 來執行TypeScript編譯器檢查編碼錯誤，然後啟用AWS CDK執行該項目的bin/hello-cdk.js文件。若要執行此操作，請從專案的根目錄以 **run** 動作執行 **npm** 命令，且需將package.json 檔案中的 **build** 命令值指定如下。

```
npm run build
```

上述命令會執行TypeScript編譯器，這增加了支持bin/hello-cdk.d.ts和lib/hello-cdk-stack.d.ts文件。這個編譯器也會將hello-cdk.ts 和 hello-cdk-stack.ts 檔案轉換為hello-cdk.js 和 hello-cdk-stack.js 檔案。

步驟 3：執行程式碼

在此步驟中，您指示 AWS CDK 在 bin/hello-cdk.js 檔案中根據程式碼建立 AWS CloudFormation 堆疊範本。然後，指示 AWS CDK 部署堆疊，這會建立 Amazon SNS 主題和 Amazon SQS 佇列，然後訂閱該主題的佇列。然後，確認已成功部署主題和佇列，做法是從主題傳送訊息到佇列。

1. 讓 AWS CDK 建立 AWS CloudFormation 堆疊範本。若要執行此操作，請在 IDE 仍開啟終端機工作階段的情況下，從專案的根目錄，以 **synth** 動作和堆疊的名稱執行 **cdk** 命令。

```
cdk synth HelloCdkStack
```

如果成功，輸出會顯示 AWS CloudFormation 堆疊範本的 Resources 部分。

2. 您第一次將 AWS CDK 應用程式部署至特定 AWS 帳戶和 AWS 區域組合的環境時，必須安裝引導堆疊。這個堆疊包含 AWS CDK 完成其各種操作所需的各種資源。例如，這個堆疊包含一個 Amazon S3 儲存貯體，AWS CDK 會在其部署過程中使用此儲存貯體來存放範本和資產。若要安裝引導堆疊，請執行 **cdk** 命令，搭配 **bootstrap** 動作。

```
cdk bootstrap
```

Note

如果您執行 `cdk bootstrap` 時不指定任何選項，則系統會使用預設的 AWS 帳戶和 AWS 區域。您也可以指定設定檔和帳戶/區域組合來引導特定環境。例如：

```
cdk bootstrap --profile test 123456789012/us-east-1
```

3. 讓 AWS CDK 執行 AWS CloudFormation 堆疊範本以部署堆疊。若要執行此操作，請自專案的根目錄執行 **cdk** 命令與 **deploy** 動作，以及堆疊的名稱。

```
cdk deploy HelloCdkStack
```

如果成功，輸出會顯示 HelloCdkStack 堆疊已部署成功，沒有出現錯誤。

Note

如果輸出訊息顯示堆疊未定義環境，而且無法從標準位置取得 AWS 憑證，或未設定任何區域，請確定您的 AWS 憑證已在 IDE 中正確設定，然後再次執行 **cdk deploy** 命令。如需詳細資訊，請參閱[從 AWS Cloud9 的環境呼叫 AWS 服務](#)。

4. 要確認已成功部署 Amazon SNS 主題和 Amazon SQS 佇列，請將訊息傳送到主題，然後檢查收到訊息的佇列。若要執行此操作，您可以使用 AWS Command Line Interface (AWS CLI) 或 AWS CloudShell 之類的工具。如需這些工具的詳細資訊，請參閱 [AWS Command Line Interface 和 aws-shell 教程 AWS Cloud9](#)。

例如，如要在 IDE 仍開啟終端機工作階段的情況下將訊息傳送到主題，請使用 AWS CLI 執行 Amazon SNS **publish** 命令，其中包括訊息的主旨和本文、主題的 AWS 區域，以及該主題的 Amazon 資源名稱 (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
```

在前述命令中，將 `arn:aws:sns:us-east-2:123456789012:HelloCdkStack-HelloCdkTopic1A234567-8BCD9EFGHIJ0K` 取代為 AWS CloudFormation 指派給主題的 ARN。若要取得 ID，您可以執行 Amazon SNS **list-topics** 命令。

```
aws sns list-topics --output table --query 'Topics[*].TopicArn'
```

如果成功，**publish** 命令的輸出會顯示已發佈訊息的 MessageId 值。

若要檢查已接收訊息的佇列，請執行 Amazon SQS **receive-message** 命令，其中包括佇列的 URL。

```
aws sqs receive-message --queue-url https://queue.amazonaws.com/123456789012/HelloCdkStack-HelloCdkQueue1A234567-8BCD9EFGHIJ0K
```

在前述命令中，將 `https://queue.amazonaws.com/123456789012/HelloCdkStack-HelloCdkQueue1A234567-8BCD9EFGHIJ0K` 取代為 AWS CloudFormation 指派給佇列的 ARN。若要取得 URL，您可以執行 Amazon SQS **list-queues** 命令。

```
aws sqs list-queues --output table --query 'QueueUrls[*]'
```

如果成功，**receive-message** 命令的輸出會顯示已接收訊息的相關資訊。

步驟 4：清理

為避免在您結束使用此範例後仍對您的 AWS 帳戶持續收費，您應刪除 AWS CloudFormation 堆疊。這麼做會刪除 Amazon SNS 主題和 Amazon SQS 佇列。您也應該刪除環境。

步驟 4.1：刪除堆疊

在 IDE 仍開啟終端機工作階段的情況下，請以 **destroy** 動作和堆疊的名稱從專案的根目錄執行 **cdk** 命令。

```
cdk destroy HelloCdkStack
```

當系統提示您刪除堆疊時，請輸入 `y`，然後按下 Enter。

如果成功，輸出會顯示 HelloCdkStack 堆疊已刪除，沒有出現錯誤。

步驟 4.2：刪除環境

若要刪除環境，請參閱[刪除 AWS Cloud9 中的環境](#)。

燈泡教程 AWS Cloud9

本教學課程可讓您在 AWS Cloud9 開發環境 Linux 中設定和執行 LAMP (MySQL、Apache HTTP 伺服器和 PHP)。

遵循本教學課程並建立此範例可能會導致您的AWS帳戶。其中包括 Amazon Elastic Compute Cloud (Amazon EC2) 等 AWS 服務的可能費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝工具](#)
- [步驟 2：設定 MySQL](#)
- [步驟 3：設定網站](#)
- [步驟 4：清理](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱 [在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱 [在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝工具

在此步驟中，您將安裝下列工具：

- Apache HTTP Server，這是 Web 伺服器主機。
- PHP，這是一種指令碼語言，尤其適用於 Web 開發並可內嵌到 HTML。
- MySQL，這是一種關聯式資料庫管理系統。

然後，先啟動 Apache HTTP Server 再啟動 MySQL，來完成此步驟。

1. 確保已在執行個體上安裝最新安全性更新和錯誤修正。若要這樣做，請在 AWS Cloud9 IDE 的終端機工作階段中，執行 **yum update** (若為 Amazon Linux) 或 **apt update** (若為 Ubuntu Server) 命令。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。

針對 Amazon Linux :

```
sudo yum -y update
```

針對 Ubuntu Server :

```
sudo apt -y update
```

2. 檢查是否已安裝 Apache HTTP Server。若要這麼做，請執行 **httpd -v** (若為 Amazon Linux) 或 **apache2 -v** (若為 Ubuntu Server) 命令。

如果成功，輸出會包含 Apache HTTP Server 版本號碼。

如果您看到錯誤，則執行 **install** 命令來安裝 Apache HTTP Server。

針對 Amazon Linux :

```
sudo yum install -y httpd24
```

針對 Ubuntu Server :

```
sudo apt install -y apache2
```

3. 執行 **php -v** 命令，確認是否已安裝 PHP。

如果成功，輸出會包含 PHP 版本號碼。

如果您看到錯誤，則執行 **install** 命令來安裝 PHP。

針對 Amazon Linux :

```
sudo yum install -y php56
```

針對 Ubuntu Server :

```
sudo apt install -y php libapache2-mod-php php-xml
```

4. 執行 **mysql --version** 命令，確認是否已安裝 MySQL。

如果成功，輸出會包含 MySQL 版本編號。

如果您看到錯誤，則執行 **install** 命令來安裝 MySQL。

針對 Amazon Linux：

```
sudo yum install -y mysql-server
```

針對 Ubuntu Server：

```
sudo apt install -y mysql-server
```

5. 在您安裝 Apache HTTP Server、PHP 和 MySQL 之後，請啟動 Apache HTTP Server，然後執行以下命令來確認它已啟動。

若為 Amazon Linux (您可能需要執行此命令兩次)：

```
sudo service httpd start && sudo service httpd status
```

若為 Ubuntu Server (返回命令提示字元，按下 q)：

```
sudo service apache2 start && sudo service apache2 status
```

6. 啟動 MySQL，然後執行以下命令來確認它已啟動。

針對 Amazon Linux：

```
sudo service mysqld start && sudo service mysqld status
```

若為 Ubuntu Server (返回命令提示字元，按下 q)：

```
sudo service mysql start && sudo service mysql status
```

步驟 2：設定 MySQL

在此步驟中，您設定 MySQL 以遵循 MySQL 安全最佳實務。這些安全最佳實務包括為根帳戶設定密碼，以及移除可從本機主機外部存取的根帳戶。要注意的其他最佳實務是移除匿名使用者、移除測試資料庫，以及移除允許任何人存取名稱開頭為 `test_` 之資料庫的權限。

然後，練習啟動與結束 MySQL 命令列用戶端，即可完成此步驟。

1. 在 AWS Cloud9 IDE 的終端機工作階段中執行以下命令，針對 MySQL 安裝實作 MySQL 安全最佳實務。

```
sudo mysql_secure_installation
```

2. 出現提示時，回答下列指定的問題。

針對 Amazon Linux：

1. 輸入根目錄目前使用的密碼 (若無密碼，請按 Enter) – 按 Enter (若沒有密碼)。
2. 設定根密碼 - 輸入 Y，然後按 Enter。
3. 新密碼 - 輸入密碼，然後按 Enter。
4. 重新輸入新密碼 - 重新輸入密碼，然後按 Enter。(請務必將密碼存放在安全位置，以供日後使用。)
5. 移除匿名使用者 - 輸入 Y，然後按 Enter。
6. 不允許從遠端進行根登入 - 輸入 Y，然後按 Enter。
7. 移除測試資料庫和其存取權 - 輸入 Y，然後按 Enter。
8. 立即重新載入權限表格 - 輸入 Y，然後按 Enter。

針對 Ubuntu Server：

1. 是否要設定 VALIDATE PASSWORD 外掛程式 - 輸入 y，然後按 Enter。
 2. 有三種層級的密碼驗證政策 - 輸入 0、1 或 2，然後按 Enter。
 3. 新密碼 - 輸入密碼，然後按 Enter。
 4. 重新輸入新密碼 - 重新輸入密碼，然後按 Enter。請務必將密碼存放在安全位置，以供日後使用。
 5. 是否要繼續使用已提供的密碼 - 輸入 y，然後按 Enter。
 6. 移除匿名使用者 - 輸入 y，然後按 Enter。
 7. 不允許從遠端進行根登入 - 輸入 y，然後按 Enter。
 8. 移除測試資料庫和其存取權 - 輸入 y，然後按 Enter。
 9. 立即重新載入權限表格 - 輸入 y，然後按 Enter。
3. 若要直接與 MySQL 互動，請執行以下命令，以根使用者身分啟動 MySQL 命令列用戶端。出現提示時，輸入您先前設定的根使用者密碼，然後按 Enter。當您使用 MySQL 命令列用戶端時，提示會變更為 `mysql>`。

```
sudo mysql -uroot -p
```

- 若要結束 MySQL 命令列用戶端，請執行以下命令。提示會變回 \$。

```
exit;
```

步驟 3：設定網站

在此步驟中，您將使用建議的擁有者和存取許可來設定 Apache HTTP Server 的預設網站根目錄。然後，在預設網站根目錄內建立 PHP 型網頁。

接著，啟用傳入的 Web 流量以檢視該網頁，方法是在 Amazon EC2 中設定安全群組，並在與此 EC2 環境相關聯的 Amazon Virtual Private Cloud (Amazon VPC) 中設定網路存取控制清單 (網路 ACL)。每個 EC2 環境必須同時與 Amazon EC2 中的安全群組以及 Amazon VPC 中的網路 ACL 相關聯。不過，即使 AWS 帳戶中的預設網路 ACL 允許環境的所有傳入和傳出流量，預設安全群組仍僅允許透過連接埠 22 使用 SSH 的傳入流量。如需詳細資訊，請參閱[the section called “Amazon VPC 設定”](#)。

接著，從 AWS Cloud9 IDE 外部成功檢視網頁，您就能完成此步驟。

- 使用建議的擁有者和存取許可來設定 Apache HTTP Server 的預設網站根目錄 (/var/www/html)。若要這樣做，在 AWS Cloud9 IDE 的終端機工作階段中依序執行以下六個命令。若要了解每個命令的作用，請閱讀每個命令的 # 字元後面的資訊。

針對 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.
```

針對 Ubuntu Server :

```
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. 執行以下命令，在 Apache HTTP Server 的預設網站根資料夾 (即 /var/www/html) 中建立名稱為 index.php 的 PHP 型網頁。

針對 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
```

上述的 Amazon Linux 命令也會將檔案的擁有者變更為 ec2-user、將檔案的群組變更為 web-content，以及為使用者將檔案的許可變更為讀取/寫入，和為群組及其他對象變更為讀取/執行。

針對 Ubuntu Server :

```
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 的上述命令也會將檔案的擁有者變更為 ubuntu、將檔案的群組變更為 web-content，以及若為使用者，將檔案的許可變更為讀取/寫入，若為群組及其他，則變更為讀取/執行。

如果成功，上述命令會以下列內容建立 index.php 檔案。

```
<?php
  phpinfo();
?>
```

3. 啟用透過連接埠 80 傳入的 Web 流量以檢視新網頁，方法是在 Amazon VPC 中設定網路 ACL，以及在與此 EC2 環境相關聯的 Amazon EC2 設定安全群組。若要這樣做，請依照下列順序執行以下八個命令，一次一個。若要了解每個命令的作用，請閱讀每個命令的 # 字元後面的資訊。

Important

執行以下命令後，所有 EC2 環境以及與此環境的安全群組和網路 ACL 相關聯的 Amazon EC2 執行個體都能接受透過連接埠 80 傳入的 Web 流量。這可能意外導致 EC2 環境以及此執行個體以外的 Amazon EC2 執行個體能夠接受透過連接埠 80 傳入的 Web 流量。

Note

以下第二個到第四個命令可讓安全群組允許透過連接埠 80 傳入的 web 流量。如果您有一個預設安全群組，僅允許透過連接埠 22 傳入的 SSH 流量，則必須先執行第一個命令，接著執行第二個到第四個命令。不過，如果您有一個自訂安全群組，已允許透過連接埠 80 傳入的 Web 流量，則可以略過執行那些命令。

以下第五個到第八個命令可讓網路 ACL 允許透過連接埠 80 傳入的 web 流量。如果您有一個預設網路 ACL，已允許透過所有連接埠傳入的流量，則可以安全地略過執行那些命令。不過，假設您有自訂網路 ACL，不允許透過連接埠 80 傳入的 Web 流量。然後，執行第一個命令，接著執行第五到第八個命令。

```
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. 取得 Web 伺服器根目錄內 index.php 檔案的 URL。若要這樣做，請執行以下命令，並使用新的 Web 瀏覽器標籤或 AWS Cloud9 IDE 以外的其他 Web 瀏覽器，前往顯示的 URL。如果成功，網頁會顯示 Apache HTTP Server、MySQL、PHP 和其他相關設定的相關資訊。

```
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.
```

步驟 4：清理

假設您想要繼續使用此環境，但想要停用透過連接埠 80 傳入的 Web 流量。然後，依照下列順序逐一執行以下八個命令，以刪除您先前在安全群組和網路 ACL 中設定，並與環境相關聯的相應傳入流量規則。若要了解每個命令的作用，請閱讀每個命令的 # 字元後面的資訊。

Important

執行以下命令後，所有 EC2 環境以及與此環境的安全群組和網路 ACL 相關聯的 Amazon EC2 執行個體都會拒絕透過連接埠 80 傳入的 Web 流量。這可能意外導致 EC2 環境以及此執行個體以外的 Amazon EC2 執行個體拒絕透過連接埠 80 傳入的 Web 流量。

Note

以下第五個到第八個命令會移除現有規則，以封鎖網路 ACL 不允許透過連接埠 80 傳入的 web 流量。如果您有一個預設網路 ACL，已允許透過所有連接埠傳入的流量，則可以略過執行那些命令。不過，假設您有一個自訂網路 ACL，其現有規則允許透過連接埠 80 傳入的 Web 流量，而且您想要刪除那些規則。然後，您需要執行第一個命令，接著執行第五到第八個命令。

```
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[].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.
```

如果您已完成使用此環境，請將環境刪除，以免您的 AWS 帳戶持續產生費用。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

WordPress 教學課程 AWS Cloud9

本教程使您可以安裝和運行 WordPress 在一個 AWS Cloud9 開發環境。WordPress 是一個廣泛用於交付 Web 內容的開源內容管理系統 (CMS)。

Note

遵循本教學課程並建立此範例可能會向您收取費用 AWS 帳戶。包括 Amazon Elastic Compute Cloud (Amazon EC2) 等服務的可能費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。
- 你有一個 up-to-date EC2 執行個體，包含所有最新軟體套件。在 AWS Cloud9 IDE 終端機視窗，您可以採用 `-y` 選項執行 `yum update` 來安裝更新，不需要要求確認。如果您要先檢查更新再安裝，則可以省略此選項。

```
sudo yum update -y
```

安裝概觀

安裝 WordPress 環境的 EC2 執行個體包含以下步驟：

1. 安裝和配置 MariaDB 服務器，這是一個開放源代碼的關聯式數據庫，用於存儲信息 WordPress 安裝
2. 安裝和配置 WordPress，其中包括編輯 `wordpress.conf` 配置文件
3. 配置託管 WordPress 現場
4. 預覽 WordPress 由 Apache 伺服器託管的網頁內容

步驟 1：安裝與設定 MariaDB 伺服器

1. 在 AWS Cloud9 IDE 中，選擇 Window (視窗)、New Terminal (新增終端機)，並輸入以下命令來安裝並啟動 MariaDB 伺服器安裝：

```
sudo yum install -y mariadb-server
sudo systemctl start mariadb
```

2. 然後，執行 `mysql_secure_installation` 指令碼來加強 MariaDB 伺服器安裝的安全性。

向指令碼提供回應時，請對第一個問題按下 Enter，以便讓根密碼保持空白。對 Set root password? 按下 n，然後對其餘每個安全選項都按下 y。


```
mysql_secure_installation
```

3. 現在創建一個數據庫表來存儲WordPress使用 MariaDB 客戶端的信息。

(系統要求您輸入密碼時按下 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. 若要登出 MariaDB 用戶端，請執行 `exit` 命令。

步驟 2：安裝和配置WordPress

1. 在 IDE 終端機視窗中，導覽到 `environment` 目錄，然後建立目錄 `config` 和 `wordpress`。然後執行 `touch` 命令，在 `config` 目錄中建立檔案並命名為 `wordpress.conf`：

```
cd /home/ec2-user/environment
mkdir config wordpress
touch config/wordpress.conf
```

2. 使用 IDE 編輯器或 `vim` 進行更新`wordpress.conf`包含允許 Apache 伺服器提供服務的主機組態資訊WordPress內容：

```
# Ensure that Apache listens on port 80
Listen 8080
<VirtualHost *:8080>
    DocumentRoot "/var/www/wordpress"
    ServerName www.example.org
    # Other directives here
</VirtualHost>
```

3. 現在運行以下命令來檢索所需的存檔文件並安裝WordPress:

```
cd /home/ec2-user/environment
wget https://wordpress.org/latest.tar.gz
tar xvf latest.tar.gz
```

4. 執行 `touch` 命令，在 `environment/wordpress` 目錄中建立檔案並命名為 `wp-config.php`：

```
touch wordpress/wp-config.php
```

5. 使用 IDE 編輯器或 vim 進行更新wp-config.php並用您的設置替換示例數據：

```
// ** 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';
```

步驟 3：設定 Apache HTTP 伺服器

1. 在 AWS Cloud9 IDE 終端機視窗中，確認您已安裝 Apache：

```
httpd -v
```

如果未安裝 Apache 伺服器，請執行下列命令：

```
sudo yum install -y httpd
```

2. 導覽至 /etc/httpd/conf.d 目錄，也就是 Apache 虛擬主機組態檔案的位置。然後使用 ln 命令將您之前建立的 wordpress.conf 連結到目前的工作目錄 (/etc/httpd/conf.d)：

```
cd /etc/httpd/conf.d
sudo ln -s /home/ec2-user/environment/config/wordpress.conf
```

3. 接著導覽至 `/var/www` 目錄，也就是 Apache 伺服器的預設根資料夾。然後使用 `ln` 命令將您之前建立的 `wordpress` 目錄連結到目前的工作目錄 (`/var/www`)：

```
cd /var/www
sudo ln -s /home/ec2-user/environment/wordpress
```

4. 執行 `chmod` 命令，允許 Apache 伺服器執行 `wordpress` 子目錄中的內容：

```
sudo chmod +x /home/ec2-user/
```

5. 接著重新啟動 Apache 伺服器使它偵測新組態：

```
sudo service httpd restart
```

步驟 4：預覽WordPress網頁內容

1. 使用 AWS Cloud9 IDE 在以下目錄建立新檔案並命名為 `index.html`：`environment/wordpress`。
2. 將 HTML 格式的文字新增至 `index.html`。例如：

```
<h1>Hello World!</h1>
```

3. 在環境視窗中，選擇 `index.html` 檔案，然後選擇預覽，預覽運行的應用。

應用程式預覽標籤中會出現 web 頁面，顯示 Hello World！訊息。若要在偏好的瀏覽器中檢視 Web 內容，請選擇 Pop Out Into a New Window (在新視窗中彈出)。

如果您刪除 `index.html` 文件並刷新應用程式預覽選項卡，WordPress 顯示配置頁面。

管理混合內容錯誤

Web 瀏覽器顯示的混合內容錯誤 WordPress 網站，如果它正在同一時間加載 HTTPS 和 HTTP 腳本或內容。錯誤訊息的用詞視您使用的 Web 瀏覽器而定，但系統會通知您與網站的連線不安全，或安全性不完整。且您的 Web 瀏覽器會封鎖混合內容的存取權。

⚠ Important

根據預設，您在 AWS Cloud9 IDE 應用程式預覽標籤中存取的所有 web 頁面會自動使用 HTTPS 通訊協定。如果頁面的 URI 具有不安全的 http 通訊協定，就會自動由 https 替代。而且您無法藉由手動將 https 改回 http 來存取不安全的內容。如需為您的網站實作 HTTPS 的指引，請參閱[WordPress 文件](#)。

爪哇教學 AWS Cloud9

⚠ Important

如果您使用的是 AWS Cloud9 開發環境，該環境由具有 2 GiB 或更多記憶體 of EC2 執行個體提供支援，則我們建議您啟用增強的 Java 支援。這可以提供各種生產力功能的存取權，例如程式碼完成、錯誤檢查、內容特定動作以及除錯選項 (如中斷點和逐步執行)。如需詳細資訊，請參閱[增強對 Java 開發的支援](#)。

本教程使您能夠在 AWS Cloud9 開發環境中運行一些 Java 代碼。

遵循本教學課程並建立此範例可能會對您的 AWS 帳戶產生費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#) 和 [Amazon S3 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：建置並執行程式碼](#)
- [步驟 4：進行使用 AWS SDK for Java 的設定](#)
- [步驟 5：在環境中設定 AWS 憑證管理](#)
- [步驟 6：新增 AWS SDK 程式碼](#)
- [步驟 7：建置並執行 AWS SDK 程式碼](#)
- [步驟 8：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您會在 AWS Cloud9 開發環境中安裝一組 Java 開發工具。如果您已在環境中安裝一組 Java 開發工具，例如 Oracle JDK 或 OpenJDK，您可以直接跳到[步驟 2：新增程式碼](#)。此範例以 OpenJDK 8 開發，完成下列程序後您就可將此範例安裝在您的環境中。

1. 確認是否已安裝 OpenJDK 8。若要執行此操作，請在 AWS Cloud9 IDE 的終端機工作階段中，使用 **-version** 選項執行 Java runner 的命令列版本。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。

```
java -version
```

依據上述命令的輸出，執行以下操作之一：

- 如果找不到 java 命令的輸出狀態，請繼續此程序中的步驟 2 以安裝 OpenJDK 8。
 - 如果輸出包含以 Java(TM)、Java Runtime Environment、Java SE、J2SE 或 Java2 起始的值，表示沒有安裝 OpenJDK 或未設定為預設的 Java 開發工具組。繼續此程序中的步驟 2 以安裝 OpenJDK 8，然後切換為使用 OpenJDK 8。
 - 如果輸出包含以 java version 1.8 和 OpenJDK 開頭的值，請直接跳到[步驟 2：新增程式碼](#)。已為此範例正確安裝 OpenJDK 8。
 - 如果輸出包含小於 java version 的 1.8 以及以 OpenJDK 起始的值，請繼續使用此程序中的安裝步驟 2，將已安裝的 OpenJDK 版本升級到 OpenJDK 8。
2. 確定已安裝最新的安全性更新和錯誤修正。若要這樣做，請以 **update** 命令執行 yum 工具 (適用於 Amazon Linux) 或 apt 工具 (適用於 Ubuntu Server)。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server :

```
sudo apt update
```

3. 安裝 OpenJDK 8。若要這樣做，請以 **install** 命令執行 yum 工具 (適用於 Amazon Linux) 或 apt 工具 (適用於 Ubuntu Server)，並指定 OpenJDK 8 套件。

針對 Amazon Linux :

```
sudo yum -y install java-1.8.0-openjdk-devel
```

針對 Ubuntu Server :

```
sudo apt install -y openjdk-8-jdk
```

如需詳細資訊，請參閱 OpenJDK 網站上的[如何下載和安裝預先建置的 OpenJDK 套件](#)。

4. 將預設的 Java 開發工具組切換或升級為 OpenJDK 8。若要這樣做，請執行 **update-alternatives** 命令與 **--config** 選項。執行此命令兩次，以切換或升級 Java runner 和編譯器的命令列版本。

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

在每個提示中，輸入 OpenJDK 8 的選擇號碼 (包含 java-1.8)。

5. 確認 Java runner 和編譯器命令列版本使用的是 OpenJDK 8。若要執行此操作，執行 Java runner 和編譯器的命令列版本並使用 **-version** 選項。

```
java -version
javac -version
```

如果 OpenJDK 8 已安裝並正確設定，Java runner 版本輸出將包含以 `openjdk version 1.8` 起始的值，而且 Java 編譯器版本輸出以 `javac 1.8` 起始的值。

步驟 2：新增程式碼

在 AWS Cloud9 IDE 中，建立含有以下程式碼的檔案，然後以 `hello.java` 名稱儲存檔案。(若要建立檔案，請在選單列上選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)。)

```
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));  
    }  
}
```

步驟 3：建置並執行程式碼

1. 使用 Java 編譯器的命令列版本，將 `hello.java` 檔案編譯為 `hello.class` 檔案。若要執行此操作，請在 AWS Cloud9 IDE 中使用終端機，從與 `hello.java` 檔案相同的目錄執行 Java 編譯器，並指定 `hello.java` 檔案。

```
javac hello.java
```

2. 使用 Java runner 的命令列版本執行 `hello.class` 檔案。若要這樣做，從與 `hello.class` 檔案相同的目錄執行 Java runner，並指定在 `hello` 檔案中宣告的 `hello.java` 類別的名稱，新增兩個整數 (例如，5 和 9)。

```
java hello 5 9
```

3. 比較您的輸出。

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```

步驟 4：進行使用 AWS SDK for Java 的設定

您可以進一步強化本範例，使用 AWS SDK for Java 建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

在此步驟中，您會在環境中安裝 [Apache Maven](#) 或 [Gradle](#)。Maven 和 Gradle 是通用建置自動化系統，可搭配使用 Java 專案。在安裝 Maven 或 Gradle 之後，您可以使用它來產生新的 Java 專案。在這個新專案中，您會新增 AWS SDK for Java 的參考。此 AWS SDK for Java 提供便利的方法，讓您從 Java 程式碼與 Amazon S3 等 AWS 服務互動。

主題

- [使用 Maven 設定](#)
- [使用 Gradle 設定](#)

使用 Maven 設定

1. 在您的環境中安裝 Maven。若要查看是否已安裝 Maven，請在 AWS Cloud9 IDE 中使用終端機，以 **-version** 選項執行 Maven。

```
mvn -version
```

如果成功，輸出會包含 Maven 版本編號。如果已安裝 Maven，請直接跳到此程序中的步驟 4，使用 Maven 在環境中產生新的 Java 專案。

2. 使用終端機執行以下命令以安裝 Maven。

如為 Amazon Linux，您可以使用下列命令獲得存放 Maven 的套件儲存庫相關資訊，然後使用此資訊安裝 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
```

如需有關上述命令的詳細資訊，請參閱 Fedora Project Wiki 網站上的 [Extra Packages for Enterprise Linux \(EPEL\)](#)。

對於 Ubuntu Server，改為執行下列命令。


```
sudo apt install -y maven
```

3. 執行 Maven 並使用 **-version** 選項以確認安裝。

```
mvn -version
```

4. 使用 Maven 產生新的 Java 專案。若要這樣做，請使用終端機從您希望 Maven 產生專案的目錄執行下列命令 (例如環境的根目錄)。

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

上述命令會在環境中建立以下專案目錄結構。

```
my-app
|- src
|   |- main
|       |- java
|           |- com
|               |- mycompany
|                   |- app
|                       |-App.java
|- test
|   |- java
|       |- com
|           |- mycompany
|               |- app
|                   |- AppTest.java
|- pom.xml
```

如需有關上述目錄結構的詳細資訊，請參閱 Apache Maven Project 網站上的 [Maven 快速入門 Archetype](#) 和 [認識標準目錄配置](#)。

5. 修改專案的專案物件模型 (POM) 檔案。(POM 檔案定義 Maven 專案的設定。) 若要執行此操作，請從 Environment (環境) 視窗開啟 my-app/pom.xml 檔案。在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 pom.xml 檔案。

```
<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 檔案包含指定宣告的專案設定，例如：

- `artifactId` 的 `my-app` 設定項目可設定專案的根目錄名稱，`group-id` 的 `com.mycompany.app` 設定項目可設定 `com/mycompany/app` 子目錄結構以及 `package` 和 `App.java` 檔案中的 `AppTest.java` 宣告。
- `artifactId` 的 `my-app` 設定項目與 `packaging` 的 `jar` 設定項目、`version` 的 `1.0-SNAPSHOT` 設定項目，以及 `descriptorRef` 的 `jar-with-dependencies` 設定項目，可設定 `my-app-1.0-SNAPSHOT-jar-with-dependencies.jar` 的輸出 JAR 檔案名稱。
- `plugin` 區段宣告將建置單一 JAR，其中包含所有相依性。
- `dependency` 區段附帶 `com.amazon.aws` 的 `groupId` 設定和 `aws-java-sdk` 的 `artifactId` 設定，包含 AWS SDK for Java 程式庫檔案。要使用的 AWS SDK for Java 版本是以 `version` 設定宣告。若要使用不同的版本，請取代此版本編號。

直接跳到 [步驟 5：在環境中設定 AWS 憑證管理](#)。

使用 Gradle 設定

1. 在您的環境中安裝 Gradle。若要查看是否已安裝 Gradle，請在 AWS Cloud9 IDE 中使用終端機，以 **-version** 選項執行 Gradle。

```
gradle -version
```

如果成功，輸出會包含 Gradle 版本號碼。如果已安裝 Gradle，請直接跳到此程序中的步驟 4，使用 Gradle 在環境中產生新的 Java 專案。

2. 使用終端機執行以下命令以安裝 Gradle。這些命令安裝和執行 SDKMAN! 工具，然後使用 SDKMAN! 安裝最新版本的 Gradle。

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk install gradle
```

如需有關前述命令的詳細資訊，請參閱 SDKMAN! 網站的[安裝](#)，以及 Gradle 網站的[使用套件軟體管理工具進行安裝](#)。

3. 執行 Gradle 並使用 **-version** 選項以確認安裝。

```
gradle -version
```

4. 使用 Gradle 在環境中產生新的 Java 專案。若要這樣做，請使用終端機執行以下命令來建立專案目錄，然後切換到該目錄。

```
mkdir my-app
cd my-app
```

5. 執行以下命令，讓 Gradle 在環境的 my-app 目錄中產生新的 Java 應用程式專案。

```
gradle init --type java-application
```

上述命令會在環境中建立以下專案目錄結構。

```
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. 修改專案的 AppTest.java。(如果您不執行此操作，專案可能不會如預期建置或執行。)若要執行此操作，請從 Environment (環境) 視窗開啟 my-app/src/test/java/AppTest.java 檔案。在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 AppTest.java 檔案。

```
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. 修改專案的 `build.gradle` 檔案。(build.gradle 檔案定義 Gradle 專案的設定。) 若要執行此操作，請從 Environment (環境) 視窗開啟 `my-app/build.gradle` 檔案。在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 `build.gradle` 檔案。

```
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` 檔案包含指定宣告的專案設定，例如：

- `io.spring.dependency-management` 外掛程式用於匯入 AWS SDK for Java Maven 物料清單 (BOM) 以管理專案的 AWS SDK for Java 依存項目。`classpath` 宣告要使用的版本。若要使用不同的版本，請取代此版本編號。
- `com.amazonaws:aws-java-sdk-s3` 包含 AWS SDK for Java 程式庫檔案的 Amazon S3 部分。`mavenBom` 會宣告要使用的版本。如果您要使用不同的版本，請取代此版本編號。

步驟 5：在環境中設定 AWS 憑證管理

每次使用 AWS SDK for Java 呼叫 AWS 服務時，您都必須隨著呼叫提供一組 AWS 登入資料。這些登入資料將判斷 AWS SDK for Java 是否具備適當許可，能夠發出該次呼叫。如果登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需其他資訊，請參閱 AWS SDK for Java 開發人員指南中的 [設定開發用的 AWS 憑證和區域](#)。

步驟 6：新增 AWS SDK 程式碼

在此步驟中，您會新增程式碼以與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

從 Environment (環境) 視窗，開啟 `my-app/src/main/java/com/mycompany/app/App.java` 檔案 (適用於 Maven) 或開啟 `my-app/src/main/java/App.java` 檔案 (適用於 Gradle)。在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 `App.java` 檔案。

```
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.getMessage());
            }
        }
    }
}
```

```
    }  
  }  
  
  // 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.getMessage());  
  }  
  
  // 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());  
  }  
}  
  
}
```

步驟 7：建置並執行 AWS SDK 程式碼

若要執行上個步驟的程式碼，請從終端機執行以下命令。這些命令將使用 Maven 或 Gradle 建立專案的可執行 JAR 檔案，然後使用 Java runner 執行 JAR。此 JAR 在執行時會使用要在 Amazon S3 中建立的儲存貯體名稱 (例如 my-test-bucket) 以及要在其中將儲存貯體建立為輸入的 AWS 區域 ID (例如 us-east-2)。

如果是 Maven，請執行下列命令。

```
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
```

如果是 Gradle，請執行下列命令。

```
gradle build
gradle run -PappArgs=["my-test-bucket', 'us-east-2']"
```

將您的結果與下列輸出比較。

```
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:
```

步驟 8：清除

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

C ++ 教程 AWS Cloud9

本教程使您能夠在 AWS Cloud9 開發環境中運行 C ++ 代碼。程式碼也會使用 [AWS SDK for C++](#) 提供的資源，這是一種模組化的跨平台開源程式庫，您可以使用它來連線到 Amazon Web Services。

遵循本教學課程並建立此範例可能會對您的 AWS 帳戶產生費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#) 和 [Amazon S3 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝 g ++ 和所需的開發套件](#)
- [步驟 2：安裝 CMake](#)

- [步驟 3：取得並建置適用於 C++ 的 SDK](#)
- [第 4 步：創建 C++ 和 C MakeLists 文件](#)
- [步驟 5：建置並執行 C++ 程式碼](#)
- [步驟 6：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝 g++ 和所需的開發套件

要建置和執行 C++ 應用程式，你需要 g++ 這類應用程式，它是由 [GNU 編譯器集合 \(GCC\)](#) 所提供的 C++ 編譯器。

您還需要新增 libcurl、libopenssl、libuuid、zlib 的標題檔案 (-dev 套件)，以及支援 Amazon Polly 的 libpulse (選用)。

視您使用的是 Amazon Linux/Amazon Linux 2 執行個體或 Ubuntu 執行個體，安裝開發工具的過程會略有不同。

Amazon Linux-based systems

您可以檢查否已經安裝 gcc，方法是在 AWS Cloud9 終端機執行下列命令：

```
g++ --version
```

如果尚未安裝 g++，您可以輕鬆地安裝套件群組中名為「開發工具」的部分。您可以使用 yum groupinstall 命令將這些工具新增至執行個體：

```
sudo yum groupinstall "Development Tools"
```

再次執行 `g++ --version` 確認編譯器已安裝。

接著使用系統的套件管理員安裝所需程式庫的套件：

```
sudo yum install libcurl-devel openssl-devel libuuid-devel pulseaudio-libs-devel
```

Ubuntu-based systems

您可以檢查否已經安裝 `gcc`，方法是在 AWS Cloud9 終端機執行下列命令：

```
g++ --version
```

如果沒有安裝 `gcc`，您可以執行以下命令，藉此安裝在以 Ubuntu 為基礎的系統上：

```
sudo apt update
sudo apt install build-essential
sudo apt-get install manpages-dev
```

再次執行 `g++ --version` 確認編譯器已安裝。

接著使用系統的套件管理員安裝所需程式庫的套件：

```
sudo apt-get install libcurl4-openssl-dev libssl-dev uuid-dev zlib1g-dev libpulse-dev
```

步驟 2：安裝 CMake

您需要安裝 `cmake` 工具，它可以將從來源碼建置可執行檔的過程自動化。

1. 在 IDE 終端機視窗中，執行下列命令來取得必要的封存檔：

```
wget https://cmake.org/files/v3.18/cmake-3.18.0.tar.gz
```

2. 從封存檔中解壓縮檔案，並導覽至包含解壓縮檔案的目錄：

```
tar xzf cmake-3.18.0.tar.gz
cd cmake-3.18.0
```

3. 接下來，執行 Bootstrap 指令碼並執行下列命令來安裝 `cmake`：

```
./bootstrap  
make  
sudo make install
```

4. 執行下列命令來確認您已安裝此工具：

```
cmake --version
```

步驟 3：取得並建置適用於 C++ 的 SDK

若要設定 C++ 的 AWS SDK，您可以直接從來源自行建置 SDK，或使用套件管理員下載程式庫。您可以在 AWS SDK for C++ 開發人員指南中的 [C++ AWS SDK 使用入門](#) 此找到可用選項的詳細資訊。

此範例將示範如何使用 git 來複製 SDK 來源碼，以及使用 cmake 來建置 C++ 的 SDK。

1. 在終端機中執行下列命令，即可為 AWS Cloud9 環境複製遠端儲存庫並以遞迴方式取得所有 Git 子模組來：

```
git clone --recurse-submodules https://github.com/aws/aws-sdk-cpp
```

2. 導覽至新的 aws-sdk-cpp 目錄，在要為 C++ 建置 AWS SDK 的位置建立子目錄，然後導覽到該子目錄：

```
cd aws-sdk-cpp  
mkdir sdk_build  
cd sdk_build
```

- 3.

