



使用者指南

AWS CloudFormation



API 版本 2010-05-15

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS CloudFormation: 使用者指南

Copyright © 2026 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 CloudFormation ?	1
簡化基礎設施管理	1
快速複製您的基礎設施	1
輕鬆控制和追蹤您基礎設施的變更	1
CloudFormation 入門	2
相關資訊	2
開始使用	3
CloudFormation 如何運作	3
重要概念	3
CloudFormation 如何運作	7
CloudFormation 入門方法	10
註冊 AWS 帳戶	10
註冊 AWS 帳戶	10
建立具有管理存取權的使用者	11
建立您的第一個堆疊	12
先決條件	13
使用主控台建立 CloudFormation 堆疊	13
監控堆疊建立	16
測試 Web 伺服器	17
疑難排解	17
清除	18
後續步驟	19
最佳實務	20
縮短回饋迴圈以提高開發速度	21
依生命週期和所有權組織您的堆疊	22
使用跨堆疊參考來傳回另一個堆疊匯出的輸出值	22
針對多帳戶和多區域部署使用 CloudFormation StackSets	23
驗證所有資源類型的配額	23
重複使用範本在多個環境中複製堆疊	23
使用模組以重複使用資源組態	23
採用基礎設施作為程式碼實務	24
請勿在您的範本中內嵌憑證	24
使用 AWS 特定的參數類型	24
使用參數限制條件	25

使用虛擬參數提升可攜性	25
使用 <code>AWS::CloudFormation::Init</code> 在 Amazon EC2 執行個體上部署軟體應用程式	26
使用最新的協助程式指令碼	26
驗證後再使用範本	26
驗證範本是否符合組織政策規範	26
使用 YAML 或 JSON 來撰寫範本	27
實作全面的標記策略	27
利用範本巨集進行進階轉換	28
透過 CloudFormation 管理所有堆疊資源	28
更新您的堆疊之前建立變更集	28
使用堆疊政策來保護資源	29
使用 AWS CloudTrail 記錄 CloudFormation 呼叫	29
使用程式碼檢閱和修訂控制管理您的範本	29
定期更新 Amazon EC2 執行個體	29
定期使用偏離偵測	29
設定自動復原的復原觸發條件	30
實作有效的堆疊重構策略	30
使用 CloudFormation Hooks 進行生命週期管理	30
使用 IaC 產生器從現有資源中建立範本	31
AWS Infrastructure Composer 用於視覺化範本設計	31
考慮將 AWS Cloud Development Kit (AWS CDK) 用於複雜的基礎設施	31
使用 IAM 控制存取	32
套用最低權限準則	32
保護敏感參數	32
使用 將政策實作為程式碼 AWS CloudFormation Guard	33
使用 範本	34
範本的儲存位置	35
驗證範本	35
開始使用範本	2
規劃使用 CloudFormation 範本參考	36
範例範本	36
範本格式	37
範本結構	37
說明	39
規格	40
進一步了解	40

使用常規表達式	40
範本區段	41
Resources	43
Parameters	51
Outputs	63
Mappings	68
Metadata	75
Rules	81
Conditions	101
Transform	112
格式版本	113
Description	113
Infrastructure Composer	114
為何使用 CloudFormation 主控台模式中的 Infrastructure Composer ?	114
此模式與 Infrastructure Composer 主控台有何差異 ?	114
AWS CloudFormation 語言伺服器	115
IDEs與 AWS CloudFormation 語言伺服器整合	115
開始使用	116
在 IDE 中初始化 CloudFormation 專案 (僅限 VS 程式碼)	121
開放原始碼	122
需要幫助 ?	122
IaC 產生器	122
考量事項	123
許可	124
常用命令	124
將範本遷移至 AWS CDK	124
啟動資源掃描	125
檢視掃描摘要	129
透過掃描的資源建立範本	130
透過掃描的資源建立堆疊	135
解決唯寫屬性	136
取得儲存在其他服務中的值	140
一般考量事項	140
取得 Systems Manager 純文字值	141
取得 Systems Manager 安全字串	144
取得 Secrets Manager 值	147

取得 AWS 值	151
語法	151
可用的虛擬參數	152
範例	154
取得堆疊輸出	158
匯出堆疊輸出值與使用巢狀堆疊的比較	158
考量事項	158
列出已匯出的輸出值	159
列出匯入匯出的輸出值的堆疊	160
在執行時期指定現有資源	161
概觀	161
範例	162
考量事項	165
支援 AWS 的特定參數類型	166
支援的 Systems Manager 參數類型	167
不支援的 Systems Manager 參數類型	168
逐步解說	168
參考在另一個 CloudFormation 堆疊中的資源輸出	169
在 Amazon EC2 上部署應用程式	173
更新堆疊	182
建置擴展和負載平衡的應用程式	198
與另一個帳戶中的 VPC 對等	211
執行 ECS 藍/綠部署	221
範本程式碼片段	243
一般	244
自動擴展	254
AWS 帳單主控台	285
CloudFormation	288
CloudFront	295
CloudWatch	307
CloudWatch Logs	313
DynamoDB	347
Amazon EC2	351
Amazon ECS	387
Amazon EFS	410
Elastic Beanstalk	435

Elastic Load Balancing	439
IAM	445
AWS Lambda	462
Amazon Redshift	468
Amazon RDS	478
Route 53	487
Amazon S3	495
Amazon SNS	503
Amazon SQS	504
Amazon Timestream	504
Windows 型堆疊	533
引導 Windows 堆疊	535
使用 CloudFormation 提供的資源類型	541
自訂資源	542
範本巨集	583
巢狀堆疊	599
等待條件	605
透過模組建立可重複使用的資源組態	610
使用模組時的考量	611
了解模組版本控制	612
在範本中使用模組	613
使用參數來指定模組值	614
參考模組資源	616
建立和管理堆疊	619
管理堆疊的介面	620
建立堆疊	621
建立堆疊	621
設定堆疊選項	624
預覽堆疊的組態	626
檢視堆疊資訊	627
更新您的堆疊範本	629
了解更新堆疊資源的行為	630
透過變更集更新堆疊	632
建立變更集	633
檢視變更集	638
偏離感知變更集	646

執行變更集	651
刪除變更集	652
變更集範例	654
巢狀堆疊變更集	666
驗證堆疊部署	672
部署前驗證的運作方式	672
考量事項	673
先決條件	673
驗證堆疊部署（主控台）	673
驗證堆疊部署 (AWS CLI)	674
驗證類型	676
資源限制	676
直接更新堆疊	691
取消堆疊更新	694
更新堆疊更新 (主控台)	694
取消堆疊更新 (AWS CLI)	695
刪除堆疊	695
相關資源	696
檢視已刪除的堆疊	697
監控堆疊進度	697
檢視堆疊事件	698
依操作檢視堆疊事件	705
檢視堆疊部署時間軸圖表	709
了解堆疊建立事件	712
監控堆疊更新	714
繼續復原更新	716
確定堆疊失敗的原因	719
堆疊失敗選項	720
警示違規時復原您的堆疊	729
新增堆疊建立或更新期間的復原觸發條件	730
將復原觸發條件新增至變更集	730
檢視堆疊的復原觸發條件	731
檢視變更集的復原觸發條件	731
偵測堆疊和資源的未受管組態變更	731
什麼是偏離？	732
漂移偵測狀態碼	733

偵測偏離時的考量	735
在整個 CloudFormation 堆疊上偵測偏離	737
在個別堆疊資源上偵測偏離	742
透過匯入作業解決漂移	745
匯入 AWS 資源	754
手動匯入 AWS 資源	755
自動匯入 AWS 資源	790
還原匯入操作	791
堆疊重構	796
堆疊重構的運作方式	797
堆疊重構考量事項	797
先決條件	798
重構堆疊 (主控台)	798
重構堆疊 (AWS CLI)	799
資源限制	804
資源類型支援	808
使用快速建立連結來建立 CloudFormation 堆疊	1003
URL 格式	1003
範例	1004
使用快速建立連結來建立堆疊	1005
AWS CLI 和 PowerShell 的範例命令	1006
CancelUpdateStack	1006
ContinueUpdateRollback	1007
CreateStack	1008
部署	1012
DeleteStack	1013
DescribeStackEvents	1014
DescribeStackResource	1016
DescribeStackResources	1017
DescribeStacks	1020
GetTemplate	1024
ListStackResources	1025
ListStacks	1027
UpdateStack	1029
ValidateTemplate	1032
上傳本機成品到 S3 儲存貯體	1035

透過 StackSets 管理堆疊	1038
StackSets 概念	1039
管理員和目標帳戶	1039
CloudFormation StackSets	1040
StackSets 的許可模型	1040
堆疊執行個體	1041
StackSet 操作	1041
StackSet 操作選項	1042
Tags (標籤)	1043
StackSets 狀態碼	1044
堆疊執行個體狀態碼	1045
先決條件	1046
啟用預設停用 AWS 區域 的	1046
授與自我管理許可	1047
啟用受信任存取	1062
使用範例範本入門	1066
先決條件	1067
從主控台使用範例範本建立 StackSet	1067
監控 StackSet 建立	1069
檢視 StackSet 結果	1070
更新您的 StackSet	1070
將堆疊新增至 StackSet	1071
清除	1071
後續步驟	1073
建立 StackSets (自我管理許可)	1073
建立具有自我管理許可的 StackSet (主控台)	1074
建立具有自我管理許可的 StackSet (AWS CLI)	1075
建立 StackSets (具有服務受管許可)	1076
考量事項	1077
建立具有服務受管許可的 StackSet (主控台)	1077
建立具有服務受管許可的 StackSet (AWS CLI)	1079
啟用-停用自動部署	1083
自動部署的運作方式	1083
考量事項	1083
啟用或停用自動部署 (主控台)	1084
啟用或停用自動部署 (AWS CLI)	1084

更新 StackSets	1085
更新您的 StackSet (主控台)	1085
更新您的 StackSet (AWS CLI)	1087
將堆疊新增至 StackSets	1089
將堆疊新增至 StackSet (主控台)	1089
將堆疊新增至 StackSet (AWS CLI)	1090
覆寫堆疊上的參數	1091
覆寫堆疊上的參數 (主控台)	1092
覆寫堆疊上的參數 (AWS CLI)	1093
從 StackSets 刪除堆疊	1094
從您的 StackSet 刪除堆疊 (主控台)	1095
從您的 StackSet 刪除堆疊 (AWS CLI)	1096
刪除 StackSets	1097
刪除 StackSet (主控台)	1098
刪除 StackSet (AWS CLI)	1098
刪除服務角色 (選用)	1098
目標帳戶闢道	1099
要求	1100
用於建立 Lambda 函式的 CloudFormation 範本	1100
選擇並行模式	1101
每個並行模式的運作方式	1101
根據部署速度，選擇嚴格容錯能力和軟性容錯能力	1104
選擇並行模式 (主控台)	1105
選擇並行模式 (AWS CLI)	1105
偵測 StackSets 上的漂移	1106
CloudFormation 在 StackSet 上執行漂移偵測的方式	1107
偵測 StackSet 上的漂移 (主控台)	1107
偵測 StackSet 上的漂移 (AWS CLI)	1109
停止 StackSet 上的漂移偵測	1115
將堆疊匯入 StackSets	1116
自我管理堆疊匯入	1116
服務受管堆疊匯入	1119
還原堆疊匯入	1124
最佳實務	1124
定義範本	1125
建立或將堆疊新增至 StackSet	1125

更新 StackSet 中的堆疊	1125
範例範本	1126
疑難排解	1127
堆疊操作失敗的常見原因	1128
重試失敗的堆疊建立或更新操作	1128
堆疊執行個體刪除失敗	1129
堆疊匯入操作失敗	1129
StackSets 操作的堆疊執行個體失敗計數	1130
使用 Git 原始程式碼來同步堆疊	1136
Git 同步的運作方式	1137
Git 同步的運作方式	1137
提取請求的註解	1138
堆疊部署檔案	1138
CloudFormation 範本檔案	1139
範本定義儲存庫	1139
先決條件	1140
Git 儲存庫	1140
CloudFormation 範本	1140
Git 同步服務角色	1140
主控台使用者的 IAM 許可	1144
從儲存庫原始程式碼建立堆疊	1146
從儲存庫原始程式碼建立堆疊	1147
從 Git 儲存庫中更新堆疊	1149
啟用提取請求的註解	1149
狀態儀表板	1150
Git 同步狀態	1150
最新同步事件	1151
支援的堆疊狀態	1151
使用 CloudFormation 登錄檔管理擴充功能	1155
相關文件	1155
概念	1156
擴充功能類型	1156
公有擴充功能	1156
已啟用的擴充功能	1157
檢視可用和已啟用擴充功能	1157
公有擴充功能	1158

IAM 角色	1159
自動使用擴充功能的新版本	1161
使用別名來稱呼擴充功能	1161
使用公有擴充功能的常用 AWS CLI 命令	1162
啟用公有擴充功能	1162
更新公有擴充功能	1166
停用公有擴充功能	1167
私有擴充功能	1168
IAM 許可	1168
使用私有擴充功能的常用 AWS CLI 命令	1169
註冊私有擴充功能	1169
取消註冊私有擴充功能	1171
編輯擴充功能的組態資料	1172
使用動態參考所需的許可	1173
編輯擴充功能的組態資料 (主控台)	1173
編輯擴充功能的組態資料 (AWS CLI)	1174
在 AWS Config 中記錄資源類型	1175
防止在組態項目中記錄感應性屬性	1176
持續交付	1177
逐步解說：建置管線以用於測試和生產堆疊	1177
先決條件	1177
逐步解說概觀	1178
步驟 1：編輯成品並將它上傳到 S3 儲存貯體	1178
步驟 2：建立管道堆疊	1180
步驟 3：檢視 WordPress 堆疊	1185
步驟 4：清除資源	1185
另請參閱	1186
組態屬性參考	1187
組態屬性 (主控台)	1187
組態屬性 (JSON 物件)	1189
另請參閱	1193
CloudFormation 成品	1194
堆疊範本檔案	1194
範本組態檔案	1194
另請參閱	1196
搭配 CodePipeline 管道使用參數覆寫函數	1196

Fn::GetArtifactAtt	1196
Fn::GetParam	1198
另請參閱	1200
範本參考指南	1201
安全	1202
防止堆疊遭到刪除	1202
控制誰可以在堆疊上變更終止保護	1204
避免更新堆疊資源	1205
範例堆疊政策	1205
定義堆疊政策	1206
設定堆疊政策	1210
更新受保護的資源	1211
修改堆疊政策	1214
更多範例堆疊政策	1215
資料保護	1218
靜態加密	1219
傳輸中加密	1219
網際網路流量隱私權	1219
控制 IAM 來控制存取	1219
定義 IAM 基於身分識別的政策	1219
認可 CloudFormation 範本中的 IAM 資源	1228
管理在 Amazon EC2 執行個體上執行之應用程式的憑證	1229
授予暫時存取 (聯合身分存取)	1229
身分型政策範例	1230
AWS 受管政策	1238
CloudFormation 服務角色	1240
預防跨服務混淆代理人	1241
FAS 請求和許可評估	1242
記錄 API 呼叫	1245
CloudTrail 中的 CloudFormation 資訊	1245
了解 CloudFormation 日誌檔案項目	1246
基礎設施安全性	1250
恢復能力	1250
法規遵循驗證	1251
組態與漏洞分析	1251
安全最佳實務	1251

使用 IAM 控制存取	1251
請勿在您的範本中內嵌憑證	1252
使用 AWS CloudTrail 記錄 CloudFormation 呼叫	1252
AWS PrivateLink	1252
CloudFormation VPC 端點的考量事項	1252
建立 的介面 VPC 端點 CloudFormation	1253
為 建立 VPC 端點政策 CloudFormation	1254
使用 EventBridge 進行監控	1256
CloudFormation 和 Git 同步事件概觀	1256
許可	1258
建立自訂事件模式	1258
事件詳細參照	1259
資源狀態變更	1260
堆疊狀態變更	1263
偏離偵測狀態變更	1265
StackSet 狀態變更	1267
StackSet 堆疊執行個體狀態變更	1268
StackSet 操作狀態變更	1271
儲存庫同步狀態變更	1273
資源同步狀態變更	1275
配額	1278
功能可用性	1283
StackSets 和巨集	1283
疑難排解	1284
故障診斷指南	1284
故障診斷錯誤	1285
刪除堆疊失敗	1285
相依性錯誤	1286
AWS Config 和 AWS Systems Manager 衝突	1286
傳遞清單時剖析參數出現錯誤	1287
IAM 許可不足	1287
無效的值或不支援的資源屬性	1287
超過配額	1287
巢狀堆疊復原失敗	1288
無需執行更新	1288
在建立、更新或刪除堆疊操作期間，資源無法穩定	1288

安全群組不存在於 VPC	1289
更新復原失敗	1289
等待條件未收到來自 Amazon EC2 執行個體所需數目的訊號	1290
從堆疊中移除但未刪除的資源	1291
聯絡支援	1291
使用 Amazon Q 進行故障診斷	1293
功能	1293
運作方式	1293
定價	1294
區域可用性	1294
限制	1294
文件歷史紀錄	1295
封存的更新	1311
.....	mcdv

什麼是 CloudFormation ?

CloudFormation 是一項服務，可協助您建立和設定 AWS 資源的模型，以減少管理這些資源的時間，並有更多時間專注於在 中執行的應用程式 AWS。您可以建立範本來描述您想要的所有 AWS 資源（例如 Amazon EC2 執行個體或 Amazon RDS 資料庫執行個體），而 CloudFormation 會負責為您佈建和設定這些資源。您不需要個別建立及設定 AWS 資源，並費心思考依存性，這些 CloudFormation 都能處理。以下案例示範 CloudFormation 可協助您的方式。

簡化基礎設施管理

針對同時包含後端資料庫的可擴展 Web 應用程式，您可以使用 Auto Scaling 群組、Elastic Load Balancing 負載平衡器和 Amazon Relational Database Service 資料庫執行個體。您可以使用每項個別服務來佈建這些資源，並且在您建立資源之後，您必須設定它們，才能讓它們一起運作。在您讓您的應用程式開始執行之前，這些任務可能會先讓一切變得更複雜且更耗費時間。

您可以改為建立或修改現有的 CloudFormation 範本。範本會描述您所有的資源及其屬性。當您使用該範本建立 CloudFormation 堆疊時，CloudFormation 會為您佈建 Auto Scaling 群組、負載平衡器和資料庫。堆疊成功建立後，您的 AWS 資源就會啟動並執行。您仍然可以輕易的刪除堆疊，同時刪除堆疊中所有的資源。使用 CloudFormation，您可以輕鬆將資源集合作為單一單位管理。

快速複製您的基礎設施

若您的應用程式需要其他可用性，您可能會在多個區域中複製它，以便在一個區域無法使用時，讓您的使用者仍然可以在其他區域使用您的應用程式。複製您應用程式的挑戰是，您必須同時複製您的資源。您不僅需要記錄所有您應用程式所需的資源，還要在每個區域內佈建及設定那些資源。

重複使用您的 CloudFormation 範本，以一致且可重複的方式建立資源。如需再次使用範本，只需要描述您的資源一次，然後在多個區域內重複佈建相同資源即可。

輕鬆控制和追蹤您基礎設施的變更

在某些情況下，您可能會有您希望累加升級的基礎資源。例如，您可能會在您的 Auto Scaling 啟動組態中變更至效能更高的執行個體類型，以減少您 Auto Scaling 群組中的最大執行個體數。若在您完成更新之後發生問題，您可能需要將您的基礎設施復原至原始設定。若要手動執行此作業，您不僅需要記得您變更了哪些資源，您也必須了解原始設定為何。

當您使用 CloudFormation 佈建您的基礎設施時，CloudFormation 範本會明確描述佈建了哪些資源以及其設定。因為這些範本都是文字檔案，您只需要追蹤您範本中的差異，便可追蹤您基礎設施的變更，與開發人員控制來源碼修訂的方式相似。例如，您可以搭配您的範本使用版本控制系統，以明確了解進行了哪些變更、變更者，以及變更的時間。若在其中任何一個時間點，您需要反轉您基礎設施的變更，您可以使用您範本的先前版本。

CloudFormation 入門

CloudFormation 可透過 CloudFormation [主控台](#)、[API](#)、[AWS CLI](#)、[AWS SDK](#)，以及透過多項整合來取得。

如需 CloudFormation 的簡介，請參閱 [CloudFormation 如何運作](#)。

要開始使用 CloudFormation，請參閱 [建立您的第一個堆疊](#)。

相關資訊

您可以在本使用者指南以及下列資源中進一步了解 CloudFormation：

- 如需產品詳細資訊和常見問題，請參閱 [AWS CloudFormation 產品頁面](#)。
- 如需定價資訊，請參閱 [AWS CloudFormation 定價](#)。

CloudFormation 入門

您可以透過從範例範本建立堆疊 AWS 管理主控台，開始透過使用 CloudFormation，這可協助您了解堆疊建立的基本概念。範本是定義堆疊內所有資源的文字檔案。堆疊是 CloudFormation 範本的部署。可以從單一範本建立多個堆疊。每個堆疊都包含可做為單一單位管理的一組 AWS 資源。

CloudFormation 是一項免費服務；不過，您需要按每個服務目前費率支付堆疊中包含 AWS 的資源費用。如需 AWS 定價的詳細資訊，請前往 <https://aws.amazon.com> 上每個產品的詳細資訊頁面。

主題

- [CloudFormation 如何運作](#)
- [註冊 AWS 帳戶](#)
- [建立您的第一個堆疊](#)

CloudFormation 如何運作

本主題說明 CloudFormation 的運作方式，並介紹使用 CloudFormation 時需要了解的重要概念。

主題

- [重要概念](#)
- [CloudFormation 如何運作](#)
- [CloudFormation 入門方法](#)

重要概念

當您使用 CloudFormation 時，即可搭配範本與堆疊使用。您可以建立範本來描述 AWS 資源及其屬性。每當建立堆疊時，CloudFormation 即會佈建範本中所述的資源。

主題

- [範本](#)
- [堆疊](#)
- [變更集](#)

範本

CloudFormation 範本屬於 YAML 或 JSON 格式的文字檔案。您可以以任何副檔名 (如 .yaml、.json、.template 或 .txt) 來儲存這些檔案。CloudFormation 使用這些範本做為建置 AWS 資源的藍圖。舉例來說，您可以在範本中說明 Amazon EC2 執行個體的執行個體類型、AMI ID、區塊型裝置映射與 Amazon EC2 金鑰對名稱等等。您在建立堆疊時還可以指定範本，讓 CloudFormation 能用來建立範本中所述的內容。

假若您採用下方範本建立堆疊，則 CloudFormation 在佈建執行個體時，會使用 ami-0ff8a91507f77f867 AMI ID、t2.micro 執行個體類型、testkey 金鑰對名稱與 Amazon EBS 磁碟區。

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A sample template",
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "t2.micro",
```



```

        "KeyName": "testkey",
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/sdm",
                "Ebs": {
                    "VolumeType": "io1",
                    "Iops": 200,
                    "DeleteOnTermination": false,
                    "VolumeSize": 20
                }
            }
        ]
    }
}
}
}
}

```

您也可以在單一範本中指定多個資源並加以設定，讓這些資源共同搭配運作。例如，您可以修改先前的範本以包括彈性 IP 地址 (EIP)，並將其與 Amazon EC2 執行個體相關聯，如以下範例所示：

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: A sample template
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
  MyEIP:
    Type: AWS::EC2::EIP
    Properties:
      InstanceId: !Ref MyEC2Instance

```

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A sample template",
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "t2.micro",
        "KeyName": "testkey",
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/sdm",
            "Ebs": {
              "VolumeType": "io1",
              "Iops": 200,
              "DeleteOnTermination": false,
              "VolumeSize": 20
            }
          }
        ]
      }
    },
    "MyEIP": {
      "Type": "AWS::EC2::EIP",
      "Properties": {
        "InstanceId": {
          "Ref": "MyEC2Instance"
        }
      }
    }
  }
}
```

儘管先前的範本是以單一 Amazon EC2 執行個體為中心，但 CloudFormation 範本還有提供其他的功能，讓您可以用來建構複雜的資源集，並在多種情況下重複使用這些範本。舉例來說，您可以新增輸入參數，其為建立 CloudFormation 堆疊時所指定的值。換言之，您不需等到建立範本期間才指定數值，建立堆疊期間即可指定執行個體類型等值，藉此在不同情況下更輕鬆地重複使用範本。

堆疊

透過 CloudFormation，您可以將相關資源視為單一單位進行管理，該單位稱為「堆疊」；意即您能夠建立、更新和刪除堆疊，藉此建立、更新並刪除資源集合。堆疊中所有的資源都是由堆疊的 CloudFormation 範本定義。假設您想建立包含 Auto Scaling 群組、Elastic Load Balancing 負載平衡器及 Amazon Relational Database Service (Amazon RDS) 資料庫執行個體的範本。則請提交您建立的範本以進行堆疊建立程序，即可建立這些資源；而 CloudFormation 會為您佈建所有這些資源。

變更集

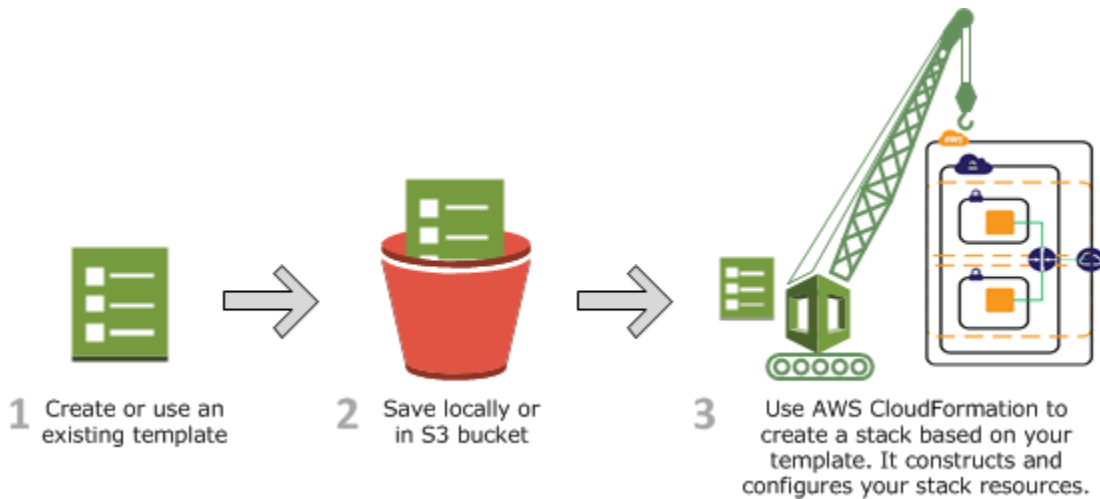
如果您需要變更堆疊內的執行中資源，請更新堆疊。在變更資源之前，您可以產生變更集以列出請求變更的摘要。藉由變更集，您可以先掌握該變更對執行中資源 (特別是重要資源) 造成的影響，再實行變更作業。

例如，若您變更 Amazon RDS 資料庫執行個體的名稱，CloudFormation 即會建立新的資料庫並刪除舊有資料庫。除非您已備份舊資料庫中的資料，否則這些資料均會遺失。如果您有產生變更集，即可發現該變更作業會導致資料庫遭取代，便能在更新堆疊前進行相應的考量。

CloudFormation 如何運作

當您使用 CloudFormation 來建立堆疊時，CloudFormation 會對 AWS 發出基礎服務呼叫，以佈建並設定範本中描述的資源。您需要許可才能建立這些資源。舉例來說，您必須擁有建立執行個體的許可，才能使用 CloudFormation 建立 EC2 執行個體。您可以使用 [AWS Identity and Access Management \(IAM\)](#) 來管理這些許可。

您的範本會宣告所有 CloudFormation 進行的呼叫。例如，假設範本可以說明執行個體類型為 t2.micro 的 EC2 執行個體；則使用該範本建立堆疊時，CloudFormation 即會呼叫 Amazon EC2 來建立執行個體 API，並將執行個體類型指定為 t2.micro。下圖概述了建立堆疊的 CloudFormation 工作流程。



建立 堆疊

1. 使用文字編輯器來建立 YAML 或 JSON 格式的 CloudFormation 範本。CloudFormation 範本會描述所需的資源及其設定。使用 [Infrastructure Composer](#) 來視覺化和驗證範本。這可協助確保範本結構正確且沒有語法錯誤。如需詳細資訊，請參閱[使用 CloudFormation 範本](#)。
2. 在本機或 Amazon S3 儲存貯體中儲存範本。
3. 透過指定範本檔案的位置 (如本機電腦上的路徑或 Amazon S3 URL) 建立 CloudFormation 堆疊。若該範本包含參數，您可以在建立堆疊期間指定輸入值。透過參數，您可以將值傳入範本，因此您每次建立堆疊時都可以自訂資源。

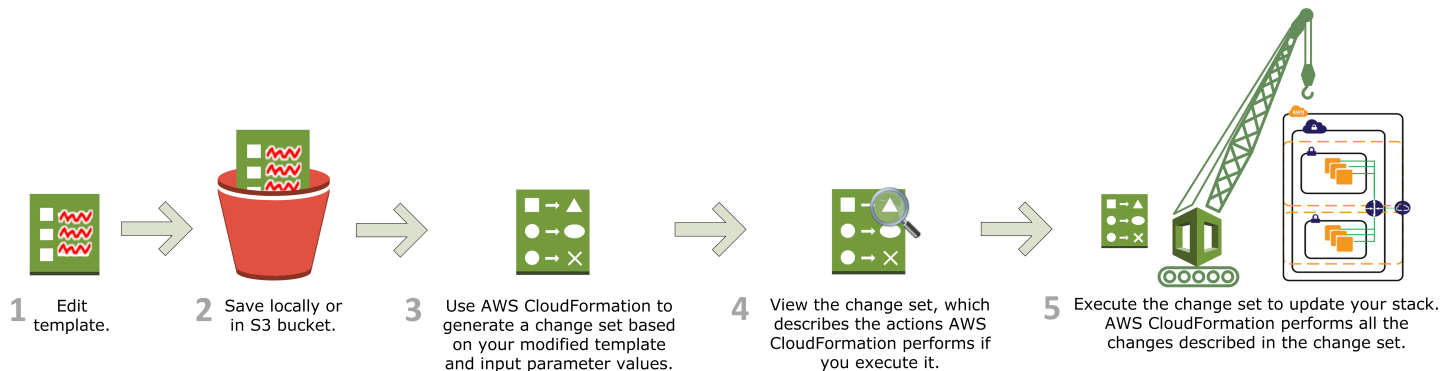
Note

如果您指定儲存在本機的範本檔案，CloudFormation 會將其上傳至您 AWS 帳戶中的 S3 儲存貯體。CloudFormation 會為您在其中上傳範本檔案的各個區域建立儲存貯體。AWS 您帳戶中具有 Amazon Simple Storage Service (Amazon S3) 許可的任何人都可以存取儲存貯體。如果 CloudFormation 已建立儲存貯體，則系統會將範本新增至該儲存貯體。您可以使用自己的儲存貯體，並手動上傳範本至 Amazon S3 以管理其許可。然後在您每次建立或更新堆疊時，指定範本檔案的 Amazon S3 URL。

建立完所有資源之後，CloudFormation 即會回報堆疊建立完成。接著，您便可以開始使用堆疊中的資源。如果堆疊建立失敗，則 CloudFormation 會刪除建立的資源來復原變更。

透過變更集更新堆疊

當您需要更新堆疊的資源時，即可修改堆疊的範本，並不需要建立新堆疊和刪除舊堆疊。若要更新堆疊，請透過提交原始堆疊範本的修改版本、不同的輸入參數值或同時提交兩者來建立變更集。CloudFormation 會比較經修改範本與原始範本，並產生變更集。而該變更集會列出建議的變更。檢閱變更之後，您可以啟動該變更集以更新堆疊，或建立新的變更集。下圖概述了更新堆疊的工作流程。



若要用變更集來更新堆疊

1. 可以使用 [Infrastructure Composer](#) 或文字編輯器來修改 CloudFormation 堆疊範本。如需詳細資訊，請參閱[更新您的堆疊範本](#)。

更新範本時，請謹記更新可能會導致中斷。根據更新中的資源與屬性，更新作業可能會中斷，甚至是取代現有資源。如需詳細資訊，請參閱[了解更新堆疊資源的行為](#)。

2. 在本機或 S3 儲存貯體中儲存 CloudFormation 範本。
3. 透過指定要更新的堆疊與修改後範本的位置 (如本機電腦上的路徑或 Amazon S3 URL)，即可建立變更集。如需建立變更集的詳細資訊，請參閱 [透過變更集更新 CloudFormation 堆疊](#)。

Note

如果您指定的範本存放於本機電腦上，CloudFormation 會自動將範本上傳至 AWS 帳戶中的 S3 儲存貯體。

4. 檢視變更集，藉此檢查 CloudFormation 是否會執行預期的變更。例如，檢查 CloudFormation 是否會取代任何重要的堆疊資源。您可以依需求建立無限個變更集，直到涵蓋所需的變更為止。

Important

變更集並不會指示系統是否能成功執行堆疊更新作業。例如，變更集並不會檢查是否超過帳戶**配額**、更新中的資源是否不支援更新，或者具備的**許可**是否不足以修改資源，上述所有情況都可能導致堆疊更新失敗。

5. 啟動要套用至堆疊的變更集。CloudFormation 透過僅更新經修改的資源來更新堆疊，並發出已成功更新堆疊的訊號。CloudFormation 在堆疊更新失敗時則會復原變更，使堆疊還原至上次已知的運作狀態。

CloudFormation 入門方法

若要使用主控台建立 hello world CloudFormation 堆疊，請參閱 [建立您的第一個堆疊](#)。

如需指導性學習，請嘗試 [CloudFormation入門](#) 工作坊，它會提供範本開發的實作體驗。

註冊 AWS 帳戶

當您註冊時 AWS，您的 AWS 帳戶會自動註冊中的所有服務 AWS，包括 CloudFormation。如果您已有 AWS 帳戶，請跳到下一個主題。如果您沒有 AWS 帳戶，請使用下列程序建立一個。

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

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或簡訊，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，AWS 帳戶根使用者會建立。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行[需要根使用者存取權的任務](#)。

AWS 會在註冊程序完成後傳送確認電子郵件給您。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。

建立具有管理存取權的使用者

註冊後 AWS 帳戶，請保護 AWS 帳戶根使用者、啟用 AWS IAM Identity Center 和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者 [AWS 管理主控台](#) 身分登入。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)。

如需說明，請參閱《IAM 使用者指南》中的 [為您的 AWS 帳戶 根使用者（主控台）啟用虛擬 MFA 裝置](#)。

建立具有管理存取權的使用者

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，將管理存取權授予使用者。

如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱 AWS IAM Identity Center 《使用者指南》中的 [使用預設值設定使用者存取 IAM Identity Center 目錄](#)。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 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](#)。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的 [建立許可集](#)。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的[新增群組](#)。

 Note

如需有關如何管理誰有權進行什麼操作的詳細資訊，請參閱 [使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

建立您的第一個堆疊

本主題會逐步引導您使用 建立您的第一個 CloudFormation 堆疊 AWS 管理主控台。遵循本教學課程，您將了解如何佈建基本 AWS 資源、監控堆疊事件和產生輸出。

在此範例中，以 YAML 格式撰寫 CloudFormation 範本。YAML 是一種人類可讀格式，廣泛用於定義基礎結構即程式碼。當您進一步了解 CloudFormation 時，可能也會遇到 JSON 格式的其他範本，但在本教學課程中，為了其可讀性選擇 YAML。

 Note

CloudFormation 是免費的，但您需要為建立的 Amazon EC2 和 Amazon S3 資源付費。不過，如果您是新手 AWS，則可以利用 [免費方案](#)，在此學習過程中將成本降至最低或消除。

主題

- [先決條件](#)
- [使用主控台建立 CloudFormation 堆疊](#)
- [監控堆疊建立](#)
- [測試 Web 伺服器](#)
- [疑難排解](#)
- [清除](#)
- [後續步驟](#)

先決條件

- 您必須具有 IAM 使用者或角色 AWS 帳戶 的存取權，該角色具有使用 Amazon EC2、Amazon S3 和 CloudFormation 的許可，或管理使用者存取權。
- 必須擁有可存取網際網路的虛擬私有雲端 (VPC)。此逐步解說範本需要預設的 VPC，較新的 AWS 帳戶會自動提供它。如果沒有預設的 VPC，或已刪除它，請參閱本主題中的疑難排解一節，了解替代解決方案。

使用主控台建立 CloudFormation 堆疊

使用主控台建立 Hello world CloudFormation 堆疊

1. 開啟 [CloudFormation 主控台](#)。
2. 選擇 Create Stack (建立堆疊)。
3. 在建立堆疊頁面中，選擇從 Infrastructure Composer 中建置，然後選擇在 Infrastructure Composer 中建立。您會進入 CloudFormation 主控台模式中的 Infrastructure Composer，可以在其中上傳和驗證範例範本。
4. 若要上傳和驗證範例範本，請執行下列動作：
 - a. 選擇範本。然後，將下列 CloudFormation 範本複製並貼到範本編輯器中：

```
AWSTemplateFormatVersion: 2010-09-09
Description: CloudFormation Template for WebServer with Security Group and EC2 Instance

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.
```

MyIP:

Description: Your IP address in CIDR format (e.g. 203.0.113.1/32).

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:**WebServerSecurityGroup:**

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP access via my IP address

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: !Ref MyIP

WebServer:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData: !Base64 |

#!/bin/bash

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/index.html

Outputs:**WebsiteURL:**

Value: !Join

- ''

- - http://

- !GetAtt WebServer.PublicDnsName

Description: Website URL

在進行下一步之前，讓我們花一點時間查看範本，並了解一些重要的 CloudFormation 概念。

- 建立堆疊時，**Parameters** 區段會宣告可傳遞至範本的值。在範本中稍後指定的資源會參考這些值並使用資料。參數是一種有效的方法，可指定您不想存放在範本本身中的資訊。也可使用其來指定可能對您要部署之特定應用程式或組態唯一的資訊。
- 範本可定義下列參數：
 - **LatestAmiId** – 從 AWS Systems Manager Parameter Store 中擷取最新的 Amazon Linux 2 AMI ID。
 - **InstanceType** – 允許選取 EC2 執行個體類型 (預設值：t2.micro，允許的值：t3.micro，t2.micro)。
 - **MyIP** – 指定 HTTP 存取的 IP 位址範圍 (預設值：0.0.0.0/0，允許從任何 IP 中存取)。
- **Resources** 區段包含您要使用範本建立 AWS 的資源定義。資源宣告是一次指定所有組態設定的有效方式。當您將資源宣告包含在範本中時，可以透過使用範本建立堆疊，來建立及設定所有宣告的資源。也可以從相同的範本中建立新堆疊，以啟動相同的資源組態。
- 此範本會建立以下資源：
 - **WebServerSecurityGroup** – EC2 安全群組，它允許來自指定 IP 範圍的連接埠 80 上的傳入 HTTP 流量。
 - **WebServer** – 具有下列組態的 EC2 執行個體：
 - 使用最新的 Amazon Linux 2 AMI
 - 套用選取的執行個體類型
 - 將 WebServerSecurityGroup 新增至 SecurityGroupIds 屬性
 - 包含使用者資料指令碼以安裝 Apache HTTP Server
 - 會在每個資源和參數宣告的開頭指定邏輯名稱。例如，WebServerSecurityGroup 是指派給 EC2 安全群組資源的邏輯名稱。然後，Ref 函數會透過其在範本其他部分中的邏輯名稱來參考資源和參數。當一個資源參考另一個資源時，會在兩者之間建立相依性。
 - **Outputs** 區段定義堆疊建立後傳回的自訂值。您可以使用輸出值，從堆疊中的資源傳回資訊，例如資源識別符或 URL。
 - 範本會定義一個輸出：
 - **WebsiteURL** – 已部署的 Web 伺服器的 URL，使用 EC2 執行個體的公有 DNS 名稱建構。Join 函數有助於將固定的 http:// 與可變的 PublicDnsName 合併為單一字串，從而輕鬆輸出 Web 伺服器的完整 URL。

- b. 選擇驗證，以確保 YAML 程式碼在上傳範本之前有效。
 - c. 接著，選擇建立範本以建立範本，並將其新增至 S3 儲存貯體。
 - d. 在開啟的對話方塊中，記下 S3 儲存貯體的名稱，以便稍後刪除它。然後，選擇確認並繼續前往 CloudFormation。這將帶您進入 CloudFormation 主控台，現在已在其中指定範本的 S3 路徑。
5. 在建立堆疊頁面中，選擇下一步。
 6. 在指定堆疊詳細資訊頁面，於堆疊名稱欄位輸入名稱。堆疊不可含有空格。在此範例中，使用 **MyTestStack**。
 7. 在參數下，指定參數值，如下所示：
 - LatestAmild：它預設為最新的 Amazon Linux 2 AMI。
 - InstanceType：為 EC2 執行個體類型選擇 t2.micro 或 t3.micro。

Note

如果您是新手 AWS，您可以使用免費方案免費啟動和使用 t2.micro 執行個體 12 個月（在 t2.micro 無法使用的區域中，您可以在免費方案下使用 t3.micro 執行個體）。

- MyIP：指定具有 /32 尾碼的實際公有 IP 位址。/32 尾碼用於 CIDR 表示法，以指定允許的單一 IP 位址。它本質上意味著允許流量進出此特定 IP 位址，而不允許其他位址。
8. 選取下一步兩次，以前往檢閱和建立頁面。在本教學課程中，可以在設定堆疊選項頁面上原樣保留預設值。
 9. 檢閱堆疊的資訊。當您滿意設定時，請選擇 Submit (提交)。

監控堆疊建立

選擇提交之後，CloudFormation 會開始建立範本中指定的資源。您的新堆疊 **MyTestStack** 會出現在 CloudFormation 主控台上方部分的清單中。其狀態應為 CREATE_IN_PROGRESS。您可以透過檢視其事件來查看堆疊的詳細狀態。

檢視堆疊的事件

1. 在 CloudFormation 主控台中，選擇清單中的堆疊 **MyTestStack**。
2. 在堆疊詳細資訊窗格中，選擇 Events (事件) 標籤。

主控台會每隔 60 秒自動重新整理事件清單，以包含最新的事件。

Events (事件) 標籤會顯示堆疊建立過程中的每個主要步驟，並依每個事件的時間排序，最新的事件放在最上方。

第一個事件 (位於事件清單的底部) 是開始堆疊建立程序：

```
2024-12-23 18:54 UTC-7 MyTestStack CREATE_IN_PROGRESS User initiated
```

接下來是標記每個資源建立開始和完成的事件。例如，建立 EC2 執行個體會產生下列項目：

```
2024-12-23 18:59 UTC-7 WebServer CREATE_COMPLETE
```

```
2024-12-23 18:54 UTC-7 WebServer CREATE_IN_PROGRESS Resource creation initiated
```

當 CloudFormation 報告已開始建立資源時，會記錄 CREATE_IN_PROGRESS 事件。當資源成功建立時，會記錄 CREATE_COMPLETE 事件。

當 CloudFormation 已成功建立堆疊時，您會在 Events (事件) 標籤頂端看到下列事件：

```
2024-12-23 19:17 UTC-7 MyTestStack CREATE_COMPLETE
```

如果 CloudFormation 無法建立資源，它會報告 CREATE_FAILED 事件，且預設會復原堆疊並刪除任何已建立的資源。Status Reason (狀態原因) 欄會顯示導致失敗的問題。

建立堆疊之後，可以前往資源索引標籤來檢視您建立的 EC2 執行個體和安全群組。

測試 Web 伺服器

成功建立堆疊後，導覽至 CloudFormation 主控台中的輸出索引標籤。尋找 WebsiteURL 欄位。這將包含 EC2 執行個體的公有 URL。

開啟瀏覽器並前往 WebsiteURL 下列出的 URL。應該會看到瀏覽器中顯示的簡單 "Hello World!" 訊息。

這確認您的 EC2 執行個體正在執行 Apache HTTP Server 並提供基本網頁。

疑難排解

如果在堆疊建立期間遇到復原，可能是因為缺少 VPC。以下是解決此問題的方法。

沒有可用的預設 VPC

此逐步解說中的範本需要預設 VPC。如果堆疊建立因 VPC 或子網路可用性錯誤而失敗，則帳戶中可能沒有預設 VPC。您有下列選項：

- 建立新的預設 VPC – 可以透過 Amazon VPC 主控台來建立新的預設 VPC。如需說明，請參閱《Amazon VPC 使用者指南》中的[建立預設 VPC](#)。
- 修改範本以指定子網路 – 如果您有非預設 VPC，可以修改範本以明確指定 VPC 和子網路 ID。將以下參數新增至範本：

```
SubnetId:
  Description: The subnet ID to launch the instance into
  Type: AWS::EC2::Subnet::Id
```

然後，更新 WebServer 資源以包含子網路 ID：

```
WebServer:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    SubnetId: !Ref SubnetId
    UserData: !Base64 |
      #!/bin/bash
      yum update -y
      yum install -y httpd
      systemctl start httpd
      systemctl enable httpd
      echo "<html><body><h1>Hello World!</h1></body></html>" > /var/www/html/
index.html
```

建立堆疊時，您需要指定具有網際網路存取權的子網路，才能連線 Web 伺服器。

清除

若要確定您不會針對任何不想要的服務支付費用，您可以透過刪除堆疊及其資源來清除這些服務。也可以刪除存放堆疊範本的 S3 儲存貯體。

刪除堆疊及其資源

1. 開啟 [CloudFormation 主控台](#)。
2. 在堆疊頁面中，選取您所建立堆疊名稱旁的選項 (**MyTestStack**)，然後選擇刪除。

3. 出現確認提示時，請選擇刪除。
4. 在事件索引標籤中監控堆疊刪除程序的進度。**MyTestStack** 的狀態會變更為 **DELETE_IN_PROGRESS**。當 CloudFormation 完成刪除堆疊時，它會從清單移除堆疊。

如果已完成使用範例範本，且不再需要 Amazon S3 儲存貯體，請將其刪除。您必須先清空儲存貯體，才能將其刪除。清空儲存貯體將會刪除其中的所有物件。

若要清空和刪除 Amazon S3 儲存貯體

1. 開啟 [Amazon S3 主控台](#)。
2. 在主控台左側的導覽窗格中，選擇 Buckets (儲存貯體)。
3. 在儲存貯體清單中，選取您為此教學課程建立的儲存貯體名稱旁的選項，然後選擇清空。
4. 在清空儲存貯體頁面上的文字欄位中輸入 **permanently delete**，以確認您要清空儲存貯體，然後選擇清空。
5. 在 Empty bucket: Status (清空儲存貯體：狀態) 頁面上監控儲存貯體清空的進度。
6. 若要返回儲存貯體清單，請選擇 Exit (結束)。
7. 選取儲存貯體名稱旁邊的選項，然後選擇刪除。
8. 出現確認提示時，請輸入儲存貯體名稱，然後選擇刪除儲存貯體。
9. 從儲存貯體清單中，監控儲存貯體刪除程序的進度。當 Amazon S3 完成刪除儲存貯體時，它會從清單移除儲存貯體。

後續步驟

恭喜您！您已成功建立堆疊、監控其建立並使用其輸出。

若要繼續學習：

- 進一步了解範本，以便您可以建立自己的範本。如需詳細資訊，請參閱[使用 CloudFormation 範本](#)。
- 請嘗試 [CloudFormation 入門](#) 工作坊，以取得有關範本建立的更多實作實務。
- 如需 [入門 CloudFormation](#) 的縮短版本，請參閱 [在 Amazon EC2 上部署應用程式](#)。本主題說明使用 CloudFormation 協助程式指令碼 `cfn-init` 來引導 Amazon EC2 執行個體的相同案例。

CloudFormation 最佳實務

最佳實務是一些建議，可協助您在整個工作流程中更有效地使用 CloudFormation 並採用安全的實務。了解如何規劃和組織您的堆疊、建立範本來描述您的資源和對其執行的軟體應用程式，以及管理您的堆疊及其資源。以下最佳實務是根據目前 CloudFormation 客戶的實際體驗。

規劃與組織

- [縮短回饋迴圈以提高開發速度](#)
- [依生命週期和所有權組織您的堆疊](#)
- [使用跨堆疊參考來傳回另一個堆疊匯出的輸出值](#)
- [針對多帳戶和多區域部署使用 CloudFormation StackSets](#)
- [重複使用範本在多個環境中複製堆疊](#)
- [驗證所有資源類型的配額](#)
- [使用模組以重複使用資源組態](#)
- [採用基礎設施作為程式碼實務](#)

建立範本

- [請勿在您的範本中內嵌憑證](#)
- [使用 AWS 特定的參數類型](#)
- [使用參數限制條件](#)
- [使用虛擬參數提升可攜性](#)
- [使用 `AWS::CloudFormation::Init` 在 Amazon EC2 執行個體上部署軟體應用程式](#)
- [使用最新的協助程式指令碼](#)
- [驗證後再使用範本](#)
- [使用 YAML 或 JSON 來撰寫範本](#)
- [實作全面的標記策略](#)
- [利用範本巨集進行進階轉換](#)

管理堆疊

- [透過 CloudFormation 管理所有堆疊資源](#)
- [更新您的堆疊之前建立變更集](#)

- [使用堆疊政策來保護資源](#)
- [使用 AWS CloudTrail 記錄 CloudFormation 呼叫](#)
- [使用程式碼檢閱和修訂控制管理您的範本](#)
- [定期更新 Amazon EC2 執行個體](#)
- [定期使用偏離偵測](#)
- [設定自動復原的復原觸發條件](#)
- [實作有效的堆疊重構策略](#)
- [使用 CloudFormation Hooks 進行生命週期管理](#)

編寫工具

- [使用 IaC 產生器從現有資源中建立範本](#)
- [AWS Infrastructure Composer 用於視覺化範本設計](#)
- [考慮將 AWS Cloud Development Kit \(AWS CDK\) 用於複雜的基礎設施](#)

安全和合規

- [使用 IAM 控制存取](#)
- [套用最低權限準則](#)
- [保護敏感參數](#)
- [使用 將政策實作為程式碼 AWS CloudFormation Guard](#)

縮短回饋迴圈以提高開發速度

採用適當的實務和工具，協祝您針對使用 CloudFormation 範本描述的基礎設施縮短回饋迴圈。這包括在工作站中對範本執行早期檢查和測試；這樣一來，在將成品提交至原始碼儲存庫之前，您還有機會發現潛在的語法和組態問題。早期發現這類問題有助於防止它們進入正式的生命週期環境，例如開發、品質保證和生產。這種早期測試、快速檢錯方法可讓您減少重工等待時間、減少潛在的影響區域，以及增加成功佈建作業的信心程度。

有助於實現快速檢錯實務的工具選擇包括 [CloudFormation Linter \(cfn-lint\)](#) 和 [TaskCat](#) 命令列工具。cfn-lint 工具可讓您根據 [CloudFormation 資源規格](#) 來驗證您的 CloudFormation 範本。這包括檢查資源屬性的有效值，以及最佳實務。cfn-lint 的外掛程式[適用於許多程式碼編輯器](#)；這可讓問題在編輯器中一目了然，為您提供直接的 linter 回饋。您也可以選擇將 cfn-lint 整合到原始碼儲存庫的組態中，以便在提交成品時執行範本驗證。如需詳細資訊，請參閱[使用 對 CloudFormation 範本進行 Git 預先遞交驗證 cfn-lint](#)。執行初始檢查並修正了 cfn-lint 可能找出的任何問題之後，您可以

使用 TaskCat 在您選擇的 AWS 區域中以程式設計方式建立堆疊來測試範本。TaskCat 也會針對您選擇的每個區域產生包含合格/失敗等級的報告。

如需有關如何使用這兩種工具縮短回饋迴圈的逐步實作演練，請按照 [CloudFormation Workshop](#) 的 [Linting and Testing 實驗室](#) 進行操作。

依生命週期和所有權組織您的堆疊

使用 AWS 資源的生命週期和擁有權，協助您決定每個堆疊中應包含的資源。一開始，您可能會將所有資源放在一個堆疊中，但隨著您的堆疊逐漸成比例成長且範圍越來越廣泛，管理單一堆疊可能既繁瑣又耗時。透過根據通用生命週期和所有權來分組資源，擁有者就可以使用自己的程序和排程來變更一組資源，無需影響其他資源。

例如，假設開發人員和工程師團隊擁有一個網站，它託管在負載平衡器後方的 Amazon EC2 Auto Scaling 執行個體上。由於網站有自己的生命週期並由網站團隊維護，因此您可以建立網站及其資源的堆疊。現在，假設網站也會使用後端資料庫，這些資料庫位於資料庫管理員擁有及維護的個別堆疊中。每當網站團隊或資料庫團隊需要更新其資源時，都可以在不影響彼此的堆疊下執行這項操作。如果所有資源都在同一個堆疊中，協調及傳達更新可能很困難。

如需組織您堆疊的其他指引，您可以使用兩個常見架構：多層架構和服務導向架構 (SOA)。

分層架構可將堆疊組織成彼此重疊的多個水平層，每個 layer 會相依於其正下方的 layer。您可以在每個 layer 中有一或多個堆疊，但在每個 layer 內，您的堆疊應該有生命週期和所有權類似的 AWS 資源。

使用服務導向架構，您可以將龐大的商務問題組織成多個易於管理的部分。每個部分都是一項服務，其具有明確定義的用途，並代表獨立的功能單位。您可以將這些服務映射至堆疊，每個堆疊都有自己的生命週期和擁有者。所有服務 (堆疊) 都可連接在一起，以便彼此互動。

使用跨堆疊參考來傳回另一個堆疊匯出的輸出值

當您根據生命週期和擁有權組織 AWS 資源時，建議您建置使用另一個堆疊中資源的堆疊。您可以硬式編碼值，或使用輸入參數來傳遞資源名稱和 ID。不過，這些方法可能會讓範本難以重複使用，或可能增加讓堆疊執行的成本。反之，請使用跨堆疊參考來傳回由另一個堆疊匯出的輸出值，以便其他堆疊可以對其進行使用。堆疊可以透過使用 `Fn::ImportValue` 函數呼叫匯出的資源，來使用這些資源。

例如，您可能有一個網路堆疊，其中包含 VPC、安全群組和子網路。您想要讓所有公有 Web 應用程式使用這些資源。透過匯出資源，即允許具有公有 Web 應用程式的所有堆疊使用這些資源。如需詳細資訊，請參閱 [從部署的 CloudFormation 堆疊取得匯出的輸出](#)。

針對多帳戶和多區域部署使用 CloudFormation StackSets

CloudFormation StackSets 可讓您透過單一操作跨多個帳戶和區域建立、更新或刪除堆疊，藉此擴充堆疊的功能。使用 StackSets 在整個組織中部署常見的基礎設施元件、合規控制或共用服務。

使用 StackSets 時，使用 實作服務受管許可 AWS Organizations，以簡化許可管理。此方法可讓您將 StackSets 部署到組織內的帳戶，而不需要在每個帳戶中手動設定 IAM 角色。

如需有關 StackSets 的詳細資訊，請參閱 [StackSets 概念](#)。

驗證所有資源類型的配額

啟動堆疊之前，請確定您可以建立所需的所有資源，而不會達到 AWS 您的帳戶限制。如果您達到限制，在您提高配額或刪除額外資源之前，CloudFormation 都無法成功建立您的堆疊。每項服務可能有各種限制，需要您在啟動堆疊前注意。例如，根據預設，您只能在 AWS 帳戶的每個區域啟動 2000 個 CloudFormation 堆疊。如需有關限制及如何提高預設限制的詳細資訊，請參閱 AWS 一般參考 中的 [AWS 服務配額](#)。

重複使用範本在多個環境中複製堆疊

設定您的堆疊和資源之後，您可以重複使用範本在多個環境中複製您的基礎設施。例如，您可以建立適用於開發、測試和生產的環境，讓您可以測試變更，再將其實作到生產環境中。為了讓範本可重複使用，請使用參數、映射和條件區段，以便在建立堆疊時能夠自訂這些堆疊。例如，對於您的開發環境，您可以指定相較於生產環境成本較低的執行個體類型，但所有其他組態和設定則保持不變。如需參數、映射和條件的詳細資訊，請參閱 [CloudFormation 範本區段](#)。

使用模組以重複使用資源組態

隨著您的基礎設施逐漸成長，可能出現的常見模式即是您會在每個範本中宣告相同的元件。模組可讓您以透明、易於管理且可重複的方式，來封裝可包含在堆疊範本之間的資源組態。模組可透過可自訂的模組化建置區塊形式，來封裝通用服務組態和最佳實務，以便您將其納入堆疊範本中。

這些建置區塊可用於單一資源，例如定義 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的最佳實務；也可以用於多個資源，以定義應用程式架構的常見模式。這些建置區塊可以嵌套到其他模組中，因此您可以將最佳實務堆疊到較高層級的建置區塊中。CloudFormation 模組在 [CloudFormation 登錄檔](#) 中可用，因此您可以像使用原生資源一樣使用這些模組。當您使用 CloudFormation 模組時，會

將模組範本被擴展到消費範本，這樣一來，您可以存取使用 [Ref](#) 或 [Fn::GetAtt](#) 模組內的資源。如需詳細資訊，請參閱[透過 CloudFormation 模組建立可跨範本納入的可重複使用資源組態](#)。

採用基礎設施作為程式碼實務

透過實作基礎結構即程式碼 (IaC) 實務，將 CloudFormation 範本視為程式碼。將您的範本存放在版本控制系統中，實作程式碼檢閱，並使用自動化測試來驗證變更。此方法可確保一致性，改善協同合作，並提供基礎設施變更的稽核軌跡。

請考慮為您的基礎設施程式碼實作 CI/CD 管道，以自動化 CloudFormation 範本的測試和部署。AWS CodePipeline AWS CodeBuild和 等工具 AWS CodeDeploy 可用於為您的基礎設施部署建立自動化工作流程。

如需實作 IaC 最佳實務的詳細資訊，請參閱[使用 AWS CloudFormation 做為 IaC 工具](#)。

如需有關搭配使用持續交付與 CloudFormation 的詳細資訊，請參閱[使用 CodePipeline 進行持續交付](#)。

請勿在您的範本中內嵌憑證

建議您在堆疊範本中使用動態參考，而不要在 CloudFormation 範本中嵌入敏感資訊。

動態參考提供精簡且強大的方式，讓您參考儲存在其他服務中的外部值，例如 AWS Systems Manager 參數存放區或 AWS Secrets Manager。當您使用動態參考時，在堆疊和變更集操作期間，CloudFormation 會在必要時擷取指定參考的值，並將值傳遞至適當的資源。不過，CloudFormation 絕不會存放實際參考值。如需詳細資訊，請參閱[使用動態參考來指定範本值](#)。

[AWS Secrets Manager](#) 可協助您安全地加密、存放和擷取資料庫和其他服務的登入資料。[AWS Systems Manager 參數存放區](#)提供安全的階層式儲存，以管理組態資料。

如需定義範本參數的詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

使用 AWS 特定的參數類型

如果您的範本需要輸入現有 AWS 特定值，例如現有 Amazon Virtual Private Cloud IDs 或 Amazon EC2 AWS 金鑰對名稱，請使用特定參數類型。例如，您可以將參數指定為類型 `AWS::EC2::KeyPair::KeyName`，這會使用您和您的堆疊 AWS 帳戶之區域中的現有金鑰對

名稱。CloudFormation 可以在建立堆疊之前快速驗證 AWS 特定參數類型的值。此外，如果您使用 CloudFormation 主控台，CloudFormation 會顯示有效值的下拉式清單，因此您不需要查詢或記住正確的 VPC ID 或金鑰對名稱。如需詳細資訊，請參閱[使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)。

使用參數限制條件

使用限制條件，您可以描述允許的輸入值，讓 CloudFormation 在建立堆疊之前攔截任何無效的值。您可以設定限制條件，例如長度下限、長度上限和允許的模式。例如，您可以對資料庫使用者名稱值設定限制條件，讓其長度至少為八個字元並只包含英數字元。如需詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

使用虛擬參數提升可攜性

您可以在範本中使用[虛擬參數](#)作為[內部函數](#)的引數，例如 Ref 和 Fn::Sub。虛擬參數為 CloudFormation 預先定義的參數。請勿在範本中宣告這些參數。在內建函數中使用虛擬參數可增加堆疊範本跨區域和帳戶的可攜性。

例如，假設您想要建立範本，其中需要針對特定資源屬性指定另一個現有資源的 [Amazon Resource Name](#) (ARN)。在此情況下，現有資源是具有以下 ARN 的 [AWS Systems Manager 參數存放區](#) 資源：arn:aws:ssm:us-east-1:123456789012:parameter/MySampleParameter。您需要針對目標 AWS 分割區、區域和帳戶 ID 調整 [ARN 格式](#)。您可以使用 AWS::Partition、AWS::Region 和 AWS::AccountId 虛擬參數提升範本可攜性，而非對此類值進行硬式編碼。在此情況下，下列範例會示範如何將 ARN 中的元素與 CloudFormation 串連：！Sub 'arn:\${AWS::Partition}:ssm:\${AWS::Region}:\${AWS::AccountId}:parameter/MySampleParameter。

針對另一個範例，假設您想要跨多個堆疊共用資源或組態。在此範例中，假設您已建立 VPC 的[子網路](#)，然後匯出其 ID 以用於相同 AWS 帳戶和區域中的其他堆疊。在另一個堆疊中，您可以在描述 Amazon EC2 執行個體時，參考子網路 ID 的匯出值。如需使用 Export 輸出欄位和 Fn::ImportValue 內部函數的詳細範例，請參閱[參考在另一個 CloudFormation 堆疊中的資源輸出](#)。

堆疊匯出在每個帳戶和區域中必須是唯一。因此，在此情況下，您可以使用 AWS::StackName 虛擬參數為匯出建立字首。由於每個帳戶和區域的堆疊名稱也必須是唯一，因此使用此虛擬參數作為字首會增加具有唯一匯出名稱的可能性，同時也促使在匯出值的堆疊間採行可重複使用的方法。或者，您可以自行選擇使用的字首。

使用 `AWS::CloudFormation::Init` 在 Amazon EC2 執行個體上部署軟體應用程式

當您啟動堆疊時，您可以使用 `cfn-init` 協助程式指令碼和 `AWS::CloudFormation::Init` 資源，在 Amazon EC2 執行個體上安裝及設定軟體應用程式。透過使用 `AWS::CloudFormation::Init`，您可以描述想要的組態，而不用編寫程序性步驟的指令碼。您也可以更新組態，無需重新建立執行個體。此外，如果您的組態發生任何錯誤，CloudFormation 會產生日誌，以供您用來調查問題。

在您的範本中，透過 `AWS::CloudFormation::Init` 資源指定安裝和組態狀態。如需有關說明如何使用 `cfn-init` 和 `AWS::CloudFormation::Init` 的演練，請參閱 [在 Amazon EC2 上部署應用程式](#)。

使用最新的協助程式指令碼

CloudFormation 協助程式指令碼會定期更新。請務必在您範本的 `UserData` 屬性中包含下列命令，再呼叫協助程式指令碼，以確保您已啟動的執行個體取得最新的協助程式指令碼：