Note

為了節省時間，此步驟僅會建置 AWS SDK for C++ 的 Amazon S3 部分。如果您想要建置完整的 SDK，請刪減 cmake 命令中的 `-DBUILD_ONLY=s3`。

為 C++ 建置完整 SDK 可能需要一個多小時才會完成，具體時間視您的 Amazon EC2 執行個體或自有伺服器中的可用運算資源而定。

執行下列命令，藉此使用 cmake 在 sdk_build 目錄中建置 C++ SDK 的 Amazon S3 部分：

```
cmake .. -DBUILD_ONLY=s3
```

4. 接著請執行 `make install` 命令，以便使建置的 SDK 可供存取：

```
sudo make install
cd ..
```

第 4 步：創建 C++ 和 C MakeLists 文件

在此步驟中，您會建立 C++ 檔案，藉此允許專案的使用者與 Amazon S3 儲存貯體互動。

您也可以建立 `CMakeLists.txt` 檔案，來提供 `cmake` 建置 C++ 程式庫時使用的指示。

1. 在 AWS Cloud9 IDE 中，建立含有以下內容的檔案，然後儲存檔案到您環境的根目錄 (/) 並命名為 `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. 再另外建立含有以下內容的檔案，然後儲存檔案到您環境的根目錄 (/) 並命名為 CMakeLists.txt。此檔案讓您能夠將程式碼建置成可執行檔。

```
# 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})
```

步驟 5：建置並執行 C++ 程式碼

1. 在您儲存 `s3-demo.cpp` 和 `CMakeLists.txt` 的環境根目錄中，執行 `cmake` 來建置您的專案：

```
cmake .
make
```

2. 您至此已可以從命令列執行程式。在下列命令中，將 `my-unique-bucket-name` 替換為 Amazon S3 儲存貯體的唯一名稱，如有必要，請將 `us-east-1` 更換為您要建立儲存貯體的其他 AWS 區域的識別碼。

```
./s3-demo my-unique-bucket-name us-east-1
```

如果程式成功執行，系統會傳回類似如下的輸出內容：

```
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.
```

步驟 6：清除

為避免在您結束此範例後您的 AWS 帳戶持續產生費用，請刪除環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

AWS Cloud9 的 Python 教學課程

本教學課程示範如何在 AWS Cloud9 開發環境中執行 Python 程式碼。

遵循此教學可能會向您的 AWS 帳戶收取費用。包括 Amazon Elastic Compute Cloud (Amazon EC2) 和 Amazon Simple Storage Service (Amazon S3) 等服務的可能費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#) 和 [Amazon S3 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝 Python](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：安裝並設定 AWS SDK for Python \(Boto3\)](#)
- [步驟 5：新增 AWS SDK 程式碼](#)
- [步驟 6：執行 AWS SDK 程式碼](#)
- [步驟 7：清除](#)

先決條件

在您使用此教學前，請務必符合下列要求。

- 您已有 AWS Cloud9 EC2 開發環境

此教學假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如需詳細資訊，請參閱 [建立 EC2 環境](#)。

如果您有不同類型的環境或作業系統，您可能需要調整此教學的操作指示。

- 您已開啟該環境的 AWS Cloud9 IDE

當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱 [在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝 Python

1. 於 AWS Cloud9 IDE 的終端機工作階段中執行 **python --version** 命令，確認是否已安裝 Python。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。) 若 Python 已安裝，請跳至 [步驟 2：新增程式碼](#)。
2. 執行 **yum update** (適用於 Amazon Linux) 或 **apt update** (適用於 Ubuntu Server) 命令，協助確保已安裝最新安全性更新和錯誤修正。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server：

```
sudo apt update
```

3. 執行 **install** 命令以安裝 Python。

針對 Amazon Linux：

```
sudo yum -y install python3
```

針對 Ubuntu Server：

```
sudo apt-get install python3
```

步驟 2：新增程式碼

在 AWS Cloud9 IDE 中，建立含有以下內容的檔案，然後儲存檔案並命名為 `hello.py`。(若要建立檔案，請在選單列上選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)。)

```
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))
```

步驟 3：執行程式碼

1. 在 AWS Cloud9 IDE 的選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Stopped ([新增] - 已停止) 索引標籤中，為 Command (命令) 輸入 `hello.py 5 9`。此程式碼的 5 代表 `sys.argv[1]`，而 9 代表 `sys.argv[2]`。
3. 選擇 Run (執行)，然後比較您的輸出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```

4. 預設情況下，AWS Cloud9 會自動為您的程式碼選取執行器。若要變更執行器，請選擇 Runner (執行器)，然後選擇 Python 2 或 Python 3。

Note

您可以為特定版本的 Python 建立自訂的執行器。如需詳細資訊，請參閱 [建立建置器或執行器](#)。

步驟 4：安裝並設定 AWS SDK for Python (Boto3)

AWS SDK for Python (Boto3) 可讓您使用 Python 程式碼與 AWS 服務 (例如 Amazon S3) 進行互動。例如，您可以使用 SDK 來建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

安裝 pip

在 AWS Cloud9 IDE 中執行 `python -m pip --version` 命令，確認已經為作用中的 Python 版本安裝 pip。如果已安裝 pip，請跳到下一節。

若要安裝 pip，請執行以下命令。由於 `sudo` 與使用者的環境不同，如果它與目前別名版本不同，則必須指定要使用的 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.
```

如需詳細資訊，請參閱 pip 網站文章[安裝](#)。

安裝 AWS SDK for Python (Boto3)

在您安裝 pip 後，請執行 `pip install` 命令來安裝 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.
```

如需詳細資訊，請參閱 [《》](#) 快速入門 AWS SDK for Python (Boto3) 的「安裝」一節。

在環境中設定憑證

每次使用 AWS SDK for Python (Boto3) 呼叫 AWS 服務時，您都必須隨著呼叫提供一組登入資料。這些登入資料會判斷 SDK 是否具備適當許可，能夠發出該次呼叫。如果登入資料未涵蓋必要的許可，呼叫會失敗。

若要將憑證存放在環境中，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 [《》](#) [章節](#) 登入資料 AWS SDK for Python (Boto3)。

步驟 5：新增 AWS SDK 程式碼

新增使用 Amazon S3 來建立儲存貯體的程式碼、列出可用的儲存貯體，並選擇性刪除您剛建立的儲存貯體。

在 AWS Cloud9 IDE 中，建立含有以下內容的檔案，然後儲存檔案並命名為 `s3.py`。

```
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()
```

步驟 6：執行 AWS SDK 程式碼

1. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 對於 Command (命令)，輸入 `s3.py my-test-bucket us-west-2`，其中 `my-test-bucket` 是要建立的儲存貯體名稱，而 `us-west-2` 是建立儲存貯體的 AWS 區域 ID。預設情況下，指令碼結束之前，會先刪除您的儲存貯體。若要保留您的儲存貯體，請將 `--keep_bucket` 新增到您的命令中。如需 AWS 區域 ID 的清單，請參閱 [AWS 一般參考](#)。

Note

不只是在您的 AWS 帳戶中，Amazon S3 儲存貯體名稱在整個 AWS 中皆不可重複。

3. 選擇 Run (執行)，然後比較您的輸出。

```
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
```

步驟 7：清除

為避免在您結束使用此教學課程後您的 AWS 帳戶仍持續產生費用，請刪除 AWS Cloud9。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

.NET 教學課程AWS Cloud9

本教程使您可以運行一些 .NET 代碼AWS Cloud9開發環境。

遵循本教學課程並建立此範例可能會向您收取費用AWS帳戶。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝必要工具](#)

- [步驟 2 \(選用\)：為 Lambda 函數安裝 .NET CLI 擴充功能](#)
- [步驟 3：建立 .NET 主控台應用程式專案](#)
- [步驟 4：新增程式碼](#)
- [步驟 5：建置並執行程式碼](#)
- [步驟 6：建立並設定 .NET 主控台應用程式專案，該專案使用AWS SDK for .NET](#)
- [步驟 7：新增 AWS SDK 程式碼](#)
- [步驟 8：建置並執行 AWS SDK 程式碼](#)
- [步驟 9：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將 .NET SDK 安裝到您的環境中，這是執行此範例所需的。

1. 確認您的環境中是否已安裝最新版本的 .NET SDK。若要執行這個操作，請在 AWS Cloud9 IDE 的終端機工作階段中，以 `--version` 選項執行 .NET Core 命令列界面 (CLI)。

```
dotnet --version
```

如果顯示的 .NET 命令列工具版本是 2.0 或更高版本，請直接跳到 [步驟 3：建立 .NET 主控台應用程式專案](#)。如果版本小於 2.0，或者如果錯誤 `bash: dotnet: command not found` 隨即顯示，請繼續以安裝 .NET SDK。

2. 對於亞馬遜 Linux，在終端會話中 AWS Cloud9 IDE 中，執行下列命令，以協助確保已安裝最新的安全性更新和錯誤修正，並安裝 `libunwind`.NET SDK 所需要的套件。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機))。

```
sudo yum -y update
sudo yum -y install libunwind
```

如使用 Ubuntu Server，請在 AWS Cloud9 IDE 的終端機工作階段中，執行下列命令來協助確保已安裝最新安全性更新和錯誤修正。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機))。

```
sudo apt -y update
```

- 執行下列命令，將 .NET SDK 安裝程式指令碼下載到您的環境中。

```
wget https://dot.net/v1/dotnet-install.sh
```

- 執行以下命令，讓目前使用者可執行安裝程式指令碼。

```
sudo chmod u=rx dotnet-install.sh
```

- 執行下列命令，執行安裝程式指令碼，該指令碼會下載並安裝 .NET SDK。

```
./dotnet-install.sh -c Current
```

- 將 .NET SDK 添加到您的 PATH。若要這樣做，請在環境的 shell 描述檔 (例如，.bashrc 檔案) 中，將 \$HOME/.dotnet 子目錄新增到環境的 PATH 變數，如下所示。

- 使用 .bashrcvi 命令，開啟 檔案以供編輯。

```
vi ~/.bashrc
```

- 如為 Amazon Linux，請使用向下鍵或 j 鍵，移到開頭為 export PATH 的那一行。

對於 Ubuntu Server，輸入 G，移到檔案的最後一行。

- 使用向右鍵或 \$ 鍵，移到行的結尾。
- 按下 i 鍵，切換到插入模式 (-- INSERT --- 將出現在顯示幕末端)。
- 如為 Amazon Linux，輸入 :\$HOME/.dotnet，以便將 \$HOME/.dotnet 子目錄加入 PATH 變數。請務必包含冒號字元 (:)。該行現在應該看起來類似於以下內容。

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet
```

對於 Ubuntu Server，按向右鍵，然後按 Enter 兩次，接著在檔案結尾單獨輸入以下這一行。

```
export PATH=$HOME/.dotnet:$PATH
```

- f. 儲存檔案。若要這麼做，請按 Esc 鍵 (-- INSERT --- 會從顯示器尾端消失)、輸入 :wq (以便寫入然後結束檔案)，然後按 Enter。

7. 載入 .NET SDK，方法是採購 .bashrc 文件。

```
. ~/.bashrc
```

8. 透過執行 .NET CLI，確認已載入 .NET SDK --help 選項。

```
dotnet --help
```

如果成功，則會顯示 .NET SDK 版本號碼，以及其他使用資訊。

9. 如果您不想再在環境中保留 .NET SDK 安裝程式指令碼，您可以按照以下步驟將其刪除。

```
rm dotnet-install.sh
```

步驟 2 (選用)：為 Lambda 函數安裝 .NET CLI 擴充功能

雖然本教學課程不需要，但您可以進行部署 AWS Lambda 功能和 AWS Serverless Application Model 使用 .NET CLI 的應用程式，如果您同時安裝 Amazon.Lambda.Tools 包裝。

1. 若要安裝此套件，請執行下列命令：

```
dotnet tool install -g Amazon.Lambda.Tools
```

2. 接著設定 PATH 和 DOTNET_ROOT 環境變數，以便指向已安裝的 Lambda 工具。在 .bashrc 檔案中找到 export PATH 區塊並加以編輯，以便使它顯示為類似如下內容 (如需編輯此檔案的詳細資訊，請參閱步驟 1)：

```
export PATH=$PATH:$HOME/.local/bin:$HOME/bin:$HOME/.dotnet:$HOME/.dotnet/tools
export DOTNET_ROOT=$HOME/.dotnet
```

步驟 3：建立 .NET 主控台應用程式專案

在此步驟中，您可以使用 .NET 建立名為的專案hello。該項目包含 .NET 需要從 IDE 中的終端運行簡單應用程式的所有文件。應用程式的程式碼是以 C# 編寫。

建立 .NET 主控台應用程式專案。若要執行這項操作，請使用 **new** 命令，指定主控台應用程式專案範本類型和要使用的程式設計語言 (在此範例中為 C#)。

-n 選項表示專案輸出到新的目錄 hello 中。接著導覽至該目錄。

```
dotnet new console -lang C# -n hello
cd hello
```

上述命令會新增名為 obj 的子目錄與數個檔案，以及一些其他的獨立檔案到 hello 目錄。您應該會注意到以下兩個重要檔案：

- hello/hello.csproj 檔案包含有關主控台應用程式專案的資訊。
- hello/Program.cs 檔案包含要執行的應用程式的程式碼。

步驟 4：新增程式碼

在此步驟中，您將新增一些程式碼到應用程式。

從 AWS Cloud9 IDE 中的 Environment (環境)視窗開啟 hello/Program.cs 檔案。

在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 Program.cs 檔案。

```
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);

}
}
}
```

步驟 5：建置並執行程式碼

在此步驟中，您將建立專案及其相依性到一組二進位檔案，包括可執行的應用程式檔案。接著執行應用程式。

1. 在 IDE 中，建立 .NET 的建置器，如下所示。
 - a. 從功能表列選擇 Run, Build System, New Build System (執行、建置系統、新建置系統)。
 - b. 在 My Builder.build (我的 Builder.build) 標籤上，使用以下程式碼取代標籤的內容。

```
{
  "cmd" : ["dotnet", "build"],
  "info" : "Building..."
}
```

- c. 選擇 File, Save As (檔案、另存新檔)。
 - d. 在 Filename (檔案名稱) 輸入 .NET.build。
 - e. 為 Folder (資料夾) 輸入 /.c9/builders。
 - f. 選擇 儲存。
2. 隨著的內容 Program.cs 文件顯示在編輯器中，選擇執行、建置系統、.NET。接著選擇 Run, Build (執行、建置)。

此建置器會新增名為 bin 的子目錄並將名為 Debug 的子目錄新增到 hello/obj 子目錄中。請注意以下三個重要檔案。

- hello/bin/Debug/netcoreapp3.1/hello.dll 檔案是可執行應用程式檔。
- hello/bin/Debug/netcoreapp3.1/hello.deps.json 檔案會列出應用程式的相依性。

- `hello/bin/Debug/netcoreapp3.1/hello.runtimeconfig.json` 檔案指定應用程式的共用執行時間及其版本。

Note

資料夾名稱，`netcoreapp3.1`，會反映此範例中使用的 .NET SDK 版本。視您安裝的版本而定，您可能會在資料夾名稱中看到不同的編號。

3. 建立 .NET 的流道，如下所示。
 - a. 在功能表列中，選擇 Run, Run With, New Runner (執行、執行工具、新增執行器)。
 - b. 在 My Runner.run (我的 Runner.run) 標籤上，使用以下程式碼取代標籤的內容。

```
{
  "cmd" : ["dotnet", "run", "$args"],
  "working_dir": "$file_path",
  "info" : "Running..."
}
```

- c. 選擇 File, Save As (檔案、另存新檔)。
 - d. 在 Filename (檔案名稱) 輸入 `.NET.run`。
 - e. 為 Folder (資料夾) 輸入 `/.c9/runners`。
 - f. 選擇 儲存。
4. 執行應用程式並搭配要新增的兩個整數 (例如，5 和 9)，如下所示。
 - a. 使用顯示在編輯器中的 `Program.cs` 檔案內容，選擇 Run, Run Configurations, New Run Configuration (執行、執行組態、新增執行組態)。
 - b. 在[新]-閒置」頁籤上，選擇亞軍：自動，然後選擇.NET。
 - c. 在 Command (命令) 方塊中，輸入 `hello 5 9`。
 - d. 選擇執行。

依預設，此執行程式會指示 .NET 執行 `hello.dll` 檔案中的 `hello/bin/Debug/netcoreapp3.1` 目錄。

將您的輸出與下列項目比較。

```
Hello, World!
```

```
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```

步驟 6：建立並設定 .NET 主控台應用程式專案，該專案使用 AWS SDK for .NET

您可以進一步強化本範例，使用 AWS SDK for .NET 建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

在這個新專案中，您會新增 AWS SDK for .NET 的參考。AWS SDK for .NET 提供便利的方式讓您從 .NET 程式碼與 Amazon S3 等 AWS 服務互動。您接著會在環境中設定 AWS 憑證管理。AWS SDK for .NET 需要這些登入資料，才能與 AWS 服務互動。

建立專案

1. 建立 .NET 主控台應用程式專案。若要執行這項操作，請使用 **new** 指令，指定主控台應用程式專案範本類型和要使用的程式設計語言。

-n 選項表示專案輸出到新的目錄 s3 中。接著導覽至該目錄。

```
dotnet new console -lang C# -n s3  
cd s3
```

2. 在 AWS SDK for .NET 中新增 Amazon S3 套件的專案參考。若要執行這項操作，請使用 **add package** 命令，指定亞馬遜 S3 套件的名稱 NuGet。(NuGet 定義如何建立、託管和使用 .NET 的套件，並為這些角色提供工具。)

```
dotnet add package AWSSDK.S3
```

當您將專案參考新增至 Amazon S3 套件時，NuGet 還將項目引用添加到其餘部分 AWS SDK for .NET。

Note

對於其他的名稱和版本 AWS 中的相關套件 NuGet，請參閱 [NuGet 使用 aws-sdk 標記的套件](#) 在 NuGet 網站。

設定 AWS 登入資料管理

每次使用 AWS SDK for .NET 呼叫 AWS 服務時，您都必須隨著呼叫提供一組 AWS 登入資料。這些登入資料將判斷 AWS SDK for .NET 是否具備適當許可，能夠發出該次呼叫。如果登入資料未涵蓋適當許可，呼叫即會失敗。

若要將憑證存放在環境中，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for .NET 開發人員指南中的 [設定 AWS 憑證](#)。

步驟 7：新增 AWS SDK 程式碼

在此步驟中，您會新增程式碼與 Amazon S3 互動，藉此建立儲存貯體、刪除您剛建立的儲存貯體，然後列出可用的儲存貯體。

從 AWS Cloud9 IDE 中的 Environment (環境)視窗開啟 s3/Program.cs 檔案。在編輯器中，使用以下程式碼取代該檔案的目前內容，然後儲存 Program.cs 檔案。

```
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);
    }
}
}
```

步驟 8：建置並執行 AWS SDK 程式碼

在此步驟中，您將建立專案及其相依性到一組二進位檔案，包括可執行的應用程式檔案。接著執行應用程式。

1. 建置專案。若要執行此操作，請使用顯示在編輯器中的 `s3/Program.cs` 檔案內容，在功能表列上選擇 **Run, Build** (執行、建置)。
2. 以要建立的 Amazon S3 儲存貯體名稱以及要在其中建立儲存貯體的 AWS 區域 ID (例如，`my-test-bucket` 和 `us-east-2`) 來執行應用程式，如下所示。
 - a. 使用仍顯示在編輯器中的 `s3/Program.cs` 檔案內容，選擇 **Run, Run Configurations, New Run Configuration** (執行、執行組態、新增執行組態)。
 - b. 在[新]-閒置」頁籤上，選擇亞軍：自動，然後選擇.NET。
 - c. 在 **Command (命令)** 方塊中，輸入應用程式的名稱、要建立的 Amazon S3 儲存貯體名稱，以及要在其中建立儲存貯體的 AWS 區域 ID (例如 `s3 my-test-bucket us-east-2`)。
 - d. 選擇執行。

依預設，此執行程式會指示 .NET 執行 `s3.dll` 檔案中的 `s3/bin/Debug/netcoreapp3.1` 目錄。

將您的結果與下列輸出比較。

```
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:
```

步驟 9：清除

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

Node.js 教學課程 AWS Cloud9

本教學課程可讓您在 AWS Cloud9 開發環境中執行一些 Node.js 指令碼。

遵循本教學課程並建立此範例可能會對您的 AWS 帳戶產生費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#) 和 [Amazon S3 定價](#)。

主題

- [必要條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：JavaScript 在 Node.js 中安裝和配置 AWS 軟體開發套件](#)
- [步驟 5：新增 AWS SDK 程式碼](#)
- [步驟 6：執行 AWS SDK 程式碼](#)
- [步驟 7：清除](#)

必要條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須擁有現有的 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱 [在 AWS Cloud9 中建立環境](#)。
- 您已開啟現有環境的 AWS Cloud9 IDE。當您開 AWS Cloud9 啟環境時，會在網頁瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱 [在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將安裝 Node.js，其為執行本範例的必要元件。

1. 在 AWS Cloud9 IDE 的終端機工作階段中，透過執行 `node --version` 命令確認是否已安裝 Node.js。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果成功，輸出會包含 Node.js 版本編號。若 Node.js 已安裝，請跳至 [步驟 2：新增程式碼](#)。
2. 執行 `yum update` (適用於 Amazon Linux) 或 `apt update` (適用於 Ubuntu Server) 命令，協助確保已安裝最新安全性更新和錯誤修正。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server：

```
sudo apt update
```

3. 若要安裝 Node.js，請先執行此命令以下載節點版本管理員 (nvm)。(nvm 是一個簡單的 Bash 外殼腳本，對於安裝和管理 Node.js 版本非常有用。如需詳細資訊，請參閱 GitHub 網站上的 [節點版本管理員](#)。)

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash
```

4. 若要開始使用 nvm，請先關閉終端機工作階段並再次啟動，或是援引 `~/.bashrc` 檔案，其中包含的命令將會載入 nvm。

```
. ~/.bashrc
```

5. 運行此命令在 Amazon Linux 2 上安裝 Node.js 16, Amazon Linux 1 和 Ubuntu 18.04. Amazon Linux 1 和 18.04 執行個體僅支援最高版本的 Node.js。

```
nvm install 16
```

運行此命令以在 Amazon 2023 和 Ubuntu 22.04 上安裝最新版本的 Node.js :

```
nvm install --lts && nvm alias default lts/*
```

Note

最新的 AL2023 AWS Cloud9 映像已經安裝了 Node.js 20，而最新的 Amazon 2 AWS Cloud9 映像已經安裝了 Node.js 18。如果您想在 Amazon Linux 2 上 AWS Cloud9 手動安裝 Node.js 18，請在 AWS Cloud9 IDE 終端機中運行以下命令：

```
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
```

步驟 2：新增程式碼

在 AWS Cloud9 IDE 中，建立包含此內容的檔案，並以名稱儲存檔案 `hello.js`。(若要建立檔案，請在選單列上選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)。)

```
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);

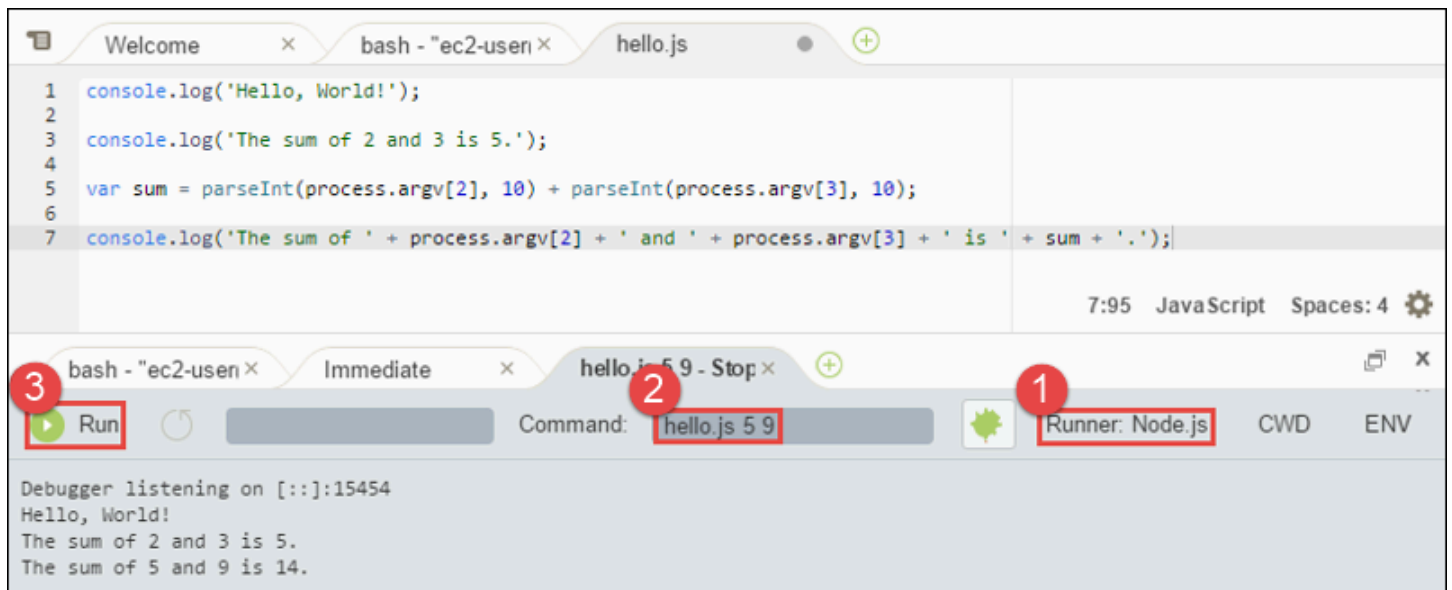
console.log('The sum of ' + process.argv[2] + ' and ' +
```

```
process.argv[3] + ' is ' + sum + '.');
```

步驟 3：執行程式碼

1. 在 AWS Cloud9 IDE 的功能表列上，選擇 [執行]、[執行組態]、[新增執行組態]。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Node.js。
3. 在 Command (命令) 中輸入 `hello.js 5 9`。在程式碼中，5 代表 `process.argv[2]`，而 9 代表 `process.argv[3]`。(`process.argv[0]` 代表名稱執行時間 (node)，而 `process.argv[1]` 代表檔案名稱 (hello.js))。
4. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
Hello, World!  
The sum of 2 and 3 is 5.  
The sum of 5 and 9 is 14.
```



步驟 4：JavaScript 在 Node.js 中安裝和配置 AWS 軟件開發套件

在中執行 Node.js 指令碼時 AWS Cloud9，您可以選擇 AWS 第 3 JavaScript 版 (V3) 的 AWS SDK 和 JavaScript 版本 2 (V2) 的舊版 SDK。與 V2 一樣，V3 可讓您輕鬆地使用 Amazon Web Services，但是已經寫入 TypeScript 並新增了幾個經常要求的功能，例如模組化套件。

AWS SDK for JavaScript (V3)

您可以增強此範例，以使用 Node.js JavaScript 中的 AWS 開發套件建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除剛建立的儲存貯體。

在此步驟中，您可以在 Node.js 中安裝和設定 AWS 開發套件的 Amazon S3 服務用戶端模組，這可讓您透過程式 JavaScript 碼與 Amazon S3 AWS 服務互動提供便利的方式。JavaScript

如果要使用其他 AWS 服務，則需要單獨安裝它們。如需有關安裝 AWS 模組的詳細資訊，請參閱[開AWS 發人員指南 \(V3\) 中的](#)。如需有關如何開始使用 JavaScript (V3) 的 Node.js 和 AWS SDK 的詳細資訊，請參閱[開 JavaScript發人員適用的AWS SDK 指南 \(V3\) 中的使用 Node.js](#) 入門。

JavaScript 在 Node.js 中安裝適用的 AWS SDK 之後，您必須在環境中設定認證管理。Node.js JavaScript 中的 AWS SDK 需要這些認證才能與 AWS 服務互動。

若要 JavaScript 在 Node.js 中安裝適用的 AWS 開發套件

使用 npm 執行 **install** 命令。

```
npm install @aws-sdk/client-s3
```

如需詳細資訊，請參閱[開AWS SDK for JavaScript 發人員指南 JavaScript中的安裝 SDK](#)。

在環境中設定憑證管理

每次您 JavaScript 在 Node.js 中使用的 AWS SDK 呼叫 AWS 服務時，都必須在呼叫中提供一組認證。這些認證會判斷 Node.js JavaScript 中的 AWS SDK 是否具有進行該呼叫的適當權限。若登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for JavaScript 開發人員指南中的[以 Node.js 設定憑證](#)。

AWS SDK for JavaScript (V2)

您可以增強此範例，以使用 Node.js JavaScript 中的 AWS 開發套件建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除剛建立的儲存貯體。

在此步驟中，您可以 JavaScript 在 Node.js 中安裝和設定的 AWS SDK，這可讓您透過程式 JavaScript碼與 Amazon S3 等 AWS 服務進行互動的便利方式。JavaScript 在 Node.js 中安裝適用

的 AWS SDK 之後，您必須在環境中設定認證管理。Node.js JavaScript 中的 AWS SDK 需要這些認證才能與 AWS 服務互動。

若要 JavaScript 在 Node.js 中安裝適用的 AWS 開發套件

使用 npm 執行 **install** 命令。

```
npm install aws-sdk
```

如需詳細資訊，請參閱[AWS SDK for JavaScript 開發人員指南 JavaScript 中的安裝 SDK](#)。

在環境中設定憑證管理

每次您 JavaScript 在 Node.js 中使用的 AWS SDK 呼叫 AWS 服務時，都必須在呼叫中提供一組認證。這些認證會判斷 Node.js JavaScript 中的 AWS SDK 是否具有進行該呼叫的適當權限。若登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for JavaScript 開發人員指南中的[以 Node.js 設定憑證](#)。

步驟 5：新增 AWS SDK 程式碼

AWS SDK for JavaScript (V3)

在此步驟中，您會再新增其他程式碼，這次是要與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。您稍後將執行此程式碼。

在 AWS Cloud9 IDE 中，建立包含此內容的檔案，並以名稱儲存檔案 `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)

在此步驟中，您會再新增其他程式碼，這次是要與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。您稍後將執行此程式碼。

在 AWS Cloud9 IDE 中，建立包含此內容的檔案，並以名稱儲存檔案 `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,
]);
```

步驟 6：執行 AWS SDK 程式碼

1. 使用 npm 執行 **install** 命令，讓程式碼能以非同步方式呼叫 Amazon S3 操作。

```
npm install async
```

2. 在 AWS Cloud9 IDE 的功能表列上，選擇 [執行]、[執行組態]、[新增執行組態]。

3. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Node.js。
4. 如果您正在使用 AWS SDK JavaScript (V3)，則用於命令類型s3.js。如果您使用 AWS SDK to Javascript (v2)，對於命令類型s3.js my-test-bucket us-east-2，其中my-test-bucket是您要創建然後刪除的存儲桶的名稱，並且us-east-2是您要在其中創建存儲桶的 AWS 區域的 ID。其他區域的 ID 請參閱 [《》 章節 Amazon Simple Storage Service \(Amazon S3\) Amazon Web Services 一般參考](#)。

Note

Amazon S3 儲存貯體名稱必須是唯一的，AWS而不僅僅是您的 AWS 帳戶。

5. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
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:
```

步驟 7：清除

若要避免 AWS 帳戶在使用完此範例後持續向您的帳戶收取費用，您應該刪除環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

對於 PHP 的教程 AWS Cloud9

本教程使您能夠在AWS Cloud9開發環境中運行一些 PHP 腳本。

遵循本教學課程並建立此範例可能會對您的AWS帳戶產生費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

主題

- [必要條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：安裝並設定 AWS SDK for PHP](#)
- [步驟 5：新增 AWS SDK 程式碼](#)
- [步驟 6：執行 AWS SDK 程式碼](#)
- [步驟 7：清除](#)

必要條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱 [在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱 [在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將安裝 PHP，這是執行此範例的必要工具。

Note

下列程序只會安裝 PHP。若要安裝 Apache Web 伺服器和 MySQL 資料庫這類相關工具，請參閱 Amazon EC2 的 Linux 執行個體使用者指南中的 [教學：在 Amazon Linux 上安裝 LAMP Web 伺服器](#)。

1. 於 AWS Cloud9 IDE 的終端機工作階段，執行 **php --version** 命令以確認是否已安裝 PHP。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果成功，輸出會包含 PHP 版本號碼。如果已安裝 PHP，請直接跳到 [步驟 2：新增程式碼](#)。

2. 執行 **yum update** (適用於 Amazon Linux) 或 **apt update** (適用於 Ubuntu Server) 命令，協助確保已安裝最新安全性更新和錯誤修正。

針對 Amazon Linux 2 和 Amazon Linux :

```
sudo yum -y update
```

針對 Ubuntu Server :

```
sudo apt update
```

3. 執行 **install** 命令，以安裝 PHP。

針對 Amazon Linux 2 :

```
sudo amazon-linux-extras install -y php7.2
```

針對 Amazon Linux :

```
sudo yum -y install php72
```

Note

您可以使用下列命令來檢視您的 Amazon Linux 版本。

```
cat /etc/system-release
```

針對 Ubuntu Server :

```
sudo apt install -y php php-xml
```

如需詳細資訊，請參閱 PHP 網站上的 [Installation and Configuration](#)。

步驟 2：新增程式碼

在 AWS Cloud9 IDE 中，建立含有此內容的檔案，然後以名稱 `hello.php` 儲存檔案。(若要建立檔案，請在選單列上，選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)，為 Filename (檔案名稱) 輸入 `hello.php`，然後選擇 Save (儲存)。

```
<?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

上述程式碼未依賴任何外部檔案。不過，如果您曾經在檔案中包含或需要其他 PHP 檔案，而且想要 AWS Cloud9 使用這些檔案在您輸入時完成程式碼，請在 Preferences (偏好設定) 中開啟 Project, PHP Support, Enable PHP code completion (專案、PHP 支援、啟用 PHP 程式碼完成) 設定，然後將這些檔案的路徑新增至 Project, PHP Support, PHP Completion Include Paths (專案、PHP 支援、PHP 完成時包括路徑) 設定。(若要檢視及變更偏好設定，請從選單列選擇 AWS Cloud9、Preferences (偏好設定))。

步驟 3：執行程式碼

1. 在 AWS Cloud9 IDE 的選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 PHP (cli)。
3. 在 Command (命令) 中輸入 `hello.php 5 9`。在程式碼中，5 代表 `$argv[1]`，而 9 代表 `$argv[2]`。(`$argv[0]` 代表檔案名稱 (`hello.php`)。)
4. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```

```

1 <?php
2 print('Hello, World!');
3
4 print("\nThe sum of 2 and 3 is 5.");
5
6 $sum = (int)$argv[1] + (int)$argv[2];
7
8 print("\nThe sum of $argv[1] and $argv[2] is $sum.");
9 ?>

```

Running PHP script /home/ec2-user/workspace/hello.php
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.

步驟 4：安裝並設定 AWS SDK for PHP

您可以進一步強化本範例，使用 AWS SDK for PHP 建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

在此步驟中，您會安裝並設定 AWS SDK for PHP，它提供的便捷方法可讓您透過 PHP 程式碼與 Amazon S3 等 AWS 服務互動。您應該先安裝 AWS SDK for PHP Composer，[才能安裝](#)。安裝 AWS SDK for PHP 之後，您必須在環境中設定憑證管理。AWS SDK for PHP 需要這些登入資料，才能與 AWS 服務互動。

安裝 Composer

搭配無訊息 () 執行 `curl -s` 命令，並顯示錯誤 (-S) 選項，以透過管線將 Composer 安裝程式放入 PHP 封存檔 (PHAR) (依慣例命名為 `composer.phar`)。

```
curl -sS https://getcomposer.org/installer | php
```

安裝 AWS SDK for PHP

對於 Ubuntu Server，安裝 Composer 安裝 AWS SDK for PHP 時所需的額外套件。

```
sudo apt install -y php-xml php-curl
```

如為 Amazon Linux 或 Ubuntu Server，請使用 `php` 命令執行 Composer 安裝程式來安裝 AWS SDK for PHP。


```
php composer.phar require aws/aws-sdk-php
```

此命令會在環境中建立數個資料夾和檔案。您會使用的主要檔案是 `autoload.php`，它位在環境的 `vendor` 資料夾中。

Note

安裝之後，Composer 可能建議您安裝額外的依存項目。您可以使用以下命令來執行此動作，並指定要安裝的依存項目清單。例如，以下命令會指示 Composer 安裝以下列出的依存項目。

```
php composer.phar require psr/log ext-curl doctrine/cache aws/aws-php-sns-message-validator
```

如需詳細資訊，請參閱 AWS SDK for PHP 開發人員指南中的[安裝](#)。

在環境中設定憑證管理

每次使用 AWS SDK for PHP 呼叫 AWS 服務時，您都必須隨著呼叫提供一組登入資料。這些登入資料將判斷 AWS SDK for PHP 是否具備適當許可，能夠發出該次呼叫。如果登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for PHP 開發人員指南中[基本用量](#)的「建立用戶端」一節。

步驟 5：新增 AWS SDK 程式碼

在此步驟中，您會再新增其他程式碼，這次是要與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。您稍後將執行此程式碼。

在 AWS Cloud9 IDE 中，建立含有此內容的檔案，然後以名稱 `s3.php` 儲存檔案。

```
<?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);

?>
```

步驟 6：執行 AWS SDK 程式碼

1. 在 AWS Cloud9 IDE 的選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 PHP (cli)。
3. 針對 Command (命令)，輸入 `s3.php America/Los_Angeles my-test-bucket us-east-2`，其中：

- `America/Los_Angeles` 是您的預設時區 ID。如需其他 ID，請參閱 PHP 網站上的「[支援的時區清單](#)」。
- `my-test-bucket` 是您想要建立然後刪除的儲存貯體名稱。

Note

不只是在您的 AWS 帳戶中，Amazon S3 儲存貯體名稱在整個 AWS 中皆不可重複。

- `us-east-2` 是您想要在其中建立儲存貯體之 AWS 區域的 ID。其他區域的 ID 請參閱 [《》 章節 Amazon Simple Storage Service \(Amazon S3\) Amazon Web Services 一般參考](#)。
4. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
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:
```

步驟 7：清除

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱[刪除 AWS Cloud9 中的環境](#)。

疑難排解 PHP 執行程式的問題 AWS Cloud9

如果您遇到 PHP CLI 執行程式的問題，您必須確定執行器已設定為 PHP，並且已啟用除錯程式模式。

紅寶石AWS Cloud9

如需有關AWS Cloud9搭配 Ruby 使用AWS SDK 的詳細資訊，請參閱 Ruby 開發人員指南中的「[AWS Cloud9搭配開發人員使用AWS SDK](#)」。AWS

Note

遵循此教學可能會向您的 AWS 帳戶收取費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

前往的教學課程 AWS Cloud9

本教程使您可以在AWS Cloud9開發環境中運行一些 Go 代碼。

遵循本教學課程並建立此範例可能會對您的AWS帳戶產生費用。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

主題

- [必要條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：安裝並設定 AWS SDK for Go](#)
- [步驟 5：新增 AWS SDK 程式碼](#)
- [步驟 6：執行 AWS SDK 程式碼](#)
- [步驟 7：清除](#)

必要條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。
- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將安裝和設定 Go，這是執行此範例的必要工具。

1. 在 AWS Cloud9 IDE 的終端機工作階段中，執行 **go version** 命令，確認是否已安裝 Go。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果成功，則輸出應該包含 Go 版本號碼。否則，應該會輸出錯誤訊息。若 Go 已安裝，請跳至[步驟 2：新增程式碼](#)。
2. 執行 **yum update** (適用於 Amazon Linux) 或 **apt update** (適用於 Ubuntu Server) 命令，協助確保已安裝最新安全性更新和錯誤修正。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server :

```
sudo apt update
```

- 若要安裝 Go，請逐一執行這些命令。

```
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.
```

上述命令採用撰寫本主題時最新穩定版本的 Go。如需詳細資訊，請參閱 [Go Programming Language](#) 網站上的 [Downloads](#)。

- 將 Go 二進位檔的路徑新增至 PATH 環境變數，如下所示。
 - 開啟您的 shell 描述檔檔案 (例如，`~/.bashrc`) 以進行編輯。
 - 在這行程式碼的結尾，輸入下列程式碼，讓程式碼現在如下所示。

```
PATH=$PATH:/usr/local/go/bin
```

- 儲存檔案。
- 取得 `~/.bashrc` 檔案，讓終端機現在可以找到您剛剛參考的 Go 二進位檔。

```
. ~/.bashrc
```

- 執行 `go version` 命令，以確認現在已成功安裝和設定 Go。如果成功，輸出會包含 Go 版本號碼。

步驟 2：新增程式碼

在 AWS Cloud9 IDE 中，建立含有此內容的檔案，然後以名稱 `hello.go` 儲存檔案。(若要建立檔案，請在選單列上選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)。)

```
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))  
}
```

步驟 3：執程式碼

1. 在 AWS Cloud9 IDE 的選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Go。

Note

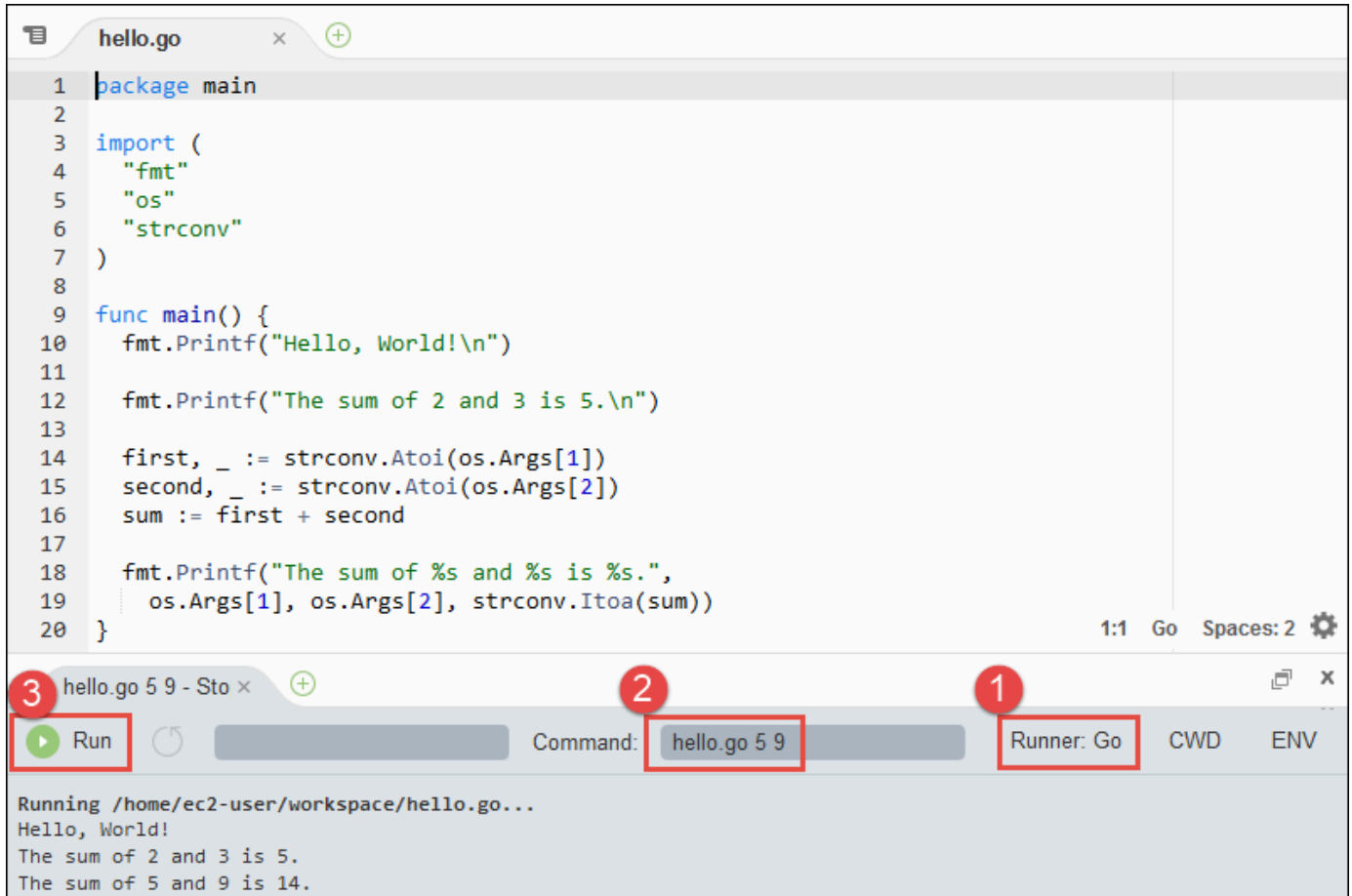
如果未提供 Go，您可以建立 Go 的自訂執行器。

1. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 New Runner (新執行器)。
2. 在 My Runner.run 標籤上，使用以下程式碼取代標籤的內容。

```
{  
    "cmd" : ["go", "run", "$file", "$args"],  
    "info" : "Running $project_path$file_name...",  
    "selector" : "source.go"  
}
```

3. 從選單列選擇 File (檔案)、Save As (另存新檔)，然後儲存檔案至 /.c9/runners 資料夾並命名為 Go.run。

4. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Go。
 5. 選擇 hello.go 標籤以使其作用。
3. 在 Command (命令) 中輸入 hello.go 5 9。此程式碼的 5 代表 os.Args[1]，而 9 代表 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. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```

步驟 4：安裝並設定 AWS SDK for Go

您可以進一步強化本範例，使用 AWS SDK for Go 建立 Amazon S3 儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。

在此步驟中，您會安裝並設定 AWS SDK for Go，它提供的便捷方法可讓您透過 Go 程式碼與 Amazon S3 等 AWS 服務互動。安裝 AWS SDK for Go 前，您必須設定 GOPATH 環境變數。安裝 AWS SDK for Go 並設定 GOPATH 環境變數之後，您必須在環境中設定憑證管理。AWS SDK for Go 需要這些登入資料，才能與 AWS 服務互動。

設定 GOPATH 環境變數

1. 開啟 `~/.bashrc` 檔案進行編輯。
2. 在檔案的最後一行後面，輸入此程式碼。

```
GOPATH=~/.environment/go  
  
export GOPATH
```

3. 儲存檔案。
4. 取得 `~/.bashrc` 檔案，讓終端機現在可以找到您剛剛參考的 GOPATH 環境變數。

```
. ~/.bashrc
```

5. 執行 `echo $GOPATH` 命令，以確認已成功設定 GOPATH 環境變數。如果成功，應該會輸出 `/home/ec2-user/environment/go` 或 `/home/ubuntu/environment/go`。

安裝 AWS SDK for Go

執行 `go get` 命令，並指定 AWS SDK for Go 來源所在位置。

```
go get -u github.com/aws/aws-sdk-go/...
```

Go 會將 AWS SDK for Go 來源安裝至 GOPATH 環境變數所指定的位置，即環境中的 `go` 資料夾。

在環境中設定憑證管理

每次使用 AWS SDK for Go 呼叫 AWS 服務時，您都必須隨著呼叫提供一組登入資料。這些登入資料將判斷 AWS SDK for Go 是否具備適當許可，能夠發出該次呼叫。如果登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for Go 開發人員指南中的 [指定憑證](#)。

步驟 5：新增 AWS SDK 程式碼

在此步驟中，您會再新增其他程式碼，這次是要與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。您稍後將執行此程式碼。

在 AWS Cloud9 IDE 中，建立含有此內容的檔案，然後以名稱 `s3.go` 儲存檔案。

```
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)
}
```

步驟 6：執行 AWS SDK 程式碼

1. 在 AWS Cloud9 IDE 的選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
2. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Go。
3. 針對 Command (命令) 輸入 `s3.go YOUR_BUCKET_NAME THE_AWS_REGION`，其中 `YOUR_BUCKET_NAME` 是您要建立再予以刪除的儲存貯體的名稱，而 `THE_AWS_REGION` 是該儲存貯體建立所在 AWS 區域的 ID。例如，若為美國東部 (俄亥俄) 區域，則使用 `us-east-2`。其他區域的 ID 請參閱 [《》 章節 Amazon Simple Storage Service \(Amazon S3\) Amazon Web Services 一般參考](#)。

Note

不只是在您的 AWS 帳戶中，Amazon S3 儲存貯體名稱在整個 AWS 中皆不可重複。

4. 選擇 Run (執行) 按鈕，然後對照您的輸出。

```
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:
```

步驟 7：清除

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

TypeScript教學課程AWS Cloud9

本教程向您展示如何使用TypeScript在一個AWS Cloud9開發環境。

遵循本教學課程並建立此範例可能會向您收取費用AWS帳戶。其中包括 Amazon EC2 和 Amazon S3 這類服務可能的費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)和 [Amazon S3 定價](#)。

主題

- [先決條件](#)
- [步驟 1：安裝必要工具](#)
- [步驟 2：新增程式碼](#)
- [步驟 3：執行程式碼](#)
- [步驟 4：安裝和配置AWS適用於的 SDKJavaScript在 Node.js 中](#)
- [步驟 5：新增 AWS SDK 程式碼](#)
- [步驟 6：執行 AWS SDK 程式碼](#)
- [步驟 7：清除](#)

先決條件

在您使用此範例前，請務必確認您的設定符合下列要求：

- 您必須已有 AWS Cloud9 EC2 開發環境。本範例假設您已具備 EC2 環境，且該環境已連線到執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。如果您有不同類型的環境或作業系

統，您可能需要依照此範例的說明來設定相關工具。如需詳細資訊，請參閱[在 AWS Cloud9 中建立環境](#)。

- 您現有環境的 AWS Cloud9 IDE 已開啟。當您開啟環境時，AWS Cloud9 會在 Web 瀏覽器中開啟該環境的 IDE。如需詳細資訊，請參閱[在 AWS Cloud9 中開啟環境](#)。

步驟 1：安裝必要工具

在此步驟中，您將安裝 TypeScript 使用節點數據包管理員 (npm)。若要安裝 npm，請使用節點套件管理工具 (nvm)。如果沒有 nvm，請先在此步驟中安裝它。

1. 在終端工作階段中 AWS Cloud9 IDE 中，確認是否 TypeScript 已經通過運行命令行安裝 TypeScript 編譯器與 `--version` 選項。(若要啟動新終端機工作階段，請在選單列上，選擇 Window (視窗)、New Terminal (新增終端機)。如果成功，輸出會包含 TypeScript 版本編號。如果 TypeScript 已安裝，請跳至[步驟 2：新增程式碼](#)。

```
tsc --version
```

2. 以 `--version` 選項執行 npm，確認是否已安裝 npm。如果成功，輸出會包含 npm 版本編號。如果 npm 已安裝，請跳至此程序中的步驟 10 以使用 npm 要安裝 TypeScript。

```
npm --version
```

3. 執行 `yum update` (適用於 Amazon Linux) 或 `apt update` (適用於 Ubuntu Server) 命令，協助確保已安裝最新安全性更新和錯誤修正。

針對 Amazon Linux：

```
sudo yum -y update
```

針對 Ubuntu Server：

```
sudo apt update
```

4. 若要安裝 npm，通過運行以下命令開始下載節點版本管理器 (nvm)。(nvm 是一個簡單的 Bash 外殼腳本，對於安裝和管理 Node.js 版本很有用。如需詳細資訊，請參閱[節點版本管理員](#)上的 GitHub 網站。)

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.0/install.sh | bash
```

- 若要開始使用 **nvm**，請先關閉終端機工作階段並再次啟動，或是援引 `~/.bashrc` 檔案，其中包含的命令將會載入 **nvm**。

```
. ~/.bashrc
```

- 以 **--version** 選項執行 **nvm**，確認已安裝 **nvm**。

```
nvm --version
```

- 通過運行來安裝最新版本 16 的 Node.js **nvm**。（**npm** 已包含在 Node.js 中。）

```
nvm install v16
```

- 執行 Node.js 命令列版本搭配 **--version** 選項，確認已安裝 Node.js。

```
node --version
```

- 以 **--version** 選項執行 **npm**，確認已安裝 **npm**。

```
npm --version
```

- 安裝 TypeScript 通過運行 **npm** 與 **-g** 選項。這會安裝 TypeScript 作為環境中的全局包。

```
npm install -g typescript
```

- 確認 TypeScript 通過運行命令行安裝 TypeScript 編譯器與 **--version** 選項。

```
tsc --version
```

步驟 2：新增程式碼

- 在 AWS Cloud9 IDE 中，建立名為 `hello.ts` 的檔案。（若要建立檔案，請在選單列上選擇 File (檔案)、New File (新增檔案)。若要儲存檔案，請選擇 File (檔案)、Save (儲存)。）
- 在 IDE 的終端機中，從與 `hello.ts` 檔案的相同目錄執行 **npm** 以安裝 `@types/node` 程式庫。

```
npm install @types/node
```

這樣會在與 `node_modules/@types/node` 檔案相同的目錄中新增 `hello.ts` 資料夾。這個新資料夾包含 Node.js 類型定義 TypeScript 稍後在此過程中需要 `console.log` 和 `process.argv` 您將新增至 `hello.ts` 文件。

3. 將下列程式碼加入到 `hello.ts` 檔案：

```
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 + '.');
```

步驟 3：執行程式碼

1. 在終端中，從相同的目錄 `hello.ts` 檔案中，執行 TypeScript 編譯器。指定 `hello.ts` 檔案和要包含的其他程式庫。

```
tsc hello.ts --lib es6
```

TypeScript 使用 `hello.ts` 文件和一組電子印刷稿 6 (ES6) 庫文件來轉換 TypeScript 在中的程式碼 `hello.ts` 文件轉換為等效 JavaScript 名為的文件中的代碼 `hello.js`。

2. 在 Environment (環境) 視窗中開啟 `hello.js` 檔案。
3. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
4. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Node.js。
5. 在 Command (命令) 中輸入 `hello.js 5 9`。在程式碼中，5 代表 `process.argv[2]`，而 9 代表 `process.argv[3]`。(`process.argv[0]` 代表名稱執行時間 (node)，而 `process.argv[1]` 代表檔案名稱 (`hello.js`))。
6. 選擇 Run (執行)，然後比較您的輸出。完成後，選擇 Stop (停止)。

```
Hello, World!
The sum of 2 and 3 is 5.
The sum of 5 and 9 is 14.
```




Note

除了在 IDE 中建立新的執行組態，您也可以從終端機執行命令 **node hello.js 5 9** 來執行此程式碼。

步驟 4：安裝和配置 AWS 適用於的 SDK JavaScript 在 Node.js 中

您可以增強此範例以使用 AWS 適用於的 SDK JavaScript 在 Node.js 中創建一個 Amazon S3 存儲桶，列出可用的存儲桶，然後刪除剛創建的存儲桶。

在此步驟中，您將安裝並配置 AWS 適用於的 SDK JavaScript 在 Node.js 中。SDK 提供了一種方便的互動方式 AWS 服務，如亞馬遜 S3，從您 JavaScript 代碼。安裝之後 AWS 適用於的 SDK JavaScript 在 Node.js 中，您必須在環境中設定認證管理。SDK 需要這些登入資料，才能與 AWS 服務互動。

若要安裝 AWS 適用於的 SDK JavaScript 在 Node.js 中

在終端工作階段中 AWS Cloud9 IDE，從相同的目錄 `hello.js` 檔案來源 [步驟 3：執行程式碼](#)，執行 **npm** 若要安裝 AWS 適用於的 SDK JavaScript 在 Node.js 中。

```
npm install aws-sdk
```

此命令會新增數個資料夾到 [步驟 3：執行程式碼](#) 中的 `node_modules` 資料夾。這些資料夾包含的原始程式碼和相依性 AWS 適用於的 SDK JavaScript 在 Node.js 中。如需詳細資訊，請參閱 [為下列項目安裝 SDK JavaScript 在 AWS SDK for JavaScript 開發者指南](#)。

在環境中設定憑證管理

每次使用AWS適用於的 SDKJavaScript在 Node.js 中調用一個AWS服務，您必須在呼叫中提供一組憑據。這些認證決定是否AWS適用於的 SDKJavaScript在 Node.js 中具有適當的權限來進行該調用。如果登入資料未涵蓋適當許可，呼叫即會失敗。

在此步驟中，您會在環境中存放您的憑證。若要這麼做，請遵循 [從 AWS Cloud9 的環境呼叫 AWS 服務](#) 中的指示，然後返回本主題。

如需詳細資訊，請參閱 AWS SDK for JavaScript 開發人員指南中的 [以 Node.js 設定憑證](#)。

步驟 5：新增 AWS SDK 程式碼

在此步驟中，您會再新增其他程式碼，這次是要與 Amazon S3 互動，藉此建立儲存貯體、列出可用的儲存貯體，然後刪除您剛建立的儲存貯體。您稍後將執行此程式碼。

1. 在 AWS Cloud9 IDE 中，前往與前述步驟的 `hello.js` 檔案相同的目錄，建立名為 `s3.ts` 的檔案。
2. 從中的終端AWS Cloud9IDE 位於相同的目錄`s3.ts`檔案中，啟用程式碼以非同步方式呼叫 Amazon S3 作業，方法是執行 **npm** 兩次安裝異步庫TypeScript並再次為JavaScript。

```
npm install @types/async # For TypeScript.
npm install async        # For JavaScript.
```

3. 將下列程式碼加入到 `s3.ts` 檔案：

```
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
]);
```

步驟 6：執行 AWS SDK 程式碼

1. 在終端中，從相同的目錄 `s3.ts` 檔案中，執行 TypeScript 編譯器。指定 `s3.ts` 檔案和要包含的其他程式庫。

```
tsc s3.ts --lib es6
```

TypeScript 使用 `s3.ts` 檔案，該 AWS 適用於的 SDK JavaScript 在 Node.js 中，異步庫和一組電子印刷稿 6 (ES6) 庫文件來轉換 TypeScript 在中的程式碼 `s3.ts` 文件轉換為等效 JavaScript 名為的文件中的代碼 `s3.js`。

2. 在 Environment (環境) 視窗中開啟 `s3.js` 檔案。
3. 在選單列上，選擇 Run (執行)、Run Configurations (執行組態)、New Run Configuration (新增執行組態)。
4. 在 [New] - Idle ([新增] - 閒置) 標籤上，選擇 Runner: Auto (執行器: 自動)，然後選擇 Node.js。
5. 針對 Command (命令)，輸入 `s3.js YOUR_BUCKET_NAME THE_AWS_REGION`，其中 `YOUR_BUCKET_NAME` 是您要建立然後刪除的儲存貯體名稱，而 `THE_AWS_REGION` 是要在其

中建立儲存貯體之 AWS 區域的 ID。例如，若為美國東部 (俄亥俄) 區域，則使用 us-east-2。其他區域的 ID 請參閱 [《》 章節 Amazon Simple Storage Service \(Amazon S3\) Amazon Web Services](#) 一般參考。

Note

不只是在您的 AWS 帳戶中，Amazon S3 儲存貯體名稱在整個 AWS 中皆不可重複。

6. 選擇 Run (執行)，然後比較您的輸出。完成後，選擇 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:
```

步驟 7：清除

為避免在您結束使用此範例後您的 AWS 帳戶持續產生費用，您應刪除該環境。如需說明，請參閱 [刪除 AWS Cloud9 中的環境](#)。

碼頭教程 AWS Cloud9

本教學說明如何將 AWS Cloud9 SSH 開發環境連接到 Amazon Amazon EC2 中亞馬遜 Linux 執行個體內的執行中 Docker 容器。這可讓您透過 AWS Cloud9 IDE 來使用 Docker 容器內的程式碼和檔案，並在該容器執行命令。如需有關 Docker 的詳細資訊，請參閱 Docker 網站上的 [什麼是 Docker](#)。

遵循本教學課程並建立此範例可能會對您的 AWS 帳戶產生費用。包括 Amazon EC2 等服務的可能花費。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

主題

- [必要條件](#)

- [步驟 1：安裝和執行 Docker](#)
- [步驟 2：建置影像](#)
- [步驟 3：執行容器](#)
- [步驟 4：建立環境](#)
- [步驟 5：執行程式碼](#)
- [步驟 6：清除](#)

必要條件

- 您應該已有執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。這個範例假設您已經有一個在 AWS 帳戶中執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體。若要啟動 Amazon EC2 執行個體，請參閱[啟動 Linux 虛擬機器](#)。在該精靈的 Choose an Amazon Machine Image (AMI) (選擇 Amazon Machine Image (AMI)) 頁面中，選擇顯示名稱開頭為 Amazon Linux AMI 或 Ubuntu Server 的 AMI。
- 如果 Amazon EC2 執行個體在 Amazon VPC 中執行，尚需符合其他需求。請參閱[AWS Cloud9 開發環境的 VPC 設定](#)。
- Amazon EC2 執行個體都應該至少有 8 到 16 GB 可用的磁碟空間。此範例使用大小超過 3 GB 的 Docker 映像和可以使用 3 GB 或更多磁碟空間的額外增量來建置映像。如果您在可用空間為 8 GB 或更少的磁碟上嘗試執行此範例，我們發現 Docker 映像可能不會建置或者 Docker 容器可能不會執行。若要確認執行個體的可用磁碟空間，您可以在執行個體上執行 `df -h` 之類的命令 (適用於以人類可讀取格式的磁碟檔案系統資訊)。若要增加現有執行個體的磁碟大小，請參閱 Amazon EC2 Linux 執行個體使用者指南中的[修改磁碟區](#)。

步驟 1：安裝和執行 Docker

在此步驟中，您會檢查 Docker 是否已安裝在 Amazon EC2 執行個體，若否則進行安裝。在安裝 Docker 後，您會在執行個體上執行它。

1. 使用 SSH 用戶端 (例如 `ssh` 公用程式或 PuTTY) 來連接至執行中的 Amazon EC2 執行個體。若要這樣做，請參閱[啟動 Linux 虛擬機器](#)中的「步驟 3：連接到執行個體」。
2. 檢查在執行個體上是否已安裝 Docker。若要這樣做，請在執行個體上執行 `docker` 命令並使用 `--version` 選項。

```
docker --version
```

如果 Docker 已安裝，即會顯示 Docker 版本和建置號碼。在這種情況下，請直接跳到此程序中稍後的步驟 5。

3. 安裝 Docker. 若要這樣做，請以 **install** 動作執行 **yum** 或 **apt** 命令，並指定要安裝的 **docker** 或 **docker.io** 套件。

針對 Amazon Linux：

```
sudo yum install -y docker
```

針對 Ubuntu Server：

```
sudo apt install -y docker.io
```

4. 確認 Docker 已安裝。若要進行這項動作，請重新執行 **docker --version** 指令。即會顯示 Docker 版本和建置號碼。
5. 執行 Docker。若要這樣做，請使用 **docker** 服務和 **start** 動作執行 **service** 命令。

```
sudo service docker start
```

6. 確認 Docker 正在執行中。若要這樣做，請執行 **docker** 命令與 **info** 動作。

```
sudo docker info
```

如果 Docker 正在執行中，即會顯示 Docker 的相關訊息。

步驟 2：建置影像

在此步驟中，您使用 Dockerfile 建置 Docker 影像到該執行個體。這個範例使用映像，其中包含 Node.js 和範例聊天伺服器應用程式。

1. 在執行個體上建立 Dockerfile。若要這樣做，在 SSH 用戶端仍然連接到執行個體的情況下，在執行個體上的 /tmp 目錄中，建立一個名為 Dockerfile 的檔案。例如，執行 **touch** 命令，如下所示。

```
sudo touch /tmp/Dockerfile
```

2. 將下列內容新增至 Dockerfile 檔案：

```
# 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.*/%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
```

若要將上述內容新增至 Dockerfile 檔案，您可以使用執行個體上的 **vi** 公用程式，如下所示。

- a. 使用AWS Cloud9開啟和編輯檔/tmp/Dockerfile案。

```
sudo vi /tmp/Dockerfile
```

- b. 將上述內容貼到 Dockerfile 檔案。如果您不確定如何執行此作業，請參閱 SSH 用戶端文件。
- c. 切換至命令模式。若要執行此操作，請按 Esc 鍵 (-- INSERT -- 從視窗底部消失)。
- d. 輸入 :wq (寫入 /tmp/Dockerfile 檔案、儲存檔案，然後結束 **vi**)，然後按下 Enter。

Note

您可以從 AWS CodeBuild 存取經常更新的 Docker 影像清單。如需詳細資訊，請參閱AWS CodeBuild使用者指南 CodeBuild中[提供的 Docker 影像](#)。

3. 在執行個體上，建立一個檔案，其中包含要使用之 Docker 容器的 AWS Cloud9 SSH 公有金鑰。若要這樣做，舉例來說，可以透過執行 Dockerfile authorized_keys 命令，在與 **touch** 檔案所在的相同目錄中，建立名為 的檔案。

```
sudo touch /tmp/authorized_keys
```

4. 將 AWS Cloud9 SSH 公有金鑰新增到 authorized_keys 檔案。若要取得 AWS Cloud9 SSH 公有金鑰，請執行下列動作：
 - a. [請在以下位置開啟AWS Cloud9主控台](https://console.aws.amazon.com/cloud9/)。 <https://console.aws.amazon.com/cloud9/>
 - b. 在 AWS 導覽列的 AWS 區域選擇工具中，選擇您在這個主題稍後要建立 AWS Cloud9 開發環境的 AWS 區域。
 - c. 若顯示歡迎頁面，請針對 New AWS Cloud9 environment (新增 AWS Cloud9 環境)，選擇 Create environment (建立環境)。否則，請選擇 Create environment (建立環境)。
 - d. 在 Name environment (為環境命名) 頁面的 Name (名稱) 內，輸入環境的名稱。(此處的名稱並不重要。您稍後還會選擇不同的名稱。)
 - e. 選擇 Next step (下一個步驟)。
 - f. 對於 Environment type (環境類型)，選擇 Connect and run in remote server (SSH) (連線並在遠端伺服器中執行 (SSH))。

- g. 展開 View public SSH key (查看公有 SSH 金鑰)。
 - h. 選擇 Copy key to clipboard (複製金鑰至剪貼簿)。(這是介於 View public SSH key (查看公有 SSH 金鑰) 和 Advanced settings (進階設定) 之間。)
 - i. 選擇取消。
 - j. 將剪貼簿的內容貼到 authorized_keys 檔案，然後儲存檔案。例如，您可以使用 **vi** 公用程式，如此步驟稍早所述。
5. 以 **build** 動作執行 **docker** 命令，並將標籤 cloud9-image:latest 新增到影像，並指定要使用的 Dockerfile 檔案路徑，藉此建置影像。

```
sudo docker build -t cloud9-image:latest /tmp
```

如果成功，建置輸出的最後兩行會顯示 Successfully built 和 Successfully tagged。

若要確認 Docker 是否順利建置映像，執行 **docker** 命令並搭配 image ls 動作。

```
sudo docker image ls
```

若成功，輸出會顯示一個項目，其中 REPOSITORY 欄位設為 cloud9-image 且 TAG 欄位設為 latest。

6. 請記下 Amazon EC2 執行個體的公有 IP 地址。您在[步驟 4：建立環境](#)時會需要此資訊。如果您不確定執行個體的公有 IP 地址為何，您可以對該執行個體執行以下命令，來取得此資訊。

```
curl http://169.254.169.254/latest/meta-data/public-ipv4
```

步驟 3：執行容器

在此步驟中，您在執行個體上執行 Docker 容器。此容器是根據您在之前步驟所建的映像。

1. 若要執行 Docker 容器，在執行個體上執行的 **docker** 命令搭配 **run** 動作和以下選項。

```
sudo docker run -d -it --expose 9090 -p 0.0.0.0:9090:22 --name cloud9 cloud9-image:latest
```

- **-d** 會在分離模式中執行容器，會在用來執行容器 (在這個範例中，也就是 SSH 用戶端) 的根程序結束時便跟著結束。

- `-it` 會使用分配的 `pseudo-TTY` 來執行容器，並讓 `STDIN` 保持開放 (如果未連接容器)。
- `--expose` 讓使用者能夠從容器使用指定的連接埠 (在這個範例中，連接埠 `9090`)。
- `-p` 讓使用者能夠透過指定的 IP 地址和連接埠來讓指定連接埠可在內部供 Amazon EC2 執行個體使用。在這個範例中，容器上的連接埠 `9090` 可透過 Amazon EC2 執行個體上的連接埠 `22` 來進行內部存取。
- `--name` 是容器之可供人類讀取的名稱 (在此範例中，`cloud9`)。
- `cloud9-image:latest` 要用來執行容器的建置映像之可供人類讀取的名稱。

若要確認 Docker 是否順利執行容器，執行 `docker` 命令並搭配 `container ls` 動作。

```
sudo docker container ls
```

若成功，輸出會顯示一個項目，其中 `IMAGE` 欄位設為 `cloud9-image:latest` 且 `NAMES` 欄位設為 `cloud9`。

2. 登入執行中容器。若要這樣做，請執行 `docker` 命令並使用 `exec` 動作和下列選項。

```
sudo docker exec -it cloud9 bash
```

- `-it` 會使用分配的 `pseudo-TTY` 來執行容器，並讓 `STDIN` 保持開放 (如果未連接容器)。
- `cloud9` 是執行中容器之可供人類讀取的名稱。
- `bash` 在執行中容器中開始標準 shell。

如果成功，終端提示變更會顯示容器的登入使用者名稱和容器 ID。

Note

如果您想要登出執行中的容器，執行 `exit` 命令。終端提示會變回會顯示執行個體的登入使用者名稱和執行個體的私有 DNS。容器仍應執行中。

3. 對於在登入後您希望 AWS Cloud9 從其開始的執行中容器上的目錄，將其存取許可設為 `rwxr-xr-x`。這表示擁有者的 `read-write-execute` 權限、群組的讀取執行權限，以及其他人的讀取執行權限。例如，如果目錄的路徑是 `~`，您可以在執行中容器中執行 `chmod` 命令，以在目錄上設定這些許可，如下所示。

```
sudo chmod u=rwx,g=rx,o=rx ~
```

- 請記下在包含 Node.js 二進位執行容器上的目錄路徑，因為您在[步驟 4：建立環境](#)時會需要此資訊。如果您不確定此路徑為何，對執行中容器執行以下命令來取得此資訊。

```
which node
```

步驟 4：建立環境

在此步驟中，您會使用 AWS Cloud9 來建立 AWS Cloud9 SSH 開發環境，並將其連接到執行中的 Docker 容器。在 AWS Cloud9 建立環境後，它會顯示 AWS Cloud9 IDE，讓您可以開始使用容器中的檔案和程式碼。

您可以使用 AWS Cloud9 主控台建立 AWS Cloud9 SSH 開發環境。您不能使用 CLI 建立 SSH 環境。

必要條件

- 先確定您已完成 [設定 AWS Cloud9](#) 中的步驟。您就能登入 AWS Cloud9 主控台並建立環境。
- 指定您希望要讓 AWS Cloud9 連線至環境的現有雲端運算執行個體 (例如您 AWS 帳戶中的 Amazon EC2 執行個體)，或您自己的伺服器。
- 確保現有的執行個體或您自己的伺服器符合所有 [SSH 主機要求](#)。這些條件包括擁有 Python、Node.js 和其他已安裝元件的特定版本；在您想要 AWS Cloud9 於登入之後從中啟動的目錄上設定特定許可；以及設定任何相關聯的 Amazon Virtual Private Cloud。

建立 SSH 環境

- 請確認您已完成前述必要條件。
- 如果您尚未連線，請使用 SSH 用戶端連線到您的現有執行個體或自己的伺服器。這能確保您可以將必要的公有 SSH 金鑰值新增至執行個體或伺服器。此程序稍後會再進一步說明。

Note

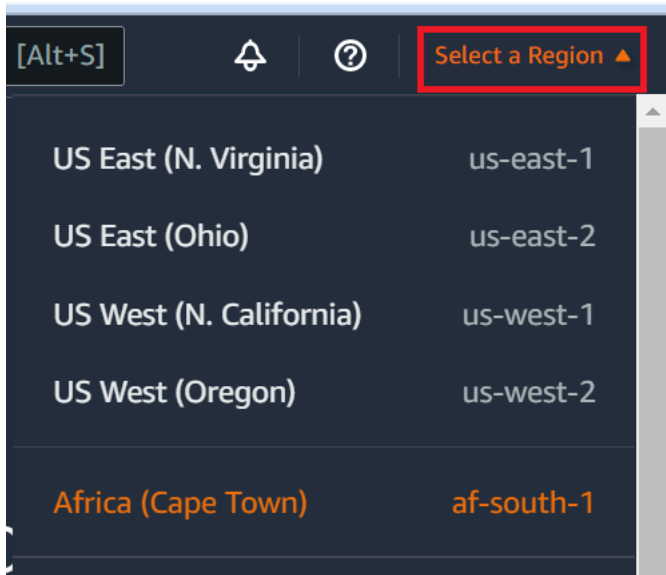
若要連線到現有的 AWS 雲端運算執行個體，請參閱下列一個或多個資源：

- 如為 Amazon EC2，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的[連線至 Linux 執行個體](#)。

- 針對 Amazon Lightsail，請參閱 [Amazon Lightsail 文件](#) 中的連線至您的 Linux/Unix 型 Lightsail 執行個體。
- 如為 AWS Elastic Beanstalk，請參閱 AWS Elastic Beanstalk 開發人員指南中的 [列出並連線到伺服器執行個體](#)。
- 如為 AWS OpsWorks，請參閱 AWS OpsWorks 使用者指南中的 [使用 SSH 登入 Linux 執行個體](#)。
- 如為其他 AWS 服務，請參閱特定服務的文件。

若要連線到您自己的伺服器，請使用 SSH。SSH 已安裝在 macOS 和 Linux 作業系統。若要在 Windows 上使用 SSH 連接伺服器，您必須安裝 [PuTTY](#)。

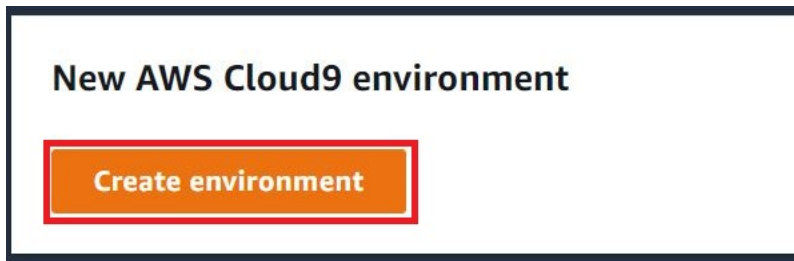
3. 登入 AWS Cloud9 主控台，網址為 <https://console.aws.amazon.com/cloud9/>。
4. 在頂端導覽列中登入 AWS Cloud9 主控台後，請選擇 AWS 區域 來建立環境。如需可用的清單 AWS 區域，請參閱 [AWS Cloud9](#) 中的 AWS 一般參考。



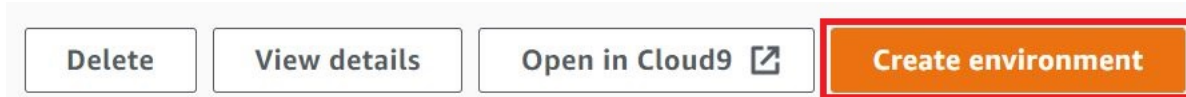
5. 如果這是您第一次建立開發環境，畫面會顯示歡迎頁面。在新的 AWS Cloud9 環境面板中，選擇 Create environment (建立環境)。

如果您先前已建立開發環境，也可以展開畫面左側的窗格。選擇 Your environments (您的環境)，然後選擇 Create environment (建立環境)。

在 welcome (歡迎) 頁面中：



或者在 Your environments (您的環境) 頁面中：



6. 在 Create environment (建立環境) 頁面上，輸入環境的名稱。
7. 在 Description (描述) 中，輸入環境的相關資訊。在本教學中，使用 This environment is for the AWS Cloud9 tutorial.
8. 針對 Environment type (環境類型)，請從下列選項中選擇 Existing Compute (現有運算)：
 - New EC2 instance (新 EC2 執行個體) – 啟動 AWS Cloud9 可以直接透過 SSH 進行連線的 Amazon EC2 執行個體。
 - Existing compute (現有運算) – 啟動不需要任何開放式傳入連接埠的 Amazon EC2 執行個體。AWS Cloud9 透過 [AWS Systems Manager](#) 連線至執行個體。
 - 如果您選取 Existing compute (現有運算) 選項，系統會建立服務角色和 IAM 執行個體描述檔，以允許 Systems Manager 代表您與 EC2 執行個體互動。您可以在界面中繼續往下拉到 Service role and instance profile for Systems Manager access (Systems Manager 存取的服務角色和執行個體描述檔) 區塊檢視兩者的名稱。如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

⚠ Warning

為環境建立 EC2 執行個體可能會讓您的 AWS 帳戶產生 Amazon EC2 的費用。使用 Systems Manager 管理 EC2 執行個體的連線不需要額外費用。

⚠ Warning

AWS Cloud9 使用 SSH 公有金鑰安全地連接到您的伺服器。若要建立安全連線，請將公有金鑰新增到您的 `~/.ssh/authorized_keys` 檔案中，並提供以下步驟中的登入憑證。選擇 Copy key to clipboard (複製金鑰至剪貼簿) 以複製 SSH 金鑰，或選擇 View public SSH key (檢視公有 SSH 金鑰) 來顯示。

9. 在 Existing compute (現有運算) 面板上，針對 User (使用者)，輸入您稍早在此程序中，用來連線至執行個體或伺服器的登入名稱。例如，針對 AWS 雲端 運算執行個體，它可能是 `ec2-user`、`ubuntu` 或 `root`。

ℹ Note

建議將登入名稱與執行個體或伺服器上的管理許可或管理員使用者建立關聯。尤其，建議這個登入名稱擁有執行個體或伺服器上的 Node.js 安裝。若要加以確認，請從執行個體或伺服器的終端機中執行命令 `ls -l $(which node)` (如果您使用 `npm`，也可以執行 `ls -l $(npm which node)`)。此命令會顯示 Node.js 安裝的擁有者名稱。同時也會顯示安裝的許可、群組名稱和位置。

10. 針對 Host (主機)，輸入執行個體或伺服器的公有 IP 地址 (偏好) 或主機名稱。
11. 針對 Port (連接埠)，輸入您想要 AWS Cloud9 用來嘗試連線至執行個體或伺服器的連接埠。或者，也可以保留預設連接埠。
12. 選擇 Additional details - optional (其他詳細資訊 - 選用)，顯示環境路徑、node.js 二進位檔路徑和 SSH 跳接主機資訊。
13. 在 Environment path (環境路徑) 中輸入您要 AWS Cloud9 在執行個體或伺服器上啟動的目錄路徑。您會在此程序稍早的先決條件中指定此路徑。如果您將此空白，則 AWS Cloud9 會在登入之後使用您的執行個體或伺服器一般用來啟動的目錄。這通常是家用或預設目錄。
14. 針對 Path to Node.js binary path (node.js 二進位檔路徑)，請輸入路徑資訊，以指定執行個體或伺服器上的 Node.js 二進位檔路徑。若要取得路徑，您可以在執行個體或伺服器上執行命令 `which node` (如果您使用 `npm`，也可以執行 `npm which node`)。例如，路徑可能是 `/usr/bin/node`。如果您保留空白，則 AWS Cloud9 會在嘗試連線 Node.js 二進位檔時嘗試猜測其所在的位置。
15. 針對 SSH jump host (SSH 跳接主機)，請輸入執行個體或伺服器使用的跳接主機相關資訊。使用格式 `USER_NAME@HOSTNAME:PORT_NUMBER` (例如 `ec2-user@ip-192-0-2-0:22`)。

跳接主機必須符合下列要求：

- 它必須可使用 SSH 透過公有網際網路存取。
 - 它必須允許使用任何 IP 地址透過指定的連接埠傳入存取。
 - 複製至現有執行個體或伺服器上 `~/.ssh/authorized_keys` 檔案的公有 SSH 金鑰值，也必須複製至跳接主機上的 `~/.ssh/authorized_keys` 檔案。
 - 必須安裝 Netcat。
16. 藉由為每個標籤提供一個索引鍵和一個值，最多新增 50 個標籤。選取 Add new tag (新增標籤) 來執行此操作。標籤將附加至 AWS Cloud9 環境做為資源標籤，並傳播至以下基礎資源：AWS CloudFormation 堆疊、Amazon EC2 執行個體，以及 Amazon EC2 安全群組。若要進一步了解標籤，請參閱 [IAM 使用者指南](#) 中的 [使用 AWS 資源標籤控制存取](#)，以及本指南中有關標籤的 [進階資訊](#)。

Warning

如果您在建立這些標籤之後予以更新，這些變更不會傳播至基礎資源。如需詳細資訊，請參閱 [標籤相關進階資訊](#) 中的 [將標籤更新傳播至基礎資源](#)。

17. 選擇 Create (建立) 以建立您的環境，然後系統會將您重新導向至首頁。成功建立帳戶後，AWS Cloud9 主控台上方會出現綠色閃爍列。您可以選取新環境，然後選擇 Open in Cloud9 (在 Cloud9 中開啟) 以啟動 IDE。

Delete

View details

Open in Cloud9 

Create environment

若無法建立帳戶，AWS Cloud9 主控台上方會出現紅色閃爍列。您的 Web 瀏覽器、AWS 存取許可、執行個體或相關的網路問題皆可能導致帳戶無法建立。您可以在 [AWS Cloud9 疑難排解區段](#) 找到可行修正方法，解決造成帳戶無法使用的問題。

Note

如果您的環境使用代理來存取網際網路，您必須提供代理詳細資訊給 AWS Cloud9 以便安裝相依性。如需詳細資訊，請參閱 [安裝相依性失敗](#)。

步驟 5：執行程式碼

在此步驟中，您會使用 AWS Cloud9 IDE 在執行中的 Docker 容器執行範例應用程式。

1. 在執行中容器的 AWS Cloud9 IDE 顯示後，開始使用範例聊天伺服器。若要這麼做，在 Environment (環境) 視窗中，以滑鼠右鍵按一下範例 workspace/server.js 檔案，然後選擇 Run (執行)。
2. 預覽範例應用程式。若要執行此操作，請從 Environment (環境) 視窗開啟 workspace/client/index.html 檔案。然後，在功能表列中，選擇 Tools, Preview, Preview Running Application (工具、預覽、預覽執行應用程式)。
3. 在應用程式預覽索引標籤上，對於 Your Name (您的姓名)，輸入名稱。對於 Message (訊息)，請輸入訊息。然後選擇 Send (傳送)。聊天伺服器會將您的名稱和訊息新增至清單。

步驟 6：清除

在此步驟中，您會刪除環境並將 AWS Cloud9 和 Docker 支援檔案從 Amazon EC2 執行個體移除。此外，為避免系統在您結束使用此範例後仍對您的 AWS 帳戶持續收費，您應終止執行 Docker 的 Amazon EC2 執行個體。

步驟 6.1：刪除環境

若要刪除環境，請參閱[刪除 AWS Cloud9 中的環境](#)。

步驟 6.2：從容器移除 AWS Cloud9 支援檔案

在刪除環境後，部分 AWS Cloud9 支援檔案仍會保留在容器中。如果您想要繼續使用容器，但不再需要這些支援檔案，請自您指定 AWS Cloud9 在登入後啟動的容器上的目錄中將 .c9 資料夾刪除。例如，如果目錄是 ~，請以 `-r` 選項執行 `rm` 命令，如下所示。

```
sudo rm -r ~/.c9
```

步驟 6.3：從執行個體移除 Docker 支援檔案

如果您希望不再保留 Docker 容器、Docker 影像和 Amazon EC2 執行個體上的 Docker，但您想要保留執行個體，您可以移除這些 Docker 支援檔案，如下所示。

1. 將 Docker 容器從執行個體中移除。若要這樣做，請以 `stop` 和 `rm` 在執行個體上執行 `docker` 命令，停止動作和容器的人類可識讀名稱。

```
sudo docker stop cloud9
sudo docker rm cloud9
```

2. 將 Docker 映像從執行個體中移除。若要這樣做，請在執行個體上執行 **docker** 命令並使用 **image rm** 動作和映像的標籤。

```
sudo docker image rm cloud9-image:latest
```

3. 移除任何可能仍然結束的額外 Docker 支援檔案。若要這樣做，請在執行個體上執行 **docker** 命令並使用 **system prune** 動作。

```
sudo docker system prune -a
```

4. 取消安裝 Docker。若要這樣做，請以 **remove** 動作在執行個體上執行 **yum** 命令，並指定要取消安裝的 **docker** 套件。

針對 Amazon Linux：

```
sudo yum -y remove docker
```

針對 Ubuntu Server：

```
sudo apt -y remove docker
```

您也可以移除您先前建立的 Dockerfile 和 authorized_keys 檔案。例如，在執行個體上執行 **rm** 命令。

```
sudo rm /tmp/Dockerfile
sudo rm /tmp/authorized_keys
```

步驟 6.4：終止執行個體

若要終止 Amazon EC2 執行個體，請參閱 Amazon EC2 Linux 執行個體使用者指南中的[終止執行個體](#)。

相關教學

- 在[關於AWS RoboMaker 開發人員指南AWS RoboMaker 中開始使用](#)。本教學課程用AWS Cloud9於修改、建置和捆綁範例自動機制應用程式。

AWS Cloud9 進階主題

這些主題包含下列資訊：

- 用於進階組態和決策的資訊。
- 與特定任務相關，且可讓您深入了解 AWS Cloud9 的資訊，但您不一定需要此資訊才能完成該任務。

主題

- [比較 AWS Cloud9 中的 EC2 環境與 SSH 環境](#)
- [AWS Cloud9 開發環境的 VPC 設定](#)
- [SSH 環境主機要求](#)
- [在 AWS Cloud9 SSH 環境中使用 AWS Cloud9 安裝程式](#)
- [AWS Cloud9 的傳入 SSH IP 地址範圍](#)
- [AWS Cloud9 EC2 開發環境的 Amazon Machine Image \(AMI\) 內容](#)
- [使用 AWS Cloud9 的服務連結角色](#)
- [使用 AWS CloudTrail 記錄 AWS Cloud9 API 呼叫](#)
- [標籤](#)

比較 AWS Cloud9 中的 EC2 環境與 SSH 環境

如[環境和運算資源介紹](#)以及[使用環境](#)所述，您的 AWS Cloud9 環境可設定為 EC2 環境或 SSH 環境。

下表指出在 AWS Cloud9 中使用 EC2 環境和 SSH 環境的異同之處。

EC2 環境	SSH 環境
AWS Cloud9 建立關聯的 Amazon EC2 執行個體，並管理執行個體的生命週期。這包括啟動、停止和終止操作。	您可以使用現有雲端運算執行個體或自己的伺服器。您必須負責管理其生命週期。
此執行個體會 Amazon Linux 或 Ubuntu Server 上執行。	您可以使用任何可執行 Linux 的雲端運算執行個體，也可以使用自己的伺服器執行 Linux。

EC2 環境	SSH 環境
AWS Cloud9 會自動設定執行個體開始使用 AWS Cloud9。	您必須手動設定執行個體或自己的伺服器來使用 AWS Cloud9。
AWS Cloud9 會在執行個體上自動設定 AWS Command Line Interface (AWS CLI)。	如果您要在執行個體或自己的伺服器上使用 AWS CLI，則必須自行設定。
執行個體可以存取數百個有用的套件，且已安裝和設定一些常見套件。範例包括 Git、Docker Node.js 和 Python。	您可能需要下載、安裝和設定其他套件以完成常見任務。
您可以維護執行個體，例如定期套用系統更新。	您可以維護執行個體或自己的伺服器。
當您刪除環境時，AWS Cloud9 會自動終止相關聯的執行個體。	刪除環境後，執行個體或您自己的伺服器會保留下來。
<p>AWS 受管臨時憑證可在 EC2 環境中使用。使用這些憑證，您可以針對呼叫者 AWS 帳戶中的所有 AWS 資源開啟或關閉所有 AWS 動作，但有一些限制。您無需為環境的 Amazon EC2 執行個體設定執行個體設定檔或存放 AWS 實體 (例如 IAM 使用者) 的永久 AWS 存取憑證。如果您環境的 Amazon EC2 執行個體啟動到私有子網路，則無法使用 AWS 受管暫時登入資料來允許 Amazon EC2 環境代表 AWS 實體 (例如 IAM 使用者) 存取 AWS 服務。</p>	<p>AWS 受管臨時憑證在 SSH 環境中不可用。您必須使用 AWS Identity and Access Management 來管理許可，這些許可可讓您使用 AWS Cloud9 和其他 AWS 服務與資源。</p>
<p>AWS 工具組、Git 面板和增強的 Java 支援可供使用。</p>	<p>AWS 工具組、Git 面板和增強的 Java 支援不可用。</p>

AWS Cloud9 開發環境的 VPC 設定

每個與 Amazon Virtual Private Cloud (Amazon VPC) 相關聯的 AWS Cloud9 開發環境都必須符合特定 VPC 要求。這些環境包括 EC2 環境，以及與在 VPC 內執行的 AWS 雲端 運算執行個體相關聯的 SSH 環境。範例包括 Amazon EC2 和 Amazon Lightsail 執行個體。

AWS Cloud9 的 Amazon VPC 需求

AWS Cloud9 所使用的 Amazon VPC 需具有以下設定。如果您已熟知這些需求，並且只想建立相容的 VPC，請跳至 [建立 VPC 以及其他 VPC 資源](#)。