```
yum install -y aws-cfn-bootstrap
```

如需有關取得最新協助程式指令碼的詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [CloudFormation 協助程式指令碼參考](#)。

驗證後再使用範本

您可以使用 CloudFormation 驗證範本，再使用範本建立或更新堆疊。驗證範本可在 CloudFormation 建立任何資源之前，協助您攔截語法和一些語意錯誤，例如循環相依性。如果您使用 CloudFormation 主控台，則主控台會在您指定輸入參數之後自動驗證範本。針對 AWS CLI 或 CloudFormation API，使用 [validate-template](#) 命令或 [ValidateTemplate](#) 操作。

在驗證期間，CloudFormation 會先檢查範本是否為有效的 JSON。如果不是，CloudFormation 會檢查範本是否為有效的 YAML。如果兩種檢查都失敗，則 CloudFormation 會傳回範本驗證錯誤。

驗證範本是否符合組織政策規範

您還可驗證範本與組織政策指導方針的符合性。AWS CloudFormation Guard (`cfn-guard`) 是一種開放原始碼命令行介面 (CLI) 工具，提供政策即程式碼語言，以定義可檢查必要和禁止之資源組態的規

則。然後，它可讓您根據這些規則驗證您的範本。例如，管理員可建立規則以確保使用者一律建立加密的 Amazon S3 儲存貯體。

您可在本機使用 `cfn-guard`，或自動作為 CI/CD 管道的一部分，以停止不合規資源的部署。

此外，`cfn-guard` 也包含功能 `rulegen`，該功能可讓您從現有的合規 CloudFormation 範本擷取規則。

如需詳細資訊，請參閱 GitHub 上的 [cfn-guard](#) 儲存庫。

使用 YAML 或 JSON 來撰寫範本

CloudFormation 支援 YAML 和 JSON 格式的範本。每個都有其優點，選擇取決於您的特定需求：

在以下情況使用 YAML

- 優先考慮人類可讀性和可維護性
- 您想要包含註解以記錄範本
- 您正在使用具有巢狀結構的複雜範本
- 您想要使用 YAML 特定功能，例如錨點和別名，以減少重複

在下列情況下使用 JSON：

- 您需要與偏好 JSON 的工具或系統整合
- 您正在使用程式設計式範本產生或操作
- 您需要嚴格的資料驗證

由於 YAML 的可讀性並支援註解，因此通常建議用於手動範本撰寫。對於複雜範本，它特別有用，因為其行首縮進式結構有助於視覺化資源階層。在自動化工作流程中或使用預期 JSON 輸入的 API 時，JSON 可能很有優勢。當您需要確保嚴格遵守特定結構時，它也很有幫助。無論選擇哪種格式，都專注於建立結構良好、有文件記錄且可維護的範本。如果使用 YAML，請利用其錨點和別名等功能來減少重複並改善可維護性。

實作全面的標記策略

為 CloudFormation 範本建立的所有資源實作一致的標記策略。標籤有助於資源組織、成本分配、存取控制和自動化。考慮包含環境、擁有者、成本中心、應用程式和用途的標籤。

使用 `AWS::CloudFormation::Stack` 資源的 `Tags` 屬性，將標籤套用至堆疊中所有支援的資源。也可以使用許多資源類型上可用的 `TagSpecifications` 屬性，在資源建立期間套用標籤。

如需詳細資訊，請參閱[資源標籤](#)。

利用範本巨集進行進階轉換

CloudFormation 巨集可讓您在範本上執行自訂處理，從查找並取代操作等簡單動作，到產生其他資源的複雜轉換。使用巨集來擴展 CloudFormation 範本的功能，並在整個組織中實作可重複使用的模式。

AWS Serverless Application Model 是巨集的範例，可簡化無伺服器應用程式的開發。請考慮為組織特定模式和需求建立自訂巨集。

如需有關在範本中使用巨集的詳細資訊，請參閱 [CloudFormation 巨集概觀](#)。

透過 CloudFormation 管理所有堆疊資源

啟動堆疊之後，使用 CloudFormation [主控台](#)、[API](#) 或 [AWS CLI](#) 來更新堆疊中的資源。請勿在 CloudFormation 外變更堆疊資源。這樣做可能會在您堆疊的範本與您堆疊資源的目前狀態之間產生不相符的情況，如果您更新或刪除堆疊，這可能會造成錯誤。這稱為偏離。如果對 CloudFormation 範本外部的資源進行變更，並且更新堆疊，則將捨棄直接對資源所做的變更，且資源組態將還原為範本中的組態。

如需有關偏離的詳細資訊，請參閱[什麼是偏離？](#)。

如需有關更新堆疊的詳細資訊，請參閱[更新 CloudFormation 堆疊](#)。

更新您的堆疊之前建立變更集

變更集可讓您先查看提議的堆疊變更對執行中資源可能造成的影響，再實作這些變更。在您執行變更集之前，CloudFormation 不會對您的堆疊進行任何變更，這讓您可以決定是繼續進行提議的變更，還是建立其他變更集。

使用變更集可查看您的變更對執行中資源 (特別是重要資源) 造成的影響。例如，如果您變更 Amazon RDS 資料庫執行個體的名稱，CloudFormation 會建立新的資料庫並刪除舊的資料庫；除非您已進行備份，否則您將會遺失舊資料庫中的資料。如果您產生變更集，您會看到變更取代您的資料庫。這可協助您先規劃，再更新您的堆疊。如需詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

使用堆疊政策來保護資源

堆疊政策可協助防止意外更新重要的堆疊資源，這可能會導致資源被中斷或甚至被取代。堆疊政策是一種 JSON 文件，描述可在指定資源上執行的更新動作。每當您建立具有重要資源的堆疊時，請指定堆疊政策。

在堆疊更新期間，您必須明確地指定要更新的受保護資源，否則不會對受保護的資源進行任何變更。如需詳細資訊，請參閱[避免更新堆疊資源](#)。

使用 AWS CloudTrail 記錄 CloudFormation 呼叫

AWS CloudTrail 會追蹤在您的 中進行 CloudFormation API 呼叫的任何人 AWS 帳戶。每當有人使用 CloudFormation API、CloudFormation 主控台、後端主控台或 CloudFormation AWS CLI 命令時，都會記錄 API 呼叫。啟用記錄，然後指定 Amazon S3 儲存貯體來存放日誌。如此一來，如果您需要，即可稽核誰在您的帳戶中發出哪些 CloudFormation 呼叫。

如需詳細資訊，請參閱[使用 記錄 CloudFormation API 呼叫 AWS CloudTrail](#)。

使用程式碼檢閱和修訂控制管理您的範本

您的堆疊範本描述 AWS 資源的組態，例如其屬性值。若要檢閱變更並保留確切的資源歷史記錄，請使用程式碼檢閱和修訂控制。這些方法可協助您追蹤不同版本的範本之間的變更，如此有助於追蹤您堆疊資源的變更。此外，透過維護歷史記錄，您就可以隨時將堆疊還原至範本的特定版本。

定期更新 Amazon EC2 執行個體

在所有 Amazon EC2 Windows 執行個體及使用 CloudFormation 建立的 Amazon EC2 Linux 執行個體上，定期執行 `yum update` 命令以更新 RPM 套件。這可確保您取得最新的修正程式和安全性更新。

定期使用偏離偵測

定期使用 CloudFormation 偏離偵測功能來識別已在 CloudFormation 管理之外修改的資源。偵測和解決偏離有助於維持基礎設施即程式碼方法的完整性，並確保您的範本可準確反映已部署資源的狀態。

請考慮將自動偏離偵測實作作為操作程序的一部分。您可以使用 Amazon EventBridge 規則觸發的 AWS Lambda 函數定期檢查偏離，並在偵測到差異時通知您的團隊。

如需有關偏離的詳細資訊，請參閱[使用偏離偵測來偵測堆疊和資源的未受管組態變更](#)。

設定自動復原的復原觸發條件

使用復原觸發來指定 CloudFormation 在堆疊建立和更新操作期間應監控的 Amazon CloudWatch 警示。如果任何指定的警示進入 ALARM 狀態，CloudFormation 會自動復原整個堆疊操作，協助確保基礎設施保持穩定狀態。

設定關鍵指標的復原觸發條件，例如應用程式錯誤率、系統資源使用率或指出應用程式和基礎設施運作狀態的自訂業務指標。

如需有關復原觸發條件的詳細資訊，請參閱[在警示違規時復原堆疊](#)。

實作有效的堆疊重構策略

隨著基礎設施的發展，您可能需要重構 CloudFormation 堆疊，以提高可維護性、降低複雜性或適應不斷變化的需求。堆疊重構涉及重組您的範本和資源，同時保留其外部行為和功能。堆疊重構有助於透過下列方式與 CloudFormation 搭配使用：

- 分割單體堆疊：將大型、複雜的堆疊分解為更小、更易於管理的堆疊，依生命週期或擁有權組織
- 合併相關資源：將來自多個堆疊的相關資源合併為單一、有凝聚力的堆疊以簡化管理
- 擷取可重複使用的元件：將常見模式移入模組或巢狀堆疊，以提升重複使用和一致性
- 改善資源組織：重組堆疊中的資源，以更好地反映其關係和相依性

如需有關重構 CloudFormation 堆疊的詳細資訊，請參閱[堆疊重構](#)。

使用 CloudFormation Hooks 進行生命週期管理

CloudFormation Hooks 提供的程式碼，可在佈建之前主動檢查 AWS 資源的組態，並執行複雜的驗證檢查。Hooks 可檢查您的資源、堆疊和變更集是否符合組織的安全性、營運和成本最佳化需求。其會在資源佈建之前提供警告，或使操作失敗並完全停止，這視其設定方式而定。違規和警告會記錄在 Amazon CloudWatch 中，以提供不合規部署的可見性。

如需這些 Hooks 最佳實務的詳細資訊，請參閱 [AWS CloudFormation Hooks 概念](#)。

如需 Hooks 可以為您的 CloudFormation 資源做什麼的詳細資訊，請參閱[什麼是 AWS CloudFormation Hooks ?](#)

使用 IaC 產生器從現有資源中建立範本

CloudFormation IaC（基礎設施即程式碼）產生器可協助您從現有 AWS 資源建立 CloudFormation 範本。當您需要複製現有基礎設施、記錄手動建立的資源或在 CloudFormation 管理下使用先前未受管的資源時，此功能特別有用。IaC 產生器適用於以下列方式建立 CloudFormation 範本：

- 加速的範本建立：從現有資源中產生範本，而不是從頭開始寫入範本
- 一致的基礎設施：使用產生的範本作為起點，確保新環境符合現有環境
- 移轉至基礎設施即程式碼：在 CloudFormation 管理下逐漸引入手動建立的資源
- 文件：以範本形式建立現有基礎設施的記錄

如需 IaC 產生器的詳細資訊，請參閱[使用 IaC 產生器從現有資源中產生範本](#)。

AWS Infrastructure Composer 用於視覺化範本設計

AWS Infrastructure Composer 是一種視覺化設計工具，可協助您使用 drag-and-drop 界面建立、視覺化和修改 CloudFormation 範本。以下列方式使用 CloudFormation 時，它特別有用：

- 架構規劃：在實作之前設計和驗證基礎設施架構
- 範本現代化：視覺化現有範本以了解其結構，並發現改善機會
- 訓練和加入：協助新團隊成員透過視覺化學習了解 CloudFormation 概念 AWS 和服務關係
- 利益相關者溝通：使用清晰的視覺化呈現方式，向非技術利益相關者展示基礎設施設計
- 合規審查：使用視覺化圖表來提升基礎設施設計的安全性和合規審查
- 合規審查：使用視覺化圖表來提升基礎設施設計的安全性和合規審查

如需 Infrastructure Composer 的詳細資訊，請參閱[什麼是 AWS Infrastructure Composer？](#)。

考慮將 AWS Cloud Development Kit (AWS CDK) 用於複雜的基礎設施

如需複雜的基礎設施需求，請考慮使用 CDK，使用 TypeScript、Python、Java 和 .NET。AWS CDK generates CloudFormation 範本等熟悉的程式設計語言來定義您的雲端資源，讓您利用 CloudFormation 的完整功能，同時使用您慣用語言的抽象和程式設計建構。

AWS CDK 提供高階建構，可封裝最佳實務並簡化常見基礎設施模式的定義。這可以大幅減少定義基礎設施所需的程式碼數量，同時確保遵守最佳實務。

如需 CDK 的詳細資訊，請參閱 [AWS Cloud Development Kit \(AWS CDK\)](#)。

使用 IAM 控制存取

IAM 是一項 AWS 服務，可用來管理 中的使用者及其許可 AWS。您可以使用 IAM 搭配 CloudFormation 來指定使用者可執行的 CloudFormation 動作，例如檢視堆疊範本、建立堆疊或刪除堆疊。此外，管理 CloudFormation 堆疊的任何人都需要這些堆疊中的資源許可。例如，如果使用者想要使用 CloudFormation 來啟動、更新或終止 Amazon EC2 執行個體，則必須具備呼叫相關 Amazon EC2 動作的許可。

在大多數情況下，使用者需要完整存取權，才能管理範本中的所有資源。CloudFormation 會呼叫來代替使用者建立、修改及刪除這些資源。若要區隔使用者與 CloudFormation 服務的許可，請使用服務角色。CloudFormation 使用服務角色的政策來呼叫，而不是使用者的政策。如需詳細資訊，請參閱 [CloudFormation 服務角色](#)。

套用最低權限準則

為 CloudFormation 服務角色或範本建立的資源設定 IAM 角色時，請務必套用最低權限原則。僅授予預期功能所需的許可，並盡可能避免使用萬用字元許可。

使用 IAM Access Analyzer 來檢閱授予給 CloudFormation 服務角色的許可，並識別可以移除的未使用許可。定期檢閱和更新 IAM 政策，以確保其符合安全需求。

保護敏感參數

如需密碼、API 金鑰和其他秘密等敏感資訊，請使用 AWS Systems Manager 參數存放區或 AWS Secrets Manager，而不是直接內嵌在您的範本中。在範本中使用動態參考，以便在堆疊操作期間安全地擷取這些值。

在範本中使用參數時，請將敏感參數的 NoEcho 屬性設為 true，以防止其值顯示在主控制台、API 回應或 CLI 輸出中。請注意，如果傳遞給可能記錄該值的其他服務或資源，則 NoEcho 不會阻止記錄該值。

如需搭配 CloudFormation 使用 AWS Systems Manager 參數存放區的詳細資訊，[請參閱從 AWS Systems Manager 參數存放區取得純文字值](#)。

如需有關使用 NoEcho 屬性的詳細資訊，請參閱 [CloudFormation 範本參數語法](#)。

如需 AWS Secrets Manager 搭配 CloudFormation 使用的詳細資訊，請參閱在 [中建立 AWS Secrets Manager 秘密 AWS CloudFormation](#)。

使用 將政策實作為程式碼 AWS CloudFormation Guard

AWS CloudFormation Guard (cfn-guard) 是一種開放原始碼 policy-as-code 工具，可讓您定義和強制執行 CloudFormation 範本的規則。使用 cfn-guard 來確保範本符合組織政策、安全最佳實務和控管要求。

將 cfn-guard 整合到您的 CI/CD 管道，以在部署之前根據您的政策規則來自動驗證範本。這有助於防止不合規資源部署到您的環境，並向開發人員提供有關政策違規的早期意見回饋。

如需 Guard 的詳細資訊，請參閱 [什麼是 AWS CloudFormation Guard ?](#)

使用 CloudFormation 範本

AWS CloudFormation 範本會定義您要在堆疊中建立、更新或刪除 AWS 的資源。它包含多個區段，但唯一必要的區段是 [Resources](#) 區段，且該區段必須宣告至少一項資源。

您可以使用下列方法建立範本：

- AWS Infrastructure Composer – 用於設計範本的視覺化介面。
- 文字編輯器 – 直接以 JSON 或 YAML 語法撰寫範本。
- IaC 產生器 – 從您帳戶中已佈建但目前未由 CloudFormation 管理的資源產生範本。IaC 產生器可搭配您所在區域中 Cloud Control API 支援的多種資源類型使用。

本節提供完整指南，說明如何使用 CloudFormation 範本的各個區段，以及如何開始建立堆疊範本。它涵蓋下列主題：

主題

- [範本的儲存位置](#)
- [驗證範本](#)
- [開始使用範本](#)
- [範例範本](#)
- [CloudFormation 範本格式](#)
- [CloudFormation 範本區段](#)
- [透過 Infrastructure Composer 視覺化建立範本](#)
- [AWS CloudFormation 語言伺服器](#)
- [使用 IaC 產生器從現有資源中產生範本](#)
- [使用動態參考取得存放在其他服務中的值](#)
- [使用虛擬參數取得 AWS 值](#)
- [從部署的 CloudFormation 堆疊取得匯出的輸出](#)
- [使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)
- [CloudFormation 逐步解說](#)
- [CloudFormation 範本程式碼片段](#)

- [使用 CloudFormation 部署 Windows 型堆疊](#)
- [使用 CloudFormation 提供的資源類型來擴展範本的功能](#)
- [透過 CloudFormation 模組建立可跨範本納入的可重複使用資源組態](#)

範本的儲存位置

Amazon S3 儲存貯體

您可以將 CloudFormation 範本儲存在 Amazon S3 儲存貯體中。建立或更新堆疊時，您可以指定範本的 S3 URL，而非直接上傳。

如果您直接透過 AWS 管理主控台 或 上傳範本 AWS CLI，系統會自動為您建立 S3 儲存貯體。如需詳細資訊，請參閱[從 CloudFormation 主控台建立堆疊](#)。

Git 儲存庫

使用 [Git 同步](#)，您可以在 Git 儲存庫中儲存範本。建立或更新堆疊時，您可以指定包含範本的 Git 儲存庫位置和分支，而非直接上傳或參考 S3 URL。CloudFormation 會自動監控指定儲存庫和分支的範本變更。如需詳細資訊，請參閱[使用 Git 同步從儲存庫原始程式碼建立堆疊](#)。

驗證範本

語法驗證

您可以使用 [validate-template](#) CLI 命令，或在主控台上指定範本，驗證範本的 JSON 或 YAML 語法。主控台會自動執行驗證。如需詳細資訊，請參閱[從 CloudFormation 主控台建立堆疊](#)。

但這些方法僅驗證您範本的語法，而不會驗證您為資源指定的屬性數值。

其他驗證工具

如需更複雜的驗證和最佳實務檢查，您可以使用下列其他工具：

- [CloudFormation Linter \(cfn-lint\)](#) – 依據 [CloudFormation 資源提供者結構描述](#) 驗證範本。包括檢查資源屬性的有效值，以及最佳實務。
- [CloudFormation Rain \(rain fmt\)](#) – 將您的 CloudFormation 範本格式化為一致的標準，或重新格式化從 JSON 到 YAML (或 YAML 到 JSON) 的範本。使用 YAML 時會保留註解，並在可行的情況下將內建函數切換為簡短語法。

開始使用範本

若要開始建立 CloudFormation 範本，請遵循下列步驟：

1. 選擇資源 – 識別您要包含在堆疊中的 AWS 資源，例如 EC2 執行個體、VPCs、安全群組等。
2. 撰寫範本 – 以 JSON 或 YAML 格式撰寫範本，定義資源及其屬性。
3. 儲存範本 – 在本機使用副檔名儲存範本，例如：`.json`、`.yaml` 或 `.txt`。
4. 驗證範本 – 使用 [驗證範本](#) 章節所述的方法驗證範本。
5. 建立堆疊 – 使用經驗證的範本建立堆疊。

規劃使用 CloudFormation 範本參考

撰寫範本時，您可以在《[AWS 資源與屬性類型參考](#)》中，找到不同資源類型的詳細語法文件。

您的堆疊範本通常需要使用內建函數，指派執行時期才會可用的屬性數值，並使用特殊屬性控制資源的行為。撰寫範本時，可參考下列資源取得指引：

- [內建函數參考](#) – 常用的內建函數包括：
 - Ref – 擷取資源的參數值或實體 ID。
 - Sub – 將字串中的預留位置替換為實際數值。
 - GetAtt – 傳回範本中某項資源的屬性數值。
 - Join – 將一組數值串聯為單一字串。
- [資源屬性參考](#) – 部分常用的特殊屬性包括：
 - DependsOn – 使用此屬性以指定必須在另一項資源後建立的資源。
 - DeletionPolicy – 使用此屬性指定 CloudFormation 應如何處理刪除資源。

範例範本

CloudFormation 提供開放原始碼堆疊範本，供您入門使用。如需詳細資訊，請參閱 GitHub 網站上的 [CloudFormation 範例範本](#)。

請注意，這些範本並非可用於正式環境的版本。您應花時間了解其運作方式、根據自身需求調整，並確保其符合公司的合規標準。

此儲存庫中的每個範本都會通過 [CloudFormation Linter](#) (cfn-lint) 檢查，以及以Center for Internet Security (CIS)前 20 為基礎的一組基本 AWS CloudFormation Guard 規則，但某些規則的例外是，在合理的情況下，讓範例專注於單一使用案例。

CloudFormation 範本格式

您可用 JSON 或 YAML 格式編寫 CloudFormation 範本。這兩種格式用途相同，但在可讀性和複雜度方面各有獨特優勢。

- JSON – JSON 是一種輕量級資料交換格式，易於機器解析和生成。但對於人類而言，撰寫和閱讀可能較為繁瑣，尤其是在處理複雜組態時。在 JSON 中，範本透過巢狀的大括號 {} 和方括號 [] 來建構，以定義資源、參數及其他元件。其語法要求每個元素都必須明確宣告，這可能使範本較為冗長，但能確保嚴格遵循結構化格式。
- YAML – YAML 設計初衷是比 JSON 更易於人類閱讀，且更精簡。它使用縮排而非括號來表示巢狀關係，能更直觀地呈現資源和參數的階層結構。YAML 因其清晰性和易用性而備受青睞，尤其在處理較複雜的範本時。但 YAML 依賴縮排，若間距不一致可能導致錯誤，因此需仔細留意以確保正確性。

範本結構

CloudFormation 範本分為多個區段，每個區段用於存放特定類型的資訊。部分區段必須按特定順序宣告，其他區段則無順序要求。不過，在您建置範本時，使用下列範例所示的邏輯順序可能有幫助，因為某個區段中的值可能會參照先前區段中的值。

建立範本時，請勿重複宣告主要區段 (例如 Resources 區段)。雖然 CloudFormation 可能會接受範本，但在處理範本時會有未定義的行為，並且可能不正確地佈建資源或傳回莫名錯誤。

JSON

下列範例顯示包含所有可用區段的 JSON 格式範本結構。

```
{
  "AWSTemplateFormatVersion" : "version date",

  "Description" : "JSON string",

  "Metadata" : {
    template metadata
  },
```

```
"Parameters" : {  
    set of parameters  
},  
  
"Rules" : {  
    set of rules  
},  
  
"Mappings" : {  
    set of mappings  
},  
  
"Conditions" : {  
    set of conditions  
},  
  
"Transform" : {  
    set of transforms  
},  
  
"Resources" : {  
    set of resources  
},  
  
"Outputs" : {  
    set of outputs  
}  
}
```

YAML

下列範例顯示包含所有可用區段的 YAML 格式範本結構。

```
---  
AWSTemplateFormatVersion: version date  
  
Description:  
    String  
  
Metadata:  
    template metadata  
  
Parameters:
```

set of parameters

Rules:

set of rules

Mappings:

set of mappings

Conditions:

set of conditions

Transform:

set of transforms

Resources:

set of resources

Outputs:

set of outputs

說明

JSON 格式的範本不支援註解。JSON 本身並未設計註解語法，因此您無法在 JSON 結構中直接新增註解。但如果您需要加入說明性備註或文件，可考慮新增中繼資料。如需詳細資訊，請參閱 [Metadata 屬性](#)。

在 YAML 格式的範本中，您可以使用 # 符號加入行內註解。

以下範例示範有內嵌評論的 YAML 範本。

```
AWS::CloudFormation::Template
Description: A sample CloudFormation template with YAML comments.
# Resources section
Resources:
  MyEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      # Linux AMI
      ImageId: ami-1234567890abcdef0
      InstanceType: t2.micro
      KeyName: MyKey
      BlockDeviceMappings:
        - DeviceName: /dev/sdm
```

```
Ebs:
  VolumeType: io1
  Iops: 200
  DeleteOnTermination: false
  VolumeSize: 20
```

規格

CloudFormation 支援以下 JSON 和 YAML 規格：

JSON

CloudFormation 遵循 ECMA-404 JSON 標準。如需 JSON 格式的詳細資訊，請參閱 <http://www.json.org>。

YAML

CloudFormation 支援 YAML 1.1 版規格，但有幾個例外狀況。CloudFormation 不支援以下功能：

- binary、omap、pairs、set 和 timestamp 標籤
- 別名
- 雜湊合併

如需 YAML 的詳細資訊，請參閱 <https://yaml.org/>。

進一步了解

對於您在範本中指定的每個資源，都需使用 JSON 或 YAML 的特定語法規則定義其屬性和值。如需每種格式之範本語法的詳細資訊，請參閱 [CloudFormation 範本區段](#)。

在 CloudFormation 範本中使用常規表達式

可以在 CloudFormation 範本的多個位置中使用規則表達式 (通常稱為 regexe)，例如建立範本 [參數](#) 時可用於 AllowedPattern 屬性。

CloudFormation 中的所有規則表達式皆符合 Java regex 語法。如需 Java regex 語法及其建構的全面描述，請參閱 java.util.regex.Pattern。

如果使用 JSON 語法撰寫 CloudFormation 範本，則必須新增額外的反斜線，以轉義規則表達式中的任何反斜線字元 (\)。這是因為 JSON 將反斜線解譯為轉義字元，而且您需要將其轉義，以確保其在規則表達式中被視為常值反斜線。

舉例來說，若您要在 JSON 範本的常規表達式中加入 \d 以比對數字字元，則必須將該字元編寫為 \\d。

在下列範例中，AllowedPattern 屬性會指定符合四個連續數字字元 (\d{4}) 的規則表達式。不過，由於在 JSON 範本中定義規則表達式，因此需要使用額外的反斜線 (\\d) 來轉義反斜線字元。

```
{
  "Parameters": {
    "MyParameter": {
      "Type": "String",
      "AllowedPattern": "\\d{4}"
    }
  }
}
```

如果使用 YAML 語法撰寫 CloudFormation 範本，則必須使用單引號 (') 括住規則表達式。不需要額外的轉義。

```
Parameters:
  MyParameter:
    Type: String
    AllowedPattern: '\d{4}'
```

Note

CloudFormation 中的規則表達式只能在特定內容中用於驗證，例如 AllowedPattern。不支援其作為 CloudFormation 內建函數中的模式比對操作，例如 Fn::Equals，其僅執行確切的字串比較，而非模式比對。

CloudFormation 範本區段

每個 CloudFormation 範本都包含一個或多個區段，每個區段都有特定用途。

資源區段是每個 CloudFormation 範本的必要項目，也是範本的核心。本節指定堆疊資源和其屬性，例如 Amazon EC2 執行個體或 Amazon S3 儲存貯體。每個資源都會以唯一的邏輯 ID、類型和特定設定詳細資訊來定義。

參數區段雖然是選用的，但在提升範本彈性方面扮演重要角色。它可讓使用者在建立或更新堆疊時於執行時期傳遞值。這些參數可在 Resources 和 Outputs 區段中參考，無需修改範本本身即可實現自訂。例如，您可使用參數指定不同部署之間會變更的執行個體類型或環境設定。

輸出同樣為選用，用於定義檢視堆疊屬性時傳回的值。輸出會提供資源識別碼或 URL 等實用資訊，可用於操作用途或與其他堆疊整合。此區段協助使用者擷取並使用範本建立之資源的重要詳細資訊。

其他選用區段包括映射，其功能類似查閱表格，用於管理條件值。透過映射，您可以定義索引鍵/值對，並將其與 Resources 和 Outputs 區段中的 `Fn::FindInMap` 內建函數搭配使用。這適用於您需要根據 AWS 區域 或 環境等條件調整組態的情況。

中繼資料和規則區段雖較少使用，但提供額外功能。Metadata 可包含有關範本的額外資訊，而 Rules 會在建立或更新堆疊時驗證單一參數或參數組合，確保其符合特定標準。條件區段透過根據環境類型等條件控制是否建立特定資源或為屬性指派值，進一步提升彈性。

最後，轉換區段可用於在範本處理期間套用巨集。對於無伺服器應用程式（也稱為 Lambda 應用程式），它會指定要使用的 [AWS 無伺服器應用程式模型 \(AWS SAM\)](#) 版本。當您指定轉換時，您可以使用 AWS SAM 語法來宣告範本中的資源。此模型定義您可以使用的語法及其處理方式。您也可以使用 `AWS::Include` 轉換，以包含與主要 CloudFormation 範本分開存放的範本程式碼片段。

下列主題提供了關於使用每個區段的更多資訊和範例。

主題

- [CloudFormation 範本 Resources 語法](#)
- [CloudFormation 範本 Parameters 語法](#)
- [CloudFormation 範本 Outputs 語法](#)
- [CloudFormation 範本 Mappings 語法](#)
- [CloudFormation 範本 Metadata 語法](#)
- [CloudFormation 範本 Rules 語法](#)
- [CloudFormation 範本 Conditions 語法](#)
- [CloudFormation 範本 Transform 區段](#)
- [CloudFormation 範本格式版本語法](#)
- [CloudFormation 範本 Description 語法](#)

CloudFormation 範本 Resources 語法

Resources 區段是 CloudFormation 範本中所需的頂層區段。它會宣告您希望 CloudFormation 在堆疊中佈建和設定 AWS 的資源。

語法

Resources 區段使用下列語法：

JSON

```
"Resources" : {  
  "LogicalResourceName1" : {  
    "Type" : "AWS::ServiceName::ResourceType",  
    "Properties" : {  
      "PropertyName1" : "PropertyValue1",  
      ...  
    }  
  },  
  
  "LogicalResourceName2" : {  
    "Type" : "AWS::ServiceName::ResourceType",  
    "Properties" : {  
      "PropertyName1" : "PropertyValue1",  
      ...  
    }  
  }  
}
```

YAML

```
Resources:  
  LogicalResourceName1:  
    Type: AWS::ServiceName::ResourceType  
    Properties:  
      PropertyName1: PropertyValue1  
      ...  
  
  LogicalResourceName2:  
    Type: AWS::ServiceName::ResourceType  
    Properties:  
      PropertyName1: PropertyValue1
```

...

邏輯 ID (也稱為邏輯名稱)

在 CloudFormation 範本內，資源會以其邏輯資源名稱來識別。這些名稱必須是英數字元 (A-Za-z0-9)，而且在範本內必須是唯一的。邏輯名稱可用於參考範本其他區段的資源。

Resource Type (資源類型)

每個資源必須有 Type 屬性，以定義該 AWS 資源的種類。Type 屬性擁有的格式為 `AWS::ServiceName::ResourceType`。例如，Amazon S3 儲存貯體的 Type 屬性為 `AWS::S3::Bucket`。

如需受支援資源類型的完整清單，請參閱 [AWS 資源和屬性類型參考](#)。

資源屬性

資源屬性是您可指定的其他選項，用於定義特定資源類型的組態詳細資訊。有些屬性是必要的，有些則為選用。某些屬性具有預設值，因此指定這些屬性為選用。

如需每個資源類型所支援屬性的詳細資訊，請參閱《[AWS 資源和屬性類型參考](#)》中的主題。

屬性值可以是常值字串、字串清單、動態參考、布林值、參數參考、虛擬參考，或函數所傳回的值。下列範例顯示如何宣告不同的屬性值類型：

JSON

```
"Properties" : {  
    "String" : "A string value",  
    "Number" : 123,  
    "LiteralList" : [ "first-value", "second-value" ],  
    "Boolean" : true  
}
```

YAML

```
Properties:  
  String: A string value  
  Number: 123  
  LiteralList:
```



```
- first-value
- second-value
Boolean: true
```

實體 ID

除了邏輯 ID 之外，特定資源也會有實體 ID，這是該資源的實際指派名稱 (例如 EC2 執行個體 ID 或 S3 儲存貯體名稱)。使用實體 ID 來識別 CloudFormation 範本外部的資源，但只在建立資源之後。例如，假設您為 EC2 執行個體資源指定邏輯 ID `MyEC2Instance`。當 CloudFormation 建立執行個體時，CloudFormation 會自動產生並指派實體 ID (例如 `i-1234567890abcdef0`) 給執行個體。您可以使用此實體 ID 來識別執行個體，以及使用 Amazon EC2 主控台來檢視其屬性 (例如 DNS 名稱)。

對於 Amazon S3 儲存貯體及許多其他資源，若您未明確指定，CloudFormation 會自動為資源產生唯一的實體名稱。此實體名稱基於 CloudFormation 堆疊的名稱、CloudFormation 範本中指定的資源邏輯名稱，以及唯一 ID 的組合。例如，若您的 Amazon S3 儲存貯體在名為 `MyStack` 的堆疊中具有邏輯名稱 `MyBucket`，CloudFormation 可能會將儲存貯體命名為下列實體名稱 `MyStack-MyBucket-abcdefghijkl`。

針對支援自訂名稱的資源，您可以指派自己的實體名稱，協助您快速識別資源。例如，您可以將存放日誌的 S3 儲存貯體命名為 `MyPerformanceLogs`。如需詳細資訊，請參閱[名稱類型](#)。

參考資源

您通常需要根據另一個資源的名稱或屬性，設定某個資源的屬性。例如，您可建立使用 EC2 安全群組的 EC2 執行個體，或 S3 儲存貯體支援的 CloudFront 分佈。所有這些資源都可在相同的 CloudFormation 範本中建立。

CloudFormation 提供內建函數，您可以用來參考其他資源及其屬性。這些函數可讓您在資源之間建立相依關係，並將值從一個資源傳遞到另一個資源。

Ref 函數

Ref 函數通常用於擷取相同 CloudFormation 範本中所定義資源的識別屬性。傳回的內容要取決於資源的類型。對於大多數資源，它會傳回資源的實體名稱。然而，對於某些資源類型，它可能會傳回不同的值，例如 `AWS::EC2::EIP` 資源的 IP 位址或 Amazon SNS 主題的 Amazon Resource Name (ARN)。

以下範例會示範屬性中 Ref 函數的使用方式。在每個範例中，Ref 函數會傳回範本中其他位置所宣告 `LogicalResourceName` 資源的實際名稱。YAML 範例中的 `!Ref` 語法範例只是撰寫 Ref 函數的較短方式。

JSON

```
"Properties" : {  
    "PropertyName" : { "Ref" : "LogicalResourceName" }  
}
```

YAML

```
Properties:  
  PropertyName1:  
    Ref: LogicalResourceName  
  PropertyName2: !Ref LogicalResourceName
```

如需 Ref 函數的詳細資訊，請參閱 [Ref](#)。

Fn::GetAtt 函數

如果為資源傳回的參數或值完全是您想要的，Ref 函數會很有幫助。不過，您可能會需要資源的其他屬性。例如，如果您想要建立有 S3 來源的 CloudFront 分佈，您需要使用 DNS 樣式地址指定儲存貯體位置。多項有其他屬性的資源，其值可用於範本中。您可以使用 Fn::GetAtt 函數取得這些屬性。

以下範例會示範屬性中 GetAtt 函數的使用方式。Fn::GetAtt 函數需要兩個參數擷取資源邏輯名稱及屬性名稱。YAML 範例中的 !GetAtt 語法範例只是撰寫 GetAtt 函數的較短方式。

JSON

```
"Properties" : {  
    "PropertyName" : {  
        "Fn::GetAtt" : [ "LogicalResourceName", "AttributeName" ]  
    }  
}
```

YAML

```
Properties:  
  PropertyName1:  
    Fn::GetAtt:  
      - LogicalResourceName  
      - AttributeName  
  PropertyName2: !GetAtt LogicalResourceName.AttributeName
```

如需 GetAtt 函數的詳細資訊，請參閱 [Fn::GetAtt](#)。

範例

下列範例說明如何宣告資源，以及 CloudFormation 範本如何參考定義在相同範本中的其他資源，以及現有的 AWS 資源。

主題

- [宣告具有自訂名稱的單一資源](#)
- [使用 Ref 函數參考其他資源](#)
- [透過 Fn::GetAtt 函數參考資源屬性](#)

宣告具有自訂名稱的單一資源

以下範例會宣告邏輯名稱為 MyBucket 的 AWS::S3::Bucket 類型單一資源。BucketName 屬性設定為 *amzn-s3-demo-bucket*，您應以 S3 儲存貯體所需的名稱取代。

若您使用此資源宣告來建立堆疊，CloudFormation 將使用預設設定建立 Amazon S3 儲存貯體。至於 Amazon EC2 執行個體或 Auto Scaling 群組等其他資源，CloudFormation 需要更多資訊。

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "BucketName": "amzn-s3-demo-bucket"
      }
    }
  }
}
```

YAML

```
Resources:
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
```

使用 **Ref** 函數參考其他資源

下列範例顯示定義 EC2 執行個體與安全群組的資源宣告。Ec2Instance 資源會使用 Ref 函數參考 InstanceSecurityGroup 資源，做為其 SecurityGroupIds 屬性的一部分。它還包含了範本中未宣告的現有安全群組 (sg-12a4c434)。您要使用文字字串參考現有的 AWS 資源。

JSON

```
{
  "Resources": {
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "SecurityGroupIds": [
          {
            "Ref": "InstanceSecurityGroup"
          },
          "sg-12a4c434"
        ],
        "KeyName": "MyKey",
        "ImageId": "ami-1234567890abcdef0"
      }
    },
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    }
  }
}
```

YAML

```
Resources:
```

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    SecurityGroupIds:
      - !Ref InstanceSecurityGroup
      - sg-12a4c434
    KeyName: MyKey
    ImageId: ami-1234567890abcdef0
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
```

透過 **Fn::GetAtt** 函數參考資源屬性

下列範例顯示定義 CloudFront 分佈資源及 S3 儲存貯體的資源宣告。MyDistribution 資源會使用 Fn::GetAtt 函數取得儲存貯體的 DomainName 屬性，以指定 MyBucket 資源的 DNS 名稱。您要通知 Fn::GetAtt 函數在陣列中列出它的兩個參數。對於需要多個參數的函數，您使用陣列指定它們的參數。

JSON

```
{
  "Resources": {
    "MyBucket": {
      "Type": "AWS::S3::Bucket"
    },
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "Origins": [
            {
              "DomainName": {
                "Fn::GetAtt": [
                  "MyBucket",
                  "DomainName"
                ]
              }
            }
          ]
        }
      }
    }
  }
}
```

```

        },
        "Id": "MyS3Origin",
        "S3OriginConfig": {}
    }
],
"Enabled": "true",
"DefaultCacheBehavior": {
    "TargetOriginId": "MyS3Origin",
    "ForwardedValues": {
        "QueryString": "false"
    },
    "ViewerProtocolPolicy": "allow-all"
}
}
}
}
}
}
}
}
}
}

```

YAML

```

Resources:
  MyBucket:
    Type: AWS::S3::Bucket
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: !GetAtt
            - MyBucket
            - DomainName
            Id: MyS3Origin
            S3OriginConfig: {}
        Enabled: 'true'
        DefaultCacheBehavior:
          TargetOriginId: MyS3Origin
          ForwardedValues:
            QueryString: 'false'
          ViewerProtocolPolicy: allow-all

```

CloudFormation 範本 Parameters 語法

使用選用的 Parameters 區段自訂您的範本。透過參數，您可以在每次建立或更新堆疊時將自訂值輸入至您的範本。透過在範本中使用參數，您能夠建立可重複使用且靈活的範本，這些範本可針對特定案例量身打造。

透過定義適當類型的參數，您可在使用主控台建立堆疊時，從現有資源的識別碼清單中選擇。如需詳細資訊，請參閱[使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)。

參數是指定堆疊資源屬性值的常用方式。但是，有些設定可能依存於區域，或有些複雜，讓使用者因其他條件或依存項目而無法找出。在這些情況下，您可能要在範本本身中放入一些邏輯，讓使用者可以指定簡單的值 (或完全不指定)，以取得他們希望的結果，如透過映射來實現。如需詳細資訊，請參閱[CloudFormation 範本 Mappings 語法](#)。

語法

您可在範本的 Parameters 區段中宣告參數，該區段使用下列一般語法：

JSON

```
"Parameters" : {  
  "ParameterLogicalID" : {  
    "Description": "Information about the parameter",  
    "Type" : "DataType",  
    "Default" : "value",  
    "AllowedValues" : ["value1", "value2"]  
  }  
}
```

YAML

```
Parameters:  
  ParameterLogicalID:  
    Description: Information about the parameter  
    Type: DataType  
    Default: value  
    AllowedValues:  
      - value1  
      - value2
```

參數包含定義其值的屬性清單以及針對其值的限制條件。唯一的必要屬性是 `Type`，可以是 `String`、`Number` 或 `CloudFormation` 提供的參數類型。您也可以新增 `Description` 屬性，描述您應該指定的值類型。您使用建立堆疊精靈中的範本時，指定參數頁面上會出現參數的名稱和描述。

Note

根據預設，CloudFormation 主控台會依據參數的邏輯 ID 字母順序來列出參數。若要覆寫此預設排序，並將相關參數分組在一起，您可在範本中使用 `AWS::CloudFormation::Interface` 中繼資料金鑰。如需詳細資訊，請參閱[使用 `AWS::CloudFormation::Interface` 中繼資料組織 CloudFormation 參數](#)。

針對有預設值的參數，除非使用者指定另一個值，否則 CloudFormation 會使用預設值。若您省略預設屬性，使用者必須指定該參數的值。但是，要求使用者輸入值並不能確保該值有效。若要驗證參數的值，您可以宣告限制條件或指定 AWS 特定的參數類型。

對於沒有預設值的參數，使用者必須在建立堆疊時，指定金鑰名稱值。否則，CloudFormation 無法建立堆疊，且會擲出例外狀況：

```
Parameters: [KeyName] must have values
```

Properties

AllowedPattern

規則表達式，代表 `String` 或 `CommaDelimitedList` 類型允許的模式。當套用於類型 `String` 參數時，模式必須匹配提供的整個參數值。當套用於類型 `CommaDelimitedList` 參數時，模式必須匹配清單中的每個值。

必要：否

AllowedValues

陣列，包含參數的允許值清單。當套用於類型 `String` 參數時，參數值必須是允許值之一。當套用於類型 `CommaDelimitedList` 參數時，清單中的每個值必須是指定的允許值之一。

必要：否

 Note

如果您使用 YAML，並且想要將 Yes 和 No 字串用於 AllowedValues，請使用單引號來防止 YAML 剖析器考慮這些布林值。

ConstraintDescription

字串，說明違反限制時的限制。例如，具有允許模式 [A-Za-z0-9]+ 的參數會在使用者指定無效值時顯示下列錯誤訊息，但沒有限制描述：

```
Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+
```

透過新增限制描述，例如只能包含字母 (大寫和小寫) 及數字，您可以顯示下列自訂錯誤訊息：

```
Malformed input-Parameter MyParameter must only contain uppercase and lowercase letters and numbers
```

必要：否

Default

建立堆疊時，範本要在未指定值時使用的適當類型值。如果您定義參數的限制，則必須指定遵循這些限制的值。

必要：否

Description

多達 4000 個字元的字串，可說明參數。

必要：否

MaxLength

整數值，決定您想要針對 String 類型允許的最大字元數。

必要：否

MaxValue

數值，決定您想要針對 Number 類型允許的最大數值。

必要：否

MinLength

整數值，決定您想要針對 String 類型允許的最小字元數。

必要：否

MinValue

數值，決定您想要針對 Number 類型允許的最小數值。

必要：否

NoEcho

是否遮罩參數值以防止其在主控台、命令列工具或 API 中顯示。若您將 NoEcho 屬性設為 true，CloudFormation 會將任何描述堆疊或堆疊事件呼叫所傳回的參數值以星號 (****) 遮罩，但儲存在以下指定位置中的資訊除外。

必要：否

Important

使用 NoEcho 屬性不會遮罩任何儲存在下列資訊中的資訊：

- Metadata 範本區段。CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。若要取得更多資訊，請參閱[中繼資料](#)。
 - Outputs 範本區段。如需詳細資訊，請參閱[輸出](#)。
 - 資源定義的 Metadata 屬性。如需詳細資訊，請參閱[Metadata 屬性](#)。
- 我們強烈建議您不要使用這些機制來包含敏感資訊，例如密碼或秘密。

Important

我們建議您不要直接在 CloudFormation 範本中嵌入敏感資訊，而是在堆疊範本中使用動態參數來參考在 CloudFormation 外部存放和管理的敏感資訊，例如在 AWS Systems Manager 參數存放區或 AWS Secrets Manager 中。

如需詳細資訊，請參閱[請勿在範本中內嵌認證](#)的最佳實務。

Important

強烈建議不要在作為資源主要識別碼一部分的資源屬性中包含 NoEcho 參數或任何敏感資料。

形成主要資源識別碼的屬性中包含 NoEcho 參數時，CloudFormation 可使用主要資源識別碼中的實際純文字值。此資源 ID 可能會出現在任何衍生輸出或目的地中。

若要判斷哪些資源屬性組成資源類型的主要識別碼，請參閱 [AWS 資源和屬性類型參考](#) 中該資源的資源參考文件。在 Return values (傳回值) 區段中，Ref 函數傳回值表示組成資源類型主要識別碼的資源屬性。

Type

參數的資料類型 (DataType)。

必要：是

CloudFormation 支援下列參數類型：

String

常值字串。您可以使用下列屬性宣告限制條件：

MinLength、MaxLength、Default、AllowedValues 和 AllowedPattern。

例如，使用者可以指定 "MyUserName"。

Number

整數或浮點數。CloudFormation 會將參數值驗證為數字；不過，當您在範本中的其他位置使用參數時 (例如，使用 Ref 內建函數)，參數值會變成字串。

您可以使用下列屬性宣告限制條件：MinValue、MaxValue、Default 和 AllowedValues。

例如，使用者可以指定 "8888"。

List<Number>

以逗號分隔的整數或浮點數陣列。CloudFormation 會將參數值驗證為數字；不過，當您在範本中的其他位置使用參數時 (例如，使用 Ref 內建函數)，參數值會變成字串清單。

例如，使用者可以指定 "80,20"，而 Ref 會產生 ["80","20"]。

CommaDelimitedList

以逗號分隔的常值字串陣列。字串總數應該比逗號總數多一個。此外，還會裁剪每個成員字串的空格。

例如，使用者可以指定 "test,dev,prod"，而 Ref 會產生 ["test","dev","prod"]。

AWS 特定參數類型

AWS 值，例如 Amazon EC2 金鑰對名稱和 VPC IDs。如需詳細資訊，請參閱[在執行時期指定現有資源](#)。

Systems Manager 參數類型

對應至 Systems Manager 參數存放區中現有參數的參數。您可以將 Systems Manager 參數索引鍵指定為 Systems Manager 參數類型的值，而 CloudFormation 會擷取參數存放區中的最新值來用於堆疊。如需詳細資訊，請參閱[在執行時期指定現有資源](#)。

參數的一般要求

以下是使用參數時適用的要求：

- CloudFormation 範本中最多可以有 200 個參數。
- 每個參數都必須獲指定邏輯名稱 (也稱為邏輯 ID)，其必須為英數字元，而且在範本的所有邏輯名稱之間必須是唯一的。
- 每個參數都必須獲指派 CloudFormation 所支援的參數類型。如需詳細資訊，請參閱[類型](#)。
- 每個參數在執行時期都必須獲指派值，以讓 CloudFormation 成功佈建堆疊。除非提供另一個值，否則您可以選擇性指定預設值以供 CloudFormation 使用。
- 參數必須在相同的範本內宣告和參考。您可以參考範本之 Resources 和 Outputs 區段中的參數。

範例

主題

- [簡單字串參數](#)
- [密碼參數](#)
- [參考參數](#)
- [逗號分隔清單參數](#)
- [逗號分隔清單參數的傳回值](#)

簡單字串參數

下列範例宣告類型 String 名為 InstanceTypeParameter 的參數。此參數可讓您指定堆疊的 Amazon EC2 執行個體類型。若在堆疊建立或更新期間未提供任何值，CloudFormation 會使用預設值 t2.micro。

JSON

```
"Parameters" : {
  "InstanceTypeParameter" : {
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro.",
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"]
  }
}
```

YAML

```
Parameters:
  InstanceTypeParameter:
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
```

密碼參數

下列範例宣告類型 String 名為 DBPwd 的參數，沒有預設值。NoEcho 屬性設定為 true，以防止參數值在堆疊描述中顯示。可指定的最小長度為 1，而可指定的最大長度為 41。此模式允許大小寫字母字元和數字。此範例也說明 AllowedPattern 屬性使用規則表達式。

JSON

```
"Parameters" : {
  "DBPwd" : {
    "NoEcho" : "true",
    "Description" : "The database admin account password",
    "Type" : "String",
```

```
    "MinLength" : "1",
    "MaxLength" : "41",
    "AllowedPattern" : "^[a-zA-Z0-9]*$"
  }
}
```

YAML

```
Parameters:
  DBPwd:
    NoEcho: true
    Description: The database admin account password
    Type: String
    MinLength: 1
    MaxLength: 41
    AllowedPattern: ^[a-zA-Z0-9]*$
```

參考參數

您可以使用 Ref 內建函數來參考參數，而 CloudFormation 使用參數值來佈建堆疊。您可以參考相同範本之 Resources 和 Outputs 區段中的參數。

在下列範例中，EC2 執行個體資源的 InstanceType 屬性參考 InstanceTypeParameter 參數值：

JSON

```
"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "InstanceType" : { "Ref" : "InstanceTypeParameter" },
    "ImageId" : "ami-0ff8a91507f77f867"
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType:
      Ref: InstanceTypeParameter
```

```
ImageId: ami-0ff8a91507f77f867
```

逗號分隔清單參數

當您需要為單一屬性提供多個值時，`CommaDelimitedList` 參數類型會非常實用。下列範例宣告名為 `DbSubnetIpBlocks` 的參數，預設值為三個以逗號分隔的 CIDR 區塊。

JSON

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three CIDR blocks",
    "Type": "CommaDelimitedList",
    "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

YAML

```
Parameters:
  DbSubnetIpBlocks:
    Description: "Comma-delimited list of three CIDR blocks"
    Type: CommaDelimitedList
    Default: "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
```

逗號分隔清單參數的傳回值

若要參照的逗號分隔清單中的特定值，請在範本的 `Fn::Select` 區段中使用 `Resources` 內建函數。傳遞所要之物件的索引值和物件清單，如下列範例所示。

JSON

```
{
  "Parameters": {
    "VPC": {
      "Type": "String",
      "Default": "vpc-123456"
    },
    "VpcAzs": {
      "Type": "CommaDelimitedList",
      "Default": "us-west-2a, us-west-2b, us-west-2c"
    },
    "DbSubnetIpBlocks": {
```

```

        "Type": "CommaDelimitedList",
        "Default": "172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26"
    }
},
"Resources": {
    "DbSubnet1": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "AvailabilityZone": {
                "Fn::Select": [
                    0,
                    {
                        "Ref": "VpcAzs"
                    }
                ]
            },
            "VpcId": {
                "Ref": "VPC"
            },
            "CidrBlock": {
                "Fn::Select": [
                    0,
                    { "Ref": "DbSubnetIpBlocks" }
                ]
            }
        }
    },
    "DbSubnet2": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "AvailabilityZone": {
                "Fn::Sub": [
                    "${AWS::Region}${AZ}",
                    {
                        "AZ": {
                            "Fn::Select": [
                                1,
                                { "Ref": "VpcAzs" }
                            ]
                        }
                    }
                ]
            },
            "VpcId": {

```



```

        "Ref": "VPC"
      },
      "CidrBlock": {
        "Fn::Select": [
          1,
          { "Ref": "DbSubnetIpBlocks" }
        ]
      }
    },
    "DbSubnet3": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "AvailabilityZone": {
          "Fn::Sub": [
            "${AWS::Region}${AZ}",
            {
              "AZ": {
                "Fn::Select": [
                  2,
                  { "Ref": "VpcAzs" }
                ]
              }
            }
          ]
        },
        "VpcId": {
          "Ref": "VPC"
        },
        "CidrBlock": {
          "Fn::Select": [
            2,
            { "Ref": "DbSubnetIpBlocks" }
          ]
        }
      }
    }
  }
}

```

YAML

Parameters:

```
VPC:
  Type: String
  Default: vpc-123456
VpcAzs:
  Type: CommaDelimitedList
  Default: us-west-2a, us-west-2b, us-west-2c
DbSubnetIpBlocks:
  Type: CommaDelimitedList
  Default: 172.16.0.0/26, 172.16.0.64/26, 172.16.0.128/26
Resources:
  DbSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Select
        - 0
        - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 0
        - !Ref DbSubnetIpBlocks
  DbSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
            - 1
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 1
        - !Ref DbSubnetIpBlocks
  DbSubnet3:
    Type: AWS::EC2::Subnet
    Properties:
      AvailabilityZone: !Sub
        - ${AWS::Region}${AZ}
        - AZ: !Select
            - 2
            - !Ref VpcAzs
      VpcId: !Ref VPC
      CidrBlock: !Select
        - 2
```

- !Ref DbSubnetIpBlocks

相關資源

CloudFormation 也支援使用動態參考來動態指定屬性值。例如，您可能需要參考在 Systems Manager 參數存放區中儲存的安全字串。如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

您也可以使用 Ref 或 Sub 函數，使用虛擬參數來動態填入值。如需詳細資訊，請參閱[使用虛擬參數取得 AWS 值](#)。

CloudFormation 範本 Outputs 語法

選用 Outputs 區段會宣告堆疊的輸出值。這些輸出值可透過多種方式使用：

- 擷取資源的重要詳細資訊 – 輸出是擷取資源重要資訊的便捷方式。例如，您可以輸出堆疊的 S3 儲存貯體名稱，以更輕鬆地找到儲存貯體。您可以在 CloudFormation 主控台的輸出索引標籤中檢視輸出值，或使用 [describe-stacks](#) CLI 命令檢視輸出值。
- 跨堆疊參考 – 您可以將輸出值匯入其他堆疊，以[在堆疊之間建立參考](#)。這在您需要跨多個堆疊共用資源或組態時相當實用。

Important

CloudFormation 不會對您在 Outputs 區段中包含的任何資訊進行修改或混淆。我們強烈建議您不要使用此區段來輸出敏感資訊，例如密碼或秘密。

堆疊操作完成後，可使用輸出值。堆疊狀態處於任何 IN_PROGRESS [狀態](#)時，堆疊輸出值將不可用。我們不建議在服務執行階段和堆疊輸出值之間建立相依性，因為輸出值可能無法在任何時候都可用。

語法

Outputs 區段由索引鍵名稱 Outputs 組成。您最多可以在範本中宣告 200 個輸出。

下列範例示範 Outputs 區段的結構。

JSON

使用括號括住所有輸出宣告。使用逗號分隔多個輸出。

```
"Outputs" : {
```

```
"OutputLogicalID" : {  
  "Description" : "Information about the value",  
  "Value" : "Value to return",  
  "Export" : {  
    "Name" : "Name of resource to export"  
  }  
}  
}
```

YAML

```
Outputs:  
  OutputLogicalID:  
    Description: Information about the value  
    Value: Value to return  
    Export:  
      Name: Name of resource to export
```

輸出欄位

Outputs 區段可以包含下列欄位。

邏輯 ID (也稱為邏輯名稱)

目前輸出的識別碼。邏輯 ID 必須是英數字元 (a-z、A-Z、0-9)，而且在範本內必須是唯一的。

Description (選用)

String 類型，說明輸出值。描述宣告值的長度必須是介於 0 與 1024 位元組之間的常值字串。您不能使用參數或函數來指定描述。

Value (必要)

[describe-stacks](#) 命令所傳回的屬性值。輸出值可以包含常值、參數參考、虛擬參數、映射值或內建函數。

Export (選用)

要針對跨堆疊參考匯出的資源輸出名稱。

您可以使用內部函數來自訂匯出的 Name 值。

如需詳細資訊，請參閱[從部署的 CloudFormation 堆疊取得匯出的輸出](#)。

若要建立條件與輸出的關聯，請在範本的 [Conditions](#) 區段中定義條件。

範例

下列範例說明堆疊輸出的運作方式。

主題

- [堆疊輸出](#)
- [使用 Fn::Sub 自訂匯出名稱](#)
- [使用 Fn::Join 自訂匯出名稱](#)
- [傳回使用 Fn::Join 建構的 URL](#)

堆疊輸出

在下列範例中，只有在 BackupLoadBalancerDNSName 條件為 true 時，名為 BackupLoadBalancer 的輸出才會傳回邏輯 ID 為 CreateProdResources 之資源的 DNS 名稱。名為 InstanceID 的輸出會傳回邏輯 ID 為 EC2Instance 之 EC2 執行個體的 ID。

JSON

```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ] },
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

YAML

```
Outputs:
  BackupLoadBalancerDNSName:
    Description: The DNSName of the backup load balancer
    Value: !GetAtt BackupLoadBalancer.DNSName
    Condition: CreateProdResources
  InstanceID:
    Description: The Instance ID
    Value: !Ref EC2Instance
```

使用 **Fn::Sub** 自訂匯出名稱

在下列範例中，名為 StackVPC 的輸出會傳回 VPC 的 ID，然後匯出跨堆疊參考的值，亦即將名稱 VPCID 附加至堆疊名稱。

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : {"Fn::Sub": "${AWS::StackName}-VPCID" }
    }
  }
}
```

YAML

```
Outputs:
  StackVPC:
    Description: The ID of the VPC
    Value: !Ref MyVPC
    Export:
      Name: !Sub "${AWS::StackName}-VPCID"
```

如需 Fn::Sub 函數的詳細資訊，請參閱「[Fn::Sub](#)」。

使用 **Fn::Join** 自訂匯出名稱

您還可以使用 Fn::Join 函數根據參數、資源屬性和其他字串來建構值。

下列範例使用 Fn::Join 函數來自訂匯出名稱，而非 Fn::Sub 函數。範例 Fn::Join 函數使用冒號做為分隔符號，將堆疊名稱與名稱 VPCID 串連。

JSON

```
"Outputs" : {
  "StackVPC" : {
    "Description" : "The ID of the VPC",
    "Value" : { "Ref" : "MyVPC" },
    "Export" : {
      "Name" : { "Fn::Join" : [ ":", [ { "Ref" : "AWS::StackName" }, "VPCID" ] ] }
    }
  }
}
```

```
    }  
  }  
}
```

YAML

```
Outputs:  
  StackVPC:  
    Description: The ID of the VPC  
    Value: !Ref MyVPC  
    Export:  
      Name: !Join [ ":", [ !Ref "AWS::StackName", VPCID ] ]
```

如需 `Fn::Join` 函數的詳細資訊，請參閱「[Fn::Join](#)」。

傳回使用 **Fn::Join** 建構的 URL

在建立 WordPress 網站的範本的下列範例中，`InstallURL` 是 `Fn::Join` 函數呼叫傳回的字串，它會串連 `http://`、資源的 DNS 名稱 `ElasticLoadBalancer`，以及 `/wp-admin/install.php`。輸出值可能類似以下內容：

```
http://mywptests-elasticl-1gb51l6sl8y5v-206169572.aws-region.elb.amazonaws.com/wp-admin/install.php
```

JSON

```
{  
  "Outputs": {  
    "InstallURL": {  
      "Value": {  
        "Fn::Join": [  
          "",  
          [  
            "http://",  
            {  
              "Fn::GetAtt": [  
                "ElasticLoadBalancer",  
                "DNSName"  
              ]  
            },  
            "/wp-admin/install.php"  
          ]  
        ]  
      }  
    }  
  }  
}
```

```

        ]
      },
      "Description": "Installation URL of the WordPress website"
    }
  }
}

```

YAML

```

Outputs:
  InstallURL:
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt
          - ElasticLoadBalancer
          - DNSName
        - /wp-admin/install.php
    Description: Installation URL of the WordPress website

```

如需 `Fn::Join` 函數的詳細資訊，請參閱「[Fn::Join](#)」。

CloudFormation 範本 Mappings 語法

選用 Mappings 區段可協助您建立索引鍵/值對，用於根據特定條件或依賴關係指定數值。

Mappings 區段的一個常見使用案例是根據部署堆疊的 AWS 區域 來設定值。可以透過使用 `AWS::Region` 虛擬參數來實現這一點。`AWS::Region` 虛擬參數是 CloudFormation 解析為堆疊建立區域的值。當您建立堆疊時，CloudFormation 會解析虛擬參數。

要擷取映射中的數值，您可以在範本的 Resources 區段內使用 `Fn::FindInMap` 內建函數。

語法

Mappings 區段使用下列語法：

JSON

```

"Mappings" : {
  "MappingLogicalName" : {
    "Key1" : {
      "Name" : "Value1"
    },

```



```
"Key2" : {  
  "Name" : "Value2"  
},  
"Key3" : {  
  "Name" : "Value3"  
}  
}  
}
```

YAML

```
Mappings:  
  MappingLogicalName:  
    Key1:  
      Name: Value1  
    Key2:  
      Name: Value2  
    Key3:  
      Name: Value3
```

- MappingLogicalName 是映射的邏輯名稱。
- 在映射內，每個映射都是後面接著另一個映射的索引鍵。
- 該金鑰必須是名稱值對的映射和映射中唯一的映射。
- 名稱/值對是一個標籤，也是要映射的值。藉由命名這些值，您可以將多組的值映射至一個索引鍵。
- 映射中的索引鍵必須是常值字串。
- 值可以是類型 String 或 List。

Note

您不能在 Mappings 區段中包含參數、虛擬參數或內部函數。
如果堆疊目前沒有使用映射中的值，您將無法更新映射本身。必須涵蓋新增、修改或刪除資源等變更作業，才能進行更新。

範例

主題

- [基本映射](#)

- [具有多個值的映射](#)
- [從映射傳回值](#)
- [輸入參數和 Fn::FindInMap](#)

基本映射

下列範例顯示具有映射 Mappings 的 RegionToInstanceType 區段，內含映射至包含單一字串值之名稱值對的五個索引鍵。索引鍵是區域名稱。每個名稱/數值組都是 T 系列的執行個體類型，且在索引鍵所代表的區域中可用。名稱值對會有名稱 (在此範例中為 InstanceType) 和值。

JSON

```
"Mappings" : {
  "RegionToInstanceType" : {
    "us-east-1"      : { "InstanceType" : "t2.micro" },
    "us-west-1"      : { "InstanceType" : "t2.micro" },
    "eu-west-1"      : { "InstanceType" : "t2.micro" },
    "eu-north-1"     : { "InstanceType" : "t3.micro" },
    "me-south-1"     : { "InstanceType" : "t3.micro" }
  }
}
```

YAML

```
Mappings:
  RegionToInstanceType:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
```

具有多個值的映射

下列範例的區域索引鍵映射至兩組值：一個名為 MyAMI1，另一個名為 MyAMI2。

Note

這些範例中顯示的 AMI ID 是用作示範用途的預留位置。盡可能考慮使用 AWS Systems Manager 參數的動態參考作為 Mappings 區段的替代方案。若要避免在每次要使用變更的 AMI 時，使用新 ID 更新所有範本，請在建立或更新堆疊時，使用 AWS Systems Manager 參數擷取最新的 AMI ID。Systems Manager 中也提供最新版本的常用 AMI 作為公有參數。如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

JSON

```
"Mappings" : {
  "RegionToAMI" : {
    "us-east-1"      : { "MyAMI1" : "ami-12345678901234567", "MyAMI2" :
"ami-23456789012345678" },
    "us-west-1"      : { "MyAMI1" : "ami-34567890123456789", "MyAMI2" :
"ami-45678901234567890" },
    "eu-west-1"      : { "MyAMI1" : "ami-56789012345678901", "MyAMI2" :
"ami-67890123456789012" },
    "ap-southeast-1" : { "MyAMI1" : "ami-78901234567890123", "MyAMI2" :
"ami-89012345678901234" },
    "ap-northeast-1" : { "MyAMI1" : "ami-90123456789012345", "MyAMI2" :
"ami-01234567890123456" }
  }
}
```

YAML