使用以下檢查清單確認 VPC 符合所有以下要求：

- VPC 可以在相同的位置 AWS 帳戶和 AWS 區域作為 AWS Cloud9 開發環境或 VPC 可以是不同的共享 VPC AWS 帳戶比環境。但是，VPC 必須在相同的位置 AWS 區域作為環境。有關亞馬遜 VPC 的更多信息 AWS 區域，請參閱 [檢視 AWS 區域的 VPC 清單](#)。有關為其創建亞馬遜 VPC 的更多說明 AWS Cloud9，請參閱 [建立 VPC 以及其他 VPC 資源](#)。如需使用共用 Amazon VPC 的相關資訊，請參閱 [使用共用 VPC](#) 在亞馬遜 VPC 用戶指南。
- VPC 必須具有公用子網路。如果子網路的流量路由到網際網路閘道，則子網路是公用的。如需 Amazon VPC 的子網路清單，請參閱 [檢視 VPC 的子網路清單](#)。
- 如果您的環境透過 SSH 直接存取其 EC2 執行個體，則您只能在公有子網路中啟動執行個體。如需確認子網路是否為公用的資訊，請參閱 [確認子網路是否為公有子網路](#)。
- 如果使用 Systems Manager 存取 [無輸入的 Amazon EC2 執行個體](#)，則可在公有或私有子網路中啟動執行個體。
- 如果您使用公有子網路，請將網際網路閘道連接至 VPC。就是這樣 AWS Systems Manager Agent (SSM Agent) 執行個體可連線至系統管理員。
- 如果您使用的是私有子網路，請允許子網路執行個體在公有子網路中代管 NAT 閘道，藉此與網際網路通訊。如需檢視或變更網際網路閘道設定的詳細資訊，請參閱 [檢視或變更網際網路閘道的設定](#)。
- 公有子網路的路由表必須具備一組採用最基本設定的路由。若要瞭解如何確認子網路是否具有路由表，請參閱 [確認子網路是否擁有路由表](#)。如需有關如何建立路由表的資訊，請參閱 [建立路由表](#)。
- 相關的 VPC 安全群組 (或相關的 AWS 雲端運算執行個體安全群組，這會依據架構而定) 必須允許一組最基本的傳入與傳出流量。如需 Amazon VPC 的安全群組清單，請參閱 [檢視 VPC 的安全群組清單](#)。如需在 Amazon VPC 中建立安全群組的詳細資訊，請參閱 [在 VPC 中建立安全群組](#)。
- 如要獲得多一層的安全保障，VPC 所具備的網路 ACL 必須允許一組最基本的傳入與傳出流量。若要確認 Amazon VPC 是否具有至少一個網路 ACL，請參閱 [確認 VPC 是否至少有一個網路 ACL](#)。若要取得有關建立網路 ACL 的資訊，請參閱 [〈建立網路 ACL〉](#)。
- 如果您的開發環境 [使用 SSM 存取 EC2 執行個體](#)，請確保該執行個體已由公有子網路指派了可在子網路中啟動的公有 IP 地址。若要這麼做，您必須為公用子網路啟用自動指派公用 IP 位址選項，並將其設定為 Yes。您可以在建立公用子網路之前啟用此功能 AWS Cloud9 子網路設定頁面中的環境。如需修改公用子網路中自動指派 IP 設定的相關步驟，請參閱 [修改子網路的公用 IPv4 位址屬性](#) 在亞馬遜 VPC 用戶指南。如需設定公用和私有子網路的詳細資訊，請參閱 [將子網路設定為公有或私有](#)

Note

如需下列程序，請登入AWS Management Console並使用管理員登入資料開啟 Amazon VPC 主控台 (<https://console.aws.amazon.com/vpc>) 或亞馬遜 EC2 控制台 (<https://console.aws.amazon.com/ec2>)。

如果您使用AWS CLI或AWS CloudShell，我們建議您設定AWS CLI或AWS CloudShell使用您的管理員的憑據AWS 帳戶。如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

檢視 AWS 區域的 VPC 清單

若要使用 Amazon VPC 主控台，請在 AWS 的導覽列中，選擇 AWS Cloud9 要建立環境的所在 AWS 區域。接著由導覽窗格中選擇 Your VPCs (您的 VPC)。

如需使用 AWS CLI 或 AWS CloudShell，請執行 Amazon EC2 **describe-vpcs** 命令，如下所示。

```
aws ec2 describe-vpcs --output table --query 'Vpcs[*].VpcId' --region us-east-2
```

在上述命令中，將 `us-east-2` 取代為 AWS Cloud9 要在其中建立環境的 AWS 區域。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含 VPC ID 的清單。

檢視 VPC 的子網路清單

若要使用 Amazon VPC 主控台，請從導覽窗格中選擇 Your VPCs (您的 VPC)。記下 VPC ID 欄內的 VPC ID。接著，從導覽窗格中選擇 Subnets (子網路)，並在 VPC 欄內尋找包含該 ID 的子網路。

如需使用 AWS CLI 或 `aws-shell`，請執行 Amazon EC2 **describe-subnets** 命令，如下所示。

```
aws ec2 describe-subnets --output table --query 'Subnets[*].[SubnetId,VpcId]' --region us-east-2
```

在上述命令中，將 `us-east-2` 取代為包含子網路的 AWS 區域。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

在輸出中，查看符合 VPC ID 的子網路。

確認子網路是否為公有子網路

Important

假設您要將環境的 EC2 執行個體啟動至私有子網路。確定允許該執行個體的輸出流量，以便其可以連線到 SSM 服務。若為私有子網路，傳出流量通常是透過網路位址轉譯 (NAT) 閘道或 VPC 端點所設定。(NAT 閘道需要公有子網路。)

假設您存取 SSM 時選擇 VPC 端點，而非 NAT 閘道。如果執行個體的自動更新和安全性修補程式依賴網際網路存取，則其可能無法運作。您可以使用其他應用程式，例如 [AWS Systems Manager 修補程式管理員](#)，以管理您的環境可能需要的任何軟體更新。AWS Cloud9 軟體將如常更新。

若要使用 Amazon VPC 主控台，請在導覽窗格中，選擇 Subnets (子網路)。選取您想要 AWS Cloud9 使用之子網路旁的方塊。在 Route Table (路由表) 索引標籤上，若 Target (目標) 欄內具有以 igw- 開頭的項目，即表示該子網路為公有子網路。

若要使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含子網路的 AWS 區域，並將 `subnet-12a3456b` 取代為子網路 ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

在輸出中，如果至少有一個開頭為 `igw-` 的結果，代表子網路為公有。

在輸出中，如果沒有結果，路由表可能與 VPC (而不是子網路) 相關聯。若要確認此情況，請針對子網路相關的 VPC (而不是子網路本身) 執行 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
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

在輸出中，如果至少有一個開頭為 `igw-` 的結果，則 VPC 包含網際網路閘道。

檢視或變更網際網路閘道的設定

若要使用 Amazon VPC 主控台，請在導覽窗格中，選擇 Internet Gateways (網際網路閘道)。選取網際網路閘道旁的方塊。查看每個標籤以檢視各項設定。若要變更任一標籤上的設定，請在適用的情況下選擇 Edit (編輯)，然後遵循螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell` 來查看設定，請執行 Amazon EC2 **`describe-internet-gateways`** 命令。

```
aws ec2 describe-internet-gateways --output table --region us-east-2 --internet-gateway-id igw-1234ab5c
```

在上述命令中，將 `us-east-2` 取代為包含網際網路閘道的 AWS 區域，並將 `igw-1234ab5c` 取代為網際網路閘道 ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

建立網際網路閘道

若要使用 Amazon VPC 主控台，請在導覽窗格中，選擇 Internet Gateways (網際網路閘道)。接著，選擇 Create internet gateway (建立網際網路閘道)，然後遵循螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell`，請執行 Amazon EC2 **`create-internet-gateway`** 命令。

```
aws ec2 create-internet-gateway --output text --query 'InternetGateway.InternetGatewayId' --region us-east-2
```

在上述命令中，將 `us-east-2` 取代為包含新網際網路閘道的 AWS 區域。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含新網際網路閘道的 ID。

將網際網路閘道連接至 VPC

若要使用 Amazon VPC 主控台，請在導覽窗格中，選擇 Internet Gateways (網際網路閘道)。選取網際網路閘道旁的方塊。在適用的情況下，選擇 Actions, Attach to VPC (動作：連接至 VPC)，然後遵循螢幕上的指示進行。

如需使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含網際網路閘道的 AWS 區域。將 `igw-a1b2cdef` 取代為網際網路閘道 ID。然後將 `vpc-1234ab56` 取代為 VPC ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

確認子網路是否擁有路由表

若要使用 Amazon VPC 主控台，請在導覽窗格中，選擇 Subnets (子網路)。選取您要供 AWS Cloud9 使用的 VPC 公有子網路旁的方塊。在 Route table (路由表) 標籤上，若 Route Table (路由表) 內出現數值，即表示該公有子網路有路由表。

若要使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含公有子網路的 AWS 區域，並將 `subnet-12a3456b` 取代為公有子網路 ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

如果輸出中有值，則公有子網路擁有至少有一個路由表。

在輸出中，如果沒有結果，路由表可能與 VPC (而不是子網路) 相關聯。若要確認此情況，請針對子網路相關的 VPC (而不是子網路本身) 執行 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
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

如果在輸出中至少有一個結果，則 VPC 至少有一個路由表。

將路由表連接至子網路

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Route Tables (路由表)。選取在您要連接之路由表旁邊的方塊。在 Subnet Associations (子網路關聯) 索引標籤中，選擇 Edit (編輯)，選擇您要將連接到其中之子網路旁的方塊，然後選擇 Save (儲存)。

如需使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含路由表的 AWS 區域 區域。將 `subnet-12a3456b` 取代為子網路 ID。然後將 `rtb-ab12cde3` 取代為路由表 ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

建立路由表

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Route Tables (路由表)。選擇 Create Route Table (建立路由表)，然後依照螢幕上的指示進行。

如需使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含新路由表的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 ("")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含新路由表的 ID。

檢視或變更路由表的設定

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Route Tables (路由表)。選取路由表旁的方塊。查看每個標籤以檢視各項設定。若要變更任一標籤上的設定，請選擇 Edit (編輯)，然後依照螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell` 來查看設定，請執行 Amazon EC2 **describe-route-tables** 命令，如下所示。

```
aws ec2 describe-route-tables --output table --region us-east-2 --route-table-ids rtb-ab12cde3
```

在上述命令中，將 `us-east-2` 取代為包含新路由表的 AWS 區域，並將 `rtb-ab12cde3` 取代為路由表 ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

AWS Cloud9 最基本的路由表建議設定

目的地	目標	狀態	傳播
CIDR-BLOCK	本機	作用中	否
0.0.0.0/0	igw-INTERNET-GATEWAY-ID	作用中	否

上述設定中，`CIDR-BLOCK` 是子網路的 CIDR 區塊，而 `igw-INTERNET-GATEWAY-ID` 是相容的網際網路閘道的 ID。

檢視 VPC 的安全群組清單

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Security Groups (安全群組)。接著，在 Search Security Groups (搜尋安全群組) 方塊中輸入 VPC 的 ID 或名稱，並按下 Enter。搜尋結果清單將顯示該 VPC 的安全群組。

若要使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 ("")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含該 VPC 安全群組 ID 的清單。

檢視 AWS 雲端運算執行個體的安全群組清單

若要使用 Amazon EC2 主控台，請在導覽窗格中展開 Instances (執行個體)，接著選擇 Instances (執行個體)。在執行個體清單中，選擇執行個體旁的方塊。該執行個體的安全群組會隨即顯示於 Security groups (安全群組) 旁的 Description (說明) 標籤。

如需使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含執行個體的 AWS 區域，並將 `i-12a3c456d789e0123` 取代為執行個體 ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含該執行個體安全群組 ID 的清單。

在 VPC 中檢視或變更安全群組的設定

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Security Groups (安全群組)。選取安全群組旁的方塊。查看每個標籤以檢視各項設定。若要變更任一標籤上的設定，請在適用的情況下選擇 Edit (編輯)，然後遵循螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell` 來查看設定，請執行 Amazon EC2 **describe-security-groups** 命令，如下所示。

```
aws ec2 describe-security-groups --output table --region us-east-2 --group-ids
sg-12a3b456
```

在上述命令中，將 `us-east-2` 取代為包含執行個體的 AWS 區域，並將 `sg-12a3b456` 取代為安全群組 ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

檢視或變更 AWS 雲端運算執行個體安全群組的設定

若要使用 Amazon EC2 主控台，請在導覽窗格中展開 Instances (執行個體)，接著選擇 Instances (執行個體)。在執行個體清單中，勾選執行個體旁的方塊；在 Description (說明) 標籤的 Security groups (安全群組) 中，選擇安全群組。請查看每個標籤；若要變更任一標籤上的設定，請在適用的情況下選擇 Edit (編輯)，然後遵循螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell` 來查看設定，請執行 Amazon EC2 **describe-security-groups** 命令，如下所示。

```
aws ec2 describe-security-groups --output table --region us-east-2 --group-ids
sg-12a3b456
```

在上述命令中，將 `us-east-2` 取代為包含執行個體的 AWS 區域，並將 `sg-12a3b456` 取代為安全群組 ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

AWS Cloud9 最基本的傳入與傳出流量設定

⚠ Important

執行個體的 IA 安全群組可能沒有傳入規則。如果發生此情況，這表示源自其他主機的傳入流量不允許送至執行個體。如需使用無輸入 EC2 執行個體的相關資訊，請參閱[使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)。

- 傳入：透過連接埠 22 使用 SSH 的所有 IP 地址。不過，您可以將這類 IP 地址限制為僅由 AWS Cloud9 使用的地址。如需詳細資訊，請參閱[AWS Cloud9 的傳入 SSH IP 地址範圍](#)。

📘 Note

對於 2018 年 7 月 31 日當天或之後建立的 EC2 環境，AWS Cloud9 會使用安全群組，限制透過連接埠 22 使用 SSH 的傳入 IP 地址。這些傳入的 IP 地址特別僅是 AWS Cloud9 使用的地址。如需詳細資訊，請參閱[AWS Cloud9 的傳入 SSH IP 地址範圍](#)。

- 傳入 (僅限網路 ACL)：對於 EC2 環境和與執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體相關聯的 SSH 環境，所有 IP 地址都會透過連接埠 32768-61000 使用 TCP。如需詳細資訊及其他 Amazon EC2 執行個體類型的連接埠範圍，請參閱 Amazon VPC 使用者指南中的[暫時性連接埠](#)。
- 傳出：使用任何通訊協定與連接埠的所有流量來源。

您可以在安全群組層級設定相應的行為。如要獲得多一層的安全保障，您也可以使用網路 ACL。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[比較安全群組和網路 ACL](#)。

例如，若要為安全群組新增傳入及傳出規則，您可以按照下述方式設定相應的規則。

傳入規則

類型	通訊協定	連接埠範圍	Source (來源)
SSH (22)	TCP (6)	22	0.0.0.0 (但請參閱以下注意事項和 AWS Cloud9 的傳入 SSH IP 地址範圍 。)

Note

對於 2018 年 7 月 31 日當天或之後建立的 EC2 環境，AWS Cloud9 會新增傳入規則，限制透過連接埠 22 使用 SSH 的傳入 IP 地址。這特別僅限於 AWS Cloud9 使用的地址。如需詳細資訊，請參閱[AWS Cloud9 的傳入 SSH IP 地址範圍](#)。

傳出規則

類型	通訊協定	連接埠範圍	Source (來源)
所有流量	ALL	ALL	0.0.0.0/0

如果您也選擇將傳入及傳出規則新增至網路 ACL，則可以設定如下的規則。

傳入規則

規則 #	類型	通訊協定	連接埠範圍	Source (來源)	允許/拒絕
100	SSH (22)	TCP (6)	22	0.0.0.0 (但請參閱 AWS Cloud9 的傳入 SSH IP 地址範圍 。)	允許
200	自訂 TCP 規則	TCP (6)	32768-61000 (適用於 Amazon Linux 和 Ubuntu Server 執行個體。其他執行個體類型請參閱 暫時性連接埠 。)	0.0.0.0/0	允許
*	所有流量	ALL	ALL	0.0.0.0/0	拒絕

傳出規則

規則 #	類型	通訊協定	連接埠範圍	Source (來源)	允許/拒絕
100	所有流量	ALL	ALL	0.0.0.0/0	允許
*	所有流量	ALL	ALL	0.0.0.0/0	拒絕

如需安全群組與網路 ACL 的詳細資訊，請參閱 Amazon VPC 使用者指南中的下列內容。

- [安全性](#)
- [VPC 的安全群組](#)
- [網路 ACL](#)

在 VPC 中建立安全群組

若要使用 Amazon VPC 或 Amazon EC2 主控台，請執行以下其中一項動作：

- 在 Amazon VPC 主控台的導覽窗格中，選擇 Security Groups (安全群組)。選擇 Create Security Group (建立安全群組)，然後依照螢幕上的指示進行。
- 在 Amazon EC2 主控台的導覽窗格中，展開 Network & Security (網路與安全)，接著選擇 Security Groups (安全群組)。選擇 Create Security Group (建立安全群組)，然後依照螢幕上的指示進行。

如需使用 AWS CLI 或 `aws-shell`，請執行 Amazon EC2 **create-security-group** 命令，如下所示。

```
aws ec2 create-security-group --region us-east-2 --vpc-id vpc-1234ab56
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

確認 VPC 是否至少有一個網路 ACL

若要使用 Amazon VPC 主控台，請從導覽窗格中選擇 Your VPCs (您的 VPC)。選擇您想要 AWS Cloud9 使用之 VPC 旁的方塊。在 Summary (摘要) 索引標籤上，若 Network ACL (網路 ACL) 內出現數值，即表示該 VPC 至少有一份網路 ACL。

若要使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

如果輸出至少包含清單中一個項目，VPC 至少有一個網路 ACL。

檢視 VPC 的網路 ACL 清單

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Network ACLs (網路 ACL)。在 Search Network ACLs (搜尋網路 ACL) 方塊中輸入 VPC 的 ID 或名稱，並按下 Enter。搜尋結果清單將顯示該 VPC 的網路 ACL。

若要使用 AWS CLI 或 `aws-shell`，請執行 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
```

在上述命令中，將 `us-east-2` 取代為包含 VPC 的 AWS 區域，並將 `vpc-1234ab56` 取代為 VPC ID。若要在 Windows 中執行上述命令，請將單引號 (') 取代為雙引號 (" ")。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

輸出包含該 VPC 網路 ACL 的清單。

檢視或變更網路 ACL 的設定

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Network ACLs (網路 ACL)。選擇網路 ACL 旁的方塊。查看每個標籤以檢視各項設定。若要變更任一標籤上的設定，請在適用的情況下選擇 Edit (編輯)，然後遵循螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell` 來查看設定，請執行 Amazon EC2 **`describe-network-acls`** 命令。

```
aws ec2 describe-network-acls --output table --region us-east-2 --network-acl-ids
acl-1234ab56
```

在上述命令中，將 `us-east-2` 取代為包含網路 ACL 的 AWS 區域，並將 `acl-1234ab56` 取代為網路 ACL ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

建立網路 ACL

若要使用 Amazon VPC 主控台，請在導覽窗格中選擇 Network ACLs (網路 ACL)。選擇 Create Network ACL (建立網路 ACL)，然後依照螢幕上的指示進行。

若要使用 AWS CLI 或 `aws-shell`，請執行 Amazon EC2 **`create-network-acl`** 命令。

```
aws ec2 create-network-acl --region us-east-2 --vpc-id vpc-1234ab56
```

在上述命令中，將 `us-east-2` 取代為包含您要將新網路 ACL 連接至其中之 VPC 的 AWS 區域。亦將 `vpc-1234ab56` 取代為 VPC ID。若要使用 `aws-shell` 執行上述命令，請省略 `aws`。

建立 VPC 以及其他 VPC 資源

使用下列程序建立 VPC 以及執行應用程式所需的其他 VPC 資源。VPC 資源包括子網路、路由表、網際網路閘道和 NAT 閘道。

如何使用主控台建立 VPC、子網路和其他 VPC 資源

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在 VPC 儀表板上，選擇 Create VPC (建立 VPC)。
3. 針對 Resources to create (建立資源)，選擇 VPC and more (VPC 等)。
4. 若要建立 VPC 資源的名稱標籤，請保留自動產生名稱標籤已選取。若要為 VPC 資源提供您自己的名稱標籤，請將其清除。
5. 對於 CIDR 區塊，您必須輸入 VPC 的 IPv4 位址範圍。建議使用的 IPv4 系列，適用於 AWS Cloud9 是 `10.0.0.0/16`。
6. (選用) 若要支援 IPv6 流量，請選擇 IPv6 CIDR 區塊 > Amazon 提供的 IPv6 CIDR 區塊。
7. 選擇租用選項。此選項會定義啟動至 VPC 的 EC2 執行個體，是否會在與其他 AWS 帳戶共用的硬體上執行，或是在僅供您使用的硬體上執行。如果您選擇 VPC 的租賃是 Default，啟動到此 VPC 的 EC2 執行個體將使用啟動執行個體時指定的租用屬性。如需詳細資訊，請參閱 [使用定義的參數啟動例證](#) 在亞馬遜 EC2 Linux 執行個體使用者指南。

如果您選擇 VPC 的租用為 Dedicated，則執行個體將會一律以硬體上之 [專用預留執行個體](#) (專供您使用) 的形式執行。如果您正在使用 AWS Outposts，您的 Outpost 需要私人連接，您必須使用 Default 租賃。

8. 對於可用區域數量 (AZ)，我們建議您至少佈建兩個子網路Availability Zones適用於生產環境。若要選擇子網路的 AZ，請展開自訂 AZ。否則，你可以讓AWS選擇適合您的 AZ。
9. 若要設定您的子網路，請選擇公有子網路數目和私有子網路數目的值。若要選擇子網路的 IP 地址範圍，請展開自訂子網路 CIDR 區塊。否則，請讓 AWS 為您選擇設定。
10. (選擇性) 如果私有子網路中的資源需要透過 IPv4 存取公用網際網路：用於NAT 閘道」下方，選擇要在其中建立 NAT 閘道的 AZ 數目。在生產環境中，建議您在每個 AZ 中部署一個 NAT 閘道，並包含需要存取公有網際網路的資源。
11. (選擇性) 如果私有子網路中的資源需要透過 IPv6 存取公用網際網路：用於僅限輸出網際網路閘道，選擇是。
12. (選擇性) 若要直接從您的虛擬私人雲端存取 Amazon S3，請選擇VPC 端點,S3 閘道。這會為 Amazon S3 建立閘道 VPC 端點。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的[閘道 VPC 端點](#)。
13. (選用) 在 DNS 選項，網域名稱解析的兩個選項都會依預設啟用。如果預設值不符合您的需求，您可以停用這些選項。
14. (選用) 若要將標籤新增至 VPC，請展開其他標籤，選擇新增標籤，然後輸入標籤金鑰和標籤值。
15. 在預覽」窗格中，您可以視覺化您設定的 VPC 資源之間的關係。實線代表資源之間的關係。虛線代表 NAT 閘道、網際網路閘道和閘道端點的網路流量。建立 VPC 後，可以隨時使用資源映射索引標籤以此格式將 VPC 中的資源視覺化。
16. 完成 VPC 的設定後，請選擇建立 VPC。

僅建立 VPC

使用 Amazon VPC 主控台，使用下列程序建立不含額外 VPC 資源的 VPC。

如何使用主控台建立不含任何其他 VPC 資源的 VPC

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在 VPC 儀表板上，選擇 Create VPC (建立 VPC)。
3. 在建立資源，選擇僅 VPC。
4. (選用) 在名稱標籤中，輸入您 VPC 的名稱。執行此作業會使用 Name 做為索引鍵，以及您指定的值來建立標籤。
5. 對於 IPv4 CIDR 區塊，執行下列其中一個動作：
 - 選擇 IPv4 CIDR 手動輸入，然後輸入 VPC 的 IPv4 地址範圍。建議使用的 IPv4 系列，適用於 AWS Cloud9是10.0.0.0/16。

- 選擇 IPAM 分配的 CIDR 塊中，選取一個亞馬遜 VPC IP 位址管理員 (IPAM) IPv4 位址集區和一個網路遮罩。CIDR 區塊的大小會受 IPAM 集區上的分配規則所限制。IPAM 是一項 VPC 功能，可協助您規劃、追蹤和監控 IP 位址 AWS 工作負載。如需詳細資訊，請參閱 [什麼是 IPAM ?](#) 在亞馬遜虛擬私有雲管理員指南。

如果您使用 IPAM 來管理 IP 位址，建議您選擇此選項。否則，您為 VPC 指定的 CIDR 區塊可能會與 IPAM CIDR 配置重疊。

6. (選用) 若要建立雙堆疊 VPC，請為您的 VPC 指定 IPv6 地址範圍。對於 IPv6 CIDR 區塊，執行下列其中一個動作：
 - 選擇 IPAM 配置的 IPv6 CIDR 區塊，然後選取您的 IPAM IPv6 地址集區和網路遮罩。CIDR 區塊的大小會受 IPAM 集區上的分配規則所限制。
 - 若要從 IPv6 位址的亞馬遜集區要求 IPv6 CIDR 區塊，請選擇亞馬遜提供的 CIDR 塊。在 Network Border Group (網路邊界群組) 欄位中，請選取 AWS 在其中公告 IP 地址的群組。Amazon 提供的 IPv6 CIDR 區塊固定大小為 /56。
 - 選擇我擁有的 IPv6 CIDR，以便使用您透過 [自攜 IP 地址 \(BYOIP\)](#) 帶到 AWS 的 IPv6 CIDR 區塊。在 Pool (集區) 欄位中，選擇要從中配置 IPv6 CIDR 區塊的 IPv6 地址集區。
7. (選用) 選擇租用選項。此選項會定義啟動至 VPC 的 EC2 執行個體，是否會在與其他 AWS 帳戶共用的硬體上執行，或是在僅供您使用的硬體上執行。如果您選擇 VPC 的租賃是 Default，啟動到此 VPC 的 EC2 執行個體將使用啟動執行個體時指定的租用屬性。如需詳細資訊，請參閱 [使用定義的參數啟動例證](#) 在亞馬遜 EC2 Linux 執行個體使用者指南。

如果您選擇 VPC 的租用為 Dedicated，則執行個體將會一律以硬體上之 [專用預留執行個體](#) (專供您使用) 的形式執行。如果您正在使用 AWS Outposts，您的 Outpost 需要私人連接，您必須使用 Default 租賃。

8. (選用) 若要將標籤新增至 VPC，請選擇新增標籤，然後輸入標籤金鑰和標籤值。
9. 選擇 Create VPC (建立 VPC)。
10. 建立 VPC 後，您可以新增子網。

為 AWS Cloud9 建立子網路

您可以使用 Amazon VPC 主控台，為與 AWS Cloud9 相容的 VPC 建立子網路。您是否可以為 EC2 執行個體建立私有子網路或公有子網路，取決於您的環境連線到執行個體的方式：

- 透過 SSH 直接存取：僅限公有子網路
- 透過 Systems Manager 存取權：公有或私有子網路皆可

在私有子網路中啟動環境的 EC2 的選項僅適用於您使用[主控台、命令列或 AWS CloudFormation](#) 建立「無輸入」EC2 環境的情況。

您必須按照可將子網路設為公有或私有的[相同子網路建立步驟](#)操作。如果子網路與路由至網際網路閘道的路由表建立關聯，就會成為公有子網路。但如果子網路與未路由至網際網路閘道的路由表建立關聯，則成為私有子網路。如需詳細資訊，請參閱[將子網路設定為公有或私有](#)

若您已遵循上述程序完成 AWS Cloud9 的 VPC 建立作業，便不需要再進行本程序，因為建立新的 VPC 精靈會自動為您建立子網路。

Important

- AWS 帳戶在環境的同一個 AWS 區域內必須已有相容的 VPC。如需詳細資訊，請參閱 [AWS Cloud9 的 Amazon VPC 需求](#) 所述的 VPC 需求。
- 在本程序中，建議您在 AWS 帳戶中使用 IAM 管理員的憑證來登入 AWS Management Console，並開啟 Amazon VPC 主控台。如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員。
- 某些組織可能不允許您自行建立子網路。若您無法建立子網路，請洽詢您的 AWS 帳戶管理員或網路管理員。

建立子網路

1. 如果尚未開啟 Amazon VPC 主控台，請前往 <https://console.aws.amazon.com/vpc> 登入 AWS Management Console 並開啟 Amazon VPC 主控台。
2. 在導覽列中，如果 AWS 區域不同於環境的區域，請選擇正確的區域。
3. 如果尚未顯示 Subnets (子網路) 頁面，從導覽窗格中選擇 Subnets (子網路)。
4. 選擇 Create Subnet (建立子網路)。
5. 在 Create Subnet (建立子網路) 對話方塊中，為 Name tag (名稱標籤) 輸入子網路的名稱。
6. 對於 VPC，選擇要與此子網路關聯的 VPC。
7. 對於 Availability Zone (可用區域)，選擇 AWS 區域內要供此子網路使用的可用區域，或選擇 No Preference (無偏好設定) 交由 AWS 為您選擇可用區域。
8. 在 IPv4 CIDR block (IPv4 CIDR 區塊) 中，以 CIDR 格式輸入要供此子網路使用的 IP 地址範圍。該 IP 地址範圍必須是 VPC 中的 IP 地址之子集。

如需 CIDR 區塊的相關資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 和子網路規模調整](#)。另請參閱 [3.1. RFC 4632 的基本概念與字首標記法](#)，或 Wikipedia 中的 [IPv4 CIDR 區塊](#)。

建立子網路後，[設定為公有或私有子網路](#)。

將子網路設定為公有或私有

建立子網路之後，您可以指定子網路與網際網路通訊的方式，藉此設為公有或私有。

公有子網路具有公有 IP 地址，且與網際網路閘道 (IGW) 連接，以便允許子網路執行個體與網際網路及其他 AWS 服務之間的通訊。

私有子網路中的執行個體具有私有 IP 地址，且會使用網路位址轉譯 (NAT) 閘道來傳送子網路執行個體與網際網路及其他 AWS 服務之間的往來流量。NAT 閘道必須託管在公有子網路中。

Public subnets

Note

即使您環境的執行個體是在私有子網路中啟動，VPC 也必須具備至少一個公有子網路。這是因為轉送執行個體往來流量的 NAT 閘道必須託管在公有子網路中。

將子網路設定為公有時，必須將網際網路閘道 (IGW) 連接到子網路，設定路由表以指定該 IGW 的路由，並且定義安全群組中的設定以便控制傳入和傳出流量。

如需瞭解如何執行這些任務，請參閱[建立 VPC 以及其他 VPC 資源](#)。

Important

如果您的開發環境[使用 SSM 存取 EC2 執行個體](#)，請確保該執行個體已由公有子網路指派了可在子網路中啟動的公有 IP 地址。若要這麼做，您必須為公用子網路啟用自動指派公用 IP 位址選項，並將其設定為 Yes。您可以在建立公用子網路之前啟用此功能 AWS Cloud9 子網路設定頁面中的環境。如需修改公用子網路中自動指派 IP 設定的相關步驟，請參閱[修改子網路的公用 IPv4 位址屬性](#)在亞馬遜 VPC 用戶指南。如需有關設定公用和私有子網路的詳細資訊，請參閱[將子網路設定為公有或私有](#)。

Private subnets

如果您建立透過 Systems Manager 存取的無輸入執行個體，您可以在私有子網路中啟動該執行個體。私有子網路沒有公有 IP 地址。因此，您需要 NAT 閘道將私有 IP 地址對應至請求的公有地址，並且還需要將公有 IP 地址對應回至回應的私有地址。

Warning

您將需要為您帳戶中建立及使用的 NAT 閘道支付費用。適用的費率為 NAT 閘道的每小時用量率及資料處理率。同時也適用 Amazon EC2 資料傳輸費。如需詳細資訊，請參閱 [Amazon VPC 定價](#)。

建立和設定 NAT 閘道前，您必須先執行以下作業：

- 建立公有 VPC 子網路來代管 NAT 閘道。
- 佈建可指派至 NAT 閘道的 [彈性 IP 地址](#)。
- 如為私有子網路，請清除 Enable auto-assign public IPv4 address (啟用自動指派公有 IPv4 地址) 核取方塊，以便為要在其中啟動的執行個體指派私有 IP 地址。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [在 VPC 設定 IP 地址](#)。

如需瞭解任務步驟，請參閱 Amazon VPC 使用者指南中的 [使用 NAT 閘道](#)。

Important

目前，如果您環境的 EC2 執行個體已在私有子網路中啟動，您就無法使用 [AWS 受管臨時憑證](#)，來允許 EC2 環境代表 AWS 實體 (例如 IAM 使用者) 存取 AWS 服務。

SSH 環境主機要求

若要指示 AWS Cloud9 將環境連線至現有雲端運算執行個體或自有伺服器，您必須建立 AWS Cloud9 SSH 開發環境。不過，在建立 SSH 環境之前，請考量建立 EC2 環境的好處。

當您建立 EC2 環境時，AWS Cloud9 會建立新的環境，請求 Amazon EC2 啟動新的執行個體，然後將新啟動的執行個體連線到新的環境。建立 EC2 環境的優點如下：

- 自動執行個體啟動。當您建立 EC2 環境時，AWS Cloud9 會同時請求 Amazon EC2 建立新的執行個體。在 SSH 環境中，您必須提供現有的雲端運算執行個體 (例如 Amazon EC2 執行個體) 或您自己的伺服器。
- 自動執行個體關機。根據預設，在為 EC2 環境連線到 IDE 的所有 Web 瀏覽器執行個體關閉 30 分鐘後，AWS Cloud9 會自動關閉 EC2 環境。您可隨時變更此行為。這個動作有助於減少您使用 Amazon EC2 的 AWS 帳戶產生額外費用。
- 自動執行個體清理。您刪除 EC2 環境時，也會自動刪除連線的 Amazon EC2 執行個體。這個動作同樣有助於減少您使用 Amazon EC2 的 AWS 帳戶產生額外費用。在連接到雲端運算執行個體的 SSH 環境中，您必須記得自行刪除該執行個體。
- AWS 的受管臨時憑證在 EC2 環境中，您可以輕鬆在呼叫者的 AWS 帳戶中開啟或關閉所有 AWS 資源的所有 AWS 動作 (存在部分限制)。您無需為環境的 Amazon EC2 執行個體設定執行個體描述檔或存放 AWS 實體 (例如 IAM 使用者) 的永久 AWS 存取憑證。

如需詳細資訊，請參閱[AWS 受管理的臨時憑](#)。

- AWS Toolkit 和 Git 面板。用於與 AWS 服務互動和使用視覺化來源控制的這些工具僅在 AWS Cloud9 環境中提供，這些環境由 Amazon EC2 執行個體建立。

如果您要改為建立 EC2 環境，請參閱[建立 EC2 環境](#)。否則，請繼續閱讀關於建立 SSH 環境的資訊。

建立 SSH 環境環境的時機與方法

如果您符合以下任何要求，就必須建立 SSH 環境，而不是 EC2 環境：

要求	指示
您不想要由於使用 AWS 雲端運算執行個體而對 AWS 帳戶產生額外費用。因此，您決定將 AWS Cloud9 連線至 AWS 外的現有雲端運算執行個體，或改為連線至自有伺服器。	<ol style="list-style-type: none"> 1. 確定您的執行個體或伺服器符合本主題稍後說明的要求。 2. 建立 SSH 環境讓 AWS Cloud9 可將您的執行個體或伺服器與其連線。
您想要使用您 AWS 帳戶中現有的 AWS 雲端運算執行個體 (例如 Amazon EC2 執行個體)，而不是讓 AWS Cloud9 在建立環境的同時啟動新的執行個體。	<ol style="list-style-type: none"> 1. 確定執行個體符合本主題稍後描述的要求。 2. 建立 SSH 環境讓 AWS Cloud9 將執行個體與其連線。

要求	指示
您想要使用 AWS Cloud9 目前不支援 EC2 環境使用的 Amazon EC2 執行個體類型 (例如 R4)。	<ol style="list-style-type: none"> 1. 根據所需的執行個體類型 啟動 Amazon EC2 執行個體。或者，找出您 AWS 帳戶中執行所需執行個體類型的現有執行個體。 2. 確定執行個體符合本主題稍後描述的 要求。 3. 建立 SSH 環境 讓 AWS Cloud9 將執行個體與其連線。
除了 Amazon Linux 或 Ubuntu Server 之外，您想要使用以 Amazon Machine Image (AMI) 為基礎的 Amazon EC2 執行個體。	<ol style="list-style-type: none"> 1. 根據所需的 AMI 啟動 Amazon EC2 執行個體。或者，找出您 AWS 帳戶中以所需 AMI 為基礎的現有執行個體。 2. 確定執行個體符合本主題稍後描述的 要求。 3. 建立 SSH 環境 讓 AWS Cloud9 將執行個體與其連線。
您希望將多個環境連線到單一現有雲端運算執行個體或您個人的伺服器。	<ol style="list-style-type: none"> 1. 確定執行個體或伺服器符合本主題稍後描述的 要求。 2. 為您希望 AWS Cloud9 連接執行個體或伺服器的每個環境 建立 SSH 環境。

Note

啟動 Amazon EC2 執行個體可能讓您的 AWS 帳戶承擔 Amazon EC2 費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

SSH 主機要求

現有的雲端運算執行個體或您個人的伺服器必須符合下列要求，AWS Cloud9 才可將它連線到 SSH 環境。

- 它必須執行 Linux。(AWS Cloud9 不支援 Windows。)
- 它不得使用 Arm 型架構。(我們目前正在審核以 Arm 處理器建置的系統支援。)
- 它必須可以使用 SSH 透過公有網際網路來聯繫。如果它只能透過 Virtual Private Cloud (VPC) 或虛擬私有網路 (VPN) 來聯繫，則該 VPC 或 VPN 必須可以存取公有網際網路。

- 如果主機是屬於 Amazon Virtual Private Cloud (Amazon VPC) 的現有 AWS 雲端運算執行個體，則還必須符合其他要求。如需詳細資訊，請參閱「[Amazon VPC 設定](#)」。
- 它必須已安裝 Python3，並在安裝 AWS Cloud9 時設定為預設 Python 版本和 pip3。若要檢查版本，請從現有執行個體或您的伺服器的終端機執行 `python --version` 命令。若要在執行個體或伺服器上安裝 Python，請參閱下列其中一項資源：
 - Python 範例中的[步驟 1：安裝必要工具](#)。
 - Python 從 Python 網站[下載](#)。

Note

若要連線至現有的 AWS 雲端運算執行個體，來驗證並滿足要求，請參閱下列一個或多個資源：

- 如為 Amazon EC2，請參閱 Linux 執行個體適用的 Amazon EC2 使用者指南中的[連線至 Linux 執行個體](#)。
- 針對 Amazon Lightsail，請參閱 [Amazon Lightsail 文件](#) 中的連線至您的 Linux/Unix 型 Lightsail 執行個體。
- 如為 AWS Elastic Beanstalk，請參閱 AWS Elastic Beanstalk 開發人員指南中的[列出並連線到伺服器執行個體](#)。
- 如為 AWS OpsWorks，請參閱 AWS OpsWorks 使用者指南中的[使用 SSH 登入 Linux 執行個體](#)。
- 如需其他 AWS 服務，請參閱服務的[文件](#)。

若要連線到您自己的伺服器，來驗證並滿足要求，請從 macOS 或 Linux 使用 "connect to a server by using the SSH command" (使用 SSH 命令連線到伺服器) 或從 Windows 使用 "connect to a server by using PuTTY" (使用 PuTTY 連線到伺服器) 這類詞句搜尋網際網路。

- 執行下列命令以安裝所有必要的套件。

針對 Amazon Linux：

```
sudo yum install -y make glibc-devel gcc gcc-c++
```

針對 Ubuntu Server：

```
sudo apt install build-essential
```

- 它必須安裝了 Node.js。建議您安裝主機作業系統支援的最新 Node.js 版本。

⚠ Warning

如果您使用 AWS Cloud9 不支援的 Node.js 版本，則建立 SSH 環境時可能會發生 AWS Cloud9 安裝問題。

若要檢查版本，請從現有執行個體或您伺服器的終端機執行 `node --version` 命令。若要在執行個體或伺服器上安裝 Node.js，請參閱下列資源：

- Node.js 範例中的 [步驟 1：安裝必要工具](#)。
- Node.js 網站上的 [Installing Node.js via package manager](#)。
- [節點版本管理員](#) 開啟 GitHub。
- 您希望 AWS Cloud9 在登入後啟動之現有執行個體或伺服器上的目錄路徑，必須將其存取許可設定為 `rwxr-xr-x`。這表示擁有者的 `read-write-run` 權限對應於您在 [設定設定] 頁面的 [\[建立使用者\] 精靈](#) 中指定的登入名稱、此擁有者所屬群組的讀取執行權限，以及其他人的讀取執行權限。

例如，如果目錄的路徑為 `~` (其中 `~` 代表您針對 Configure settings (組態設定) 頁面上的 User (使用者) 所指定的登入名稱主目錄)，則您可以在執行個體或伺服器上使用以下命令和後續指示執行 `chmod` 命令，藉此在目錄上設定這些許可。

```
sudo chmod u=rwx,g=rx,o=rx ~
```

- 在現有的執行個體或伺服器上 [下載並執行 AWS Cloud9 安裝程式](#)。
- 您可以選擇性地限制透過 SSH 傳入的流量只可傳入 AWS Cloud9 使用的 IP 地址。若要執行此作業，請將傳入 SSH 流量設定為 [AWS Cloud9 的傳入 SSH IP 地址範圍](#) 中說明的 IP 範圍。

在您確定您的執行個體或伺服器符合前述要求後，就可 [建立目標 SSH 環境](#) 讓 AWS Cloud9 與之連線。

在 AWS Cloud9 SSH 環境中使用 AWS Cloud9 安裝程式

在建立 AWS Cloud9 開發環境之前，雲端運算執行個體 (例如 Amazon EC2 執行個體) 或您想要連線到環境的自有伺服器必須符合 [SSH 主機要求](#)。其中一個要求就是您必須在執行個體或伺服器上下載並執行 AWS Cloud9 安裝程式。AWS Cloud9 安裝程式是一種 Linux Shell 指令碼，其會檢查執行個體或伺服器是否正在 AWS Cloud9 支援的作業系統平台和架構上執行。如果此檢查成功，則這個指令碼會嘗試在執行個體或伺服器上安裝 AWS Cloud9 所需的元件及其相依項目。

本主題描述如何在目標執行個體或伺服器上下載並執行此安裝程式指令碼。

- [下載並執行 AWS Cloud9 安裝程式](#)
- [AWS Cloud9 安裝程式疑難排解](#)

下載並執行 AWS Cloud9 安裝程式

1. 確定雲端運算執行個體或您想要連線到環境的自有伺服器符合 [SSH 主機要求](#)。這些條件包括已安裝 Python 和 Node.js 的特定版本、在您想要 AWS Cloud9 於登入之後從中啟動的目錄上設定特定許可，以及設定任何相關聯的 Amazon Virtual Private Cloud。
2. 當您連接到執行個體或伺服器時，請在該執行個體或伺服器上執行以下其中一個命令。您必須先安裝 gcc 才能執行其中一個指令。

```
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. 如果顯示 Done (完成) 訊息，且沒有出現任何錯誤，則您可以 [建立 SSH 環境](#)。

如果顯示錯誤訊息，請參閱下節以取得疑難排解資訊。

AWS Cloud9 安裝程式疑難排解

本節描述常見問題、可能的原因，以及建議的解決方案，用於 AWS Cloud9 安裝程式錯誤的疑難排解。

如果您的問題未列於其中，或者，如果您需要額外協助，請參閱 [AWS Cloud9 開發論壇](#)。(當您進入此論壇時，AWS 可能會要求您登入。) 您也可以直接 [聯絡我們](#)。

- [-bash: wget: command not found](#)
- [Error: please install make to proceed](#)
- [Error: please install gcc to proceed](#)
- [configure: error: curses not found](#)

-bash: wget: command not found

問題：執行安裝程式指令碼時，顯示以下訊息：-bash: wget: command not found。

可能原因：**wget** 公用程式未安裝在執行個體或伺服器上。

建議的解決方案：改在具有 **curl** 公用程式的執行個體或伺服器上執行安裝程式指令碼。

Error: please install make to proceed

問題：執行安裝程式指令碼時，顯示以下訊息：Error: please install make to proceed。

可能原因：**make** 公用程式未安裝在執行個體或伺服器上。

建議的解決方案：安裝 **make** 公用程式，然後重新在執行個體或伺服器上嘗試執行安裝程式指令碼。

若要安裝 **make** 公用程式，您可以在執行個體或伺服器上執行下列其中一個命令。

- 若是 Amazon Linux、Amazon Linux 2，以及在 Amazon EC2 執行的 Red Hat Enterprise Linux (RHEL)：**sudo yum -y groupinstall "Development Tools"**
- 若是在 Amazon EC2 執行的 Ubuntu Server：**sudo apt install -y build-essential**
- 若為 SUSE：**sudo zypper install -y make**

Error: please install gcc to proceed

問題：執行安裝程式指令碼時，顯示以下訊息：Error: please install gcc to proceed。

可能原因：**gcc** 公用程式未安裝在執行個體或伺服器上。

建議的解決方案：安裝 **gcc** 公用程式，然後重新在執行個體或伺服器上嘗試執行安裝程式指令碼。

若要安裝 **gcc** 公用程式，您可以在執行個體或伺服器上執行下列其中一個命令。

- 若是 Amazon Linux、Amazon Linux 2，以及在 Amazon EC2 執行的 Red Hat Enterprise Linux (RHEL)：**sudo yum -y groupinstall "Development Tools"**
- 若是在 Amazon EC2 執行的 Ubuntu Server：**sudo apt install -y build-essential**
- 若為 SUSE：**sudo zypper install -y gcc**
- 若為其他作業系統，請參閱[安裝 GCC](#)。

configure: error: curses not found

問題：執行安裝程式指令碼時，顯示以下訊息：configure: error: curses not found。

可能原因：**ncurses** 終端機控制程式庫未安裝在執行個體或伺服器上。

建議的解決方案：安裝 **ncurses** 終端機控制程式庫 (在某些作業系統上，則為 **glibc-static** 程式庫)，然後重新在執行個體或伺服器上嘗試執行安裝程式指令碼。

若要安裝 **ncurses** 終端機控制程式庫 (在某些作業系統上，則為 **glibc-static** 程式庫)，您可以在執行個體或伺服器上執行下列其中一個命令。

- 若是 Amazon Linux、Amazon Linux 2，以及在 Amazon EC2 執行的 Red Hat Enterprise Linux (RHEL)：**sudo yum -y install ncurses-devel**
- 若為 SUSE：**sudo zypper install -y ncurses-devel** 和 **sudo zypper install -y glibc-static**

AWS Cloud9 的傳入 SSH IP 地址範圍

您可以將傳入流量限制為僅傳入 AWS Cloud9 透過 SSH 連線到 Amazon VPC 的 AWS 雲端運算執行個體 (例如 Amazon EC2 執行個體) 或您網路中的自有伺服器時使用的 IP 地址範圍。

Note

您可以限制傳入流量，僅有 AWS Cloud9 透過 SSH 連線的 IP 地址範圍。如果您是在 2018 年 7 月 31 日當天或之後建立 EC2 環境，則可以略過此主題。這是因為 AWS Cloud9 會自動將該環境的傳入 SSH 流量限制為只傳入本主題稍後說明的 IP 地址。AWS Cloud9 的做法是自動新增規則至與環境的 Amazon EC2 執行個體關聯的安全群組。此規則將透過連接埠 22 傳入的 SSH 流量限制在關聯 AWS 區域的 IP 地址。針對網路中您的自有伺服器，您仍須遵照本主題稍後所述的步驟。

多數 AWS 區域的 IP 地址範圍位在 `ip-ranges.json` 檔案中，如中的 [AWS IP 地址範圍](#) 所述 AWS 一般參考。

Note

請參閱 [下方](#) 了解目前不包含在 `ip-ranges.json` 檔案中的亞太區域 (香港)、歐洲 (米蘭) 和中東 (巴林) 區域 IP 地址範圍。

尋找 `ip-ranges.json` 檔案中的 IP 範圍：

- 若是 Windows，請使用 AWS Tools for Windows PowerShell 執行以下命令。

```
Get-AWSPublicIpAddressRange -ServiceKey CLOUD9
```

- 若是 Linux，請下載 [ip-ranges.json](#) 檔案。然後，您可以使用 **jq** 這類工具，透過執行以下命令來進行查詢。

```
jq '.prefixes[] | select(.service=="CLOUD9")' < ip-ranges.json
```

這些 IP 範圍可能偶爾會變更。只要有所變更，我們就會傳送 AmazonIpSpaceChanged 主題的通知給訂閱者。若要取得這些通知，請參閱中的 [AWSIP 地址範圍通知AWS 一般參考](#)。

若要將使用 AWS 雲端運算執行個體的環境設定為使用這些 IP 地址範圍，請參閱 [AWS Cloud9 開發環境的 VPC 設定](#)。另外，如果您選擇限制 EC2 環境的傳入流量，或與執行 Amazon Linux 或 Ubuntu Server 的 Amazon EC2 執行個體相關聯的 SSH 環境傳入流量，請務必最少允許使用 TCP 透過 32768-61000 連接埠傳入的所有 IP 地址。如需詳細資訊及其他 AWS 雲端運算執行個體類型的連接埠範圍，請參閱 Amazon VPC 使用者指南中的 [暫時性連接埠](#)。

若要將使用您自有網路的 SSH 環境設定為使用這些 IP 地址範圍，請參閱您網路的文件或聯絡您的網路管理員。

不在 `ip-ranges.json` 中的 IP 地址

以下 AWS 區域的 AWS Cloud9 IP 地址範圍目前不包含在 `ip-ranges.json` 檔案中：亞太區域 (香港)、歐洲 (米蘭) 和中東 (巴林)。下表列出這些區域的 IP 範圍。

Note

每個區域都有兩個 IP 地址範圍，以支援 AWS Cloud9 的控制平面 (資訊路由) 和資料平面 (資訊處理) 服務。

AWS 區域	代碼	IP 範圍 (CIDR 標記法)
亞太區域 (香港)	ap-east1	18.163.201.96/27
		18.163.139.32/27
歐洲 (米蘭)	eu-south-1	15.161.135.64/27

AWS 區域	代碼	IP 範圍 (CIDR 標記法)
		15.161.135.96/27
Middle East (Bahrain)	me-south-1	15.185.141.160/27 15.185.91.32/27

AWS Cloud9 EC2 開發環境的 Amazon Machine Image (AMI) 內容

使用以下資訊來取得 AWS Cloud9 用於 EC2 環境的 Amazon Machine Image (AMI) 詳細資訊。

Important

如果您環境的 Amazon EC2 執行個體是以 Amazon Linux 2023 AMI 或 Amazon Linux 2 AMI 範本為基礎，則安全性更新會在執行個體啟動後立即安裝在執行個體上。接下來，安全性修補程式會每小時自動套用到執行個體。這些更新是由背景處理程序所套用，不會影響執行個體的使用。

在 Ubuntu EC2 環境中，執行個體也會在啟動後立即安裝安全性更新。接下來，unattended-upgrades 套件就會每天自動安裝可用更新。

主題

- [Amazon Amazon](#)
- [Ubuntu Server](#)

Amazon Amazon

Important

我們建議您在[使用主控台建立 Amazon EC2 環境時選擇亞馬遜 Linux 2023](#) 選項。Amazon Linux 2023 AMI 除了提供安全、穩定和高效能的運行時間環境外，還包括到 2024 年的長期支持。

若要顯示 Amazon Linux 執行個體的版本，請從連線環境的 AWS Cloud9 IDE，或從 SSH 公用程式 (例如 ssh 命令或 PuTTY) 執行下列命令。


```
cat /etc/system-release
```

若要顯示 Amazon Linux 執行個體上安裝的套件清單，請執行下列一或多個命令。

若要將所有已安裝的套件顯示為一份清單：

```
sudo yum list installed
```

若要顯示套件名稱包含所指定文字的安裝套件清單：

```
sudo yum list installed | grep YOUR_SEARCH_TERM
```

在上述命令中，將 YOUR_SEARCH_TERM 取代為部分的套件名稱。例如，若要顯示名稱包含 sql 的所有安裝套件清單：

```
sudo yum list installed | grep sql
```

若要顯示所有已安裝的套件清單，請一次一頁顯示一個套件：

```
sudo yum list installed | less
```

若要捲動顯示的頁面：

- 若要下移一行，請按 **j**。
- 若要上移一行，請按 **k**。
- 若要下移一頁，請按 **Ctrl-F**。
- 若要上移一頁，請按 **Ctrl-B**。
- 若要退出，請按 **q**。

Note

使用 Amazon Linux 2 時，您可用 Extras Library 將應用程式和軟體更新安裝至執行個體。這些軟體更新即為主題。如需詳細資訊，請參閱 [Amazon EC2 Linux 執行個體使用者指南中的 Extras Library \(Amazon Linux 2\)](#)。

如需其他選項，請執行 `man yum` 命令。另亦請參閱下列資源：

- Amazon 2023 : [AMI 頁面](#)。
- Amazon Linux : [Amazon Linux AMI 2018.03 套件](#)。

Ubuntu Server

若要顯示 Ubuntu Server 執行個體的版本，請從連線環境的 AWS Cloud9 IDE，或從 SSH 公用程式 (例如 `ssh` 命令或 PuTTY) 執行下列命令。

```
lsb_release -a
```

此版本會顯示在 Description (描述) 欄位旁邊。

若要顯示 Ubuntu Server 上安裝的套件清單，請執行下列一或多個命令。

若要將所有已安裝的套件顯示為一份清單：

```
sudo apt list --installed
```

若要顯示套件名稱包含所指定文字的安裝套件清單：

```
sudo apt list --installed | grep YOUR_SEARCH_TERM
```

在上述命令中，將 `YOUR_SEARCH_TERM` 取代為部分的套件名稱。例如，若要顯示名稱包含 `sql` 的所有安裝套件清單：

```
sudo apt list --installed grep sql
```

若要顯示所有已安裝的套件清單，請一次一頁顯示一個套件：

```
sudo apt list --installed | less
```

若要捲動顯示的頁面：

- 若要下移一行，請按 **j**。
- 若要上移一行，請按 **k**。

- 若要下移一頁，請按 **Ctrl-F**。
- 若要上移一頁，請按 **Ctrl-B**。
- 若要退出，請按 **q**。

如需其他選項，請執行 `man apt` 命令。另請參閱 Ubuntu 網站上的 [Ubuntu 套件搜尋](#)。

使用 AWS Cloud9 的服務連結角色

AWS Cloud9 會使用 AWS Identity and Access Management (IAM) 的 [服務連結角色](#)。服務連結角色是直接連結至 AWS Cloud9 的一種特殊 IAM 角色類型。服務連結角色由 AWS Cloud9 預先定義，且內含該服務代您呼叫其他 AWS 服務所需的所有許可。

服務連結的角色可讓設定 AWS Cloud9 更為簡單，因為您不必新增必要的許可。AWS Cloud9 會定義其服務連結角色的許可，且僅有 AWS Cloud9 可以擔任角色。定義的許可包括信任政策和許可政策，並且該許可政策不能附加到任何其他 IAM 實體。

您必須先刪除角色的相關資源，才能刪除角色。如此可保護您 AWS Cloud9 的資源，避免您不小心移除資源的存取許可。

如需關於支援服務連結角色的其他服務的資訊，請參閱 [可搭配 IAM 運作的 AWS 服務](#)，並尋找服務連結角色欄顯示為是的服務。選擇具有連結的 Yes (是)，以檢視該服務的服務連結角色文件。

- [AWS Cloud9 的服務連結角色許可](#)
- [為 AWS Cloud9 建立服務連結角色](#)
- [為 AWS Cloud9 編輯服務連結角色](#)
- [為 AWS Cloud9 刪除服務連結角色](#)
- [AWS Cloud9 服務連結角色的支援區域](#)

AWS Cloud9 的服務連結角色許可

AWS Cloud9 使用名為 `AWSServiceRoleForAWSCloud9` 的服務連結角色。此服務連結角色信任 `cloud9.amazonaws.com` 服務擔任其角色。

此服務連結角色的許可政策名為 `AWSCloud9ServiceRolePolicy`，它可 AWS Cloud9 允許在指定的資源上完成下政策中列出的動作。

⚠ Important

如果您正在使用 License Manager，並且收到 `unable to access your environment` 錯誤，則需要將舊的服務連結角色取代為支援 License Manager 的版本。您可以透過直接刪除來取代舊角色。然後，更新過的角色會自動建立。

```
{
  "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"
      }
    }
  }
]
}

```

您必須設定許可，以允許 AWS Cloud9 代表 IAM 實體 (如使用者、群組或角色) 建立服務連結角色。

為使能 AWS Cloud9 夠建立 AWS Service Role for AWS Cloud 9 服務連結角色，請將以下陳述式加入至 AWS Cloud9 需要代為建立服務連結角色的 IAM 實體許可政策。

```

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

```

或者，您也可為 IAM 實體新增 AWS 受管政策 `AWSCloud9User` 或 `AWSCloud9Administrator`。

若要允許 IAM 實體刪除AWSServiceRoleForAWSCloud 9 服務連結角色，請將以下陳述式加入至需要刪除服務連結角色的 IAM 實體許可政策。

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": "cloud9.amazonaws.com"
    }
  }
}
```

為 AWS Cloud9 建立服務連結角色

您不需要建立服務連結角色。建立 AWS Cloud9 開發環境後，AWS Cloud9 會為您建立服務連結角色。

為 AWS Cloud9 編輯服務連結角色

您無法在編輯AWSServiceRoleForAWSCloud 9 服務連結角色AWS Cloud9。例如，您一旦建立了服務連結角色，即無法變更該角色的名稱，因為各種實體皆可能會參考該角色。然而，您可使用 IAM 來編輯角色描述。如需更多資訊，請參閱 IAM 使用者指南中的[編輯服務連結角色](#)。

為 AWS Cloud9 刪除服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。

在 IAM 中刪除服務連結角色

使用 IAM 刪除服務連結角色之前，您必須先移除該角色所使用的任何 AWS Cloud9 資源。若要移除 AWS Cloud9 資源，請參閱[刪除環境](#)。

您可以使用 IAM 主控台刪除AWSServiceRoleForAWSCloud 9 服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的[刪除服務連結角色](#)。

AWS Cloud9 服務連結角色的支援區域

AWS Cloud9 支援在所有提供服務的區域中，使用服務連結角色。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [AWS Cloud9](#)。

使用 AWS CloudTrail 記錄 AWS Cloud9 API 呼叫

AWS Cloud9 整合了 CloudTrail，後者是一項可記錄使用者、角色或 AWS 服務在 AWS Cloud9 中所採取的動作的服務。CloudTrail 會將 AWS Cloud9 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 AWS Cloud9 主控台的呼叫，以及對 AWS Cloud9 API 的程式碼呼叫。如果您建立追蹤，就可以將 CloudTrail 事件持續提供給 Amazon Simple Storage Service (Amazon S3) 儲存貯體，包括 AWS Cloud9 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台內的 Event history (事件歷史記錄) 檢視最新事件。您可以利用 CloudTrail 所收集的資訊來判斷向 AWS Cloud9 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [《AWS CloudTrail 使用者指南》](#)。

CloudTrail 中的 AWS Cloud9 資訊


當您建立帳戶時，系統即會在 AWS 帳戶中啟用 CloudTrail。當 AWS Cloud9 中發生活動時，該活動會記錄在 CloudTrail 事件中，其他 AWS 服務事件則記錄於 Event history (事件歷史記錄) 中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷史記錄檢視事件](#)。

如需您 AWS 帳戶中正在進行事件的記錄 (包含 AWS Cloud9 的事件)，請建立追蹤。追蹤能讓 CloudTrail 將日誌檔交付至 Amazon S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案](#)

AWS Cloud9 支援將下列 API 動作記錄為 CloudTrail 日誌檔案中的事件：

- CreateEnvironmentEC2
- CreateEnvironmentSSH
- CreateEnvironmentMembership
- DeleteEnvironment
- DeleteEnvironmentMembership
- DescribeEnvironmentMemberships
- DescribeEnvironments
- DescribeEnvironmentStatus
- ListEnvironments
- ListTagsForResource
- TagResource
- UntagResource
- UpdateEnvironment
- UpdateEnvironmentMembership

 Note

AWS Cloud9 的部分 CloudTrail 事件並不是由公有 API 操作導致。相反地，下列事件會由對使用者驗證和受管臨時憑證有影響的內部更新所啟動：

- DisableManagedCredentialsByCollaborator
- EnvironmentTokenSuccessfullyCreated
- ManagedCredentialsUpdatedOnEnvironment

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 AWS Identity and Access Management IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務 服務提出。

如需詳細資訊，請參閱 [《CloudTrail userIdentity 元素》](#)。

了解 AWS Cloud9 日誌檔案項目

追蹤是一種組態，可讓事件以日誌檔案的形式交付至您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌項目。事件即為來自任何來源的單一請求，其中包含請求動作、動作日期和時間，以及請求參數的相關資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

- [CreateEnvironmentEC2](#)
- [CreateEnvironmentSSH](#)
- [CreateEnvironmentMembership](#)
- [DeleteEnvironment](#)
- [DeleteEnvironmentMembership](#)
- [DescribeEnvironmentMemberships](#)
- [DescribeEnvironments](#)
- [DescribeEnvironmentStatus](#)
- [ListEnvironments](#)
- [ListTagsForResource](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdateEnvironment](#)
- [UpdateEnvironmentMembership](#)

CreateEnvironmentEC2

以下範例顯示的是展示 CreateEnvironmentEC2 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 CreateEnvironmentSSH 動作的 CloudTrail 日誌項目。

```

{
  "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

以下範例顯示的是展示 CreateEnvironmentMembership 動作的 CloudTrail 日誌項目。

```
{
```

```
"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

以下範例顯示的是展示 DeleteEnvironment 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 DeleteEnvironmentMembership 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 DescribeEnvironmentMemberships 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 DescribeEnvironments 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 DescribeEnvironmentStatus 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 ListEnvironments 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 ListTagsForResource 動作的 CloudTrail 日誌項目。

```

{
  "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:3XXXXXXXXX6a4221b7d8e07d9ef6ee21"
  },

```

```
"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

以下範例顯示的是展示 TagResource 動作的 CloudTrail 日誌項目。

```
{
  "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

以下範例顯示的是展示 UntagResource 動作的 CloudTrail 日誌項目。

```

{
    "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:3XXXXXXXXXX6a4221b7d8e07d9ef6ee21",
    "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

以下範例顯示的是展示 UpdateEnvironment 動作的 CloudTrail 日誌項目。

```

{
  "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

以下範例顯示的是展示 UpdateEnvironmentMembership 動作的 CloudTrail 日誌項目。

```

{
  "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"
}
[]}
```

標籤

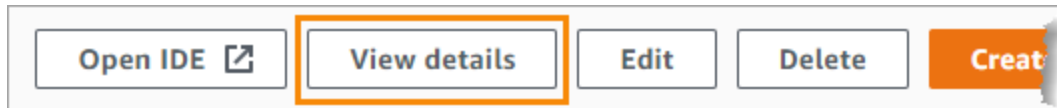
標籤是您或 AWS 附加至 AWS 資源的標籤或屬性。每個標籤皆包含索引鍵與成對的值。您可以使用標籤來控制 AWS Cloud9 資源的存取權，相關說明請參考 [IAM 使用者指南](#) 中的 [使用 AWS 資源標籤控制存取](#)。標籤也可以幫助您管理帳單資訊，如 [使用者定義的成本分配標籤](#) 中所述。

當您 [建立 AWS Cloud9 EC2 開發環境後](#)，AWS Cloud9 會包含管理環境所需的某些系統標籤。系統標籤開頭為「aws:」。在此建立程序期間，您也可以新增自己的資源標籤。

建立環境之後，您可以檢視附加至環境的標籤、將新資源標籤新增至環境，或修改或移除您先前新增的標籤。您最多可以將 50 個使用者定義標籤附加至 AWS Cloud9 環境。

使用下列一或多個方法檢視或更新標籤。

- 在 [AWS Cloud9 主控台](#) 中，選取您感興趣的環境，然後選擇 View Details (檢視詳細資訊)。



- 使用以下 AWS Cloud9 CLI 命令：[list-tags-for-resource](#)、[tag-resource](#) 以及 [untag-resource](#)。
- 使用下列 AWS Cloud9 API 動作：[ListTagsForResourceTagResource](#)、和 [UntagResource](#)。

Warning

您使用上述方法為 AWS Cloud9 建立或更新的標籤不會自動傳播至基礎資源。如需有關如何執行這項操作的資訊，請參閱下一節，[將標籤更新傳播至基礎資源](#)。

將標籤更新傳播至基礎資源

當您使用 AWS Cloud9 CLI 命令或 API 動作以新增、修改或移除附加至 AWS Cloud9 環境的標籤時，這些變更不會自動傳播至基礎資源，例如 AWS CloudFormation 堆疊、Amazon EC2 執行個體，以及 Amazon EC2 安全群組。您必須手動傳播這些變更。

若要更輕鬆地使用下列程序，您可以針對感興趣的環境取得其環境 ID。若您想要執行此操作，請遵循下列步驟：

- 在 [AWS Cloud9 主控台](#) 中，選取所需環境，然後選擇 View Details (檢視詳細資訊)。
- 尋找 Environment ARN (環境 ARN) 屬性並記錄環境 ID，其屬於「environment:」之後的環境 ARN 的一部分。

視您的標籤使用目的而定，您可能需要將標籤更新傳播至以下一或多個位置。

將標籤更新傳播至 AWS CloudFormation 堆疊

Note

將標籤更新至 AWS CloudFormation 堆疊時，這些更新會自動傳播到與堆疊相關聯的 Amazon EC2 執行個體和 Amazon EC2 安全群組。

1. 導覽至 [AWS CloudFormation 主控台](#)。
2. 尋找並選擇與您所需的 AWS Cloud9 環境對應的堆疊。如果您已記錄環境 ID，您可以使用該 ID 來篩選環境。
3. 在 Stack info (堆疊資訊) 索引標籤的 Tags (標籤) 區段中，檢閱標籤清單。
4. 如果您需要更新標籤，請選擇頁面頂端附近的 Update (更新)，並依照指示進行操作。如需詳細資訊，請參閱 [AWS CloudFormation 使用者指南](#) 中的 [直接更新堆疊](#)。

您也可以使用 [describe-stacks](#) 和 [update-stack](#) CLI 命令來更新標籤。

將標籤更新傳播至 Amazon EC2 執行個體

1. 導覽至 [Amazon EC2 主控台](#)。
2. 尋找並選取與您所需的 AWS Cloud9 環境的對應的 Amazon EC2 執行個體。如果您先前已記錄環境 ID，您可以使用該 ID 來篩選環境。
3. 在 Tags (標籤) 索引標籤上，視需要檢視及更新標籤。

您也可以使用 [describe-tags](#)、[create-tags](#) 以及 [delete-tags](#) CLI 命令來更新標籤。

將標籤更新傳播至 Amazon EC2 安全群組

1. 導覽至 [Amazon EC2 安全群組 主控台](#)。
2. 尋找並選取與您所需的 AWS Cloud9 環境對應的安全群組。如果您先前已記錄環境 ID，您可以使用該 ID 來篩選環境。
3. 開啟 Tags (標籤) 索引標籤，視需要檢視及更新標籤。

您也可以使用 [describe-tags](#)、[create-tags](#) 以及 [delete-tags](#) CLI 命令來更新標籤。

安全性 AWS Cloud9

雲端安全是 Amazon Web Services (AWS) 最重視的一環。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。安全是 AWS 與您之間共同承擔的責任。[共同責任模型](#) 將此描述為雲端本身的安全和雲端內部的安全。

雲的安全性 — AWS 負責保護運行 AWS 雲中提供的所有服務的基礎設施，並為您提供可以安全使用的服務。我們的安全責任是我們的首要任務 AWS，並且我們的安全性有效性是由第三方審計師定期測試和驗證，作為[AWS 合規計劃](#)的一部分。

雲端安全性 — 您的責任取決於您使用的 AWS 服務，以及其他因素，包括資料的敏感性、組織的需求，以及適用的法律和法規。

AWS Cloud9 透過其支援的特定 AWS 服務，遵循[共同的責任模式](#)。如需 AWS 服務安全性資訊，請參閱[AWS 服務安全性說明文件頁面](#)和符合性計劃 [AWS 遵循工作範圍的 AWS 服務](#)。

下列主題說明如何設定 AWS Cloud9 以符合安全性與合規性目標。

主題

- [資料保護 AWS Cloud9](#)
- [的 Identity and Access Management AWS Cloud9](#)
- [登錄和監控 AWS Cloud9](#)
- [符合性驗證 AWS Cloud9](#)
- [韌性 AWS Cloud9](#)
- [基礎結構安全 AWS Cloud9](#)
- [軟體更新與修補](#)
- [安全性最佳做法 AWS Cloud9](#)

資料保護 AWS Cloud9

AWS [共用責任模型](#)適用於中的資料保護 AWS Cloud9。如此模型中所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API AWS Cloud9 或 AWS SDK 時 AWS 服務 使用或其他使用時。AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

資料加密是指在傳輸中 (在資料 AWS Cloud9 與您的 AWS 帳戶之間傳輸時) 和靜態 (同時儲存在 AWS Cloud9 組態存放區和 AWS 雲端運算執行個體中) 時保護資料。

在的內容中 AWS Cloud9，下列類型的資料可能需要透過加密進行保護：

您的內容與資料

您運用、收集和存放的資訊。以下是這種資料類型的範例：

- 您的程式碼檔案
- 連接的 EC2 環境或 SSH 環境的組態、應用程式和資料

AWS Cloud9 中繼資料

AWS Cloud9 操作、收集和儲存的資料。以下是這種資料類型的範例：

- IDE 設定，例如標籤狀態、開啟的檔案，以及 IDE 偏好設定
- AWS Cloud9 開發環境中繼資料，例如環境名稱和說明
- AWS Cloud9 服務 API 和控制台日誌

- 服務日誌，例如 HTTP 請求

AWS Cloud9 也會透過其資料層服務傳輸部分內容和資料。這包括您的檔案、終端機、輸出文字，以及一些 IDE 命令 (例如儲存檔案)。

靜態加密

靜態加密是指在存放時對資料進行加密，以保護您的資料免受未經授權的存取。儲存在 AWS Cloud9 環境中的任何客戶資料，例如程式碼檔案、套件或相依性，一律會儲存在客戶的資源中。如果客戶使用 Amazon EC2 環境，資料會存放在其 AWS 帳戶中相關聯的 Amazon Elastic Block Store (Amazon EBS) 磁碟區中。如果客戶使用 SSH 環境，資料會存放在其 Linux 伺服器上的本機儲存體中。

為 AWS Cloud9 開發環境建立 Amazon EC2 執行個體時，會建立未加密的 Amazon EBS 磁碟區並將其連接到該執行個體。想要加密資料的客戶需要建立加密的 EBS 磁碟區，並將其連接至 EC2 執行個體。AWS Cloud9 並附加的 Amazon EBS 磁碟區支援 Amazon EBS 預設加密，此加密預設為區域特定的設定。如需詳細資訊，請參閱 AWS Elastic Compute Cloud 使用者指南中的[預設加密](#)。

有關 AWS Cloud9 開發環境的中繼資料 (例如環境名稱、環境成員和 IDE 設定) 會儲存在客戶資源中 AWS，而不是儲存在客戶資源中。客戶特定資訊 (例如環境說明和 IDE 設定) 會經過加密。

傳輸中加密

傳輸中的加密指的是保護您的資料免於在通訊端點間移動時遭到攔截。客戶端和 AWS Cloud9 服務之間傳輸的所有數據都通過 HTTPS，WSS 和加密的 SSH 進行加密。

- HTTPS — 確保客戶的 Web 瀏覽器和 AWS Cloud9 服務之間的安全請求。AWS Cloud9 此外，還會從 Amazon 從客戶的瀏覽器透過 HTTPS CloudFront 傳送的資產載入。
- WSS (WebSocket 安全) — 在客戶的 Web 瀏覽器和 AWS Cloud9 服務 WebSockets 之間啟用安全的雙向通訊。
- 加密 SSH (安全殼層)：允許客戶端的 Web 瀏覽器和 AWS Cloud9 服務之間的數據的安全傳輸。

HTTPS、WSS 和 SSH 通訊協定的使用取決於您使用支援的瀏覽器。AWS Cloud9請參閱[AWS Cloud9 支援的瀏覽器](#)。

Note

根據預設，加密通訊協定會在 AWS Cloud9中實作。客戶無法變更 encryption-in-transit 設定。

金鑰管理

AWS Key Management Service (AWS KMS) 是一種用於創建和控制的託管服務 AWS KMS keys , 用於加密客戶數據的加密密鑰。 AWS Cloud9 生成和管理加密密鑰 , 以代表客戶加密數據。

網際網路流量隱私權

SSH 環境會連線到內部部署的客戶自有運算和儲存體。加密 SSH、HTTPS 和 WSS 連線支援在服務和 SSH 環境間傳輸資料。

您可以設定要在特定 VPC 和子網路中啟動的 AWS Cloud9 EC2 開發環境 (由 Amazon EC2 執行個體支援)。如需 Amazon Virtual Private Cloud 設定的詳細資訊 , 請參閱 [AWS Cloud9 開發環境的 VPC 設定](#)。

的 Identity and Access Management AWS Cloud9

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。 AWS 服務 IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有權限) 來使用 AWS Cloud9 資源。 IAM 是您可以使用的 AWS 服務 , 無需額外付費。

主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [如何與 IAM AWS Cloud9 搭配使用](#)
- [AWS Cloud9的身分型政策範例](#)
- [疑難排解 AWS Cloud9 身分和存取](#)
- [如何使 AWS Cloud9 用 IAM 資源和作業](#)
- [AWS 受管理的政策 AWS Cloud9](#)
- [建立客戶管理的政策 AWS Cloud9](#)
- [AWS Cloud9 權限參考](#)
- [AWS 受管理的臨時憑](#)

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 AWS Cloud9。

服務使用者 — 如果您使用 AWS Cloud9 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS Cloud9 功能來完成工作時，您可能需要其他權限。了解存取的管理方式可協助您向管理員請求正確的許可。若您無法存取 AWS Cloud9 中的某項功能，請參閱 [疑難排解 AWS Cloud9 身分和存取](#)。

服務管理員 — 如果您負責公司的 AWS Cloud9 資源，您可能擁有完整的存取權 AWS Cloud9。決定您的服務使用者應該存取哪些 AWS Cloud9 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM AWS Cloud9，請參閱 [如何與 IAM AWS Cloud9 搭配使用](#)。

IAM 管理員：如果您是 IAM 管理員，建議您掌握如何撰寫政策以管理 AWS Cloud9 存取權的詳細資訊。若要檢視可在 IAM 中使用的 AWS Cloud9 基於身分的政策範例，請參閱 [AWS Cloud9 的身分型政策範例](#)

使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中的 [如何登入](#) 您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的 [簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [多重要素驗證](#) 和 IAM 使用者指南中的 [在 AWS 中使用多重要素驗證 \(MFA\)](#)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的[需要根使用者憑證的任務](#)。

聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 AWS IAM Identity Center 使用者指南中的[什麼是 IAM Identity Center ?](#)。

IAM 使用者和群組

[IAM 使用者](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以[切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或

AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 [IAM 使用者指南](#) 中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的[許可集](#)。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的[IAM 角色與資源類型政策的差異](#)。
- 跨服務訪問 — 有些 AWS 服務使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
 - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務向下游服務發出要求。只有當服務收到需要與其 AWS 服務他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。
- 服務角色 – 服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的[建立角色以委派許可給 AWS 服務](#)。
- 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的[建立 IAM 角色 \(而非使用者\) 的時機](#)。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的[JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的[建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的[在受管政策和內嵌政策間選擇](#)。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF 若要進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的 [存取控制清單 \(ACL\) 概觀](#)。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 實體許可範圍](#)。
- 服務控制策略 (SCP) — SCP 是 JSON 策略，用於指定中組織或組織單位 (OU) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制策略 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 [SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的 [工作階段政策](#)。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的 [政策評估邏輯](#)。

如何與 IAM AWS Cloud9 搭配使用

在您使用 IAM 管理存取權限之前 AWS Cloud9，請先了解哪些 IAM 功能可搭配使用 AWS Cloud9。

您可以搭配使用的 IAM 功能 AWS Cloud9

IAM 功能	AWS Cloud9 支持
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
轉送存取工作階段 (FAS)	是
服務角色	是
服務連結角色	是

若要深入瞭解如何以 AWS Cloud9 及其他 AWS 服務如何使用大多數 IAM 功能，請參閱 IAM 使用者指南中的搭配 IAM 使用的[AWS 服務](#)。

以身分識別為基礎的原則 AWS Cloud9

支援身分型政策	是
---------	---

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。

以身分識別為基礎的原則範例 AWS Cloud9

若要檢視以 AWS Cloud9 身分為基礎的原則範例，請參閱 [AWS Cloud9 的身分型政策範例](#)

以資源為基礎的政策 AWS Cloud9

支援以資源基礎的政策

否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 角色與資源型政策有何差異](#)。

AWS Cloud9 不支援以資源為基礎的政策，但您仍然可以透過 AWS Cloud9 API 和 AWS Cloud9 IDE 控制 AWS Cloud9 AWS Cloud9 環境成員的環境資源權限。

的政策動作 AWS Cloud9

支援政策動作

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS Cloud9 動作清單，請參閱服務授權參考 AWS Cloud9 中 [所定義的動作](#)。

中的策略動作在動作之前 AWS Cloud9 使用下列前置詞：

```
account
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "account:action1",  
  "account:action2"  
]
```

若要檢視以 AWS Cloud9 身為基礎的原則範例，請參閱 [AWS Cloud9 的身分型政策範例](#)

的政策資源 AWS Cloud9

支援政策資源

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 AWS Cloud9 資源類型及其 ARN 的清單，請參閱服務授權參考 AWS Cloud9 中 [所定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [AWS Cloud9 定義的動作](#)。

若要檢視以 AWS Cloud9 身為基礎的原則範例，請參閱 [AWS Cloud9 的身分型政策範例](#)

的政策條件索引鍵 AWS Cloud9

支援服務特定政策條件金鑰 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的[IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

若要查看 AWS Cloud9 條件索引鍵清單，請參閱服務授權參考 AWS Cloud9 中的[條件金鑰](#)。若要瞭解您可以使用條件索引鍵的動作和資源，請參閱[定義的動作 AWS Cloud9](#)。

若要檢視以 AWS Cloud9 身為基礎的原則範例，請參閱。[AWS Cloud9 的身分型政策範例](#)

ACL 位於 AWS Cloud9

支援 ACL 否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

阿巴克與 AWS Cloud9

支援 ABAC (政策中的標籤) 是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件金鑰，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的 [什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的 [使用屬性型存取控制 \(ABAC\)](#)。

使用臨時登入資料 AWS Cloud9

支援臨時憑證 是

當您使用臨時憑據登錄時，某些 AWS 服務不起作用。如需其他資訊，包括哪些 AWS 服務與臨時登入資料 [搭配 AWS 服務使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的 [切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 [IAM 中的暫時性安全憑證](#)。

轉寄存取工作階段 AWS Cloud9

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，

才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

AWS Cloud9的服務角色

支援服務角色	是
--------	---

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務](#)。

Warning

變更服務角色的權限可能會中斷 AWS Cloud9 功能。只有在 AWS Cloud9 提供指引時才編輯服務角色。

服務連結角色 AWS Cloud9

支援服務連結角色	是
----------	---

服務連結角色是一種連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服务連結角色的詳細資訊，請參閱 [可搭配 IAM 運作的 AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇 Yes (是) 連結，以檢視該服務的服務連結角色文件。

AWS Cloud9的身分型政策範例

根據預設，使用者和角色不具備建立或修改 AWS Cloud9 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的 [建立 IAM 政策](#)。

如需有關由定義的動作和資源類型的詳細資訊 AWS Cloud9，包括每個資源類型的 ARN 格式，請參閱服務授權參考 AWS Cloud9 中的動作、資源和條件索引 [鍵](#)。

主題

- [政策最佳實務](#)
- [使用 AWS Cloud9 主控台](#)
- [允許使用者檢視他們自己的許可](#)

政策最佳實務

以身分識別為基礎的政策會決定某人是否可以建立、存取或刪除您帳戶中的 AWS Cloud9 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始授與使用者和工作負載的權限，請使用可授與許多常見使用案例權限的 AWS 受管理原則。它們可用在您的 AWS 帳戶。建議您透過定義特定於您的使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。

使用 AWS Cloud9 主控台

若要存取 AWS Cloud9 主控台，您必須擁有最少一組權限。這些權限必須允許您列出和檢視有關 AWS 帳戶。AWS Cloud9 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保使用者和角色仍可使用 AWS Cloud9 主控台，請同時將 AWS Cloud9 *ConsoleAccess* 或受 *ReadOnly* AWS 管理的原則附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
  "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": "*"
    }
  ]
}
```

```
}
```

疑難排解 AWS Cloud9 身分和存取

使用下列資訊可協助您診斷和修正使用和 IAM 時可能會遇到的 AWS Cloud9 常見問題。

主題

- [我沒有執行操作的授權 AWS Cloud9](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想允許我以外的人訪 AWS 帳戶 問我的 AWS Cloud9 資源](#)

我沒有執行操作的授權 AWS Cloud9

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 `awes:GetWidget` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
awes:GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 `awes:GetWidget` 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我沒有授權執行 iam : PassRole

如果您收到錯誤，告知您未獲授權執行 `iam:PassRole` 動作，您的政策必須更新，允許您將角色傳遞給 AWS Cloud9。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

名為 marymajor 的 IAM 使用者嘗試使用主控台在 AWS Cloud9中執行動作時，發生下列範例錯誤。但是，動作要求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我想允許我以外的人訪 AWS 帳戶 問我的 AWS Cloud9 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解是否 AWS Cloud9 支援這些功能，請參閱[如何與 IAM AWS Cloud9 搭配使用](#)。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 [IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限](#)。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的[提供第三方 AWS 帳戶 擁有的存取權](#)。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的[將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 [IAM 角色 與資源型政策的差異](#)。

如何使 AWS Cloud9 用 IAM 資源和作業

AWS Identity and Access Management 用於管理允許您同時處理 AWS Cloud9 開發環境和其他 AWS 服務 和資源的權限。

AWS Cloud9 資源與營運

在中 AWS Cloud9，主要資源是 AWS Cloud9 開發環境。在政策中，您使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。下表列有環境 ARN。如需詳細資訊，請參閱《Amazon Web Services 一般參考》中的 [Amazon Resource Name \(ARN\) 與 AWS 服務命名空間](#)。

資源類型	ARN 格式
環境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i></code>
指定 AWS 區域中的指定帳戶擁有的每個環境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :environment:*</code>
指定區域中的指定帳戶擁有的每個環境	<code>arn:aws:cloud9: <i>REGION_ID</i> :<i>ACCOUNT_ID</i> :*</code>
每個 AWS Cloud9 資源，無論帳號和區域為何	<code>arn:aws:cloud9:*</code>

例如，您可以在陳述式中使用環境的 Amazon Resource Name (ARN) 指定特定環境，如下所示。

```
"Resource": "arn:aws:cloud9:us-east-2:123456789012:environment:70d899206236474f9590d93b7c41dfEX"
```

若要指定所有資源，請在 * 元素中使用萬用字元 (Resource)。

```
"Resource": "*"
```

若要在單一陳述式中指定多項資源，請使用逗號分隔其 Amazon Resource Name (ARN)。

```
"Resource": [
  "arn:aws:cloud9:us-east-2:123456789012:environment:70d899206236474f9590d93b7c41dfEX",
  "arn:aws:cloud9:us-east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
]
```

AWS Cloud9 提供了一組操作來處理 AWS Cloud9 資源。如需清單，請參閱 [AWS Cloud9 權限參考](#)。

了解資源所有權

AWS 帳戶 帳號擁有在帳號中建立的資源，無論是誰建立資源。

請考慮下列使用案例：

- 假設您使用的根帳戶憑據 AWS 帳戶 來創建一個 AWS Cloud9 開發環境。雖然可能，但不建議這樣做。在這種情況下，您 AWS 帳戶 是環境的所有者。
- 假設您在中建立 IAM 使用者，AWS 帳戶 並授與建立環境的權限給該使用者。然後，使用者可以建立一個環境。但是 AWS 帳戶，您的使用者所屬，仍然擁有該環境。
- 假設您在 AWS 帳戶 具有權限的中建立 IAM 角色以建立環境。然後，可以擔任該角色的任何人都能建立環境。您的 AWS 帳戶(即該角色所屬的帳戶) 擁有環境。

Note

如果您刪除身為一或多個 AWS Cloud9 環境之 ARN 擁有者的使用者帳戶，則這些環境將沒有擁有者。此案例的解決方法是使用 AWS Cloud9 SDK 新增另一個具有讀取和寫入權限的 IAM 使用者，使用 `CreateEnvironmentMembership` 動作和 `EnvironmentMember` 資料類型。新增此 IAM 使用者之後，您可以將環境檔案複製到新 AWS Cloud9 環境，並將此擁有者設為 ARN 擁有者。如需有關此動作的詳細資訊，請參閱 [CreateEnvironmentMembership](#)，以取得有關此資料類型的詳細資訊，請參閱《AWS Cloud9 API 參考指南》[EnvironmentMember](#) 中的。

管理資源存取

許可政策說明誰可以存取哪些資源。

Note

本節討論如何在 AWS Cloud9 中使用 IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱 IAM 使用者指南中的 [什麼是 IAM?](#)。如需有關 IAM 政策語法和說明的資訊，請參閱 IAM 使用者指南中的 [IAM JSON 政策參考](#)。

連接到 IAM 身分的政策稱為 身分型政策 (或 IAM 政策)。附加至資源的策略稱為以資源為基礎的策略。AWS Cloud9 支援以身分識別為基礎和以資源為基礎的政策。

以下每個 API 動作只需要將 IAM 政策連接到希望呼叫這些 API 動作的 IAM 身分：

- `CreateEnvironmentEC2`
- `DescribeEnvironments`

下列 API 動作需要以資源為基礎的政策。不需要 IAM 政策，但如果 IAM 政策附加到想要呼叫這些 API 動作的 IAM 身分，則 AWS Cloud9 會使用 IAM 政策。以資源為基礎的策略必須套用至所需的 AWS Cloud9 資源：

- CreateEnvironmentMembership
- DeleteEnvironment
- DeleteEnvironmentMembership
- DescribeEnvironmentMemberships
- DescribeEnvironmentStatus
- UpdateEnvironment
- UpdateEnvironmentMembership

如需其中每個 API 動作執行內容的詳細資訊，請參閱 [AWS Cloud9 API 參考](#)。

您無法將以資源為基礎的策略直接附加到 AWS Cloud9 資源。而是在您新增、修改、更新或刪除環境成員時，AWS Cloud9 將適當的以 AWS Cloud9 資源為基礎的原則附加至資源。

若要授與使用者對 AWS Cloud9 資源執行動作的權限，您可以將許可政策附加至使用者所屬的 IAM 群組。我們建議您盡可能附加 AWS 受管理的 (預先定義) 政策。AWS Cloud9 AWS 受管理的原則包含一般使用案例和使用者類型的預先定義存取權限集合，例如環境的完整管理、環境使用者，以及只有環境唯讀存取權限的使用者。如需的 AWS 受管理策略清單 AWS Cloud9，請參閱 [AWS 受管理的政策 AWS Cloud9](#)。

如需更詳細的使用案例和獨特使用者類型，您可以建立並連接自己的客戶受管政策。請參閱 [AWS Cloud9 的其他設定選項 \(小組和企業\)](#) 和 [建立客戶管理的政策 AWS Cloud9](#)。

若要將 IAM 政策 (AWS 受管或客戶管理) 附加至 IAM 身分，請參閱 [IAM 使用者指南中的附加 IAM 政策 \(主控台\)](#)。

API 操作的工作階段許可

使用 AWS CLI 或 AWS API 以程式設計方式為角色或同盟使用者建立暫時工作階段時，您可以將工作階段原則當做參數傳遞，以擴充角色工作階段的範圍。也就是說，工作階段的有效許可會是 [角色的身分類型政策和工作階段政策的交集](#)。

在工作階段期間發出存取資源的請求時，如果工作階段政策中沒有適用的 Deny 陳述式，也沒有適用的 Allow 陳述式，則政策評估的結果會是 [隱含拒絕](#)。(如需詳細資訊，請參閱 IAM 使用者指南中的 [判斷帳戶中的請求獲得允許或遭到拒絕](#)。)

但是，對於需要以資源為基礎的政策（請參閱上文）的 AWS Cloud9 API 操作，如果在資源策略中將其指定為，則會將許可授予給正Principal在呼叫的 IAM 實體。此明確權限的優先順序高於對工作階段原則的隱含拒絕，因此可讓工作階段成功呼叫 AWS Cloud9 API 作業。

AWS 受管理的政策 AWS Cloud9

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住，AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的[客戶管理政策](#)，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別（使用者、群組和角色）。AWS 當新的啟動或新 AWS 服務的 API 操作可用於現有服務時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的[AWS 受管政策](#)。

AWS 受管理的策略：AWSCloud9Administrator

您可將 AWSCloud9Administrator 政策連接到 IAM 身分。

AWS Cloud9#

許可詳細資訊

此政策包含以下許可。

- AWS Cloud9 — 在他們的所有 AWS Cloud9 行動 AWS 帳戶。
- Amazon EC2 — 取得其 AWS 帳戶中多個 Amazon VPC 和子網路資源的相關資訊。
- IAM — 取得 IAM 使用者的相關資訊 AWS 帳戶，並視需要在其 AWS 帳戶 中建立 AWS Cloud9 服務連結角色。
- Systems Manager：允許使用者呼叫 StartSession 以啟動工作階段管理員工作階段之執行個體的連線。使用者如果要透過 Systems Manager 開啟與 EC2 執行個體通訊的環境，就需要此許可。如需詳細資訊，請參閱[使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)