```
Mappings:
  RegionToAMI:
    us-east-1:
      MyAMI1: ami-12345678901234567
      MyAMI2: ami-23456789012345678
    us-west-1:
      MyAMI1: ami-34567890123456789
      MyAMI2: ami-45678901234567890
    eu-west-1:
      MyAMI1: ami-56789012345678901
      MyAMI2: ami-67890123456789012
    ap-southeast-1:
      MyAMI1: ami-78901234567890123
```

```

MyAMI2: ami-89012345678901234
ap-northeast-1:
MyAMI1: ami-90123456789012345
MyAMI2: ami-01234567890123456

```

從映射傳回值

您可以使用 `Fn::FindInMap` 函數，以根據指定的索引鍵來傳回命名值。下列範例範本包含 Amazon EC2 資源，而其 `InstanceType` 屬性是由 `FindInMap` 函數指派。`FindInMap` 函數會將索引鍵指定為建立堆疊 AWS 區域的（使用 `AWS::Region` 虛擬參數），並將索引鍵 `InstanceType` 指定為要映射的值名稱。`ImageId` 會使用 Systems Manager 參數動態擷取最新的 Amazon Linux 2 AMI。如需這些虛擬參數的詳細資訊，請參閱[使用虛擬參數取得 AWS 值](#)。

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionToInstanceType" : {
      "us-east-1"      : { "InstanceType" : "t2.micro" },
      "us-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-west-1"      : { "InstanceType" : "t2.micro" },
      "eu-north-1"     : { "InstanceType" : "t3.micro" },
      "me-south-1"     : { "InstanceType" : "t3.micro" }
    }
  },
  "Resources" : {
    "myEC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap" : [ "RegionToInstanceType", { "Ref" :
"AWS::Region" }, "InstanceType" ]}
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09

```

```

Mappings:
  RegionToInstanceType:
    us-east-1:
      InstanceType: t2.micro
    us-west-1:
      InstanceType: t2.micro
    eu-west-1:
      InstanceType: t2.micro
    eu-north-1:
      InstanceType: t3.micro
    me-south-1:
      InstanceType: t3.micro
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}'
      InstanceType: !FindInMap [RegionToInstanceType, !Ref 'AWS::Region', InstanceType]

```

輸入參數和 **Fn::FindInMap**

下列範例範本示範如何使用多個映射建立 EC2 執行個體。範本使用巢狀映射，根據目標和環境類型 (Dev 或) 自動選取適當的執行個體類型 AWS 區域 和安全群組Prod。它也會使用 Systems Manager 參數來動態擷取最新的 Amazon Linux 2 AML。

JSON

```

{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "EnvironmentType" : {
      "Description" : "The environment type (Dev or Prod)",
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : [ "Dev", "Prod" ]
    }
  },
  "Mappings" : {
    "RegionAndEnvironmentToInstanceType" : {
      "us-east-1"      : { "Dev" : "t3.micro", "Prod" : "c5.large" },
      "us-west-1"      : { "Dev" : "t2.micro", "Prod" : "m5.large" }
    }
  },

```

```

    "RegionAndEnvironmentToSecurityGroup" : {
      "us-east-1"      : { "Dev" : "sg-12345678", "Prod" : "sg-abcdef01" },
      "us-west-1"      : { "Dev" : "sg-ghijkl23", "Prod" : "sg-45678abc" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "InstanceType" : { "Fn::FindInMap": [ "RegionAndEnvironmentToInstanceType",
        { "Ref": "AWS::Region" }, { "Ref": "EnvironmentType" } ] },
        "SecurityGroupIds" : [{ "Fn::FindInMap" :
        [ "RegionAndEnvironmentToSecurityGroup", { "Ref" : "AWS::Region" }, { "Ref" :
        "EnvironmentType" } ]}]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvironmentType:
    Description: The environment type (Dev or Prod)
    Type: String
    Default: Dev
    AllowedValues:
      - Dev
      - Prod
Mappings:
  RegionAndEnvironmentToInstanceType:
    us-east-1:
      Dev: t3.micro
      Prod: c5.large
    us-west-1:
      Dev: t2.micro
      Prod: m5.large
  RegionAndEnvironmentToSecurityGroup:
    us-east-1:
      Dev: sg-12345678

```

```
    Prod: sg-abcdef01
  us-west-1:
    Dev: sg-ghijkl23
    Prod: sg-45678abc
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      InstanceType: !FindInMap [RegionAndEnvironmentToInstanceType, !Ref
'AWS::Region', !Ref EnvironmentType]
      SecurityGroupIds:
        - !FindInMap [RegionAndEnvironmentToSecurityGroup, !Ref 'AWS::Region', !Ref
EnvironmentType]
```

相關資源

當您開發使用 `Fn::FindInMap` 函數的範本時，這些相關主題會很有幫助。

- [Fn::FindInMap](#)
- [Fn::FindInMap 增強功能](#)
- [Fn::Sub](#)

CloudFormation 範本 Metadata 語法

Metadata 會使用 JSON 或 YAML 物件儲存額外資訊。您可在範本中使用的範本層級中繼資料類型包括：

自訂中繼資料

儲存使用者定義的索引鍵/值對。例如，您可提供不影響資源建立，但能說明基礎設施、團隊或部署詳情的額外資訊。

AWS::CloudFormation::Interface

定義輸入參數在 CloudFormation 主控台中顯示的群組和排序。根據預設，CloudFormation 主控台會依據參數的邏輯 ID 字母順序來排序參數。

AWS::CloudFormation::Designer

CloudFormation 設計工具 (Designer) 已於 2025 年 2 月 5 日結束生命週期。

Important

在更新堆疊期間，您無法自行更新 Metadata 區段。只有在包含可新增、修改或刪除資源的變更時，才能予以更新。

CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。因此，我們強烈建議您不要使用此區段來儲存機密資訊，例如密碼或秘密。

語法

若要在您的 CloudFormation 範本中宣告自訂中繼資料，請使用下列語法：

JSON

```
"Metadata" : {  
  "Instances" : {"Description" : "Information about the instances"},  
  "Databases" : {"Description" : "Information about the databases"}  
}
```

YAML

```
Metadata:  
  Instances:  
    Description: "Information about the instances"  
  Databases:  
    Description: "Information about the databases"
```

如需 `AWS::CloudFormation::Interface` 的語法，請參閱 [使用 AWS::CloudFormation::Interface 中繼資料組織 CloudFormation 參數](#)。

使用 `AWS::CloudFormation::Interface` 中繼資料組織 CloudFormation 參數

`AWS::CloudFormation::Interface` 是一種中繼資料金鑰，可定義參數在 CloudFormation 主控台的分組和排序方式。預設情況下，當您在主控台中建立或更新堆疊時，主控台會依據輸入參數的邏輯 ID 字母順序來列出參數。使用此金鑰時，您可以定義自己的參數分組和排序，以讓使用者有效地指定參數值。例如，您可以將所有 EC2 相關參數分組為一個群組，而將所有 VPC 相關參數分組為另一個群組。

在中繼資料金鑰中，您可以指定要建立的群組、每個群組中所要包含的參數，以及主控台顯示群組內每個參數的順序。

也可以定義參數的標籤。標籤是要在主控台中顯示的易讀名稱或描述，而非參數的邏輯 ID。標籤有助使用者了解可為每個參數指定的值。例如，您可以為 KeyPair 參數設定 Select an EC2 key pair 標籤。

您在中繼資料金鑰中參考的所有參數都會宣告於範本的 Parameters 區段中。

Note

只有 CloudFormation 主控台會使用 `AWS::CloudFormation::Interface` 中繼資料金鑰。AWS CLI 而 API 呼叫不會使用此金鑰。

語法

若要在您的 CloudFormation 範本中宣告此實體，請使用下列語法：

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups": [
      {
        "Label": {
          "default": "Group Label"
        },
        "Parameters": [
          "Parameter1",
          "Parameter2"
        ]
      }
    ],
    "ParameterLabels": {
      "Parameter1": {
        "default": "Friendly Name for Parameter1"
      }
    }
  }
}
```

YAML

```
Metadata:
```

```
AWS::CloudFormation::Interface:
  ParameterGroups:
    - Label:
        default: Group Label
      Parameters:
        - Parameter1
        - Parameter2
  ParameterLabels:
    Parameter1:
      default: Friendly Name for Parameter1
```

Properties

ParameterGroups

參數群組類型清單；您可以在其中指定群組名稱、每個群組中的參數，以及參數的顯示順序。

必要：否

Label

參數群組的名稱。

必要：否

default

CloudFormation 主控台用來命名參數群組的預設標籤。

必要：否

類型：字串

Parameters

群組要包含之區分大小寫的參數邏輯 ID 清單。這些參數必須已在範本的 Parameters 區段內定義。一個參數只能包含在一個參數群組中。

您不會在 Other parameters 群組中，按字母順序建立主控台所列出參數與參數群組的關聯。

必要：否

類型：字串值清單

ParameterLabels

參數和其易讀名稱的映射；在建立或更新堆疊時，CloudFormation 主控台會顯示這些項目。

必要：否

參數標籤

參數的標籤。此標籤定義易用的名稱或描述，建立或更新堆疊時，CloudFormation 主控台會在 Specify Parameters (指定參數) 頁面上顯示這些名稱或描述。參數標籤必須是有效參數的邏輯 ID (區分大小寫)，而且這些參數必須已在範本的 Parameters 區段中宣告。

必要：否

default

CloudFormation 主控台用來命名參數的預設標籤。

必要：否

類型：字串

範例

下列範例定義兩個參數群組：Network Configuration 和 Amazon EC2 Configuration。Network Configuration 群組包含 VPCID、SubnetId 和 SecurityGroupID 參數，這些參數是在範本的 Parameters 區段中定義 (未顯示)。在主控台中，這些參數的顯示順序是由參數的列出順序來定義，從 VPCID 參數開始。本範例會以類似方式來分組和排序 Amazon EC2 Configuration 參數。

本範例也會定義 VPCID 參數的標籤。主控台會顯示 Which VPC should this be deployed to? (此項目應該部署至哪個 VPC?) 而非參數的邏輯 ID (VPCID)。

JSON

```
"Metadata" : {
  "AWS::CloudFormation::Interface" : {
    "ParameterGroups" : [
      {
        "Label" : { "default" : "Network Configuration" },
        "Parameters" : [ "VPCID", "SubnetId", "SecurityGroupID" ]
      }
    ]
  }
}
```

```
    },
    {
      "Label" : { "default":"Amazon EC2 Configuration" },
      "Parameters" : [ "InstanceType", "KeyName" ]
    }
  ],
  "ParameterLabels" : {
    "VPCID" : { "default" : "Which VPC should this be deployed to?" }
  }
}
```

YAML

```
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Network Configuration"
        Parameters:
          - VPCID
          - SubnetId
          - SecurityGroupId
      - Label:
          default: "Amazon EC2 Configuration"
        Parameters:
          - InstanceType
          - KeyName
    ParameterLabels:
      VPCID:
        default: "Which VPC should this be deployed to?"
```

主控台中的參數群組

下圖會使用此範例的中繼資料金鑰來說明建立或更新堆疊時，主控台會如何顯示參數群組：主控台內的參數群組

CloudFormation 範本 Rules 語法

Rules 區段為 CloudFormation 範本的選用部分，可啟用自訂驗證邏輯。若包含此區段，其中的規則函式會在 CloudFormation 建立或更新任何資源前驗證參數值。

當標準參數限制不足時，規則很有用。例如，啟用 SSL 時，必須提供憑證和網域名稱。規則可以確保滿足這些相依性。

語法

Rules 區段使用下列語法：

JSON

範本的 Rules 區塊包含了金鑰名稱 Rules，後面接著單一冒號。您必須使用大括號來括住所有規則宣告。如果宣告多項規則，這些規則會以逗號分隔。對於每項規則，您會在引號中宣告其邏輯名稱，後面依序接著冒號和括號，括號之中是規則條件與宣告。

```
{
  "Rules": {
    "LogicalRuleName1": {
      "RuleCondition": {
        "rule-specific intrinsic function": "Value"
      }
    }
  }
}
```

```

    },
    "Assertions": [
      {
        "Assert": {
          "rule-specific intrinsic function": "Value"
        },
        "AssertDescription": "Information about this assert"
      },
      {
        "Assert": {
          "rule-specific intrinsic function": "Value"
        },
        "AssertDescription": "Information about this assert"
      }
    ]
  },
  "LogicalRuleName2": {
    "Assertions": [
      {
        "Assert": {
          "rule-specific intrinsic function": "Value"
        },
        "AssertDescription": "Information about this assert"
      }
    ]
  }
}

```

YAML

```

Rules:
  LogicalRuleName1:
    RuleCondition:
      rule-specific intrinsic function: Value
    Assertions:
      - Assert:
          rule-specific intrinsic function: Value
          AssertDescription: Information about this assert
      - Assert:
          rule-specific intrinsic function: Value
          AssertDescription: Information about this assert
  LogicalRuleName2:

```

```
Assertions:
  - Assert:
      rule-specific intrinsic function: Value
      AssertDescription: Information about this assert
```

規則欄位

Rules 區段可以包含下列欄位。

邏輯 ID (也稱為邏輯名稱)

每條規則的唯一識別碼。

RuleCondition (選用)

決定規則何時生效的屬性。如果您未定義規則條件，則該規則的宣告一律生效。對於每個規則，您只能定義一個規則條件。

Assertions (必要)

一個或多個陳述式，指定特定參數的可接受值。

Assert

必須等於 true 的條件。

AssertDescription

宣告失敗時顯示的訊息。

規則特定的內部函數

若要定義您的規則，您必須使用規則特定的函數，這些函式只能在範本的 Rules 區塊中使用。雖然可以巢狀這些函數，但規則條件或宣告的最終結果必須為 true 或 false。

現在可以使用下列規則：

- [Fn::And](#)
- [Fn::Contains](#)
- [Fn::EachMemberEquals](#)
- [Fn::EachMemberIn](#)
- [Fn::Equals](#)

- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)
- [Fn::RefAll](#)
- [Fn::ValueOf](#)
- [Fn::ValueOfAll](#)

這些函數用於規則的條件或宣告中。由條件屬性決定 CloudFormation 是否套用宣告。若條件計算結果為 true，CloudFormation 會評估宣告以驗證在佈建產品建立或更新時，某個參數值是否有效。若參數值為無效，CloudFormation 便不會建立或更新堆疊。如果條件評估為 false，CloudFormation 便不會檢查參數值，而直接繼續運作堆疊。

範例

主題

- [有條件地驗證參數值](#)
- [交互參數驗證](#)

有條件地驗證參數值

在下列範例中，兩個規則會檢查 InstanceType 參數的值。視環境參數 (test 或 prod) 的值而定，使用者必須針對 t3.medium 參數指定 t3.large 或 InstanceType。InstanceType 與 Environment 參數必須在同一個範本的 Parameters 區塊中宣告。

JSON

```
{
  "Rules": {
    "testInstanceType": {
      "RuleCondition": {
        "Fn::Equals": [
          {"Ref": "Environment"},
          "test"
        ]
      },
      "Assertions": [
        {
          "Assert": {
```



```

        "Fn::Contains": [
            ["t3.medium"],
            {"Ref": "InstanceType"}
        ]
    },
    "AssertDescription": "For a test environment, the instance type must be
t3.medium"
}
]
},
"prodInstanceType": {
    "RuleCondition": {
        "Fn::Equals": [
            {"Ref": "Environment"},
            "prod"
        ]
    },
    "Assertions": [
        {
            "Assert": {
                "Fn::Contains": [
                    ["t3.large"],
                    {"Ref": "InstanceType"}
                ]
            },
            "AssertDescription": "For a production environment, the instance type must be
t3.large"
        }
    ]
}
}
}
}

```

YAML

```

Rules:
  testInstanceType:
    RuleCondition: !Equals
      - !Ref Environment
      - test
    Assertions:
      - Assert:
          'Fn::Contains':

```

```

        - - t3.medium
        - !Ref InstanceType
        AssertDescription: 'For a test environment, the instance type must be
t3.medium'
    prodInstanceType:
        RuleCondition: !Equals
        - !Ref Environment
        - prod
        Assertions:
        - Assert:
            'Fn::Contains':
            - - t3.large
            - !Ref InstanceType
            AssertDescription: 'For a production environment, the instance type must be
t3.large'

```

交互參數驗證

下列範例範本示範如何使用規則進行跨參數驗證。這些範本會建立一個執行在負載平衡器後方 Auto Scaling 群組上的範例網站。根據輸入參數，網站可透過連接埠 80 或 443 存取。Auto Scaling 群組中的執行個體可設定為接聽任何連接埠 (預設為 8888)。

此範本中的規則會在建立堆疊前驗證輸入參數。其會驗證所有子網路是否屬於指定的 VPC，並確保當 UseSSL 參數設定為 Yes 時，同時提供 SSL 憑證 ARN 和託管區域名稱。

Note

如果您從此範本建立堆疊，您將需要支付所使用的 AWS 資源費用。

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "VpcId": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
      "ConstraintDescription": "must be the VPC Id of an existing Virtual Private
Cloud."
    },
    "Subnets": {

```

```

    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
    "ConstraintDescription": "must be a list of at least two existing subnets
associated with at least two different availability zones."
  },
  "InstanceType": {
    "Description": "WebServer EC2 instance type",
    "Type": "String",
    "Default": "t2.micro",
    "AllowedValues": ["t2.micro", "t3.micro"],
    "ConstraintDescription": "must be a valid EC2 instance type."
  },
  "KeyName": {
    "Description": "Name of an existing EC2 KeyPair to enable SSH access to the
instances",
    "Type": "AWS::EC2::KeyPair::KeyName",
    "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
  },
  "SSHLocation": {
    "Description": "The IP address range that can be used to SSH to the EC2
instances",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "UseSSL": {
    "AllowedValues": ["Yes", "No"],
    "Default": "No",
    "Description": "Select \"Yes\" to implement SSL, \"No\" to skip (default).",
    "Type": "String"
  },
  "ALBSSLCertificateARN": {
    "Default": "",
    "Description": "[Optional] The ARN of the SSL certificate to be used for the
Application Load Balancer",
    "Type": "String"
  },
  "HostedZoneName": {
    "AllowedPattern": "^$|((([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\\-\\-]*[a-zA-Z0-9])\\
\\.)*([A-Za-z0-9]|[A-Za-z0-9][A-Za-z0-9\\-\\-]*[A-Za-z0-9]))$",
    "Default": "",

```

```

        "Description": "[Optional] The domain name of a valid Hosted Zone on AWS.",
        "Type": "String"
    },
    "Conditions": {
        "UseALBSSL": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]}
    },
    "Rules": {
        "SubnetsInVPC": {
            "Assertions": [
                {
                    "Assert": {"Fn::EachMemberEquals": [{"Fn::ValueOf": ["Subnets", "VpcId"]}, {"Ref": "VpcId"}]},
                    "AssertDescription": "All subnets must be in the VPC"
                }
            ]
        },
        "ValidateHostedZone": {
            "RuleCondition": {"Fn::Equals": [{"Ref": "UseSSL"}, "Yes"]},
            "Assertions": [
                {
                    "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "ALBSSLCertificateARN"}, ""]}]},
                    "AssertDescription": "ACM Certificate value cannot be empty if SSL is required"
                },
                {
                    "Assert": {"Fn::Not": [{"Fn::Equals": [{"Ref": "HostedZoneName"}, ""]}]},
                    "AssertDescription": "Route53 Hosted Zone Name is mandatory when SSL is required"
                }
            ]
        }
    },
    "Resources": {
        "WebServerGroup": {
            "Type": "AWS::AutoScaling::AutoScalingGroup",
            "Properties": {
                "VPCZoneIdentifier": {"Ref": "Subnets"},
                "LaunchTemplate": {
                    "LaunchTemplateId": {"Ref": "LaunchTemplate"},
                    "Version": {"Fn::GetAtt": ["LaunchTemplate", "LatestVersionNumber"]}
                },
                "MinSize": "2",

```

```

    "MaxSize": "2",
    "TargetGroupARNs": [{"Ref": "ALBTargetGroup"}]
  },
  "CreationPolicy": {
    "ResourceSignal": {"Timeout": "PT15M"}
  },
  "UpdatePolicy": {
    "AutoScalingRollingUpdate": {
      "MinInstancesInService": "1",
      "MaxBatchSize": "1",
      "PauseTime": "PT15M",
      "WaitOnResourceSignals": true
    }
  }
},
"LaunchTemplate": {
  "Type": "AWS::EC2::LaunchTemplate",
  "Metadata": {
    "Comment": "Install a simple application",
    "AWS::CloudFormation::Init": {
      "config": {
        "packages": {"yum": {"httpd": []}},
        "files": {
          "/var/www/html/index.html": {
            "content": {"Fn::Join": ["\n", ["<h1>Congratulations, you have
successfully launched the AWS CloudFormation sample.</h1>"]]},
            "mode": "000644",
            "owner": "root",
            "group": "root"
          },
          "/etc/cfn/cfn-hup.conf": {
            "content": {"Fn::Join": ["", [
              "[main]\n",
              "stack=", {"Ref": "AWS::StackId"}, "\n",
              "region=", {"Ref": "AWS::Region"}, "\n"
            ]]},
            "mode": "000400",
            "owner": "root",
            "group": "root"
          },
          "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
            "content": {"Fn::Join": ["", [
              "[cfn-auto-reloader-hook]\n",
              "triggers=post.update\n",

```

```

        "path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init\n",
        "action=/opt/aws/bin/cfn-init -v ",
        "        --stack ", {"Ref": "AWS::StackName"},
        "        --resource LaunchTemplate ",
        "        --region ", {"Ref": "AWS::Region"}, "\n",
        "runas=root\n"
    ]],
    "mode": "000400",
    "owner": "root",
    "group": "root"
}
},
"services": {
    "sysvinit": {
        "httpd": {
            "enabled": "true",
            "ensureRunning": "true"
        },
        "cfn-hup": {
            "enabled": "true",
            "ensureRunning": "true",
            "files": [
                "/etc/cfn/cfn-hup.conf",
                "/etc/cfn/hooks.d/cfn-auto-reloader.conf"
            ]
        }
    }
}
},
"Properties": {
    "LaunchTemplateName": {"Fn::Sub": "${AWS::StackName}-launch-template"},
    "LaunchTemplateData": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
        "SecurityGroupIds": [{"Ref": "InstanceSecurityGroup"}],
        "InstanceType": {"Ref": "InstanceType"},
        "KeyName": {"Ref": "KeyName"},
        "UserData": {
            "Fn::Base64": {"Fn::Join": ["", [
                "#!/bin/bash\n",
                "yum install -y aws-cfn-bootstrap\n",
                "/opt/aws/bin/cfn-init -v ",

```

```

        "        --stack ", {"Ref": "AWS::StackName"},
        "        --resource LaunchTemplate ",
        "        --region ", {"Ref": "AWS::Region"}, "\n",
        "/opt/aws/bin/cfn-signal -e $? ",
        "        --stack ", {"Ref": "AWS::StackName"},
        "        --resource WebServerGroup ",
        "        --region ", {"Ref": "AWS::Region"}, "\n"
    ]}]
    }
}
},
"ELBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Allow access to the ELB",
        "VpcId": {"Ref": "VpcId"},
        "SecurityGroupIngress": [{
            "Fn::If": [
                "UseALBSSL",
                {
                    "IpProtocol": "tcp",
                    "FromPort": 443,
                    "ToPort": 443,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "CidrIp": "0.0.0.0/0"
                }
            ]
        }]
    }
},
"ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
        "Subnets": {"Ref": "Subnets"},
        "SecurityGroups": [{"Ref": "ELBSecurityGroup"}]
    }
},
"ALBListener": {

```

```

    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "DefaultActions": [{
        "Type": "forward",
        "TargetGroupArn": {"Ref": "ALBTargetGroup"}
      }],
      "LoadBalancerArn": {"Ref": "ApplicationLoadBalancer"},
      "Port": {"Fn::If": ["UseALBSSL", 443, 80]},
      "Protocol": {"Fn::If": ["UseALBSSL", "HTTPS", "HTTP"]},
      "Certificates": [{
        "Fn::If": [
          "UseALBSSL",
          {"CertificateArn": {"Ref": "ALBSSLCertificateARN"}},
          {"Ref": "AWS::NoValue"}
        ]
      }
    ]
  },
  "ALBTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "HealthCheckIntervalSeconds": 30,
      "HealthCheckTimeoutSeconds": 5,
      "HealthyThresholdCount": 3,
      "Port": 80,
      "Protocol": "HTTP",
      "UnhealthyThresholdCount": 5,
      "VpcId": {"Ref": "VpcId"}
    }
  },
  "InstanceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Enable SSH access and HTTP access on the inbound port",
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "SourceSecurityGroupId": {"Fn::Select": [0, {"Fn::GetAtt":
["ApplicationLoadBalancer", "SecurityGroups"]}]}
        },
        {
          "IpProtocol": "tcp",

```



```

        "FromPort": 22,
        "ToPort": 22,
        "CidrIp": {"Ref": "SSHLocation"}
    }
],
    "VpcId": {"Ref": "VpcId"}
}
},
"RecordSet": {
    "Type": "AWS::Route53::RecordSetGroup",
    "Condition": "UseALBSSL",
    "Properties": {
        "HostedZoneName": {"Fn::Join": [ "", [ {"Ref": "HostedZoneName"}, "." ] ] },
        "RecordSets": [ {
            "Name": {"Fn::Join": [ "", [
                {"Fn::Select": [ "0", {"Fn::Split": [ ".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"] } ] ] ] },
                ".",
                {"Ref": "HostedZoneName"},
                "."
            ] ] },
            "Type": "A",
            "AliasTarget": {
                "DNSName": {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"] },
                "EvaluateTargetHealth": true,
                "HostedZoneId": {"Fn::GetAtt": ["ApplicationLoadBalancer",
"CanonicalHostedZoneID"] }
            }
        } ]
    }
},
"Outputs": {
    "URL": {
        "Description": "URL of the website",
        "Value": {"Fn::Join": [ "", [
            {"Fn::If": [
                "UseALBSSL",
                {"Fn::Join": [ "", [
                    "https://",
                    {"Fn::Join": [ "", [
                        {"Fn::Select": [ "0", {"Fn::Split": [ ".", {"Fn::GetAtt":
["ApplicationLoadBalancer", "DNSName"] } ] ] ] },
                        ".",

```

```

        {"Ref": "HostedZoneName"},
        "."
    ]}]
  ]}],
  {"Fn::Join": [ "", [
    "http://",
    {"Fn::GetAtt": ["ApplicationLoadBalancer", "DNSName"]}
  ]}]
  ]}
  ]}]
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VpcId of your existing Virtual Private Cloud (VPC)
    ConstraintDescription: must be the VPC Id of an existing Virtual Private Cloud.
  Subnets:
    Type: List<AWS::EC2::Subnet::Id>
    Description: The list of SubnetIds in your Virtual Private Cloud (VPC)
    ConstraintDescription: >-
      must be a list of at least two existing subnets associated with at least
      two different availability zones. They should be residing in the selected
      Virtual Private Cloud.
  InstanceType:
    Description: WebServer EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
    ConstraintDescription: must be a valid EC2 instance type.
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  SSHLocation:
    Description: The IP address range that can be used to SSH to the EC2 instances

```

```

    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  UseSSL:
    AllowedValues:
      - 'Yes'
      - 'No'
    ConstraintDescription: Select Yes to create a HTTPS Listener
    Default: 'No'
    Description: 'Select "Yes" to implement SSL, "No" to skip (default).'
    Type: String
  ALBSSLCertificateARN:
    Default: ''
    Description: >-
      [Optional] The ARN of the SSL certificate to be used for the Application
      Load Balancer
    Type: String
  HostedZoneName:
    AllowedPattern: >-
      ^$|(([a-zA-Z0-9]|[a-zA-Z0-9][a-zA-Z0-9\-\-]*[a-zA-Z0-9])\.)*([A-Za-z0-9]|[A-Za-
      z0-9][A-Za-z0-9\-\-]*[A-Za-z0-9])$
    Default: ''
    Description: '[Optional] The domain name of a valid Hosted Zone on AWS.'
    Type: String
  Conditions:
    UseALBSSL: !Equals
      - !Ref UseSSL
      - 'Yes'
  Rules:
    SubnetsInVPC:
      Assertions:
        - Assert:
            'Fn::EachMemberEquals':
              - 'Fn::ValueOf':
                  - Subnets
                  - VpcId
              - Ref: VpcId
            AssertDescription: All subnets must be in the VPC
    ValidateHostedZone:
      RuleCondition: !Equals
        - !Ref UseSSL

```

```

    - 'Yes'
  Assertions:
    - Assert: !Not
      - !Equals
        - !Ref ALBSSLCertificateARN
        - ''
      AssertDescription: ACM Certificate value cannot be empty if SSL is required
    - Assert: !Not
      - !Equals
        - !Ref HostedZoneName
        - ''
      AssertDescription: Route53 Hosted Zone Name is mandatory when SSL is required
  Resources:
    WebServerGroup:
      Type: AWS::AutoScaling::AutoScalingGroup
      Properties:
        VPCZoneIdentifier: !Ref Subnets
        LaunchTemplate:
          LaunchTemplateId: !Ref LaunchTemplate
          Version: !GetAtt LaunchTemplate.LatestVersionNumber
        MinSize: '2'
        MaxSize: '2'
        TargetGroupARNs:
          - !Ref ALBTargetGroup
      CreationPolicy:
        ResourceSignal:
          Timeout: PT15M
      UpdatePolicy:
        AutoScalingRollingUpdate:
          MinInstancesInService: '1'
          MaxBatchSize: '1'
          PauseTime: PT15M
          WaitOnResourceSignals: 'true'
    LaunchTemplate:
      Type: AWS::EC2::LaunchTemplate
      Metadata:
        Comment: Install a simple application
        AWS::CloudFormation::Init:
          config:
            packages:
              yum:
                httpd: []
            files:
              /var/www/html/index.html:

```

```

        content: !Join
          - |+
          - - >-
            <h1>Congratulations, you have successfully launched the AWS
            CloudFormation sample.</h1>
        mode: '000644'
        owner: root
        group: root
    /etc/cfn/cfn-hup.conf:
        content: !Sub |
          [main]
          stack=${AWS::StackId}
          region=${AWS::Region}
        mode: '000400'
        owner: root
        group: root
    /etc/cfn/hooks.d/cfn-auto-reloader.conf:
        content: !Sub |-
          [cfn-auto-reloader-hook]
          triggers=post.update
          path=Resources.LaunchTemplate.Metadata.AWS::CloudFormation::Init
          action=/opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
          runas=root
        mode: '000400'
        owner: root
        group: root
    services:
      sysvinit:
        httpd:
          enabled: 'true'
          ensureRunning: 'true'
        cfn-hup:
          enabled: 'true'
          ensureRunning: 'true'
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf

    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
      SecurityGroupIds:

```

```

- !Ref InstanceSecurityGroup
InstanceType: !Ref InstanceType
KeyName: !Ref KeyName
UserData: !Base64
  Fn::Sub: |
    #!/bin/bash
    yum install -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchTemplate --region ${AWS::Region}
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerGroup --region ${AWS::Region}
ELBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow access to the ELB
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - !If
        - UseALBSSL
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets: !Ref Subnets
    SecurityGroups:
      - !Ref ELBSecurityGroup
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: !If
      - UseALBSSL
      - 443
      - 80

```

```

    Protocol: !If
      - UseALBSSL
      - HTTPS
      - HTTP
    Certificates:
      - !If
        - UseALBSSL
        - CertificateArn: !Ref ALBSSLCertificateARN
        - !Ref 'AWS::NoValue'
  ALBTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      HealthCheckIntervalSeconds: 30
      HealthCheckTimeoutSeconds: 5
      HealthyThresholdCount: 3
      Port: 80
      Protocol: HTTP
      UnhealthyThresholdCount: 5
      VpcId: !Ref VpcId
  InstanceSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable SSH access and HTTP access on the inbound port
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          SourceSecurityGroupId: !Select
            - 0
            - !GetAtt
              - ApplicationLoadBalancer
              - SecurityGroups
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref SSHLocation
      VpcId: !Ref VpcId
  RecordSet:
    Type: AWS::Route53::RecordSetGroup
    Condition: UseALBSSL
    Properties:
      HostedZoneName: !Join
        - ''
        - - !Ref HostedZoneName

```

```

    - .
RecordSets:
  - Name: !Join
    - ''
    - - !Select
      - '0'
      - !Split
        - .
        - !GetAtt
          - ApplicationLoadBalancer
          - DNSName
    - .
    - !Ref HostedZoneName
    - .
Type: A
AliasTarget:
  DNSName: !GetAtt
    - ApplicationLoadBalancer
    - DNSName
  EvaluateTargetHealth: true
  HostedZoneId: !GetAtt
    - ApplicationLoadBalancer
    - CanonicalHostedZoneID
Outputs:
  URL:
    Description: URL of the website
    Value: !Join
      - ''
      - - !If
        - UseALBSSL
        - !Join
          - ''
          - - 'https://'
            - !Join
              - ''
              - - !Select
                - '0'
                - !Split
                  - .
                  - !GetAtt
                    - ApplicationLoadBalancer
                    - DNSName
              - .
            - !Ref HostedZoneName

```



```

- !Join
- ''
- - 'http://'
- !GetAtt
- ApplicationLoadBalancer
- DNSName

```

CloudFormation 範本 Conditions 語法

選用的 Conditions 區段包含陳述式來定義在哪些情況下建立或設定實體。例如，您可以建立條件，並將它與資源或輸出建立關聯，使得 CloudFormation 只在條件為 true 時才建立資源或輸出。同樣地，您可以將條件與屬性建立關聯，使得 CloudFormation 只在條件為 true 時才將屬性設為特定的值。如果條件為 false，CloudFormation 會將屬性設為您指定的替代值。

當您想要重複使用範本以在不同的內容 (例如測試環境與生產環境) 中建立資源時，可以使用條件。例如，可以在範本中新增 EnvironmentType 輸入參數，此參數接受 prod 或 test 作為輸入。在 prod 環境中，可以包含具有特定功能的 EC2 執行個體；而在 test 環境中，就要使用精簡的功能來節省資金。此條件定義可讓您針對每種環境類型來定義要建立哪些資源和其設定方式。

語法

Conditions 區段由索引鍵名稱 Conditions 組成。每個條件宣告都包含邏輯 ID 和一個或多個內建函數。

JSON

```

"Conditions": {
  "LogicalConditionName1": {
    "Intrinsic function": ...[
  },

  "LogicalConditionName2": {
    "Intrinsic function": ...
  }
}

```

YAML

```

Conditions:
  LogicalConditionName1:
    Intrinsic function:

```

```
...  
  
LogicalConditionName2:  
  Intrinsic function:  
  ...
```

條件的運作方式

若要使用條件，請遵循下列步驟：

1. 新增參數定義 – 定義您的條件將在範本的 Parameters 區段中評估的輸入。條件會根據這些輸入參數值而評估為 true 或 false。請注意，虛擬參數會自動提供，而且不需要 Parameters 區段中的明確定義。如需這些虛擬參數的詳細資訊，請參閱[使用虛擬參數取得 AWS 值](#)。
2. 新增條件定義 – 使用 Fn::If 或 Fn::Equals 等內建函數，在 Conditions 區段中定義條件。這些條件決定 CloudFormation 何時建立相關聯的資源。這些條件的依據為：
 - 輸入或虛擬參數值
 - 其他條件
 - 映射值

不過，您無法在條件中參考資源邏輯 ID 或其屬性。

3. 將條件與資源或輸出建立關聯 – 使用 Condition 金鑰和條件的邏輯 ID 來參考資源或輸出中的條件。或者，在範本的其他部分 (例如屬性值) 中使用 Fn::If，根據條件來設定值。如需詳細資訊，請參閱[使用 Condition 金鑰](#)。

CloudFormation 會在建立或更新堆疊時評估條件。CloudFormation 會建立與 true 條件建立關聯的實體，並忽略與 false 條件建立關聯的實體。在修改任何資源之前，CloudFormation 也會在每次堆疊更新期間重新評估這些條件。會更新與 true 條件保持關聯的實體，而與 false 條件相關聯的實體則會被刪除。

Important

在更新堆疊期間，您無法自行更新條件。唯有涵蓋新增、修改或刪除資源等變更作業時，才能更新條件。

條件內部函數

您可以使用下列內部函數來定義條件：

- [Fn::And](#)
- [Fn::Equals](#)
- [Fn::ForEach](#)
- [Fn::If](#)
- [Fn::Not](#)
- [Fn::Or](#)

Note

只有在範本的 Resources 區段和 Outputs 區段的中繼資料屬性、更新政策屬性和屬性值中，才支援 Fn::If。

使用 **Condition** 金鑰

定義條件後，可以使用 Condition 金鑰將其套用至範本中的多個位置，例如 Resources 和 Outputs。Condition 金鑰會參考條件的邏輯名稱，並傳回指定條件的評估結果。

主題

- [將條件與資源建立關聯](#)
- [將條件與輸出建立關聯](#)
- [其他條件中的參考條件](#)
- [使用 Fn::If 有條件地傳回屬性值](#)

將條件與資源建立關聯

若要有條件地建立資源，請將 Condition 金鑰和條件的邏輯 ID 新增為資源的屬性。只有當條件評估為 true 時，CloudFormation 才會建立資源。

JSON

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Condition" : "IsProduction",
  "Properties" : {
    "Size" : "100",
```

```
    "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone" ] }  
  }  
}
```

YAML

```
NewVolume:  
  Type: AWS::EC2::Volume  
  Condition: IsProduction  
  Properties:  
    Size: 100  
    AvailabilityZone: !GetAtt EC2Instance.AvailabilityZone
```

將條件與輸出建立關聯

也可以將條件與輸出建立關聯。CloudFormation 只會在關聯條件評估為 true 時建立輸出。

JSON

```
"Outputs" : {  
  "VolumeId" : {  
    "Condition" : "IsProduction",  
    "Value" : { "Ref" : "NewVolume" }  
  }  
}
```

YAML

```
Outputs:  
  VolumeId:  
    Condition: IsProduction  
    Value: !Ref NewVolume
```

其他條件中的參考條件

在 Conditions 區段中定義條件時，可以使用 Condition 金鑰來參考其他條件。這可讓您結合多個條件來建立更複雜的條件式邏輯。

在下列範例中，只有在 IsProduction 和 IsFeatureEnabled 條件評估為 true 時，IsProdAndFeatureEnabled 條件才會評估為 true。

JSON

```
"Conditions": {
  "IsProduction" : {"Fn::Equals" : [{"Ref" : "Environment"}, "prod"]},
  "IsFeatureEnabled" : { "Fn::Equals" : [{"Ref" : "FeatureFlag"}, "enabled"]},
  "IsProdAndFeatureEnabled" : {
    "Fn::And" : [
      {"Condition" : "IsProduction"},
      {"Condition" : "IsFeatureEnabled"}
    ]
  }
}
```

YAML

```
Conditions:
  IsProduction:
    !Equals [!Ref Environment, "prod"]
  IsFeatureEnabled:
    !Equals [!Ref FeatureFlag, "enabled"]
  IsProdAndFeatureEnabled: !And
    - !Condition IsProduction
    - !Condition IsFeatureEnabled
```

使用 **Fn::If** 有條件地傳回屬性值

如需更精細的控制，可以使用 `Fn::If` 內建函數，在資源或輸出中有條件地傳回兩個屬性值的其中之一。此函數會評估條件，如果條件為 `true`，則傳回一個值，如果條件為 `false`，則傳回另一個值。

條件式屬性值

下列範例示範如何根據環境條件來設定 EC2 執行個體類型。如果 `IsProduction` 條件評估為 `true`，執行個體類型會設為 `c5.xlarge`。否則，它會設定為 `t3.small`。

JSON

```
"Properties" : {
  "InstanceType" : {
    "Fn::If" : [
      "IsProduction",
      "c5.xlarge",
```

```
        "t3.small"
    ]
}
}
```

YAML

```
Properties:
  InstanceType: !If
    - IsProduction
    - c5.xlarge
    - t3.small
```

條件式屬性移除

也可以使用 `AWS::NoValue` 虛擬參數作為傳回值來移除條件為 `false` 時的對應屬性。

JSON

```
"DBSnapshotIdentifier" : {
  "Fn::If" : [
    "UseDBSnapshot",
    {"Ref" : "DBSnapshotName"},
    {"Ref" : "AWS::NoValue"}
  ]
}
```

YAML

```
DBSnapshotIdentifier: !If
  - UseDBSnapshot
  - !Ref DBSnapshotName
  - !Ref "AWS::NoValue"
```

範例

主題

- [基於環境的資源建立](#)
- [多條件資源佈建](#)

基於環境的資源建立

以下範例會佈建 EC2 執行個體，並且只有在環境類型為 `prod` 時，才会有條件地建立新的 EBS 磁碟區。如果環境為 `test`，則其只會建立沒有額外磁碟區的 EC2 執行個體。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Description": "Environment type",
      "Default": "test",
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ],
      "ConstraintDescription": "must specify prod or test"
    }
  },
  "Conditions": {
    "IsProduction": {
      "Fn::Equals": [
        {
          "Ref": "EnvType"
        },
        "prod"
      ]
    }
  },
  "Resources": {
    "EC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-1234567890abcdef0",
        "InstanceType": "c5.xlarge"
      }
    },
    "MountPoint": {
      "Type": "AWS::EC2::VolumeAttachment",
      "Condition": "IsProduction",
      "Properties": {
        "InstanceId": {
```

```

        "Ref": "EC2Instance"
      },
      "VolumeId": {
        "Ref": "NewVolume"
      },
      "Device": "/dev/sdh"
    }
  },
  "NewVolume": {
    "Type": "AWS::EC2::Volume",
    "Condition": "IsProduction",
    "Properties": {
      "Size": 100,
      "AvailabilityZone": {
        "Fn::GetAtt": [
          "EC2Instance",
          "AvailabilityZone"
        ]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Description: Environment type
    Default: test
    Type: String
    AllowedValues:
      - prod
      - test
    ConstraintDescription: must specify prod or test
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance

```



```
Properties:
  ImageId: ami-1234567890abcdef0
  InstanceType: c5.xlarge
MountPoint:
  Type: AWS::EC2::VolumeAttachment
  Condition: IsProduction
  Properties:
    InstanceId: !Ref EC2Instance
    VolumeId: !Ref NewVolume
    Device: /dev/sdh
NewVolume:
  Type: AWS::EC2::Volume
  Condition: IsProduction
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt
      - EC2Instance
      - AvailabilityZone
```

多條件資源佈建

以下範例會在提供儲存貯體名稱時有條件地建立 S3 儲存貯體，並只在環境設定為 prod 時連接儲存貯體政策。如果未指定儲存貯體名稱或環境為 test，則不會建立資源。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EnvType": {
      "Type": "String",
      "AllowedValues": [
        "prod",
        "test"
      ]
    },
    "BucketName": {
      "Default": "",
      "Type": "String"
    }
  },
  "Conditions": {
    "IsProduction": {
```

```

        "Fn::Equals": [
            {
                "Ref": "EnvType"
            },
            "prod"
        ]
    },
    "CreateBucket": {
        "Fn::Not": [
            {
                "Fn::Equals": [
                    {
                        "Ref": "BucketName"
                    },
                    ""
                ]
            }
        ]
    },
    "CreateBucketPolicy": {
        "Fn::And": [
            {
                "Condition": "IsProduction"
            },
            {
                "Condition": "CreateBucket"
            }
        ]
    }
},
"Resources": {
    "Bucket": {
        "Type": "AWS::S3::Bucket",
        "Condition": "CreateBucket",
        "Properties": {
            "BucketName": {
                "Ref": "BucketName"
            }
        }
    }
},
"Policy": {
    "Type": "AWS::S3::BucketPolicy",
    "Condition": "CreateBucketPolicy",
    "Properties": {

```

```

        "Bucket": {
            "Ref": "Bucket"
        },
        "PolicyDocument": { ... }
    }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  EnvType:
    Type: String
    AllowedValues:
      - prod
      - test
  BucketName:
    Default: ''
    Type: String
Conditions:
  IsProduction: !Equals
    - !Ref EnvType
    - prod
  CreateBucket: !Not
    - !Equals
      - !Ref BucketName
      - ''
  CreateBucketPolicy: !And
    - !Condition IsProduction
    - !Condition CreateBucket
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Condition: CreateBucket
    Properties:
      BucketName: !Ref BucketName
  Policy:
    Type: AWS::S3::BucketPolicy
    Condition: CreateBucketPolicy
    Properties:
      Bucket: !Ref Bucket

```

PolicyDocument: ...

在此範例中，CreateBucketPolicy 條件示範如何使用 Condition 金鑰來參考其他條件。只有在 IsProduction 和 CreateBucket 條件都評估為 true 時，才會建立政策。

Note

如需使用條件的更複雜範例，請參閱《CloudFormation 範本參考指南》中的 [Condition 屬性主題](#)。

CloudFormation 範本 Transform 區段

選用的 Transform 區段指定 CloudFormation 用來以某種方式處理您範本的一或多個巨集。

巨集可執行簡單任務，如文字查詢與替換，也可對整個範本進行更全面的轉換。CloudFormation 會按照巨集的指定順序執行。當您建立變更集時，CloudFormation 產生的變更集會包含處理過的範本內容。然後，您可以檢閱變更並執行變更集。如需有關巨集運作方式的詳細資訊，請參閱 [使用範本巨集在 CloudFormation 範本上執行自訂處理](#)。

CloudFormation 亦支援轉換，這些是由 CloudFormation 託管的巨集。在執行順序和範圍方面，CloudFormation 對待這些轉換的方式與您建立的任何巨集相同。如需詳細資訊，請參閱 [轉換參考](#)。

若要宣告多個巨集，請使用清單格式，並指定一或多個巨集。

例如，在以下範例中，CloudFormation 會評估範本 MyMacro，再評估 AWS::Serverless，這兩者都可以處理整個範本，因為其都納入 Transform 區段中。

```
# Start of processable content for MyMacro and AWS::Serverless
Transform:
  - MyMacro
  - 'AWS::Serverless'
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
```

```
    Tags: [{"key": "value"}]
    CorsConfiguration: []
MyEc2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-1234567890abcdef0
# End of processable content for MyMacro and AWS::Serverless
```

CloudFormation 範本格式版本語法

AWSTemplateFormatVersion 區段 (選用) 可識別範本所遵循的範本格式版本。最新的範本格式版本是 2010-09-09，而且是目前唯一的有效值。

範本格式版本與 API 版本不同。範本格式版本可以與 API 版本分開變更。

範本格式版本宣告的值必須是常值字串。您不能使用參數或函數來指定範本格式版本。如果未指定值，則 CloudFormation 會採用最新的範本格式版本。下列程式碼片段是有效範本格式版本宣告範例：

JSON

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

YAML

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

CloudFormation 範本 Description 語法

Description 區段 (選用) 讓您能夠包含描述範本的文字字串。此區段一律必須遵循範本格式版本區段。

描述宣告值的長度必須是介於 0 與 1024 位元組之間的常值字串。您不能使用參數或函數來指定描述。下列程式碼片段是描述宣告範例：

Important

在更新堆疊期間，您無法自行更新 Description 區段。只有在包含可新增、修改或刪除資源的變更時，才能予以更新。

JSON

```
"Description" : "Here are some details about the template."
```

YAML

```
Description: > Here are some details about the template.
```

透過 Infrastructure Composer 視覺化建立範本

AWS Infrastructure Composer (先前稱為 Application Composer) 可協助您以視覺化方式編寫和設定現代應用程式 AWS。您無需撰寫程式碼，只需拖放不同的資源即可視覺化建立應用程式。

CloudFormation 主控台模式中的 Infrastructure Composer 是建議用於視覺化處理 CloudFormation 範本的工具。您可以從 CloudFormation 主控台存取的此 Infrastructure Composer 版本，是從稱為 CloudFormation 設計工具的舊工具進行改善。

透過 CloudFormation 主控台模式中的 Infrastructure Composer，您可以將各種稱為卡片的資源拖放、設定並連結至視覺化畫布。這種視覺化方式讓您無需直接操作範本，就能輕鬆設計並編輯應用程式架構。要從 [CloudFormation 主控台](#) 存取此模式，請從左側導航選單中選取 Infrastructure Composer。

如需詳細資訊，請參閱《AWS Infrastructure Composer 開發人員指南》中的[如何在 AWS Infrastructure Composer 中編寫](#)。

為何使用 CloudFormation 主控台模式中的 Infrastructure Composer？

在 Infrastructure Composer 中視覺化呈現範本，可協助您找出 CloudFormation 範本與應用程式架構中的不足之處及待改進之處。Infrastructure Composer 透過視覺化建立及修改 CloudFormation 堆疊的便捷性與效率，提升您的開發體驗。您可以從初步草稿開始，建立可部署的程式碼，並將開發人員工作流程與 Infrastructure Composer 的視覺化設計工具進行整合。

此模式與 Infrastructure Composer 主控台有何差異？

儘管 CloudFormation 主控台版本的 Infrastructure Composer 與標準 Infrastructure Composer 主控台具備相似功能，但仍有部分差異。Lambda 相關卡片 (Lambda 函式與 Lambda Layer) 需要程式碼建置及套件化解決方案，而這些功能在 CloudFormation 主控台模式的 Infrastructure Composer 中並不提供。此模式也不支援本機同步功能。

不過，您可在 [Infrastructure Composer 主控台](#) 或 AWS Toolkit for Visual Studio Code 中使用這些 Lambda 相關卡片及本機同步功能。如需詳細資訊，請參閱《AWS Infrastructure Composer 開發人員指南<https://docs.aws.amazon.com/infrastructure-composer/latest/dg/what-is-composer.html>》和《AWS Toolkit for Visual Studio Code 使用者指南》中的 [Infrastructure Composer](#)。

AWS CloudFormation 語言伺服器

AWS CloudFormation 語言伺服器提供加速編寫infrastructure-as-code (IaC) 以及安全且自信地部署 AWS 資源的功能。它遵循[語言伺服器通訊協定](#) (LSP) 來提供有關暫留、自動完成、透過靜態驗證診斷的文件，請前往定義和程式碼動作。除了這些傳統語言伺服器功能之外，伺服器還新增線上功能，以透過 CloudFormation 探索和部署 AWS 資源。這包括使用變更集驗證和部署範本的能力；檢視堆疊差異、事件、資源和輸出；依類型列出堆疊和瀏覽資源；以及將即時資源狀態直接插入 CloudFormation 範本。

主題

- [IDEs與 AWS CloudFormation 語言伺服器整合](#)
- [開始使用](#)
- [在 IDE 中初始化 CloudFormation 專案（僅限 VS 程式碼）](#)
- [開放原始碼](#)
- [需要幫助？](#)

IDEs與 AWS CloudFormation 語言伺服器整合

AWS 透過 AWS Toolkit 為下列 IDEs 提供off-the-shelf整合：CloudFormation

- [Visual Studio 程式碼](#)
- [JetBrains IDEs](#)(2025.3 版或更新版本)，包括 IntelliJ IDEA、WebStorm 和 PyCharm

下列 IDEs也支援 CloudFormation 語言伺服器：

- [開羅](#)
- [游標](#)
- 大多數 VS 程式碼分支和分佈

CloudFormation Language Server 遵循[語言伺服器通訊協定](#) (LSP)，因此其他整合是可設定的。如需將語言伺服器與其他編輯器整合的說明，請參閱 [安裝指南](#)。

開始使用

主題

- [先決條件](#)
- [步驟 1：安裝或升級 AWS Toolkit](#)
- [步驟 2：存取 AWS Toolkit 中的 CloudFormation](#)
- [步驟 3：驗證、測試和精簡您的範本](#)
- [步驟 4：導覽範本](#)
- [步驟 5：驗證和部署](#)

先決條件

在開始之前，請確認：

- 您在支援的作業系統 (macOS、Windows 或) 上使用支援的 IDELinux。
- 您已為 IDE 安裝或升級至最新版本的 AWS Toolkit。

AWS CloudFormation 語言伺服器中的某些功能需要作用中 AWS 帳戶 且設定的登入資料。您必須使用有效的登入資料，AWS 帳戶 透過 AWS Toolkit 登入您的。

步驟 1：安裝或升級 AWS Toolkit

從 IDE 的延伸模組或外掛程式管理員安裝或更新至最新版本的 AWS Toolkit，然後重新啟動 IDE。

安裝後，AWS Toolkit 會自動啟用 CloudFormation IDE 支援。當您第一次使用 AWS CloudFormation 語言伺服器安裝或升級 AWS Toolkit 時，系統會提示您授予 AWS 收集匿名使用資料的許可。此資料有助於 AWS 改善 CloudFormation 語言伺服器，並增強撰寫體驗。不會收集任何敏感資訊，AWS 也不會記錄或存放範本內容、資源組態或任何可識別身分的客戶資料。您可以隨時從 IDE 設定變更遙測偏好設定。重新啟動 IDE 以使變更生效。所收集的用量資料僅著重於特徵互動和效能指標。這些洞見有助於 AWS 識別和排定改善的優先順序，例如更快的驗證、增強的自動完成和更好的錯誤診斷。

步驟 2：存取 AWS Toolkit 中的 CloudFormation

安裝 AWS Toolkit 之後，請在 IDE 中開啟 CloudFormation 面板。在 VS 程式碼中，從活動列開啟 AWS Toolkit 面板，然後選擇 CLOUDFORMATION。在 JetBrains IDEs 中，從側邊列開啟 AWS Toolkit 工具視窗，然後選取 CloudFormation 標籤。

CloudFormation 面板包含下列區段：

- **區域：**顯示目前的 AWS 區域。在 VS 程式碼中，您可以透過選取區域名稱或使用 AWS CloudFormation：從命令面板選取區域命令來變更它。在 JetBrains IDEs 中，區域是透過 AWS Toolkit 連線設定來設定。
- **堆疊：**顯示您帳戶中 CloudFormation 堆疊的分頁清單。展開堆疊以檢視其變更集節點，其中列出與該堆疊相關聯的變更集。使用檢視堆疊詳細資訊動作來開啟堆疊詳細資訊檢視，其中會顯示堆疊概觀、事件、輸出和資源。
- **資源：**新增資源類型之後，面板會在您的帳戶中顯示該類型的 AWS 資源。您可以檢視、重新整理、複製或匯入範本。

在 JetBrains IDEs 中，樹狀目錄上方的工具列可讓您快速存取常見動作，包括驗證和部署、重新執行驗證和部署、新增資源類型和重新整理。動作也可以透過在樹狀節點上按一下滑鼠右鍵的內容功能表來使用。

步驟 3：驗證、測試和精簡您的範本

當您撰寫 CloudFormation 範本時，IDE 會提供智慧型撰寫協助，協助您更快速地建立準確且合規的基礎設施。CloudFormation 語言伺服器會在背景執行，並提供下列撰寫功能：

- **程式碼完成：**根據 CloudFormation 結構描述建議資源類型、參數和屬性。
- **新增現有 AWS 資源：**可讓您將現有資源從 匯入 AWS 帳戶 範本。IDE 使用 [AWS 雲端控制 API \(CCAPI\)](#) 擷取資源的即時組態和屬性，協助您複製或重複使用範本中的現有基礎設施。
- **擷取至 參數：**當您的游標位於範本中的常值（例如，類似的字串 `t2.micro`）上時，IDE 會提供重構動作，將值擷取至 `Parameters` 區段，並將常值取代 `!Ref` 為新參數的。如果相同的常值出現在多個位置，您可以選擇一次擷取所有出現的項目。

將資源新增至範本

- **新增資源類型：**在 AWS 工具組 CloudFormation 面板的資源下，新增要瀏覽的資源類型。在 VS 程式碼中，按一下新增 + 圖示，或使用命令調色盤中的 `AWS CloudFormation：新增資源類型` 命令。在 JetBrains 中，按一下工具列中的新增資源類型按鈕，或以滑鼠右鍵按一下資源節點。

- 搜尋資源類型：在搜尋對話方塊中，輸入您要新增 AWS 的資源類型。範例：
 - `AWS::S3::Bucket`
 - `AWS::Lambda::Function`
- 瀏覽資源：在資源區段下，會顯示您帳戶中偵測到 AWS 的資源分頁清單。如果您有許多資源，則只會顯示第一頁。使用導覽控制項來瀏覽其他頁面，並檢視所有可用的資源。
- 選擇您要包含在範本中的資源。
- 視您的目標而定，您可以透過兩種方式將資源插入範本：
 - 複製現有資源：使用現有 AWS 資源的即時組態和屬性，在範本中建立新的資源。
 - 匯入現有資源：使用即時狀態將實際資源新增至您的範本，藉此將實際資源插入您的堆疊。

提示

- 您可以隨時重新整理資源區段，以檢視帳戶或區域中可用資源的最新清單。
- 如果您要匯入資源，請勿新增已屬於相同帳戶中現有 CloudFormation 堆疊的資源。
- 若要確認資源是否已由 CloudFormation 管理，請使用資源旁邊的資訊動作。在 VS 程式碼中，按一下  圖示。在 JetBrains IDEs 中，在資源上按一下滑鼠右鍵，然後選擇取得堆疊管理資訊。

新增相關資源

在 VS 程式碼中，您可以使用命令 `AWS CloudFormation`：依類型新增相關資源，將相關資源新增至選取的資源。從範本中已定義的資源類型中選取資源類型後，IDE 會顯示通常與該類型相關聯或相依的資源清單。例如，如果您選取 `AWS::EC2::Instance`，IDE 可能會建議新增相關資源，例如 `AWS::EC2::SecurityGroup` 或 `AWS::EC2::Subnet`。此功能可協助您快速建置連線的基礎設施元件，而無需手動搜尋相容的資源類型。JetBrains IDEs 目前不支援此功能。

靜態驗證

CloudFormation Language Server 提供內建的靜態驗證，採用 [AWS CloudFormation Linter \(cfn-lint\)](#) 和 技術 [AWS CloudFormation Guard](#)。當您編寫範本時，這些驗證會在幕後執行，協助您在部署之前識別語法錯誤、合規差距和最佳實務問題。

靜態驗證概觀

您會在 IDE 中看到兩種類型的即時靜態驗證：

- CloudFormation Linter (cfn-lint)：根據 CloudFormation 資源規格和結構描述規則驗證您的範本。

- Guard (cfn-guard)：根據合規規則和組織政策套件驗證您的範本。

CloudFormation Linter (cfn-lint)

CloudFormation Linter 已整合至 IDE，可在您輸入時自動檢查範本語法和結構。

- 結構描述驗證：偵測語法和結構描述錯誤，以確保您的範本符合 CloudFormation 資源結構描述。
- 錯誤反白顯示：在問題下顯示內嵌標記，代表部署封鎖程式或警告。
- 暫留說明：當您將滑鼠游標停留在錯誤上時，IDE 會顯示與該問題相關聯的診斷訊息。如果有快速修正，也會提供。

Guard 整合

Guard 會根據定義合規和安全性政策的規則集來驗證您的範本。IDE 透過 CloudFormation 語言伺服器即時執行 Guard 驗證，在您撰寫範本時提供立即意見回饋。

- 預設規則套件：IDE 包含一組預先註冊的 Guard 規則，著重於資源安全和組態衛生的基礎最佳實務。若要進一步了解，請參閱 [防護規則登錄檔](#)。
- 新增規則套件：若要新增或修改規則集，請開啟 IDE 設定並導覽至 Guard 組態區段，以選取或上傳其他 Guard 規則套件。

秘訣：了解診斷指標

- 藍色指標：最佳實務提示或最佳化建議。
- 黃色指標：非封鎖問題的警告（例如，缺少標籤或參數）。
- 紅色指標：部署封鎖程式，例如無效的屬性名稱、缺少必要欄位或結構描述不相符。

步驟 4：導覽範本

IDE 提供 CloudFormation 範本的結構化階層檢視，組織成 Parameters、Outputs、Resources 和等區段 Mappings，顯示每個資源類型和邏輯 ID。這可讓您輕鬆地快速尋找和導覽至大型範本中的特定資源或參數。在 VS 程式碼中，Explorer 邊欄中的大綱面板會顯示此結構。在 JetBrains IDEs 中，開啟結構工具視窗以檢視目前開啟檔案的範本結構。

您可以針對 GetAtt 和 等內部函數使用 Go to DefinitionRef，可讓您直接跳到範本中參考的資源或參數。這可協助您追蹤相依性、了解資源關係，並更有效率地進行編輯。

步驟 5：驗證和部署

當您準備好部署 CloudFormation 範本時，請使用驗證和部署功能來建立變更集。IDE 會驗證您的範本，如果找不到封鎖錯誤，則會繼續建立 [偏離感知的變更集](#)。IDE 接著會顯示差異檢視，讓您可以在執行變更集之前檢閱所有提議的變更。

在 VS 程式碼中，開啟命令面板並執行 AWS CloudFormation：驗證和部署。命令調色盤會逐步引導您選取範本、堆疊名稱、參數、功能和其他部署選項。在 JetBrains IDEs 中，使用驗證和部署工具列按鈕，在編輯器中的範本檔案上按一下滑鼠右鍵，或在樹狀目錄中的堆疊上按一下滑鼠右鍵。JetBrains 會顯示精靈對話方塊，您可以在其中設定所有部署選項，包括範本選擇、堆疊名稱、參數、功能、標籤和進階選項。

驗證的運作方式

IDE 會在 [部署之前自動執行驗證檢查](#)，並根據常見的失敗原因驗證您的範本，包括：

- 屬性語法或結構描述不相符無效：這些問題通常在撰寫 cfn-lint 期間由發現，但如果您繼續部署而不解決，CloudFormation 的部署時間驗證會在建立或更新堆疊之前顯示相同的錯誤。
- 資源名稱與您帳戶中的現有資源衝突。
- 服務特定的限制條件，例如 S3 儲存貯體名稱衝突或缺少加密。

如果驗證偵測到錯誤，IDE 會直接在範本中反白問題，並在診斷面板中列出錯誤。每個問題都包含導致失敗的特定屬性或資源，以及建議的修正。如果沒有封鎖錯誤，您可以繼續部署階段。

如果發現警告（非封鎖問題），則會出現一個對話方塊，允許您繼續部署或取消並進行更正。

IDE 會開啟 [偏離感知變更集](#)，顯示目前範本與部署堆疊組態之間的任何差異。這可讓您在執行之前檢閱、確認或取消變更集。取消部署會刪除變更集。

偏離感知變更集可讓您安全地處理堆疊偏離，以增強 CloudFormation 部署程序。當資源的實際狀態與 CloudFormation 範本中定義的狀態不同時，就會發生堆疊偏離，通常是由於透過 AWS 管理主控台、CLI 或 SDK 進行的手動變更。CloudFormation [偏離感知變更集](#) 會比較您處理過的堆疊組態與即時資源狀態，而 IDE 會呈現這些差異，讓您可以在部署之前將資源恢復合規。

檢視堆疊事件

部署開始時，您可以從 CloudFormation 面板即時監控進度。在堆疊事件下，您會看到部署期間執行的操作清單。每個事件都包含詳細資訊，例如：

- 時間戳記：事件發生的時間

- 資源：正在建立、更新或刪除的特定 AWS 資源
- 狀態：操作的目前狀態（例如 CREATE_IN_PROGRESS、UPDATE_COMPLETE 或 ROLLBACK_IN_PROGRESS）
- 原因：其他內容或錯誤訊息，如適用

您也可以從此面板檢視堆疊的資源和輸出。堆疊事件檢視可協助您追蹤部署進度、識別潛在問題，並確認堆疊何時成功完成。

在 IDE 中初始化 CloudFormation 專案（僅限 VS 程式碼）

在 IDE 中初始化 CloudFormation 專案可協助您設定具有正確資料夾、環境組態和 AWS 憑證的結構化工作區，以便您可以可靠地驗證和部署範本。您可以直接從 IDE 初始化新的 CloudFormation 專案，以建立此建議設定。此功能目前僅適用於 VS 程式碼，JetBrains IDEs 不支援此功能。

若要初始化 CloudFormation 專案：

- 開啟命令調色盤
 - 從 VS 程式碼中，開啟命令調色盤 (Ctrl+Shift+P Cmd+Shift+P 上的 或 macOS)。
 - 選擇 AWS CloudFormation : CFN Init : 初始化專案。
- 選擇專案目錄
 - 根據預設，IDE 會使用您目前的工作目錄。
 - 您可以將此路徑變更為您要存放 CloudFormation 範本的任何資料夾。
- 選取您的 AWS 登入資料設定檔
 - 系統會提示您選擇 AWS 登入資料設定檔。選取的設定檔用於環境偵測、驗證和部署。
- 設定您的環境
 - 系統會提示您建立或選取環境。
 - 環境會定義部署或驗證範本的位置和方式（例如 dev、beta 或 production）。您可以使用 AWS CloudFormation : CFN Init : 新增環境來選取或變更您的環境。
 - 您可以使用 AWS CloudFormation : CFN Init : Remove Environment 來移除您選取的環境。
- （選用）匯入參數檔案
 - 如果您已經有現有的參數檔案，IDE 可讓您在初始化期間匯入它們。
 - IDE 會自動偵測相容的檔案，並將其連結至您的專案，以用於範本驗證和部署。
- 命名並完成專案
 - 提供專案名稱，例如 beta-environment，並完成設定。

- IDE 會為您建立初始專案結構和組態檔案。

您可以直接從 IDE 執行驗證、預覽部署或切換環境。

開放原始碼

AWS CloudFormation 語言伺服器是根據 Apache-2.0 授權進行開放原始碼，為客戶提供範本診斷、結構描述驗證和靜態分析執行方式的完整透明度。對於在採用工具之前需要來源層級可見性的客戶，這可以減少安全和合規摩擦。

程式碼庫可在 GitHub 上公開取得：<https://github.com/aws-cloudformation/cloudformation-languageserver/>。

需要幫助？

試用 [CloudFormation 社群](#) AWS re:Post。

使用 IaC 產生器從現有資源中產生範本

使用 CloudFormation 基礎設施做為程式碼產生器 (IaC 產生器)，您可以使用您帳戶中佈建但尚未由 CloudFormation 管理 AWS 的資源產生範本。

以下是 IaC 產生器的優點：

- 將全部應用程式置於 CloudFormation 管理下，或將其移轉至 AWS CDK 應用程式。
- 產生範本，而不必依屬性來描述資源屬性，然後轉換為 JSON 或 YAML 語法。
- 使用範本在新帳戶或區域中複寫資源。

IaC 產生程序包含下列步驟：

1. 掃描資源 – 第一步是開始掃描資源。此掃描適用於整個區域，會在 30 天後過期。在此期間，可以從相同的掃描中建立多個範本。
2. 建立範本 – 若要建立範本，可使用兩個選項：
 - 從頭開始建立新範本，並將掃描的資源和相關資源新增至其中。
 - 使用現有的 CloudFormation 堆疊作為起點，並將掃描的資源和相關資源新增至其範本。
3. 匯入資源 – 使用您的範本將資源匯入為 CloudFormation 堆疊，或將其遷移至 AWS CDK 應用程式。

laC 產生器功能適用於所有商業區域，並支援許多常見的 AWS 資源類型。如需所支援資源的完整清單，請參閱[資源類型支援](#)。

主題

- [考量事項](#)
- [掃描資源所需的 IAM 許可](#)
- [範本產生、管理和刪除的常用命令](#)
- [將範本遷移至 AWS CDK](#)
- [透過 CloudFormation laC 產生器啟動資源掃描](#)
- [在 CloudFormation 主控台中檢視掃描摘要](#)
- [透過 laC 產生器掃描的資源建立 CloudFormation 範本](#)
- [透過掃描的資源建立 CloudFormation 堆疊](#)
- [解決唯寫屬性](#)

考量事項

您可以為具有讀取存取權 AWS 的資源產生 JSON 或 YAML 範本。laC 產生器功能的範本能夠可靠且快速地模型化雲端資源，而無需依屬性來描述資源屬性。

下表列出可用於 laC 產生功能的配額。

名稱	完整掃描	部分掃描
掃描中可處理的資源數量上限	100,000	100,000
每天的掃描數 (適用於資源少於 10,000 的掃描)	10	10
每天的掃描數 (適用於資源超過 10,000 的掃描)	1	1
每個帳戶產生的範本並行數量	5	5
針對一個範本產生而建模的資源並行數量	5	5
可在一個範本中建模的資源總數	500	500
每個帳戶產生的範本數量上限	1,000	1,000

⚠ Important

IaC 產生器僅支援 區域中 Cloud Control API 支援 AWS 的資源。如需詳細資訊，請參閱[資源類型支援](#)。

掃描資源所需的 IAM 許可

若要使用 IaC 產生器來掃描資源，則 IAM 主體 (使用者、角色或群組) 必須擁有下列權限：

- CloudFormation 掃描許可
- 目標 AWS 服務的讀取許可

掃描範圍僅限於您有權讀取的資源。缺少許可不會導致掃描失敗，但會排除這些資源。

如需可授予掃描和範本管理許可的 IAM 政策範例，請參閱 [允許所有 IaC 產生器操作](#)。

範本產生、管理和刪除的常用命令

使用 IaC 產生器的常用命令包括：

- [start-resource-scan](#)，開始掃描 AWS 區域中帳戶的資源。
- [describe-resource-scan](#)，監控資源掃描的進度。
- [list-resource-scans](#)，列出 AWS 區域中的資源掃描。
- [list-resource-scan-resources](#)，列出資源掃描期間找到的資源。
- [list-resource-scan-related-resources](#)，列出與掃描資源相關的資源。
- [create-generated-template](#)，從一組掃描的資源中產生 CloudFormation 範本。
- [update-generated-template](#)，更新產生的範本。
- [describe-generated-template](#)，傳回所產生範本的相關資訊。
- [list-generated-templates](#)，列出帳戶和目前區域中所有產生的範本。
- [delete-generated-template](#)，刪除產生的範本。

將範本遷移至 AWS CDK

AWS Cloud Development Kit (AWS CDK) 是開放原始碼軟體開發架構，可讓您使用熱門的程式設計語言來開發、管理和部署 CloudFormation 資源。

CLI AWS CDK 提供與 IaC 產生器的整合。使用 AWS CDK CLI `cdk migrate` 命令來轉換 CloudFormation 範本，並建立新的 CDK 應用程式，其中包含您的資源。然後，您可以使用 AWS CDK 來管理您的資源並部署到 CloudFormation。

如需詳細資訊，請參閱《AWS Cloud Development Kit (AWS CDK) 開發人員指南》中的[移轉至 AWS CDK](#)。

透過 CloudFormation IaC 產生器啟動資源掃描

從現有資源建立範本前，您必須先啟動資源掃描，以探索目前的資源及其彼此間的關係。

您可以透過下列其中一種方式啟動資源掃描。對於 IaC 產生器的首次使用者，建議選擇第一種方式。

- 掃描所有資源 (完整掃描) – 掃描目前帳戶和區域中的所有現有資源。對於 1,000 個資源，此掃描程序可能需要長達 10 分鐘。
- 掃描特定資源 (部分掃描) – 手動選擇要在目前帳戶和區域中掃描的資源類型。此選項提供更快且更具針對性的掃描程序，非常適用於反覆式範本開發。

掃描完成後，您可以選擇要在產生範本時納入哪些資源及其相關資源。使用部分掃描時，相關資源僅在下列任一情況下會在範本產生期間可供使用：

- 您在啟動掃描前已明確選取它們，或
- 它們是探索您所選資源類型必需的資源。

例如，若您選取 `AWS::EKS::Nodegroup`，但未選取 `AWS::EKS::Cluster`，IaC 產生器會自動將 `AWS::EKS::Cluster` 資源納入掃描，因為探索節點群組需先探索叢集。在所有其他情況下，掃描僅會包含您明確選取的資源。

Note

繼續操作之前，請確認您具備使用 IaC 產生器所需的許可。如需詳細資訊，請參閱[掃描資源所需的 IAM 許可](#)。

主題

- [啟動資源掃描 \(主控台\)](#)
- [啟動資源掃描 \(AWS CLI\)](#)

啟動資源掃描 (主控台)

啟動所有資源類型的資源掃描 (完整掃描)

1. 開啟 CloudFormation 主控台的 [laC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，選擇包含掃描資源的 AWS 區域。
3. 從掃描面板中，選擇開始新的掃描，再選擇掃描所有資源。

啟動特定資源類型的資源掃描 (部分掃描)

1. 開啟 CloudFormation 主控台的 [laC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，選擇包含掃描資源的 AWS 區域。
3. 從掃描面板中，選擇啟動新的掃描，再選擇掃描特定資源。
4. 在啟動部分掃描對話方塊中，選取最多 100 種資源類型，然後選擇啟動掃描。

啟動資源掃描 (AWS CLI)

啟動所有資源類型的資源掃描 (完整掃描)

使用以下 [start-resource-scan](#) 命令。將 *us-east-1* 取代 AWS 區域 為包含要掃描之資源的。