```
{
  "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 受管理的策略：AWSCloud9User

您可將 AWSCloud9User 政策連接到 IAM 身分。

此政策向###授予建立 AWS Cloud9 開發環境並管理自有環境的許可。

許可詳細資訊

此政策包含以下許可。

- AWS Cloud9 — 建立並取得其環境的相關資訊，以及取得和變更其環境的使用者設定。
- Amazon EC2 — 取得其 AWS 帳戶中多個 Amazon VPC 和子網路資源的相關資訊。
- IAM — 取得 IAM 使用者的相關資訊 AWS 帳戶，並視需要在其 AWS 帳戶 中建立 AWS Cloud9 服務連結角色。
- Systems Manager：允許使用者呼叫 StartSession 以啟動工作階段管理員工作階段之執行個體的連線。使用者如果要透過 Systems Manager 開啟與 EC2 執行個體通訊的環境，就需要此許可。如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)

```

{
  "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 受管理的策略：AWSCloud9EnvironmentMember

您可將 AWSCloud9EnvironmentMember 政策連接到 IAM 身分。

此原則#授與提供加入 AWS Cloud9 共用環境之能力的成員資格權限。

許可詳細資訊

此政策包含以下許可：

- AWS Cloud9 — 取得有關其環境的資訊，以及取得和變更其環境的使用者設定。
- IAM — 取得 IAM 使用者的相關資訊 AWS 帳戶，並視需要在其 AWS 帳戶 中建立 AWS Cloud9 服務連結角色。
- Systems Manager：允許使用者呼叫 StartSession 以啟動工作階段管理員工作階段之執行個體的連線。使用者如果要透過 Systems Manager 開啟與 EC2 執行個體通訊的環境，就需要此許可。如需詳細資訊，請參閱 [使用 AWS Systems Manager 存取無輸入的 EC2 執行個體](#)

```
{
  "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 受管理的策略：AWSCloud9ServiceRolePolicy

服務連結角色 AWSServiceRoleForAWSCloud9 使用此政策允許 AWS Cloud9 環境與 Amazon EC2 和 AWS CloudFormation 資源互動。

許可詳細資訊

AWSCloud9ServiceRolePolicy 授予 AWSServiceRoleForAWSCloud 9 個必要的許可，以 AWS Cloud9 允許與建立和執行開發環境所需的 AWS 服務 (Amazon EC2 和 AWS CloudFormation) 互動。

AWS Cloud9 定義其服務連結角色的權限，而且只 AWS Cloud9 能擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

如需如何 AWS Cloud9 使用服務連結角色的詳細資訊，請參閱 [使用 AWS Cloud9 的服務連結角色](#)。

```

{
  "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"
    }
  }
}
]
}

```

AWS Cloud9 AWS 受管理策略的更新

檢視 AWS Cloud9 自此服務開始追蹤這些變更以來的 AWS 受管理策略更新詳細資料。如需有關此頁面變更的自動警示，請訂閱「AWS Cloud9 文件歷史記錄」頁面上的 RSS 摘要。

變更	描述	日期
新動作已新增至AWSCloud9 UserAWSCloud9Administrator和AWSCloud9EnvironmentMember原則。	動作ssm:GetConnectionStatus 已新增至AWSCloud9 UserAWSCloud9Administrator和AWSCloud9EnvironmentMember策略。此動作將授與使用者檢查 SSM 連線狀態的權限。cloud9:ValidateEnvironmentName API 已從AWSCloud9 User政策中移除，因為它已被取代。	2023 年 10 月 12 日
API 已新增至AWSCloud9 User和AWSCloud9Administrator政策。	兩個新的 API 已添加到AWSCloud9 User和AWSCloud9 Administrator策略中，這些 API 的是ec2:DescribeInstanceTypeOfferings 和ec2:DescribeRouteTables 。這些	2023年8月02 日

變更	描述	日期
	API 的目的是 AWS Cloud9 要驗證預設子網路是否支援客戶在建立 AWS Cloud9 環境時選擇的執行個體類型。	
更新至 AWSCloud9ServiceRolePolicy	AWSCloud9ServiceRolePolicy 已更新，AWS Cloud9 允許啟動和停止由 License Manager 授權組態管理的 Amazon EC2 執行個體。	2022 年 1 月 12 日
AWS Cloud9 開始追蹤變更	AWS Cloud9 開始追蹤其 AWS 受管理策略的變更。	2021 年 3 月 15 日

建立客戶管理的政策 AWS Cloud9

如果所有 AWS 受管政策都不符合您的存取控制需求，您可以建立並附加自己的客戶管理政策。

若要建立客戶受管政策，請參閱 IAM 使用者指南中的 [建立 IAM 政策 \(主控台\)](#)。

主題

- [指定政策元素：效果、委託人、動作和資源](#)
- [客戶受管政策範例](#)

指定政策元素：效果、委託人、動作和資源

服務會針對每個 AWS Cloud9 資源定義一組 API 作業。若要授與這些 API 作業的權限，請 AWS Cloud9 定義一組您可以在政策中指定的動作。

以下是基本的政策元素：

- **Effect** - 您可以指定使用者請求動作時會有什麼效果 (允許或拒絕)。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕存取資源。如要確保即使在其他政策授予存取權時，使用者仍無法存取資源，您可以採取此動作。
- **Principal** - 在以身分為基礎的政策 (IAM 政策) 中，政策連接的使用者是隱含主體。對於以資源為基礎的政策，您可以指定希望獲得許可的使用者、帳戶、服務或其他實體。

- Resource - 使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。
- Action - 使用動作關鍵字來識別要允許或拒絕的資源操作。例如，cloud9:CreateEnvironmentEC2 許可授予使用者執行 CreateEnvironmentEC2 操作的許可。

如需進一步了解 IAM 政策語法和說明，請參閱 IAM 使用者指南中的 [IAM JSON 政策參考](#)。

如需顯示所有 AWS Cloud9 API 動作及其套用資源的表格，請參閱 [AWS Cloud9 權限參考](#)。

客戶受管政策範例

在本節中，您可以找到授予 AWS Cloud9 動作許可的政策範例。您可以調整下列 IAM 政策範例來允許或明確拒絕 IAM 身分的 AWS Cloud9 存取。

若要建立客戶受管政策或將其連接到 IAM 身分，請參閱 IAM 使用者指南中的 [建立 IAM 政策 \(主控台\)](#) 和 [連接 IAM 政策 \(主控台\)](#)。

Note

下列範例使用美國東部 (俄亥俄 us-east-2) 區域 ()、虛構 AWS 帳戶 ID (123456789012) 和虛構的 AWS Cloud9 開發環境 ID ()。81e900317347585a0601e04c8d52eaEX

主題

- [取得環境的相關資訊](#)
- [建立 EC2 環境](#)
- [建立具有特定 Amazon EC2 執行個體類型的 EC2 環境](#)
- [在特定 Amazon VPC 子網路中建立 EC2 環境](#)
- [建立具有特定環境名稱的 EC2 環境](#)
- [只建立 SSH 環境](#)
- [更新環境或防止更新環境](#)
- [取得環境成員清單](#)
- [只與特定使用者共享環境](#)
- [防止共享環境](#)
- [變更或防止變更環境成員的設定](#)

- [移除或防止移除環境成員](#)
- [刪除或防止刪除環境](#)
- [建立 SSM 環境的自訂 IAM 政策](#)

取得環境的相關資訊

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例讓該實體取得其帳戶中任何環境的相關資訊。

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

Note

先前的存取權限已包含在 AWS 受管理的策略 `AWSCloud9Administrator` 和 `AWSCloud9User` 中。

建立 EC2 環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體在其帳戶中建立 AWS Cloud9 EC2 開發環境。

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

Note

先前的存取權限已包含在 AWS 受管理的策略 `AWSCloud9Administrator` 和中 `AWSCloud9User`。

建立具有特定 Amazon EC2 執行個體類型的 EC2 環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體在其帳戶中建立 AWS Cloud9 EC2 開發環境。不過，EC2 環境只能使用指定的 Amazon EC2 執行個體類型類別。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "cloud9:InstanceType": "t3.*"
        }
      }
    }
  ]
}
```

Note

如果受 AWS 管政策 `AWSCloud9Administrator` 或 `AWSCloud9User` 已附加至 IAM 實體，則該 AWS 受管政策會覆寫前述 IAM 政策聲明的行為。這是因為這些 AWS 受管理的政策較寬鬆。

在特定 Amazon VPC 子網路中建立 EC2 環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體在其帳戶中建立 AWS Cloud9 EC2 開發環境。不過，EC2 環境只能使用指定的 Amazon VPC 子網路。

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "cloud9:CreateEnvironmentEC2",
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "cloud9:SubnetId": [
          "subnet-12345678",
          "subnet-23456789"
        ]
      }
    }
  }
]
}

```

Note

如果受 AWS 管政策 `AWSCloud9Administrator` 或 `AWSCloud9User` 已附加至 IAM 實體，則該 AWS 受管政策會覆寫前述 IAM 政策聲明的行為。這是因為這些 AWS 受管理的政策較寬鬆。

建立具有特定環境名稱的 EC2 環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體在其帳戶中建立 AWS Cloud9 EC2 開發環境。不過，EC2 環境只能使用指定的名稱。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:CreateEnvironmentEC2",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloud9:EnvironmentName": "my-demo-environment"
        }
      }
    }
  ]
}

```



```

    }
  ]
}
```

Note

如果受 AWS 管政策 `AWSCloud9Administrator` 或 `AWSCloud9User` 已附加至 IAM 實體，則該 AWS 受管政策會覆寫前述 IAM 政策聲明的行為。這是因為這些 AWS 受管理的政策較寬鬆。

只建立 SSH 環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體在其帳戶中建立 AWS Cloud9 SSH 開發環境。但是，實體無法建立 AWS Cloud9 EC2 開發環境。

```

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

更新環境或防止更新環境

下列附加至 IAM 實體的 IAM 政策陳述式範例可讓該實體變更其帳戶中任何 AWS Cloud9 開發環境的相關資訊。

```

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

```

    "Effect": "Allow",
    "Action": "cloud9:UpdateEnvironment",
    "Resource": "*"
  }
]
}

```

Note

AWS 受管理策略中已包含先前的存取權限AWSCloud9Administrator。

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例明確防止該實體變更具有指定 Amazon Resource Name (ARN) 的環境相關資訊。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:UpdateEnvironment",
      "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}

```

取得環境成員清單

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可讓該實體取得帳戶中任何環境的成員清單。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:DescribeEnvironmentMemberships",
      "Resource": "*"
    }
  ]
}

```

Note

AWS 受管理策略中已包含先前的存取權限 `AWSCloud9Administrator`。此外，先前的存取權限比 AWS 受管理原則中的對等存取權限更為寬鬆。 `AWSCloud9User`

只與特定使用者共享環境

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例讓該實體只能與指定使用者共用帳戶中的任何環境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloud9:CreateEnvironmentMembership"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloud9:UserArn": "arn:aws:iam::123456789012:user/MyDemoUser"
        }
      }
    }
  ]
}
```

Note

如果受 AWS 管政策 `AWSCloud9Administrator` 或 `AWSCloud9User` 已附加至 IAM 實體，則這些 AWS 受管政策會覆寫前述 IAM 政策聲明的行為。這是因為這些 AWS 受管理的政策較寬鬆。

防止共享環境

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例會防止該實體共用帳戶中的任何環境。

```
{
```

```

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

```

變更或防止變更環境成員的設定

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可讓該實體變更帳戶中任何環境裡的成員設定。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:UpdateEnvironmentMembership",
      "Resource": "*"
    }
  ]
}

```

Note

AWS 受管理策略中已包含先前的存取權限 `AWSCloud9Administrator`。

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可明確防止該實體在具有指定 Amazon Resource Name (ARN) 的環境中變更成員設定。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",

```

```

    "Action": "cloud9:UpdateEnvironmentMembership",
    "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
  }
]
}

```

移除或防止移除環境成員

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可讓該實體移除帳戶中任何環境裡的成員。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloud9:DeleteEnvironmentMembership",
      "Resource": "*"
    }
  ]
}

```

Note

AWS 受管理策略中已包含先前的存取權限AWSCloud9Administrator。

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可明確防止該實體在具有指定 Amazon Resource Name (ARN) 的環境中移除任何成員。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:DeleteEnvironmentMembership",
      "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}

```

刪除或防止刪除環境

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可讓該實體刪除帳戶中的任何環境。

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

Note

AWS 受管理策略中已包含先前的存取權限AWSCloud9Administrator。

下列是連接到 IAM 實體的 IAM 政策陳述式範例，此範例可明確防止該實體刪除具有指定 Amazon Resource Name (ARN) 的環境。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "cloud9:DeleteEnvironment",
      "Resource": "arn:aws:cloud9:us-
east-2:123456789012:environment:81e900317347585a0601e04c8d52eaEX"
    }
  ]
}
```

建立 SSM 環境的自訂 IAM 政策

當建立附加AWSCloud9Administrator或AWSCloud9User原則的 SSM 環境時，會發生目前的權限問題。下列 IAM 政策陳述式範例連接至 IAM 實體時，可讓使用者附加和使用受 AWS 管政策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 權限參考

您可以在 AWS Cloud9 原則中使用 AWS 寬條件金鑰來表示條件。如需取得清單，請參閱 IAM 使用者指南中的 [IAM JSON 政策元素：條件](#)。

您可以在政策的 Action 欄位中指定動作。若要指定動作，請使用後接 API 操作名稱的 cloud9: 字首 (例如，"Action": "cloud9:DescribeEnvironments")。若要在單一陳述式中指定多個動作，請用逗號加以分隔 (例如 "Action": ["cloud9:UpdateEnvironment", "cloud9:DeleteEnvironment"])。

使用萬用字元

您可以使用或不使用萬用字元 (*)，指定 ARN 做為政策之 Resource 欄位中的資源值。您可以使用萬用字元指定多個動作或資源。例如，cloud9:* 指定所有 AWS Cloud9 動作，並 cloud9:Describe* 指定所有以開頭的 AWS Cloud9 動作 Describe。

下列範例可讓 IAM 實體取得其帳戶中任何環境與環境成員資格的相關資訊。

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

Note

AWS 受管理策略中已包含先前的存取權限 `AWSCloud9Administrator`。此外，先前的存取權限比 AWS 受管理原則中的對等存取權限更寬鬆。 `AWSCloud9User`

AWS Cloud9 API 操作和動作所需的權限

Note

當您設定存取控制及撰寫可連接到 IAM 身分的許可政策 (身分類型政策) 時，您可以使用下表做為參考。

[Public API operations](#) 資料表列出了可由使用 SDK 和 AWS Command Line Interface 的客戶呼叫的 API 操作。

[Permission-only API operations](#) 列出無法由客戶程式碼或 AWS Command Line Interface 直接呼叫的 API 操作。但是 IAM 使用者透過主控台執行 AWS Cloud9 動作時，確實需要呼叫這些操作的許可。

公有 API 操作

AWS Cloud9 操作	必要許可 (API 動作)	資源
CreateEnvironmentEC2	cloud9:CreateEnvironmentEC2 建立 AWS Cloud9 EC2 開發環境所需。	*
CreateEnvironmentMembership	cloud9:CreateEnvironmentMembership 必須具備才能將成員新增至環境。	arn:aws:cloud9:REGION_ID:ACCOUNT_ID:environment:ENVIRONMENT_ID
DeleteEnvironment	cloud9>DeleteEnvironment 必須具備才能刪除環境。	arn:aws:cloud9:REGION_ID:ACCOUNT_ID:environment:ENVIRONMENT_ID
DeleteEnvironmentMembership	cloud9>DeleteEnvironmentMembership 必須具備才能從環境移除成員。	arn:aws:cloud9:REGION_ID:ACCOUNT_ID:environment:ENVIRONMENT_ID

AWS Cloud9 操作	必要許可 (API 動作)	資源
DescribeEnvironmentMemberships	cloud9:DescribeEnvironmentMemberships 必須具備才能取得環境中的成員清單。	*
DescribeEnvironments	cloud9:DescribeEnvironments 必須具備才能取得環境的相關資訊。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
DescribeEnvironmentStatus	cloud9:DescribeEnvironmentStatus 必須具備才能取得環境的狀態相關資訊。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
UpdateEnvironment	cloud9:UpdateEnvironment 必須具備才能更新環境的設定。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>
UpdateEnvironmentMembership	cloud9:UpdateEnvironmentMembership 必須具備才能更新環境中成員的設定。	arn:aws:cloud9: <i>REGION_ID</i> : <i>ACCOUNT_ID</i> :environment: <i>ENVIRONMENT_ID</i>

需要許可的 API 操作

AWS Cloud9 操作	描述	主控台文件
ActivateEC2Remote	cloud9:ActivateEC2Remote	在 AWS Cloud9 中開啟環境

AWS Cloud9 操作	描述	主控台文件
	啟動 AWS Cloud9 IDE 連線到的亞馬遜 EC2 執行個體。	
CreateEnvironmentSSH	cloud9:CreateEnvironmentSSH 創建一個 AWS Cloud9 SSH 開發環境。	建立 SSH 環境
CreateEnvironmentToken	cloud9:CreateEnvironmentToken 建立身分驗證字符，以允許在 AWS Cloud9 IDE 和使用者的環境之間形成連線。	建立 EC2 環境
DescribeEC2Remote	cloud9:DescribeEC2Remote 擷取與 EC2 開發環境連線的相關詳細資訊，包括主機、使用者和連接埠。	建立 EC2 環境
DescribeSSHRemote	cloud9:DescribeSSHRemote 擷取與 SSH 開發環境連線的相關詳細資訊，包括主機、使用者和連接埠。	建立 SSH 環境
GetEnvironmentConfig	cloud9:GetEnvironmentConfig 取得用來初始化 AWS Cloud9 IDE 的組態資訊。	使用 AWS Cloud9 整合開發環境 (IDE)

AWS Cloud9 操作	描述	主控台文件
GetEnvironmentSettings	<p>cloud9:GetEnvironmentSettings</p> <p>取得指定開發環境的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 整合開發環境 (IDE)
GetMembershipSettings	<p>cloud9:GetMembershipSettings</p> <p>取得指定環境成員的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 的共用環境
GetUserPublicKey	<p>cloud9:GetUserPublicKey</p> <p>取得使用者的公開安全殼層金鑰，此金鑰可用 AWS Cloud9 來連線至 SSH 開發環境。</p>	建立 SSH 環境
GetUserSettings	<p>cloud9:GetUserSettings</p> <p>取得指定使用者的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 整合開發環境 (IDE)
ModifyTemporaryCredentialsOnEnvironmentEC2	<p>cloud9:ModifyTemporaryCredentialsOnEnvironmentEC2</p> <p>在 AWS Cloud9 整合式開發環境 (IDE) 使用的 Amazon EC2 執行個體上設定 AWS 受管的臨時登入資料。</p>	AWS 受管理的臨時憑

AWS Cloud9 操作	描述	主控台文件
UpdateEnvironmentSettings	<p>cloud9:UpdateEnvironmentSettings</p> <p>更新指定開發環境的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 整合開發環境 (IDE)
UpdateMembershipSettings	<p>cloud9:UpdateMembershipSettings</p> <p>更新指定環境成員的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 的共用環境
UpdateSSHRemote	<p>cloud9:UpdateSSHRemote</p> <p>更新與 SSH 開發環境連線的相關詳細資訊，包括主機、使用者和連接埠。</p>	建立 SSH 環境
UpdateUserSettings	<p>cloud9:UpdateUserSettings</p> <p>更新指定使用者的 AWS Cloud9 IDE 設定。</p>	使用 AWS Cloud9 整合開發環境 (IDE)
GetMigrationExperiences	<p>cloud9:GetMigrationExperiences</p> <p>授予 AWS Cloud9 使用者取得從移轉體驗 AWS Cloud9 到的權限 CodeCatalyst。</p>	

AWS 受管理的臨時憑

如果您只是在尋找 AWS 管理臨時憑據支持的操作列表，請跳到[AWS 受管理暫時認證支援的動作](#)。

對於 AWS Cloud9 EC2 開發環境，可 AWS Cloud9 讓您在環境中使用臨時 AWS 存取登入資料。我們將這些憑證稱為 AWS 受管臨時憑證。其具備下列優點：

- 您不需要將 AWS 實體 (例如 IAM 使用者) 的永久 AWS 存取登入資料儲存在環境中的任何位置。這可防止環境成員在您不知悉且未核准的情況下存取這些憑證。
- 您不需要手動設定、管理執行個體描述檔，或將其連接到與環境連線的 Amazon EC2 執行個體。執行個體設定檔是管理臨時 AWS 存取認證的另一種方法。
- AWS Cloud9 會持續更新其暫時登入資料，因此只能在有限的時間內使用單一認證集。這是安 AWS 全性最佳作法。如需詳細資訊，請參閱 [建立和更新 AWS 受管理的臨時認證](#)。
- AWS Cloud9 對於如何使用其臨時登入資料來存取環境中的 AWS 動作和資源，提供額外的限制。這也是安 AWS 全性最佳作法。

Important

目前，如果您環境的 EC2 執行個體啟動到私有子網路，則無法使用 AWS 受管暫時登入資料來允許 EC2 環境代表 AWS 實體 (例如 IAM 使用者) 存取 AWS 服務。

如需詳細了解可在私有子網路中啟動 EC2 執行個體的時機，請參閱 [為 AWS Cloud9 建立子網路](#)。

Note

當您使用受 AWS 管理的臨時登入資料時，請考慮使用 AWS 受管原則而非內嵌政策。

以下是每當 EC2 環境嘗試代表 AWS 實體 (例如 IAM 使用者) 存取時，AWS 受管臨時登入資料的運作方式：AWS 服務

1. AWS Cloud9 檢查呼叫 AWS 實體 (例如 IAM 使用者) 是否具有針對中所請求資源採取要求動作的權限 AWS。如果許可不存在或已明確拒絕，則請求失敗。
 2. AWS Cloud9 檢查 AWS Managed 臨時認證，以查看其權限是否允許針對請求的資源執行請求的操作 AWS。如果許可不存在或已明確拒絕，則請求失敗。如需 AWS 管理暫時登入資料支援的權限清單，請參閱 [AWS 受管理暫時認證支援的動作](#)。
- 如果 AWS 實體和 AWS 受管理的臨時認證都允許對請求的資源執行請求的動作，則請求成功。

- 如果 AWS 實體或 AWS Managed 臨時認證明確拒絕或無法明確允許請求的資源所請求的動作，則請求將失敗。這表示即使呼叫 AWS 實體具有正確的權限，如果要求 AWS Cloud9 未明確允許，也會失敗。同樣地，如果 AWS Cloud9 允許針對特定資源採取特定動作，則如果 AWS 實體也未明確允許，則要求會失敗。

EC2 環境的擁有者可以隨時開啟或關閉該環境的 AWS 受管暫時登入資料，如下所示：

1. 在環境開啟的情況下，在 AWS Cloud9 IDE 的功能表列上選擇「AWS Cloud9 偏好設定」。
2. 在 Preferences (偏好設定) 索引標籤的導覽窗格中，選擇 AWS Settings, Credentials (設定、憑證)。
3. 使用 AWS managed temporary credentials (AWS 受管臨時憑證) 將 AWS 受管臨時憑證設為開啟或關閉。

Note

您也可以透過呼叫 AWS Cloud9 API 作業 [UpdateEnvironment](#) 並將值指派給 `managedCredentialsAction` 參數，來開啟或關閉 AWS 受管理的臨時登入資料。您可以使用標準 AWS 工具 (例如 AWS SDK 和 AWS CLI

如果您關閉 AWS 受管理的臨時登入資料，則無論提出要求的 AWS 實體為何 AWS 服務，環境都無法存取任何登入資料。但是，假設您不能或不想要為環境開啟 AWS 受管理的臨時登入資料，而且您仍然需要該環境才能存取 AWS 服務。然後，請考慮下列替代方案：

- 將執行個體描述檔連接到連線至環境的 Amazon EC2 執行個體。如需相關指示，請參閱「[建立及使用執行個體描述檔來管理暫時登入資料](#)」。
- 將您的永久 AWS 存取認證儲存在環境中，例如透過設定特殊的環境變數或執行 `aws configure` 指令。如需說明，請參閱 [在環境中建立並存放永久存取憑證](#)。

上述替代方案會覆寫 EC2 環境中 AWS 受管臨時憑證所允許 (或拒絕) 的所有許可。

AWS 受管理暫時認證支援的動作

對於 AWS Cloud9 EC2 開發環境，AWS 受管臨時登入資料允許呼叫者中的所有 AWS 資源執行所有 AWS 動作 AWS 帳戶，但有以下限制：

- 對於 AWS Cloud9，只允許下列動作：

- `cloud9:CreateEnvironmentEC2`
- `cloud9:CreateEnvironmentSSH`
- `cloud9:DescribeEnvironmentMemberships`
- `cloud9:DescribeEnvironments`
- `cloud9:DescribeEnvironmentStatus`
- `cloud9:UpdateEnvironment`
- 在 IAM 方面，只允許下列動作：
 - `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
- 所有與角色互動的 IAM 動作只可使用以 Cloud9- 開頭的角色名稱。不過，iam:PassRole 可使用所有角色名稱。
- 對於 AWS Security Token Service (AWS STS)，只允許下列動作：
 - sts:GetCallerIdentity
 - sts:DecodeAuthorizationMessage
- 所有支援的 AWS 動作僅限於環境的 IP 位址。這是安 AWS 全性最佳作法。

如果 AWS Cloud9 不支援您需要 EC2 環境存取的動作或資源，或者 EC2 環境的 AWS 受管暫時登入資料已關閉且無法重新開啟，請考慮下列替代方法：

- 將執行個體描述檔連接到連線至 EC2 環境的 Amazon EC2 執行個體。如需說明，請參閱[建立及使用執行個體描述檔來管理臨時憑證](#)。
- 將永久存 AWS 取登入資料儲存在 EC2 環境中，例如透過設定特殊的環境變數或執行aws configure命令。如需說明，請參閱[在環境中建立並存放永久存取憑證](#)。

上述替代方案會覆寫 EC2 環境中 AWS 受管臨時憑證所允許 (或拒絕) 的所有許可。

建立和更新 AWS 受管理的臨時認證

對於 AWS Cloud9 EC2 開發環境，會在您第一次開啟環境時建立 AWS 受管的臨時登入資料。

AWS 受管理的臨時登入資料會在下列任一情況下更新：

- 每經過特定的一段時間時。目前是每五分鐘一次。
- 每次您重新載入顯示環境 IDE 的 Web 瀏覽器標籤時。
- 達到環境的 ~/.aws/credentials 檔案中所列的時間戳記時。
- 在 AWS managed temporary credentials (AWS 受管臨時憑證) 設定設為關閉的情況下，每次您再重新開啟時。(若要檢視或變更此設定，請在 IDE 選單列中，選擇AWS Cloud9,Preferences (AWS Cloud9、偏好設定)。在 Preferences (偏好設定) 索引標籤的導覽窗格中，選擇 AWS Settings, Credentials (AWS 設定、憑證)。)

- 為了安全起見，AWS 受管理的臨時登入資料會在 15 分鐘後自動 若要重新整理憑證，環境擁有者必須透過 IDE 連線至 AWS Cloud9 環境。如需環境擁有者的角色詳細資訊，請參閱 [控制 AWS 受管臨時憑證的存取權](#)。

控制 AWS 受管臨時憑證的存取權

具有 AWS 受管理臨時登入資料的共同作業人員可以用 AWS Cloud9 來與其他 AWS 服務人互動。為了確保只有受信任的協作者才能獲得 AWS 受管臨時憑證，如果環境擁有者以外的任何人新增了成員，這些憑證就會停用。憑證會因 `~/.aws/credentials` 檔案的刪除而停用。

Important

AWS 受管理的臨時登入資料也會每 15 分鐘自動過期。若要重新整理認證，以便協同合作者可以繼續使用它們，環境擁有者必須透過 IDE 連線至 AWS Cloud9 環境。

只有環境擁有者可以重新啟用 AWS 受管理的臨時登入資料，以便與其他成員共用。當環境擁有者開啟 IDE 時，會出現一個對話方塊，確認 AWS 受管理的暫時認證已停用。環境擁有者可以重新啟用所有成員的憑證，或維持停用所有成員的憑證。

Warning

為了符合最佳安全性作法，如果您對上次新增至環境的使用者身分有疑慮，請將受管臨時憑證維持停用。您可以在 [Collaborate](#) (協作) 視窗檢查具有讀取/寫入權限的成員清單。

登錄和監控 AWS Cloud9

監視活動 CloudTrail

AWS Cloud9 與 (提供中的使用者 AWS CloudTrail、角色或服務所採取的動作記錄) 的 AWS 服務整合 AWS Cloud9。CloudTrail 擷取 AWS Cloud9 作為事件的所有 API 呼叫。擷取的呼叫包括來自 AWS Cloud9 主控台的呼叫，以及來自 AWS Cloud9 API 的程式碼呼叫。

如果您建立追蹤，您可以啟用持續交付 CloudTrail 事件到 Amazon Simple Storage Service (Amazon S3) 儲存貯體，包括 AWS Cloud9。

如果您未設定追蹤，您仍然可以在 [事件歷程記錄] 中檢視 CloudTrail 主控台中最近的事件。使用收集的資訊 CloudTrail，您可以判斷提出的要求 AWS Cloud9、提出要求的 IP 位址、提出要求的人員、提出要求的時間，以及其他詳細資訊。

如需詳細資訊，請參閱 [使用 AWS CloudTrail 記錄 AWS Cloud9 API 呼叫](#)。

監控 EC2 環境效能

如果您使用 AWS Cloud9 EC2 開發環境，則可以監控關聯 Amazon EC2 執行個體的可靠性、可用性和效能。透過執行個體狀態監控，您可以快速判斷 Amazon EC2 是否有偵測到任何可能會防止您執行個體執行應用程式的問題。

如需詳細資訊，請參閱適用於 Linux 執行個體的 Amazon EC2 使用者指南中的 [監控 Amazon EC2](#)。

符合性驗證 AWS Cloud9

協力廠商稽核人員會評估 AWS 服務的安全性與合規性，作為多項 AWS 合規計畫的一部分。

AWS Cloud9 符合以下合規計畫的範圍：

SOC

AWS 系統與組織控制 (SOC) 報告是獨立的第三方檢驗報告，展示如何 AWS 達成關鍵合規性控制與目標。

服務	SDK	SOC 1、2、3
AWS Cloud9	Cloud9	✓

PCI

支付卡行業數據安全標準 (PCI DSS) 是由 PCI 安全標準委員會管理的專有信息安全標準，該委員會由美國運通，發現金融服務，JCB 國際，全 MasterCard 球和 Visa Inc 成立。

服務	SDK	PCI
AWS Cloud9	Cloud9	✓

FedRAMP

聯邦風險與授權管理計劃 (FedRAMP) 是一項美國政府整體計劃，提供標準化的方法，為雲端產品和服務進行安全評估、授權和持續監控。

透過 FedRAMP 評估和授權的服務將具有以下狀態：

- 第三方評估組織 (3PAO) 評估：此服務目前正在進行我們的第三方評估人員的評估。
- 聯合授權委員會 (JAB) 審查：此服務目前正在進行 JAB 的審查。

服務	SDK	FedRAMP Moderate (East/West)	FedRAMP () GovCloud
AWS Cloud9	Cloud9	JAB 審查	N/A

DoD CC SRG

國防部 (DoD) 雲端運算安全要求指南 (SRG) 提供標準化的評定和授權程序，讓雲端服務提供者 (CSP) 取得 DoD 臨時授權，以便為 DoD 客戶提供服務。

透過 DoD CC SRG 評估和授權的服務將具有以下狀態：

- 第三方評估組織 (3PAO) 評估：此服務目前正在進行我們的第三方評估人員的評估。
- 聯合授權委員會 (JAB) 審查：此服務目前正在進行 JAB 的審查。
- 國防通訊局 (DISA) 審查：此服務目前正在進行 DISA 的審查。

服務	軟體開發套件	DoD CC SRG IL2 (East/West)	DoD 抄送 SRG IL2 () GovCloud	DoD 抄送 SRG IL4 () GovCloud	DoD 抄送 SRG IL5 () GovCloud	DoD CC SRG IL6 (秘密區域) AWS
AWS Cloud9	Cloud9	JAB 審查	N/A	N/A	N/A	N/A

HIPAA BAA

1996 年健康保險流通與責任法案 (HIPAA) 是一項聯邦法律，要求制定國家標準以保護敏感的患者健康資訊在未經患者同意或不知情的情況下公開。

AWS 使受 HIPAA 規範的涵蓋實體及其業務夥伴能夠安全地處理、儲存和傳輸受保護的健康資訊 (PHI)。此外，截至 2013 年 7 月，為此類客戶 AWS 提供標準化的商業夥伴增補合約 (BAA)

服務	SDK	HIPAA BAA
AWS Cloud9	Cloud9	✓

IRAP

資訊安全註冊評估人員計劃 (IRAP) 可讓澳洲政府客戶驗證是否有適當的控制措施，並決定適當的責任模式，以便滿足由澳洲網路安全中心 (ACSC) 所制定的澳洲政府資訊安全手冊 (ISM) 要求。

服務	命名空間*	IRAP 受保護
AWS Cloud9	Cloud9	✓

* 命名空間可協助您識別環境中的 AWS 服務。例如，當您建立 IAM 政策時，請使用 Amazon 資源名稱 (ARN) 和讀取 AWS CloudTrail 日誌。

C5

雲端運算合規控制目錄 (C5) 是德國政府支持的認證計劃，由德國聯邦資訊安全辦公室 (BSI) 在德國推出，旨在協助組織在德國政府的「雲端供應商安全建議」範圍內使用雲端服務時展示針對常見網路攻擊的操作安全性。

服務	SDK	C5
AWS Cloud9	Cloud9	✓

FINMA

FINMA 是瑞士獨立的金融市場監管機構。Amazon Web Services (AWS) 已完成 FINMA ISAE 3000 第 2 類報告。

服務	SDK	FINMA
AWS Cloud9	Cloud9	✓

GSMA

GSM 協會是代表全球行動網路營運商利益的行業組織。Amazon Web Services (AWS) 歐洲 (巴黎) 和美國東部 (俄亥俄) 區域現已通過 GSM 協會 (GSMA) 的安全認證計劃訂閱管理 (SAS-SM) 認證，範圍包括資料中心營運和管理 (DCOM)。這種符合 GSMA 要求的做法證明我們一直致力於滿足雲端服務供應商的更高期待。

服務	美國東部 (俄亥俄)	歐洲 (巴黎)
AWS Cloud9	✓	✓

PiTuKri

AWS PiTuKri 符合要求，證明我們持續致力於滿足芬蘭交通與通訊局 Traficom 對雲端服務提供者的高度期望。

服務	SDK	PiTuKri
AWS Cloud9	Cloud9	✓

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱[AWS 服務 遵循規範計劃](#)方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱[AWS 規範計劃](#)。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載中的報告中的](#) AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於您資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- [安全性與合規性快速入門指南](#) — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- [在 Amazon Web Services 上架構 HIPAA 安全性與合規性](#) — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 應用程式。

Note

並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源AWS](#) — 此工作簿和指南集合可能適用於您的產業和所在地。
- [AWS 客戶合規指南](#) — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- [使用AWS Config 開發人員指南中的規則評估資源](#) — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub](#)— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [AWS Audit Manager](#)— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

韌性 AWS Cloud9

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。AWS 區域提供多個實體分離和隔離的可用區域，這些區域透過低延遲、高輸送量和高度備援的網路連線。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需區域和可用區域的相關 AWS 資訊，請參閱[AWS 全域基礎結構](#)。

除了 AWS 全球基礎架構外，還 AWS Cloud9 支援特定功能，以支援您的資料復原和備份需求。

- AWS Cloud9 與 Amazon Web Services 託管的版本控制服務整合 AWS CodeCommit，您可以使用該服務在雲端中私有存放和管理資產 (例如文件、原始程式碼和二進位檔案)。如需詳細資訊，請參閱《AWS CodeCommit 使用指南》AWS CodeCommit中的「[整合 AWS Cloud9](#)」。
- 在 AWS Cloud9 開發環境中使用 Git 版本控制系統來備份遠端 GitHub 儲存庫上的檔案和資料。如需詳細資訊，請參閱 [使用 Git 面板的視覺化來源控制](#)。

基礎結構安全 AWS Cloud9

作為託管服務，AWS Cloud9 受到 AWS 全球網絡安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱[AWS 雲端安全](#)。若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱[安全性支柱架構](#)。良好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過 AWS Cloud9 過網路進行存取。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

Note

依預設，AWS Cloud9 EC2 開發環境會自動為執行個體的系統套件安裝安全性修補程式。

軟體更新與修補

AWS Cloud9 開發環境在雲端運算資源之上執行。雲端運算資源可以是 EC2 環境的 Amazon EC2 執行個體，也可以是您自己的 SSH 環境雲端運算資源。如需這兩個選項的詳細資訊，請參閱〈[環境與運算資源節](#)〉。

AWS Cloud9 EC2 環境會在環境啟動後自動安裝作業系統安全性修補程式和更新。AWS Cloud9 環境也包含運作和支援 IDE 所需的軟體套件。當環境載入時，會自動修補這些套件。特定的開發工具已預先安裝在 AWS Cloud9 環境中。AWS Cloud9 更新 AMI 中的這些工具，但我們不會在您的環境中自動更新它們。如需如何更新這些工具的詳細資訊，請參閱下列各節：

- [安裝或更新《AWS Command Line Interface 使用指南》AWS CLI中的最新版本](#)。
- 在AWS Serverless Application Model 開發人員指南中[管理 AWS SAM CLI 版本](#)。
- [安裝 AWS CDK在AWS Cloud Development Kit \(AWS CDK\)開發人員指南中](#)。

無論基礎雲端運算資源或自動更新的頻率為何，AWS Cloud9 使用者或其 AWS Cloud9 管理員都必須負責確保已修補雲端運算資源並保持最新狀態。

如需深入了解哪些客戶在[共同責任模型](#)之下負有責任，請參閱[資料保護 AWS Cloud9](#)。

安全性最佳做法 AWS Cloud9

以下最佳實務為一般準則，並不代表完整的安全解決方案。因為這些最佳實務可能不適合或無法滿足您的環境，所以請將其視為實用建議，不要當成指示。

一些安全性最佳做法 AWS Cloud9

- 將您的程式碼安全地存放在版本控制系統中，例如 [AWS CodeCommit](#)。
- 對於 AWS Cloud9 EC2 開發環境，請設定和使用 [Amazon 彈性區塊存放區](#) 加密磁碟區。
- 在 EC2 環境中，請使用 [標籤](#) 來控制您 AWS Cloud9 資源的存取權。
- 針對您的共用 AWS Cloud9 開發環境，請遵循它們的 [最佳做法](#)。

AWS Cloud9 疑難排解

使用以下資訊來查明與解決 AWS Cloud9 的相關問題。

如果您的問題未列於其中，或者，如果您需要額外協助，請參閱 [AWS Cloud9 開發論壇](#)。當您進入此論壇時，您可能需要登入。您也可以直接[聯絡我們](#)。

主題

- [Installer \(安裝程式\)](#)
- [AWS Cloud9 環境](#)
- [Amazon EC2](#)
- [其他 AWS 服務](#)
- [申請預覽](#)
- [效能](#)
- [第三方應用程式和服務](#)

Installer (安裝程式)

下節概述了與AWS Cloud9安裝程式相關的疑難排解問題。

AWS Cloud9 安裝程式停止回應或失敗

問題：當您[下載並執行 AWS Cloud9 安裝程式](#)時，發生一個或多個錯誤，且安裝程式指令碼未顯示 Done。

原因：AWS Cloud9 安裝程式發生一個或多個無法復原的錯誤，因此失敗。

解決方案：如需詳細資訊，請參閱 [AWS Cloud9 安裝程式疑難排解](#)。請參閱所提供的常見問題、可能的原因，以及建議解決方案。

AWS Cloud9 安裝程式未在顯示「Package Cloud9 IDE 1」(套件 Cloud9 IDE 1) 之後完成

問題：建立 SSH 開發環境的過程中會在您現有的 Amazon EC2 執行個體或自有伺服器上安裝 AWS Cloud9。當您在 AWS Cloud9 安裝程式對話方塊上看到此訊息：「套件 Cloud9 IDE 1」後，安裝程式

呈現暫停狀態。如果您選擇 Cancel (取消)，您會看到下列訊息：「安裝失敗。」此錯誤發生在無法於客戶的 SSH 主機安裝 AWS Cloud9 套件時。

原因：SSH 主機需要您已安裝 Node.js。建議您安裝主機作業系統支援的最新 Node.js 版本。如果您的主機上有 AWS Cloud9 不支援的 Node.js 版本，則可能會發生安裝錯誤。

建議解決方案：在 SSH 主機上安裝 AWS Cloud9 支援的 Node.js 版本。

安裝相依性失敗

問題：AWS Cloud9 需要網際網路存取權才能下載相依性。

可能原因：

- 如果您的 AWS Cloud9 環境使用 Proxy 來存取網際網路，則 AWS Cloud9 需要 Proxy 詳細資料才能安裝相依性。如果您沒有提供代理詳細資訊給 AWS Cloud9，會出現此錯誤。
- 造成這種情況的另一個原因可能是如果您的環境不允許輸出流量。

建議解決方案：

- 要提供您的代理詳細信息 AWS Cloud9，請將以下代碼附加到環境 `~/.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`，請將以下幾行新增至您的 `~/.bashrc` 檔案，並設定 `NODE_EXTRA_CA_CERTS` 為 PEM 格式的憑證授權單位檔案路徑。如需此變數的詳細資訊，請參閱 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]
```