```
aws cloudformation start-resource-scan --region us-east-1
```

如果成功，此命令會傳回掃描的 ARN。請注意 ResourceScanId 屬性中的 ARN。您建立範本時會需要用到它。

```
{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60"
}
```

啟動特定資源類型的資源掃描 (部分掃描)

1. 使用下列 [cat](#) 命令，將要掃描的資源類型儲存至主目錄中名為 config.json 的 JSON 檔案。下列是掃描設定範例，會掃描 Amazon EC2 執行個體、安全群組及所有 Amazon S3 資源。

```
$ cat > config.json
```

```
[
  {
    "Types": [
      "AWS::EC2::Instance",
      "AWS::EC2::SecurityGroup",
      "AWS::S3::*"
    ]
  }
]
```

2. 使用 [start-resource-scan](#) 命令，搭配 `--scan-filters` 選項和您建立的 `config.json` 檔案，以啟動部分掃描。將 `us-east-1` 取代為包含要掃描資源的 AWS 區域。

```
aws cloudformation start-resource-scan --scan-filters file://config.json --
region us-east-1
```

如果成功，此命令會傳回掃描的 ARN。請注意 `ResourceScanId` 屬性中的 ARN。您建立範本時會需要用到它。

```
{
  "ResourceScanId":
    "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60"
}
```

監控資源掃描的進度

使用 [describe-resource-scan](#) 命令。對於 `--resource-scan-id` 選項，請將範例 ARN 取代為實際 ARN。

```
aws cloudformation describe-resource-scan --region us-east-1 \
--resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60
```

如果此命令成功執行，您會看到類似如下的輸出傳回：

```
{
  "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
  "Status": "COMPLETE",
}
```

```

    "StartTime": "2023-08-21T03:10:38.485000+00:00",
    "EndTime": "2023-08-21T03:20:28.485000+00:00",
    "PercentageCompleted": 100.0,
    "ResourceTypes": [
        "AWS::CloudFront::CachePolicy",
        "AWS::CloudFront::OriginRequestPolicy",
        "AWS::EC2::DHCPOptions",
        "AWS::EC2::InternetGateway",
        "AWS::EC2::KeyPair",
        "AWS::EC2::NetworkAcl",
        "AWS::EC2::NetworkInsightsPath",
        "AWS::EC2::NetworkInterface",
        "AWS::EC2::PlacementGroup",
        "AWS::EC2::Route",
        "AWS::EC2::RouteTable",
        "AWS::EC2::SecurityGroup",
        "AWS::EC2::Subnet",
        "AWS::EC2::SubnetCidrBlock",
        "AWS::EC2::SubnetNetworkAclAssociation",
        "AWS::EC2::SubnetRouteTableAssociation",
        ...
    ],
    "ResourcesRead": 676
}

```

如需部分掃描，輸出格式應類似以下內容：

```

{
    "ResourceScanId": "arn:aws:cloudformation:region:account-id:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60",
    "Status": "COMPLETE",
    "StartTime": "2025-03-06T18:24:19.542000+00:00",
    "EndTime": "2025-03-06T18:25:23.142000+00:00",
    "PercentageCompleted": 100.0,
    "ResourceTypes": [
        "AWS::EC2::Instance",
        "AWS::EC2::SecurityGroup",
        "AWS::S3::Bucket",
        "AWS::S3::BucketPolicy"
    ],
    "ResourcesRead": 65,
    "ScanFilters": [
        {

```

```
        "Types": [
            "AWS::EC2::Instance",
            "AWS::EC2::SecurityGroup",
            "AWS::S3::*"
        ]
    }
}
```

如需輸出中欄位的說明，請參閱《AWS CloudFormation API 參考》中的 [DescribeResourceScan](#)。

在 CloudFormation 主控台中檢視掃描摘要

掃描完成後，可以檢視掃描期間找到的資源視覺化，以協助您識別不同產品類型的資源集中度。

檢視掃描期間找到之資源的相關資訊

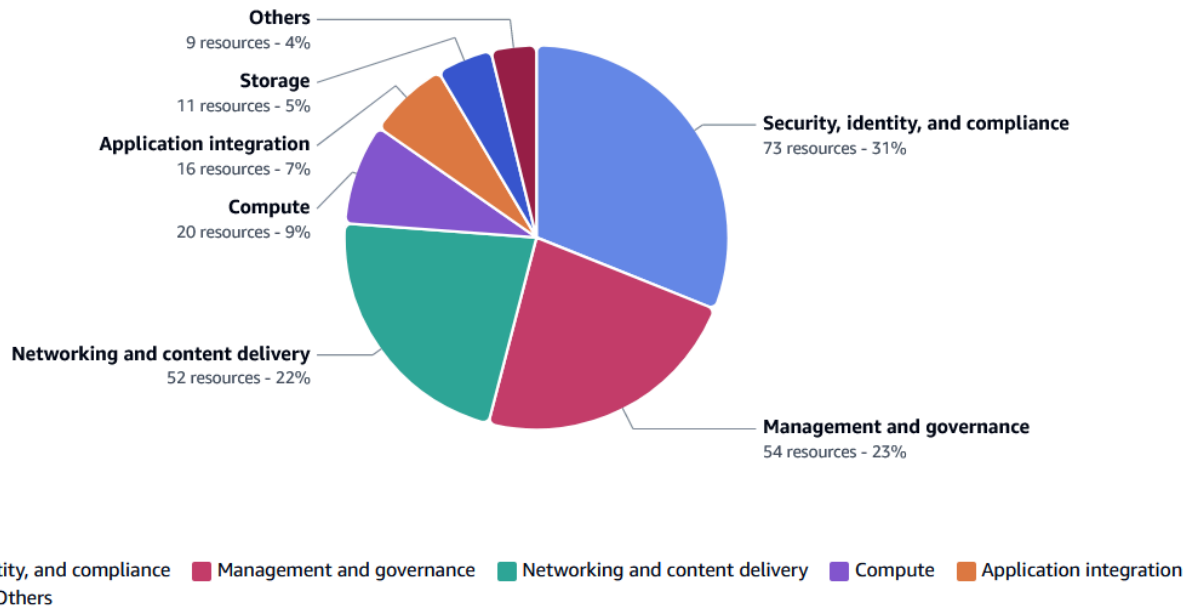
1. 開啟 CloudFormation 主控台的 [IaC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，選擇包含要檢視之資源掃描 AWS 區域 的。
3. 在導覽窗格中，選擇 IaC 產生器。
4. 在掃描資源明細中，可以依產品類型找到掃描資源的視覺化明細，例如運算和儲存。
5. 若要自訂顯示的產品類型數量，請選擇篩選顯示的資料。這可協助您量身打造視覺化效果，以專注於您最感興趣的產品類型。
6. 頁面右側是掃描摘要詳細資訊面板。如要開啟該面板，請選擇開啟面板圖示。

Scanned resources breakdown 235

View a visual breakdown of your scanned resources by product types. Select a product type for a further breakdown.

Filter displayed data

Filter by product type ▼



透過 IaC 產生器掃描的資源建立 CloudFormation 範本

本主題說明如何從透過 IaC 產生器功能掃描的資源建立範本。

透過掃描的資源建立範本 (主控台)

透過掃描的資源建立堆疊範本

1. 開啟 CloudFormation 主控台的 [IaC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，選擇包含掃描資源的 AWS 區域。
3. 從範本區段，選擇建立範本。
4. 選擇從新範本開始。
 - a. 對於範本名稱，提供範本的名稱。
 - b. (選用) 設定您的刪除政策和更新取代政策。
 - c. 選擇下一步，將掃描的資源新增至範本。

5. 對於新增掃描的資源，瀏覽掃描的資源清單，並選取您要新增至範本的資源。您可以依據資源識別碼、資源類型或標籤篩選資源。這些篩選條件是相互包含的。
6. 將所有需要的資源新增至範本後，選擇下一步以離開新增掃描的資源頁面，然後繼續前往新增相關資源頁面。
7. 檢閱建議的相關資源清單。Amazon EC2 執行個體與安全群組等相關資源是相互依存的，通常歸屬於相同的工作負載。選取您要納入產生範本的相關資源。

Note

建議您將所有相關資源新增至此範本。

8. 檢閱範本詳細資訊、掃描的資源及相關資源。
9. 選擇建立範本，以離開檢閱和建立頁面，並建立範本。

透過掃描的資源建立範本 (AWS CLI)

透過掃描的資源建立堆疊範本

1. 使用 [list-resource-scan-resources](#) 命令列出掃描期間找到的資源，並可選擇性指定限制輸出的 `--resource-identifier` 選項。對於 `--resource-scan-id` 選項，請將範例 ARN 取代為實際 ARN。

```
aws cloudformation list-resource-scan-resources \
  --resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resource-identifier MyApp
```

下列是回應範例，其中 `ManagedByStack` 表示 CloudFormation 是否已管理該資源。複製輸出內容。下一個步驟需要此值。

```
{
  "Resources": [
    {
      "ResourceType": "AWS::EKS::Cluster",
      "ResourceIdentifier": {
        "ClusterName": "MyAppClusterName"
      },
      "ManagedByStack": false
    }
  ]
}
```

```
    },
    {
      "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
      "ResourceIdentifier": {
        "AutoScalingGroupName": "MyAppASGName"
      },
      "ManagedByStack": false
    }
  ]
}
```

如需輸出中欄位的說明，請參閱《AWS CloudFormation API 參考》中的 [ScannedResource](#)。

2. 使用 `cat` 命令，將資源類型和識別碼儲存至主目錄中名為 `resources.json` 的 JSON 檔案。以下是基於上述步驟中範例命令的範例 JSON。

```
$ cat > resources.json
[
  {
    "ResourceType": "AWS::EKS::Cluster",
    "ResourceIdentifier": {
      "ClusterName": "MyAppClusterName"
    }
  },
  {
    "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
    "ResourceIdentifier": {
      "AutoScalingGroupName": "MyAppASGName"
    }
  }
]
```

3. 使用 [list-resource-scan-related-resources](#) 命令，搭配您建立的 `resources.json` 檔案，列出與您掃描資源相關的資源。

```
aws cloudformation list-resource-scan-related-resources \
  --resource-scan-id arn:aws:cloudformation:us-east-1:123456789012:resourceScan/0a699f15-489c-43ca-a3ef-3e6ecfa5da60 \
  --resources file://resources.json
```

下列是回應範例，其中 `ManagedByStack` 表示 CloudFormation 是否已管理該資源。將這些資源新增至您在上一步驟建立的 JSON 檔案。您建立範本時會需要用到它。


```
{
  "RelatedResources": [
    {
      "ResourceType": "AWS::EKS::Nodegroup",
      "ResourceIdentifier": {
        "NodegroupName": "MyAppNodegroupName"
      },
      "ManagedByStack": false
    },
    {
      "ResourceType": "AWS::IAM::Role",
      "ResourceIdentifier": {
        "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
      },
      "ManagedByStack": false
    }
  ]
}
```

如需輸出中欄位的說明，請參閱《AWS CloudFormation API 參考》中的 [ScannedResource](#)。

Note

輸入的資源清單長度不得超過 100。若要列出超過 100 個資源的相關資源，請以每批次 100 個的方式執行 `list-resource-scan-related-resources` 命令，並合併結果。請注意，輸出清單中可能包含重複的資源。

4. 若要建立新的啟動範本，請按如下所示使用 [create-generated-template](#) 命令，且進行這些修改：

- *us-east-1* 將取代 AWS 區域 為包含掃描資源的。
- 將 *MyTemplate* 取代為要建立的範本名稱。

```
aws cloudformation create-generated-template --region us-east-1 \
  --generated-template-name MyTemplate \
  --resources file://resources.json
```

以下是範例 `resources.json` 檔案。

```
[
```

```
{
  "ResourceType": "AWS::EKS::Cluster",
  "LogicalResourceId": "MyCluster",
  "ResourceIdentifier": {
    "ClusterName": "MyAppClusterName"
  }
},
{
  "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
  "LogicalResourceId": "MyASG",
  "ResourceIdentifier": {
    "AutoScalingGroupName": "MyAppASGName"
  }
},
{
  "ResourceType": "AWS::EKS::Nodegroup",
  "LogicalResourceId": "MyNodegroup",
  "ResourceIdentifier": {
    "NodegroupName": "MyAppNodegroupName"
  }
},
{
  "ResourceType": "AWS::IAM::Role",
  "LogicalResourceId": "MyRole",
  "ResourceIdentifier": {
    "RoleId": "arn:aws::iam::account-id:role/MyAppIAMRole"
  }
}
]
```

如果成功，此命令會傳回下列內容：

```
{
  "Arn":
    "arn:aws:cloudformation:region:account-id:generatedtemplate/7fc8512c-d8cb-4e02-b266-d39c48344e48",
  "Name": "MyTemplate"
}
```

透過掃描的資源建立 CloudFormation 堆疊

建立範本後，您可在建立堆疊和匯入掃描的資源之前，透過 Infrastructure Composer 預覽產生的範本。這能協助您視覺化完整的應用程式架構，以及資源與其彼此間的關係。如需有關 Infrastructure Composer 的更多資訊，請參閱 [透過 Infrastructure Composer 視覺化建立範本](#)。

建立堆疊並匯入掃描的資源

1. 開啟 CloudFormation 主控台的 [IaC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，AWS 區域 為您的範本選擇。
3. 選擇範本索引標籤，然後您的範本名稱以檢視更多資訊。您的範本名稱以檢視更多資訊。
4. 在範本定義索引標籤的範本區段頂端，您可根據偏好設定，將範本語法在 YAML 和 JSON 之間切換。
5. 檢閱範本的詳細資訊，確保所有設定皆正確無誤。為了便於檢閱和理解範本，您可以透過 Infrastructure Composer 將預設的程式碼檢視切換為範本中所描述基礎設施的圖形化檢視。為此，請在範本下選擇畫布，而不是範本。

畫面動作

- 若要著重檢視範本中特定資源的詳細資訊，請按兩下卡片以開啟資源屬性面板。
 - 若要在畫布上視覺化排列和整理卡片，請從畫布左上角選擇排列。
 - 要放大和縮小畫布，請使用畫布右下角的縮放控制項。
6. 要在主控台中檢視特定資源，請選擇範本資源索引標籤，然後選擇要查看的資源實體 ID。這會將您導向該特定資源的主控台。您也可以透過範本資源索引標籤，在範本定義中新增、移除和重新同步資源。
 7. 在範本定義索引標籤上，IaC 產生器可能會對包含唯寫屬性的資源發出警告。檢閱警告後，您可下載產生的範本並進行任何必要變更。如需詳細資訊，請參閱[解決唯寫屬性](#)。
 8. 當您對範本定義感到滿意時，在範本定義索引標籤上，選擇匯入至堆疊，然後選擇下一步。
 9. 在指定堆疊詳細資訊頁面的指定堆疊面板中，輸入您的堆疊的名稱，然後選擇下一步。
 10. 檢閱並輸入堆疊的參數。選擇下一步。
 11. 在檢閱變更頁面上檢閱您的選項，然後選擇下一步。
 12. 在檢閱和匯入頁面上檢閱您的詳細資訊，然後選擇匯入資源。

解決唯寫屬性

使用 CloudFormation IaC 產生器，您可以使用已在帳戶中佈建且尚未由 CloudFormation 管理的資源來產生範本。不過，某些資源屬性會指定為唯寫，這表示其可以寫入，但 CloudFormation 無法讀取，例如資料庫密碼。

透過現有資源產生 CloudFormation 範本時，唯寫屬性會帶來挑戰。在大多數情況下，CloudFormation 會將這些屬性轉換為所產生範本中的參數。這可讓您在匯入操作期間將屬性輸入為參數值。不過，在某些情況下無法進行此轉換，CloudFormation 會以不同的方式處理這些案例。

互斥屬性

有些資源有多個互斥屬性集，其中至少有些是唯讀屬性。在這些情況下，IaC 產生器無法判斷在建立期間將哪組專用屬性套用至資源。例如，可以使用其中一個屬性集來提供 [AWS::Lambda::Function](#) 的程式碼。

- Code/S3Bucket、Code/S3Key 和可選的 Code/S3ObjectVersion
- Code/ImageUri
- Code/ZipFile

所有這些屬性都是唯寫屬性。IaC 產生器會選取其中一個專用屬性集，並將其新增至產生的範本。系統會為每個唯寫屬性新增參數。參數名稱包含 OneOf，參數說明指出對應的屬性可以取代為其他專用屬性。IaC 產生器會為包含的屬性設定 MUTUALLY_EXCLUSIVE_PROPERTIES 的警告類型。

互斥類型

在某些情況下，唯寫屬性可以是多種資料類型。例如，[AWS::ApiGateway::RestApi](#) 的 Body 屬性可以是 object 或 string。在這種情況下，IaC 產生器會使用 string 的類型在產生的範本中包含屬性，並設定 MUTUALLY_EXCLUSIVE_TYPES 的警告類型。

Array 屬性

如果唯寫屬性具有 array 類型，則 IaC 產生器無法將其包含在產生的範本中，因為參數只能是純量值。在此情況下，會從範本中省略該屬性，並設定 UNSUPPORTED_PROPERTIES 的警告類型。

可選屬性

對於選用的唯寫屬性，IaC 產生器無法偵測在設定資源時是否使用屬性。在此情況下，會從產生的範本中省略該屬性，並設定 UNSUPPORTED_PROPERTIES 的警告類型。

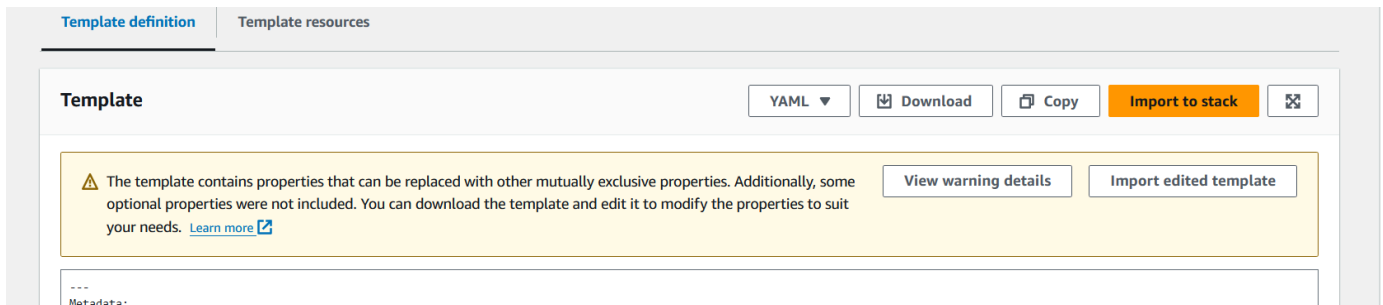
警告和後續步驟

若要確定哪些屬性為唯讀屬性，必須查看 IaC 產生器主控台傳回的警告。[AWS 資源和屬性類型參考](#)不會指出屬性是否為唯讀屬性，或者它是否支援多種類型。

或者，可以從資源提供者結構描述中查看哪些屬性是唯讀屬性。若要下載資源提供者結構描述，請參閱[CloudFormation 資源提供者結構描述](#)。

若要解決唯寫屬性的問題

1. 開啟 CloudFormation 主控台的 [IaC 產生器頁面](#)。
2. 在畫面頂端的導覽列上，AWS 區域 為您的範本選擇。
3. 選擇範本索引標籤，然後選擇建立的範本名稱。
4. 在範本定義索引標籤中，當產生的範本包含具有唯寫屬性的資源時，IaC 產生器主控台會顯示警告，其中包含問題類型的摘要。例如：



5. 選擇檢視警告詳細資訊以取得更多詳細資訊。由所產生的範本和資源類型中使用的邏輯 ID 來識別具有唯寫屬性的資源。

使用警告清單來識別具有唯寫屬性的資源，並查看每個資源，以確定需要對產生的範本進行哪些變更 (如果有的話)。

Resource warnings (1)

▼ LambdaFunction00IaClambda002fD6K (AWS::Lambda::Function) [Learn more](#)

Unsupported properties	Mutually exclusive properties
1. SnapStart/ApplyOn Applying SnapStart setting on function resource type. Possible values: [PublishedVersions, None] This property can be replaced with other types	1. Code/S3Bucket An Amazon S3 bucket in the same AWS Region as your function. The bucket can be in a different AWS account. This property can be replaced with other exclusive properties
2. Code/S3ObjectVersion For versioned objects, the version of the deployment package object to use. This property can be replaced with other exclusive properties	2. Code/S3Key The Amazon S3 key of the deployment package. This property can be replaced with other exclusive properties

6. 如果必須更新範本以解決唯寫屬性的問題，請執行下列動作：

- a. 選擇下載以下載範本副本。
- b. 編輯範本。
- c. 變更完成後，可以選擇匯入已編輯的範本按鈕以繼續匯入程序。

如何解決 AWS::ApiGateway::RestAPI 資源中唯寫屬性的問題

本主題說明如何在使用 IaC 產生器時解決 [AWS::ApiGateway::RestApi](#) 資源中唯寫屬性的問題。

問題

當產生的範本包含 AWS::ApiGateway::RestApi 資源時，會產生警告，指出 Body、BodyS3Location 和 CloneFrom 屬性已標識為 UNSUPPORTED_PROPERTIES。這是因為其是選用的唯寫屬性。IaC 產生器不知道這些屬性是否曾經套用至資源。因此，它會在產生的範本中忽略這些屬性。

Resolution

若要設定 REST API 的 Body 屬性，請更新產生的範本。

1. 使用 Amazon API Gateway [GetExport](#) API 動作來下載 API。例如，使用 [aws apigateway get-export](#) AWS CLI 命令。如需詳細資訊，請參閱《API Gateway 開發人員指南》中的[從 API Gateway 匯出 REST API](#)。
2. 從 GetExport API 動作的回應中擷取 Body 屬性。將其上傳至 Amazon S3 儲存貯體。
3. 下載產生的範本。
4. 將 BodyS3Location/Bucket 和 BodyS3Location/Key 屬性新增至範本，指定存放 Body 的儲存貯體名稱和金鑰。
5. 在 IaC 產生器主控台中開啟產生的範本，然後選擇匯入已編輯的範本。

如何解決 AWS::Lambda::Function 資源中唯寫屬性的問題

本主題說明如何在使用 IaC 產生器時解決 [AWS::Lambda::Function](#) 資源中唯寫屬性的問題。

問題

AWS::Lambda::Function 資源有三個互斥的屬性集，用於指定 Lambda 程式碼：

- Code/S3Bucket 和 Code/S3Key 屬性，以及選用的 Code/S3ObjectVersion 屬性

- Code/ImageUri 屬性
- Code/ZipFile 屬性

只有其中一個集合可用於指定的 `AWS::Lambda::Function` 資源。

IaC 產生器無法確定用於建立或更新資源的專屬唯寫屬性集。因此，它只會在產生的範本中包含第一組屬性。忽略 Code/ImageUri 和 Code/ZipFile 屬性。

此外，IaC 產生器會發出下列警告：

- **MUTUALLY_EXCLUSIVE_PROPERTIES** – 警告 Code/S3Bucket 和 Code/S3Key 被標識為互斥屬性。
- **UNSUPPORTED_PROPERTIES** – 警告 Code/S3ObjectVersion 屬性不受支援。

若要在產生的範本中包含 `AWS::Lambda::Function` 資源，必須下載並更新具有正確程式碼屬性的範本。

Resolution

如果將 Lambda 程式碼存放在 Amazon S3 儲存貯體中，但不使用 **S3ObjectVersion** 屬性，可以匯入產生的範本，無需進行任何修改。在匯入操作期間，IaC 產生器會要求您使用 Amazon S3 儲存貯體和金鑰作為範本參數。

如果將 Lambda 程式碼儲存為 Amazon ECR 儲存庫，可以使用下列指示來更新範本：

1. 下載產生的範本。
2. 從產生的範本中移除 Code/S3Bucket 和 Code/S3Key 屬性的屬性和對應參數。
3. 使用 Code/ImageUri 屬性取代所產生範本中已移除的屬性，指定 Amazon ECR 儲存庫的 URL。
4. 在 IaC 產生器主控台中開啟產生的範本，然後選擇匯入已編輯的範本按鈕。

如果將 Lambda 程式碼儲存為 zip 檔案，可以使用下列指示來更新範本：

1. 下載產生的範本。
2. 從產生的範本中移除 Code/S3Bucket 和 Code/S3Key 屬性的屬性和對應參數。
3. 使用 Code/ZipFile 屬性來取代所產生範本中已移除的屬性。

4. 在 IaC 產生器主控台中開啟產生的範本，然後選擇匯入已編輯的範本按鈕。

如果沒有 Lambda 程式碼的副本，可以使用下列指示來更新範本：

1. 使用 API AWS Lambda [GetFunction](#) 動作（例如，使用 [aws lambda get-function](#) AWS CLI 命令）。
2. 在回應中，如果程式碼位於 Amazon S3 儲存貯體中，則 RepositoryType 參數為 S3，如果程式碼位於 Amazon ECR 儲存庫中，則為 ECR。
3. 在回應中，Location 參數包含預先簽章的 URL，可用來下載部署套件，時長 10 分鐘。下載程式碼。
4. 將程式碼上傳至 Amazon S3 儲存貯體。
5. 使用產生的範本來執行匯入操作，並提供儲存貯體名稱和金鑰作為參數值。

使用動態參考取得存放在其他服務中的值

動態參考可讓您方便地指定在其他服務中存放且管理的外部值，並從基礎設施即程式碼範本中分離敏感資訊。在堆疊和變更集操作期間，CloudFormation 會在必要時擷取指定參考的值。

透過動態參考，您可以：

- 使用安全字串 – 對於敏感資料，請一律使用 AWS Systems Manager 參數存放區中的安全字串參數或中的秘密 AWS Secrets Manager，以確保靜態加密您的資料。
- 限制存取 – 僅將 Parameter Store 參數或 Secrets Manager 秘密的存取權限制給已授權的主體和角色。
- 輪換憑證 – 定期輪換儲存在 Parameter Store 或 Secrets Manager 中的敏感資料，以維持高層級的安全性。
- 自動化輪換 – 利用 Secrets Manager 的自動輪換功能，在應用程式和環境中定期更新和分發敏感資料。

一般考量事項

以下是在 CloudFormation 範本中指定動態參考之前要考慮的一般考量事項：

- 不要在作為資源主識別碼一部分的資源屬性中包含動態引用或任何敏感資料。CloudFormation 可能會在主要資源識別符中使用實際的純文字值，這可能是安全風險。此資源 ID 可能會出現在任何衍生輸出或目的地中。

若要確定哪些資源屬性組成資源類型的主要識別碼，請參閱 [AWS 資源和屬性類型參考](#) 中該資源的資源參考文件。在 Return values (傳回值) 區段中，Ref 函數傳回值表示組成資源類型主要識別碼的資源屬性。

- 您在一個堆疊範本中最多可以包含 60 個動態參考。
- 如果使用轉換 (例如 `AWS::Include` 或 `AWS::Serverless`)，CloudFormation 不會在套用轉換之前解析動態參考。反之，它會將動態參考的常值字串傳遞至轉換，並在您使用範本執行變更集時解析參考。
- 動態參考無法用於自訂資源中的安全值 (例如儲存在 Parameter Store 或 Secrets Manager 中的值)。
- `AWS::CloudFormation::Init` 中繼資料和 Amazon EC2 UserData 屬性也不支援動態參考。
- 請勿建立以反斜線 (\) 結尾的動態參考。CloudFormation 無法解析這些參考，這會導致堆疊操作失敗。

下列主題提供使用動態參考的資訊和其他考量事項。

主題

- [從 Systems Manager Parameter Store 取得純文字值](#)
- [從 Systems Manager Parameter Store 中取得安全字串值](#)
- [從 Secrets Manager 中取得秘密或秘密值](#)

從 Systems Manager Parameter Store 取得純文字值

建立 CloudFormation 範本時，建議您使用儲存在參數存放區中的純文字值。參數存放區是 `aws:ssm` 的功能 AWS Systems Manager。如需 Parameter Store 的簡介，請參閱《AWS Systems Manager 使用者指南》中的 [AWS Systems Manager Parameter Store](#)。

若要在範本中使用 Parameter Store 的純文字值，請使用 `ssm` 動態參考。此參考可讓您存取 Parameter Store 中 String 或 StringList 參數類型的值。

若要驗證將在堆疊操作中使用的 `ssm` 動態參考版本，請為堆疊操作建立變更集。然後，在範本索引標籤中檢閱已處理的範本。如需詳細資訊，請參閱[建立 CloudFormation 堆疊的變更集](#)。

使用 `ssm` 動態參考時，需要謹記幾個重要事項：

- CloudFormation 不支援動態參考的偏離偵測。對於尚未指定參數版本的 `ssm` 動態參考，如果更新 Systems Manager 中的參數版本，建議也可以對包含 `ssm` 動態參考的任何堆疊執行堆疊更新操作，以便擷取最新的參數版本。

- 若要在 CloudFormation 範本的 Parameters 區段中使用 ssm 動態參考，必須包含版本編號。CloudFormation 不允許您在本節中參考沒有版本號碼的參數存放區值。或者，可以在範本中將參數定義為 Systems Manager 參數類型。執行此操作時，可以將 Systems Manager 參數金鑰指定為參數的預設值。然後，CloudFormation 將從 Parameter Store 中擷取參數值的最新版本，而無需指定版本編號。這可讓您的範本更簡單且更容易維護。如需詳細資訊，請參閱[使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)。
- 對於自訂資源，CloudFormation 在將請求傳送到自訂資源之前會解析 ssm 動態參考。
- CloudFormation 不支援使用動態參考來參考從另一個 AWS 帳戶共用的參數。
- CloudFormation 在動態參考中不支援使用 Systems Manager 參數標籤。

許可

若要指定存放在 Systems Manager 參數存放區的參數，您必須擁有許可才能呼叫 [GetParameters](#) 以取得指定的參數。若要了解如何建立可提供特定 Systems Manager 參數存取權的 IAM 政策，請參閱《AWS Systems Manager 使用者指南》中的[使用 IAM 政策限制對 Systems Manager 參數的存取](#)。

參考模式

若要在 CloudFormation 範本中參考存放在 Systems Manager Parameter Store 中的純文字值，請使用下列 ssm 參考模式。

```
{{resolve:ssm:parameter-name:version}}
```

針對 parameter-name 和 version，您的參考必須遵守下列規則表達式模式：

```
{{resolve:ssm:[a-zA-Z0-9_.\-/+](:\d+)?}}
```

parameter-name

參數存放區中的參數名稱。函數名稱區分大小寫。

必要。

version

整數，指定要使用的參數版本。如果您沒有指定確切的版本，CloudFormation 會在您建立或更新堆疊時使用最新版本的參數。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的[使用參數版本](#)。

選用。

範例

主題

- [公有 AMI ID 參數](#)
- [自訂 AMI ID 參數](#)

公有 AMI ID 參數

下列範例會建立參考公有 AMI 參數的 EC2 執行個體。動態參考會從公有參數中擷取最新的 Amazon Linux 2023 AMI ID。如需公有參數的詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的[探索參數存放區中的公有參數](#)。

JSON

```
{
  "Resources": {
    "MyInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
        "InstanceType": "t2.micro"
      }
    }
  }
}
```

YAML

```
Resources:
  MyInstance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}'
      InstanceType: t2.micro
```

自訂 AMI ID 參數

下列範例會建立 EC2 啟動範本，它可參考儲存在 Parameter Store 中的自訂 AMI ID。動態參考會在執行個體從啟動範本啟動時，從 *golden-ami* 參數的 2 版本中擷取 AMI ID。

JSON

```
{
  "Resources": {
    "MyLaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Properties": {
        "LaunchTemplateName": {
          "Fn::Sub": "${AWS::StackName}-launch-template"
        },
        "LaunchTemplateData": {
          "ImageId": "{{resolve:ssm:golden-ami:2}}",
          "InstanceType": "t2.micro"
        }
      }
    }
  }
}
```

YAML

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: '{{resolve:ssm:golden-ami:2}}'
        InstanceType: t2.micro
```

從 Systems Manager Parameter Store 中取得安全字串值

在 CloudFormation 中，您可以將敏感資料儲存為 AWS Systems Manager 參數存放區中的「安全字串」，以使用密碼或授權金鑰等敏感資料，而無需直接在範本中公開。如需 Parameter Store 的簡介，請參閱《AWS Systems Manager 使用者指南》中的 [AWS Systems Manager Parameter Store](#)。

若要在範本中使用 Parameter Store 安全字串，請使用 `ssm-secure` 動態參考。CloudFormation 永遠不會存放實際的安全字串值。相反，它僅存放文字動態參考，其中包含安全字串的純文字參數名稱。

在堆疊建立或更新期間，CloudFormation 會視需要存取安全字串值，而不會公開實際值。安全字串僅用於有支援 `ssm-secure` 動態參考模式的資源屬性。如需詳細資訊，請參閱 [對安全字串支援動態參數模式的資源](#)。

CloudFormation 不會在任何 API 呼叫中傳回安全字串的實際參數值。它只會傳回常值動態參考。使用變更集來比較變更時，CloudFormation 只會比較文字動態參考字串。它不會解析和比較實際安全字串值。

使用 `ssm-secure` 動態參考時，需要謹記幾個重要事項：

- CloudFormation 無法從其他 AWS 帳戶中存取 Parameter Store 值。
- CloudFormation 在動態參考中不支援使用 Systems Manager 參數標籤或公有參數。
- 在 `cn-north-1` 和 `cn-northwest-1` 區域中，Systems Manager 不支援安全字串。
- 自訂資源中目前不支援安全值的動態參考，例如 `ssm-secure`。
- 如果 CloudFormation 需要復原堆疊更新，且之前指定的安全字串參數版本不再可用，則復原操作將失敗。在這種情況下，您有兩個選項：
 - 使用 `CONTINUE_UPDATE_ROLLBACK` 略過資源。
 - 在 Systems Manager 參數存放區中重新建立安全字串參數，並更新它，直到參數版本達到範本中使用的版本。然後，在不跳過資源的情況下使用 `CONTINUE_UPDATE_ROLLBACK`。

對安全字串支援動態參數模式的資源

支援 `ssm-secure` 動態參考模式的資源包括：

資源	屬性類型	Properties
AWS::DirectoryService::MicrosoftAD		Password
AWS::DirectoryService::SimpleAD		Password
AWS::ElastiCache::ReplicationGroup		AuthToken
AWS::IAM::User	LoginProfile	Password
AWS::KinesisFirehose::DeliveryStream	RedshiftDestinationConfiguration	Password
AWS::OpsWorks::App	來源	Password

資源	屬性類型	Properties
AWS::OpsWorks::Stack	CustomCookbooksSource	Password
AWS::OpsWorks::Stack	RdsDbInstances	DbPassword
AWS::RDS::DBCluster		MasterUserPassword
AWS::RDS::DBInstance		MasterUserPassword
AWS::Redshift::Cluster		MasterUserPassword

參考模式

若要在 CloudFormation 範本中參考 Systems Manager Parameter Store 的安全字串值，請使用下列 `ssm-secure` 參考模式。

```
{{resolve:ssm-secure:parameter-name:version}}
```

針對 `parameter-name` 和 `version`，您的參考必須遵守下列規則表達式模式：

```
{{resolve:ssm-secure:[a-zA-Z0-9_.\-/]+(:\d+)?}}
```

parameter-name

參數存放區中的參數名稱。函數名稱區分大小寫。

必要。

version

整數，指定要使用的參數版本。如果您沒有指定確切的版本，CloudFormation 會在您建立或更新堆疊時使用最新版本的參數。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的[使用參數版本](#)。

選用。

範例

以下範例使用 `ssm-secure` 動態參考，將 IAM 使用者的密碼設為存放在 Parameter Store 的安全字串。依規定，CloudFormation 會使用 `IAMUserPassword` 參數的 `10` 版本，用於堆疊和變更集操作。

JSON

```
"MyIAMUser": {
  "Type": "AWS::IAM::User",
  "Properties": {
    "UserName": "MyUserName",
    "LoginProfile": {
      "Password": "{{resolve:ssm-secure:IAMUserPassword:10}}"
    }
  }
}
```

YAML

```
MyIAMUser:
  Type: AWS::IAM::User
  Properties:
    UserName: 'MyUserName'
    LoginProfile:
      Password: '{{resolve:ssm-secure:IAMUserPassword:10}}'
```

從 Secrets Manager 中取得秘密或秘密值

Secrets Manager 是一項服務，可讓您安全地存放和管理秘密，例如資料庫憑證、密碼和第三方 API 金鑰。您可以使用 Secrets Manager 集中存放或控制對這些秘密的存取，因此您可以將程式碼中的硬式編碼憑證 (包括密碼)，取代為對 Secrets Manager 的 API 呼叫，以程式設計方式擷取秘密。如需詳細資訊，請參閱 AWS Secrets Manager 《使用者指南》中的 [什麼是 AWS Secrets Manager？](#)。

若要在 CloudFormation 範本內使用存放於 Secrets Manager 中的全部秘密或秘密值，您可使用 `secretsmanager` 動態參考。

最佳實務

在 CloudFormation 範本中使用 Secrets Manager 動態參考時，請遵循下列最佳實務：

- 使用 CloudFormation 範本的無版本參考 – 將憑證儲存在 Secrets Manager 中，並使用動態參考，而不指定 `version-stage` 或 `version-id` 參數，以支援適當的秘密輪換工作流程。

- 利用自動輪換 – 搭配使用 Secrets Manager 的自動輪換功能與無版本動態參考，進行憑證管理。這可確保定期更新憑證，而不需要變更範本。如需詳細資訊，請參閱[輪換 AWS Secrets Manager 秘密](#)。
- 謹慎使用版本控制的參考 – 僅針對特定案例指定明確的 `version-stage` 或 `version-id` 參數，例如測試或復原情況。

考量事項

使用 `secretsmanager` 動態參考時，請注意以下重要考量：

- CloudFormation 不會追蹤先前部署中使用的秘密版本。在實作動態參考之前，請仔細規劃您的秘密管理策略。盡可能使用無版本參考，以利用自動秘密輪換。在變更動態參考組態時 (例如從未版本化動態參考轉換為版本化動態參考時) 監控和驗證資源更新，反之亦然。
- 僅更新 Secrets Manager 中的秘密值不會自動導致 CloudFormation 擷取新值。CloudFormation 只會在修改包含動態參考之資源的資源建立或更新期間擷取秘密值。

例如，假設範本包含 [AWS::RDS::DBInstance](#) 資源，其中 `MasterPassword` 屬性設定為 Secrets Manager 動態參考。從此範本建立堆疊後，可以在 Secrets Manager 中更新秘密的值。不過，`MasterPassword` 屬性會保留舊密碼值。

若要套用新的秘密值，需要修改 CloudFormation 範本中的 `AWS::RDS::DBInstance` 資源，並執行堆疊更新。

為了避免未來出現此手動程序，請考慮使用 Secrets Manager 自動輪換秘密。

- 自訂資源中目前不支援安全值的動態參考，例如 `secretsmanager`。
- 所有資源屬性中可以使用 `secretsmanager` 動態參考。使用 `secretsmanager` 動態參考表明 Secrets Manager 和 CloudFormation Logs 都不應存留任何已解析的秘密值。但是，秘密值在服務的資源中使用時，秘密值就可能出現在此服務中。檢閱您的使用情況，以避免洩露秘密資料。

許可

若要指定存放在 Secrets Manager 中的秘密，您必須擁有許可才能呼叫 [GetSecretValue](#) 以取得秘密。

參考模式

若要在 CloudFormation 範本中參考 Secrets Manager 秘密，請使用下列 `secretsmanager` 參考模式。


```
{{resolve:secretsmanager:secret-id:secret-string:json-key:version-stage:version-id}}
```

secret-id

秘密的名稱或 ARN。

若要存取 中的秘密 AWS 帳戶，您只需指定秘密名稱。若要存取不同 中的秘密 AWS 帳戶，請指定秘密的完整 ARN。

必要。

secret-string

唯一支援的值為 SecretString。預設值為 SecretString。

json-key

您要擷取值的鍵值組的索引鍵名稱。如果您不指定 json-key，CloudFormation 會擷取整個秘密文字。

此區段可能不可包含冒號字元 (:)。

version-stage

要使用的秘密版本之預備標籤。Secrets Manager 在輪換程序期間使用預備標籤來追蹤不同版本。如果您使用 version-stage，請不要指定 version-id。如果您未指定 version-stage 或 version-id，則預設為 AWSCURRENT 版本。

此區段可能不可包含冒號字元 (:)。

version-id

您要使用的秘密版本的唯一識別碼。如果您指定 version-id，則請不要指定 version-stage。如果您未指定 version-stage 或 version-id，則預設為 AWSCURRENT 版本。

此區段可能不可包含冒號字元 (:)。

範例

主題

- [從秘密中擷取使用者名稱和密碼值](#)
- [擷取整個 SecretString](#)
- [從特定版本的秘密中擷取值](#)

- [從另一個擷取秘密 AWS 帳戶](#)

從秘密中擷取使用者名稱和密碼值

以下 [AWS::RDS::DBInstance](#) 範例會擷取存放在 *MySecret* 秘密中的使用者名稱和密碼值。此範例顯示無版本動態參考的建議模式，它會自動使用 AWSCURRENT 版本並支援 Secrets Manager 輪換工作流程，而不需要變更範本。

JSON

```
{
  "MyRDSInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName": "MyRDSInstance",
      "AllocatedStorage": "20",
      "DBInstanceClass": "db.t2.micro",
      "Engine": "mysql",
      "MasterUsername":
        "{{resolve:secretsmanager:MySecret:SecretString:username}}",
      "MasterUserPassword":
        "{{resolve:secretsmanager:MySecret:SecretString:password}}"
    }
  }
}
```

YAML

```
MyRDSInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName: MyRDSInstance
    AllocatedStorage: '20'
    DBInstanceClass: db.t2.micro
    Engine: mysql
    MasterUsername: '{{resolve:secretsmanager:MySecret:SecretString:username}}'
    MasterUserPassword: '{{resolve:secretsmanager:MySecret:SecretString:password}}'
```

擷取整個 SecretString

下列動態參考會擷取 *MySecret* 的 SecretString。

```
{{resolve:secretsmanager:MySecret}}
```

或使用：

```
{{resolve:secretsmanager:MySecret:::}}
```

從特定版本的秘密中擷取值

下列動態參考會擷取 *MySecret* 的 *AWSPREVIOUS* 版本的 *password* 值。

```
{{resolve:secretsmanager:MySecret:SecretString:password:AWSPREVIOUS}}
```

從另一個 擷取秘密 AWS 帳戶

下列動態參考會擷取另一個 AWS 帳戶中 *MySecret* 的 *SecretString*。您必須指定完整的秘密 ARN，才能存取另一個秘密 AWS 帳戶。

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret}}
```

下列動態參考會擷取另一個 AWS 帳戶中 *MySecret* 的 *password* 值。您必須指定完整的秘密 ARN，才能存取另一個秘密 AWS 帳戶。

```
{{resolve:secretsmanager:arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret:SecretString:password}}
```

使用虛擬參數取得 AWS 值

虛擬參數是內建變數，可讓您存取重要的 AWS 環境資訊，例如帳戶 IDs、區域名稱，以及可在部署或環境之間變更的堆疊詳細資訊。

您可以使用虛擬參數，而不是硬式編碼值，讓您的範本更方便攜帶，並更輕鬆地在不同 AWS 帳戶 和 區域重複使用。

語法

您可以透過 `Ref` 內建函數或 `Fn::Sub` 內建函數參考虛擬參數。

Ref

Ref 內建函數使用下列一般語法。如需詳細資訊，請參閱 [Ref](#)。

JSON

```
{ "Ref" : "AWS::PseudoParameter" }
```

YAML

```
!Ref AWS::PseudoParameter
```

Fn::Sub

Fn::Sub 內建函數使用不同格式，在虛擬參數前後加上 `${}` 語法。如需詳細資訊，請參閱 [Fn::Sub](#)。

JSON

```
{ "Fn::Sub" : "${AWS::PseudoParameter}" }
```

YAML

```
!Sub '${AWS::PseudoParameter}'
```

可用的虛擬參數

AWS::AccountId

傳回要在其中建立堆疊之帳戶的 AWS 帳戶 ID，例如 123456789012。

此虛擬參數常用於定義 IAM 角色、原則及其他涉及帳戶專屬 ARN 的資源原則。

AWS::NotificationARNs

傳回接收堆疊事件通知的 Amazon SNS 主題對應的 Amazon Resource Name (ARN) 清單。您可以在建立 AWS CLI 或更新堆疊時，透過 `--notification-arns` 選項或透過 主控台 指定這些 ARNs。

不同於其他傳回單一數值的虛擬參數，`AWS::NotificationARNs` 會傳回 ARN 清單。要存取清單中的特定 ARN，請使用 `Fn::Select` 內建函數。如需詳細資訊，請參閱[Fn::Select](#)。

AWS::NoValue

若在 `Fn::If` 內部函數中將此參數指定為傳回值，則系統會移除對應的資源屬性。如需詳細資訊，請參閱[Fn::If](#)。

此虛擬參數在建立僅需於特定條件下納入的條件式資源屬性時特別實用。

AWS::Partition

傳回資源所在的分割區。對於標準 AWS 區域，分割區為 `aws`。如果資源處於其他分割區，則會傳回 `aws-partitionname` 分割區。例如，中國（北京和寧夏）區域中資源的分割區為 `aws-cn`，AWS GovCloud（美國西部）區域中資源的分割區為 `aws-us-gov`。

分區是資源 ARN 的一部分。使用 `AWS::Partition` 可確保您的範本在不同 AWS 分割區之間正確運作。

AWS::Region

傳回建立包容性資源區域的代表字串，例如：`us-west-2`。

這是最常用的虛擬參數之一，因為它允許範本在不修改 AWS 區域的情況下適應不同的參數。

AWS::StackId

傳回堆疊的 ID (ARN)，例如 `arn:aws:cloudformation:us-west-2:123456789012:stack/teststack/51af3dc0-da77-11e4-872e-1234567db123`。

AWS::StackName

傳回堆疊的名稱，例如 `teststack`。

堆疊名稱常用於建立獨特的資源名稱，這類名稱易於識別為屬於特定堆疊。

AWS::URLSuffix

在部署 AWS 區域堆疊的中，傳回 AWS 網域的尾碼。尾碼通常為 `amazonaws.com`，但對於中國（北京）區域，尾碼為 `amazonaws.com.cn`。

此參數在建構 AWS 服務端點URLs 時特別有用。

範例

主題

- [基本使用](#)
- [使用 AWS::NotificationARNs](#)
- [具備 AWS::NoValue 的條件屬性](#)

基本使用

下列範例建立兩個資源：一個 Amazon SNS 主題，以及一個向該主題傳送通知的 CloudWatch 警示。它們使用 `AWS::StackName`、`AWS::Region` 和 `AWS::AccountId` 將堆疊名稱、目前 AWS 區域和帳戶 ID 動態插入資源名稱、描述和 ARNs。

JSON

```
{
  "Resources": {
    "MyNotificationTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "DisplayName": { "Fn::Sub": "Notifications for ${AWS::StackName}" }
      }
    },
    "CPUALarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": { "Fn::Sub": "Alarm for high CPU in ${AWS::Region}" },
        "AlarmName": { "Fn::Sub": "${AWS::StackName}-HighCPUALarm" },
        "MetricName": "CPUUtilization",
        "Namespace": "AWS/EC2",
        "Statistic": "Average",
        "Period": 300,
        "EvaluationPeriods": 1,
        "Threshold": 80,
        "ComparisonOperator": "GreaterThanThreshold",
        "AlarmActions": [{ "Fn::Sub": "arn:aws:sns:${AWS::Region}:${AWS::AccountId}:${MyNotificationTopic}" }]
      }
    }
  }
}
```

```
    }
  }
}
```

YAML

```
Resources:
  MyNotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: !Sub Notifications for ${AWS::StackName}
  CPUAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: !Sub Alarm for high CPU in ${AWS::Region}
      AlarmName: !Sub ${AWS::StackName}-HighCPUAlarm
      MetricName: CPUUtilization
      Namespace: AWS/EC2
      Statistic: Average
      Period: 300
      EvaluationPeriods: 1
      Threshold: 80
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - !Sub arn:aws:sns:${AWS::Region}:${AWS::AccountId}:${MyNotificationTopic}
```

使用 AWS::NotificationARNs

下列範例設定 Auto Scaling 群組，以便在執行個體啟動事件及啟動錯誤時傳送通知。此設定使用 `AWS::NotificationARNs` 虛擬參數，該參數提供建立堆疊時指定的 Amazon SNS 主題 ARN 清單。Fn::Select 函數會從該清單中選擇第一個 ARN。

JSON

```
"myASG": {
  "Type": "AWS::AutoScaling::AutoScalingGroup",
  "Properties": {
    "LaunchTemplate": {
      "LaunchTemplateId": { "Ref": "myLaunchTemplate" },
      "Version": { "Fn::GetAtt": [ "myLaunchTemplate", "LatestVersionNumber" ] }
    },
    "MaxSize": "1",
    "MinSize": "1",
```

```

    "VPCZoneIdentifier": [
        "subnetIdAz1",
        "subnetIdAz2",
        "subnetIdAz3"
    ],
    "NotificationConfigurations" : [{
        "TopicARN" : { "Fn::Select" : [ "0", { "Ref" : "AWS::NotificationARNs" } ] },
        "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH",
"autoscaling:EC2_INSTANCE_LAUNCH_ERROR" ]
    }]
}
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MinSize: '1'
    MaxSize: '1'
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    NotificationConfigurations:
      - TopicARN:
          Fn::Select:
            - '0'
            - Ref: AWS::NotificationARNs
        NotificationTypes:
          - autoscaling:EC2_INSTANCE_LAUNCH
          - autoscaling:EC2_INSTANCE_LAUNCH_ERROR

```

具備 AWS::NoValue 的條件屬性

下列範例會建立 Amazon RDS 資料庫執行個體，僅在提供快照 ID 時才使用快照。若 UseDBSnapshot 條件評估為 true，CloudFormation 會對 DBSnapshotIdentifier 屬性使用 DBSnapshotName 參數值。若條件評估為 false，CloudFormation 即會移除 DBSnapshotIdentifier 屬性。

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
        "UseDBSnapshot",
        { "Ref" : "DBSnapshotName" },
        { "Ref" : "AWS::NoValue" }
      ]
    }
  }
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    EngineVersion: '5.5'
    MasterUsername:
      Ref: DBUser
    MasterUserPassword:
      Ref: DBPassword
    DBParameterGroupName:
      Ref: MyRDSParamGroup
    DBSnapshotIdentifier:
      Fn::If:
        - UseDBSnapshot
        - Ref: DBSnapshotName
        - Ref: AWS::NoValue
```

從部署的 CloudFormation 堆疊取得匯出的輸出

當您在相同 AWS 帳戶和區域中有多個堆疊時，您可能想要在它們之間共用資訊。當一個堆疊需要使用另一個堆疊建立的資源時，這項功能相當實用。

例如，您可能有一個堆疊專門為 Web 伺服器建立網路資源，如子網路和安全群組。而建立實際 Web 伺服器的其他堆疊，就能使用第一個堆疊建立的網路資源。您不必在堆疊範本中硬式編碼資源 ID，也不需要以輸入參數形式傳遞 ID。

若要在堆疊之間共用資訊，可從一個堆疊匯出輸出值，再將其匯入另一個堆疊。以下是其運作方式：

1. 在第一個堆疊的範本 (例如，聯網堆疊) 中，可使用 Outputs 區段的 Export 欄位來定義匯出的特定值。如需詳細資訊，請參閱[CloudFormation 範本 Outputs 語法](#)。
2. 當您建立或更新該堆疊時，CloudFormation 會匯出輸出值，使其可供相同 AWS 帳戶和區域中的其他堆疊使用。
3. 在另一個堆疊的範本中，您可以使用 [Fn::ImportValue](#) 函數從第一個堆疊匯入匯出的值。
4. 建立或更新第二個堆疊 (例如 Web 伺服器堆疊) 時，CloudFormation 會自動從第一個堆疊擷取已匯出的值並套用。

如需演練和範例範本，請參閱[參考在另一個 CloudFormation 堆疊中的資源輸出](#)。

匯出堆疊輸出值與使用巢狀堆疊的比較

巢狀堆疊是指藉由使用 `AWS::CloudFormation::Stack` 資源在另一個堆疊中建立的堆疊。使用巢狀堆疊時，您可以透過單一堆疊來部署和管理所有資源。您可以將巢狀堆疊群組中某個堆疊的輸出，作為群組中其他堆疊的輸入。這與匯出值的情況不同。

如果您想要將資訊的共享隔離在巢狀堆疊群組內，則建議您使用巢狀堆疊。若要與其他堆疊 (而不只是在巢狀堆疊的群組內) 共享資訊，則匯出值。例如，您可以建立含子網路的單一堆疊，然後匯出其 ID。其他堆疊可透過匯入該 ID 來使用該子網路。每個堆疊無需單獨建立子網路。只要有堆疊匯入子網路 ID，您就無法變更或刪除該 ID。

如需巢狀堆疊的詳細資訊，請參閱[運用巢狀堆疊，將範本分割成可重複使用的部分](#)。

考量事項

下列限制適用於跨堆疊參考：

- 對於每個名稱 AWS 帳戶，Export 名稱在區域中必須是唯一的。

- 您無法跨區域建立跨堆疊參考。您可以使用內部函數 `Fn::ImportValue`，僅匯入已在同一區域內匯出的值。
- 對於輸出，`Export` 的 `Name` 屬性值無法使用依賴某個資源的 `Ref` 或者 `GetAtt` 函數。

同樣，`ImportValue` 函數不能包含依賴某個資源的 `Ref` 或 `GetAtt` 函數。

- 當其他堆疊匯入輸出值之後，您就無法刪除匯出輸出值的堆疊，也無法修改匯出的輸出值。您必須先移除所有匯入，之後才可以刪除匯出堆疊或修改輸出值。

列出已匯出的輸出值

若需檢視堆疊已匯出的輸出值，可使用下列其中一種方法：

列出已匯出的輸出值 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在左側導覽窗格中，選擇匯出。

列出已匯出的輸出值 (AWS CLI)

使用下列 [list-exports](#) 命令。將 `us-east-1` 取代為您的 AWS 區域。

```
aws cloudformation list-exports --region us-east-1
```

以下為範例輸出。

```
{
  "Exports": [
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-a",
      "Value": "subnet-07b410xmplddcfa03"
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-subnet-b",
      "Value": "subnet-075ed3xmplebd2fb1"
    }
  ]
}
```

```
    },
    {
      "ExportingStackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
      "Name": "private-vpc-vpcid",
      "Value": "vpc-011d7xmpl1100e9841"
    }
  ]
}
```

CloudFormation 會針對目前匯出的區域和堆疊，顯示已匯出的輸出名稱和值。若要使用其他堆疊範本中的已匯出輸出值，請使用匯出名稱和 `Fn::ImportValue` 函數進行參考。

列出匯入匯出的輸出值的堆疊

若要刪除或變更匯出的輸出值，必須先找出哪些堆疊正在匯入輸出值。

若需檢視匯入已匯出輸出值的堆疊，可使用下列其中一種方法：

列出匯入匯出的輸出值的堆疊 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在左側導覽窗格中，選擇匯出。
3. 若要查看匯入指定匯出值的堆疊，請選擇該匯出值的 Export Name (匯出名稱)。CloudFormation 會顯示匯出詳細資訊頁面，列出匯入該值的所有堆疊。

列出匯入匯出的輸出值的堆疊 (AWS CLI)

使用 [list-imports](#) 命令。將 *us-east-1* 取代為您的 AWS 區域，並將 *private-vpc-vpcid* 取代為匯出輸出值的名稱。

```
aws cloudformation list-imports --region us-east-1 \
  --export-name private-vpc-vpcid
```

CloudFormation 傳回匯入該值的堆疊清單。

```
{
  "Imports": [
    "my-app-stack"
  ]
}
```

```
}
```

確認哪些堆疊正在匯入特定已匯出值後，您需要修改這些堆疊，移除參考該輸出值的 `Fn::ImportValue` 函數。您必須移除所有參考匯出的輸出值的匯入，才能刪除或修改匯出的輸出值。

使用 CloudFormation 提供的參數類型，在執行時期指定現有資源

建立範本時，您可以使用 CloudFormation 提供的專用參數類型，建立參數，要求使用者輸入現有 AWS 資源或 Systems Manager 參數的識別符。

主題

- [概觀](#)
- [範例](#)
- [考量事項](#)
- [支援 AWS 的特定參數類型](#)
- [支援的 Systems Manager 參數類型](#)
- [不支援的 Systems Manager 參數類型](#)

概觀

在 CloudFormation 中，可以透過在堆疊建立或更新期間提供輸入值，使用參數來自訂堆疊。此功能使得您的範本在不同案例中可重複使用且靈活。

在 CloudFormation 範本的 `Parameters` 區段中定義參數。每個參數都有名稱和類型，並且可以有額外的設定，例如預設值和允許的值。如需詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

參數類型會決定參數可以接受的輸入值類型。例如，`Number` 只接受數值，而 `String` 接受文字輸入。

CloudFormation 提供數種額外的參數類型，您可以在範本中用來參考現有的 AWS 資源和 Systems Manager 參數。

這些參數類型分為兩個類別：

- **AWS 特定參數類型** – CloudFormation 提供一組參數類型，可協助在建立或更新堆疊時擷取無效的值。當您使用這些參數類型時，使用範本的任何人都必須從他們建立堆疊所在的 AWS 帳戶和區域指定有效值。

如果他們使用 AWS 管理主控台，CloudFormation 會提供其帳戶和區域中現有值的預先填入清單。這樣，使用者不需要記住和正確輸入特定名稱或 ID。相反，他們只需從下拉式清單中選取值。在某些情況下，他們甚至可以依 ID、名稱或 Name 標籤值來搜尋值。

- **Systems Manager 參數類型** – CloudFormation 也提供對應於 Systems Manager 參數存放區中現有參數的參數類型。當您使用這些參數類型時，使用範本的任何人都必須指定參數存放區金鑰作為 Systems Manager 參數類型的值，然後 CloudFormation 會從參數存放區擷取最新的值，以便在其堆疊中使用。當您需要使用新的屬性值 (例如新的 Amazon Machine Image (AMI) ID) 來頻繁更新應用程式時，這會很有用。如需參數存放區的相關資訊，請參閱 [Systems Manager 參數存放區](#)。

在 Parameters 區段中定義參數後，可以使用 Ref 函數在整個 CloudFormation 範本中參考參數值。

範例

下列範例顯示使用下列參數類型的範本。

- `AWS::EC2::VPC::Id`
- `AWS::EC2::Subnet::Id`
- `AWS::EC2::KeyPair::KeyName`
- `AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>`

若要從此範本中建立堆疊，必須在您的帳戶中指定現有的 VPC ID、子網路 ID 和金鑰對名稱。也可以指定參考所需 AMI ID 的現有參數存放區金鑰，或保留預設值 `/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2`。此公有參數是最新 Amazon Linux 2 AMI 的區域 AMI ID 別名。如需公有參數的詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 [探索參數存放區中的公有參數](#)。

JSON

```
{
  "Parameters": {
    "VpcId": {
      "Description": "ID of an existing Virtual Private Cloud (VPC).",
      "Type": "AWS::EC2::VPC::Id"
    },
    "PublicSubnetId": {
      "Description": "ID of an existing public subnet within the specified VPC.",
      "Type": "AWS::EC2::Subnet::Id"
    }
  }
}
```

```

    },
    "KeyName": {
      "Description": "Name of an existing EC2 key pair to enable SSH access to
the instance.",
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
    "AMIId": {
      "Description": "Name of a Parameter Store parameter that stores the ID of
the Amazon Machine Image (AMI).",
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
    }
  },
  "Resources": {
    "InstanceSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Enable SSH access via port 22",
        "VpcId": { "Ref": "VpcId" },
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    },
    "Ec2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "KeyName": { "Ref": "KeyName" },
        "ImageId": { "Ref": "AMIId" },
        "NetworkInterfaces": [
          {
            "AssociatePublicIpAddress": "true",
            "DeviceIndex": "0",
            "SubnetId": { "Ref": "PublicSubnetId" },
            "GroupSet": [{ "Ref": "InstanceSecurityGroup" }]
          }
        ]
      }
    }
  }
}

```

```

    },
    "Outputs": {
      "InstanceId": {
        "Value": { "Ref": "Ec2Instance" }
      }
    }
  }
}

```

YAML

Parameters:

VpcId:

Description: ID of an existing Virtual Private Cloud (VPC).

Type: AWS::EC2::VPC::Id

PublicSubnetId:

Description: ID of an existing public subnet within the specified VPC.

Type: AWS::EC2::Subnet::Id

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance.

Type: AWS::EC2::KeyPair::KeyName

AMIId:

Description: Name of a Parameter Store parameter that stores the ID of the Amazon Machine Image (AMI).

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:

InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Enable SSH access via port 22

VpcId: !Ref VpcId

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

Ec2Instance:

Type: AWS::EC2::Instance

Properties:

KeyName: !Ref KeyName

ImageId: !Ref AMIId

NetworkInterfaces:

- AssociatePublicIpAddress: "true"


```
    DeviceIndex: "0"
    SubnetId: !Ref PublicSubnetId
    GroupSet:
      - !Ref InstanceSecurityGroup
Outputs:
  InstanceId:
    Value: !Ref Ec2Instance
```

AWS CLI 命令來建立堆疊

下列 [create-stack](#) 命令會根據範例範本來建立堆疊。

```
aws cloudformation create-stack --stack-name MyStack \
  --template-body file://sampletemplate.json \
  --parameters \
    ParameterKey="VpcId",ParameterValue="vpc-a123baa3" \
    ParameterKey="PublicSubnetId",ParameterValue="subnet-123a351e" \
    ParameterKey="KeyName",ParameterValue="MyKeyName" \
    ParameterKey="AMIId",ParameterValue="MyParameterKey"
```

若要使用接受字串清單的參數類型，例如 `List<AWS::EC2::Subnet::Id>`，您必須使用雙反斜線來轉義 `ParameterValue` 內的逗號，如下列範例所示。

```
--parameters ParameterKey="SubnetIDs",ParameterValue="subnet-5ea0c127\\, subnet-6194ea3b\\, subnet-c87f2be0"
```

考量事項

強烈建議您使用動態參考來限制存取敏感組態定義，例如第三方憑證。如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

如果您想要允許範本使用者指定來自不同的值 AWS 帳戶，請勿使用 AWS 特定的參數類型。反之，請定義類型 `String` 或 `CommaDelimitedList` 的參數。

使用 Systems Manager 參數類型時需要記住幾件事：

- 可以在主控台中或透過執行 [describe-stacks](#) 或 [describe-change-set](#)，查看堆疊的參數索引標籤中的已解析參數值。請謹記，這些值是在建立或更新堆疊時所設定，因此可能會與參數存放區中的最新值不同。
- 對於堆疊更新，當您使用使用現有值選項 (或 `UsePreviousValue` 設定為 `true`) 時，這表示您想要繼續使用相同的參數存放區金鑰，而不是其值。CloudFormation 始終會擷取最新的值。

- 如果您指定任何允許的值或其他限制條件，CloudFormation 會根據您指定的參數索引鍵進行驗證，但不會驗證其值。您應該驗證參數存放區本身中的值。
- 當您建立或更新堆疊並建立變更集時，CloudFormation 會使用參數存放區中當時存在的任何值。如果指定的參數不存在於發起人的參數存放區中 AWS 帳戶，CloudFormation 會傳回驗證錯誤。
- 當您執行變更集時，CloudFormation 會使用變更集中所指定的值。您應該先檢閱這些值，再執行變更集，因為它們可能會在您建立變更集與執行變更集之間於參數存放區中變更。
- 對於存放在相同中的參數存放區參數 AWS 帳戶，您必須提供參數名稱。對於另一個 AWS 帳戶共用的參數存放區參數，必須提供完整的參數 ARN。

支援 AWS 的特定參數類型

CloudFormation AWS 支援下列特定類型：

`AWS::EC2::AvailabilityZone::Name`

可用區域，例如 `us-west-2a`。

`AWS::EC2::Image::Id`

Amazon EC2 映像 ID，例如 `ami-0ff8a91507f77f867`。請注意，CloudFormation 主控台不會顯示此參數類型的下拉式值清單。

`AWS::EC2::Instance::Id`

Amazon EC2 執行個體 ID，例如 `i-1e731a32`。

`AWS::EC2::KeyPair::KeyName`

Amazon EC2 金鑰對名稱。

`AWS::EC2::SecurityGroup::GroupName`

預設 VPC 安全群組名稱，例如 `my-sg-abc`。

`AWS::EC2::SecurityGroup::Id`

安全群組 ID，例如 `sg-a123fd85`。

`AWS::EC2::Subnet::Id`

子網路 ID，例如 `subnet-123a351e`。

`AWS::EC2::Volume::Id`

Amazon EBS 磁碟區 ID，例如 `vol-3cdd3f56`。

`AWS::EC2::VPC::Id`

VPC ID，例如 `vpc-a123baa3`。

`AWS::Route53::HostedZone::Id`

Amazon Route 53 託管區域 ID，例如 `Z23YXV40VPL04A`。

`List<AWS::EC2::AvailabilityZone::Name>`

區域的可用區域陣列，例如 `us-west-2a`，`us-west-2b`。

`List<AWS::EC2::Image::Id>`

Amazon EC2 映像 ID 陣列，例如 `ami-0ff8a91507f77f867`，`ami-0a584ac55a7631c0c`。請注意，CloudFormation 主控台不會顯示此參數類型的下拉式值清單。

`List<AWS::EC2::Instance::Id>`

Amazon EC2 執行個體 ID 陣列，例如 `i-1e731a32`，`i-1e731a34`。

`List<AWS::EC2::SecurityGroup::GroupName>`

預設 VPC 安全群組名稱陣列，例如 `my-sg-abc`，`my-sg-def`。

`List<AWS::EC2::SecurityGroup::Id>`

安全群組 ID 陣列，例如 `sg-a123fd85`，`sg-b456fd85`。

`List<AWS::EC2::Subnet::Id>`

子網路 ID 陣列，例如 `subnet-123a351e`，`subnet-456b351e`。

`List<AWS::EC2::Volume::Id>`

Amazon EBS 磁碟區 ID 陣列，例如 `vol-3cdd3f56`，`vol-4cdd3f56`。

`List<AWS::EC2::VPC::Id>`

VPC ID 陣列，例如 `vpc-a123baa3`，`vpc-b456baa3`。

`List<AWS::Route53::HostedZone::Id>`

Amazon Route 53 託管區域 ID 陣列，例如 `Z23YXV40VPL04A`，`Z23YXV40VPL04B`。

支援的 Systems Manager 參數類型

CloudFormation 支援下列 Systems Manager 參數類型：

`AWS::SSM::Parameter::Name`

Systems Manager 參數索引鍵的名稱。僅使用此參數類型來檢查必要的參數是否存在。CloudFormation 不會擷取與參數相關聯的實際值。

`AWS::SSM::Parameter::Value<String>`

值為字串的 Systems Manager 參數。這對應至參數存放區中的 String 參數類型。

`AWS::SSM::Parameter::Value<List<String>>` 或

`AWS::SSM::Parameter::Value<CommaDelimitedList>`

值為字串清單的 Systems Manager 參數。這對應至參數存放區中的 StringList 參數類型。

`AWS::SSM::Parameter::Value<AWS-specific parameter type>`

值為 AWS 特定參數類型的 Systems Manager 參數。

例如，以下指定 `AWS::EC2::KeyPair::KeyName` 類型：

- `AWS::SSM::Parameter::Value<AWS::EC2::KeyPair::KeyName>`

`AWS::SSM::Parameter::Value<List<AWS-specific parameter type>>`

Systems Manager 參數，其值是 AWS 特定參數類型的清單。

例如，以下指定 `AWS::EC2::KeyPair::KeyName` 類型清單：

- `AWS::SSM::Parameter::Value<List<AWS::EC2::KeyPair::KeyName>>`

不支援的 Systems Manager 參數類型

CloudFormation 不支援下列 Systems Manager 參數類型：

- Systems Manager 參數類型清單 — 例如：`List<AWS::SSM::Parameter::Value<String>>`

此外，CloudFormation 不支援將範本參數定義為 SecureString Systems Manager 參數類型。不過，您可以將安全字串指定為某些資源的參數值。如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

CloudFormation 逐步解說

本文件提供一系列逐步解說，旨在讓您透過實際操作練習堆疊部署。

- [參考在另一個 CloudFormation 堆疊中的資源輸出](#) – 此逐步解說說明如何在一個 CloudFormation 堆疊中參考另一個堆疊的輸出內容。您可以在個別堆疊中建立相關 AWS 資源，以建立更模組化且可重複使用的範本，而不是將所有資源包含在單一堆疊中。
- [在 Amazon EC2 上部署應用程式](#) – 了解如何使用 CloudFormation 自動安裝、設定和啟動 Amazon EC2 執行個體上的應用程式。這樣可讓您輕鬆地重複部署和更新現有安裝，無需直接連接到執行個體。
- [更新 CloudFormation 堆疊](#) – 您可以使用 CloudFormation，逐步完成更新運作中堆疊的簡易程序。
- [建置擴展和負載平衡的應用程式](#) – 了解如何使用 CloudFormation 建立可擴展且負載平衡的應用程式。本逐步解說涵蓋建立 Auto Scaling 群組、負載平衡器及其他相關資源，以確保您的應用程式能應對變化的流量負載並保持高可用性。
- [在另一個中使用 VPC 的對等 AWS 帳戶](#) – 此演練會引導您完成在不同 VPCs 之間建立虛擬私有雲端 (VPC) 對等互連的程序 AWS 帳戶。VPC 對等互連可協助您在兩個 VPC 之間路由流量，並像存取同一網路中的資源一樣存取這些資源。
- [使用 CloudFormation 透過 CodeDeploy 進行 ECS 藍色/綠色部署](#) – 探索如何使用 CloudFormation 在 Amazon ECS 上執行 AWS CodeDeploy 藍/綠部署。藍綠部署是一種以最短的停機時間更新應用程式或服務的方式。

參考在另一個 CloudFormation 堆疊中的資源輸出

此逐步解說說明如何在一個 CloudFormation 堆疊中參考另一個堆疊的輸出內容，以建立更模組化且可重複使用的範本。

您可以在個別堆疊中建立相關 AWS 資源，而不是將所有資源包含在單一堆疊中。然後，您可以從其他堆疊參考必要的資源輸出。透過將跨堆疊參考限制在輸出，您可以控制堆疊中由其他堆疊參考的部分。

例如，您可能有一個使用 VPC、安全群組，以及公有 Web 應用程式子網路的網路堆疊，以及另一個公有 Web 應用程式堆疊。若要確保 Web 應用程式使用來自網路堆疊的安全群組和子網路，您可以建立跨堆疊參考，允許 Web 應用程式堆疊參考網路堆疊的資源輸出。使用跨堆疊參考，Web 應用程式堆疊的擁有者便不需要建立或維護網路規則或資產。

若要建立跨堆疊參考，請使用 `Export` 輸出欄位，標記要匯出的資源輸出值。然後，使用 `Fn::ImportValue` 內部函數匯入值。如需詳細資訊，請參閱[從部署的 CloudFormation 堆疊取得匯出的輸出](#)。

 Note

CloudFormation 是免費服務。不過，您需要按目前費率支付堆疊中包含 AWS 的資源費用。如需 AWS 定價的詳細資訊，請參閱[各項產品的詳細資訊頁面](#)。

主題

- [使用範例範本建立網路堆疊](#)
- [使用範例範本建立 Web 應用程式堆疊](#)
- [驗證堆疊是否按設計運作](#)
- [疑難排解 AMI 映射錯誤](#)
- [清除您的資源](#)

使用範例範本建立網路堆疊

在您開始此演練前，請確認您已有使用下列所有服務的 IAM 許可：Amazon VPC、Amazon EC2 和 CloudFormation。

網路堆疊包含您會在 Web 應用程式堆疊中使用的 VPC、安全群組和子網路。除了這些資源之外，網路堆疊也會建立網際網路閘道及路由表，以啟用公有存取。

您必須在您建立 Web 應用程式堆疊前建立此堆疊。若您先建立 Web 應用程式堆疊，它將不會擁有安全群組或子網路。

堆疊範本可在以下 URL 中找到：<https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template>。若要查看堆疊建立的資源，請選擇連結，開啟範本。在 Outputs 區段中，您可以查看範例範本匯出的網路資源。匯出資源的名稱會加上堆疊名稱的字首，以在您從其他堆疊匯出網路資源時進行區別。當使用者匯入網路資源時，他們便可以指定要匯入哪些堆疊的資源。

建立網路堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。
3. 選擇選擇現有範本，然後在指定範本區段中選擇 Amazon S3 URL。

4. 對於 Amazon S3 URL，貼上下列 URL：**`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleNetworkCrossStack.template`**。
5. 選擇下一步。
6. 針對 Stack name (堆疊名稱)，輸入 **SampleNetworkCrossStack**，然後選擇 Next (下一步)。

 Note

記錄此堆疊的名稱。當您在啟動 Web 應用程式堆疊時，您將會需要堆疊名稱。

7. 選擇下一步。針對此演練，您不需要新增標籤或指定進階設定。
8. 確認堆疊名稱和範本 URL 正確，然後選擇 Create stack (建立堆疊)。

CloudFormation 可能需要幾分鐘的時間來建立您的堆疊。等待所有資源成功建立，再繼續建立 Web 應用程式堆疊。

9. 若要監控進度，請檢視堆疊事件。如需詳細資訊，請參閱[監控堆疊進度](#)。

使用範例範本建立 Web 應用程式堆疊

Web 應用程式堆疊會建立使用網路堆疊之安全群組和子網路的 EC2 執行個體。

您必須在與網路堆疊 AWS 區域 相同的 中建立此堆疊。

堆疊範本可在以下 URL 中找到：<https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>。若要查看堆疊建立的資源，請選擇連結，開啟範本。在 Resources 區段中，檢視 EC2 執行個體的屬性。您可以使用 Fn::ImportValue 函數查看從另一個堆疊匯入網路資源的方式。

建立 Web 應用程式堆疊

1. 從堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。
2. 選擇選擇現有範本，然後在指定範本區段中選擇 Amazon S3 URL。
3. 對於 Amazon S3 URL，貼上下列 URL：**`https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template`**。
4. 選擇下一步。
5. 在 Stack Name (堆疊名稱) 中輸入 **SampleWebAppCrossStack**。在 Parameters (參數) 區段中，針對 NetworkStackName 參數使用預設值，然後選擇 Next (下一步)。

範例範本會使用參數值指定要從哪個堆疊匯入值。

6. 選擇下一步。針對此演練，您不需要新增標籤或指定進階設定。
7. 確認堆疊名稱和範本 URL 正確，然後選擇 Create stack (建立堆疊)。

CloudFormation 可能需要幾分鐘的時間來建立您的堆疊。

驗證堆疊是否按設計運作

在堆疊建立之後，檢視其資源並注意執行個體 ID。如需檢視堆疊資源的詳細資訊，請參閱[在 CloudFormation 主控台中檢視堆疊資訊](#)。

若要驗證執行個體的安全群組和子網路，請在 [Amazon EC2 主控台](#) 檢視執行個體的屬性。若執行個體使用來自 SampleNetworkCrossStack 堆疊的安全群組和子網路，表示您已成功建立跨堆疊參考。

使用主控台檢視堆疊輸出和範例網站 URL，驗證 Web 應用程式已在執行中。如需詳細資訊，請參閱[在 CloudFormation 主控台中檢視堆疊資訊](#)。

疑難排解 AMI 映射錯誤

若您收到錯誤，Template error: Unable to get mapping for AWSRegionArch2AMI::[region]::HVM64 範本不會包含您 AWS 區域的 AMI 映射。我們建議使用 Systems Manager 公有參數動態參考最新的 AMI，而非更新映射：

1. 從以下網址下載 SampleWebAppCrossStack 範本到您的本機電腦：<https://s3.amazonaws.com/cloudformation-examples/user-guide/cross-stack/SampleWebAppCrossStack.template>。
2. 刪除整個 AWSRegionArch2AMI 映射區段。
3. 新增下列 Systems Manager 參數：

```
"LatestAmiId": {
  "Description": "The latest Amazon Linux 2 AMI from the Parameter Store",
  "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
  "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
}
```

4. 取代現有的 ImageId 參考：

```
"ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" } ,
  "HVM64" ] },
```

取代為：


```
"ImageId": { "Ref": "LatestAmiId" },
```

此參數會自動解析為您部署堆疊所在區域的最新 Amazon Linux 2 AMI。

對於其他 Linux 發行版本，請使用對應的參數路徑。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的[探索參數存放區中的公有參數](#)。

5. 將修改的範本上傳至您帳戶中的 S3 儲存貯體：

```
aws s3 cp SampleWebAppCrossStack.template s3://amzn-s3-demo-bucket/
```

6. 建立堆疊時，指定您的 S3 範本 URL，而非範例 URL。

清除您的資源

為了確保您不需要為不必要的服務支付費用，請刪除堆疊。

刪除堆疊

1. 在 CloudFormation 主控台中，選擇 SampleWebAppCrossStack 堆疊。
2. 選擇 Actions (動作)，然後選擇 Delete stack (刪除堆疊)。
3. 在確認訊息中，選擇 Delete (刪除)。
4. 在堆疊刪除後，為 SampleNetworkCrossStack 堆疊重複相同的步驟。

Note

等到 CloudFormation 完全刪除 SampleWebAppCrossStack 堆疊。若 EC2 執行個體仍在 VPC 中執行，CloudFormation 不會刪除 SampleNetworkCrossStack 堆疊中的 VPC。

在 Amazon EC2 上部署應用程式

可以使用 CloudFormation 自動安裝、設定和啟動 Amazon EC2 執行個體上的應用程式。這樣可讓您輕鬆地重複部署和更新現有安裝，無需直接連接到執行個體，為您節省了許多時間和精力。

CloudFormation 包含一組以 ccloud-init 為基礎的協助程式指令碼 (cfn-init、cfn-signal、cfn-get-metadata 和 cfn-hup)。您從 CloudFormation 範本呼叫這些協助程式指令

碼，在相同範本中的 Amazon EC2 執行個體上安裝、設定和更新應用程式。如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [CloudFormation 協助程式指令碼參考](#)。

在[入門教學課程](#)中，搭配使用 UserData 與基礎 bash 指令碼來建立簡單的 Web 伺服器。這僅適用於簡單的「Hello World」頁面，真正的應用程式通常需要更複雜的組態，包括：

- 以正確的順序安裝多個軟體套件。
- 使用特定內容建立的複雜組態檔案。
- 服務已啟動並設定為自動執行。
- 設定程序的錯誤處理和驗證。

相較於 UserData 中的基礎 bash 指令碼，CloudFormation 的協助程式指令碼可提供更強大且可維護的方式來設定 EC2 執行個體。cfn-init 協助程式指令碼會從範本的中繼資料中讀取組態資料，並以系統化的方式將其套用至執行個體。

在本教學課程中，您將了解如何使用 cfn-init 協助程式指令碼並監控引導程序。

Note

CloudFormation 是免費的，但需要為您建立的 Amazon EC2 資源付費。不過，如果您是新手 AWS，則可以利用 [免費方案](#)，在此學習過程中將成本降至最低或消除。

主題

- [先決條件](#)
- [了解引導概念](#)
- [從簡單的引導範例開始](#)
- [新增檔案和命令](#)
- [新增網路安全](#)
- [完整的引導範本](#)
- [使用主控台建立堆疊](#)
- [監控引導程序](#)
- [測試引導的 Web 伺服器](#)
- [對引導問題進行疑難排解](#)
- [清除資源](#)

- [後續步驟](#)

先決條件

- 您必須已完成 [建立您的第一個堆疊](#) 教學課程，或具有 CloudFormation 基礎知識的同等經驗。
- 您必須具有具有 IAM 使用者或角色 AWS 帳戶 的 存取權，該角色具有使用 Amazon EC2 和 CloudFormation 的許可，或管理使用者存取權。
- 必須擁有可存取網際網路的虛擬私有雲端 (VPC)。此教學範本需要預設 VPC，它會自動隨附較新的 AWS 帳戶。如果您沒有預設 VPC，或已刪除它，請參閱 [建立您的第一個堆疊](#) 教學課程中的疑難排解一節，了解替代解決方案。

了解引導概念

讓我們先了解讓引導運作的關鍵概念，然後再建立範本。

cfn-init helper 指令碼

CloudFormation 提供 Python 協助程式指令碼，您可以使用其在 Amazon EC2 執行個體上安裝軟體及啟動服務。cfn-init 指令碼會從範本中讀取資源中繼資料，並將組態套用至執行個體。

程序的運作方式如下：

1. 可以在 EC2 資源的 Metadata 區段中定義組態。
2. 可以透過 UserData 指令碼呼叫 cfn-init。
3. cfn-init 會讀取中繼資料並套用組態。
4. 根據您的規格設定您的執行個體。

中繼資料結構

在 EC2 執行個體中的特定結構中定義組態。

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:                                # Metadata section for the resource
      AWS::CloudFormation::Init:           # Required key that cfn-init looks for
        config:                             # Configuration name (you can have multiple)
          packages:                         # Install packages
```

```
files:                # Create files
commands:             # Run commands
services:             # Start/stop services
```

cfn-init 指令碼會按照特定順序處理這些區段：
packages、groups、users、sources、files、commands、services。

從簡單的引導範例開始

讓我們從安裝和啟動 Apache 的最低引導範例開始。

```
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:          # Install Apache web server
            yum:
              httpd: []
          services:          # Start Apache and enable it to start on boot
            systemd:
              httpd:
                enabled: true
                ensureRunning: true
    Properties:
      ImageId: !Ref LatestAmiId
      InstanceType: !Ref InstanceType
      UserData: !Base64          # Script that runs when instance starts
      Fn::Sub: |
        #!/bin/bash
        yum install -y aws-cfn-bootstrap
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource EC2Instance --
        region ${AWS::Region}
```

這個簡單的範例示範了核心概念：

- packages 區段會使用 yum 安裝 httpd 套件。這適用於 Amazon Linux 和使用 yum 的其他 Linux 發行版本。
- services 區段可確保 httpd 自動啟動和執行。
- UserData 安裝最新的引導工具並呼叫 cfn-init。

新增檔案和命令

現在，讓我們透過在 EC2 執行個體的 `/var/log` 目錄中新增自訂網頁和日誌檔案來增強我們的範例。

建立檔案

`files` 區段可讓您在具有特定內容的執行個體上建立檔案。垂直管道 (`|`) 可讓您傳遞文字區塊 (HTML 程式碼) 作為檔案的內容 (`/var/www/html/index.html`)。

```
files:
  /var/www/html/index.html:
    content: |
      <body>
        <h1>Congratulations, you have successfully launched the AWS CloudFormation
sample.</h1>
      </body>
```

執行命令

`commands` 區段可讓您在引導程序期間執行 Shell 命令。此命令會在 EC2 執行個體上的 `/var/log/welcome.txt` 建立日誌檔案。若要檢視它，需要 Amazon EC2 金鑰對來用於 SSH 存取，以及可用於 SSH 到執行個體的 IP 位址範圍 (此處未涵蓋)。

```
commands:
  createWelcomeLog:
    command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
```

新增網路安全

由於我們正在設定 Web 伺服器，我們需要允許 Web 流量 (HTTP) 來連接我們的 EC2 執行個體。因此，我們會建立一個安全群組，它允許來自 IP 位址的連接埠 80 的傳入流量。EC2 執行個體也需要將流量傳送到網際網路，例如安裝套件更新。根據預設，安全群組允許所有傳出流量。然後，我們將使用 `SecurityGroupIds` 屬性來關聯此安全群組與 EC2 執行個體。

```
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow HTTP access from my IP address
    SecurityGroupIngress:
      - IpProtocol: tcp
        Description: HTTP
        FromPort: 80
```

```
ToPort: 80
CidrIp: !Ref MyIP
```

完整的引導範本

現在，讓我們將所有部分放在一起。以下是合併我們討論的所有概念的完整範本。

```
AWSTemplateFormatVersion: 2010-09-09
Description: Bootstrap an EC2 instance with Apache web server using cfn-init

Parameters:
  LatestAmiId:
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'

  InstanceType:
    Description: EC2 instance type
    Type: String
    Default: t2.micro
    AllowedValues:
      - t3.micro
      - t2.micro
    ConstraintDescription: must be a valid EC2 instance type.

  MyIP:
    Description: Your IP address in CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.(\d{1,2})$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow HTTP access from my IP address
      SecurityGroupIngress:
        - IpProtocol: tcp
          Description: HTTP
          FromPort: 80
```

```

        ToPort: 80
        CidrIp: !Ref MyIP

WebServer:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
        files:
          /var/www/html/index.html:
            content: |
              <body>
                <h1>Congratulations, you have successfully launched the AWS
CloudFormation sample.</h1>
              </body>
        commands:
          createWelcomeLog:
            command: "echo 'cfn-init ran successfully!' > /var/log/welcome.txt"
        services:
          systemd:
            httpd:
              enabled: true
              ensureRunning: true
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    UserData: !Base64
      Fn::Sub: |
        #!/bin/bash
        yum install -y aws-cfn-bootstrap
        /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServer --
region ${AWS::Region}
    Tags:
      - Key: Name
        Value: Bootstrap Tutorial Web Server

Outputs:
  WebsiteURL:
    Value: !Sub 'http://${WebServer.PublicDnsName}'

```

Description: EC2 instance public DNS name

使用主控台建立堆疊

下列程序涉及從檔案上傳範例堆疊範本。在本機電腦上打開文字編輯器，並新增範本。儲存檔案，並將其命名為 `samplelinux2stack.template`。

啟動堆疊範本

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 選擇 Create stack (建立堆疊)、With new resources (standard) (使用新資源 (標準))。
3. 在指定範本下，選擇上傳範本檔案，然後選擇選擇檔案以上傳 `samplelinux2stack.template` 檔案。
4. 選擇下一步。
5. 在指定詳細資訊頁面上，輸入 **BootstrapTutorialStack** 作為堆疊名稱。
6. 在參數中，請執行下列動作。
 - LatestAmild：保留預設值。
 - InstanceType：為 EC2 執行個體類型選擇 `t2.micro` 或 `t3.micro`。
 - MyIP：輸入尾碼為 /32 的公有 IP 位址。
7. 選擇下一步兩次，然後選擇提交以建立堆疊。

監控引導程序

引導程序需要的時間比簡單的 EC2 啟動時間更長，因為正在安裝和設定其他軟體。

若要監控引導進度

1. 在 CloudFormation 主控台中，選取堆疊並開啟事件索引標籤。
2. 請留意 WebServer CREATE_IN_PROGRESS 事件。引導程序會在執行個體啟動後開始。
3. 引導程序通常需要幾分鐘時間。您會在完成後看到 WebServer CREATE_COMPLETE。

如果想要查看引導程序期間發生的情況，可以檢查執行個體日誌。

若要檢視引導日誌 (選用)

1. 開啟 [EC2 主控台](#) 並找到執行個體。
2. 選取執行個體，然後再依序選擇動作、監控和故障診斷、取得系統日誌，以查看引導進度。
3. 如果沒有立即看到日誌，請等待並重新整理頁面。

測試引導的 Web 伺服器

當堆疊顯示 CREATE_COMPLETE 時，請測試 Web 伺服器。

若要測試 Web 伺服器

1. 在 CloudFormation 主控台中，前往堆疊的輸出索引標籤。
2. 按一下網站 URL 值，在新索引標籤中開啟您的 Web 伺服器。
3. 應該會看到包含訊息 Congratulations, you have successfully launched the AWS CloudFormation sample 的自訂網頁。

Note

如果頁面未立即載入，請等待一分鐘，然後再試一次。即使堆疊顯示 CREATE_COMPLETE，引導程序仍可能正在完成。

對引導問題進行疑難排解

如果引導程序失敗或 Web 伺服器無法運作，以下是常見問題和解決方案。

常見問題

- 堆疊建立失敗 – 檢查事件索引標籤是否有特定錯誤訊息。
- 無法存取 Web 伺服器 – 在 MyIP 參數中確認 IP 位址正確。請記得在結尾包含 /32。
- 引導程序失敗 – 執行個體可能會啟動，但 cfn-init 失敗。如監控章節所述，檢查系統日誌。

清除資源

若要避免持續收費，可以透過刪除堆疊及其資源來清除。

刪除堆疊及其資源

1. 開啟 [CloudFormation 主控台](#)。
2. 在堆疊頁面中，選取您所建立堆疊名稱旁的選項 (**BootstrapTutorialStack**)，然後選擇刪除。
3. 出現確認提示時，請選擇刪除。
4. 在事件索引標籤中監控堆疊刪除程序的進度。**BootstrapTutorialStack** 的狀態會變更為 **DELETE_IN_PROGRESS**。當 CloudFormation 完成刪除堆疊時，它會從清單移除堆疊。

後續步驟

恭喜您！您已成功了解如何使用 CloudFormation 來啟動 EC2 執行個體。您現在了解：

- 如何使用 `cfn-init` 協助程式指令碼
- 如何建構用於引導的中繼資料
- 如何安裝套件、建立檔案、執行命令和管理服務
- 如何監控引導問題

若要繼續學習：

- 了解如何更新執行中的堆疊並使用 `cfn-hup` 協助程式指令碼。如需詳細資訊，請參閱 [更新 CloudFormation 堆疊](#)。
- 了解如何引導 Windows 堆疊。如需詳細資訊，請參閱 [引導 Windows 型 CloudFormation 堆疊](#)。
- 探索具有多個組態集的更複雜引導案例。如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [cfn-init](#) 和 [AWS::CloudFormation::Init](#)。
- 了解有關用於報告引導完成狀態的 `cfn-signal`。如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [cfn-signal](#)。

更新 CloudFormation 堆疊

Note

本教學課程以在 [Amazon EC2 上部署應用程式](#) 教學課程中的概念為基礎。如果您尚未完成該教學課程，建議您先了解使用 CloudFormation 的 EC2 引導。

本主題示範了對執行中堆疊的簡單更新進度。我們將逐步解說以下步驟：

1. 建立初始堆疊 – 使用基本 Amazon Linux 2 AMI 建立堆疊，並使用 CloudFormation 協助程式指令碼安裝 Apache Web Server 和簡單的 PHP 應用程式。
2. 更新應用程式 – 更新應用程式中的其中一個檔案，並使用 CloudFormation 部署軟體。
3. 新增金鑰對 – 將 Amazon EC2 金鑰對新增至執行個體，然後更新安全群組以允許 SSH 存取執行個體。
4. 更新執行個體類型 – 變更基礎 Amazon EC2 執行個體的執行個體類型。
5. 更新 AMI – 變更堆疊中 Amazon EC2 執行個體的 Amazon Machine Image (AMI)。

Note

CloudFormation 是免費的，但需要為您建立的 Amazon EC2 資源付費。不過，如果您是新手 AWS，則可以利用 [免費方案](#)，在此學習過程中將成本降至最低或消除。

主題

- [步驟 1：建立初始堆疊](#)
- [步驟 2：更新應用程式](#)
- [步驟 3：使用金鑰對新增 SSH 存取](#)
- [步驟 4：更新執行個體類型](#)
- [步驟 5：更新 AMI](#)
- [可用性與影響考量事項](#)
- [相關資源](#)

步驟 1：建立初始堆疊

首先，我們會建立堆疊，供本主題的其餘部分使用。我們提供了簡單的範本，可啟動託管在上的單一執行個體 PHP Web 應用程式，Apache Web Server 並在 Amazon Linux 2 AMI 上執行。

Apache Web Server、PHP 和簡單的 PHP 應用程式都是由預設安裝在 Amazon Linux 2 AMI 上的 CloudFormation 協助程式指令碼所安裝。下列範本程式碼片段顯示描述要安裝的套件和檔案的中繼資料，在此情況下，Amazon Linux 2 AMI 儲存 Yum 庫中的 Apache Web Server 和 PHP 基礎設施。程式碼片段也會顯示 Services 區段，這可確保 Apache Web Server 正在執行。

```
WebServerInstance:
  Type: AWS::EC2::Instance
  Metadata:
    AWS::CloudFormation::Init:
      config:
        packages:
          yum:
            httpd: []
            php: []
        files:
          /var/www/html/index.php:
            content: |
              <?php
              echo '<h1>Hello World!</h1>';
              ?>
            mode: '000644'
            owner: apache
            group: apache
        services:
          systemd:
            httpd:
              enabled: true
              ensureRunning: true
```

應用程式本身是完全在範本中定義的「Hello World」範例。若是實際的應用程式，系統可能會將檔案存放在 Amazon S3、GitHub 或其他儲存庫上，並參考範本的內容。CloudFormation 能夠下載 RPM 或 RubyGems 等套件、參考個別檔案以及解壓縮 .zip 與 .tar 檔案，藉此在 Amazon EC2 執行個體上建立應用程式成品。

範本會啟用並設定 cfn-hup 協助程式，以接聽 Amazon EC2 執行個體中繼資料中定義的組態變更。透過使用 cfn-hup 協助程式，您可以更新應用程式軟體，例如 Apache 或 PHP 的版本，也可以從 CloudFormation 更新 PHP 應用程式檔案本身。範本中相同 Amazon EC2 資源的下列程式碼片段顯示設定 cfn-init 每兩分鐘 cfn-hup 呼叫以通知中繼資料並套用更新所需的片段。否則，cfn-init 只會在啟動時執行一次。

```
files:
  /etc/cfn/cfn-hup.conf:
    content: !Sub |
      [main]
      stack=${AWS::StackId}
      region=${AWS::Region}
```

```

    # The interval used to check for changes to the resource metadata in minutes.
    Default is 15
    interval=2
    mode: '000400'
    owner: root
    group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
  content: !Sub |
    [cfn-auto-reloader-hook]
    triggers=post.update
    path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
    action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --region
    ${AWS::Region}
    runas=root
services:
  systemd:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf

```

為了完成堆疊，在 Amazon EC2 執行個體定義的 Properties 區段中，UserData 屬性包含呼叫 `cfn-init` 來安裝套件和檔案的 `cloud-init` 指令碼。如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [CloudFormation 協助程式指令碼參考](#)。範本也會建立 Amazon EC2 安全群組。

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

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
```

```
  InstanceType:
```

```
    Description: WebServer EC2 instance type
```

```
    Type: String
```

```
    Default: t3.micro
```

```
    AllowedValues:
```

- t3.nano
- t3.micro
- t3.small

- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

yum update -y aws-cfn-bootstrap

Run cfn-init

/opt/aws/bin/cfn-init -v --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region} || error_exit 'Failed to run cfn-init'

Start up the cfn-hup daemon to listen for changes to the EC2 instance

metadata

/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'

Signal success or failure

```

    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerInstance --region ${AWS::Region}
Metadata:
  AWS::CloudFormation::Init:
    config:
      packages:
        yum:
          httpd: []
          php: []
      files:
        /var/www/html/index.php:
          content: |
            <?php
            echo "<h1>Hello World!</h1>";
            ?>
          mode: '000644'
          owner: apache
          group: apache
        /etc/cfn/cfn-hup.conf:
          content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
            mode: '000400'
            owner: root
            group: root
        /etc/cfn/hooks.d/cfn-auto-reloader.conf:
          content: !Sub |
            [cfn-auto-reloader-hook]
            triggers=post.update
            path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
            runas=root
      services:
        systemd:
          httpd:
            enabled: true
            ensureRunning: true
          cfn-hup:
            enabled: true

```

```
        ensureRunning: true
        files:
          - /etc/cfn/cfn-hup.conf
          - /etc/cfn/hooks.d/cfn-auto-reloader.conf
    CreationPolicy:
      ResourceSignal:
        Timeout: PT5M

    WebServerSecurityGroup:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupDescription: Enable HTTP access via port 80
        SecurityGroupIngress:
          - IpProtocol: tcp
            FromPort: 80
            ToPort: 80
            CidrIp: 0.0.0.0/0

    Outputs:
      WebsiteURL:
        Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
        Description: URL of the web application
```

從此範本啟動堆疊

1. 複製範本，並將其儲存為文字檔案於本機系統。請記下檔案位置，因為後續步驟仍需使用到該檔案。
2. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
3. 選擇使用新資源建立堆疊（標準）。
4. 選擇選擇現有的範本。
5. 在指定範本下，選擇上傳範本檔案，然後瀏覽至您在第一個步驟中建立的檔案，然後選擇下一步。
6. 在指定堆疊詳細資訊頁面上，輸入 **UpdateTutorial** 做為堆疊名稱。
7. 在參數下，保持所有參數相同，然後選擇下一步兩次。
8. 在檢閱和建立畫面上，選擇提交。

堆疊的狀態為 之後 CREATE_COMPLETE，輸出索引標籤會顯示您網站的 URL。如果您選擇 WebsiteURL 輸出的值，您會看到新的 PHP 應用程式正在運作。

步驟 2：更新應用程式

既然我們已經將堆疊部署完成，接下來便要更新應用程式。我們會簡單地更改應用程式所印出的文字。若要執行此作業，便需要將 echo 命令新增至 index.php 檔案，如下列範本程式碼片段所示：

```
files:
  /var/www/html/index.php:
    content: |
      <?php
      echo "<h1>Hello World!</h1>";
      echo "<p>This is an updated version of our application.</p>";
      ?>
    mode: '000644'
    owner: apache
    group: apache
```

透過文字編輯器，即可手動編輯儲存在本機上的範本檔案。

現在，請更新堆疊。

使用已更新的範本更新堆疊

1. 在 CloudFormation 主控台中，選取您的**UpdateTutorial**堆疊。
2. 選擇更新，直接更新。
3. 選擇取代現有範本。
4. 在指定範本下，選擇上傳範本檔案並上傳修改後的範本檔案，然後選擇下一步。
5. 在指定堆疊詳細資訊頁面上，保持所有參數相同，然後選擇下一步兩次。
6. 在檢閱頁面上，檢閱變更。在變更下，您應該會看到 CloudFormation 會更新 **WebServerInstance** 資源。
7. 選擇提交。

當您的堆疊處於 UPDATE_COMPLETE 狀態時，您可以再次選擇 **WebsiteURL** 輸出值，以確認應用程式的變更已生效。cfn-hup 協助程式每 2 分鐘執行一次，因此更新堆疊後，應用程式最多可能需要 2 分鐘才能變更。

前往 CloudFormation 主控台，即可查看更新後的資源集。您可以在 Events (事件) 標籤上檢視堆疊事件。在此情況下，Amazon EC2 執行個體的中繼資料 **WebServerInstance** 已更新，這會導致 CloudFormation 也重新評估其他資源 (**WebServerSecurityGroup**)，以確保沒有其他變更。系統

並不會修改其他堆疊資源。唯有堆疊中的資源受到任何堆疊變更影響時，CloudFormation 才會更新這些資源。這類變更可以是直接的，例如屬性或中繼資料變更，也可能是由於相依性或透過 Ref、GetAtt或其他內部範本函數的資料流程所造成。如需詳細資訊，請參閱[內建函數參考](#)。

這個簡單的更新說明了程序。不過，您可以對部署到 Amazon EC2 執行個體的檔案和套件進行更複雜的變更。例如，您可能會認為需要將 MySQL 與 MySQL 的 PHP 支援新增至執行個體。若要執行此作業，只需將其他套件、檔案與任何其他服務一併新增至組態，接著更新堆疊，即可部署變更。

```
packages:
  yum:
    httpd: []
    php: []
    mysql: []
    php-mysql: []
    mysql-server: []
    mysql-libs: []

    ...

services:
  systemd:
    httpd:
      enabled: true
      ensureRunning: true
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - /etc/cfn/cfn-hup.conf
        - /etc/cfn/hooks.d/cfn-auto-reloader.conf
    mysqld:
      enabled: true
      ensureRunning: true
```

您可以更新 CloudFormation 中繼資料，藉此將應用程式所使用的套件更新至新版本。在上述範例中，每個套件的版本屬性都是空的，表示 cfn-init 應該安裝最新版本的套件。

```
packages:
  yum:
    httpd: []
    php: []
```

您可以選擇指定套件的版本字串；若您在後續的更新堆疊呼叫中更改該版本字串，則系統將部署新版本的套件。下方為使用 RubyGems 套件版本編號的範例。而支援版本控制的任何套件皆可擁有特定版本。

```
packages:
  rubygems:
    mysql: []
    rubygems-update:
      - "1.6.2"
    rake:
      - "0.8.7"
    rails:
      - "2.3.11"
```

步驟 3：使用金鑰對新增 SSH 存取

您也可以更新範本中的資源，以新增最初未在範本中指定的屬性。為了說明這一點，我們會將 Amazon EC2 金鑰對新增至現有的 EC2 執行個體，然後在 Amazon EC2 安全群組中開啟連接埠 22，以便您可以使用 Secure Shell (SSH) 存取執行個體。

在現有的 Amazon EC2 執行個體中新增 SSH 存取

1. 將兩個額外參數新增至範本，藉此傳遞現有的 Amazon EC2 金鑰對名稱與 SSH 位置。

```
Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the
instance
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.

  SSHLocation:
    Description: The IP address that can be used to SSH to the EC2 instances in
CIDR format (e.g. 203.0.113.1/32)
    Type: String
    MinLength: 9
    MaxLength: 18
    Default: 0.0.0.0/0
    AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\.(\d{1,2})$'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
```

2. 將 KeyName 屬性新增至 Amazon EC2 執行個體。

```

WebServerInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: !Ref LatestAmiId
    InstanceType: !Ref InstanceType
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup

```

3. 將連接埠 22 與 SSH 位置新增至 Amazon EC2 安全群組的輸入規則。

```

WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation

```

4. 使用與 中說明的相同步驟來更新堆疊[步驟 2：更新應用程式](#)。

步驟 4：更新執行個體類型

現在，讓我們示範如何透過變更執行個體類型來更新基礎基礎設施。

到目前為止，我們建置的堆疊使用 t3.micro Amazon EC2 執行個體。假設您新建立的網站取得的流量超過 t3.micro 執行個體可以處理的流量，現在您想要移至 m5.large Amazon EC2 執行個體類型。如果執行個體類型的架構變更，則必須使用不同的 AMI 建立執行個體。不過，t3.micro 和 m5.large 都使用相同的 CPU 架構，並執行 Amazon Linux 2 (x86_64) AMIs。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[變更執行個體類型的相容性](#)。

讓我們使用我們在上一個步驟中修改的範本來變更執行個體類型。由於 InstanceType 是範本的輸入參數，因此我們不需要修改範本；我們可以在指定堆疊詳細資訊頁面上變更 參數的值。

使用新的參數值更新堆疊

1. 在 CloudFormation 主控台中，選取您的**UpdateTutorial**堆疊。
2. 選擇更新，直接更新。
3. 選擇使用現有範本，然後選擇下一步。
4. 在指定堆疊詳細資訊頁面上，將 InstanceType 文字方塊的值從 t3.micro 變更為 m5.large。然後，選擇下一步兩次。
5. 在檢閱頁面上，檢閱變更。在變更下，您應該會看到 CloudFormation 會更新 WebServerInstance 資源。
6. 選擇提交。

透過啟動和停用執行個體，即可動態變更 EBS 後端 Amazon EC2 執行個體的執行個體類型。CloudFormation 會更新執行個體類型並重新啟動該執行個體，藉此嘗試將變更內容最佳化，讓執行個體 ID 不會因而改變。然而，重新啟動執行個體時，該執行個體的公有 IP 地址會隨之變更。為了確保彈性 IP 位址在變更後仍能正確繫結，CloudFormation 將一併更新彈性 IP 地址。您可以在 CloudFormation CloudFormation 主控台的事件索引標籤上查看變更。

若要從 檢查執行個體類型 AWS 管理主控台，請開啟 Amazon EC2 主控台，並在該處找到您的執行個體。

步驟 5：更新 AMI

現在，讓我們更新堆疊以使用 Amazon Linux 2023，這是新一代的 Amazon Linux。

更新 AMI 是一項需要取代執行個體的主要變更。我們不能只啟動和停止執行個體來修改 AMI；CloudFormation 會將此視為對資源不可變屬性的變更。CloudFormation 必須啟動替代資源，才能更改不可變屬性；此處將以新的 Amazon EC2 執行個體執行新的 AMI 為例。

我們來看看如何更新堆疊範本以使用 Amazon Linux 2023。關鍵變更包括更新 AMI 參數，以及從 yum 變更為 dnf 套件管理員。

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

```
Parameters:
```

```
  LatestAmiId:
```

```
    Description: The latest Amazon Linux 2023 AMI from the Parameter Store
```

```
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
```

```
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64'
```

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t3.micro

AllowedValues:

- t3.nano
- t3.micro
- t3.small
- t3.medium
- t3a.nano
- t3a.micro
- t3a.small
- t3a.medium
- m5.large
- m5.xlarge
- m5.2xlarge
- m5a.large
- m5a.xlarge
- m5a.2xlarge
- c5.large
- c5.xlarge
- c5.2xlarge
- r5.large
- r5.xlarge
- r5.2xlarge
- r5a.large
- r5a.xlarge
- r5a.2xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Description: Name of an existing EC2 KeyPair to enable SSH access to the instance

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address that can be used to SSH to the EC2 instances in CIDR format (e.g. 203.0.113.1/32)

Type: String

MinLength: 9

MaxLength: 18

Default: 0.0.0.0/0

AllowedPattern: '^(\d{1,3}\.){3}\d{1,3}\/\d{1,2}\$'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

Resources:

WebServerInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref WebServerSecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash -xe

Get the latest CloudFormation package

dnf update -y aws-cfn-bootstrap

Run cfn-init

/opt/aws/bin/cfn-init -v --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region} || error_exit 'Failed to run cfn-init'

Start up the cfn-hup daemon to listen for changes to the EC2 instance

metadata

/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'

Signal success or failure

/opt/aws/bin/cfn-signal -e \$? --stack \${AWS::StackName} --resource

WebServerInstance --region \${AWS::Region}

Metadata:

AWS::CloudFormation::Init:

config:

packages:

dnf:

httpd: []

php: []

files:

/var/www/html/index.php:

content: |

<?php

echo "<h1>Hello World!</h1>";

echo "<p>This is an updated version of our application.</p>";

echo "<p>Running on Amazon Linux 2023!</p>";

?>

mode: '000644'

owner: apache

group: apache

/etc/cfn/cfn-hup.conf:

```

        content: !Sub |
            [main]
            stack=${AWS::StackId}
            region=${AWS::Region}
            # The interval used to check for changes to the resource metadata in
minutes. Default is 15
            interval=2
            mode: '000400'
            owner: root
            group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
        content: !Sub |
            [cfn-auto-reloader-hook]
            triggers=post.update
            path=Resources.WebServerInstance.Metadata.AWS::CloudFormation::Init
            action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerInstance --
region ${AWS::Region}
            runas=root
    services:
        systemd:
            httpd:
                enabled: true
                ensureRunning: true
            cfn-hup:
                enabled: true
                ensureRunning: true
            files:
                - /etc/cfn/cfn-hup.conf
                - /etc/cfn/hooks.d/cfn-auto-reloader.conf
    CreationPolicy:
        ResourceSignal:
            Timeout: PT5M

WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
        GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
        SecurityGroupIngress:
            - IpProtocol: tcp
              FromPort: 80
              ToPort: 80
              CidrIp: 0.0.0.0/0
            - IpProtocol: tcp
              FromPort: 22

```



```
ToPort: 22
CidrIp: !Ref SSHLocation
```

Outputs:

```
WebsiteURL:
  Value: !Sub 'http://${WebServerInstance.PublicDnsName}'
  Description: URL of the web application
```

使用與 中說明的相同步驟來更新堆疊[步驟 2：更新應用程式](#)。

新的執行個體開始運作後，CloudFormation 即會更新堆疊中的其他資源，進而指向新資源。系統會進行稱為 UPDATE_CLEANUP 的程序，藉此建立新資源、刪除舊資源。此時，堆疊中執行個體的執行個體 ID 與應用程式 URL，皆會因更新作業而有所變更。事件資料表中的事件包含描述「請求的更新已變更不可變屬性，因此建立新的實體資源」，以指出資源已取代。

或者：如果您將應用程式碼寫入您要更新的 AMI，您可以使用相同的堆疊更新機制來更新 AMI 以載入新的應用程式。

使用自訂應用程式碼更新 AMI

1. 建立包含應用程式或作業系統變更的新 AMI。如需詳細資訊，請參閱 [《Amazon EC2 使用者指南》中的建立 Amazon EBS 後端 AMI](#)。Amazon EC2
2. 更新您的範本以納入新的 AMI ID。
3. 使用與 中說明的相同步驟來更新堆疊[步驟 2：更新應用程式](#)。

更新堆疊時，CloudFormation 即會偵測到 AMI ID 有所變更，然後啟動堆疊更新作業 (與上述更新的觸發方式相同)。

可用性與影響考量事項

不同屬性會對堆疊中的資源造成不同影響。您可以善用 CloudFormation 來更新任何屬性；但進行任何變更之前，應考量下列問題：

1. 更新作業對資源本身有何影響？舉例來說，更新警示閾值會導致警示在更新期間處於非作用中狀態。正如我們在本章所見，您需要停用並重新啟動執行個體，才能變更執行個體類型。CloudFormation 會透過基礎資源的更新或修改動作來更改資源。如需了解更新造成的影響，請查閱特定資源的文件。
2. 更改內容屬於可變或不可變項目？基礎服務並不支援部分資源屬性變更操作，例如：更改 Amazon EC2 執行個體上的 AMI。如果更改內容屬於可變項目，CloudFormation 將針對基

基礎資源使用 Update (更新) 或 Modify (修改) 類型的 API。如果屬性更改屬於不可變項目，則 CloudFormation 會使用更新後的屬性來建立新資源，並將這些資源連結至堆疊，接著刪除舊資源。雖然 CloudFormation 會嘗試縮短堆疊資源的停機時間，但替換資源屬於多步驟程序，需要一些時間才能完成。重新設定堆疊期間，應用程式將無法完全正常運作。例如，應用程式可能無法處理請求或存取資料庫；

相關資源

如需使用 CloudFormation 啟動應用程式以及與其他組態和部署服務整合的詳細資訊，例如 Puppet 和 Opscode Chef，請參閱下列白皮書：

- [透過 CloudFormation 引導應用程式](#)
- [將 CloudFormation 與 整合 Opscode Chef](#)
- [將 CloudFormation 與 整合 Puppet](#)

建置擴展和負載平衡的應用程式

在此逐步解說中，您將建立可協助您設定已擴展和已負載平衡應用程式的堆疊。逐步解說提供您將用於建立堆疊的範例範本。範例範本佈建 Auto Scaling 群組、Application Load Balancer、控制負載平衡器和 Auto Scaling 群組流量的安全群組，以及用以發布擴展活動相關通知的 Amazon SNS 通知組態。

此範本會建立一或多個 Amazon EC2 執行個體和一個 Application Load Balancer。若您從此範本建立堆疊，您必須為使用的 AWS 資源支付費用。

完整堆疊範本

讓我們從範本開始。

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Parameters:
  InstanceType:
    Description: The EC2 instance type
    Type: String
    Default: t3.micro
    AllowedValues:
      - t3.micro
      - t3.small
      - t3.medium
```

```
KeyName:
  Description: Name of an existing EC2 key pair to allow SSH access to the instances
  Type: AWS::EC2::KeyPair::KeyName
LatestAmiId:
  Description: The latest Amazon Linux 2 AMI from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2'
OperatorEmail:
  Description: The email address to notify when there are any scaling activities
  Type: String
SSHLocation:
  Description: The IP address range that can be used to SSH to the EC2 instances
  Type: String
  MinLength: 9
  MaxLength: 18
  Default: 0.0.0.0/0
  ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
Subnets:
  Type: 'List<AWS::EC2::Subnet::Id>'
  Description: At least two public subnets in different Availability Zones in the
selected VPC
VPC:
  Type: AWS::EC2::VPC::Id
  Description: A virtual private cloud (VPC) that enables resources in public subnets
to connect to the internet
Resources:
  ELBSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: ELB Security Group
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
  EC2SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: EC2 Security Group
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
```

```
    ToPort: 80
    SourceSecurityGroupId:
      Fn::GetAtt:
        - ELBSecurityGroup
        - GroupId
    - IpProtocol: tcp
      FromPort: 22
      ToPort: 22
      CidrIp: !Ref SSHLocation
EC2TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 30
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 15
    HealthyThresholdCount: 5
    Matcher:
      HttpCode: '200'
    Name: EC2TargetGroup
    Port: 80
    Protocol: HTTP
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: '20'
    UnhealthyThresholdCount: 3
    VpcId: !Ref VPC
ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref EC2TargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    Subnets: !Ref Subnets
    SecurityGroups:
      - !GetAtt ELBSecurityGroup.GroupId
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
```

Properties:

LaunchTemplateName: !Sub \${AWS::StackName}-launch-template

LaunchTemplateData:

ImageId: !Ref LatestAmiId

InstanceType: !Ref InstanceType

KeyName: !Ref KeyName

SecurityGroupIds:

- !Ref EC2SecurityGroup

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "<h1>Hello World!</h1>" > /var/www/html/index.html

NotificationTopic:

Type: AWS::SNS::Topic

Properties:**Subscription:**

- Endpoint: !Ref OperatorEmail

Protocol: email

WebServerGroup:

Type: AWS::AutoScaling::AutoScalingGroup

Properties:**LaunchTemplate:**

LaunchTemplateId: !Ref LaunchTemplate

Version: !GetAtt LaunchTemplate.LatestVersionNumber

MaxSize: '3'

MinSize: '1'

NotificationConfigurations:

- TopicARN: !Ref NotificationTopic

NotificationTypes: ['autoscaling:EC2_INSTANCE_LAUNCH',

'autoscaling:EC2_INSTANCE_LAUNCH_ERROR', 'autoscaling:EC2_INSTANCE_TERMINATE',

'autoscaling:EC2_INSTANCE_TERMINATE_ERROR']

TargetGroupARNs:

- !Ref EC2TargetGroup

VPCZoneIdentifier: !Ref Subnets

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
```

```
"Parameters":{
  "InstanceType":{
    "Description":"The EC2 instance type",
    "Type":"String",
    "Default":"t3.micro",
    "AllowedValues":[
      "t3.micro",
      "t3.small",
      "t3.medium"
    ]
  },
  "KeyName":{
    "Description":"Name of an existing EC2 key pair to allow SSH access to the
instances",
    "Type":"AWS::EC2::KeyPair::KeyName"
  },
  "LatestAmiId":{
    "Description":"The latest Amazon Linux 2 AMI from the Parameter Store",
    "Type":"AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
    "Default":"/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
  },
  "OperatorEmail":{
    "Description":"The email address to notify when there are any scaling
activities",
    "Type":"String"
  },
  "SSHLocation":{
    "Description":"The IP address range that can be used to SSH to the EC2
instances",
    "Type":"String",
    "MinLength":9,
    "MaxLength":18,
    "Default":"0.0.0.0/0",
    "ConstraintDescription":"Must be a valid IP CIDR range of the form x.x.x.x/x."
  },
  "Subnets":{
    "Type":"List<AWS::EC2::Subnet::Id>",
    "Description":"At least two public subnets in different Availability Zones in the
selected VPC"
  },
  "VPC":{
    "Type":"AWS::EC2::VPC::Id",
    "Description":"A virtual private cloud (VPC) that enables resources in public
subnets to connect to the internet"
```

```
    }
  },
  "Resources":{
    "ELBSecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
      "Properties":{
        "GroupDescription":"ELB Security Group",
        "VpcId":{
          "Ref":"VPC"
        },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "CidrIp":"0.0.0.0/0"
          }
        ]
      }
    },
    "EC2SecurityGroup":{
      "Type":"AWS::EC2::SecurityGroup",
      "Properties":{
        "GroupDescription":"EC2 Security Group",
        "VpcId":{
          "Ref":"VPC"
        },
        "SecurityGroupIngress":[
          {
            "IpProtocol":"tcp",
            "FromPort":80,
            "ToPort":80,
            "SourceSecurityGroupId":{
              "Fn::GetAtt":[
                "ELBSecurityGroup",
                "GroupId"
              ]
            }
          },
          {
            "IpProtocol":"tcp",
            "FromPort":22,
            "ToPort":22,
            "CidrIp":{
```

```

        "Ref": "SSHLocation"
      }
    }
  ]
}
},
"EC2TargetGroup": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "HealthCheckIntervalSeconds": 30,
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeoutSeconds": 15,
    "HealthyThresholdCount": 5,
    "Matcher": {
      "HttpCode": "200"
    },
    "Name": "EC2TargetGroup",
    "Port": 80,
    "Protocol": "HTTP",
    "TargetGroupAttributes": [
      {
        "Key": "deregistration_delay.timeout_seconds",
        "Value": "20"
      }
    ],
    "UnhealthyThresholdCount": 3,
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"ALBListener": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [
      {
        "Type": "forward",
        "TargetGroupArn": {
          "Ref": "EC2TargetGroup"
        }
      }
    ],
    "LoadBalancerArn": {
      "Ref": "ApplicationLoadBalancer"
    }
  }
}
},