- 如果您使用的是無輸入的 Amazon EC2 執行個體，則必須確保已設定適用於 Amazon S3 的 Amazon VPC 端點。如需詳細資訊，請參閱 [針對 Amazon S3 下載相依性設定 Amazon VPC 端點](#)。

SSH 環境錯誤:Python 安裝 pty.js 需要三版本」

問題：開啟安AWS Cloud9全殼層開發環境之後，AWS Cloud9IDE 中的終端機會顯示以「安裝 pty.js 時需要 Python 版本 3」開頭的訊息。

原因:要按預期工作，SSH 環境需要安裝 Python 版本 3。

解決方案：在環境中安裝 Python 版本 3。若要檢查您的版本，請從伺服器的終端機執行 `python --version` 命令。若要在您的伺服器上安裝 Python 3，請參閱下列其中一項：

- Python 範例中的 [步驟 1：安裝 Python](#)。
- 在 [Python 網站上下載](#)。

AWS Cloud9 環境

下節概述了與AWS Cloud9環境相關的疑難排解問題。

環境建立錯誤：「我們無法建立 EC2 執行個體...」

問題：在嘗試建立 AWS Cloud9 開發環境時，會顯示一則訊息表示：「We are unable to create EC2 instances in your account during account verification and activation.」（我們無法在帳戶驗證和啟用期間，在您的帳戶中建立 EC2 執行個體。）

原因：AWS 目前正在驗證和啟用您的 AWS 帳戶。啟用完成前（最多可能需要 24 小時），您都無法建立這個執行個體或其他環境。

解決方案：嘗試稍候重新建立環境。如果您在 24 小時後仍然收到此訊息，請聯絡[支援](#)部門。此外，請務必了解，即使嘗試建立環境失敗，AWS CloudFormation 仍會在您的帳戶中建立相關的堆疊。這些堆疊會計入您帳戶的堆疊建立配額中。為了避免耗盡堆疊建立配額，您可以刪除這些失敗的堆疊。如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的[刪除 AWS CloudFormation 主控台的堆疊](#)。

環境建立錯誤：「未授權執行 sts：AssumeRole」

問題：當您嘗試建立新環境時，您會看到此錯誤：「未授權執行 sts:」AssumeRole，且未建立環境。

可能原因：您的 AWS 帳戶 帳戶中沒有 AWS Cloud9 服務連結角色。

建議解決方案：在您的 AWS 帳戶 中建立 AWS Cloud9 服務連結角色。您可以在 AWS Command Line Interface (AWS CLI) 或 AWS CloudShell 中執行下列命令，即可執行這項操作。

```
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.
```

如果您無法執行此操作，請洽詢您的 AWS 帳戶 管理員。

執行上述命令後，再次嘗試建立環境。

聯合身分無法建立環境

問題：當您嘗試使用 AWS 聯合身分建立 AWS Cloud9 開發環境時，出現存取錯誤訊息而未能建立環境。

原因：AWS Cloud9 使用服務連結角色。在帳戶中使用 `iam:CreateServiceLinkedRole` 呼叫首次建立環境後，服務連結角色會隨之建立。不過，聯合身分使用者無法呼叫 IAM API。如需詳細資訊，請參閱《AWS Security Token Service API 參考》中的 [GetFederationToken](#)。

解決方案：請 AWS 帳戶 管理員在 IAM 主控台為 AWS Cloud9 建立服務連結角色，或以 AWS Command Line Interface (AWS CLI) 執行此命令來建立角色：

```
aws iam create-service-linked-role --aws-service-name cloud9.amazonaws.com
```

或者，使用 AWS-shell 來執行此命令：

```
iam create-service-linked-role --aws-service-name cloud9.amazonaws.com
```

如需詳細資訊，請參閱 [IAM 使用者指南](#) 中的使用服務連結角色。

主控台錯誤：「User is not authorized to perform action on resource」(使用者未獲授權對資源執行動作)

問題：當您嘗試使用 AWS Cloud9 主控台來建立或管理 AWS Cloud9 開發環境時，您會看到錯誤，其中包含類似「使用者 `arn:aws:iam::123456789012:user/MyUser` 未獲授權在資源 `arn:aws:cloud9:us-east-2:123456789012:environment:12a34567b8cd9012345ef67abcd890e1` 上執行 `cloud9:action`」的語句，其中：

- `arn:aws:iam::123456789012:user/MyUser` 是發出請求之使用者的 Amazon Resource Name (ARN)。
- `action` 是接收請求之使用者的操作名稱。
- `arn:aws:cloud9:us-east-2:123456789012:environment:12a34567b8cd9012345ef67abcd890e1` 是使用者請求執行操作的環境的 ARN。

原因：您用於登入 AWS Cloud9 主控台的使用者未具備正確的 AWS 存取許可，以致無法執行該動作。

解決方案：確認使用者具備正確的 AWS 存取許可，然後再次嘗試執行該動作。如需詳細資訊，請參閱下列內容：

- 小組設定中的 [步驟 3：為群組新增 AWS Cloud9 存取許可](#)
- 企業設定中的 [步驟 6：讓組織內的群組和使用者使用 AWS Cloud9](#)
- 使用共用環境中的 [關於環境成員的存取角色](#)

無法連線到環境

問題：使用者無法連接到環境，停滯在「連接」階段。

原因：如果您變更 `~/ .ssh/authorized_keys` 檔案的權限，從該檔案移除 AWS Cloud9 金鑰或完全移除檔案，則可能會發生此問題。

解決方案：不要刪除此檔案。如果您刪除它，就必須重新建立您的環境，並且可能需要將現有環境的 [EBS 磁碟區](#) 連接到新的 EC2 環境。這是為了擷取遺失的資料。如果缺少許可，請確保檔案具有 Read-Write 許可。這是為了允許 SSH 常駐程式讀取它。

無法開啟環境

問題：當您嘗試開啟環境時，IDE 超過五分鐘還未顯示。

可能原因：

- 登入 AWS Cloud9 主控台的 IAM 使用者未具備開啟環境所需的 AWS 存取許可。
- 如果環境與 AWS 雲端運算執行個體 (如 Amazon EC2 執行個體) 相關聯，則可能是下列原因：
 - 該執行個體的相關聯 VPC 未設為 AWS Cloud9 所適用的正確設定。

- 當 AWS Cloud9 嘗試連接至該執行個體時，執行個體正在轉換狀態或自動狀態檢查失敗。
- 若環境是 SSH 環境，表示相關聯的雲端運算執行個體或自有伺服器並未正確設為允許 AWS Cloud9 進行存取。

建議解決方案：

- 確定登入 AWS Cloud9 主控台的 IAM 使用者具備開啟環境所需的 AWS 存取許可。然後再次嘗試開啟環境。如需詳細資訊，請參閱以下內容或洽詢您的 AWS 帳戶 管理員：
 - 小組設定中的 [步驟 3：為群組新增 AWS Cloud9 存取許可](#)
 - 身分驗證與存取控制中的 [AWS 受管理的政策 AWS Cloud9](#)
 - 進階小組設定中的 [使用 AWS Cloud9 的小組適用的客戶受管政策範例](#)
 - 身分驗證與存取控制中的 [客戶受管政策範例](#)
 - 《IAM 使用者指南》中的 [變更 IAM 使用者的許可](#)
 - IAM 使用者指南中的 [IAM 政策故障診斷](#)

若是登入的 IAM 使用者仍然無法開啟環境，請嘗試先登出，再以 AWS 帳戶 根使用者身分或帳戶中的管理員使用者身分重新登入。然後再次嘗試開啟環境。如果這麼做能夠開啟環境，問題很可能就跟 IAM 使用者存取許可有關。

- 如果環境與 AWS 雲端運算執行個體 (如 Amazon EC2 執行個體) 相關聯，請執行下列操作：
 - 確定該執行個體的相關聯 VPC 已設為 AWS Cloud9 適用的正確設定，然後再次嘗試開啟環境。如需詳細資訊，請參閱[AWS Cloud9 的 Amazon VPC 需求](#)。

如果 AWS 雲端運算執行個體相關聯的 VPC 已設為 AWS Cloud9 適用的正確設定，但仍然無法開啟環境，則可能是因為執行個體的安全群組被禁止存取 AWS Cloud9。以下僅做為疑難排解技巧之用：請檢查安全群組，確認至少允許所有 IP 地址透過連接埠 22 傳入 SSH 流量 (Anywhere 或 0.0.0.0/0)。如需相關指示，請參閱 Amazon EC2 的 Linux 執行個體使用者指南中的[說明您的安全群組](#)和[更新安全群組規則](#)。

如需其他 VPC 疑難排解步驟，請觀看相關的 5 分鐘影片[AWS 知識中心影片：如果無法連線至 VPC 中的執行個體，該如何檢查？](#) 上 YouTube。

Warning

完成疑難排解時，請務必將傳入規則設為適當的地址範圍。如需詳細資訊，請參閱[the section called “傳入 SSH IP 地址範圍”](#)。

- 重新啟動該執行個體、確定執行個體運作中並已通過所有系統檢查，然後再次嘗試開啟環境。如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的[重新啟動您的執行個體](#)和[檢視狀態檢查](#)。
- 若環境是 SSH 環境，請確認相關聯的雲端運算執行個體或自有伺服器已正確設為允許 AWS Cloud9 進行存取。然後再次嘗試開啟環境。如需詳細資訊，請參閱[SSH 環境主機要求](#)。

無法開啟 AWS Cloud9 環境：「This environment cannot be currently accessed by collaborators. Please wait until the removal of managed temporary credentials is complete, or contact the owner of this environment.」(此環境目前無法供協作者存取。請等候受管臨時憑證的移除作業完成，或連絡此環境的擁有者。)

問題：如果新的協作者是由環境擁有者以外的人新增至環境中，AWS 受管臨時憑證會遭到停用。憑證會因您刪除 `~/.aws/credentials` 檔案而停用。正在刪除 `~/.aws/credentials` 檔案時，新的協作者無法存取 AWS Cloud9 環境。

原因：在 AWS 受管臨時憑證刪除過程中防止環境遭存取是一種安全性措施。這樣可讓環境擁有者確認只有受信任的協作者才能存取受管憑證。如果環境擁有者認為協作者名單沒有問題，就可以重新啟用受管憑證，讓憑證可以共用。如需詳細資訊，請參閱[控制 AWS 受管臨時憑證的存取權](#)。

建議解決方案：等候 `~/.aws/credentials` 檔案刪除完畢，然後再次嘗試開啟 AWS Cloud9 環境。憑證到期的最長等待時間為 15 分鐘。或者，請要求環境擁有者重新啟用或停用受管臨時憑證。重新啟用或停用憑證之後，協作者就可以立即存取環境。透過將受管憑證的狀態切換為 ENABLED 或 DISABLED，環境擁有者就可確保憑證不會保留在中繼狀態。中繼統計資料可以防止協作者存取環境。

Note

假設環境擁有者和協同合作者屬於同一個 AWS 帳戶。然後，協作者可以在主控台的 Your environments (您的環境) 頁面查看環境卡片，藉此找到要聯絡的環境擁有者。環境擁有者也會列在 Environment details (環境詳細資訊) 頁面上。

環境刪除錯誤：「One or more environments failed to delete」(無法刪除一或多個環境)

問題：當您嘗試在 AWS Cloud9 主控台中刪除一或多個環境，會有一則訊息顯示，內容指出「one or more environments failed to delete」(無法刪除一或多個環境)，且至少有一個環境未刪除。

可能的原因：AWS CloudFormation 在刪除一或多個環境時可能發生問題。AWS Cloud9 倚賴 AWS CloudFormation 來建立和刪除環境。

建議解決方案：嘗試使用 AWS CloudFormation 來刪除每個未刪除的環境。

1. 造訪網址 <https://console.aws.amazon.com/cloudformation> 以開啟 AWS CloudFormation 主控台：
2. 在 AWS 導覽列中，選擇環境的 AWS 區域。
3. 在 AWS CloudFormation 堆疊的清單中，選取 Stack name (堆疊名稱) 包含未刪除的環境名稱且 Status (狀態) 是 DELETE_FAILED 的項目。例如，如果環境名稱是 **my-demo-environment**，請選擇以 aws-cloud-9-名稱開頭的堆疊。my-demo-environment(請選擇環境名稱旁的方塊或選項，而非環境名稱本身。)
4. 選擇 Actions (動作)、Delete Stack (刪除堆疊)。
5. 出現提示時，選擇 Yes, Delete (是，刪除)。

刪除堆疊的程序可能需要幾分鐘的時間。

如果堆疊從清單中消失，則表示環境已刪除。

如果幾分鐘過後堆疊仍然顯示 DELETE_FAILED，則表示環境仍未刪除。您可以嘗試手動刪除已失敗堆疊的每個資源。

Note

手動刪除失敗堆疊的資源並不會從您的 AWS 帳戶中移除堆疊本身。

若要手動刪除這些資源，請執行下列動作。在 AWS CloudFormation 主控台中選擇失敗的堆疊，然後選取 Resources (資源) 區段。移至此清單中每個資源在 AWS 中的主控台，然後使用該主控台來刪除資源。

變更 AWS Cloud9 IDE 中環境的逾時時間

問題：使用者想要更新 Amazon EC2 環境的逾時時間。

原因:預設逾時時間為 30 分鐘。對於某些用戶來說，這可能太短了。

建議解決方案：

1. 開啟您要設定的環境。
2. 在 AWS Cloud9IDE 的功能表列上，選擇「AWS Cloud9偏好設定」。
3. 在「偏好設定」視窗中，捲動至 Amazon EC2 執行個體區段。
4. 從可用清單中選取逾時值並進行更新。

在本機 AWS Toolkit 執行 SAM 應用程式時發生錯誤，因為 AWS Cloud9 環境沒有足夠的磁碟空間

問題：使用 AWS Toolkit 為 SAM 範本所定義的應用程式執行 AWS SAM CLI 命令時發生錯誤。

可能的原因：當您使用 AWS Toolkit 在本機執行無伺服器應用程式並除錯時，AWS SAM 會使用 Docker 映像。這些映像提供執行時間環境和建置工具，而這些工具會模擬您計劃要部署至其中的 Lambda 環境。

不過，如果您的環境缺乏足夠的磁碟空間，提供這些功能的 Docker 映像就無法建置，且您的本機 SAM 應用程式無法執行。如果發生這個問題，您可能會在 Output (輸出) 索引標籤收到錯誤，內容如下。

```
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.
```

此錯誤與使用 Python 執行時間建置的 SAM 應用程式相關。根據您為應用程式選擇的執行時間而定，您收到的訊息可能稍有不同。

建議解決方案：釋放環境中的磁碟空間，以便 Docker 映像可以進行建置。在 IDE 的終端機中執行以下命令，藉此移除任何未使用的 Docker 映像。

```
docker image prune -a
```

如果您因為磁碟空間的限制而反覆遇到 SAM CLI 命令的問題，請切換到使用不同[執行個體類型](#)的開發環境。

[\(回到頁首\)](#)

無法使用舊版 Microsoft Edge 瀏覽器載入 IDE

問題：嘗試使用 Microsoft Edge Web 瀏覽器載入 AWS Cloud9 IDE 時傳回 HTTP403：FORBIDDEN 錯誤。

可能原因：AWS Cloud9 IDE 不支援特定的舊版 Microsoft Edge。

建議解決方案：若要更新瀏覽器，請選擇 Microsoft Edge 工具列中的省略符號 (...) 按鈕。從功能表中，選擇 Settings (設定)，然後選擇 About Microsoft Edge (關於 Microsoft Edge)。如果需要更新，系統會自動下載並安裝更新。

[\(回到頁首\)](#)

無法在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment。

問題：在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment 時，您會收到一條錯誤訊息，表明無法開啟此目錄。

可能原因：目前無法使用 AWS Cloud9 IDE 的檔案系統，在相同名稱的資料夾內建立子資料夾結構 /home/ec2-user/environment。您將無法從 AWS Cloud9 IDE 檔案總管存取此目錄中的任何檔案，但是可以使用命令列存取這些檔案。此問題只影響檔案路徑 /home/ec2-user/environment/home/ec2-user/environment，/test/home/ec2-user/environment 和 /home/ec2-user/environment/test 等檔案路徑應該可以運作。這是已知問題，僅影響 AWS Cloud9 IDE 檔案總管。

建議解決方案：使用不同的檔案名稱和結構。

[\(回到頁首\)](#)

無法在 IDE 的文件資源管理器中創建子文件夾結構/項目/項目。AWS Cloud9 CodeCatalyst

問題：當您在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /專案/專案時 CodeCatalyst，您會收到一則錯誤訊息，指出無法開啟此目錄。

可能的原因：目前無法使用 AWS Cloud9 IDE 的檔案總管，在相同名稱的資料夾內建立子資料夾結構/專案。CodeCatalyst您將無法從 AWS Cloud9 IDE 檔案總管存取此目錄中的任何檔案，但是可以使用命令列存取這些檔案。此問題只會影響檔案路徑 /projects/projects，/test/projects 和 /projects/test 等檔案路徑應該可以運作。這是已知的問題，AWS Cloud9只會影響 CodeCatalyst。

建議解決方案：使用不同的檔案名稱和結構。

[\(回到頁首\)](#)

因為 **tmux** 工作階段發生錯誤，所以無法與 AWS Cloud9 中的終端機視窗互動。

問題：當您嘗試在 AWS Cloud9 中啟動新的終端機視窗，無法使用預期的命令列介面。沒有命令提示字元，您無法輸入文字。系統會傳回諸如 `tmux: need UTF-8 locale (LC_CTYPE)` 或 `invalid LC_ALL, LC_CTYPE or LANG` 等錯誤訊息。

可能原因：無回應終端機可能是由 **tmux** 錯誤所造成。AWS Cloud9 會使用 [tmux](#) 公用程式。如此，顯示在終端機中的資訊，即使在頁面重新載入或您重新連線至開發環境時也會持續存在。

在 **tmux** 工作階段中，終端機視窗中顯示的內容會由用戶端處理。用戶端會與可管理多個工作階段的伺服器通訊。伺服器和用戶端會透過位於 `tmp` 資料夾中的通訊端進行通訊。如果 `tmp` 資料夾在開發環境中遺失，或者套用了過於嚴格的許可，**tmux** 工作階段就無法執行。如果發生這種情況，IDE 中的終端機視窗將不會有回應。

建議的解決方案：如果 **tmux** 錯誤使您無法與終端機視窗互動，請使用替代方式來建立具有適當許可的 `tmp` 資料夾。這樣，**tmux** 工作階段就能執行。其中一種解決方法是在 `.bash_profile` 或 `.bashrc` 檔案中匯出 `LC_CTYPE`。另一項建議方法是使用 AWS Systems Manager 來設定主機管理組態。如此可以透過 Amazon EC2 主控台存取相關執行個體。

設定主機管理

1. 首先，在 AWS Cloud9 主控台，找到您環境執行個體的名稱。若要這麼做，請選擇 **Your environments** (您的環境) 中的相關面板，並選擇 **View details** (檢視詳細資訊)。在 **Environment details** (環境詳細資訊) 頁面中，選擇 **Go to Instance** (前往執行個體)。在 Amazon EC2 主控台中，確認您需要存取的執行個體名稱。
2. 現在前往 AWS Systems Manager 主控台，接著在導覽窗格中，選擇 **Quick Setup** (快速設定)。
3. 在 **Quick Setup** (快速設定) 頁面中，選擇 **Create** (建立)。
4. 對於 **Configuration types** (組態類型)，前往 **Host Management** (主機管理)，然後選擇 **Create** (建立)。
5. 對於 **Customize Host Management configuration options** (自訂主機管理組態選項)，在 **Targets** (目標) 區段中，選擇 **Manual** (手動)。
6. 選取您要存取的 EC2 執行個體，然後選擇 **Create** (建立)。

連線至執行個體並執行命令

Note

以下步驟適用於全新 EC2 主控台。

1. 在 Amazon EC2 主控台的導覽窗格中，選擇 Instances (執行個體)，然後選取您要連線的執行個體。
2. 選擇連線。

如果 Connect (連線) 並未啟用，您可能需要先啟動該執行個體。

3. 在 Connect to your instance (連線至您的執行個體) 窗格中，對於 Connection method (連線方法)，選擇 Session Manager (工作階段管理員)，然後選擇 Connect (連線)。
4. 在出現的終端機工作階段視窗中，輸入下列指令。這些命令會建立具有適當許可的 tmp 資料夾，讓 tmux 通訊端可供使用。

```
sudo mkdir /tmp
sudo chmod 777 /tmp
sudo rmdir /tmp/tmux-*
```

[\(回到頁首\)](#)

Amazon EC2

下一節概述了與 Amazon EC2 相關的疑難排解問題。

Amazon EC2 執行個體未自動更新

問題：最近的系統更新未自動套用到連接至 AWS Cloud9 開發環境的 Amazon EC2 執行個體。

原因：自動套用最近的系統更新可能導致您的程式碼或 Amazon EC2 執行個體在未經您事先知悉或核准的情況下發生行為異常。

建議解決方案：

依照 Amazon EC2 使用者指南中[更新執行個體軟體](#)的指示，定期對 Amazon EC2 執行個體套用系統更新。

若要在執行個體上執行命令，您可以從連接至執行個體的環境中使用 AWS Cloud9 IDE 的終端機工作階段。

或者，您可以使用 SSH 遠端存取公用程式 (例如 ssh 或 PuTTY) 連線至執行個體。若要這樣做，請從您的本機電腦中使用 ssh-keygen 或 PuTTYgen 之類的 SSH 金鑰對建立公用程式。從連接至執行個體的環境中使用 AWS Cloud9 IDE，將產生的公有金鑰存放於執行個體。接著使用 SSH 遠端存取公用程式連同產生的私密金鑰以存取執行個體。如需詳細資訊，請參閱公用程式的說明文件。

EC2 環境中的 AWS CLI 或 AWS-shell 錯誤：「The security token included in the request is invalid」(包含在請求中的安全權杖無效)

問題：當您嘗試在 EC2 環境的 AWS Cloud9 IDE 中使用 AWS Command Line Interface (AWS CLI) 或 AWS-shell 執行命令時，出現「The security token included in the request is invalid.」(包含在請求中的安全權杖無效。) 錯誤。

原因：如果您已啟用 AWS 受管臨時憑證，且發生下列其中一個事項，就可能產生無效的安全權杖：

- 您嘗試執行 AWS 受管臨時憑證不允許的命令。如需允許命令的清單，請參閱 [AWS 受管理暫時認證支援的動作](#)。
- AWS 受管臨時憑證會在 15 分鐘後自動過期。
- 共用環境的 AWS 受管臨時憑證遭停用，因為有環境擁有者以外的他人新增了成員。

建議解決方案：

- 只執行 AWS 受管臨時憑證允許的命令。如果您必須執行 AWS 受管臨時憑證不允許的命令，則可使用一組永久憑證在環境中設定 AWS CLI 或 AWS-shell，如此便能移除這個限制。這可消除此限制。如需說明，請參閱 [在環境中建立並存放永久存取憑證](#)。
- 憑證遭到停用或過期時，請確保環境擁有者開啟環境，如此一來 AWS Cloud9 才可重新整理環境中的臨時憑證。如需詳細資訊，請參閱 [控制 AWS 受管臨時憑證的存取權](#)。

無法連線到 EC2 環境，因為 VPC 的 IP 地址是由 Docker 使用

問題：使用 EC2 環境時，如果您在採用 IPv4 無類別域間路由 (CIDR) 區塊 172.17.0.0/16 的 Amazon VPC 中啟動 EC2 執行個體，則在您嘗試開啟該環境時，連線可能會停滯。

原因：Docker 使用稱為橋接網路的連結層裝置，讓連接到相同橋接網路的容器進行通訊。AWS Cloud9 會建立使用預設橋接器的容器來進行容器通訊。預設橋接器通常會使用 172.17.0.0/16 子網路來進行容器網路連線。

如果環境執行個體的 VPC 子網路使用 Docker 已經使用的相同地址範圍，可能會發生 IP 地址衝突。因此，如果 AWS Cloud9 嘗試連線到其執行個體，則該連線會由閘道路由表路由到 Docker 橋接器。這可以防止 AWS Cloud9 連線到支援開發環境的 EC2 執行個體。

建議解決方案：若要解決 Amazon VPC 和 Docker 使用相同的 IPv4 CIDR 地址區塊所造成的 IP 地址衝突，請為支援 EC2 環境的執行個體設定新的 VPC。請為這個新的 VPC 設定有別於 172.17.0.0/16 的 CIDR 區塊。(您不能變更現有 VPC 或子網路的 IP 地址範圍。)

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 [VPC 和子網路規模調整](#)。

無法在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment。

問題：在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /home/ec2-user/environment/home/ec2-user/environment 時，您會收到一條錯誤訊息，表明無法開啟此目錄。

可能原因：目前無法使用 AWS Cloud9 IDE 的檔案系統，在相同名稱的資料夾內建立子資料夾結構 /home/ec2-user/environment。您將無法從 AWS Cloud9 IDE 檔案總管存取此目錄中的任何檔案，但是可以使用命令列存取這些檔案。此問題只影響檔案路徑 /home/ec2-user/environment/home/ec2-user/environment，/test/home/ec2-user/environment 和 /home/ec2-user/environment/test 等檔案路徑應該可以運作。這是已知問題，僅影響 AWS Cloud9 IDE 檔案總管。

建議解決方案：使用不同的檔案名稱和結構。

無法從主控台啟動 AWS Cloud9，原因是 AWS License Manager 授權組態與 Amazon EC2 執行個體相關聯

問題：當您嘗試從主控台啟動 AWS Cloud9 EC2 環境時，回報了一個 unable to access your environment 錯誤訊息。

可能原因：AWS License Manager 簡化了跨 AWS 雲端軟體廠商授權的管理。設定 License Manager 時，您可以建立授權組態，它們是以您的企業協議條款為根據的授權規則集。這些授權組態可以連接到某種機制，例如：Amazon Machine Image (AMI) 或 AWS CloudFormation。您可以使用其中一種機制來啟動 EC2 執行個體。

AWSServiceRoleForAWSCloud9 服務連結角色 (SLR) 的 AWSCloud9ServiceRolePolicy 舊版目前不包含 license-configuration 資源條件。因此，不允許 AWS Cloud9 啟動和停止其執行個體。因此 AWS Cloud9 被拒絕存取其 Amazon EC2 執行個體，並傳回錯誤。

建議的解決方案：如果您無法存取現有AWS Cloud9環境並使用 License Manager，請將舊的AWSCloud9ServiceRolePolicy服務連結角色取代為 [SLR 版本](#)，該版本會在license-configuration套用至執行個體時明確允許 EC2 動作。您可以透過直接刪除來取代舊角色。然後，更新過的角色會自動建立。

無法在 EC2 環境中執行某些命令或指令碼

問題：開啟 AWS Cloud9 EC2 開發環境後，您無法安裝某些類型的套件、執行 yum 或 apt 等命令，或是執行含有 (通常會與 Linux 作業系統搭配運作的) 命令的指令碼。

原因：AWS Cloud9 用於 EC2 環境的 Amazon EC2 執行個體倚賴 Amazon Linux (其架構基礎為 Red Hat Enterprise Linux (RHEL)) 或 Ubuntu Server。

解決方案：如果您在 EC2 環境的 IDE 中安裝或管理套件，或執行命令或指令碼，請視環境的執行個體而定，確認這些項目可與 RHEL (適用於 Amazon Linux) 或 Ubuntu Server 相容。

使用創建 EC2 環境時，錯誤消息報告「實例配置文 AWSCloud9SSMInstanceProfile 件不存在於帳戶中」AWS CloudFormation

問題：使用 [AWS::Cloud9::EnvironmentEC2](#) AWS CloudFormation 資源建立 EC2 環境時，使用者會收到帳戶中 AWSCloud9SSMInstanceProfile 不存在執行個體設定檔的錯誤訊息。

原因：建立無輸入 EC2 環境時，您必須建立服務角色 AWSCloud9SSMAccessRole 和執行個體描述檔 AWSCloud9SSMInstanceProfile。這些 IAM 資源可讓 Systems Manager 管理可為開發環境提供支援的 EC2 執行個體。

如果您使用主控台建立無輸入環境，AWSCloud9SSMAccessRole 和 AWSCloud9SSMInstanceProfile 會自動建立。但是當您使用 AWS CloudFormation 或 AWS CLI 來建立第一個無輸入環境時，您必須手動建立這些 IAM 資源。

建議解決方案：如需了解如何編輯 AWS CloudFormation 範本和更新 IAM 許可，請參閱 [使用 AWS CloudFormation 建立無輸入 EC2 環境](#)

使用 AWS CloudFormation 建立 EC2 環境時，有錯誤訊息回報「未授權對資源執行 **perform: ssm:StartSession**」

問題：使用 [AWS::Cloud9::EnvironmentEC2](#) AWS CloudFormation 資源建立 EC2 環境時，使用者會收到AccessDeniedException並通知他們「未授權執行：ssm:StartSession在資源上」。

原因：使用者缺少呼叫 StartSession API 的許可，此許可對於使用 Systems Manager 管理無輸入執行個體的 EC2 環境而言，是組態中必需的部分。

建議解決方案：如需了解如何編輯 AWS CloudFormation 範本和更新 IAM 許可，請參閱 [使用 AWS CloudFormation 建立無輸入 EC2 環境](#)。

使用 AWS CLI 建立 EC2 環境時，有錯誤訊息回報未授權「在資源：執行個體設定檔 **AWSCloud9SSMInstanceProfile** 上執行：**iam:GetInstanceProfile**」

問題：使用建立 EC2 環境時，使用者會收到 AccessDeniedException 並通知其 AWS Cloud9 環境未獲授權「GetInstanceProfile 在資源：執行個體設定檔上執行 iam：」AWSCloud9SSMInstanceProfile。 [AWS CLI](#)

原因：AWS Cloud9 缺少呼叫 StartSession API 的許可，此許可對於使用 Systems Manager 管理無輸入執行個體的 EC2 環境而言，是組態中必需的部分。

建議解決方案：如需有關新增所需 AWSCloud9SSMAccessRole 服務角色和 AWSCloud9SSMInstanceProfile 到您的 AWS Cloud9 環境的資訊，請參閱 [使用 AWS CLI 管理 Systems Manager 的執行個體描述檔](#)。

將預設加密套用至 Amazon EBS 磁碟區時，無法建立環境

問題：嘗試建立 Amazon EC2 環境時傳回 Failed to create environments. The development environment '[environment-ID]' failed to create 錯誤。

可能原因：如果您的 AWS Cloud9 IDE 使用預設為已加密的 Amazon EBS 磁碟區，AWS Cloud9 的 AWS Identity and Access Management 服務連結角色需要針對這些 EBS 磁碟區存取 AWS KMS keys。如果未提供存取權，AWS Cloud9 IDE 可能無法啟動，且可能也難以進行除錯。

建議解決方案：若要提供存取權，請將 AWS Cloud9、AWSServiceRoleForAWSCloud9 的服務連結角色新增至 Amazon EBS 磁碟區所使用的客戶受管金鑰。

如需此任務的詳細資訊，請參閱 AWS 規定指引模式中的 [建立以預設加密方式使用 Amazon EBS 磁碟區的 AWS Cloud9](#)。

EC2-Classic 帳戶的 VPC 錯誤：「無法存取您的環境」

問題：EC2-Classic 導入了 Amazon EC2 的原始版本。如果您使用的是 2013 年 12 月 4 日以前設定的 AWS 帳戶，且您在建立 AWS Cloud9 EC2 開發環境時未設定 Amazon VPC 和子網路，就可能發生此錯誤。

如果您接受預設 VPC 設定，則 Amazon EC2 執行個體會在 EC2-Classic 網路中啟動。此執行個體不會啟動至預設 VPC 的子網路。當建立環境失敗時會顯示下列訊息：

環境錯誤

無法存取您的環境

建立環境失敗，出現錯誤：下列資源無法建立：[執行個體].. 使用者請求轉返..

您可以確認錯誤是因為 EC2 執行個體不在預設 VPC 中所造成的。使用 AWS CloudFormation 檢視開發環境的堆疊事件歷程記錄。

1. 開啟 AWS CloudFormation 主控台。如需詳細資訊，請參閱[登入 AWS CloudFormation 主控台](#)。
2. 在 AWS CloudFormation 主控台，選擇 Stacks (堆疊)。
3. 在 Stacks (堆疊) 頁面上，選擇無法建立的開發環境名稱。
4. 在 Stack details (堆疊詳細資訊) 頁面上，選擇 Events (事件) 標籤，然後檢查下列項目：

狀態：CREATE_FAILED

狀態原因：僅 VPC 啟動支持該 AssociatePublicIpAddress 參數。 [...]

原因：AWS Cloud9 開發環境必須與符合特定 VPC 需求的 Amazon VPC 相關聯。如果是啟用了 EC2-Classic 的帳戶，在[建立 EC2 環境](#)時接受預設網路設定，則所需的 EC2 執行個體不會在 VPC 中啟動。相反地，該執行個體會在 EC2-Classic 網路中啟動。

建議解決方案：使用 EC2-Classic 帳戶，您必須在[建立 EC2 環境](#)時選取 VPC 和子網路。在 Configure settings (配置設定) 頁面的 Network settings (advanced) (網路設定 (進階)) 區段中，選取您可以在 VPC 和子網路中啟動的 EC2 執行個體。

其他 AWS 服務

下節概述了與其他 AWS 服務相關的疑難排解問題。

無法在 IDE 的文件資源管理器中創建子文件夾結構/項目/項目。AWS Cloud9 CodeCatalyst

問題：當您在 AWS Cloud9 IDE 檔案總管中建立子資料夾結構 /專案/專案時 CodeCatalyst，您會收到一則錯誤訊息，指出無法開啟此目錄。

可能的原因：目前無法使用 AWS Cloud9 IDE 的檔案總管，在相同名稱的資料夾內建立子資料夾結構/專案。CodeCatalyst 您將無法從 AWS Cloud9 IDE 檔案總管存取此目錄中的任何檔案，但是可以使用命令列存取這些檔案。此問題只會影響檔案路徑 /projects/projects，/test/projects 和 /projects/test 等檔案路徑應該可以運作。這是已知的問題，AWS Cloud9 只會影響 CodeCatalyst。

建議解決方案：使用不同的檔案名稱和結構。

無法從 IDE 外部顯示執行中的應用程式

問題：當您或其他人嘗試從 IDE 外部由 Web 瀏覽器標籤顯示執行中的應用程式時，Web 瀏覽器標籤顯示錯誤或一片空白。

可能原因：

- 應用程式未在 IDE 中執行。
- 應用程式使用 IP 127.0.0.1 或 localhost 執行。
- 應用程式正在 AWS Cloud9 EC2 環境中執行。此外，與對應的 Amazon EC2 執行個體相關聯的一個或多個安全群組不允許透過應用程式所需的通訊協定、連接埠或 IP 地址傳入流量。
- 應用程式正在 AWS 雲端運算執行個體 (例如 Amazon EC2 執行個體) 的 AWS Cloud9 SSH 開發環境中執行。此外，與對應的執行個體相關聯虛擬私有雲端 (VPC) 中的子網路設有網路 ACL，不允許透過應用程式所需的通訊協定、連接埠或 IP 地址傳入流量。
- URL 不正確。
- 請求的 URL 是應用程式預覽標籤上的網址，而不是執行個體的公有 IP 地址。
- 您正嘗試前往的 IP 地址包含 127.0.0.1 或 localhost。這些 IP 會嘗試存取本機電腦上的資源，而非環境中的資源。
- 執行個體的公有 IP 地址已變更。
- Web 請求源自於虛擬私有網路 (VPN)，會封鎖透過應用程式所需的通訊協定、連接埠或 IP 地址傳送流量。
- 應用程式正在 SSH 環境中執行。此外，伺服器或相關聯的網路並不允許透過應用程式所需的通訊協定、連接埠或 IP 地址傳送流量。

建議解決方案：

- 確認應用程式是在 IDE 中執行。
- 確認應用程式未使用 IP 127.0.0.1 或 localhost 執行。如需 Node.js 和 Python 方面的範例，請參閱[執行應用程式](#)。
- 假設應用程式是在 AWS 雲端運算執行個體 (例如 Amazon EC2 執行個體) 中執行。接著，確認對應的執行個體其相關聯的所有安全群組皆允許透過應用程式所需的通訊協定、連接埠及 IP 地址傳入流量。如需相關指示，請參閱透過網際網路共用執行中的應用程式中的[步驟 2：為執行個體設定安全群組](#)。另亦請參閱 Amazon VPC 使用者指南中的[VPC 的安全群組](#)。
- 假設應用程式是在 AWS 雲端運算執行個體中執行。此外，VPC 中與對應執行個體相關聯的子網路存在網路 ACL。接著，確認網路 ACL 允許透過應用程式所需的通訊協定、連接埠及 IP 地址傳入流量。如需相關指示，請參閱透過網際網路共用執行中的應用程式中的[步驟 3：為執行個體設定子網路](#)。亦請參閱 Amazon VPC 使用者指南中的[網路 ACL](#)。
- 確認請求的 URL 正確無誤，包括通訊協定且可能必須指定連接埠。如需詳細資訊，請參閱透過網際網路共用執行中的應用程式中的[步驟 4：共用執行中應用程式的 URL](#)。
- 我們不建議使用 `https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/` 格式來請求 URL (其中的 12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 指派給環境的 ID，而 us-east-2 是環境的 AWS 區域 ID)。此 URL 僅限於環境的 IDE 已開啟且應用程式由同一個 Web 瀏覽器執行時才有效。
- 假設您正嘗試前往的 IP 地址包含 127.0.0.1 或 localhost。改為嘗試前往正確的執行中應用程式非本機地址。如需詳細資訊，請參閱[透過網際網路共用執行中的應用程式](#)。
- 假設應用程式是在 AWS 雲端運算執行個體中執行。查明執行個體的公有 IP 地址是否已變更。執行個體的公有 IP 地址可能因執行個體重新啟動而隨時變更。為避免此 IP 地址發生變更，您可以配置彈性 IP 地址，然後將該地址指派給運作中的執行個體。如需詳細資訊，請參閱透過網際網路共用執行中的應用程式中的[步驟 4：共用執行中應用程式的 URL](#)。
- 若 Web 請求源自於 VPN，請確認該 VPN 允許透過應用程式所需的通訊協定、連接埠及 IP 地址傳送流量。若您無法對該 VPN 進行變更，請洽詢您的網路管理員。或者如果可行，從其他網路提出 Web 請求。
- 假設應用程式在您自己伺服器的 SSH 環境中執行。確認您的伺服器和相關聯的網路允許透過應用程式所需的通訊協定、連接埠及 IP 地址傳送流量。若您無法對伺服器或相關聯的網路進行變更，請洽詢您的伺服器或網路管理員。
- 執行 `curl` 命令，後面加上 URL，藉此嘗試從環境的終端機執行應用程式。如果此命令顯示錯誤訊息，可能另有一些其他與 AWS Cloud9 無關的問題存在。

執行 AWS Toolkit 時發生錯誤：「Your environment is running out of inodes, please increase 'fs.inotify.max_user_watches' limit.」(您的環境即將用盡 Inode，請提高『fs.inotify.max_user_watches』的上限。)

問題：AWS Toolkit 使用的檔案監看員公用程式即將接近目前可監看的檔案數量上限或配額。

原因：AWS Toolkit 會使用檔案監看員公用程式來監視檔案和目錄的變更。當公用程式幾乎達到目前可監看的檔案數量配額時，會顯示警告訊息。

建議解決方案：若要增加檔案監看員可處理的最大檔案數量，請執行下列動作：

1. 在選單列上選擇 Window (視窗)、New Terminal (新增終端機)，啟動終端機工作階段。
2. 輸入以下命令。

```
sudo bash -c 'echo "fs.inotify.max_user_watches=524288" >> /etc/sysctl.conf' &&  
sudo sysctl -p
```

Lambda 本機函數執行錯誤：無法安裝 SAM Local

問題：您嘗試在 AWS Cloud9 IDE 中執行本機版本的 AWS Lambda 函數，顯示一個對話方塊。對話方塊會指出在 AWS Cloud9 安裝 SAM 本機時遇到問題。AWS Cloud9 需要 SAM 本地地在 IDE 中運行 AWS Lambda 函數的本地版本。除非安裝 SAM Local，否則您無法在 IDE 中執行本機版本的 Lambda 函式。

原因：AWS Cloud9 在環境的預期路徑 (即 ~/.c9/bin/sam) 找不到 SAM Local。這是因為 SAM Local 尚未安裝，或已安裝但 AWS Cloud9 未能在上述位置找到該元件。

建議解決方案：您可以等待 AWS Cloud9 嘗試完成 SAM Local 的安裝，或者也可自行安裝。

若要查看 AWS Cloud9 嘗試安裝 SAM Local 的情形，請從選單列選擇 Window, Installer (視窗、安裝程式)。

若要自行安裝 SAM Local，請遵循 AWS Serverless Application Model 開發人員指南中的 [在 Linux 上安裝 AWS SAM CLI](#) 進行操作。

AWS Control Tower嘗試使用以下方式創建 Amazon EC2 環境時出錯AWS Cloud9：「環境創建失敗，出現錯誤：以下鉤子失敗：[:ControlTower: Guard:: Hook]」。

問題：存在與AWS Cloud9AWS Control Tower主動控制 CT.EC2.PR. 8 的相容性問題。如果啟用此控制項，則無法在中建立 EC2 環境AWS Cloud9。

原AWS Control Tower因：期望AssociatePublicIpAddress參數位於AWS CloudFormation模板中。此參數目前無法新增。

建議的解決方案：從AWS Control Tower主控台停用控制 CT.EC2.PR.8，然後在中重新建立環境。AWS Cloud9

將預設加密套用至 Amazon EBS 磁碟區時，無法建立環境

問題：嘗試建立 Amazon EC2 環境時傳回 Failed to create environments. The development environment '[environment-ID]' failed to create 錯誤。

可能原因：如果您的 AWS Cloud9 IDE 使用預設為已加密的 Amazon EBS 磁碟區，AWS Cloud9 的 AWS Identity and Access Management 服務連結角色需要針對這些 EBS 磁碟區存取 AWS KMS keys。如果未提供存取權，AWS Cloud9 IDE 可能無法啟動，且可能也難以進行除錯。

建議解決方案：若要提供存取權，請將 AWS Cloud9、AWSServiceRoleForAWSCloud9 的服務連結角色新增至 Amazon EBS 磁碟區所使用的客戶受管金鑰。

如需此任務的詳細資訊，請參閱 AWS 規定指引模式中的[建立以預設加密方式使用 Amazon EBS 磁碟區的 AWS Cloud9](#)。

[\(回到頁首\)](#)

無法從主控台啟動 AWS Cloud9，原因是 AWS License Manager 授權組態與 Amazon EC2 執行個體相關聯

問題：當您嘗試從主控台啟動 AWS Cloud9 EC2 環境時，回報了一個 unable to access your environment 錯誤訊息。

可能原因：AWS License Manager 簡化了跨 AWS 雲端軟體廠商授權的管理。設定 License Manager 時，您可以建立授權組態，它們是以您的企業協議條款為根據的授權規則集。這些授權組態可以連接到

某種機制，例如：Amazon Machine Image (AMI) 或 AWS CloudFormation。您可以使用其中一種機制來啟動 EC2 執行個體。

AWSServiceRoleForAWSCloud9 服務連結角色 (SLR) 的AWSCloud9ServiceRolePolicy舊版目前不包含license-configuration資源條件。因此，不允許 AWS Cloud9 啟動和停止其執行個體。因此 AWS Cloud9 被拒絕存取其 Amazon EC2 執行個體，並傳回錯誤。

建議的解決方案：如果您無法存取現有AWS Cloud9環境並使用 License Manager，請將舊的AWSCloud9ServiceRolePolicy服務連結角色取代為 [SLR 版本](#)，該版本會在license-configuration套用至執行個體時明確允許 EC2 動作。您可以透過直接刪除來取代舊角色。然後，更新過的角色會自動建立。

([回到頁首](#))

申請預覽

下節概述了與應用程式預覽相關的疑難排解問題。

重新載入環境後，您必須重新整理應用程式預覽

問題：在您重新載入顯示應用程式預覽標籤的環境後，該標籤並未顯示應用程式預覽。

原因：有時使用者編寫的程式碼會執行無限迴圈。或者其代碼可能會使用太多記憶體，以至於 AWS Cloud9 IDE 可能會在應用程式預覽執行時暫停或停止。為避免發生這種情況，每次環境重新載入時，AWS Cloud9 並不會重新載入應用程式預覽標籤。

解決方案：在您重新載入顯示應用程式預覽標籤的環境後，從該標籤上選擇 Click to load the page (按一下以載入頁面) 按鈕，即可顯示應用程式預覽。

應用程式預覽或檔案預覽通知：「Third-party cookies disabled」(已停用第三方 Cookie)

問題：當您嘗試預覽[應用程式](#)或[檔案](#)時，畫面會通知下列訊息：「Preview functionality is disabled because your browser has third-party cookies disabled.」(已停用預覽功能，因為您的瀏覽器已停用第三方 Cookie。)

原因：開啟 AWS Cloud9 IDE 不需要第三方 Cookie，不過，您必須啟用第三方 Cookie，才能使用「應用程式預覽」或「檔案預覽」功能。

解決方案：在 Web 瀏覽器中啟用第三方 Cookie、重新載入您的 IDE，然後嘗試再次開啟預覽。

- Apple Safari : Apple 支援網站上的[在 Safari 中管理 Cookie 和網站資料](#)。
- Google Chrome : Google Chrome 說明網站上，[在 Chrome 中清除、啟用與管理 Cookie](#) 中的更改您的 Cookie 設定。
- Internet Explorer : Microsoft 支援網站上，[刪除與管理 Cookie](#) 中的封鎖或允許 Cookie。
- Microsoft Edge : Microsoft 支援網站上的[封鎖第三方 Cookie](#)。
- Mozilla Firefox : Mozilla 支援網站上，[啟用與停用網站追蹤您的偏好設定時使用的 Cookie](#) 中的接受第三方 Cookie 設定。
- 其他 Web 瀏覽器 : 請參閱 Web 瀏覽器的文件。

您的 Web 瀏覽器必須允許此資料粒度，您才能為 AWS Cloud9 啟用第三方 Cookie。若要執行此操作，請根據您使用 AWS Cloud9 的支援 AWS 區域，指定下列網域。

AWS 區域	網域
美國東部 (維吉尼亞北部)	*.vfs.cloud9.us-east-1.amazonaws.com vfs.cloud9.us-east-1.amazonaws.com
美國東部 (俄亥俄)	*.vfs.cloud9.us-east-2.amazonaws.com vfs.cloud9.us-east-2.amazonaws.com
美國西部 (加利佛尼亞北部)	*.vfs.cloud9.us-west-1.amazonaws.com vfs.cloud9.us-west-1.amazonaws.com
美國西部 (奧勒岡)	*.vfs.cloud9.us-west-2.amazonaws.com vfs.cloud9.us-west-2.amazonaws.com

AWS 區域	網域
非洲 (開普敦)	<code>*.vfs.cloud9.af-south-1.amazonaws.com</code> <code>vfs.cloud9.af-south-1.amazonaws.com</code>
亞太區域 (香港)	<code>*.vfs.cloud9.ap-east-1.amazonaws.com</code> <code>vfs.cloud9.ap-east-1.amazonaws.com</code>
亞太區域 (孟買)	<code>*.vfs.cloud9.ap-south-1.amazonaws.com</code> <code>vfs.cloud9.ap-south-1.amazonaws.com</code>
亞太區域 (大阪)	<code>*.vfs.cloud9.ap-northeast-3.amazonaws.com</code> <code>vfs.cloud9.ap-northeast-3.amazonaws.com</code>
亞太區域 (首爾)	<code>*.vfs.cloud9.ap-northeast-2.amazonaws.com</code> <code>vfs.cloud9.ap-northeast-2.amazonaws.com</code>
亞太區域 (新加坡)	<code>*.vfs.cloud9.ap-southeast-1.amazonaws.com</code> <code>vfs.cloud9.ap-southeast-1.amazonaws.com</code>

AWS 區域	網域
亞太區域 (雪梨)	*.vfs.cloud9.ap-southeast-2.amazonaws.com vfs.cloud9.ap-southeast-2.amazonaws.com
亞太區域 (東京)	*.vfs.cloud9.ap-northeast-1.amazonaws.com vfs.cloud9.ap-northeast-1.amazonaws.com
加拿大 (中部)	*.vfs.cloud9.ca-central-1.amazonaws.com vfs.cloud9.ca-central-1.amazonaws.com
歐洲 (法蘭克福)	*.vfs.cloud9.eu-central-1.amazonaws.com vfs.cloud9.eu-central-1.amazonaws.com
歐洲 (愛爾蘭)	*.vfs.cloud9.eu-west-1.amazonaws.com vfs.cloud9.eu-west-1.amazonaws.com
歐洲 (倫敦)	*.vfs.cloud9.eu-west-2.amazonaws.com vfs.cloud9.eu-west-2.amazonaws.com

AWS 區域	網域
歐洲 (米蘭)	*.vfs.cloud9.eu-south-1.amazonaws.com vfs.cloud9.eu-south-1.amazonaws.com
歐洲 (巴黎)	*.vfs.cloud9.eu-west-3.amazonaws.com vfs.cloud9.eu-west-3.amazonaws.com
歐洲 (斯德哥爾摩)	*.vfs.cloud9.eu-north-1.amazonaws.com vfs.cloud9.eu-north-1.amazonaws.com
中東 (巴林)	*.vfs.cloud9.me-south-1.amazonaws.com vfs.cloud9.me-south-1.amazonaws.com
南美洲 (聖保羅)	*.vfs.cloud9.sa-east-1.amazonaws.com vfs.cloud9.sa-east-1.amazonaws.com

應用程式預覽標籤顯示錯誤或一片空白

問題：當您從 IDE 的選單列選擇 Preview、Preview Running Application (預覽、預覽執行中的應用程式) 或 Tools、Preview、Preview Running Application (工具、預覽、預覽執行中的應用程式) 嘗試讓 IDE 的預覽標籤顯示您的應用程式時，該標籤顯示錯誤或一片空白。

可能原因：

- 您的應用程式未在 IDE 中執行。
- 您的應用程式未使用 HTTP 執行。
- 您的應用程式透過多個連接埠執行。
- 您的應用程式透過 8080、8081 或 8082 以外的連接埠執行。
- 您的應用程式使用 127.0.0.1、localhost 或 0.0.0.0 以外的 IP 執行。
- 預覽標籤上的 URL 未指定連接埠 (8080、8081 或 8082)。
- 您的網路封鎖了對連接埠 8080、8081 或 8082 的傳入流量。
- 您正嘗試前往的 IP 地址包含 127.0.0.1、localhost 或 0.0.0.0。根據預設，AWS Cloud9 IDE 會嘗試前往您的本機電腦。它不會嘗試移至連線至環境的執行個體或您自己的伺服器。

建議解決方案：

- 確認應用程式是在 IDE 中執行。
- 確認應用程式是使用 HTTP 執行。如需 Node.js 和 Python 方面的範例，請參閱[執行應用程式](#)。
- 確認應用程式只透過單一連接埠執行。如需 Node.js 和 Python 方面的範例，請參閱[執行應用程式](#)。
- 確認應用程式是透過連接埠 8080、8081 或 8082 執行。如需 Node.js 和 Python 方面的範例，請參閱[執行應用程式](#)。
- 確認應用程式是使用 IP 127.0.0.1、localhost 或 0.0.0.0 執行。如需 Node.js 和 Python 方面的範例，請參閱[執行應用程式](#)。
- 為預覽標籤上的 URL 添加 :8080、:8081 或 :8082。
- 確認您的網路允許透過連接埠 8080、8081 或 8082 傳入流量。若您無法對網路進行變更，請洽詢您的網路管理員。
- 如果您嘗試移至包含 IP 127.0.0.1、localhost 或 0.0.0.0 的地址，請嘗試改為移至以下地址：`https://12a34567b8cd9012345ef67abcd890e1.vfs.cloud9.us-east-2.amazonaws.com/`。在這個地址中，12a34567b8cd9012345ef67abcd890e1 是 AWS Cloud9 指派給環境的 ID。us-east-2 是 AWS 區域 指派給環境的 ID。您也可以嘗試移到 IDE 之外的此地址。不過，這僅限於環境的 IDE 已開啟且應用程式由同一個 Web 瀏覽器執行時才有效。
- 確認符合上述所有條件後，嘗試停止應用程式並將其重新啟動。
- 如果您已停止應用程式並將其重新啟動，請再次嘗試從選單列選擇 Preview、Preview Running Application (預覽、預覽執行中的應用程式) 或 Tools、Preview、Preview Running Application (工具、預覽、預覽執行中的應用程式)。或者，若已能夠看到相應的應用程式預覽標籤，則由該標籤上選擇 Refresh (重新整理) 按鈕 (圓形箭頭)。

無法在 IDE 中預覽 Web 內容，因為與網站的連線不安全

問題：當您嘗試存取 Web 內容 (例如 AWS Cloud9 EC2 環境中託管的網 WordPress 站) 時，IDE 預覽視窗無法顯示該內容。

可能原因：根據預設，您在 AWS Cloud9 IDE 應用程式預覽標籤中存取的所有 web 頁面會自動使用 HTTPS 通訊協定。如果頁面的 URI 具有不安全的 http 通訊協定，就會自動由 https 替代。而且您無法藉由手動將 https 改回 http 來存取不安全的內容。

建議解決方案：從您嘗試在 IDE 中預覽的網站中移除不安全的 HTTP 指令碼或內容。請按照 Web 伺服器或內容管理系統的指示，以取得實作 HTTPS 的指引。

預覽檔案時傳回 499 錯誤

問題：當您嘗試使用 AWS Cloud9 IDE 來預覽包含 `<script>` 元素的檔案，而該元素又包含 `src` 屬性，且 `type` 屬性設為 `module` 時，發生 499 錯誤，且指令碼未如預期般執行。

原因：AWS Cloud9 IDE 中的檔案預覽擷取請求需要經由 Web 瀏覽器傳送 Cookie，才能進行驗證。根據預設，Web 瀏覽器會針對一般指令碼請求傳送 Cookie。它們不會針對模組指令碼請求傳送 Cookie，除非您新增 `crossorigin` 屬性。

解決方案：將 `crossorigin` 屬性新增至 `<script>` 元素。例如 `<script type="module" src="index.js" crossorigin></script>`。然後，儲存已變更的檔案，並嘗試再次進行預覽。

效能

下節概述了與效能相關的疑難排解問題。

AWS Cloud9IDE 凍結大量時間

問題：在啟動期間以及執行重新整理時，AWS Cloud9IDE 終端機會凍結大量時間，因此無法使用。

原因：您的環境中可能有大量檔案，這些檔案正在被的AWS Cloud9檔案監看模組遞迴監視。

建議的解決方案：您可以減少檔案監看深度 (最小值為 1)，並考慮將與原始程式碼無關的大型資料夾或資料夾 (建置輸出/成品、協力廠商套件) 新增至忽略的模式。若要執行此操作，請導覽至喜好設定 > 使用者設定 > 檔案監視 請注意，這將導致 CodeLenses 工具包無法正常工作。

另一個可能的解決方案是考慮忽略與源代碼無關的大文件和文件夾，方法是減少要搜索的文件的最大數量。若要執行此操作，請導覽至「偏好設定 > 專案設定 > 多檔尋找」。請注意，這將導致忽略的資料夾不會顯示在檔案搜尋中。

主控台警告：「Switching to the minimal code completion engine...」(切換到最少程式碼完成引擎...)

問題：在 AWS Cloud9 主控台工作時 (例如，開啟 IDE 或重新整理 IDE Web 頁面時)，您會看到此訊息：「一或多個工作階段或協作者在此環境中處於作用中的狀態。切換到最少程式碼完成引擎以節省記憶體。」與此訊息相互關聯的程式碼完成行為可能會變慢或斷斷續續。

原因：執程式碼完成引擎會需要環境中的記憶體和 CPU 週期。此外，每個協作者和每個額外的工作階段都需要個別的程式碼完成引擎。為了避免使用太多資源，特別是在 t2.nano 和 t2.micro 之類的小型執行個體大小，AWS Cloud9 會切換到最少程式碼完成引擎。

建議解決方案：如果您計畫經常或長時間進行協作，請在建立 EC2 環境時選擇較大的 Amazon EC2 執行個體。或者，您也可以將 SSH 環境連線到容量更大的執行個體。

Note

選擇較大的 Amazon EC2 執行個體可能會使您的 AWS 帳戶產生額外費用。如需詳細資訊，請參閱 [Amazon EC2 定價](#)。

IDE 警告：「This environment is running low on memory」(此環境的記憶體過低) 或 「This environment has high CPU load」(此環境有高負載 CPU)

問題：執行 IDE 時，您會看到包含以下文字的訊息：「This environment is running low on memory」(此環境的記憶體過低) 或 「This environment has high CPU load」(此環境有高負載 CPU)。

原因：此 IDE 可能沒有足夠的運算資源，無法在不延遲或當機的情況下繼續執行。

建議解決方案：

- 停用一或多個執程序以便釋放可用記憶體。若要停用，請在環境的 IDE 選單列中，選擇 Tools, Process List (工具、程序清單)。針對您想要停用的每個程序，選擇程序後，在選取 Force Kill (強制終止)。
- 在環境中建立置換檔。置換檔為環境中的檔案，可供作業系統當做虛擬記憶體。

若要確認環境目前是否正在使用交換記憶體，請在環境的終端機工作階段中執行 **top** 命令。如果正在使用置換記憶體，輸出顯示為非零的 Swap 記憶體統計資料 (例如：Swap: 499996k total, 1280k used, 498716 free, 110672k cached)。若要停止顯示即時記憶體資訊，請按 Ctrl + C。

若要建立置換檔，請在環境中執行如下命令。

```
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
```

上述命令執行以下事項：

1. 在 /var 目錄中，建立稱為 swapfile 的 512 MB 檔案。
2. 變更 swapfile 檔的存取許可，僅限擁有者讀取或寫入。
3. 將 swapfile 檔設定為置換檔。
4. 將資訊寫入 /etc/fstab file。這使得每當系統重新啟動時都可以使用此置換檔。

您執行上述命令後，若要立即讓置換檔可用，請執行下列命令。

```
sudo swapon /var/swapfile
```

- 使用多個運算資源將環境移動到 (或先調整環境的大小再移動) 執行個體或伺服器，。若要移動 Amazon EC2 執行個體或調整其大小，請參閱 [移動環境並調整 Amazon EBS 磁碟區大小或予以加密](#)。其他執行個體或伺服器類型，請參閱您的執行個體或伺服器文件。

無法在 AWS Cloud9 IDE 中上傳檔案

問題：使用者無法在 AWS Cloud9 IDE 中上傳大型檔案。這些上傳失敗。

原因：AWS Cloud9限制上傳速度到 AWS Cloud9 IDE，結果文件上傳請求超時。

建議的解決方案：我們建議將檔案上傳到 Amazon S3，然後使用 Amazon S3 透過 AWS Cloud9 IDE 中的 CLI 將檔案下載到環境中。如需將物件上傳到 Amazon S3 的詳細資訊，請參閱 Amazon S3 使用者指南中的[上傳物件](#)。

AWS Cloud9IDE 中的下載速度慢

問題：嘗試從 AWS Cloud9 IDE 下載檔案時，使用者正在處理較慢的下載速度。

原因：當您從 IDE 下載文件到本地文件系統時，傳輸速度將被限制為 0.1 兆字節/秒的速度。

建議的解決方案：若要提高檔案傳輸速度，請使用 AWS Cloud9 IDE 中的 CLI 將檔案上傳到 Amazon S3，然後使用 Amazon S3 從該處下載檔案。

無法在 IDE 中預覽 Web 內容，因為與網站的連線不安全

問題：當您嘗試存取 Web 內容 (例如 AWS Cloud9 EC2 環境中託管的網 WordPress 站) 時，IDE 預覽視窗無法顯示該內容。

可能原因：根據預設，您在 AWS Cloud9 IDE 應用程式預覽標籤中存取的所有 web 頁面會自動使用 HTTPS 通訊協定。如果頁面的 URI 具有不安全的 http 通訊協定，就會自動由 https 替代。而且您無法藉由手動將 https 改回 http 來存取不安全的內容。

建議解決方案：從您嘗試在 IDE 中預覽的網站中移除不安全的 HTTP 指令碼或內容。請按照 Web 伺服器或內容管理系統的指示，以取得實作 HTTPS 的指引。

[\(回到頁首\)](#)

第三方應用程式和服務

下節概述了與協力廠商應用程式和服務相關的疑難排解問題。

因為 tmux 工作階段發生錯誤，所以無法與 AWS Cloud9 中的終端機視窗互動。

問題：當您嘗試在 AWS Cloud9 中啟動新的終端機視窗，無法使用預期的命令列介面。沒有命令提示字元，您無法輸入文字。系統會傳回諸如 `tmux: need UTF-8 locale (LC_CTYPE)` 或 `invalid LC_ALL, LC_CTYPE or LANG` 等錯誤訊息。

可能原因：無回應終端機可能是由 tmux 錯誤所造成。AWS Cloud9 會使用 [tmux](#) 公用程式。如此，顯示在終端機中的資訊，即使在頁面重新載入或您重新連線至開發環境時也會持續存在。

在 tmux 工作階段中，終端機視窗中顯示的內容會由用戶端處理。用戶端會與可管理多個工作階段的伺服器通訊。伺服器和用戶端會透過位於 tmp 資料夾中的通訊端進行通訊。如果 tmp 資料夾在開發環境中遺失，或者套用了過於嚴格的許可，tmux 工作階段就無法執行。如果發生這種情況，IDE 中的終端機視窗將不會有回應。

建議的解決方案：如果 tmux 錯誤使您無法與終端機視窗互動，請使用替代方式來建立具有適當許可的 tmp 資料夾。這樣，tmux 工作階段就能執行。其中一種解決方法是在 `.bash_profile` 或 `.bashrc` 檔案中匯出 `LC_CTYPE`。另一項建議方法是使用 AWS Systems Manager 來設定主機管理組態。如此可以透過 Amazon EC2 主控台存取相關執行個體。

設定主機管理

1. 首先，在 AWS Cloud9 主控台，找到您環境執行個體的名稱。若要這麼做，請選擇 Your environments (您的環境) 中的相關面板，並選擇 View details (檢視詳細資訊)。在 Environment details (環境詳細資訊) 頁面中，選擇 Go to Instance (前往執行個體)。在 Amazon EC2 主控台中，確認您需要存取的執行個體名稱。
2. 現在前往 AWS Systems Manager 主控台，接著在導覽窗格中，選擇 Quick Setup (快速設定)。
3. 在 Quick Setup (快速設定) 頁面中，選擇 Create (建立)。
4. 對於 Configuration types (組態類型)，前往 Host Management (主機管理)，然後選擇 Create (建立)。
5. 對於 Customize Host Management configuration options (自訂主機管理組態選項)，在 Targets (目標) 區段中，選擇 Manual (手動)。
6. 選取您要存取的 EC2 執行個體，然後選擇 Create (建立)。

連線至執行個體並執行命令

Note

以下步驟適用於全新 EC2 主控台。

1. 在 Amazon EC2 主控台的導覽窗格中，選擇 Instances (執行個體)，然後選取您要連線的執行個體。
2. 選擇連線。

如果 Connect (連線) 並未啟用，您可能需要先啟動該執行個體。

3. 在 Connect to your instance (連線至您的執行個體) 窗格中，對於 Connection method (連線方法)，選擇 Session Manager (工作階段管理員)，然後選擇 Connect (連線)。
4. 在出現的終端機工作階段視窗中，輸入下列指令。這些命令會建立具有適當許可的 tmp 資料夾，讓 tmux 通訊端可供使用。

```
sudo mkdir /tmp
sudo chmod 777 /tmp
sudo rmdir /tmp/tmux-*
```



```
make --with-python=/usr/bin/python3 &&
```

4. 安裝除錯器。

```
sudo make -C gdb install
```

5. 確認已安裝除錯器的更新版本。

```
gdb --version
```

PHP 運行器的問題 AWS Cloud9

問題：使用者無法在 PHP CLI 執行程式終端機中檢視任何輸出。

原因:CLI 運行器需要設置為 PHP，並且需要啟用調試器模式。

推薦的解決方案：將 CLI 運行器設置為 PHP，並確保啟用調試器模式。

與 Node.js 相關的格林錯誤

問題：使用者無法執行 Node.js 並收到 GLIBC 錯誤。以下是這些錯誤訊息的範例：

```
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:可能是與正在使用的實例相關的 Node.js 版本問題。

建議的解決方案：如需有關如何安裝 Node.js 的資訊，請參閱[步驟 1：安裝必要工具](#)本節AWS Cloud9。

AWS Cloud9 支援的瀏覽器

下表會列出 AWS Cloud9 支援的瀏覽器。

瀏覽器	版本
Google Chrome	最近三個版本
Mozilla Firefox	最近三個版本
Microsoft Edge	最近三個版本
適用於 macOS 的 Apple Safari	最近兩個版本

Warning

如果您使用 Mozilla 火狐瀏覽器作為您首選的瀏覽器與AWS Cloud9 IDE，有一個第三方 cookie 設置，可以防止AWS Cloud9網絡視圖和工AWS具包在瀏覽器中正常工作。為了解決此問題，您必須確保您未在瀏覽器設定的「隱私權與安全性」區段中封鎖 Cookie，如下圖所示。

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 'Fingerprints' (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

下列資料表列出 AWS Cloud9 和相關 AWS 服務的限制。

- [AWS Cloud9 限制](#)
- [相關AWS服務限制](#)

AWS Cloud9 限制

下表提供 AWS 帳戶的 AWS Cloud9 預設限制。除非另有說明，否則每個限制都是區域特定規定。您可以使用AWS管理主控台或 AWS CLI 要求增加。若[要要求提高配額](#)，請參閱 [Service Quotas 使用者指南中的要求增加配額](#)。

提高的限制並不會立即獲准，因此可能需要花費幾天才能使其生效。

資源	預設限制	可調整
AWS Cloud9 EC2 開發環境的數量上限	<ul style="list-style-type: none"> • 每位使用者 100 個 • 每個帳戶 200 個 	是
SSH 環境數量上限	<ul style="list-style-type: none"> • 每位使用者 100 個 • 每個帳戶 200 個 	是
環境中的成員數量上限	<p>預設的成員數量上限等於該環境的執行個體記憶體除以 60 MB (結果會無條件捨去)。例如，具有 1 GiB 記憶體的執行個體最多可有 17 個成員 (1 GiB 除以 60 MB (無條件捨去))。</p> <p>如果 AWS Cloud9 無法判斷執行個體的記憶體，則會預設為與該執行個體相關聯的每個環境最多 8 個使用者。</p>	否 ¹

資源	預設限制	可調整
	環境的成員絕對數量上限為 25。	
可編輯檔案大小上限	8 MB	否

¹ 您可以[移動環境](#)來嘗試增加預設成員數量上限。不過，環境的成員絕對數量上限仍為 25。

AWS Cloud9IDE 下載限制

當您將文件從 AWS Cloud9 IDE 下載到本地文件系統時，傳輸速度將限制為 0.1 兆字節/秒的速度。若要提高檔案傳輸速度，請使用 AWS Cloud9 IDE 中的 CLI 將檔案上傳到 Amazon S3，然後使用 Amazon S3 從該處下載檔案。

相關 AWS 服務限制

Amazon Elastic Block Store (Amazon EBS) 磁碟區的數量上限	5,000 如需詳細資訊，請參閱《》中的「 Amazon Elastic Block Store (Amazon EBS) 限制 Amazon Web Services 一般參考」。
AWS CloudFormation 堆疊數目上限	200 如需詳細資訊，請參閱 AWS CloudFormation 使用者指南中的 AWS CloudFormation 限制 。
Amazon EC2 限制	請參閱《》中的「 Amazon Elastic Compute Cloud (Amazon EC2) 限制 Amazon Web Services 一般參考」。

AWS Cloud9 使用者指南的文件歷程記錄

本主題包含 AWS Cloud9 使用者指南的重大變更清單。如需有關此文件更新的通知，您可以訂閱 [RSS 摘要](#)。

最近更新

下表說明 2019 年 3 月之後 AWS Cloud9 使用者指南的重要變更。

變更	描述	日期
Support Amazon Linux 2023 AWS Cloud9 添加。	AWS Cloud9現在支持 Amazon 2023。	2023 年 12 月 15 日
對 Node.js 教學課程所做的更新。	已對與支援 Amazon Linux 2 和 Node.js 18 相關的 Node.js 教學進行了更新。	2023 年 10 月 23 日
更新了有關使用 Amazon VPC 儀表板創建 Amazon VPC 的部分	更新了有關使用 Amazon VPC 儀表板創建 Amazon VPC 的部分。	2023 年 7 月 27 日
有關使用 Amazon EventBridge 模式的部分	已新增使用的工AWS具組使用 Amazon EventBridge 架構描述的區段AWS Cloud9。	2022 年 12 月 15 日
CodeCatalyst 部分已添加	有關新 Amazon CodeCatalyst 服務的部分已添加。	2022 年 12 月 2 日
已新增 AWS IoT 內容	已新增有關使用 AWS IoT 的章節。	2022 年 11 月 1 日
適用於 AWS Cloud9 IDE 的 Amazon ECS 服務概觀	已新增可在 AWS Cloud9 IDE 中存取之 Amazon ECS 服務功能的概觀和逐步解說。	2022 年 10 月 20 日
在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CDK	關於在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS CDK 的一節。	2022 年 10 月 5 日

添加了 Amazon ECR 的內容	添加有關使用 AWS Amazon ECR 的部分。	2022 年 10 月 4 日
合規驗證	已更新 AWS Cloud9 範圍內的合規計劃清單。	2022 年 3 月 4 日
增強的 Java 支援	附加的語言支援，以改善您在使用 Java 時的開發體驗。關鍵的生產力功能包括程式碼完成、錯誤檢查、內容特定動作以及除錯選項 (如中斷點和逐步執行)。	2022 年 1 月 18 日
已更新 AWSServiceRoleForAWS Cloud 9	更新了服務連結角色以支援使用 License Manager 的 EC2 執行個體。	2022 年 1 月 12 日
Step Functions 文件支援	新增描述使用 Step Functions 建立、編輯和運行狀態機器的內容。	2021 年 12 月 20 日
AWS Systems Manager 文件支援	已新增介紹 Systems Manager 自動化文件的內容。	2021 年 12 月 20 日
建立了 Amazon Elastic Container Service Exec 的使用者指南	這是 Amazon ECS Exec 的概觀。	2021 年 12 月 13 日
已建立 AWS IoT AWS Cloud9 IDE 服務的使用者指南	本使用者指南介紹如何開始使用 AWS Cloud9 IDE 的 AWS IoT 服務。	2021 年 11 月 22 日
對 AWS 資源的支援	新增存取資源類型的支援，以及檢視資源和相關文件的介面選項。	2021 年 11 月 5 日
適用於 AWS Cloud9 IDE 的 Amazon ECR 服務概觀	已新增可在 AWS Cloud9 IDE 中存取的 Amazon ECR 服務特性和功能的概觀和逐步解說	2021 年 10 月 14 日

App Runner 支援	新增對 AWS App Runner 的支援至 AWS Toolkit。	2021 年 9 月 30 日
AWS Cloud9 也開放在非洲 (開普敦) 和亞太區域 (大阪) 使用	AWS Cloud9 現在也開放在以下區域使用：非洲 (開普敦) 和亞太區域 (大阪)。如需有關與這些區域和其他AWS區域相關聯之服務端點和服務配額的詳細資訊，請參閱 AWS Cloud9 中的Amazon Web Services 一般參考。	2021 年 9 月 1 日
CloudWatch AWS工具包中的日誌和 Amazon S3	增加了對 CloudWatch 日誌 AWS的支持AWS Cloud9。新功能允許將目前的檔案上傳至 Amazon S3 儲存貯體。	2021 年 7 月 16 日
Amazon S3 的 VPC 端點	新增將 Amazon S3 的 VPC 端點設為可下載相依性的支援功能。	2021 年 4 月 22 日
可透過 Git 面板使用的視覺化來源控制	如果您是開發人員，則可以使用 Git 面板在使用者界面中執行 Git 命令。	2021 年 2 月 1 日
在私有子網路中啟動環境執行個體	為透過 Systems Manager 存取的 EC2 執行個體新增支援功能，以便在私有子網路中啟動。	2021 年 1 月 21 日
AWS Toolkit 的整合	您現在可以透過 AWS Explorer 視窗，使用 AWS Toolkit 來導覽 AWS 服務 並與之互動。	2020 年 12 月 11 日
AWS CloudFormation 和無輸入 EC2 環境	使用 AWS CloudFormation 範本建立無輸入 EC2 環境的相關文件已擴充內容。	2020 年 10 月 29 日

以 Amazon Linux 2 為基礎的 EC2 環境	您主控台中建立 EC2 環境時，可以為 EC2 執行個體選擇 Amazon Linux 2 AMI。	2020 年 10 月 7 日
使用 Systems Manager 的無輸入 EC2 執行個體	新增了使用 AWS Systems Manager 存取私有 EC2 執行個體的支援功能。	2020 年 8 月 12 日
強化的 AWS 無伺服器應用程式本機偵錯功能	已為 AWS 無伺服器應用程式支援新的本機偵錯功能。	2020 年 7 月 30 日
AWS Cloud9 也開放在歐洲區域 (米蘭) 使用	AWS Cloud9 現在也開放在歐洲區域 (米蘭) 使用。如需有關與此區域和其他AWS區域相關聯之服務端點和服務配額的詳細資訊，請參閱 AWS Cloud9 中的Amazon Web Services 一般參考。	2020 年 7 月 29 日
Amazon EBS 加密	此節說明如何為 AWS Cloud9 開發環境使用的 EC2 執行個體加密 Amazon EBS 磁碟區。	2020 年 7 月 3 日
新增 AWS Cloud9 的區域支援	AWS Cloud9 現在已在下列區域開放使用：美國西部 (加利佛尼亞北部)、亞太區域 (香港)、歐洲 (巴黎)、中東 (巴林) 和南美洲 (聖保羅)。如需有關與這些區域和其他AWS區域相關聯之服務端點和服務配額的詳細資訊，請參閱 AWS Cloud9 中的Amazon Web Services 一般參考。	2020 年 5 月 7 日
安全性	AWS Cloud9 使用者指南中已新增安全性章節。	2020 年 4 月 30 日

Tags (標籤)	使用標籤以幫助您控制對 AWS Cloud9 資源的存取，並協助您管理帳單資訊。	2020 年 1 月 22 日
新增 AWS Cloud9 的區域支援	AWS Cloud9 現在已在下列區域開放使用：亞太區域 (孟買)、亞太區域 (首爾)、亞太區域 (雪梨)、加拿大 (中部)、歐洲 (倫敦) 和歐洲 (斯德哥爾摩)。如需有關與這些區域和其他AWS區域相關聯之服務端點和服務配額的詳細資訊，請參閱 AWS Cloud9 中的Amazon Web Services 一般參考。	2019 年 12 月 18 日
已更新：故障診斷、無法開啟環境	不再需要第三方 Cookie 來開啟 IDE。	2019 年 11 月 6 日
已新增：故障診斷中，已停用第三方 Cookie	不再需要第三方 Cookie 來開啟 IDE。不過，「應用程式預覽」或「檔案預覽」功能需要這些功能。您可以在「疑難排解」主題中找到有關此問題的資訊。	2019 年 11 月 6 日
文件組織	我們已將組織變更套用到使用者指南來協助導覽，特別適合第一次使用的人員。	2019 年 8 月 15 日
AWS Cloud9 也開放在歐洲區域 (法蘭克福) 使用	AWS Cloud9 現在也開放在歐洲區域 (法蘭克福) 使用。如需有關與此區域和其他AWS區域相關聯之服務端點和服務配額的詳細資訊，請參閱 AWS Cloud9 中的Amazon Web Services 一般參考。	2019 年 5 月 15 日

[新增了 LAMP 範例](#)

增加了新範例，示範如何搭配 LAMP (Linux、Apache HTTP Server、MySQL 及 PHP) 使用 AWS Cloud9。如需詳細資訊，請參閱 [AWS Cloud9 的 LAMP 範例](#)。

2019 年 5 月 10 日

[WordPress 樣品已添加](#)

已新增示範如何 AWS Cloud9 搭配 WordPress 使用的新範例。如需詳細資訊，請參閱的 [WordPress 範例 AWS Cloud9](#)。

2019 年 4 月 19 日

[AWS Cloud9 也開放在亞太區域 \(東京\) 使用](#)

AWS Cloud9 現在也開放在亞太區域 (東京) 使用。如需有關與此區域和其他 AWS 區域相關聯之服務端點和服務配額的詳細資訊，請參閱 [AWS Cloud9](#) 中的 Amazon Web Services 一般參考。

2019 年 4 月 4 日

[新增了 EC2 環境中 Ubuntu Server 支援的相關資訊](#)

新增了使用 AWS Cloud9 主控台建立可連線到 Ubuntu Server 的 AWS Cloud9 EC2 環境的指示。如需詳細資訊，請參閱 [建立 EC2 環境](#)。

2019 年 4 月 2 日

請注意，您目前無法使用程式碼來建立連線到 Ubuntu 伺服器的 AWS Cloud9 EC2 開發環境，例如 AWS CLI、AWS CloudFormation，透過使用、AWS 軟體開發套件 PowerShell、Windows 工具或 AWS Cloud9 API。未來預計將支援這些方法。

舊版更新

下表說明 2019 年 4 月前 AWS Cloud9 使用者指南的重要變更。

變更	Description	變更日期
新增了學生、教育工作者和企業適用的入門指示	AWS Cloud9 入門的指示已擴展為包含學生、教育工作者和企業適用的步驟。如需詳細資訊，請參閱 設定 AWS Cloud9 。	2019 年 2 月 7 日
新增了 AWS CloudTrail 支援	AWS CloudTrail 現在支援 AWS Cloud9。如需詳細資訊，請參閱 使用 AWS CloudTrail 記錄 AWS Cloud9 API 呼叫 。	2019 年 1 月 21 日
新增了共用的 VPC 支援	AWS Cloud9 現在支援在 Amazon VPC 共用 VPC。如需詳細資訊，請參閱 AWS Cloud9 的 Amazon VPC 需求 。	2018 年 12 月 7 日
AWS RoboMaker 加入整合	AWS Cloud9 現在支援 AWS RoboMaker 這項服務可讓您輕鬆大規模開發、測試和部署智慧型機器人應用程式。如需詳細資訊，請參閱 開 AWS RoboMaker 發 AWS RoboMaker 人員指南 AWS Cloud9 中的入門與開發 。	2018 年 11 月 26 日
新增了語言專案適用的其他生產力功能相關資訊	AWS Cloud9 IDE 現在可在語言專案當中提供某些語言適用的額外生產力功能。如需詳細資訊，請參閱 增強的 TypeScript 支援與功能 。	2018 年 10 月 2 日

變更	Description	變更日期
<p>新增了 Go (執行) 視窗，移除了 Navigate (導覽) 和 Commands (命令) 視窗</p>	<p>Go (執行) 視窗已新增到 2018 年 10 月 2 日當天或之後為環境所建立的 AWS Cloud9 IDE。這個新視窗取代了 Navigate (導覽) 和 Commands (命令) 視窗，這兩者均已在 2018 年 10 月 2 日當天或之後從為環境所建立的 IDE 中移除。如需詳細資訊，請參閱 導覽 IDE 中的 步驟 10：執行視窗。</p>	<p>2018 年 10 月 2 日</p>
<p>新增了 AWS CDK 範例</p>	<p>新增了範例，此範例會示範如何搭配 AWS Cloud9 Cloud Development Kit (AWS) 使用 AWS CDK。如需更多資訊，請參閱 AWS CDK 教學課程 AWS Cloud9。</p>	<p>2018 年 8 月 30 日</p>
<p>新增了自動新增到 EC2 環境的 SSH IP 地址限制相關資訊</p>	<p>針對 2018 年 7 月 31 日當天或之後所建立的 AWS Cloud9 EC2 開發環境，AWS Cloud9 現在會自動將傳入 SSH 流量限制在 AWS Cloud9 用於透過 SSH 連線的 IP 地址範圍內。如需詳細資訊，請參閱 AWS Cloud9 的傳入 SSH IP 地址範圍。</p>	<p>2018 年 7 月 31 日</p>
<p>新增了 Docker 範例</p>	<p>新增示範如何使用 AWS Cloud9 搭配 Docker 的新範例。如需更多資訊，請參閱 碼頭教程 AWS Cloud9。</p>	<p>2018 年 6 月 19 日</p>

變更	Description	變更日期
針對 Java、.NET 核心和新增的範例 TypeScript	已新增示範如何搭AWS Cloud9配 Java、.NET 核心和 TypeScript. 如需詳細資訊，請參閱 爪哇教學 AWS Cloud9 、 .NET 教學課程AWS Cloud9 和 TypeScript教學課程 AWS Cloud9 。	2018 年 5 月 29 日
新增了支援的瀏覽器清單	新增 AWS Cloud9 支援的瀏覽器資訊。如需詳細資訊，請參閱 AWS Cloud9 支援的瀏覽器 。	2018 年 5 月 23 日
新增了 SSH IP 流量限制資訊	新增如何限制傳入流量到 AWS Cloud9 用於透過 SSH 連線到主機 IP 地址範圍的資訊。如需詳細資訊，請參閱 AWS Cloud9 的傳入 SSH IP 地址範圍 。	2018 年 4 月 19 日
新增了預覽應用程式和共用執行中應用程式的故障診斷工具	新增預覽應用程式和共享執行中應用程式的故障診斷。如需更多詳細資訊，請參閱 應用程式預覽標籤顯示錯誤或一片空白 及 無法從 IDE 外部顯示執行中的應用程式 。	2018 年 4 月 19 日
新增了 File Revision History (檔案修訂歷史記錄) 資訊	新增使用 IDE 中的 File Revision History (檔案修訂歷史紀錄) 窗格的相關資訊。如需詳細資訊，請參閱在 AWS Cloud9 整合開發環境 (IDE) 中使用檔案修訂 。	2018 年 4 月 19 日

變更	Description	變更日期
<p>新增了開啟環境時的故障診斷工具</p>	<p>新增了開啟 AWS Cloud9 開發環境時專用的故障診斷工具。如需詳細資訊，請參閱無法開啟環境。</p>	<p>2018 年 3 月 19 日</p>
<p>新增了 AWS Cloud9 安裝程式的故障診斷工具</p>	<p>新增 AWS Cloud9 安裝程式的故障診斷。如需詳細資訊，請參閱AWS Cloud9 安裝程式停止回應或失敗。</p>	<p>2018 年 3 月 19 日</p>
<p>新增了 AWS CodePipeline 資訊</p>	<p>新增如何搭配 AWS CodePipeline 使用 AWS Cloud9 的相關資訊。如需詳細資訊，請參閱在 AWS Cloud9 整合開發環境 (IDE) 中使用 AWS CodePipeline。</p>	<p>2018 年 2 月 13 日</p>
<p>新增了 AWS CloudShell 資訊</p>	<p>新增了如何搭配使用 AWS Cloud9 與 AWS CloudShell 的相關資訊。如需更多資訊，請參閱AWS Command Line Interface和aws-shell 教程 AWS Cloud9。</p>	<p>2018 年 1 月 19 日</p>
<p>文檔可用性上 GitHub 添加</p>	<p>本指南現已在上提供 GitHub。您也可以針對本指南的內容提交意見反應和變更要求。GitHub 如需詳細資訊，請選擇快顯功能表導覽列中的「編輯於」GitHub 圖示，或參閱網站上的awsdocs/aws-cloud9 使用者指南儲存庫。GitHub</p>	<p>2018 年 1 月 10 日</p>

變更	Description	變更日期
Kindle 格式可用性	本指南現已在 Amazon Kindle 格式上可用。如需詳細資訊，請選擇快顯功能表導覽列中的「開啟 Kindle」圖示。	2018 年 1 月 2 日
新增 Amazon Lightsail 相關資訊	新增如何以 Amazon Lightsail 使用 AWS Cloud9 的相關資訊。如需詳細資訊，請參閱在 AWS Cloud9 整合開發環境 (IDE) 使用 Amazon Lightsail 執行個體 。	2017 年 12 月 19 日
新增 AWS 的環境設定說明	針對 AWS Cloud9 開發環境新增特定 AWS 設定的說明。如需詳細資訊，請參閱在 AWS Cloud9 整合式開發環境 (IDE) 中使用 AWS 專案和使用者設定 。	2017 年 12 月 7 日
新增了 AWS 帳戶根使用者適用的入門指示和小組適用的進階設定步驟	新增了以 AWS 帳戶根使用者的身分使用 AWS Cloud9 的設定步驟。為團隊使用的 AWS Cloud9 新增進階設定步驟。如需詳細資訊，請參閱 設定 AWS Cloud9 。	2017 年 12 月 5 日
針對環境需求擴充涵蓋範圍	已針對 Amazon EC2 執行個體或您的伺服器連線到 AWS Cloud9 SSH 開發環境的需求擴充了涵蓋範圍。如需詳細資訊，請參閱 SSH 環境主機要求 。	2017 年 12 月 4 日
初始文件版本	此為 AWS Cloud9 使用者指南的初始版本。	2017 年 11 月 30 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。