```



```
    },
    "Port":80,
    "Protocol":"HTTP"
  }
},
"ApplicationLoadBalancer":{
  "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties":{
    "Scheme":"internet-facing",
    "Subnets":{
      "Ref":"Subnets"
    },
    "SecurityGroups":[
      {
        "Fn::GetAtt":[
          "ELBSecurityGroup",
          "GroupId"
        ]
      }
    ]
  }
},
"LaunchTemplate":{
  "Type":"AWS::EC2::LaunchTemplate",
  "Properties":{
    "LaunchTemplateName":{
      "Fn::Sub":"${AWS::StackName}-launch-template"
    },
    "LaunchTemplateData":{
      "ImageId":{
        "Ref":"LatestAmiId"
      },
      "InstanceType":{
        "Ref":"InstanceType"
      },
      "KeyName":{
        "Ref":"KeyName"
      },
      "SecurityGroupIds":[
        {
          "Ref":"EC2SecurityGroup"
        }
      ],
      "UserData":{
```

```

        "Fn::Base64":{
            "Fn::Join":[
                "",
                [
                    "#!/bin/bash\n",
                    "yum update -y\n",
                    "yum install -y httpd\n",
                    "systemctl start httpd\n",
                    "systemctl enable httpd\n",
                    "echo \"<h1>Hello World!</h1>\" > /var/www/html/index.html"
                ]
            ]
        }
    },
    "NotificationTopic":{
        "Type":"AWS::SNS::Topic",
        "Properties":{
            "Subscription":[
                {
                    "Endpoint":{
                        "Ref":"OperatorEmail"
                    },
                    "Protocol":"email"
                }
            ]
        }
    },
    "WebServerGroup":{
        "Type":"AWS::AutoScaling::AutoScalingGroup",
        "Properties":{
            "LaunchTemplate":{
                "LaunchTemplateId":{
                    "Ref":"LaunchTemplate"
                },
                "Version":{
                    "Fn::GetAtt":[
                        "LaunchTemplate",
                        "LatestVersionNumber"
                    ]
                }
            }
        }
    },

```

```
    "MaxSize": "3",
    "MinSize": "1",
    "NotificationConfigurations": [
      {
        "TopicARN": {
          "Ref": "NotificationTopic"
        },
        "NotificationTypes": [
          "autoscaling:EC2_INSTANCE_LAUNCH",
          "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
          "autoscaling:EC2_INSTANCE_TERMINATE",
          "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
        ]
      }
    ],
    "TargetGroupARNs": [
      {
        "Ref": "EC2TargetGroup"
      }
    ],
    "VPCZoneIdentifier": {
      "Ref": "Subnets"
    }
  }
}
```

範本演練

此範本的第一部分會指定 `Parameters`。每個參數都必須在執行時間指派一個值，CloudFormation 才能成功佈建堆疊。在範本中稍後指定的資源會參考這些值並使用資料。

- `InstanceType`：Amazon EC2 Auto Scaling 佈建的 EC2 執行個體類型。如果未指定，則會使用預設的 `t3.micro`。
- `KeyName`：允許 SSH 存取執行個體的現有 EC2 金鑰對。
- `LatestAmiId`：執行個體的 Amazon Machine Image (AMI)。如果未指定，您的執行個體會使用 維護的公有 AWS Systems Manager 參數，透過 Amazon Linux 2 AMI 啟動 AWS。如需詳細資訊，請參閱《AWS Systems Manager 使用者指南》中的 [尋找公有參數](#)。
- `OperatorEmail`：您要傳送擴展活動通知的電子郵件地址。
- `SSHLocation`：可用於對執行個體進行 SSH 存取的 IP 地址範圍。

- Subnets：至少兩個公有子網路位於不同的可用區域。
- VPC：您帳戶中的虛擬私有雲端 (VPC)，可讓公有子網路中的資源連線至網際網路。

Note

您可以使用預設 VPC 和預設子網路，以允許執行個體存取網際網路。在使用自己的 VPC 的情況下，請確定您的 VPC 有一個子網路映射到您正在使用之區域的每個可用區域。必須至少有兩個可用的公有子網路，您才能建立負載平衡器。

此範本的下一部分會指定 Resources。此區段指定堆疊資源及其屬性。

[AWS::EC2::SecurityGroup](#) 資源 ELBSecurityGroup

- SecurityGroupIngress 包含 TCP 輸入規則，以允許從連接埠 80 上的所有 IP 地址進行存取 ("CidrIp" : "0.0.0.0/0")。

[AWS::EC2::SecurityGroup](#) 資源 EC2SecurityGroup

- SecurityGroupIngress 包含兩條輸入規則：1) 一條允許從您為 SSHLocation 輸入參數提供的 IP 地址範圍進行 SSH 存取 (連接埠 22) 的 TCP 輸入規則，以及 2) 一條透過指定負載平衡器的安全群組允許從負載平衡器進行存取的 TCP 輸入規則。[GetAtt](#) 函數用於獲取具有邏輯名稱 ELBSecurityGroup 的安全群組的 ID。

[AWS::ElasticLoadBalancingV2::TargetGroup](#) 資源 EC2TargetGroup

- Port、Protocol 和 HealthCheckProtocol 指定 EC2 執行個體連接埠 (80) 和協定 (HTTP)，ApplicationLoadBalancer 向該連接埠和協定路由流量而 Elastic Load Balancing 使用它們來檢查 EC2 執行個體的運作狀態。
- HealthCheckIntervalSeconds 指定 EC2 執行個體在運作狀態檢查間具有 30 秒的間隔。HealthCheckTimeoutSeconds 的定義為 Elastic Load Balancing 等待來自運作狀態檢查目標回應的時間長度 (此範例中為 15 秒)。在逾時期間過後，Elastic Load Balancing 便會將 EC2 執行個體的運作狀態檢查標記為不良。當 EC2 執行個體連續三次運作狀態檢查失敗時 (UnhealthyThresholdCount)，Elastic Load Balancing 將停止將流量路由到該 EC2 執行個體，直到其連續五次運作狀態檢查的結果為狀態良好為止 (HealthyThresholdCount)。此時，Elastic Load Balancing 會將執行個體視為狀態良好，並再次開始將流量路由到該執行個體。

- `TargetGroupAttributes` 將目標群組的取消註冊延遲值更新為 20 秒。依預設，Elastic Load Balancing 會等待 300 秒，然後再完成取消註冊程序。

[AWS::ElasticLoadBalancingV2::Listener](#) 資源 `ALBListener`

- `DefaultActions` 指定負載平衡器監聽的連接埠、負載平衡器轉寄請求的目標群組，以及用來路由請求的通訊協定。

[AWS::ElasticLoadBalancingV2::LoadBalancer](#) 資源 `ApplicationLoadBalancer`

- `Subnets` 將 `Subnets` 輸入參數的值做為公有子網路的清單，負載平衡器節點將在該清單中建立。
- `SecurityGroup` 獲取安全群組的 ID，該安全群組充當您的負載平衡器節點的虛擬防火牆，以控制傳入的流量。[GetAtt](#) 函數用於獲取具有邏輯名稱 `ELBSecurityGroup` 的安全群組的 ID。

[AWS::EC2::LaunchTemplate](#) 資源 `LaunchTemplate`

- `ImageId` 將 `LatestAmiId` 輸入參數的值做為要使用的 AMI。
- `KeyName` 將 `KeyName` 輸入參數的值做為要使用的 EC2 金鑰對。
- `SecurityGroupIds` 獲取具有邏輯名稱 `EC2SecurityGroup` 安全群組 ID，該安全群組充當您的 EC2 執行個體的虛擬防火牆，以控制傳入的流量。
- `UserData` 是在執行個體啟動並運行後執行的組態指令碼。在此範例中，指令碼會安裝 Apache，並建立 `index.html` 檔案。

[AWS::SNS::Topic](#) 資源 `NotificationTopic`

- 當有任何擴展活動時，`Subscription` 會將 `OperatorEmail` 輸入參數的值做為通知收件人的電子郵件地址。

[AWS::AutoScaling::AutoScalingGroup](#) 資源 `WebServerGroup`

- `MinSize` 及 `MaxSize` 設定 Auto Scaling 群組中 EC2 執行個體的最小及最大數。
- `TargetGroupARNs` 將接受具有邏輯名稱 `EC2TargetGroup` 的目標群組 ARN。當此 Auto Scaling 群組擴展時，它會自動向此目標群組註冊和取消註冊執行個體。
- `VPCZoneIdentifier` 將 `Subnets` 輸入參數的值做為公有子網路的清單，EC2 執行個體將在該清單中建立。

步驟 1：啟動堆疊

啟動堆疊之前，請檢查您是否具有使用下列所有服務的 AWS Identity and Access Management (IAM) 許可：Amazon EC2、Amazon EC2 Auto Scaling、AWS Systems Manager、Elastic Load Balancing、Amazon SNS 和 CloudFormation。

下列程序涉及從檔案上傳範例堆疊範本。在本機電腦上打開文字編輯器，然後新增其中一個範本。儲存檔案，並將其命名為 `sampleloadbalancedappstack.template`。

啟動堆疊範本

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 選擇 **Create stack (建立堆疊)**、**With new resources (standard)** (使用新資源 (標準))。
3. 在指定範本下，選擇上傳範本檔案，然後選擇選擇檔案以上傳 `sampleloadbalancedappstack.template` 檔案。
4. 選擇下一步。
5. 在指定詳細資訊頁面上，輸入堆疊名稱 (例如 **SampleLoadBalancedAppStack**)。
6. 在參數下，檢閱堆疊的參數並為無預設值的所有參數提供值，包括 `OperatorEmail`、`SSHLocation`、`KeyName`、`VPC` 和 `Subnets`。
7. 選擇 **Next (下一步)** 兩次。
8. 在檢視 頁面上，檢視和確認的設定。
9. 選擇提交。

您可以在狀態欄的 CloudFormation 主控台中檢視堆疊的狀態。當 CloudFormation 成功建立堆疊時，您會收到 `CREATE_COMPLETE` 狀態。

Note

在建立堆疊之後，您必須先確認訂閱，電子郵件地址才能開始接收通知。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 [在 Auto Scaling 群組擴展時取得 Amazon SNS 通知](#)。

步驟 2：清除您的資源範例

若要確保您不會為未使用的資源範例付費，請刪除堆疊。

刪除堆疊

1. 在 CloudFormation 主控台中，選取 SampleLoadBalancedAppStack 堆疊。
2. 選擇 刪除。
3. 在確認訊息中，選擇刪除堆疊。

SampleLoadBalancedAppStack 的狀態變更為 DELETE_IN_PROGRESS。當 CloudFormation 完成刪除堆疊時，會從清單中移除堆疊。

使用此逐步解說的範例範本建置您自己的堆疊範本。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的[教程：設定擴展和負載平衡的應用程式](#)。

在另一個 中使用 VPC 的對等 AWS 帳戶

您可以使用 與其他虛擬私有雲端 (VPC) AWS 帳戶 對等[AWS::EC2::VPCPeeringConnection](#)。這會在兩個 VPC 之間建立網路連線，讓您能在兩者之間路由流量，使其如同在相同的網路中通訊。VPC 對等互連有助於促進資料存取和資料傳輸。

若要建立 VPC 對等互連，您需要在單一 CloudFormation 堆疊內授權兩個獨立 AWS 帳戶 帳戶。

如需詳細資訊，請參閱《[Amazon VPC 互連指南](#)》中的 VPC 互連限制。

先決條件

1. 您需要對等互連的對等 VPC ID、對等 AWS 帳戶 ID 和[跨帳戶存取角色](#)。

Note

本演練涉及兩個帳戶：第一個是允許跨帳戶對等的帳戶 (「接受者帳戶」)。第二個是請求對等連線的帳戶 (「請求者帳戶」)。

2. 若要接受 VPC 對等互連，您必須擔任跨帳戶存取角色。該資源的行為方式與同一帳戶中的 VPC 對等互連資源相同。如需有關 IAM 管理員如何授予許可以擔任跨帳戶角色的資訊，請參閱《IAM 使用者指南》中的[向使用者授予切換角色的許可](#)。

步驟 1：建立 VPC 和跨帳戶角色

在此步驟中，您需要在「接受者帳戶」中建立 VPC 和角色。

建立 VPC 和跨帳戶存取權角色

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。
3. 對於先決條件 - 準備範本，選擇選擇現有範本，然後選擇上傳範本檔案、選擇檔案。
4. 在本機電腦上打開文字編輯器，然後新增其中一個下列範本。儲存檔案，並返回主控台將其選為範本檔案。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and an assumable role for cross account VPC peering.",
  "Parameters": {
    "PeerRequesterAccountId": {
      "Type": "String"
    }
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.1.0.0/16",
        "EnableDnsSupport": false,
        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
      }
    },
    "peerRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Principal": {
                "AWS": {
                  "Ref": "PeerRequesterAccountId"
                }
              }
            }
          ],
          "Action": [
```



```

        "sts:AssumeRole"
      ],
      "Effect": "Allow"
    }
  ]
},
"Path": "/",
"Policies": [
  {
    "PolicyName": "root",
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "ec2:AcceptVpcPeeringConnection",
          "Resource": "*"
        }
      ]
    }
  }
]
}
},
"Outputs": {
  "VPCId": {
    "Value": {
      "Ref": "vpc"
    }
  },
  "RoleARN": {
    "Value": {
      "Fn::GetAtt": [
        "peerRole",
        "Arn"
      ]
    }
  }
}
}
}

```

Example YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Principal:
              AWS: !Ref PeerRequesterAccountId
            Action:
              - 'sts:AssumeRole'
            Effect: Allow
      Path: /
      Policies:
        - PolicyName: root
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action: 'ec2:AcceptVpcPeeringConnection'
                Resource: '*'
Outputs:
  VPCId:
    Value: !Ref vpc
  RoleARN:
    Value: !GetAtt
      - peerRole
      - Arn
```

5. 選擇下一步。

6. 為堆疊命名 (例如, **VPC-owner**), 然後在 `PeerRequesterAccountId` 欄位中輸入請求者帳戶的 AWS 帳戶 ID。
7. 接受預設值, 然後選擇 **Next** (下一步)。
8. 選擇我確認 CloudFormation 可能會建立 IAM 資源, 然後選擇建立堆疊。

步驟 2：建立包含 **AWS::EC2::VPCPeeringConnection** 的範本

現在您已建立 VPC 和跨帳戶角色, 您可以使用另一個 AWS 帳戶 (申請者帳戶) 與 VPC 對等。

建立其中包含 [AWS::EC2::VPCPeeringConnection](#) 資源的範本

1. 返回 CloudFormation 主控台首頁。
2. 從堆疊頁面的右上角, 選擇建立堆疊, 並選擇使用新資源 (標準)。
3. 對於先決條件 - 準備範本, 選擇選擇現有範本, 然後選擇上傳範本檔案、選擇檔案。
4. 在本機電腦上打開文字編輯器, 然後新增其中一個下列範本。儲存檔案, 並返回主控台將其選為範本檔案。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a VPC and a VPC Peering connection using the PeerRole to accept.",
  "Parameters": {
    "PeerVPCAccountId": {
      "Type": "String"
    },
    "PeerVPCId": {
      "Type": "String"
    },
    "PeerRoleArn": {
      "Type": "String"
    }
  },
  "Resources": {
    "vpc": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.2.0.0/16",
        "EnableDnsSupport": false,
```

```

        "EnableDnsHostnames": false,
        "InstanceTenancy": "default"
    },
    "vpcPeeringConnection": {
        "Type": "AWS::EC2::VPCPeeringConnection",
        "Properties": {
            "VpcId": {
                "Ref": "vpc"
            },
            "PeerVpcId": {
                "Ref": "PeerVPCId"
            },
            "PeerOwnerId": {
                "Ref": "PeerVPCAccountId"
            },
            "PeerRoleArn": {
                "Ref": "PeerRoleArn"
            }
        }
    },
    "Outputs": {
        "VPCId": {
            "Value": {
                "Ref": "vpc"
            }
        },
        "VPCPeeringConnectionId": {
            "Value": {
                "Ref": "vpcPeeringConnection"
            }
        }
    }
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and a VPC Peering connection using the PeerRole to
  accept.
Parameters:
  PeerVPCAccountId:

```

```

    Type: String
  PeerVPCId:
    Type: String
  PeerRoleArn:
    Type: String
Resources:
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.2.0.0/16
      EnableDnsSupport: false
      EnableDnsHostnames: false
      InstanceTenancy: default
  vpcPeeringConnection:
    Type: AWS::EC2::VPCPeeringConnection
    Properties:
      VpcId: !Ref vpc
      PeerVpcId: !Ref PeerVPCId
      PeerOwnerId: !Ref PeerVPCAccountId
      PeerRoleArn: !Ref PeerRoleArn
Outputs:
  VPCId:
    Value: !Ref vpc
  VPCPeeringConnectionId:
    Value: !Ref vpcPeeringConnection

```

5. 選擇下一步。
6. 命名堆疊 (例如 **VPC-peering-connection**)。
7. 接受預設值，然後選擇 Next (下一步)。
8. 選擇我確認 CloudFormation 可能會建立 IAM 資源，然後選擇建立堆疊。

建立具有高度限制政策的範本

在將您的 VPC 與另一個 AWS 帳戶對等時，您可能需要建立高度限制的政策。

下列範例範本說明如何變更 VPC 對等擁有者範本 (在上述步驟 1 中建立的接受者帳戶)，使其限制更嚴格。

Example JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",

```

```

    "Description": "Create a VPC and an assumable role for cross account VPC peering.",
    "Parameters": {
      "PeerRequesterAccountId": {
        "Type": "String"
      }
    },
    "Resources": {
      "peerRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
          "AssumeRolePolicyDocument": {
            "Statement": [
              {
                "Action": [
                  "sts:AssumeRole"
                ],
                "Effect": "Allow",
                "Principal": {
                  "AWS": {
                    "Ref": "PeerRequesterAccountId"
                  }
                }
              }
            ]
          }
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyDocument": {
              "Statement": [
                {
                  "Action": "ec2:acceptVpcPeeringConnection",
                  "Effect": "Allow",
                  "Resource": {
                    "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}"
                  }
                },
                {
                  "Action": "ec2:acceptVpcPeeringConnection",
                  "Condition": {
                    "StringEquals": {
                      "ec2:AcceptorVpc": {
                        "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}"

```

```

        }
      }
    },
    "Effect": "Allow",
    "Resource": {
      "Fn::Sub": "arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-
peering-connection/*"
    }
  ],
  "Version": "2012-10-17"
},
"PolicyName": "root"
}
]
}
},
"vpc": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": "10.1.0.0/16",
    "EnableDnsHostnames": false,
    "EnableDnsSupport": false,
    "InstanceTenancy": "default"
  }
}
},
"Outputs": {
  "RoleARN": {
    "Value": {
      "Fn::GetAtt": [
        "peerRole",
        "Arn"
      ]
    }
  },
  "VPCId": {
    "Value": {
      "Ref": "vpc"
    }
  }
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a VPC and an assumable role for cross account VPC peering.
Parameters:
  PeerRequesterAccountId:
    Type: String
Resources:
  peerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Action:
              - 'sts:AssumeRole'
            Effect: Allow
            Principal:
              AWS:
                Ref: PeerRequesterAccountId
        Path: /
      Policies:
        - PolicyDocument:
            Statement:
              - Action: 'ec2:acceptVpcPeeringConnection'
                Effect: Allow
                Resource:
                  'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/${vpc}'
              - Action: 'ec2:acceptVpcPeeringConnection'
                Condition:
                  StringEquals:
                    'ec2:AcceptorVpc':
                      'Fn::Sub': 'arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc/
${vpc}'
                Effect: Allow
                Resource:
                  'Fn::Sub': >-
                    arn:aws:ec2:${AWS::Region}:${AWS::AccountId}:vpc-peering-
connection/*
            Version: 2012-10-17
            PolicyName: root
  vpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.1.0.0/16

```



```
    EnableDnsHostnames: false
    EnableDnsSupport: false
    InstanceTenancy: default
Outputs:
  RoleARN:
    Value:
      'Fn::GetAtt':
        - peerRole
        - Arn
  VPCId:
    Value:
      Ref: vpc
```

若要存取 VPC，您可以使用上述步驟 2 中使用的請求者範本。

如需詳細資訊，請參閱《Amazon VPC 對等互連指南》中的[適用於 VPC 對等互連的身分與存取管理](#)。

使用 CloudFormation 透過 CodeDeploy 進行 ECS 藍色/綠色部署

若要更新在 Amazon Elastic Container Service (Amazon ECS) 上執行的應用程式，可以使用 CodeDeploy 藍/綠部署策略。此策略有助於將變更應用程式版本所造成的中斷降至最低。

在藍/綠部署中，可在目前的即時環境 (稱為藍色) 旁邊建立新的應用程式環境 (稱為綠色)。這可讓您監控和測試綠色環境，然後再將即時流量從藍色環境路由到綠色環境。在綠色環境提供即時流量之後，您可以安全地終止藍色環境。

若要使用 CloudFormation 在 ECS 上執行 CodeDeploy 藍綠部署，請在堆疊範本中包含下列資訊：

- 描述 `AWS::CodeDeploy::BlueGreen` 勾點的 Hooks 區段。
- 指定 `AWS::CodeDeployBlueGreen` 轉換的 Transform 區段。

下列主題會引導您在 ECS 上設定藍/綠部署的 CloudFormation 範本。

主題

- [關於藍/綠部署](#)
- [使用 CloudFormation 管理 ECS 藍/綠部署時的考量事項](#)
- [AWS::CodeDeploy::BlueGreen 勾點語法](#)
- [藍/綠部署範本範例](#)

關於藍/綠部署

本主題概述如何使用 CloudFormation 執行藍/綠部署。它還說明如何為藍/綠部署準備 CloudFormation 範本。

主題

- [運作方式](#)
- [啟動綠色部署的資源更新](#)
- [準備範本以執行 ECS 藍/綠部署](#)
- [使用 CloudFormation 資源建模您的藍/綠部署](#)
- [變更集](#)
- [監控堆疊事件](#)
- [藍/綠部署的 IAM 許可](#)

運作方式

使用 CloudFormation 透過 CodeDeploy 執行 ECS 藍/綠部署時，首先要建立可定義藍色和藍/綠應用程式環境資源的堆疊範本，包括指定要使用的流量路由和穩定設定。接著，從該範本中建立堆疊。這會產生您的藍色 (目前) 應用程式。CloudFormation 只會在堆疊建立期間建立藍色資源。綠色部署的資源必須等到有需要時才會建立。

然後，如果在未來的堆疊更新中更新藍色應用程式中的工作定義或工作集資源，CloudFormation 會執行下列動作：

- 產生所有必要的綠色應用程式環境資源
- 根據指定的流量路由參數轉移流量
- 刪除藍色資源

如果在綠色部署成功並完成之前的任何時間點發生錯誤，CloudFormation 會將堆疊復原為起始整個綠色部署之前的狀態。

啟動綠色部署的資源更新

當您執行更新特定 ECS 資源特定屬性的堆疊更新時，CloudFormation 會啟動綠色部署程序。啟動此程序的資源包括：

- [AWS::ECS::TaskDefinition](#)
- [AWS::ECS::TaskSet](#)

不過，如果這些資源的更新不涉及需要取代的屬性變更，則不會啟動綠色部署。如需詳細資訊，請參閱[了解更新堆疊資源的行為](#)。

無法在同一堆疊更新中包含上述資源的更新與其他資源的更新。如果需要同時更新列出的資源和相同堆疊中的其他資源，您有兩個選項：

- 執行兩個獨立的堆疊更新操作：一個只包含上述資源的更新，另一個包含任何其他資源變更的單獨堆疊更新。
- 從範本中移除 Transform 和 Hooks 區段，然後執行堆疊更新。在此情況下，CloudFormation 將不會執行綠色部署。

準備範本以執行 ECS 藍/綠部署

若要在堆疊上啟用藍/綠部署，請在執行堆疊更新之前，在堆疊範本中包含下列區段。

- 將對 `AWS::CodeDeployBlueGreen` 轉換的參考新增到您的範本中：

```
"Transform": [  
  "AWS::CodeDeployBlueGreen"  
],
```

- 加入可叫用 `AWS::CodeDeploy::BlueGreen` 掛接的 Hooks 區段，並指定部署的內容。如需詳細資訊，請參閱[AWS::CodeDeploy::BlueGreen 勾點語法](#)。
- 在 Resources 區段中定義了您部署的藍色和綠色資源。

您可以在第一次建立範本時 (也就是在建立堆疊本身之前) 加入這些區段，也可以在執行堆疊更新之前將這些區段加入至現有範本。如果您為新堆疊指定藍/綠部署，則 CloudFormation 只會在堆疊建立期間建立藍色資源 - 在堆疊更新期間需要綠色部署的資源前，系統不會建立這些資源。

使用 CloudFormation 資源建模您的藍/綠部署

若要在 ECS 上執行 CodeDeploy 藍/綠部署，您的 CloudFormation 範本需要包含模型部署的資源，例如 Amazon ECS 服務和負載平衡器。如需有關這些資源所代表內容的詳細資訊，請見《AWS CodeDeploy 使用者指南》中[在開始 Amazon ECS 部署之前](#)。

需求	資源	必要/選用	取代時是否啟動藍/綠部署？
Amazon ECS 叢集	AWS::ECS::Cluster	選用。可以使用預設叢集。	否
Amazon ECS 服務	AWS::ECS::Service	必要。	否
應用程式或 Network Load Balancer	AWS::ECS::Service LoadBalancer	必要。	否
生產接聽程式	AWS::ElasticLoadBalancingV2::Listener	必要。	否
測試接聽程式	AWS::ElasticLoadBalancingV2::Listener	選用。	否
兩個目標群組	AWS::ElasticLoadBalancingV2::TargetGroup	必要。	否
Amazon ECS 任務定義	AWS::ECS::TaskDefinition	必要。	是
Amazon ECS 應用程式的容器	AWS::ECS::TaskDefinition Container Definition Name	必要。	否
取代任務集的連接埠	AWS::ECS::TaskDefinition PortMapping ContainerPort	必要。	否

變更集

強烈建議您在執行將起始綠色部署的堆疊更新之前建立變更集。這可讓您在執行堆疊更新前，查看將對堆疊所做的實際變更。請注意，資源變更可能不會以堆疊更新期間執行的順序列出。如需詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

監控堆疊事件

您可以使用 AWS CLI，在 Stack (堆疊) 頁面的 Events (事件) 索引標籤上檢視 ECS 部署每個步驟所產生的堆疊事件。如需詳細資訊，請參閱[監控堆疊進度](#)。

藍/綠部署的 IAM 許可

為了讓 CloudFormation 成功執行藍/綠部署，您必須具有以下 CodeDeploy 許可：

- `codedeploy:Get*`
- `codedeploy:CreateCloudFormationDeployment`

如需詳細資訊，請參閱服務授權參考中的 [CodeDeploy 的動作、資源和條件索引鍵](#)。

使用 CloudFormation 管理 ECS 藍/綠部署時的考量事項

使用 CloudFormation 透過 CodeDeploy 來執行 ECS 藍/綠部署的程序與僅使用 CodeDeploy 的標準 ECS 部署不同。如需詳細了解這些差異，請參閱《AWS CodeDeploy 使用者指南》中的[透過 CodeDeploy 和 CloudFormation 進行 Amazon ECS 藍/綠部署之間的差異](#)。

使用 CloudFormation 管理藍/綠部署時，請注意特定限制和考量事項：

- 只有特定資源的更新才會啟動綠色部署。如需詳細資訊，請參閱[啟動綠色部署的資源更新](#)。
- 無法在相同堆疊更新中包含觸發藍/綠部署的資源更新以及對其他資源的更新。如需詳細資訊，請參閱[啟動綠色部署的資源更新](#)。
- 您只能指定單一 ECS 服務作為部署目標。
- CodeDeploy 無法在綠色部署期間更新其值被 CloudFormation 模糊化的參數，這會導致錯誤和堆疊更新失敗。其中包含：
 - 使用 NoEcho 屬性定義的參數。
 - 使用動態參照從外部服務擷取其值的參數。如需動態參考的詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。
- 若要取消仍在進行中的綠色部署，請取消 CloudFormation 中的堆疊更新，而不是 CodeDeploy 或 ECS。如需詳細資訊，請參閱[取消堆疊更新](#)。更新完成之後，您便無法將其取消。不過，您可以使用任何之前的設定重新更新堆疊。
- 對於定義 ECS 藍/綠部署的範本，目前不支援下列 CloudFormation 功能：
 - 宣告輸出或使用 `Fn::ImportValue`，從其他堆疊中匯入值。
 - 匯入資源。如需有關匯入資源的詳細資訊，請參閱[將 AWS 資源匯入 CloudFormation 堆疊](#)。

- 在包含巢狀堆疊資源的範本中使用 `AWS::CodeDeploy::BlueGreen` 勾點。如需巢狀堆疊的詳細資訊，請參閱[運用巢狀堆疊，將範本分割成可重複使用的部分](#)。
- 在巢狀堆疊中使用 `AWS::CodeDeploy::BlueGreen` 勾點。

AWS::CodeDeploy::BlueGreen 勾點語法

下列語法說明 ECS 藍/綠部署的 `AWS::CodeDeploy::BlueGreen` 勾點結構。

語法

```
"Hooks": {
  "Logical ID": {
    "Type": "AWS::CodeDeploy::BlueGreen",
    "Properties": {
      "TrafficRoutingConfig": {
        "Type": "Traffic routing type",
        "TimeBasedCanary": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        },
        "TimeBasedLinear": {
          "StepPercentage": Integer,
          "BakeTimeMins": Integer
        }
      },
      "AdditionalOptions": {"TerminationWaitTimeInMinutes": Integer},
      "LifecycleEventHooks": {
        "BeforeInstall": "FunctionName",
        "AfterInstall": "FunctionName",
        "AfterAllowTestTraffic": "FunctionName",
        "BeforeAllowTraffic": "FunctionName",
        "AfterAllowTraffic": "FunctionName"
      },
      "ServiceRole": "CodeDeployServiceRoleName",
      "Applications": [
        {
          "Target": {
            "Type": "AWS::ECS::Service",
            "LogicalID": "Logical ID of AWS::ECS::Service"
          },
          "ECSAttributes": {
            "TaskDefinitions": [
```

```
"Logical ID of AWS::ECS::TaskDefinition (Blue)",  
  "Logical ID of AWS::ECS::TaskDefinition (Green)"  
],  
  "TaskSets": [  
    "Logical ID of AWS::ECS::TaskSet (Blue)",  
    "Logical ID of AWS::ECS::TaskSet (Green)"  
  ],  
  "TrafficRouting": {  
    "ProdTrafficRoute": {  
      "Type": "AWS::ElasticLoadBalancingV2::Listener",  
      "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener  
(Production)"  
    },  
    "TestTrafficRoute": {  
      "Type": "AWS::ElasticLoadBalancingV2::Listener",  
      "LogicalID": "Logical ID of AWS::ElasticLoadBalancingV2::Listener  
(Test)"  
    },  
    "TargetGroups": [  
      "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Blue)",  
      "Logical ID of AWS::ElasticLoadBalancingV2::TargetGroup (Green)"  
    ]  
  }  
}  
  
}  
  
]  
  
}  
  
}
```

Properties

邏輯 ID (也稱為邏輯名稱)

範本的 Hooks 區段中宣告之勾點的邏輯 ID。邏輯 ID 必須是英數字元 (A-Za-z0-9)，而且在範本內必須是唯一的。

必要：是

Type

掛接的類型。AWS::CodeDeploy::BlueGreen

必要：是

Properties

掛接的屬性。

必要：是

TrafficRoutingConfig

流量路由組態設定。

必要：否

預設組態是以時間為基礎的 Canary 流量轉移，具有 15% 步驟百分比和 5 分鐘的封裝時間。

Type

部署組態所使用的流量轉移類型。

有效值：AllAtOnce | TimeBasedCanary | TimeBasedLinear

必要：是

TimeBasedCanary

指定將流量以兩個增量為單位，從一個版本的部署轉移到另一個版本的規劃。

必要：有條件：如果您將 TimeBasedCanary 指定為流量路由類型，則必須包含 TimeBasedCanary 參數。

StepPercentage

在 TimeBasedCanary 部署的第一個增量中，要轉移的流量百分比。步驟百分比必須是 14% 或更大。

必要：否

BakeTimeMins

TimeBasedCanary 部署第一次和第二次流量轉移之間的分鐘數。

必要：否

TimeBasedLinear

指定將流量從一個版本的部署轉移到另一個版本的組態，每個增量之間的分鐘數相等。

必要：有條件：如果您將 TimeBasedLinear 指定為流量路由類型，則必須包含 TimeBasedLinear 參數。

StepPercentage

每個 TimeBasedLinear 部署增量開始時移動的流量百分比。步驟百分比必須是 14% 或更大。

必要：否

BakeTimeMins

TimeBasedLinear 部署的每個遞增流量轉移之間的分鐘數。

必要：否

AdditionalOptions

藍/綠部署的其他選項。

必要：否

TerminationWaitTimeInMinutes

指定終止藍色資源之前的等待時間 (以分鐘為單位)。

必要：否

LifecycleEventHooks

使用生命週期事件掛接來指定 CodeDeploy 可呼叫以驗證部署的 Lambda 函數。針對部署生命週期事件，您可以使用相同的函數或不同的函數。完成驗證測試後，Lambda AfterAllowTraffic 函數會回呼 CodeDeploy，並傳送 Succeeded 或 Failed 的結果。如需詳細資訊，請參閱 [《AWS CodeDeploy 使用者指南》](#) 中的 AppSpec 「勾點」 章節。

必要：否

BeforeInstall

在建立替換任務之前，用來執行任務的函數。

必要：否

AfterInstall

建立替代任務集後，用來執行任務，且與其中一個目標群組相關的函數。

必要：否

AfterAllowTestTraffic

在測試接聽程式將流量轉發至替換任務集後，用來執行任務的函數。

必要：否

BeforeAllowTraffic

在第二個目標群組與替換任務集建立關聯後，但在流量轉移到替換任務集前，用來執行任務的函數。

必要：否

AfterAllowTraffic

在第二個目標群組將流量轉發至替換任務集後，用來執行任務的函數。

必要：否

ServiceRole

CloudFormation 用來執行藍/綠部署的執行角色。如需必要許可的清單，請參閱 [藍/綠部署的 IAM 許可](#)。

必要：否

Applications

指定 Amazon ECS 應用程式的屬性。

必要：是

Target

必要：是

Type

資源的類型。

必要：是

LogicalID

資源的邏輯 ID。

必要：是

ECSAttributes

代表 Amazon ECS 應用程式部署各種需求的資源。

必要：是

TaskDefinitions

[AWS::ECS::TaskDefinition](#) 資源的邏輯 ID，以執行包含 Amazon ECS 應用程式的 Docker 容器。

必要：是

TaskSets

用作應用程式任務集的 [AWS::ECS::TaskSet](#) 資源邏輯 ID。

必要：是

TrafficRouting

指定用於流量路由的資源。

必要：是

ProdTrafficRoute

供負載平衡器用來將流量導向到您目標群組的接聽程式。

必要：是

Type

資源的類型。AWS::ElasticLoadBalancingV2::Listener

必要：是

LogicalID

資源的邏輯 ID。

必要：是

TestTrafficRoute

供負載平衡器用來將流量導向到您目標群組的接聽程式。

必要：是

Type

資源的類型。AWS::ElasticLoadBalancingV2::Listener

必要：是

LogicalID

資源的邏輯 ID。

必要：否

TargetGroups

用作目標群組的資源邏輯 ID，以將流量路由至已登記的目標。

必要：是

藍/綠部署範本範例

下列範例範本在 ECS 上設定 CodeDeploy 藍/綠部署，流量路由進度為每步驟 15%，每個步驟之間的穩定期間為 5 分鐘。

使用範本建立堆疊將佈建部署的初始組態。如果隨後對需要取代該資源之 BlueTaskSet 資源中的屬性進行任何變更，則 CloudFormation 將啟動綠色部署作為堆疊更新的一部分。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": { "Type": "AWS::EC2::VPC::Id" },
    "Subnet1": { "Type": "AWS::EC2::Subnet::Id" },
    "Subnet2": { "Type": "AWS::EC2::Subnet::Id" }
  },
  "Transform": [ "AWS::CodeDeployBlueGreen" ],
  "Hooks": {
    "CodeDeployBlueGreenHook": {
      "Type": "AWS::CodeDeploy::BlueGreen",
      "Properties": {
        "TrafficRoutingConfig": {
          "Type": "TimeBasedCanary",
          "TimeBasedCanary": {
            "StepPercentage": 15,
            "BakeTimeMins": 5
          }
        }
      }
    },
    "Applications": [
      {
        "Target": {
```

```

        "Type": "AWS::ECS::Service",
        "LogicalID": "ECSDemoService"
    },
    "ECSAttributes": {
        "TaskDefinitions": [ "BlueTaskDefinition", "GreenTaskDefinition" ],
        "TaskSets": [ "BlueTaskSet", "GreenTaskSet" ],
        "TrafficRouting": {
            "ProdTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
                "LogicalID": "ALBListenerProdTraffic"
            },
            "TargetGroups": [ "ALBTargetGroupBlue", "ALBTargetGroupGreen" ]
        }
    }
}
]
}
}
},
"Resources": {
    "ExampleSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Security group for ec2 access",
            "VpcId": { "Ref": "Vpc" },
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 8080,
                    "ToPort": 8080,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "CidrIp": "0.0.0.0/0"
                }
            ]
        }
    }
}

```

```

    ]
  }
},
"ALBTargetGroupBlue":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{ "HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{ "Ref":"Vpc" }
  }
},
"ALBTargetGroupGreen":{
  "Type":"AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties":{
    "HealthCheckIntervalSeconds":5,
    "HealthCheckPath":"/",
    "HealthCheckPort":"80",
    "HealthCheckProtocol":"HTTP",
    "HealthCheckTimeoutSeconds":2,
    "HealthyThresholdCount":2,
    "Matcher":{ "HttpCode":"200" },
    "Port":80,
    "Protocol":"HTTP",
    "Tags":[{ "Key":"Group","Value":"Example" }],
    "TargetType":"ip",
    "UnhealthyThresholdCount":4,
    "VpcId":{ "Ref":"Vpc" }
  }
},
"ExampleALB":{
  "Type":"AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties":{
    "Scheme":"internet-facing",
    "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],

```

```

        "Subnets": [{ "Ref": "Subnet1" }, { "Ref": "Subnet2" } ],
        "Tags": [{ "Key": "Group", "Value": "Example" } ],
        "Type": "application",
        "IpAddressType": "ipv4"
    }
},
"ALBListenerProdTraffic": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
        "DefaultActions": [
            {
                "Type": "forward",
                "ForwardConfig": {
                    "TargetGroups": [
                        {
                            "TargetGroupArn": { "Ref": "ALBTargetGroupBlue" },
                            "Weight": 1
                        }
                    ]
                }
            }
        ],
        "LoadBalancerArn": { "Ref": "ExampleALB" },
        "Port": 80,
        "Protocol": "HTTP"
    }
},
"ALBListenerProdRule": {
    "Type": "AWS::ElasticLoadBalancingV2::ListenerRule",
    "Properties": {
        "Actions": [
            {
                "Type": "forward",
                "ForwardConfig": {
                    "TargetGroups": [
                        {
                            "TargetGroupArn": { "Ref": "ALBTargetGroupBlue" },
                            "Weight": 1
                        }
                    ]
                }
            }
        ],
        "Conditions": [

```

```

        {
            "Field": "http-header",
            "HTTPHeaderConfig": {
                "HttpHeaderName": "User-Agent",
                "Values": [ "Mozilla" ]
            }
        }
    ],
    "ListenerArn": { "Ref": "ALBListenerProdTraffic" },
    "Priority": 1
}
},
"ECSTaskExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ecs-tasks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    ]
},
"ManagedPolicyArns": [ "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" ]
},
"BlueTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
        "ExecutionRoleArn": {
            "Fn::GetAtt": [ "ECSTaskExecutionRole", "Arn" ]
        },
        "ContainerDefinitions": [
            {
                "Name": "DemoApp",
                "Image": "nginxdemos/hello:latest",
                "Essential": true,
                "PortMappings": [

```



```
        {
            "HostPort":80,
            "Protocol":"tcp",
            "ContainerPort":80
        }
    ]
}
],
"RequiresCompatibilities":[ "FARGATE" ],
"NetworkMode":"awsvpc",
"Cpu":"256",
"Memory":"512",
"Family":"ecs-demo"
}
},
"ECSDemoCluster":{
    "Type":"AWS::ECS::Cluster",
    "Properties":{}
},
"ECSDemoService":{
    "Type":"AWS::ECS::Service",
    "Properties":{
        "Cluster":{"Ref":"ECSDemoCluster" },
        "DesiredCount":1,
        "DeploymentController":{"Type":"EXTERNAL" }
    }
},
"BlueTaskSet":{
    "Type":"AWS::ECS::TaskSet",
    "Properties":{
        "Cluster":{"Ref":"ECSDemoCluster" },
        "LaunchType":"FARGATE",
        "NetworkConfiguration":{
            "AwsVpcConfiguration":{
                "AssignPublicIp":"ENABLED",
                "SecurityGroups":[{ "Ref":"ExampleSecurityGroup" }],
                "Subnets":[{ "Ref":"Subnet1" },{ "Ref":"Subnet2" }]
            }
        }
    },
    "PlatformVersion":"1.4.0",
    "Scale":{
        "Unit":"PERCENT",
        "Value":100
    }
},
```

```

    "Service":{ "Ref":"ECSDemoService"},
    "TaskDefinition":{ "Ref":"BlueTaskDefinition" },
    "LoadBalancers":[
      {
        "ContainerName":"DemoApp",
        "ContainerPort":80,
        "TargetGroupArn":{ "Ref":"ALBTargetGroupBlue" }
      }
    ]
  },
  "PrimaryTaskSet":{
    "Type":"AWS::ECS::PrimaryTaskSet",
    "Properties":{
      "Cluster":{ "Ref":"ECSDemoCluster" },
      "Service":{ "Ref":"ECSDemoService" },
      "TaskSetId":{ "Fn::GetAtt":[ "BlueTaskSet","Id" ]
    }
  }
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  Vpc:
    Type: AWS::EC2::VPC::Id
  Subnet1:
    Type: AWS::EC2::Subnet::Id
  Subnet2:
    Type: AWS::EC2::Subnet::Id
Transform:
  - 'AWS::CodeDeployBlueGreen'
Hooks:
  CodeDeployBlueGreenHook:
    Type: AWS::CodeDeploy::BlueGreen
    Properties:
      TrafficRoutingConfig:
        Type: TimeBasedCanary
      TimeBasedCanary:
        StepPercentage: 15

```

```
    BakeTimeMins: 5
  Applications:
    - Target:
        Type: AWS::ECS::Service
        LogicalID: ECSDemoService
    ECSAttributes:
      TaskDefinitions:
        - BlueTaskDefinition
        - GreenTaskDefinition
      TaskSets:
        - BlueTaskSet
        - GreenTaskSet
      TrafficRouting:
        ProdTrafficRoute:
          Type: AWS::ElasticLoadBalancingV2::Listener
          LogicalID: ALBListenerProdTraffic
        TargetGroups:
          - ALBTargetGroupBlue
          - ALBTargetGroupGreen
```

Resources:

```
ExampleSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for ec2 access
    VpcId: !Ref Vpc
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 8080
        ToPort: 8080
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
  ALBTargetGroupBlue:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      HealthCheckIntervalSeconds: 5
      HealthCheckPath: /
      HealthCheckPort: '80'
```

```
HealthCheckProtocol: HTTP
HealthCheckTimeoutSeconds: 2
HealthyThresholdCount: 2
Matcher:
  HttpCode: '200'
Port: 80
Protocol: HTTP
Tags:
  - Key: Group
    Value: Example
TargetType: ip
UnhealthyThresholdCount: 4
VpcId: !Ref Vpc
ALBTargetGroupGreen:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 5
    HealthCheckPath: /
    HealthCheckPort: '80'
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 2
    HealthyThresholdCount: 2
    Matcher:
      HttpCode: '200'
    Port: 80
    Protocol: HTTP
    Tags:
      - Key: Group
        Value: Example
    TargetType: ip
    UnhealthyThresholdCount: 4
    VpcId: !Ref Vpc
ExampleALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    SecurityGroups:
      - !Ref ExampleSecurityGroup
    Subnets:
      - !Ref Subnet1
      - !Ref Subnet2
    Tags:
      - Key: Group
        Value: Example
```

```
Type: application
IpAddressType: ipv4
ALBListenerProdTraffic:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
      ForwardConfig:
        TargetGroups:
          - TargetGroupArn: !Ref ALBTargetGroupBlue
            Weight: 1
    LoadBalancerArn: !Ref ExampleALB
    Port: 80
    Protocol: HTTP
ALBListenerProdRule:
  Type: AWS::ElasticLoadBalancingV2::ListenerRule
  Properties:
    Actions:
      - Type: forward
      ForwardConfig:
        TargetGroups:
          - TargetGroupArn: !Ref ALBTargetGroupBlue
            Weight: 1
    Conditions:
      - Field: http-header
      HttpHeaderConfig:
        HttpHeaderName: User-Agent
        Values:
          - Mozilla
    ListenerArn: !Ref ALBListenerProdTraffic
    Priority: 1
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Sid: ''
          Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy'
```

```
BlueTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ExecutionRoleArn: !GetAtt
      - ECSTaskExecutionRole
      - Arn
    ContainerDefinitions:
      - Name: DemoApp
        Image: 'nginxdemos/hello:latest'
        Essential: true
        PortMappings:
          - HostPort: 80
            Protocol: tcp
            ContainerPort: 80
    RequiresCompatibilities:
      - FARGATE
    NetworkMode: awsvpc
    Cpu: '256'
    Memory: '512'
    Family: ecs-demo
ECSDemoCluster:
  Type: AWS::ECS::Cluster
  Properties: {}
ECSDemoService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref ECSDemoCluster
    DesiredCount: 1
    DeploymentController:
      Type: EXTERNAL
BlueTaskSet:
  Type: AWS::ECS::TaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
      SecurityGroups:
        - !Ref ExampleSecurityGroup
      Subnets:
        - !Ref Subnet1
        - !Ref Subnet2
    PlatformVersion: 1.4.0
```

```
Scale:
  Unit: PERCENT
  Value: 100
Service: !Ref ECSDemoService
TaskDefinition: !Ref BlueTaskDefinition
LoadBalancers:
  - ContainerName: DemoApp
    ContainerPort: 80
    TargetGroupArn: !Ref ALBTargetGroupBlue
PrimaryTaskSet:
  Type: AWS::ECS::PrimaryTaskSet
  Properties:
    Cluster: !Ref ECSDemoCluster
    Service: !Ref ECSDemoService
    TaskSetId: !GetAtt
      - BlueTaskSet
      - Id
```

CloudFormation 範本程式碼片段

本節提供數個範例藍本，可用來了解如何宣告各種 CloudFormation 範本部分。您也可以使用程式碼片段做為自訂範本各區段的起點。

Note

由於 CloudFormation 範本必須符合 JSON 規範，因此沒有行接續字元的佈建。如果行超過 68 個字元，則包裝本文件中的程式碼片段可能是隨機的。

主題

- [一般範本程式碼片段](#)
- [自動擴展 CloudFormation 範本程式碼片段](#)
- [AWS 帳單主控台範本程式碼片段](#)
- [CloudFormation 範本程式碼片段](#)
- [Amazon CloudFront 範本程式碼片段](#)
- [Amazon CloudWatch 範本程式碼片段](#)
- [Amazon CloudWatch Logs 範本程式碼片段](#)
- [Amazon DynamoDB 範本程式碼片段](#)

- [Amazon EC2 CloudFormation 範本程式碼片段](#)
- [Amazon Elastic Container Service 範例範本](#)
- [Amazon Elastic File System 範例範本](#)
- [Elastic Beanstalk 範本程式碼片段](#)
- [Elastic Load Balancing 範本程式碼片段](#)
- [AWS Identity and Access Management 範本程式碼片段](#)
- [AWS Lambda 範本](#)
- [Amazon Redshift 範本程式碼片段](#)
- [Amazon RDS 範本程式碼片段](#)
- [Route 53 範本程式碼片段](#)
- [Amazon S3 範本程式碼片段](#)
- [Amazon SNS 範本程式碼片段](#)
- [Amazon SQS 範本程式碼片段](#)
- [Amazon Timestream 範本程式碼片段](#)

一般範本程式碼片段

以下範例說明非 AWS 服務專屬的不同 CloudFormation 範本功能。

主題

- [經 Base64 編碼的 UserData 屬性](#)
- [經 Base64 編碼的 UserData 屬性，包含 AccessKey 和 SecretKey](#)
- [具有一個文字字串參數的 Parameters 區段](#)
- [字串參數使用一般表達式限制條件的 Parameters 區段](#)
- [數值參數使用 MinValue 和 MaxValue 限制條件的 Parameters 區段](#)
- [數值參數使用 AllowedValues 限制條件的 Parameters 區段](#)
- [具有一個文字 CommaDelimitedList 參數的 Parameters 區段](#)
- [參數值以虛擬參數為基礎的 Parameters 區段](#)
- [具有三個映射的 Mapping 區段](#)
- [根據常值字串的 Description](#)
- [具有一個文字字串輸出的 Outputs 區段](#)
- [具有一個資源參考和一個虛擬參考輸出的 Outputs 區段](#)

- [具有輸出的 Outputs 區段，此輸出以函數、文字字串、參考和虛擬參數為基礎](#)
- [範本格式版本](#)
- [AWS Tags 屬性](#)

經 Base64 編碼的 UserData 屬性

此範例示範使用 Fn::Base64 和 Fn::Join 函數的 UserData 屬性。參考 MyValue 和 MyName 是必須在範本 Parameters 區段中定義的參數。文字字串 Hello World 只是本範例傳入，屬於 UserData 的另一個值。

JSON

```
"UserData" : {
  "Fn::Base64" : {
    "Fn::Join" : [ ",", [
      { "Ref" : "MyValue" },
      { "Ref" : "MyName" },
      "Hello World" ] ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    Ref: MyValue
    Ref: MyName
    Hello World
```

經 Base64 編碼的 UserData 屬性，包含 AccessKey 和 SecretKey

此範例示範使用 Fn::Base64 和 Fn::Join 函數的 UserData 屬性。它包含 AccessKey 和 SecretKey 資訊。參考 AccessKey 和 SecretKey 是必須在範本 Parameters (參數) 區段中定義的參數。

JSON

```
"UserData" : {
  "Fn::Base64" : {
```

```

        "Fn::Join" : [ "", [
            "ACCESS_KEY=", { "Ref" : "AccessKey" },
            "SECRET_KEY=", { "Ref" : "SecretKey" } ]
        ]
    }
}

```

YAML

```

UserData:
  Fn::Base64: !Sub |
    ACCESS_KEY=${AccessKey}
    SECRET_KEY=${SecretKey}

```

具有一個文字字串參數的 Parameters 區段

下列範例說明有效的 Parameters (參數) 區段宣告，在此宣告單一 String 類型參數。

JSON

```

"Parameters" : {
    "UserName" : {
        "Type" : "String",
        "Default" : "nonadmin",
        "Description" : "Assume a vanilla user if no command-line spec provided"
    }
}

```

YAML

```

Parameters:
  UserName:
    Type: String
    Default: nonadmin
    Description: Assume a vanilla user if no command-line spec provided

```

字串參數使用一般表達式限制條件的 Parameters 區段

下列範例說明有效的 Parameters (參數) 區段宣告，在此宣告單一 String 類型參數。AdminUserAccount 參數的預設值為 admin。此參數值長度下限為 1 個字元、長度上限為 16 個字元，而且包含英數字元，但必須以字母字元開頭。

JSON

```
"Parameters" : {
  "AdminUserAccount": {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "The admin account user name",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "16",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
  }
}
```

YAML

```
Parameters:
  AdminUserAccount:
    Default: admin
    NoEcho: true
    Description: The admin account user name
    Type: String
    MinLength: 1
    MaxLength: 16
    AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
```

數值參數使用 MinValue 和 MaxValue 限制條件的 Parameters 區段

下列範例說明有效的 Parameters (參數) 區段宣告，在此宣告單一 Number 類型參數。WebServerPort 參數的預設值為 80，最小值 1 和最大值 65535。

JSON

```
"Parameters" : {
  "WebServerPort": {
    "Default": "80",
    "Description" : "TCP/IP port for the web server",
    "Type": "Number",
    "MinValue": "1",
    "MaxValue": "65535"
  }
}
```

YAML

```
Parameters:
  WebServerPort:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    MinValue: 1
    MaxValue: 65535
```

數值參數使用 AllowedValues 限制條件的 Parameters 區段

下列範例說明有效的 Parameters (參數) 區段宣告，在此宣告單一 Number 類型參數。WebServerPort 參數的預設值為 80，僅允許值 80 和 8888。

JSON

```
"Parameters" : {
  "WebServerPortLimited": {
    "Default": "80",
    "Description" : "TCP/IP port for the web server",
    "Type": "Number",
    "AllowedValues" : ["80", "8888"]
  }
}
```

YAML

```
Parameters:
  WebServerPortLimited:
    Default: 80
    Description: TCP/IP port for the web server
    Type: Number
    AllowedValues:
      - 80
      - 8888
```

具有一個文字 CommaDelimitedList 參數的 Parameters 區段

下列範例說明有效的 Parameters 區段宣告，在此宣告單一 CommaDelimitedList 類型參數。NoEcho 屬性被設定為 TRUE，這將在 describe-stacks 輸出中使用星號 (*****) 遮罩其值，但儲存在下列指定位置的資訊除外。

⚠ Important

使用 NoEcho 屬性不會遮罩任何儲存在下列資訊中的資訊：

- Metadata 範本區段。CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。若要取得更多資訊，請參閱[中繼資料](#)。
- Outputs 範本區段。如需詳細資訊，請參閱[輸出](#)。
- 資源定義的 Metadata 屬性。如需詳細資訊，請參閱[Metadata 屬性](#)。

我們強烈建議您不要使用這些機制來包含敏感資訊，例如密碼或秘密。

⚠ Important

我們建議您不要直接在 CloudFormation 範本中嵌入敏感資訊，而是在堆疊範本中使用動態參數來參考在 CloudFormation 外部存放和管理的敏感資訊，例如在 AWS Systems Manager 參數存放區或中 AWS Secrets Manager。

如需詳細資訊，請參閱[請勿在範本中內嵌認證](#)的最佳實務。

JSON

```
"Parameters" : {  
  "UserRoles" : {  
    "Type" : "CommaDelimitedList",  
    "Default" : "guest,newhire",  
    "NoEcho" : "TRUE"  
  }  
}
```

YAML

```
Parameters:  
  UserRoles:  
    Type: CommaDelimitedList  
    Default: "guest,newhire"  
    NoEcho: true
```

參數值以虛擬參數為基礎的 Parameters 區段

以下範例顯示 EC2 使用者資料中使用虛擬參數 `AWS::StackName` 和 `AWS::Region` 的命令。如需這些虛擬參數的詳細資訊，請參閱[使用虛擬參數取得 AWS 值](#)。

JSON

```
"UserData"      : { "Fn::Base64" : { "Fn::Join" : ["", [
    "#!/bin/bash -xe\n",
    "yum install -y aws-cfn-bootstrap\n",

    "/opt/aws/bin/cfn-init -v ",
    "      --stack ", { "Ref" : "AWS::StackName" },
    "      --resource LaunchConfig ",
    "      --region ", { "Ref" : "AWS::Region" }, "\n",

    "/opt/aws/bin/cfn-signal -e $? ",
    "      --stack ", { "Ref" : "AWS::StackName" },
    "      --resource WebServerGroup ",
    "      --region ", { "Ref" : "AWS::Region" }, "\n"
  ]]]}
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    yum update -y aws-cfn-bootstrap
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource LaunchConfig --
region ${AWS::Region}
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource WebServerGroup
--region ${AWS::Region}
```

具有三個映射的 Mapping 區段

以下範例說明包含三個映射之有效的 Mapping 區段宣告。符合 Stop、SlowDown 或 Go 映射金鑰時，映射會提供指派給所對應 RGBColor 屬性的 RGB 值。

JSON

```
"Mappings" : {
```

```
"LightColor" : {
  "Stop" : {
    "Description" : "red",
    "RGBColor" : "RED 255 GREEN 0 BLUE 0"
  },
  "SlowDown" : {
    "Description" : "yellow",
    "RGBColor" : "RED 255 GREEN 255 BLUE 0"
  },
  "Go" : {
    "Description" : "green",
    "RGBColor" : "RED 0 GREEN 128 BLUE 0"
  }
}
```

YAML

```
Mappings:
  LightColor:
    Stop:
      Description: red
      RGBColor: "RED 255 GREEN 0 BLUE 0"
    SlowDown:
      Description: yellow
      RGBColor: "RED 255 GREEN 255 BLUE 0"
    Go:
      Description: green
      RGBColor: "RED 0 GREEN 128 BLUE 0"
```

根據常值字串的 Description

以下範例說明有效的 Description 區段宣告，其值是以文字字串為基礎。此程式碼片段適用於範本、參數、資源、屬性或輸出。

JSON

```
"Description" : "Replace this value"
```

YAML

```
Description: "Replace this value"
```

具有一個文字字串輸出的 Outputs 區段

此範例示範以文字字串為基礎的輸出指派。

JSON

```
"Outputs" : {
  "MyPhone" : {
    "Value" : "Please call 555-5555",
    "Description" : "A random message for aws cloudformation describe-stacks"
  }
}
```

YAML

```
Outputs:
  MyPhone:
    Value: Please call 555-5555
    Description: A random message for aws cloudformation describe-stacks
```

具有一個資源參考和一個虛擬參考輸出的 Outputs 區段

此範例示範具有兩個輸出指派的 Outputs 區段。一個以資源為基礎，另一個以虛擬參考為基礎。

JSON

```
"Outputs" : {
  "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
  "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

YAML

```
Outputs:
  SNSTopic:
    Value: !Ref MyNotificationTopic
  StackName:
    Value: !Ref AWS::StackName
```


具有輸出的 Outputs 區段，此輸出以函數、文字字串、參考和虛擬參數為基礎

此範例示範具有一個輸出指派的 Outputs (輸出) 區段。使用 Join 函數串連值，百分比符號為分隔符號。

JSON

```
"Outputs" : {
  "MyOutput" : {
    "Value" : { "Fn::Join" :
      [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
    }
  }
}
```

YAML

```
Outputs:
  MyOutput:
    Value: !Join [ %, [ 'A-string', !Ref 'AWS::StackName' ] ]
```

範本格式版本

下列程式碼片段說明有效的 AWSTemplateFormatVersion 區段宣告。

JSON

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

YAML

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

AWS Tags 屬性

此範例顯示 屬性 AWS Tags。您可以在資源的 Properties (屬性) 區段內指定此屬性。資源建立後，會以您宣告的標籤標記。

JSON

```
"Tags" : [
```

```
{
  "Key" : "keyname1",
  "Value" : "value1"
},
{
  "Key" : "keyname2",
  "Value" : "value2"
}
]
```

YAML

```
Tags:
-
  Key: "keyname1"
  Value: "value1"
-
  Key: "keyname2"
  Value: "value2"
```

自動擴展 CloudFormation 範本程式碼片段

藉助 Amazon EC2 Auto Scaling，您可以透過擴展政策或排程擴展來自動擴展 Amazon EC2 執行個體。Auto Scaling 群組是 Amazon EC2 執行個體的集合，可支援自動擴展和機群管理功能，例如擴展政策、排程動作、運作狀態檢查、lifecycle hook 和負載均衡。

Application Auto Scaling 可透過擴展政策或排程擴展，提供 Amazon EC2 以外的不同資源的自動擴展。

您可以使用 CloudFormation 範本建立和設定 Auto Scaling 群組、擴展政策、排程動作和其他自動擴展資源，做為基礎設施的一部分。範本可讓您以可重複且一致的方式，輕鬆管理和自動化自動擴展資源的部署。

下列範例範本程式碼片段說明 Amazon EC2 Auto Scaling 和 Application Auto Scaling CloudFormation 的資源或元件。這些程式碼片段設計用於整合至範本中，而且並非設計為獨立執行。

程式碼片段類別

- [使用 設定 Amazon EC2 Auto Scaling 資源 CloudFormation](#)
- [使用 設定 Application Auto Scaling 資源 CloudFormation](#)

使用 設定 Amazon EC2 Auto Scaling 資源 CloudFormation

下列範例示範要在範本中包含的不同程式碼片段，以與 Amazon EC2 Auto Scaling 搭配使用。

程式碼片段類別

- [建立單一執行個體 Auto Scaling 群組](#)
- [透過連接的負載平衡器建立 Auto Scaling 群組](#)
- [透過通知建立 Auto Scaling 群組](#)
- [建立使用 CreationPolicy 和 UpdatePolicy 的 Auto Scaling 群組](#)
- [建立步驟擴展政策](#)
- [混合執行個體群組範例](#)
- [啟動組態範例](#)

建立單一執行個體 Auto Scaling 群組

此範例顯示具有單一執行個體的 [AWS::AutoScaling::AutoScalingGroup](#) 資源，以協助您開始使用。Auto Scaling 群組的 `VPCZoneIdentifier` 屬性指定三個不同可用區域中的現有子網路清單。您必須先從帳戶指定適用的子網路 ID，然後才能建立堆疊。LaunchTemplate 屬性參考 [AWS::EC2::LaunchTemplate](#) 資源，該資源具有範本中其他處所定義的邏輯名稱 `myLaunchTemplate`。

Note

如需啟動範本的範例，請參閱 Amazon EC2 程式碼片段章節中的 [透過 CloudFormation 建立啟動範本](#)，以及 `AWS::EC2::LaunchTemplate` 資源中的 [範例](#) 章節。

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
```

```

    },
    "Version" : {
      "Fn::GetAtt" : [
        "myLaunchTemplate",
        "LatestVersionNumber"
      ]
    }
  },
  "MaxSize" : "1",
  "MinSize" : "1"
}
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '1'
    MinSize: '1'

```

透過連接的負載平衡器建立 Auto Scaling 群組

此範例顯示 [AWS::AutoScaling::AutoScalingGroup](#) 資源，適用於多個伺服器之間的負載平衡。它指定在相同範本中其他位置宣告 AWS 的資源邏輯名稱。

1. `VPCZoneIdentifier` 屬性指定兩個 [AWS::EC2::Subnet](#) 資源的邏輯名稱 (將在其中建立 Auto Scaling 群組的 EC2 執行個體) : `myPublicSubnet1` 和 `myPublicSubnet2`。
2. `LaunchTemplate` 屬性指定 [AWS::EC2::LaunchTemplate](#) 資源，該資源具有邏輯名稱 `myLaunchTemplate`。
3. `TargetGroupARNs` 屬性針對將流量路由到 Auto Scaling 群組所使用的 Application Load Balancer 或 Network Load Balancer，列出目標群組。在此範例中，指定了一個目標群組，[AWS::ElasticLoadBalancingV2::TargetGroup](#) 資源，該資源具有邏輯名稱 `myTargetGroup`。

JSON

```
"myServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ { "Ref" : "myPublicSubnet1" }, { "Ref" :
"myPublicSubnet2" } ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "myLaunchTemplate"
      },
      "Version" : {
        "Fn::GetAtt" : [
          "myLaunchTemplate",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "5",
    "MinSize" : "1",
    "TargetGroupARNs" : [ { "Ref" : "myTargetGroup" } ]
  }
}
```

YAML

```
myServerGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref myPublicSubnet1
      - !Ref myPublicSubnet2
    LaunchTemplate:
      LaunchTemplateId: !Ref myLaunchTemplate
      Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    TargetGroupARNs:
      - !Ref myTargetGroup
```

另請參閱

如需根據適用於 Application Load Balancer 的 ALBRequestCountPerTarget 預先定義指標建立具有目標追蹤擴展政策之 Auto Scaling 群組的詳細範例，請參閱 [AWS::AutoScaling::ScalingPolicy](#) 資源中的 [範例](#) 章節。

透過通知建立 Auto Scaling 群組

此範例顯示的 [AWS::AutoScaling::AutoScalingGroup](#) 資源會在指定的事件發生時傳送 Amazon SNS 通知。NotificationConfigurations 屬性指定 SNS 主題，其中 CloudFormation 會傳送通知，以及會導致 CloudFormation 傳送通知的事件。當指定的事件 NotificationTypes 發生時，CloudFormation 會傳送通知至指定的 SNS 主題 TopicARN。當您啟動堆疊時，CloudFormation 會建立在相同範本中宣告 [AWS::SNS::Subscription](#) 的資源 (snsTopicForAutoScalingGroup)。

Auto Scaling 群組的 VPCZoneIdentifier 屬性指定三個不同可用區域中的現有子網路清單。您必須先從帳戶指定適用的子網路 ID，然後才能建立堆疊。LaunchTemplate 屬性參考在相同範本中其他位置宣告的 [AWS::EC2::LaunchTemplate](#) 資源邏輯名稱。

JSON

```
"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "DependsOn": [
    "snsTopicForAutoScalingGroup"
  ],
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate" : {
      "LaunchTemplateId" : {
        "Ref" : "logicalName"
      },
      "Version" : {
        "Fn::GetAtt" : [
          "logicalName",
          "LatestVersionNumber"
        ]
      }
    },
    "MaxSize" : "5",
    "MinSize" : "1",
    "NotificationConfigurations" : [
      {
```

```

    "TopicARN" : { "Ref" : "snsTopicForAutoScalingGroup" },
    "NotificationTypes" : [
      "autoscaling:EC2_INSTANCE_LAUNCH",
      "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
      "autoscaling:EC2_INSTANCE_TERMINATE",
      "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
      "autoscaling:TEST_NOTIFICATION"
    ]
  }
]
}
}
}
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - snsTopicForAutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref logicalName
      Version: !GetAtt logicalName.LatestVersionNumber
    MaxSize: '5'
    MinSize: '1'
    NotificationConfigurations:
      - TopicARN: !Ref snsTopicForAutoScalingGroup
        NotificationTypes:
          - autoscaling:EC2_INSTANCE_LAUNCH
          - autoscaling:EC2_INSTANCE_LAUNCH_ERROR
          - autoscaling:EC2_INSTANCE_TERMINATE
          - autoscaling:EC2_INSTANCE_TERMINATE_ERROR
          - autoscaling:TEST_NOTIFICATION

```

建立使用 **CreationPolicy** 和 **UpdatePolicy** 的 Auto Scaling 群組

下列範例顯示如何將 CreationPolicy 和 UpdatePolicy 屬性新增至 [AWS::AutoScaling::AutoScalingGroup](#) 資源。

範例建立政策可防止 Auto Scaling 群組達到CREATE_COMPLETE狀態，直到群組準備就緒時 CloudFormation 收到成功訊號Count的數量為止。若要發出 Auto Scaling 群組已就緒的訊號，需在執行個體上執行新增到啟動範本使用者資料 (未顯示) 的 cfn-signal 協助程式指令碼。如果執行個體未在指定的 Timeout 內傳送訊號，CloudFormation 將假設未建立執行個體、資源建立失敗，且 CloudFormation 將復原堆疊。

範例更新政策使用 AutoScalingRollingUpdate 屬性指示 CloudFormation，以執行滾動更新。滾動更新根據 MaxBatchSize 和以 PauseTime 為基礎的更新批次間的暫停時間，小量批次變更 Auto Scaling 群組 (在此範例中為依執行個體逐一變更)。MinInstancesInService 屬性指定在 CloudFormation 更新舊執行個體時，Auto Scaling 群組中必須為服務中的執行個體的最少數量。

WaitOnResourceSignals 屬性會設定為 true。CloudFormation 必須在指定的 PauseTime 內先接到每個新執行個體的訊號，才會繼續更新。當堆疊正在更新時，會暫停以下 EC2 Auto Scaling 程序：HealthCheck、ReplaceUnhealthy、AZRebalance、AlarmNotification 和 ScheduledActions。注意：請勿暫停 Launch、Terminate 或 AddToLoadBalancer (如果 Auto Scaling 群組與 Elastic Load Balancing 搭配使用) 程序類型，因為這麼做會讓滾動更新無法正常運作。

Auto Scaling 群組的 VPCZoneIdentifier 屬性指定三個不同可用區域中的現有子網路清單。您必須先從帳戶指定適用的子網路 ID，然後才能建立堆疊。LaunchTemplate 屬性參考在相同範本中其他位置宣告的 [AWS::EC2::LaunchTemplate](#) 資源邏輯名稱。

如需 CreationPolicy 和 UpdatePolicy 屬性的詳細資訊，請參閱[資源屬性參考](#)。

JSON

```
{
  "Resources":{
    "myASG":{
      "CreationPolicy":{
        "ResourceSignal":{
          "Count":"3",
          "Timeout":"PT15M"
        }
      },
      "UpdatePolicy":{
        "AutoScalingRollingUpdate":{
          "MinInstancesInService":"3",
          "MaxBatchSize":"1",
          "PauseTime":"PT12M5S",
          "WaitOnResourceSignals":"true",
          "SuspendProcesses":[
```



```

        "HealthCheck",
        "ReplaceUnhealthy",
        "AZRebalance",
        "AlarmNotification",
        "ScheduledActions",
        "InstanceRefresh"
    ]
}
},
"Type": "AWS::AutoScaling::AutoScalingGroup",
"Properties": {
    "VPCZoneIdentifier": [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchTemplate": {
        "LaunchTemplateId": {
            "Ref": "logicalName"
        },
        "Version": {
            "Fn::GetAtt": [
                "logicalName",
                "LatestVersionNumber"
            ]
        }
    },
    "MaxSize": "5",
    "MinSize": "3"
}
}
}
}
}

```

YAML

```

---
Resources:
  myASG:
    CreationPolicy:
      ResourceSignal:
        Count: '3'
        Timeout: PT15M
    UpdatePolicy:
      AutoScalingRollingUpdate:
        MinInstancesInService: '3'
        MaxBatchSize: '1'

```

```

    PauseTime: PT12M5S
    WaitOnResourceSignals: true
    SuspendProcesses:
      - HealthCheck
      - ReplaceUnhealthy
      - AZRebalance
      - AlarmNotification
      - ScheduledActions
      - InstanceRefresh
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - subnetIdAz1
      - subnetIdAz2
      - subnetIdAz3
    LaunchTemplate:
      LaunchTemplateId: !Ref logicalName
      Version: !GetAtt logicalName.LatestVersionNumber
    MaxSize: '5'
    MinSize: '3'

```

建立步驟擴展政策

此範例顯示使用步進擴展政策擴增 Auto Scaling 群組的 [AWS::AutoScaling::ScalingPolicy](#) 資源。AdjustmentType 屬性指定 ChangeInCapacity，這表示 ScalingAdjustment 代表要新增 (如果 ScalingAdjustment 為正數) 或要刪除 (如果該屬性為負數) 的執行個體數量。在本範例中，ScalingAdjustment 為 1。因此，在超過警示閾值時，政策即會在群組中逐次遞增 1 個 EC2 執行個體。

[AWS::CloudWatch::Alarm](#) 資源 CPUAlarmHigh 可指定擴展政策 ASGScalingPolicyHigh 作為警示處於 ALARM 狀態時要執行的動作 (AlarmActions)。Dimensions 屬性參考在相同範本中其他位置宣告的 [AWS::AutoScaling::AutoScalingGroup](#) 資源邏輯名稱。

JSON

```

{
  "Resources": {
    "ASGScalingPolicyHigh": {
      "Type": "AWS::AutoScaling::ScalingPolicy",
      "Properties": {
        "AutoScalingGroupName": { "Ref": "logicalName" },

```

```

        "PolicyType": "StepScaling",
        "AdjustmentType": "ChangeInCapacity",
        "StepAdjustments": [
            {
                "MetricIntervalLowerBound": 0,
                "ScalingAdjustment": 1
            }
        ]
    },
    "CPUAlarmHigh": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "EvaluationPeriods": "2",
            "Statistic": "Average",
            "Threshold": "90",
            "AlarmDescription": "Scale out if CPU > 90% for 2 minutes",
            "Period": "60",
            "AlarmActions": [ { "Ref": "ASGScalingPolicyHigh" } ],
            "Namespace": "AWS/EC2",
            "Dimensions": [
                {
                    "Name": "AutoScalingGroupName",
                    "Value": { "Ref": "logicalName" }
                }
            ],
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "MetricName": "CPUUtilization"
        }
    }
}
}
}

```

YAML

```

---
Resources:
  ASGScalingPolicyHigh:
    Type: AWS::AutoScaling::ScalingPolicy
    Properties:
      AutoScalingGroupName: !Ref logicalName
      PolicyType: StepScaling
      AdjustmentType: ChangeInCapacity

```

```

    StepAdjustments:
      - MetricIntervalLowerBound: 0
        ScalingAdjustment: 1
  CPUAlarmHigh:
    Type: AWS::CloudWatch::Alarm
    Properties:
      EvaluationPeriods: 2
      Statistic: Average
      Threshold: 90
      AlarmDescription: 'Scale out if CPU > 90% for 2 minutes'
      Period: 60
      AlarmActions:
        - !Ref ASGScalingPolicyHigh
      Namespace: AWS/EC2
      Dimensions:
        - Name: AutoScalingGroupName
          Value:
            !Ref logicalName
      ComparisonOperator: GreaterThanThreshold
      MetricName: CPUUtilization

```

另請參閱

如需更多擴展政策的範例範本，請參閱 `AWS::AutoScaling::ScalingPolicy` 資源中的[範例](#)章節。

混合執行個體群組範例

使用屬性型執行個體類型選取範圍來建立 Auto Scaling 群組

此範例顯示 [AWS::AutoScaling::AutoScalingGroup](#) 資源，該資源包含使用基於屬性的執行個體類型選擇啟動混合執行個體群組之資訊。您需指定 `VCpuCount` 屬性的下限值和上限值，以及 `MemoryMiB` 屬性的下限值。Auto Scaling 群組使用的任何執行個體類型都必須與所需執行個體屬性相符。

Auto Scaling 群組的 `VPCZoneIdentifier` 屬性指定三個不同可用區域中的現有子網路清單。您必須先從帳戶指定適用的子網路 ID，然後才能建立堆疊。LaunchTemplate 屬性參考在相同範本中其他位置宣告的 [AWS::EC2::LaunchTemplate](#) 資源邏輯名稱。

JSON

```

{
  "Resources": {

```

```
"myASG":{
  "Type":"AWS::AutoScaling::AutoScalingGroup",
  "Properties":{
    "VPCZoneIdentifier":[
      "subnetIdAz1",
      "subnetIdAz2",
      "subnetIdAz3"
    ],
    "MixedInstancesPolicy":{
      "LaunchTemplate":{
        "LaunchTemplateSpecification":{
          "LaunchTemplateId":{
            "Ref":"logicalName"
          },
          "Version":{
            "Fn::GetAtt":[
              "logicalName",
              "LatestVersionNumber"
            ]
          }
        }
      },
      "Overrides":[
        {
          "InstanceRequirements":{
            "VCpuCount":{
              "Min":2,
              "Max":4
            },
            "MemoryMiB":{
              "Min":2048
            }
          }
        }
      ]
    },
    "MaxSize":"5",
    "MinSize":"1"
  }
}
```

YAML

```
---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      VPCZoneIdentifier:
        - subnetIdAz1
        - subnetIdAz2
        - subnetIdAz3
      MixedInstancesPolicy:
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref logicalName
            Version: !GetAtt logicalName.LatestVersionNumber
          Overrides:
            - InstanceRequirements:
                VCpuCount:
                  Min: 2
                  Max: 4
                MemoryMiB:
                  Min: 2048
      MaxSize: '5'
      MinSize: '1'
```

啟動組態範例

建立啟動組態

此範例顯示 Auto Scaling 群組的 [AWS::AutoScaling::LaunchConfiguration](#) 資源，您在此資源中為 ImageId、InstanceType 和 SecurityGroups 屬性指定值。SecurityGroups 屬性指定範本中其他處指定之 [AWS::EC2::SecurityGroup](#) 資源的邏輯名稱，以及名為 myExistingEC2SecurityGroup 的現有 EC2 安全群組。

JSON

```
"mySimpleConfig" : {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
  "Properties" : {
    "ImageId" : "ami-02354e95b3example",
```

```

    "InstanceType" : "t3.micro",
    "SecurityGroups" : [ { "Ref" : "logicalName" }, "myExistingEC2SecurityGroup" ]
  }
}

```

YAML

```

mySimpleConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: ami-02354e95b3example
    InstanceType: t3.micro
    SecurityGroups:
      - !Ref logicalName
      - myExistingEC2SecurityGroup

```

使用啟動組態建立 Auto Scaling 群組

此範例顯示具有單一執行個體的 [AWS::AutoScaling::AutoScalingGroup](#) 資源。Auto Scaling 群組的 `VPCZoneIdentifier` 屬性指定三個不同可用區域中的現有子網路清單。您必須先從帳戶指定適用的子網路 ID，然後才能建立堆疊。LaunchConfigurationName 屬性參考 [AWS::AutoScaling::LaunchConfiguration](#) 資源，該資源具有範本中所定義的邏輯名稱 `mySimpleConfig`。

JSON

```

"myASG" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "VPCZoneIdentifier" : [ "subnetIdAz1", "subnetIdAz2", "subnetIdAz3" ],
    "LaunchConfigurationName" : { "Ref" : "mySimpleConfig" },
    "MaxSize" : "1",
    "MinSize" : "1"
  }
}

```

YAML

```

myASG:
  Type: AWS::AutoScaling::AutoScalingGroup

```

```
Properties:
  VPCZoneIdentifier:
    - subnetIdAz1
    - subnetIdAz2
    - subnetIdAz3
  LaunchConfigurationName: !Ref mySimpleConfig
  MaxSize: '1'
  MinSize: '1'
```

使用 設定 Application Auto Scaling 資源 CloudFormation

本節提供 Application Auto Scaling 擴展政策和不同 AWS 資源排程動作的 CloudFormation 範本範例。

Important

範本中包含 Application Auto Scaling 程式碼片段時，您可能需要使用 DependsOn 屬性宣告對透過範本建立之特定可擴展資源的相依性。這會覆寫預設平行處理原則，並引導 CloudFormation 依指定順序在資源上操作。否則，擴展組態可能會在完全設定資源前套用。如需詳細資訊，請參閱 [DependsOn 屬性](#)。

程式碼片段類別

- [建立 AppStream 機群的擴展政策](#)
- [建立 Aurora 資料庫叢集的擴展原則](#)
- [建立 DynamoDB 資料表的擴展政策](#)
- [建立 Amazon ECS 服務的擴展政策 \(指標：平均 CPU 和記憶體\)](#)
- [建立 Amazon ECS 服務的擴展政策 \(指標：每個目標的平均請求計數\)](#)
- [使用 Cron 表達式建立 Lambda 函式的排程動作](#)
- [使用 at 表達式建立 Spot 機群的排程動作](#)

建立 AppStream 機群的擴展政策

此程式碼片段示範如何使用 [AWS::ApplicationAutoScaling::ScalingPolicy](#) 資源建立政策，並將其套用到 [AWS::AppStream::Fleet](#) 資源。[AWS::ApplicationAutoScaling::ScalableTarget](#) 資源宣告將套用此政策的可擴展目標。Application Auto Scaling 可擴展的機群執行個體數量下限為 1 個執行個體、上限為 20 個執行個體。該政策使機群的平均容量使用率保持在 75%，擴增和縮減的冷卻時間為 300 秒 (5 分鐘)。

其使用 `Fn::Join` 和 `Rev` 內建函數來建構 `ResourceId` 屬性，該屬性具有相同範本中指定的 `AWS::AppStream::Fleet` 資源的邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

JSON

```
{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 20,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/appstream.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_AppStreamFleet" },
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "fleet",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    },
    "ScalingPolicyAppStreamFleet" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu75" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "appstream",
        "ScalableDimension" : "appstream:fleet:DesiredCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "fleet",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    }
  }
}
```



```

ResourceId: !Join
  - /
  - - fleet
    - !Ref logicalName
TargetTrackingScalingPolicyConfiguration:
  TargetValue: 75
  PredefinedMetricSpecification:
    PredefinedMetricType: AppStreamAverageCapacityUtilization
  ScaleInCooldown: 300
  ScaleOutCooldown: 300

```

建立 Aurora 資料庫叢集的擴展原則

在此程式碼片段中，您將註冊 [AWS::RDS::DBCluster](#) 資源。[AWS::ApplicationAutoScaling::ScalableTarget](#) 資源表明應該動態縮放資料庫叢集，使其具有 1 到 8 個 Aurora 複本。您也可使用 [AWS::ApplicationAutoScaling::ScalingPolicy](#) 資源將目標追蹤擴展政策套用至叢集。

在此組態中，會使用 `RDSReaderAverageCPUUtilization` 這個預先定義的指標，根據 Aurora 資料庫叢集中所有 Aurora 複本平均 CPU 使用率百分之 40，來調整 Aurora 資料庫叢集。這個組態設定分別提供了 10 分鐘的規模縮減冷卻時間，和 5 分鐘的規模擴展冷卻時間。

此範例使用 `Fn::Sub` 內建函數來建構 `ResourceId` 屬性，該屬性具有相同範本中指定的 `AWS::RDS::DBCluster` 資源的邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

JSON

```

{
  "Resources" : {
    "ScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 8,
        "MinCapacity" : 1,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster" },
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" }
      }
    },
  },

```

```

    "ScalingPolicyDBCluster" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
      "Properties" : {
        "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu40" },
        "PolicyType" : "TargetTrackingScaling",
        "ServiceNamespace" : "rds",
        "ScalableDimension" : "rds:cluster:ReadReplicaCount",
        "ResourceId" : { "Fn::Sub" : "cluster:${logicalName}" },
        "TargetTrackingScalingPolicyConfiguration" : {
          "TargetValue" : 40,
          "PredefinedMetricSpecification" : {
            "PredefinedMetricType" : "RDSReaderAverageCPUUtilization"
          },
          "ScaleInCooldown" : 600,
          "ScaleOutCooldown" : 300
        }
      }
    }
  }
}

```

YAML

```

---
Resources:
  ScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 8
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/rds.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_RDSCluster'
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount
      ResourceId: !Sub cluster:${logicalName}
  ScalingPolicyDBCluster:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu40
      PolicyType: TargetTrackingScaling
      ServiceNamespace: rds
      ScalableDimension: rds:cluster:ReadReplicaCount

```

```

ResourceId: !Sub cluster:${logicalName}
TargetTrackingScalingPolicyConfiguration:
  TargetValue: 40
  PredefinedMetricSpecification:
    PredefinedMetricType: RDSReaderAverageCPUUtilization
  ScaleInCooldown: 600
  ScaleOutCooldown: 300

```

建立 DynamoDB 資料表的擴展政策

此程式碼片段示範如何使用 [AWS::ApplicationAutoScaling::ScalingPolicy](#) 資源建立 TargetTrackingScaling 政策類型的政策，並將其套用到 [AWS::DynamoDB::Table](#) 資源。[AWS::ApplicationAutoScaling::ScalableTarget](#) 資源會宣告要將此政策套用至哪個可擴展的目標，下限為五個寫入容量單位，上限為 15 個。此擴展政策會根據 DynamoDBWriteCapacityUtilization 預先定義指標擴展資料表的寫入容量輸送量，以將目標使用率維持在 50%。

其使用 Fn::Join 和 Ref 內建函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 AWS::DynamoDB::Table 資源的邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

Note

如需如何建立 DynamoDB 資源 CloudFormation 範本的詳細資訊，請參閱 資料庫部落格上的部落格文章[如何使用 CloudFormation 設定 Amazon DynamoDB 資料表和索引的自動擴展 AWS](#)。

JSON

```

{
  "Resources" : {
    "WriteCapacityScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 15,
        "MinCapacity" : 5,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam:${AWS::AccountId}:role/aws-service-role/dynamodb.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },
        "ServiceNamespace" : "dynamodb",
        "ScalableDimension" : "dynamodb:table:WriteCapacityUnits",

```

```

        "ResourceId" : {
            "Fn::Join" : [
                "/",
                [
                    "table",
                    {
                        "Ref" : "logicalName"
                    }
                ]
            ]
        }
    },
    "WriteScalingPolicy" : {
        "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties" : {
            "PolicyName" : "WriteScalingPolicy",
            "PolicyType" : "TargetTrackingScaling",
            "ScalingTargetId" : { "Ref" : "WriteCapacityScalableTarget" },
            "TargetTrackingScalingPolicyConfiguration" : {
                "TargetValue" : 50.0,
                "ScaleInCooldown" : 60,
                "ScaleOutCooldown" : 60,
                "PredefinedMetricSpecification" : {
                    "PredefinedMetricType" : "DynamoDBWriteCapacityUtilization"
                }
            }
        }
    }
}

```

YAML

```

---
Resources:
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      RoleARN:

```

```

    Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
    ServiceNamespace: dynamodb
    ScalableDimension: dynamodb:table:WriteCapacityUnits
    ResourceId: !Join
      - /
      - - table
        - !Ref LogicalName
WriteScalingPolicy:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: WriteScalingPolicy
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref WriteCapacityScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 50.0
      ScaleInCooldown: 60
      ScaleOutCooldown: 60
      PredefinedMetricSpecification:
        PredefinedMetricType: DynamoDBWriteCapacityUtilization

```

建立 Amazon ECS 服務的擴展政策 (指標：平均 CPU 和記憶體)

此程式碼片段示範如何使用 [AWS::ApplicationAutoScaling::ScalingPolicy](#) 資源建立政策，並將其套用到 [AWS::ECS::Service](#) 資源。

[AWS::ApplicationAutoScaling::ScalableTarget](#) 資源宣告將套用此政策的可擴展目標。Application Auto Scaling 可擴展的任務數量下限為 1 個任務、上限為 6 個任務。

這會建立兩個具有 TargetTrackingScaling 政策類型的擴展政策。這些政策用於根據服務的平均 CPU 和記憶體用量，以擴展 ECS 服務。其使用 Fn::Join 和 Ref 內建函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::ECS::Cluster](#) (myContainerCluster) 和 AWS::ECS::Service (myService) 資源之邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

JSON

```

{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : "6",
        "MinCapacity" : "1",

```

```

    "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService" },
    "ServiceNamespace" : "ecs",
    "ScalableDimension" : "ecs:service:DesiredCount",
    "ResourceId" : {
        "Fn::Join" : [
            "/",
            [
                "service",
                {
                    "Ref" : "myContainerCluster"
                },
                {
                    "Fn::GetAtt" : [
                        "myService",
                        "Name"
                    ]
                }
            ]
        ]
    },
    },
    "ServiceScalingPolicyCPU" : {
        "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties" : {
            "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-cpu70" },
            "PolicyType" : "TargetTrackingScaling",
            "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
            "TargetTrackingScalingPolicyConfiguration" : {
                "TargetValue" : 70.0,
                "ScaleInCooldown" : 180,
                "ScaleOutCooldown" : 60,
                "PredefinedMetricSpecification" : {
                    "PredefinedMetricType" : "ECSServiceAverageCPUUtilization"
                }
            }
        }
    },
    "ServiceScalingPolicyMem" : {
        "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
        "Properties" : {
            "PolicyName" : { "Fn::Sub" : "${AWS::StackName}-target-tracking-mem90" },

```



```

    "PolicyType" : "TargetTrackingScaling",
    "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
    "TargetTrackingScalingPolicyConfiguration" : {
      "TargetValue" : 90.0,
      "ScaleInCooldown" : 180,
      "ScaleOutCooldown" : 60,
      "PredefinedMetricSpecification" : {
        "PredefinedMetricType" : "ECSServiceAverageMemoryUtilization"
      }
    }
  }
}
}
}
}
}
}
}

```

YAML

```

---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 6
      MinCapacity: 1
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyCPU:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: !Sub ${AWS::StackName}-target-tracking-cpu70
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 70.0
        ScaleInCooldown: 180

```

```

    ScaleOutCooldown: 60
    PredefinedMetricSpecification:
      PredefinedMetricType: ECSServiceAverageCPUUtilization
ServiceScalingPolicyMem:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub ${AWS::StackName}-target-tracking-mem90
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ECSScalableTarget
    TargetTrackingScalingPolicyConfiguration:
      TargetValue: 90.0
      ScaleInCooldown: 180
      ScaleOutCooldown: 60
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageMemoryUtilization

```

建立 Amazon ECS 服務的擴展政策 (指標：每個目標的平均請求計數)

下列範例會將具有 `ALBRequestCountPerTarget` 預先定義指標的目標追蹤擴展政策套用至 ECS 服務。當每一目標 (每分鐘) 的請求計數超過目標值時，此政策會新增 ECS 服務的容量。因為 `DisableScaleIn` 的值設為 `true`，所以目標追蹤政策不會從可擴展的目標移除容量。

其使用 `Fn::Join` 和 `Fn::GetAtt` 內建函數來建構 `ResourceLabel` 屬性，該屬性具有相同範本中指定的 [AWS::ElasticLoadBalancingV2::LoadBalancer](#) (`myLoadBalancer`) 和 [AWS::ElasticLoadBalancingV2::TargetGroup](#) (`myTargetGroup`) 資源之邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

可擴展目標的 `MaxCapacity` 和 `MinCapacity` 屬性和擴展政策的 `TargetValue` 屬性會參考建立或更新堆疊時傳遞至範本的參數值。

JSON

```

{
  "Resources" : {
    "ECSScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : { "Ref" : "MaxCount" },
        "MinCapacity" : { "Ref" : "MinCount" },
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService" },
        "ServiceNamespace" : "ecs",

```

```

    "ScalableDimension" : "ecs:service:DesiredCount",
    "ResourceId" : {
      "Fn::Join" : [
        "/",
        [
          "service",
          {
            "Ref" : "myContainerCluster"
          },
          {
            "Fn::GetAtt" : [
              "myService",
              "Name"
            ]
          }
        ]
      ]
    }
  },
  "ServiceScalingPolicyALB" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties" : {
      "PolicyName" : "alb-requests-per-target-per-minute",
      "PolicyType" : "TargetTrackingScaling",
      "ScalingTargetId" : { "Ref" : "ECSScalableTarget" },
      "TargetTrackingScalingPolicyConfiguration" : {
        "TargetValue" : { "Ref" : "ALBPolicyTargetValue" },
        "ScaleInCooldown" : 180,
        "ScaleOutCooldown" : 30,
        "DisableScaleIn" : true,
        "PredefinedMetricSpecification" : {
          "PredefinedMetricType" : "ALBRequestCountPerTarget",
          "ResourceLabel" : {
            "Fn::Join" : [
              "/",
              [
                {
                  "Fn::GetAtt" : [
                    "myLoadBalancer",
                    "LoadBalancerFullName"
                  ]
                }
              ]
            ]
          },
          {

```

```
"Fn::GetAtt" : [
    "myTargetGroup",
    "TargetGroupName"
]
}
```

YAML

```

---
Resources:
  ECSScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: !Ref MaxCount
      MinCapacity: !Ref MinCount
      RoleARN:
        Fn::Sub: 'arn:aws:iam:::${AWS::AccountId}:role/aws-service-role/ecs.application-
autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService'
      ServiceNamespace: ecs
      ScalableDimension: 'ecs:service:DesiredCount'
      ResourceId: !Join
        - /
        - - service
          - !Ref myContainerCluster
          - !GetAtt myService.Name
  ServiceScalingPolicyALB:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: alb-requests-per-target-per-minute
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref ECSScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: !Ref ALBPolicyTargetValue
        ScaleInCooldown: 180

```

```

ScaleOutCooldown: 30
DisableScaleIn: true
PredefinedMetricSpecification:
  PredefinedMetricType: ALBRequestCountPerTarget
  ResourceLabel: !Join
    - '/'
    - - !GetAtt myLoadBalancer.LoadBalancerFullName
      - !GetAtt myTargetGroup.TargetGroupFullName

```

使用 Cron 表達式建立 Lambda 函式的排程動作

此程式碼片段會使用 [AWS::ApplicationAutoScaling::ScalableTarget](#) 資源，為名為 BLUE 的函數別名 ([AWS::Lambda::Alias](#)) 註冊佈建並行。還建立排程動作，並使用 Cron 運算式設定重複排程。重複排程的時區為 UTC。

它會使用 RoleARN 屬性中的 Fn::Join 與 Ref 內建函數來指定服務連結角色的 ARN。其使用 Fn::Sub 內建函數來建構 ResourceId 屬性，該屬性具有相同範本中指定的 [AWS::Lambda::Function](#) 或 [AWS::Serverless::Function](#) 資源的邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

Note

您無法在指向未發佈版本 (\$LATEST) 的別名上配置佈建並行。
如需如何建立 Lambda 資源 CloudFormation 範本的詳細資訊，請參閱 AWS 運算部落格上的[部落格文章排程 AWS Lambda 佈建並行的週期性尖峰用量](#)。

JSON

```

{
  "ScalableTarget" : {
    "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties" : {
      "MaxCapacity" : 250,
      "MinCapacity" : 0,
      "RoleARN" : {
        "Fn::Join" : [
          ":",
          [
            "arn:aws:iam:",
            {

```

```

        "Ref" : "AWS::AccountId"
      },
      "role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency"
    ]
  ]
},
"ServiceNamespace" : "lambda",
"ScalableDimension" : "lambda:function:ProvisionedConcurrency",
"ResourceId" : { "Fn::Sub" : "function:${logicalName}:BLUE" },
"ScheduledActions" : [
  {
    "ScalableTargetAction" : {
      "MinCapacity" : "250"
    },
    "ScheduledActionName" : "my-scale-out-scheduled-action",
    "Schedule" : "cron(0 18 * * ? *)",
    "EndTime" : "2022-12-31T12:00:00.000Z"
  }
]
}
}
}
}

```

YAML

```

ScalableTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: 250
    MinCapacity: 0
    RoleARN: !Join
      - ':'
      - - 'arn:aws:iam:'
        - !Ref 'AWS::AccountId'
        - role/aws-service-role/lambda.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_LambdaConcurrency
    ServiceNamespace: lambda
    ScalableDimension: lambda:function:ProvisionedConcurrency
    ResourceId: !Sub function:${logicalName}:BLUE
    ScheduledActions:
      - ScalableTargetAction:
          MinCapacity: 250

```

```
ScheduledActionName: my-scale-out-scheduled-action
Schedule: 'cron(0 18 * * ? *)'
EndTime: '2022-12-31T12:00:00.000Z'
```

使用 **at** 表達式建立 Spot 機群的排程動作

此程式碼片段示範如何使用 [AWS::ApplicationAutoScaling::ScalableTarget](#) 資源，為 [AWS::EC2::SpotFleet](#) 資源建立僅發生一次的兩個排程動作。每個一次性排程動作的時區都是 UTC。

其使用 `Fn::Join` 和 `Ref` 內建函數來建構 `ResourceId` 屬性，該屬性具有相同範本中指定的 `AWS::EC2::SpotFleet` 資源的邏輯名稱。如需詳細資訊，請參閱[內建函數參考](#)。

Note

Spot 機群請求必須具有 `maintain` 的請求類型。一次性請求或 Spot 區塊不支援自動擴展功能。

JSON

```
{
  "Resources" : {
    "SpotFleetScalableTarget" : {
      "Type" : "AWS::ApplicationAutoScaling::ScalableTarget",
      "Properties" : {
        "MaxCapacity" : 0,
        "MinCapacity" : 0,
        "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ec2.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest" },
        "ServiceNamespace" : "ec2",
        "ScalableDimension" : "ec2:spot-fleet-request:TargetCapacity",
        "ResourceId" : {
          "Fn::Join" : [
            "/",
            [
              "spot-fleet-request",
              {
                "Ref" : "logicalName"
              }
            ]
          ]
        }
      }
    }
  }
}
```

```

    ]
  },
  "ScheduledActions" : [
    {
      "ScalableTargetAction" : {
        "MaxCapacity" : "10",
        "MinCapacity" : "10"
      },
      "ScheduledActionName" : "my-scale-out-scheduled-action",
      "Schedule" : "at(2022-05-20T13:00:00)"
    },
    {
      "ScalableTargetAction" : {
        "MaxCapacity" : "0",
        "MinCapacity" : "0"
      },
      "ScheduledActionName" : "my-scale-in-scheduled-action",
      "Schedule" : "at(2022-05-20T21:00:00)"
    }
  ]
}
}
}
}
}

```

YAML

```

---
Resources:
  SpotFleetScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 0
      MinCapacity: 0
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-service-role/ec2.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest'
      ServiceNamespace: ec2
      ScalableDimension: 'ec2:spot-fleet-request:TargetCapacity'
      ResourceId: !Join
        - /
        - - spot-fleet-request
          - !Ref logicalName

```



```
ScheduledActions:
  - ScalableTargetAction:
      MaxCapacity: 10
      MinCapacity: 10
      ScheduledActionName: my-scale-out-scheduled-action
      Schedule: 'at(2022-05-20T13:00:00)'
  - ScalableTargetAction:
      MaxCapacity: 0
      MinCapacity: 0
      ScheduledActionName: my-scale-in-scheduled-action
      Schedule: 'at(2022-05-20T21:00:00)'
```

AWS 帳單主控台範本程式碼片段

此範例會建立一個包含 10% 全域標記定價規則的定價計劃。此定價計劃會連接至帳單群組。此帳單群組也有兩個自訂明細項目，在帳單群組總成本之上收取 10 美元的費用和 10% 的費用。

JSON

```
{
  "Parameters": {
    "LinkedAccountIds": {
      "Type": "ListNumber"
    },
    "PrimaryAccountId": {
      "Type": "Number"
    }
  },
  "Resources": {
    "TestPricingRule": {
      "Type": "AWS::BillingConductor::PricingRule",
      "Properties": {
        "Name": "TestPricingRule",
        "Description": "Test pricing rule created through Cloudformation. Mark everything by 10%.",
        "Type": "MARKUP",
        "Scope": "GLOBAL",
        "ModifierPercentage": 10
      }
    },
    "TestPricingPlan": {
      "Type": "AWS::BillingConductor::PricingPlan",
      "Properties": {
```

```

        "Name": "TestPricingPlan",
        "Description": "Test pricing plan created through Cloudformation.",
        "PricingRuleArns": [
            {"Fn::GetAtt": ["TestPricingRule", "Arn"]}
        ]
    },
    "TestBillingGroup": {
        "Type": "AWS::BillingConductor::BillingGroup",
        "Properties": {
            "Name": "TestBillingGroup",
            "Description": "Test billing group created through Cloudformation with 1
linked account. The linked account is also the primary account.",
            "PrimaryAccountId": {
                "Ref": "PrimaryAccountId"
            },
            "AccountGrouping": {
                "LinkedAccountIds": null
            },
            "ComputationPreference": {
                "PricingPlanArn": {
                    "Fn::GetAtt": ["TestPricingPlan", "Arn"]
                }
            }
        }
    },
    "TestFlatCustomLineItem": {
        "Type": "AWS::BillingConductor::CustomLineItem",
        "Properties": {
            "Name": "TestFlatCustomLineItem",
            "Description": "Test flat custom line item created through Cloudformation
for a $10 charge.",
            "BillingGroupArn": {
                "Fn::GetAtt": ["TestBillingGroup", "Arn"]
            },
            "CustomLineItemChargeDetails": {
                "Flat": {
                    "ChargeValue": 10
                },
                "Type": "FEE"
            }
        }
    },
    "TestPercentageCustomLineItem": {

```

```

        "Type": "AWS::BillingConductor::CustomLineItem",
        "Properties": {
            "Name": "TestPercentageCustomLineItem",
            "Description": "Test percentage custom line item created through
Cloudformation for a %10 additional charge on the overall total bill of the billing
group.",
            "BillingGroupArn": {
                "Fn::GetAtt": ["TestBillingGroup", "Arn"]
            },
            "CustomLineItemChargeDetails": {
                "Percentage": {
                    "PercentageValue": 10,
                    "ChildAssociatedResources": [
                        {"Fn::GetAtt": ["TestBillingGroup", "Arn"]}
                    ]
                },
                "Type": "FEE"
            }
        }
    }
}

```

YAML

```

Parameters:
  LinkedAccountIds:
    Type: ListNumber
  PrimaryAccountId:
    Type: Number
Resources:
  TestPricingRule:
    Type: AWS::BillingConductor::PricingRule
    Properties:
      Name: 'TestPricingRule'
      Description: 'Test pricing rule created through Cloudformation. Mark everything
by 10%.'
      Type: 'MARKUP'
      Scope: 'GLOBAL'
      ModifierPercentage: 10
  TestPricingPlan:
    Type: AWS::BillingConductor::PricingPlan
    Properties:

```

```

    Name: 'TestPricingPlan'
    Description: 'Test pricing plan created through Cloudformation.'
    PricingRuleArns:
      - !GetAtt TestPricingRule.Arn
  TestBillingGroup:
    Type: AWS::BillingConductor::BillingGroup
    Properties:
      Name: 'TestBillingGroup'
      Description: 'Test billing group created through Cloudformation with 1 linked
account. The linked account is also the primary account.'
      PrimaryAccountId: !Ref PrimaryAccountId
      AccountGrouping:
        LinkedAccountIds: !Ref LinkedAccountIds
      ComputationPreference:
        PricingPlanArn: !GetAtt TestPricingPlan.Arn
  TestFlatCustomLineItem:
    Type: AWS::BillingConductor::CustomLineItem
    Properties:
      Name: 'TestFlatCustomLineItem'
      Description: 'Test flat custom line item created through Cloudformation for a $10
charge.'
      BillingGroupArn: !GetAtt TestBillingGroup.Arn
      CustomLineItemChargeDetails:
        Flat:
          ChargeValue: 10
          Type: 'FEE'
  TestPercentageCustomLineItem:
    Type: AWS::BillingConductor::CustomLineItem
    Properties:
      Name: 'TestPercentageCustomLineItem'
      Description: 'Test percentage custom line item created through Cloudformation for
a %10 additional charge on the overall total bill of the billing group.'
      BillingGroupArn: !GetAtt TestBillingGroup.Arn
      CustomLineItemChargeDetails:
        Percentage:
          PercentageValue: 10
          ChildAssociatedResources:
            - !GetAtt TestBillingGroup.Arn
      Type: 'FEE'

```

CloudFormation 範本程式碼片段

主題

- [巢狀堆疊](#)
- [等待條件](#)

巢狀堆疊

在範本中建立巢狀堆疊

本範例範本中包含名為 myStack 的巢狀堆疊資源；當從範本 CloudFormation 建立堆疊時，它會建立 myStack，其範本是在 TemplateURL 屬性中指定。輸出值 StackRef 會傳回 myStack 的堆疊 ID，值 OutputFromNestedStack 則會從 myStack 資源中傳回輸出值 BucketName。保留 Outputs.*nestedstackoutputname* 格式以供您指定巢狀堆疊中的輸出值，且隨時隨地都能在涵蓋範本中使用該格式。

如需詳細資訊，請參閱[AWS::CloudFormation::Stack](#)。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template",
        "TimeoutInMinutes" : "60"
      }
    }
  },
  "Outputs": {
    "StackRef": {"Value": { "Ref" : "myStack"}},
    "OutputFromNestedStack" : {
      "Value" : { "Fn::GetAtt" : [ "myStack", "Outputs.BucketName" ] }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
```

```
myStack:
  Type: AWS::CloudFormation::Stack
  Properties:
    TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
S3_Bucket.template
    TimeoutInMinutes: '60'
Outputs:
  StackRef:
    Value: !Ref myStack
  OutputFromNestedStack:
    Value: !GetAtt myStack.Outputs.BucketName
```

在範本中使用輸入參數建立巢狀堆疊

本範例範本所包含的堆疊資源會指定輸入參數；當從此範本 CloudFormation 建立堆疊時，它會使用 Parameters 屬性中宣告的值對，做為用於建立myStackWithParams堆疊之範本的輸入參數。本範例指定的參數為 InstanceType 與 KeyName。

如需詳細資訊，請參閱[AWS::CloudFormation::Stack](#)。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myStackWithParams" : {
      "Type" : "AWS::CloudFormation::Stack",
      "Properties" : {
        "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-1/EC2ChooseAMI.template",
        "Parameters" : {
          "InstanceType" : "t2.micro",
          "KeyName" : "mykey"
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
```

```
myStackWithParams:
  Type: AWS::CloudFormation::Stack
  Properties:
    TemplateURL: https://s3.amazonaws.com/cloudformation-templates-us-east-1/
    EC2ChooseAMI.template
  Parameters:
    InstanceType: t2.micro
    KeyName: mykey
```

等待條件

使用 Amazon EC2 執行個體的等待條件

Important

針對 Amazon EC2 和 Auto Scaling 資源，我們建議您使用 `CreationPolicy` 屬性，而非等待條件。請將 `CreationPolicy` 屬性新增至這類資源，並使用 `cfn-signal` 的 helper 指令碼在執行個體建立流程成功完成後發出訊號。

如果您無法使用建立政策，請檢視以下宣告 Amazon EC2 執行個體等待條件的範例範本。myWaitCondition 等待條件會使用 myWaitConditionHandle 來發送訊號，並使用 DependsOn 屬性來指定 Amazon EC2 執行個體資源成功建立後，系統會觸發的等待條件；其亦會採用 Timeout 屬性將等待條件的持續期間指定為 4500 秒。此外，系統會透過 Ec2Instance 資源的 UserData 屬性，將發出等待條件訊號的預先簽章之 URL 傳遞至 Amazon EC2 執行個體，因此該 Amazon EC2 執行個體上執行的應用程式或指令碼皆可擷取預先簽章的 URL，並使用此 URL 發出等待條件的成功或失敗訊號。您需要使用 `cfn-signal`，或您所建立的應用程式或指令碼必須能發出等待條件訊號。而輸出值 ApplicationData 將包含等待條件訊號傳回的資料。

如需詳細資訊，請參閱[在 CloudFormation 範本中建立等待條件](#)。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
        "AMI" : "ami-0123456789abcdef0"
      },
      "us-west-1" : {
```

```

        "AMI" : "ami-0987654321fedcba0"
    },
    "eu-west-1" : {
        "AMI" : "ami-0abcdef123456789a"
    },
    "ap-northeast-1" : {
        "AMI" : "ami-0fedcba987654321b"
    },
    "ap-southeast-1" : {
        "AMI" : "ami-0c1d2e3f4a5b6c7d8"
    }
}
},
"Resources" : {
    "Ec2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
            "UserData" : { "Fn::Base64" : {"Ref" : "myWaitHandle"} },
            "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ] }
        }
    },
    "myWaitHandle" : {
        "Type" : "AWS::CloudFormation::WaitConditionHandle",
        "Properties" : {
        }
    },
    "myWaitCondition" : {
        "Type" : "AWS::CloudFormation::WaitCondition",
        "DependsOn" : "Ec2Instance",
        "Properties" : {
            "Handle" : { "Ref" : "myWaitHandle" },
            "Timeout" : "4500"
        }
    }
},
"Outputs" : {
    "ApplicationData" : {
        "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ] },
        "Description" : "The data passed back as part of signalling the
WaitCondition."
    }
}
}

```



```
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Mappings:
  RegionMap:
    us-east-1:
      AMI: ami-0123456789abcdef0
    us-west-1:
      AMI: ami-0987654321fedcba0
    eu-west-1:
      AMI: ami-0abcdef123456789a
    ap-northeast-1:
      AMI: ami-0fedcba987654321b
    ap-southeast-1:
      AMI: ami-0c1d2e3f4a5b6c7d8
Resources:
  Ec2Instance:
    Type: AWS::EC2::Instance
    Properties:
      UserData:
        Fn::Base64: !Ref myWaitHandle
      ImageId:
        Fn::FindInMap:
          - RegionMap
          - Ref: AWS::Region
          - AMI
  myWaitHandle:
    Type: AWS::CloudFormation::WaitConditionHandle
    Properties: {}
  myWaitCondition:
    Type: AWS::CloudFormation::WaitCondition
    DependsOn: Ec2Instance
    Properties:
      Handle: !Ref myWaitHandle
      Timeout: '4500'
Outputs:
  ApplicationData:
    Value: !GetAtt myWaitCondition.Data
    Description: The data passed back as part of signalling the WaitCondition.
```

使用 cfn-signal 協助程式指令碼發出等待條件訊號

本範例所顯示的 cfn-signal 命令列會發出等待條件的成功訊號。您需要在 EC2 執行個體的 UserData 屬性中定義命令列。

JSON

```
"UserData": {
  "Fn::Base64": {
    "Fn::Join": [
      "",
      [
        "#!/bin/bash -xe\n",
        "/opt/aws/bin/cfn-signal --exit-code 0 '",
        {
          "Ref": "myWaitHandle"
        },
        "'\n"
      ]
    ]
  }
}
```

YAML

```
UserData:
  Fn::Base64: !Sub |
    #!/bin/bash -xe
    /opt/aws/bin/cfn-signal --exit-code 0 '${myWaitHandle}'
```

使用 Curl 發出等待條件訊號

本範例所顯示的 Curl 命令列會發出等待條件的成功訊號。

```
curl -T /tmp/a "https://cloudformation-waitcondition-test.s3.amazonaws.com/
arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo
%3D"
```

其中 /tmp/a 檔案包含下列 JSON 結構：

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

本範例所顯示的 Curl 命令列將傳送相同的成功訊號，但會傳送 JSON 作為命令列中的參數。

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" :
"Configuration Complete","UniqueId" : "ID1234","Data" : "Application
has completed configuration."}' "https://cloudformation-waitcondition-
test.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack
%2Fstack-gosar-20110427004224-test-stack-with-WaitCondition--VEYW
%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVoo
%3D"
```

Amazon CloudFront 範本程式碼片段

在 CloudFormation 中將這些範例範本程式碼片段與 Amazon CloudFront 分佈資源搭配使用。如需詳細資訊，請參閱 [Amazon CloudFront 資源類型參考](#)。

主題

- [具有 Amazon S3 來源的 Amazon CloudFront 分佈資源](#)
- [Amazon CloudFront 分佈資源與自訂原始伺服器](#)
- [Amazon CloudFront 分佈與多重原始伺服器支援](#)
- [以 Lambda 函式做為原始伺服器的 Amazon CloudFront 分佈](#)
- [另請參閱](#)

具有 Amazon S3 來源的 Amazon CloudFront 分佈資源

以下範例範本顯示使用 [S3Origin](#) 和 [舊版原始存取身分 \(OAI\)](#) 的 [Amazon CloudFront 分佈](#)。如需關於使用原始伺服器存取控制 (OAC) 的詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的 [限制對 Amazon Simple Storage Service 原始伺服器的存取](#)。

JSON

```
{
```

```

    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myDistribution" : {
            "Type" : "AWS::CloudFront::Distribution",
            "Properties" : {
                "DistributionConfig" : {
                    "Origins" : [ {
                        "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
                        "Id" : "myS3Origin",
                        "S3OriginConfig" : {
                            "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
                        }
                    }
                ],
                "Enabled" : "true",
                "Comment" : "Some comment",
                "DefaultRootObject" : "index.html",
                "Logging" : {
                    "IncludeCookies" : "false",
                    "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
                    "Prefix" : "myprefix"
                },
                "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
                "DefaultCacheBehavior" : {
                    "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
                    "TargetOriginId" : "myS3Origin",
                    "ForwardedValues" : {
                        "QueryString" : "false",
                        "Cookies" : { "Forward" : "none" }
                    },
                    "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
                    "ViewerProtocolPolicy" : "allow-all"
                },
                "PriceClass" : "PriceClass_200",
                "Restrictions" : {
                    "GeoRestriction" : {
                        "RestrictionType" : "whitelist",
                        "Locations" : [ "AQ", "CV" ]
                    }
                },
                "ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }
            }
        }
    }

```

```
    }  
  }  
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'  
Resources:  
  myDistribution:  
    Type: AWS::CloudFront::Distribution  
    Properties:  
      DistributionConfig:  
        Origins:  
          - DomainName: amzn-s3-demo-bucket.s3.amazonaws.com  
            Id: myS3Origin  
            S3OriginConfig:  
              OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z  
        Enabled: 'true'  
        Comment: Some comment  
        DefaultRootObject: index.html  
        Logging:  
          IncludeCookies: 'false'  
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com  
          Prefix: myprefix  
        Aliases:  
          - mysite.example.com  
          - yoursite.example.com  
        DefaultCacheBehavior:  
          AllowedMethods:  
            - DELETE  
            - GET  
            - HEAD  
            - OPTIONS  
            - PATCH  
            - POST  
            - PUT  
          TargetOriginId: myS3Origin  
          ForwardedValues:  
            QueryString: 'false'  
            Cookies:  
              Forward: none  
          TrustedSigners:  
            - 1234567890EX
```

```
- 1234567891EX
ViewerProtocolPolicy: allow-all
PriceClass: PriceClass_200
Restrictions:
  GeoRestriction:
    RestrictionType: whitelist
    Locations:
      - AQ
      - CV
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

Amazon CloudFront 分佈資源與自訂原始伺服器

以下範例範本顯示使用 [CustomOrigin](#) 的 Amazon CloudFront [分佈](#)。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "DomainName" : "www.example.com",
            "Id" : "myCustomOrigin",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          } ],
          "Enabled" : "true",
          "Comment" : "Somecomment",
          "DefaultRootObject" : "index.html",
          "Logging" : {
            "IncludeCookies" : "true",
            "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
            "Prefix" : "myprefix"
          },
          "Aliases" : [
```



```
DistributionConfig:
  Origins:
  - DomainName: www.example.com
    Id: myCustomOrigin
    CustomOriginConfig:
      HTTPPort: '80'
      HTTPSPort: '443'
      OriginProtocolPolicy: http-only
  Enabled: 'true'
  Comment: Somecomment
  DefaultRootObject: index.html
  Logging:
    IncludeCookies: 'true'
    Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
    Prefix: myprefix
  Aliases:
  - mysite.example.com
  - "*.yoursite.example.com"
  DefaultCacheBehavior:
    TargetOriginId: myCustomOrigin
    SmoothStreaming: 'false'
    ForwardedValues:
      QueryString: 'false'
      Cookies:
        Forward: all
    TrustedSigners:
    - 1234567890EX
    - 1234567891EX
    ViewerProtocolPolicy: allow-all
  CustomErrorResponses:
  - ErrorCode: '404'
    ResponsePagePath: "/error-pages/404.html"
    ResponseCode: '200'
    ErrorCachingMinTTL: '30'
  PriceClass: PriceClass_200
  Restrictions:
    GeoRestriction:
      RestrictionType: whitelist
      Locations:
      - AQ
      - CV
  ViewerCertificate:
    CloudFrontDefaultCertificate: 'true'
```


Amazon CloudFront 分佈與多重原始伺服器支援

以下範例範本說明如何宣告具有多重原始伺服器支援的 CloudFront [分佈](#)。在 [DistributionConfig](#) 中，提供原始伺服器清單，並設定 [DefaultCacheBehavior](#)。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDistribution" : {
      "Type" : "AWS::CloudFront::Distribution",
      "Properties" : {
        "DistributionConfig" : {
          "Origins" : [ {
            "Id" : "myS3Origin",
            "DomainName" : "amzn-s3-demo-bucket.s3.amazonaws.com",
            "S3OriginConfig" : {
              "OriginAccessIdentity" : "origin-access-identity/
cloudfront/E127EXAMPLE51Z"
            }
          },
          {
            "Id" : "myCustomOrigin",
            "DomainName" : "www.example.com",
            "CustomOriginConfig" : {
              "HTTPPort" : "80",
              "HTTPSPort" : "443",
              "OriginProtocolPolicy" : "http-only"
            }
          }
        ],
        "Enabled" : "true",
        "Comment" : "Some comment",
        "DefaultRootObject" : "index.html",
        "Logging" : {
          "IncludeCookies" : "true",
          "Bucket" : "amzn-s3-demo-logging-bucket.s3.amazonaws.com",
          "Prefix" : "myprefix"
        },
        "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
        "DefaultCacheBehavior" : {
          "TargetOriginId" : "myS3Origin",
```

```

        "ForwardedValues" : {
            "QueryString" : "false",
            "Cookies" : { "Forward" : "all" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "100",
        "SmoothStreaming" : "true"
    },
    "CacheBehaviors" : [ {
        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myS3Origin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",
        "PathPattern" : "images1/*.jpg"
    },
    {
        "AllowedMethods" : [ "DELETE", "GET", "HEAD", "OPTIONS",
"PATCH", "POST", "PUT" ],
        "TargetOriginId" : "myCustomOrigin",
        "ForwardedValues" : {
            "QueryString" : "true",
            "Cookies" : { "Forward" : "none" }
        },
        "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
        "ViewerProtocolPolicy" : "allow-all",
        "MinTTL" : "50",
        "PathPattern" : "images2/*.jpg"
    }
    ],
    "CustomErrorResponses" : [ {
        "ErrorCode" : "404",
        "ResponsePagePath" : "/error-pages/404.html",
        "ResponseCode" : "200",
        "ErrorCachingMinTTL" : "30"
    } ],
    "PriceClass" : "PriceClass_All",
    "ViewerCertificate" : { "CloudFrontDefaultCertificate" : "true" }

```

```

    }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Origins:
          - Id: myS3Origin
            DomainName: amzn-s3-demo-bucket.s3.amazonaws.com
            S3OriginConfig:
              OriginAccessIdentity: origin-access-identity/cloudfront/E127EXAMPLE51Z
          - Id: myCustomOrigin
            DomainName: www.example.com
            CustomOriginConfig:
              HTTPPort: '80'
              HTTPSPort: '443'
              OriginProtocolPolicy: http-only
        Enabled: 'true'
        Comment: Some comment
        DefaultRootObject: index.html
        Logging:
          IncludeCookies: 'true'
          Bucket: amzn-s3-demo-logging-bucket.s3.amazonaws.com
          Prefix: myprefix
      Aliases:
        - mysite.example.com
        - yoursite.example.com
      DefaultCacheBehavior:
        TargetOriginId: myS3Origin
        ForwardedValues:
          QueryString: 'false'
          Cookies:
            Forward: all
        TrustedSigners:
          - 1234567890EX

```

```
- 1234567891EX
ViewerProtocolPolicy: allow-all
MinTTL: '100'
SmoothStreaming: 'true'
CacheBehaviors:
- AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myS3Origin
  ForwardedValues:
    QueryString: 'true'
    Cookies:
      Forward: none
  TrustedSigners:
    - 1234567890EX
    - 1234567891EX
  ViewerProtocolPolicy: allow-all
  MinTTL: '50'
  PathPattern: images1/*.jpg
- AllowedMethods:
  - DELETE
  - GET
  - HEAD
  - OPTIONS
  - PATCH
  - POST
  - PUT
  TargetOriginId: myCustomOrigin
  ForwardedValues:
    QueryString: 'true'
    Cookies:
      Forward: none
  TrustedSigners:
    - 1234567890EX
    - 1234567891EX
  ViewerProtocolPolicy: allow-all
  MinTTL: '50'
  PathPattern: images2/*.jpg
CustomErrorResponses:
```

```
- ErrorCode: '404'
  ResponsePagePath: "/error-pages/404.html"
  ResponseCode: '200'
  ErrorCachingMinTTL: '30'
PriceClass: PriceClass_All
ViewerCertificate:
  CloudFrontDefaultCertificate: 'true'
```

以 Lambda 函式做為原始伺服器的 Amazon CloudFront 分佈

下列範例建立 CloudFront 分佈，該分佈會面向指定的 Lambda 函式 URL (做為參數提供)，啟用僅限 HTTPS 的存取、快取、壓縮和全域交付。它將 Lambda URL 設定為自訂 HTTPS AWS 原始伺服器，並套用標準快取政策。此分佈已針對效能進行了最佳化，支援 HTTP/2 和 IPv6，並輸出 CloudFront 網域名稱，讓使用者可透過安全的 CDN 支援端點存取 Lambda 函式。如需詳細資訊，請參閱 AWS 部落格上的[搭配使用 Amazon CloudFront AWS Lambda 做為原始伺服器，以加速您的 Web 應用程式](#)。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "LambdaEndpoint": {
      "Type": "String",
      "Description": "The Lambda function URL endpoint without the 'https://'"
    }
  },
  "Resources": {
    "MyDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
        "DistributionConfig": {
          "PriceClass": "PriceClass_All",
          "HttpVersion": "http2",
          "IPV6Enabled": true,
          "Origins": [
            {
              "DomainName": {
                "Ref": "LambdaEndpoint"
              },
              "Id": "LambdaOrigin",
              "CustomOriginConfig": {
                "HTTPSPort": 443,
                "OriginProtocolPolicy": "https-only"
              }
            }
          ]
        }
      }
    }
  }
}
```

```

        }
      }
    ],
    "Enabled": "true",
    "DefaultCacheBehavior": {
      "TargetOriginId": "LambdaOrigin",
      "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6",
      "ViewerProtocolPolicy": "redirect-to-https",
      "SmoothStreaming": "false",
      "Compress": "true"
    }
  }
}
},
"Outputs": {
  "CloudFrontDomain": {
    "Description": "CloudFront default domain name configured",
    "Value": {
      "Fn::Sub": "https://${MyDistribution.DomainName}/"
    }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Parameters:
  LambdaEndpoint:
    Type: String
    Description: The Lambda function URL endpoint without the 'https://'
Resources:
  MyDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        PriceClass: PriceClass_All
        HttpVersion: http2
        IPV6Enabled: true
        Origins:
          - DomainName: !Ref LambdaEndpoint
            Id: LambdaOrigin

```

```
CustomOriginConfig:
  HTTPSPort: 443
  OriginProtocolPolicy: https-only
Enabled: 'true'
DefaultCacheBehavior:
  TargetOriginId: LambdaOrigin
  CachePolicyId: '658327ea-f89d-4fab-a63d-7e88639e58f6'
  ViewerProtocolPolicy: redirect-to-https
  SmoothStreaming: 'false'
  Compress: 'true'

Outputs:
  CloudFrontDomain:
    Description: CloudFront default domain name configured
    Value: !Sub https://${MyDistribution.DomainName}/
```

另請參閱

若需有關在 Route 53 記錄中新增自訂別名，為 CloudFront 分佈建立易記名稱的範例，請參閱 [CloudFront 分佈的別名資源紀錄集](#)。

Amazon CloudWatch 範本程式碼片段

使用這些範例範本程式碼片段，以協助您在 CloudFormation 中描述 Amazon CloudWatch 資源。如需詳細資訊，請參閱 [Amazon CloudWatch 資源類型參考](#)。

主題

- [帳單警示](#)
- [CPU 使用率警示](#)
- [復原 Amazon Elastic Compute Cloud 執行個體](#)
- [建立基本儀表板](#)
- [建立並排 Widget 的儀表板](#)

帳單警示

在下列範例中，AWS 帳戶的費用超過警示閾值時，Amazon CloudWatch 會傳送電子郵件通知。若要取得使用量通知，請啟用帳單提醒。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的 [建立帳單警示以監控您的預估 AWS 費用](#)。>

JSON

```

"SpendingAlarm": {
  "Type": "AWS::CloudWatch::Alarm",
  "Properties": {
    "AlarmDescription": { "Fn::Join": [ "", [
      "Alarm if AWS spending is over $",
      { "Ref": "AlarmThreshold" }
    ] ] },
    "Namespace": "AWS/Billing",
    "MetricName": "EstimatedCharges",
    "Dimensions": [{
      "Name": "Currency",
      "Value": "USD"
    }],
    "Statistic": "Maximum",
    "Period": "21600",
    "EvaluationPeriods": "1",
    "Threshold": { "Ref": "AlarmThreshold" },
    "ComparisonOperator": "GreaterThanOrEqualToThreshold",
    "AlarmActions": [{
      "Ref": "BillingAlarmNotification"
    }],
    "InsufficientDataActions": [{
      "Ref": "BillingAlarmNotification"
    }]
  }
}

```

YAML

```

SpendingAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription:
      'Fn::Join':
        - ''
        - - Alarm if AWS spending is over $
          - !Ref: AlarmThreshold
    Namespace: AWS/Billing
    MetricName: EstimatedCharges
    Dimensions:
      - Name: Currency

```



```

    Value: USD
    Statistic: Maximum
    Period: '21600'
    EvaluationPeriods: '1'
    Threshold:
      !Ref: "AlarmThreshold"
    ComparisonOperator: GreaterThanThreshold
    AlarmActions:
      - !Ref: "BillingAlarmNotification"
    InsufficientDataActions:
      - !Ref: "BillingAlarmNotification"

```

CPU 使用率警示

下列範本程式碼片段會建立警示，在 Amazon EC2 執行個體的平均 CPU 使用率於三段評估期間長達 60 秒以上超過 90% 時，傳送通知。

JSON

```

"CPULAlarm" : {
  "Type" : "AWS::CloudWatch::Alarm",
  "Properties" : {
    "AlarmDescription" : "CPU alarm for my instance",
    "AlarmActions" : [ { "Ref" : "logical name of an AWS::SNS::Topic resource" } ],
    "MetricName" : "CPUUtilization",
    "Namespace" : "AWS/EC2",
    "Statistic" : "Average",
    "Period" : "60",
    "EvaluationPeriods" : "3",
    "Threshold" : "90",
    "ComparisonOperator" : "GreaterThanThreshold",
    "Dimensions" : [ {
      "Name" : "InstanceId",
      "Value" : { "Ref" : "logical name of an AWS::EC2::Instance resource" }
    } ]
  }
}

```

YAML

```

CPULAlarm:
  Type: AWS::CloudWatch::Alarm
  Properties:

```

```
AlarmDescription: CPU alarm for my instance
AlarmActions:
- !Ref: "logical name of an AWS::SNS::Topic resource"
MetricName: CPUUtilization
Namespace: AWS/EC2
Statistic: Average
Period: '60'
EvaluationPeriods: '3'
Threshold: '90'
ComparisonOperator: GreaterThanThreshold
Dimensions:
- Name: InstanceId
  Value: !Ref: "logical name of an AWS::EC2::Instance resource"
```

復原 Amazon Elastic Compute Cloud 執行個體

當 EC2 執行個體的状态檢查連續 15 分鐘發生故障時，以下 CloudWatch 警示會復原執行個體。如需有關警示動作的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的[建立警示來停止、終止、重新啟動或復原 EC2 執行個體](#)。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "RecoveryInstance" : {
      "Description" : "The EC2 instance ID to associate this alarm with.",
      "Type" : "AWS::EC2::Instance::Id"
    }
  },
  "Resources": {
    "RecoveryTestAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Trigger a recovery when instance status check fails for 15 consecutive minutes.",
        "Namespace": "AWS/EC2" ,
        "MetricName": "StatusCheckFailed_System",
        "Statistic": "Minimum",
        "Period": "60",
        "EvaluationPeriods": "15",
        "ComparisonOperator": "GreaterThanThreshold",
        "Threshold": "0",
```

```

        "AlarmActions": [ {"Fn::Join" : [ "", [ "arn:aws:automate:", { "Ref" :
"AWS::Region" }, ":ec2:recover" ] ] } ],
        "Dimensions": [ { "Name": "InstanceId", "Value": { "Ref": "RecoveryInstance" } } ]
    }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  RecoveryInstance:
    Description: The EC2 instance ID to associate this alarm with.
    Type: AWS::EC2::Instance::Id
Resources:
  RecoveryTestAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: Trigger a recovery when instance status check fails for 15
        consecutive minutes.
      Namespace: AWS/EC2
      MetricName: StatusCheckFailed_System
      Statistic: Minimum
      Period: '60'
      EvaluationPeriods: '15'
      ComparisonOperator: GreaterThanThreshold
      Threshold: '0'
      AlarmActions: [ !Sub "arn:aws:automate:${AWS::Region}:ec2:recover" ]
      Dimensions:
        - Name: InstanceId
          Value: !Ref: RecoveryInstance

```

建立基本儀表板

以下範例會建立一個簡單的 CloudWatch 儀表板，有一個顯示 CPU 使用率的指標 Widget 和一個顯示訊息的文字 Widget。

JSON

```

{
  "BasicDashboard": {
    "Type": "AWS::CloudWatch::Dashboard",

```

```

    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\\"widgets\\":[{\\"type\\":\\"metric\\",\\"x\\":0,\\"y\\":0,
\\"width\\":12,\\"height\\":6,\\"properties\\":{\\"metrics\\":[[\\"AWS/EC2\\",\\"CPUUtilization
\\",\\"InstanceId\\",\\"i-012345\\"]],\\"period\\":300,\\"stat\\":\\"Average\\",\\"region\\":\\"us-
east-1\\",\\"title\\":\\"EC2 Instance CPU\\"}],{\\"type\\":\\"text\\",\\"x\\":0,\\"y\\":7,\\"width
\\":3,\\"height\\":3,\\"properties\\":{\\"markdown\\":\\"Hello world\\"}}]}]"
    }
  }
}

```

YAML

```

BasicDashboard:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]],"period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"text","x":0,"y":7,"width":3,"height":3,"properties":{"markdown":"Hello
world"}}]}'

```

建立並排 Widget 的儀表板

以下範例會建立有兩個指標 Widget 並排的儀表板。

JSON

```

{
  "DashboardSideBySide": {
    "Type": "AWS::CloudWatch::Dashboard",
    "Properties": {
      "DashboardName": "Dashboard1",
      "DashboardBody": "{\\"widgets\\":[{\\"type\\":\\"metric\\",\\"x\\":0,\\"y\\":0,
\\"width\\":12,\\"height\\":6,\\"properties\\":{\\"metrics\\":[[\\"AWS/EC2\\",\\"CPUUtilization
\\",\\"InstanceId\\",\\"i-012345\\"]],\\"period\\":300,\\"stat\\":\\"Average\\",\\"region\\":
\\"us-east-1\\",\\"title\\":\\"EC2 Instance CPU\\"}],{\\"type\\":\\"metric\\",\\"x\\":12,\\"y\\":0,
\\"width\\":12,\\"height\\":6,\\"properties\\":{\\"metrics\\":[[\\"AWS/S3\\",\\"BucketSizeBytes\\",
\\"BucketName\\",\\"amzn-s3-demo-bucket\\"]],\\"period\\":86400,\\"stat\\":\\"Maximum\\",\\"region
\\":\\"us-east-1\\",\\"title\\":\\"amzn-s3-demo-bucket bytes\\"}}]}]"
    }
  }
}

```

```

    }
  }
}
```

YAML

```

DashboardSideBySide:
  Type: AWS::CloudWatch::Dashboard
  Properties:
    DashboardName: Dashboard1
    DashboardBody: '{"widgets":
[{"type":"metric","x":0,"y":0,"width":12,"height":6,"properties":{"metrics":[["AWS/
EC2","CPUUtilization","InstanceId","i-012345"]], "period":300,"stat":"Average","region":"us-
east-1","title":"EC2 Instance CPU"}},
{"type":"metric","x":12,"y":0,"width":12,"height":6,"properties":
{"metrics":[["AWS/S3","BucketSizeBytes","BucketName","amzn-s3-demo-
bucket"]], "period":86400,"stat":"Maximum","region":"us-east-1","title":"amzn-s3-demo-
bucket bytes"}]}]}'
```

Amazon CloudWatch Logs 範本程式碼片段

Amazon CloudWatch Logs 可監控您的系統、應用程式，以及來自 Amazon EC2 執行個體或其他來源的自訂日誌檔案。您可以使用 CloudFormation 來佈建和管理日誌群組和指標篩選條件。如需 Amazon CloudWatch Logs 的詳細資訊，請參閱 [《Amazon CloudWatch Logs 使用者指南》](#)。

主題

- [從 Linux 執行個體傳送日誌到 CloudWatch Logs](#)
- [從 Windows 執行個體傳送日誌到 CloudWatch Logs](#)
- [另請參閱](#)

從 Linux 執行個體傳送日誌到 CloudWatch Logs

以下範本示範如何透過 CloudWatch Logs 整合，在 Amazon Linux 2023 上設定 Web 伺服器。此範本會執行以下任務：

- 安裝 Apache 與 PHP。
- 設定 CloudWatch 代理程式，將 Apache 存取日誌轉送至 CloudWatch Logs。

- 設定 IAM 角色，允許 CloudWatch 代理程式將日誌資料傳送至 CloudWatch Logs。
- 建立自訂警示與通知，用於監控 404 錯誤或高頻寬用量。

Web 伺服器的日誌事件會為 CloudWatch 警示提供指標資料。這兩個指標篩選條件描述日誌資訊如何轉換成 CloudWatch 指標。404 指標會計算 404 事件的數量。大小指標會追蹤請求的大小。如果在 2 分鐘內發生兩次以上的 404，或 10 分鐘內的平均請求大小超過 3500 KB，這兩個 CloudWatch 警示會傳送通知。

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Sample template that sets up and configures CloudWatch Logs on Amazon Linux 2023 instance.",
    "Parameters": {
        "KeyName": {
            "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instances",
            "Type": "AWS::EC2::KeyPair::KeyName",
            "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
        },
        "SSHLocation": {
            "Description": "The IP address range that can be used to SSH to the EC2 instances",
            "Type": "String",
            "MinLength": "9",
            "MaxLength": "18",
            "Default": "0.0.0.0/0",
            "AllowedPattern": "(\\d{1,3})\\.((\\d{1,3})|\\.(\\d{1,3})|\\.(\\d{1,3}))/(\\d{1,2})",
            "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
        },
        "OperatorEmail": {
            "Description": "Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)",
            "Type": "String"
        }
    },
    "Resources": {
        "LogRole": {
            "Type": "AWS::IAM::Role",
```

```

    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "ec2.amazonaws.com"
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ]
          }
        ]
      },
      "Path": "/",
      "Policies": [
        {
          "PolicyName": "LogRolePolicy",
          "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
              {
                "Effect": "Allow",
                "Action": [
                  "logs:PutLogEvents",
                  "logs:DescribeLogStreams",
                  "logs:DescribeLogGroups",
                  "logs:CreateLogGroup",
                  "logs:CreateLogStream"
                ],
                "Resource": "*"
              }
            ]
          }
        ]
      ]
    },
    "LogRoleInstanceProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Properties": {

```

```

        "Path": "/",
        "Roles": [{"Ref": "LogRole"}]
    },
    "WebServerSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable HTTP access via port 80 and SSH access via
port 22",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "CidrIp": {"Ref": "SSHLocation"}
                }
            ]
        }
    },
    "WebServerHost": {
        "Type": "AWS::EC2::Instance",
        "Metadata": {
            "Comment": "Install a simple PHP application on Amazon Linux 2023",
            "AWS::CloudFormation::Init": {
                "config": {
                    "packages": {
                        "dnf": {
                            "httpd": [],
                            "php": [],
                            "php-fpm": []
                        }
                    },
                    "files": {
                        "/etc/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.json": {
                            "content": {
                                "logs": {
                                    "logs_collected": {

```



```

        "files": {
            "collect_list": [{
                "file_path": "/var/log/httpd/
access_log",
                "log_group_name": {"Ref":
                    "WebServerLogGroup"},
                "log_stream_name": "{instance_id}/
apache.log",
                "timestamp_format": "%d/%b/%Y:%H:
%M:%S %z"
            }]
        }
    },
    {
        "mode": "000644",
        "owner": "root",
        "group": "root"
    },
    {
        "content": "<?php\\necho '<h1>AWS CloudFormation sample
PHP application on Amazon Linux 2023</h1>';\\n?>\\n",
        "mode": "000644",
        "owner": "apache",
        "group": "apache"
    },
    {
        "content": {
            "Fn::Join": [
                "",
                [
                    "[main]\\n",
                    "stack=",
                    {"Ref": "AWS::StackId"},
                    "\\n",
                    "region=",
                    {"Ref": "AWS::Region"},
                    "\\n"
                ]
            ]
        },
        "mode": "000400",
        "owner": "root",
        "group": "root"
    }
}

```

```

    },
    "/etc/cfn/hooks.d/cfn-auto-reloader.conf": {
      "content": {
        "Fn::Join": [
          "",
          [
            "[cfn-auto-reloader-hook]\n",
            "triggers=post.update\n",
            "\n",
            "path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init\n",
            "action=/opt/aws/bin/cfn-init -s ",
            {"Ref": "AWS::StackId"},
            " -r WebServerHost ",
            " --region ",
            {"Ref": "AWS::Region"},
            "\n",
            "runas=root\n"
          ]
        ]
      }
    },
    "services": {
      "systemd": {
        "httpd": {
          "enabled": "true",
          "ensureRunning": "true"
        },
        "php-fpm": {
          "enabled": "true",
          "ensureRunning": "true"
        }
      }
    },
    "CreationPolicy": {
      "ResourceSignal": {
        "Timeout": "PT5M"
      }
    },
    "Properties": {

```

```

      "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64}}",
      "KeyName": {"Ref": "KeyName"},
      "InstanceType": "t3.micro",
      "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
      "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
      "UserData": {"Fn::Base64": {"Fn::Join": [ "", [
        "#!/bin/bash\n",
        "dnf update -y aws-cfn-bootstrap\n",
        "dnf install -y amazon-cloudwatch-agent\n",
        "/opt/aws/bin/cfn-init -v --stack ", {"Ref": "AWS::StackName"}, "
--resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n",
        "\n",
        "# Verify Apache log directory exists and create if needed\n",
        "mkdir -p /var/log/httpd\n",
        "\n",
        "# Start CloudWatch agent\n",
        "/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -
a fetch-config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json
-s\n",
        "\n",
        "# Signal success\n",
        "/opt/aws/bin/cfn-signal -e $? --stack ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "\n"
]]}}
    }
  },
  "WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "Properties": {
      "RetentionInDays": 7
    }
  },
  "404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
      "LogGroupName": {"Ref": "WebServerLogGroup"},
      "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code = 404, size, ...]",
      "MetricTransformations": [
        {
          "MetricValue": "1",

```

```

        "MetricNamespace": "test/404s",
        "MetricName": "test404Count"
    }
    ]
},
"BytesTransferredMetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {"Ref": "WebServerLogGroup"},
        "FilterPattern": "[ip, identity, user_id, timestamp, request,
status_code, size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "$size",
                "MetricNamespace": "test/BytesTransferred",
                "MetricName": "testBytesTransferred"
            }
        ]
    }
},
"404Alarm": {
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
        "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
        "MetricName": "test404Count",
        "Namespace": "test/404s",
        "Statistic": "Sum",
        "Period": "60",
        "EvaluationPeriods": "2",
        "Threshold": "2",
        "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
        "ComparisonOperator": "GreaterThanThreshold"
    }
},
"BandwidthAlarm": {
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
        "AlarmDescription": "The average volume of traffic is greater 3500 KB
over 10 minutes",
        "MetricName": "testBytesTransferred",
        "Namespace": "test/BytesTransferred",
        "Statistic": "Average",

```

```

        "Period": "300",
        "EvaluationPeriods": "2",
        "Threshold": "3500",
        "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
        "ComparisonOperator": "GreaterThanThreshold"
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    },
    "Outputs": {
        "InstanceId": {
            "Description": "The instance ID of the web server",
            "Value": {"Ref": "WebServerHost"}
        },
        "WebsiteURL": {
            "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
            "Description": "URL for the web server"
        },
        "PublicIP": {
            "Description": "Public IP address of the web server",
            "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
        },
        "CloudWatchLogGroupName": {
            "Description": "The name of the CloudWatch log group",
            "Value": {"Ref": "WebServerLogGroup"}
        }
    }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Sample template that sets up and configures CloudWatch Logs on Amazon
  Linux 2023 instance.
Parameters:
  KeyName:

```

Description: Name of an existing EC2 KeyPair to enable SSH access to the instances

Type: AWS::EC2::KeyPair::KeyName

ConstraintDescription: must be the name of an existing EC2 KeyPair.

SSHLocation:

Description: The IP address range that can be used to SSH to the EC2 instances

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

OperatorEmail:

Description: Email address to notify when CloudWatch alarms are triggered (404 errors or high bandwidth usage)

Type: String

Resources:

LogRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Principal:

Service:

- ec2.amazonaws.com

Action:

- 'sts:AssumeRole'

Path: /

Policies:

- PolicyName: LogRolePolicy

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

- 'logs:PutLogEvents'

- 'logs:DescribeLogStreams'

- 'logs:DescribeLogGroups'

- 'logs:CreateLogGroup'

- 'logs:CreateLogStream'

Resource: '*'

LogRoleInstanceProfile:

Type: AWS::IAM::InstanceProfile

```

Properties:
  Path: /
  Roles:
    - !Ref LogRole
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: !Ref SSHLocation
WebServerHost:
  Type: AWS::EC2::Instance
  Metadata:
    Comment: Install a simple PHP application on Amazon Linux 2023
    'AWS::CloudFormation::Init':
      config:
        packages:
          dnf:
            httpd: []
            php: []
            php-fpm: []
        files:
          /etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json:
            content: !Sub |
              {
                "logs": {
                  "logs_collected": {
                    "files": {
                      "collect_list": [
                        {
                          "file_path": "/var/log/httpd/access_log",
                          "log_group_name": "${WebServerLogGroup}",
                          "log_stream_name": "{instance_id}/apache.log",
                          "timestamp_format": "%d/%b/%Y:%H:%M:%S %z"
                        }
                      ]
                    }
                  }
                }
              }

```

```

        }
    }
}
mode: '000644'
owner: root
group: root
/var/www/html/index.php:
content: |
    <?php echo '<h1>AWS CloudFormation sample PHP application on Amazon
Linux 2023</h1>';
    ?>
mode: '000644'
owner: apache
group: apache
/etc/cfn/cfn-hup.conf:
content: !Sub |
    [main]
    stack=${AWS::StackId}
    region=${AWS::Region}
mode: '000400'
owner: root
group: root
/etc/cfn/hooks.d/cfn-auto-reloader.conf:
content: !Sub |
    [cfn-auto-reloader-hook]
    triggers=post.update
    path=Resources.WebServerHost.Metadata.AWS::CloudFormation::Init
    action=/opt/aws/bin/cfn-init -s ${AWS::StackId} -r WebServerHost --
region ${AWS::Region}
    runas=root
services:
  systemd:
    httpd:
      enabled: 'true'
      ensureRunning: 'true'
    php-fpm:
      enabled: 'true'
      ensureRunning: 'true'
CreationPolicy:
  ResourceSignal:
    Timeout: PT5M
Properties:
  ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-
kernel-6.1-x86_64}}'

```



```

KeyName: !Ref KeyName
InstanceType: t3.micro
SecurityGroupIds:
  - !Ref WebServerSecurityGroup
IamInstanceProfile: !Ref LogRoleInstanceProfile
UserData: !Base64
  Fn::Sub: |
    #!/bin/bash
    dnf update -y aws-cfn-bootstrap
    dnf install -y amazon-cloudwatch-agent
    /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource WebServerHost
--region ${AWS::Region}

    # Verify Apache log directory exists and create if needed
    mkdir -p /var/log/httpd

    # Start CloudWatch agent
    /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -c file:/etc/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json -s

    # Signal success
    /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
WebServerHost --region ${AWS::Region}
    echo "Done"
WebServerLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    RetentionInDays: 7
404MetricFilter:
  Type: AWS::Logs::MetricFilter
  Properties:
    LogGroupName: !Ref WebServerLogGroup
    FilterPattern: >-
      [ip, identity, user_id, timestamp, request, status_code = 404, size, ...]
    MetricTransformations:
      - MetricValue: '1'
        MetricNamespace: test/404s
        MetricName: test404Count
BytesTransferredMetricFilter:
  Type: AWS::Logs::MetricFilter
  Properties:
    LogGroupName: !Ref WebServerLogGroup

```

```
    FilterPattern: '[ip, identity, user_id, timestamp, request, status_code,
size, ...]'
```

 MetricTransformations:

- MetricValue: \$size
- MetricNamespace: test/BytesTransferred
- MetricName: testBytesTransferred

404Alarm:

 Type: AWS::CloudWatch::Alarm

 Properties:

- AlarmDescription: The number of 404s is greater than 2 over 2 minutes
- MetricName: test404Count
- Namespace: test/404s
- Statistic: Sum
- Period: '60'
- EvaluationPeriods: '2'
- Threshold: '2'
- AlarmActions:
- !Ref AlarmNotificationTopic
- ComparisonOperator: GreaterThanThreshold

BandwidthAlarm:

 Type: AWS::CloudWatch::Alarm

 Properties:

- AlarmDescription: The average volume of traffic is greater 3500 KB over 10 minutes
- MetricName: testBytesTransferred
- Namespace: test/BytesTransferred
- Statistic: Average
- Period: '300'
- EvaluationPeriods: '2'
- Threshold: '3500'
- AlarmActions:
- !Ref AlarmNotificationTopic
- ComparisonOperator: GreaterThanThreshold

AlarmNotificationTopic:

 Type: AWS::SNS::Topic

 Properties:

- Subscription:
- Endpoint: !Ref OperatorEmail
- Protocol: email

Outputs:

 InstanceId:

- Description: The instance ID of the web server
- Value: !Ref WebServerHost

 WebsiteURL:

```

    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for the web server
  PublicIP:
    Description: Public IP address of the web server
    Value: !GetAtt WebServerHost.PublicIp
  CloudWatchLogGroupName:
    Description: The name of the CloudWatch log group
    Value: !Ref WebServerLogGroup

```

從 Windows 執行個體傳送日誌到 CloudWatch Logs

以下範本會設定 CloudWatch Logs 以適用於 Windows 2012R2 執行個體。

Windows 上的 CloudWatch Logs 代理程式 (Windows 2012R2 和 Windows 2016 AMI 上的 SSM 代理程式) 只會傳送啟動之後的日誌，因此啟動之前產生的任何日誌都不會傳送。若要解決此問題，此範本可協助確保在代理程式啟動之後，才會寫入任何日誌：

- 將代理程式的設定，設定為 cfn-init config 中的第一個 configSets 項目。
- 使用 waitAfterCompletion 在啟動代理程式的命令之後插入一個暫停。

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Sample template that sets up and configures CloudWatch Logs on Windows 2012R2 instance.",
  "Parameters": {
    "KeyPair": {
      "Description": "Name of an existing EC2 KeyPair to enable RDP access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
    },
    "RDPLocation": {
      "Description": "The IP address range that can be used to RDP to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",

```

```

        "ConstraintDescription": "must be a valid IP CIDR range of the form
x.x.x.x/x."
    },
    "OperatorEmail": {
        "Description": "Email address to notify when CloudWatch alarms are
triggered (404 errors)",
        "Type": "String"
    }
},
"Resources": {
    "WebServerSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable HTTP access via port 80 and RDP access via
port 3389",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": "80",
                    "ToPort": "80",
                    "CidrIp": "0.0.0.0/0"
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": "3389",
                    "ToPort": "3389",
                    "CidrIp": {"Ref": "RDPLocation"}
                }
            ]
        }
    },
    "LogRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        }
                    }
                ]
            }
        }
    },

```

```

        "Action": [
            "sts:AssumeRole"
        ]
    },
    ],
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
    ],
    "Path": "/",
    "Policies": [
        {
            "PolicyName": "LogRolePolicy",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "logs:Create*",
                            "logs:PutLogEvents",
                            "s3:GetObject"
                        ],
                        "Resource": [
                            "arn:aws:logs:*:*:*:*",
                            "arn:aws:s3:::*"
                        ]
                    }
                ]
            }
        }
    ],
    "LogRoleInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Path": "/",
            "Roles": [{"Ref": "LogRole"}]
        }
    },
    "WebServerHost": {
        "Type": "AWS::EC2::Instance",
        "CreationPolicy": {

```

```

        "ResourceSignal": {
            "Timeout": "PT15M"
        }
    },
    "Metadata": {
        "AWS::CloudFormation::Init": {
            "configSets": {
                "config": [
                    "00-ConfigureCWLogs",
                    "01-InstallWebServer",
                    "02-ConfigureApplication",
                    "03-Finalize"
                ]
            },
            "00-ConfigureCWLogs": {
                "files": {
                    "C:\\Program Files\\Amazon\\SSM\\Plugins\\awsCloudWatch\\
\\AWS.EC2.Windows.CloudWatch.json": {
                        "content": {
                            "EngineConfiguration": {
                                "Components": [
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "ApplicationEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "Application"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SystemEventLog",
                                        "Parameters": {
                                            "Levels": "7",
                                            "LogName": "System"
                                        }
                                    },
                                    {
                                        "FullName":
"\"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch\"",
                                        "Id": "SecurityEventLog",
                                        "Parameters": {

```

```

        "Levels": "7",
        "LogName": "Security"
    }
},
{
    "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Id": "EC2ConfigLog",
    "Parameters": {
        "CultureName": "en-US",
        "Encoding": "ASCII",
        "Filter": "EC2ConfigLog.txt",
        "LogDirectoryPath": "C:\\Program

Files\\Amazon\\Ec2ConfigService\\Logs",

        "TimeZoneKind": "UTC",
        "TimestampFormat": "yyyy-MM-

ddTHH:mm:ss.fffZ:"
    }
},
{
    "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Id": "CfnInitLog",
    "Parameters": {
        "CultureName": "en-US",
        "Encoding": "ASCII",
        "Filter": "cfn-init.log",
        "LogDirectoryPath": "C:\\cfn\\log",
        "TimeZoneKind": "Local",
        "TimestampFormat": "yyyy-MM-dd

HH:mm:ss,fff"
    }
},
{
    "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Id": "IISLogs",
    "Parameters": {
        "CultureName": "en-US",
        "Encoding": "UTF-8",
        "Filter": "",
        "LineCount": "3",
        "LogDirectoryPath": "C:\\inetpub\\

logs\\LogFiles\\W3SVC1",

```

```

        "TimeZoneKind": "UTC",
        "TimestampFormat": "yyyy-MM-dd

HH:mm:ss"
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E
        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

ApplicationEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

SystemEventLog",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
}

```



```

    },
    {
      "FullName":
        "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
      "Id": "CloudWatchSecurityEventLog",
      "Parameters": {
        "AccessKey": "",
        "LogGroup": {"Ref": "LogGroup"},
        "LogStream": "{instance_id}/
SecurityEventLog",

        "Region": {"Ref": "AWS::Region"},
        "SecretKey": ""
      }
    },
    {
      "FullName":
        "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
      "Id": "CloudWatchEC2ConfigLog",
      "Parameters": {
        "AccessKey": "",
        "LogGroup": {"Ref": "LogGroup"},
        "LogStream": "{instance_id}/
EC2ConfigLog",

        "Region": {"Ref": "AWS::Region"},
        "SecretKey": ""
      }
    },
    {
      "FullName":
        "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
      "Id": "CloudWatchCfnInitLog",
      "Parameters": {
        "AccessKey": "",
        "LogGroup": {"Ref": "LogGroup"},
        "LogStream": "{instance_id}/
CfnInitLog",

        "Region": {"Ref": "AWS::Region"},
        "SecretKey": ""
      }
    },
    {
      "FullName":
        "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
      "Id": "CloudWatchIISLogs",

```

```

        "Parameters": {
            "AccessKey": "",
            "LogGroup": {"Ref": "LogGroup"},
            "LogStream": "{instance_id}/

IISLogs",

            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
            "AccessKey": "",
            "NameSpace": "Windows/Default",
            "Region": {"Ref": "AWS::Region"},
            "SecretKey": ""
        }
    }
],
"Flows": {
    "Flows": [

"ApplicationEventLog,CloudWatchApplicationEventLog",

"SystemEventLog,CloudWatchSystemEventLog",

"SecurityEventLog,CloudWatchSecurityEventLog",

"EC2ConfigLog,CloudWatchEC2ConfigLog",
"CfnInitLog,CloudWatchCfnInitLog",
"IISLogs,CloudWatchIISLogs",
"MemoryPerformanceCounter,CloudWatch"

    ]
},
    "PollInterval": "00:00:05"
},
    "IsEnabled": true
}
    },
    "commands": {
        "0-enableSSM": {

```

```

        "command": "powershell.exe -Command \"Set-Service -Name
AmazonSSMAgent -StartupType Automatic\" ",
        "waitAfterCompletion": "0"
    },
    "1-restartSSM": {
        "command": "powershell.exe -Command \"Restart-Service
AmazonSSMAgent \"",
        "waitAfterCompletion": "30"
    }
},
"01-InstallWebServer": {
    "commands": {
        "01_install_webserver": {
            "command": "powershell.exe -Command \"Install-
WindowsFeature Web-Server -IncludeAllSubFeature\"",
            "waitAfterCompletion": "0"
        }
    }
},
"02-ConfigureApplication": {
    "files": {
        "c:\\\\Inetpub\\\\wwwroot\\\\index.htm": {
            "content": "<html> <head> <title>Test Application
Page</title> </head> <body> <h1>Congratulations!! Your IIS server is configured.</h1>
</body> </html>"
        }
    }
},
"03-Finalize": {
    "commands": {
        "00_signal_success": {
            "command": {
                "Fn::Sub": "cfn-signal.exe -e 0 --resource
WebServerHost --stack ${AWS::StackName} --region ${AWS::Region}"
            },
            "waitAfterCompletion": "0"
        }
    }
}
},
"Properties": {
    "KeyName": {

```

```

        "Ref": "KeyPair"
    },
    "ImageId": "{{resolve:ssm:/aws/service/ami-windows-latest/
Windows_Server-2012-R2_RTM-English-64Bit-Base}}",
    "InstanceType": "t2.xlarge",
    "SecurityGroupIds": [{"Ref": "WebServerSecurityGroup"}],
    "IamInstanceProfile": {"Ref": "LogRoleInstanceProfile"},
    "UserData": {
        "Fn::Base64": {
            "Fn::Join": [
                "",
                [
                    "<script>\n",
                    "wmic product where \"description='Amazon SSM Agent' \"
uninstall\n",
                    "wmic product where \"description='aws-cfn-bootstrap'
\" uninstall \n",
                    "start /wait c:\\\\Windows\\\\system32\\\\msiexec /passive /
qn /i https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi\n",
                    "powershell.exe -Command \"iwr https://
s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe -UseBasicParsing -OutFile C:\\\\AmazonSSMAgentSetup.exe\"\\n",
                    "start /wait C:\\\\AmazonSSMAgentSetup.exe /install /
quiet\n",
                    "cfn-init.exe -v -c config -s ", {"Ref":
"AWS::StackName"}, " --resource WebServerHost --region ", {"Ref": "AWS::Region"}, "
\n",
                    "</script>\n"
                ]
            ]
        }
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",
        "Properties": {
            "RetentionInDays": 7
        }
    },
    "404MetricFilter": {
        "Type": "AWS::Logs::MetricFilter",
        "Properties": {

```

```

        "LogGroupName": {"Ref": "LogGroup"},
        "FilterPattern": "[timestamps, serverip, method, uri, query, port,
dash, clientip, useragent, status_code = 404, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    },
    "404Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmDescription": "The number of 404s is greater than 2 over 2
minutes",
            "MetricName": "test404Count",
            "Namespace": "test/404s",
            "Statistic": "Sum",
            "Period": "60",
            "EvaluationPeriods": "2",
            "Threshold": "2",
            "AlarmActions": [{"Ref": "AlarmNotificationTopic"}],
            "ComparisonOperator": "GreaterThanThreshold"
        }
    },
    "AlarmNotificationTopic": {
        "Type": "AWS::SNS::Topic",
        "Properties": {
            "Subscription": [{"Endpoint": {"Ref": "OperatorEmail"}, "Protocol":
"email"}]
        }
    }
},
"Outputs": {
    "InstanceId": {
        "Description": "The instance ID of the web server",
        "Value": {"Ref": "WebServerHost"}
    },
    "WebsiteURL": {
        "Value": {"Fn::Sub": "http://${WebServerHost.PublicDnsName}"},
        "Description": "URL for the web server"
    }
},

```

```

    "PublicIP": {
      "Description": "Public IP address of the web server",
      "Value": {"Fn::GetAtt": ["WebServerHost", "PublicIp"]}
    },
    "CloudWatchLogGroupName": {
      "Description": "The name of the CloudWatch log group",
      "Value": {"Ref": "LogGroup"}
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Sample template that sets up and configures CloudWatch Logs on Windows 2012R2
  instance.
Parameters:
  KeyPair:
    Description: Name of an existing EC2 KeyPair to enable RDP access to the instances
    Type: AWS::EC2::KeyPair::KeyName
    ConstraintDescription: must be the name of an existing EC2 KeyPair.
  RDPLocation:
    Description: The IP address range that can be used to RDP to the EC2 instances
    Type: String
    MinLength: '9'
    MaxLength: '18'
    Default: 0.0.0.0/0
    AllowedPattern: '(\d{1,3})\.\d{1,3})\.\d{1,3})\.\d{1,3})/(\d{1,2})'
    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
  OperatorEmail:
    Description: Email address to notify when CloudWatch alarms are triggered (404
    errors)
    Type: String
Resources:
  WebServerSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable HTTP access via port 80 and RDP access via port 3389
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'

```

```
    CidrIp: 0.0.0.0/0
  - IpProtocol: tcp
    FromPort: '3389'
    ToPort: '3389'
    CidrIp: !Ref RDPLocation
LogRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - 'sts:AssumeRole'
    ManagedPolicyArns:
      - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'
  Path: /
  Policies:
    - PolicyName: LogRolePolicy
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 'logs:Create*'
              - 'logs:PutLogEvents'
              - 's3:GetObject'
            Resource:
              - 'arn:aws:logs:*:*:*'
              - 'arn:aws:s3::*:*'
LogRoleInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref LogRole
WebServerHost:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT15M
```

```

Metadata:
  'AWS::CloudFormation::Init':
    configSets:
      config:
        - 00-ConfigureCWLogs
        - 01-InstallWebServer
        - 02-ConfigureApplication
        - 03-Finalize
    00-ConfigureCWLogs:
      files:
        'C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch
\AWS.EC2.Windows.CloudWatch.json':
          content: !Sub |
            {
              "EngineConfiguration": {
                "Components": [
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "ApplicationEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Application"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SystemEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "System"
                    }
                  },
                  {
                    "FullName":
"AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
                    "Id": "SecurityEventLog",
                    "Parameters": {
                      "Levels": "7",
                      "LogName": "Security"
                    }
                  }
                ]
              }
            }

```



```

        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "EC2ConfigLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "EC2ConfigLog.txt",
            "LogDirectoryPath": "C:\\\\Program Files\\\\Amazon\\
\Ec2ConfigService\\\\Logs",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-ddTHH:mm:ss.fffZ:"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CfnInitLog",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "ASCII",
            "Filter": "cfn-init.log",
            "LogDirectoryPath": "C:\\\\cfn\\\\log",
            "TimeZoneKind": "Local",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss,fff"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "IISLogs",
        "Parameters": {
            "CultureName": "en-US",
            "Encoding": "UTF-8",
            "Filter": "",
            "LineCount": "3",
            "LogDirectoryPath": "C:\\\\inetpub\\\\logs\\\\LogFiles\\
\W3SVC1",
            "TimeZoneKind": "UTC",
            "TimestampFormat": "yyyy-MM-dd HH:mm:ss"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.PerformanceCounterInputComponent,AWS.E

```

```

        "Id": "MemoryPerformanceCounter",
        "Parameters": {
            "CategoryName": "Memory",
            "CounterName": "Available MBytes",
            "DimensionName": "",
            "DimensionValue": "",
            "InstanceName": "",
            "MetricName": "Memory",
            "Unit": "Megabytes"
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchApplicationEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/ApplicationEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSystemEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SystemEventLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchSecurityEventLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/SecurityEventLog",
            "Region": "${AWS::Region}",

```

```

        "SecretKey": ""
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchEC2ConfigLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/EC2ConfigLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchCfnInitLog",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/CfnInitLog",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatchIISLogs",
        "Parameters": {
            "AccessKey": "",
            "LogGroup": "${LogGroup}",
            "LogStream": "{instance_id}/IISLogs",
            "Region": "${AWS::Region}",
            "SecretKey": ""
        }
    },
    {
        "FullName":
"AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {

```

```

        "AccessKey": "",
        "Namespace": "Windows/Default",
        "Region": "${AWS::Region}",
        "SecretKey": ""
    }
},
"Flows": {
    "Flows": [
        "ApplicationEventLog,CloudWatchApplicationEventLog",
        "SystemEventLog,CloudWatchSystemEventLog",
        "SecurityEventLog,CloudWatchSecurityEventLog",
        "EC2ConfigLog,CloudWatchEC2ConfigLog",
        "CfnInitLog,CloudWatchCfnInitLog",
        "IISLogs,CloudWatchIISLogs",
        "MemoryPerformanceCounter,CloudWatch"
    ]
},
"PollInterval": "00:00:05"
},
"IsEnabled": true
}
commands:
  0-enableSSM:
    command: >-
      powershell.exe -Command "Set-Service -Name AmazonSSMAgent
        -StartupType Automatic"
    waitAfterCompletion: '0'
  1-restartSSM:
    command: powershell.exe -Command "Restart-Service AmazonSSMAgent "
    waitAfterCompletion: '30'
01-InstallWebServer:
  commands:
    01_install_webserver:
      command: >-
        powershell.exe -Command "Install-WindowsFeature Web-Server
          -IncludeAllSubFeature"
      waitAfterCompletion: '0'
02-ConfigureApplication:
  files:
    'c:\Inetpub\wwwroot\index.htm':
      content: >-
        <html> <head> <title>Test Application Page</title> </head>
        <body> <h1>Congratulations !! Your IIS server is

```

```

        configured.</h1> </body> </html>
03-Finalize:
  commands:
    00_signal_success:
      command: !Sub >-
        cfn-signal.exe -e 0 --resource WebServerHost --stack
          ${AWS::StackName} --region ${AWS::Region}
      waitAfterCompletion: '0'
  Properties:
    KeyName: !Ref KeyPair
    ImageId: "{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2012-
R2_RTM-English-64Bit-Base}}"
    InstanceType: t2.xlarge
    SecurityGroupIds:
      - !Ref WebServerSecurityGroup
    IamInstanceProfile: !Ref LogRoleInstanceProfile
    UserData: !Base64
      'Fn::Sub': >
        <script>

          wmic product where "description='Amazon SSM Agent' " uninstall

          wmic product where "description='aws-cfn-bootstrap' " uninstall

          start /wait c:\\Windows\\system32\\msiexec /passive /qn /i
            https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-win64-
latest.msi

          powershell.exe -Command "iwr
            https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/windows_amd64/
AmazonSSMAgentSetup.exe
            -UseBasicParsing -OutFile C:\\AmazonSSMAgentSetup.exe"

          start /wait C:\\AmazonSSMAgentSetup.exe /install /quiet

          cfn-init.exe -v -c config -s ${AWS::StackName} --resource
            WebServerHost --region ${AWS::Region}

        </script>
    LogGroup:
      Type: AWS::Logs::LogGroup
      Properties:
        RetentionInDays: 7
    404MetricFilter:

```

```
Type: AWS::Logs::MetricFilter
Properties:
  LogGroupName: !Ref LogGroup
  FilterPattern: >-
    [timestamps, serverip, method, uri, query, port, dash, clientip,
     useragent, status_code = 404, ...]
  MetricTransformations:
    - MetricValue: '1'
      MetricNamespace: test/404s
      MetricName: test404Count
404Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmDescription: The number of 404s is greater than 2 over 2 minutes
    MetricName: test404Count
    Namespace: test/404s
    Statistic: Sum
    Period: '60'
    EvaluationPeriods: '2'
    Threshold: '2'
    AlarmActions:
      - !Ref AlarmNotificationTopic
    ComparisonOperator: GreaterThanThreshold
AlarmNotificationTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: !Ref OperatorEmail
        Protocol: email
Outputs:
  InstanceId:
    Description: The instance ID of the web server
    Value: !Ref WebServerHost
  WebsiteURL:
    Value: !Sub 'http://${WebServerHost.PublicDnsName}'
    Description: URL for the web server
  PublicIP:
    Description: Public IP address of the web server
    Value: !GetAtt
      - WebServerHost
      - PublicIp
  CloudWatchLogGroupName:
    Description: The name of the CloudWatch log group
```

Value: !Ref LogGroup

另請參閱

如需有關 CloudWatch Logs 資源的詳細資訊，請參閱 [AWS::Logs::LogGroup](#) 或 [AWS::Logs::MetricFilter](#)。

Amazon DynamoDB 範本程式碼片段

主題

- [Application Auto Scaling 搭配 Amazon DynamoDB 資料表](#)
- [另請參閱](#)

Application Auto Scaling 搭配 Amazon DynamoDB 資料表

此範例會為 `AWS::DynamoDB::Table` 資源設定應用程式 Auto Scaling。範本所定義的 `TargetTrackingScaling` 擴展政策，則可向上擴展資料表的 `WriteCapacityUnits` 傳輸量。

JSON

```
{
  "Resources": {
    "DDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [
          {
            "AttributeName": "ArtistId",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Concert",
            "AttributeType": "S"
          },
          {
            "AttributeName": "TicketSales",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
```

```

        "AttributeName": "ArtistId",
        "KeyType": "HASH"
    },
    {
        "AttributeName": "Concert",
        "KeyType": "RANGE"
    }
],
"GlobalSecondaryIndexes": [
    {
        "IndexName": "GSI",
        "KeySchema": [
            {
                "AttributeName": "TicketSales",
                "KeyType": "HASH"
            }
        ],
        "Projection": {
            "ProjectionType": "KEYS_ONLY"
        },
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        }
    }
],
"ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
}
},
"WriteCapacityScalableTarget": {
    "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties": {
        "MaxCapacity": 15,
        "MinCapacity": 5,
        "ResourceId": {
            "Fn::Join": [
                "/",
                [
                    "table",
                    {
                        "Ref": "DDBTable"
                    }
                ]
            ]
        }
    }
}

```



```
    }  
  ]  
]  
},  
  "RoleARN" : { "Fn::Sub" : "arn:aws:iam::${AWS::AccountId}:role/  
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable" },  
  "ScalableDimension": "dynamodb:table:WriteCapacityUnits",  
  "ServiceNamespace": "dynamodb"  
}  
},  
"WriteScalingPolicy": {  
  "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",  
  "Properties": {  
    "PolicyName": "WriteAutoScalingPolicy",  
    "PolicyType": "TargetTrackingScaling",  
    "ScalingTargetId": {  
      "Ref": "WriteCapacityScalableTarget"  
    },  
    "TargetTrackingScalingPolicyConfiguration": {  
      "TargetValue": 50,  
      "ScaleInCooldown": 60,  
      "ScaleOutCooldown": 60,  
      "PredefinedMetricSpecification": {  
        "PredefinedMetricType": "DynamoDBWriteCapacityUtilization"  
      }  
    }  
  }  
}  
}  
}
```

YAML

```
Resources:
  DDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
        - AttributeName: "ArtistId"
          AttributeType: "S"
        - AttributeName: "Concert"
          AttributeType: "S"
```

```

    - AttributeName: "TicketSales"
      AttributeType: "S"
  KeySchema:
    - AttributeName: "ArtistId"
      KeyType: "HASH"
    - AttributeName: "Concert"
      KeyType: "RANGE"
  GlobalSecondaryIndexes:
    - IndexName: "GSI"
      KeySchema:
        - AttributeName: "TicketSales"
          KeyType: "HASH"
      Projection:
        ProjectionType: "KEYS_ONLY"
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 5
  WriteCapacityScalableTarget:
    Type: AWS::ApplicationAutoScaling::ScalableTarget
    Properties:
      MaxCapacity: 15
      MinCapacity: 5
      ResourceId: !Join
        - /
        - - table
          - !Ref DDBTable
      RoleARN:
        Fn::Sub: 'arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable'
      ScalableDimension: dynamodb:table:WriteCapacityUnits
      ServiceNamespace: dynamodb
  WriteScalingPolicy:
    Type: AWS::ApplicationAutoScaling::ScalingPolicy
    Properties:
      PolicyName: WriteAutoScalingPolicy
      PolicyType: TargetTrackingScaling
      ScalingTargetId: !Ref WriteCapacityScalableTarget
      TargetTrackingScalingPolicyConfiguration:
        TargetValue: 50.0
        ScaleInCooldown: 60

```

```
ScaleOutCooldown: 60
PredefinedMetricSpecification:
  PredefinedMetricType: DynamoDBWriteCapacityUtilization
```

另請參閱

如需詳細資訊，請參閱 資料庫部落格上的部落格文章[如何使用 CloudFormation 設定 DynamoDB 資料表和索引的自動擴展](#) AWS。

如需有關 DynamoDB 資源的詳細資訊，請參閱 [AWS::DynamoDB::Table](#)。

Amazon EC2 CloudFormation 範本程式碼片段

Amazon EC2 在 AWS 雲端中提供可擴展的運算容量。您可使用 Amazon EC2 按需要啟動任意數量的虛擬伺服器，設定安全性和聯網功能以及管理儲存。這些虛擬伺服器 (稱為執行個體) 可執行各種作業系統和應用程式，並且可進行自訂，以滿足您的特定需求。Amazon EC2 可讓您縱向擴展或縮減，以處理需求變更或用量尖峰。

您可使用 CloudFormation 範本來定義和佈建 Amazon EC2 執行個體，作為基礎設施的一部分。範本可讓您以可重複且一致的方式，輕鬆管理和自動化 Amazon EC2 資源的部署。

下列範例範本程式碼片段描述了 Amazon EC2 的 CloudFormation 資源或元件。這些程式碼片段設計用於整合至範本中，而且並非設計為獨立執行。

程式碼片段類別

- [使用 CloudFormation 設定 Amazon EC2 執行個體](#)
- [透過 CloudFormation 建立啟動範本](#)
- [使用 CloudFormation 管理安全群組](#)
- [配置彈性 IP 位址並與 CloudFormation 產生關聯](#)
- [使用 CloudFormation 來設定 Amazon VPC 資源](#)

使用 CloudFormation 設定 Amazon EC2 執行個體

下列程式碼片段示範了如何使用 CloudFormation 設定 Amazon EC2 執行個體。

程式碼片段類別

- [一般 Amazon EC2 組態](#)
- [指定執行個體的區塊型裝置映射](#)

一般 Amazon EC2 組態

下列程式碼片段示範了使用 CloudFormation 設定 Amazon EC2 執行個體的一般組態。

範例程式碼片段

- [在指定的可用區域建立 Amazon EC2 執行個體](#)
- [使用 EBS 磁碟區和使用者資料設定標記的 Amazon EC2 執行個體](#)
- [在 Amazon EC2 執行個體啟動的使用者資料中定義 DynamoDB 資料表名稱](#)
- [使用 DeletionPolicy 建立 Amazon EBS 磁碟區](#)

在指定的可用區域建立 Amazon EC2 執行個體

下列程式碼片段使用 [AWS::EC2::Instance](#) 資源，在指定的可用區域建立 Amazon EC2 執行個體。可用區域的代碼為其區域代碼，後續跟著一個字母識別符。您可在單一可用區域啟動執行個體。

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "AvailabilityZone": "aa-example-1a",
    "ImageId": "ami-1234567890abcdef0"
  }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0
```

使用 EBS 磁碟區和使用者資料設定標記的 Amazon EC2 執行個體

下列程式碼片段會建立具有標籤、EBS 磁碟區和使用者資料的 Amazon EC2 執行個體。它會使用 [AWS::EC2::Instance](#) 資源。在相同的範本中，您必須定義 [AWS::EC2::SecurityGroup](#) 資源、[AWS::SNS::Topic](#) 資源和 [AWS::EC2::Volume](#) 資源。必須在範本的 Parameters 區段中定義 KeyName。

標籤可協助您根據您的偏好設定分類 AWS 資源，例如依用途、擁有者或環境。使用者資料允許在啟動期間，將自訂指令碼或資料佈建至執行個體。此資料有助於在初始化期間，對執行個體執行任務自動化、軟體組態設定、套件安裝及其他動作。

如需有關標記您的資源的詳細資訊，請參閱《[Amazon EC2 使用者指南](#)》中的標記您的 Amazon EC2 資源。

如需關於使用者資料的資訊，請參閱《Amazon EC2 使用者指南》中的[使用執行個體中繼資料管理您的 EC2 執行個體](#)。

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups": [ { "Ref": "Ec2SecurityGroup" } ],
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [ ":", [
          "PORT=80",
          "TOPIC=",
          { "Ref": "MySNSTopic" }
        ]
        ]
      }
    },
    "InstanceType": "aa.size",
    "AvailabilityZone": "aa-example-1a",
    "ImageId": "ami-1234567890abcdef0",
    "Volumes": [
      {
        "VolumeId": { "Ref": "MyVolumeResource" },
        "Device": "/dev/sdk"
      }
    ],
    "Tags": [ { "Key": "Name", "Value": "MyTag" } ]
  }
}
```

YAML

```
Ec2Instance:
```

```

Type: AWS::EC2::Instance
Properties:
  KeyName: !Ref KeyName
  SecurityGroups:
    - !Ref Ec2SecurityGroup
  UserData:
    Fn::Base64:
      Fn::Join:
        - ":"
        - - "PORT=80"
          - "TOPIC="
          - !Ref MySNSTopic
  InstanceType: aa.size
  AvailabilityZone: aa-example-1a
  ImageId: ami-1234567890abcdef0
  Volumes:
    - VolumeId: !Ref MyVolumeResource
      Device: "/dev/sdk"
  Tags:
    - Key: Name
      Value: MyTag

```

在 Amazon EC2 執行個體啟動的使用者資料中定義 DynamoDB 資料表名稱

下列程式碼片段會建立 Amazon EC2 執行個體，並在使用者資料中定義 DynamoDB 資料表名稱，以便在啟動時傳遞至執行個體。它會使用 [AWS::EC2::Instance](#) 資源。您可在使用者資料中定義參數或動態值，以便在啟動時傳遞 EC2 執行個體。

如需關於使用者資料的更多資訊，請參閱《Amazon EC2 使用者指南》中的[使用執行個體中繼資料管理您的 EC2 執行個體](#)。

JSON

```

"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [
          "",
          [
            "TableName=",
            {

```

```

        "Ref": "DynamoDBTableName"
      }
    ]
  }
},
"AvailabilityZone": "aa-example-1a",
"ImageId": "ami-1234567890abcdef0"
}
}

```

YAML

```

Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    UserData:
      Fn::Base64:
        Fn::Join:
          - ''
          - - 'TableName='
            - Ref: DynamoDBTableName
    AvailabilityZone: aa-example-1a
    ImageId: ami-1234567890abcdef0

```

使用 **DeletionPolicy** 建立 Amazon EBS 磁碟區

下列程式碼片段使用 Amazon EC2 [AWS::EC2::Volume](#) 資源來建立 Amazon EBS 磁碟區。您可使用 `Size` 或 `SnapshotID` 屬性來定義磁碟區，但不能同時使用兩者。`DeletionPolicy` 屬性設定為在刪除堆疊時建立磁碟區快照。

如需有關 `DeletionPolicy` 屬性的詳細資訊，請參閱 [DeletionPolicy 屬性](#)。

如需有關建立 Amazon EBS 磁碟區的詳細資訊，請參閱 [建立 Amazon EBS 磁碟區](#)。

JSON

此程式碼片段會建立具有指定大小的 Amazon EBS 磁碟區。大小設定為 10，但您可視需進行調整。[AWS::EC2::Volume](#) 資源允許您指定大小或快照 ID，但不能同時指定兩者。

```

"MyEBSVolume": {
  "Type": "AWS::EC2::Volume",

```

```
"Properties": {
  "Size": "10",
  "AvailabilityZone": {
    "Ref": "AvailabilityZone"
  }
},
"DeletionPolicy": "Snapshot"
}
```

此程式碼片段使用提供的快照 ID 來建立 Amazon EBS 磁碟區。[AWS::EC2::Volume](#) 資源允許您指定大小或快照 ID，但不能同時指定兩者。

```
"MyEBSVolume": {
  "Type": "AWS::EC2::Volume",
  "Properties": {
    "SnapshotId" : "snap-1234567890abcdef0",
    "AvailabilityZone": {
      "Ref": "AvailabilityZone"
    }
  },
  "DeletionPolicy": "Snapshot"
}
```

YAML

此程式碼片段會建立具有指定大小的 Amazon EBS 磁碟區。大小設定為 10，但您可視需進行調整。[AWS::EC2::Volume](#) 資源允許您指定大小或快照 ID，但不能同時指定兩者。

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 10
    AvailabilityZone:
      Ref: AvailabilityZone
  DeletionPolicy: Snapshot
```

此程式碼片段使用提供的快照 ID 來建立 Amazon EBS 磁碟區。[AWS::EC2::Volume](#) 資源允許您指定大小或快照 ID，但不能同時指定兩者。

```
MyEBSVolume:
  Type: AWS::EC2::Volume
  Properties:
```



```
SnapshotId: snap-1234567890abcdef0
AvailabilityZone:
  Ref: AvailabilityZone
DeletionPolicy: Snapshot
```

指定執行個體的區塊型裝置映射

區塊型裝置映射定義區塊型儲存裝置，其中包括要附接至執行個體的執行個體儲存體磁碟區和 EBS 磁碟區。您可在建立 AMI 時指定區塊型裝置映射，如此從 AMI 啟動的所有執行個體都會使用該映射。或者，也可在啟動執行個體時指定區塊型裝置映射，如此該映射會覆寫啟動執行個體之 AMI 中指定的映射。

您可使用下列範本程式碼片段，以使用 [AWS::EC2::Instance](#) 資源的 `BlockDeviceMappings` 屬性為 EBS 或執行個體儲存體磁碟區指定區塊型裝置映射。

如需區塊型儲存設備映射的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Amazon EC2 執行個體上磁碟區的區塊型儲存設備映射](#)。

案例

- [為兩個 EBS 磁碟區指定區塊型裝置映射](#)
- [為執行個體儲存體磁碟區指定區塊型裝置映射](#)

為兩個 EBS 磁碟區指定區塊型裝置映射

JSON

```
"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
    "KeyName": { "Ref": "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups": [{ "Ref": "Ec2SecurityGroup" }],
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/sda1",
        "Ebs": { "VolumeSize": "50" }
      },
      {
        "DeviceName": "/dev/sdm",
```

```

        "Ebs": { "VolumeSize": "100" }
      }
    ]
  }
}

```

YAML

```

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      -
        DeviceName: /dev/sda1
        Ebs:
          VolumeSize: 50
      -
        DeviceName: /dev/sdm
        Ebs:
          VolumeSize: 100

```

為執行個體儲存體磁碟區指定區塊型裝置映射

JSON

```

"Ec2Instance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "ImageId" : "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2}}",
    "KeyName" : { "Ref" : "KeyName" },
    "InstanceType": { "Ref": "InstanceType" },
    "SecurityGroups" : [ { "Ref" : "Ec2SecurityGroup" } ],
    "BlockDeviceMappings" : [
      {
        "DeviceName" : "/dev/sdc",

```

```
        "VirtualName" : "ephemeral0"
      }
    ]
  }
}
```

YAML

```
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    KeyName: !Ref KeyName
    InstanceType: !Ref InstanceType
    SecurityGroups:
      - !Ref Ec2SecurityGroup
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        VirtualName: ephemeral0
```

透過 CloudFormation 建立啟動範本

本節提供一個透過 CloudFormation 建立 Amazon EC2 啟動範本的範例。啟動範本可讓您建立用於在 AWS 中設定和佈建 Amazon EC2 執行個體的範本。使用啟動範本，您可以來存放啟動參數，如此您就不需要在每次啟動執行個體時指定參數。如需更多範例，請參閱 `AWS::EC2::LaunchTemplate` 中的 [範例](#) 區段。

如需有關啟動範本的詳細資訊，請參閱《[Amazon EC2 使用者指南](#)》中的在 Amazon EC2 啟動範本中儲存執行個體啟動參數。

如需有關使用 Auto Scaling 群組建立啟動範本的資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的 [Auto Scaling 啟動範本](#)。

程式碼片段類別

- [建立指定安全群組、標籤、使用者資料及 IAM 角色的啟動範本](#)

建立指定安全群組、標籤、使用者資料及 IAM 角色的啟動範本

此程式碼片段顯示 `AWS::EC2::LaunchTemplate` 資源，其中包含啟動執行個體的組態資訊。您可為 `ImageId`、`InstanceType`、`SecurityGroups`、`UserData` 以及 `TagSpecifications` 屬性指定

值。SecurityGroups 屬性指定現有 EC2 安全群組以及新的安全群組。Ref 函數取得在堆疊範本中其他處宣告的 [AWS::EC2::SecurityGroup](#) 資源 myNewEC2SecurityGroup 之 ID。

啟動範本包含自訂使用者資料的區段。您可以在本區段中執行個體啟動時傳入執行的組態任務和指令碼。在此範例中，使用者資料會安裝 AWS Systems Manager 代理程式並啟動 代理程式。

啟動範本還包含 IAM 角色，該角色允許在執行個體上執行的應用程式代表您執行動作。此範例顯示啟動範本的 [AWS::IAM::Role](#) 資源，此資源使用 IamInstanceProfile 屬性指定 IAM 角色。Ref 函數取得 [AWS::IAM::InstanceProfile](#) 資源 myInstanceProfile 的名稱。若要設定 IAM 角色的許可，請為 ManagedPolicyArns 屬性指定值。

JSON

```
{
  "Resources":{
    "myLaunchTemplate":{
      "Type":"AWS::EC2::LaunchTemplate",
      "Properties":{
        "LaunchTemplateName":{" Fn::Sub": "${AWS::StackName}-launch-template" },
        "LaunchTemplateData":{
          "ImageId":"ami-02354e95b3example",
          "InstanceType":"t3.micro",
          "IamInstanceProfile":{
            "Name":{
              "Ref":"myInstanceProfile"
            }
          },
        },
        "SecurityGroupIds":[
          {
            "Ref":"myNewEC2SecurityGroup"
          },
          "sg-083cd3bfb8example"
        ],
        "UserData":{
          "Fn::Base64":{
            "Fn::Join": [
              "", [
                "#!/bin/bash\n",
                "cd /tmp\n",
                "yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm\n",
                "systemctl enable amazon-ssm-agent\n",
                "systemctl start amazon-ssm-agent\n"
              ]
            ]
          }
        }
      }
    }
  }
```

```

        ]
      ]
    }
  },
  "TagSpecifications":[
    {
      "ResourceType":"instance",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    },
    {
      "ResourceType":"volume",
      "Tags":[
        {
          "Key":"environment",
          "Value":"development"
        }
      ]
    }
  ]
}
},
"myInstanceRole":{
  "Type":"AWS::IAM::Role",
  "Properties":{
    "RoleName":"InstanceRole",
    "AssumeRolePolicyDocument":{
      "Version": "2012-10-17",
      "Statement":[
        {
          "Effect":"Allow",
          "Principal":{
            "Service":[
              "ec2.amazonaws.com"
            ]
          },
          "Action":[
            "sts:AssumeRole"
          ]
        }
      ]
    }
  }
}

```

```

    }
  ]
},
"ManagedPolicyArns":[
  "arn:aws:iam::aws:policy/myCustomerManagedPolicy"
]
}
},
"myInstanceProfile":{
  "Type":"AWS::IAM::InstanceProfile",
  "Properties":{
    "Path":"/",
    "Roles":[
      {
        "Ref":"myInstanceRole"
      }
    ]
  }
}
}
}
}
}

```

YAML

```

---
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: ami-02354e95b3example
        InstanceType: t3.micro
        IamInstanceProfile:
          Name: !Ref myInstanceProfile
        SecurityGroupIds:
          - !Ref myNewEC2SecurityGroup
          - sg-083cd3bfb8example
        UserData:
          Fn::Base64: !Sub |
            #!/bin/bash
            cd /tmp

```

```

yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/
latest/linux_amd64/amazon-ssm-agent.rpm
systemctl enable amazon-ssm-agent
systemctl start amazon-ssm-agent

```

TagSpecifications:

- ResourceType: instance

Tags:

- Key: *environment*

Value: *development*

- ResourceType: volume

Tags:

- Key: *environment*

Value: *development*

myInstanceRole:

Type: AWS::IAM::Role

Properties:

RoleName: InstanceRole

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Principal:

Service:

- 'ec2.amazonaws.com'

Action:

- 'sts:AssumeRole'

ManagedPolicyArns:

- 'arn:aws:iam::aws:policy/*myCustomerManagedPolicy*'

myInstanceProfile:

Type: AWS::IAM::InstanceProfile

Properties:

Path: '/'

Roles:

- !Ref myInstanceRole

使用 CloudFormation 管理安全群組

下列程式碼片段示範如何使用 CloudFormation 管理安全群組和 Amazon EC2 執行個體，以控制對 AWS 資源的存取。

程式碼片段類別

- [將 Amazon EC2 執行個體與安全群組建立關聯](#)

- [使用傳入規則建立安全群組](#)
- [使用安全群組傳入規則建立 Elastic Load Balancer](#)

將 Amazon EC2 執行個體與安全群組建立關聯

下列範例程式碼片段示範了如何使用 CloudFormation，將 Amazon EC2 執行個體與預設 Amazon VPC 安全群組建立關聯。

範例程式碼片段

- [將 Amazon EC2 執行個體與預設 VPC 安全群組建立關聯](#)
- [建立具有附接磁碟區和安全群組的 Amazon EC2 執行個體](#)

將 Amazon EC2 執行個體與預設 VPC 安全群組建立關聯

下列程式碼片段會建立 Amazon VPC、VPC 內的子網路以及 Amazon EC2 執行個體。VPC 使用 [AWS::EC2::VPC](#) 資源建立。VPC 的 IP 地址範圍在較大的範本中定義，並且 MyVPCCIDRRange 參數會參考該範圍。

子網路在 VPC 內使用 [AWS::EC2::Subnet](#) 資源。子網路與 VPC 關聯，以 MyVPC 做為參考。

EC2 執行個體在 VPC 和子網路內使用 [AWS::EC2::Instance](#) 資源啟動。此資源會指定用於啟動執行個體的 Amazon Machine Image (AMI)、執行執行個體所在的子網路，以及要與執行個體建立關聯的安全群組。ImageId 會使用 Systems Manager 參數動態擷取最新的 Amazon Linux 2 AMI。

安全群組 ID 使用 Fn::GetAtt 函數取得，該函數會從 MyVPC 資源擷取預設安全群組。

執行個體會放置在程式碼片段定義的 MySubnet 資源內。

當您使用 CloudFormation 建立 VPC 時，AWS 會自動在 VPC 內建立預設資源，包括預設安全群組。但是，若您在 CloudFormation 範本內定義 VPC，在建立範本時，可能無法存取這些預設資源的 ID。若要存取和使用範本中指定的預設資源，您可以使用內部函數，例如 Fn::GetAtt。此函數可讓您使用 CloudFormation 自動建立的預設資源。

JSON

```
"MyVPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": {
```



```

        "Ref": "MyVPCCIDRRange"
      },
      "EnableDnsSupport": false,
      "EnableDnsHostnames": false,
      "InstanceTenancy": "default"
    }
  },
  "MySubnet": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "CidrBlock": {
        "Ref": "MyVPCCIDRRange"
      },
      "VpcId": {
        "Ref": "MyVPC"
      }
    }
  },
  "MyInstance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
      "SecurityGroupIds": [
        {
          "Fn::GetAtt": [
            "MyVPC",
            "DefaultSecurityGroup"
          ]
        }
      ],
      "SubnetId": {
        "Ref": "MySubnet"
      }
    }
  }
}

```

YAML

```

MyVPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock:

```

```

    Ref: MyVPCCIDRRange
    EnableDnsSupport: false
    EnableDnsHostnames: false
    InstanceTenancy: default
MySubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock:
      Ref: MyVPCCIDRRange
    VpcId:
      Ref: MyVPC
MyInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2}}'
    SecurityGroupIds:
      - Fn::GetAtt:
          - MyVPC
          - DefaultSecurityGroup
    SubnetId:
      Ref: MySubnet

```

建立具有附接磁碟區和安全群組的 Amazon EC2 執行個體

下列程式碼片段會使用 [AWS::EC2::Instance](#) 資源來建立 Amazon EC2 執行個體，該資源從指定的 AMI 啟動。執行個體與安全群組關聯，允許使用 [AWS::EC2::SecurityGroup](#) 資源，在連接埠 22 上從指定的 IP 地址傳入 SSH 流量。它會使用 [AWS::EC2::Volume](#) 資源來建立 100 GB 的 Amazon EBS 磁碟區。磁碟區建立在與執行個體相同的可用區域 (如 GetAtt 函數所指定)，並掛接至 /dev/sdh 裝置上的執行個體。

如需有關建立 Amazon EBS 磁碟區的詳細資訊，請參閱[建立 Amazon EBS 磁碟區](#)。

JSON

```

"Ec2Instance": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "SecurityGroups": [
      {
        "Ref": "InstanceSecurityGroup"
      }
    ],

```

```
        "ImageId": "ami-1234567890abcdef0"
    },
    "InstanceSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Enable SSH access via port 22",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": "22",
                    "ToPort": "22",
                    "CidrIp": "192.0.2.0/24"
                }
            ]
        }
    },
    "NewVolume": {
        "Type": "AWS::EC2::Volume",
        "Properties": {
            "Size": "100",
            "AvailabilityZone": {
                "Fn::GetAtt": [
                    "Ec2Instance",
                    "AvailabilityZone"
                ]
            }
        }
    },
    "MountPoint": {
        "Type": "AWS::EC2::VolumeAttachment",
        "Properties": {
            "InstanceId": {
                "Ref": "Ec2Instance"
            },
            "VolumeId": {
                "Ref": "NewVolume"
            },
            "Device": "/dev/sdh"
        }
    }
}
```

YAML

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    SecurityGroups:
      - !Ref InstanceSecurityGroup
    ImageId: ami-1234567890abcdef0
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24
NewVolume:
  Type: AWS::EC2::Volume
  Properties:
    Size: 100
    AvailabilityZone: !GetAtt [Ec2Instance, AvailabilityZone]
MountPoint:
  Type: AWS::EC2::VolumeAttachment
  Properties:
    InstanceId: !Ref Ec2Instance
    VolumeId: !Ref NewVolume
    Device: /dev/sdh
```

使用傳入規則建立安全群組

下列範例程式碼片段示範了如何透過使用 CloudFormation 的特定傳入規則來設定安全群組。

程式碼片段

- [使用傳入規則建立安全群組以進行 SSH 和 HTTP 存取](#)
- [使用傳入規則建立安全群組，以從指定的 CIDR 範圍進行 HTTP 和 SSH 存取](#)
- [使用傳入規則建立相互參照的安全群組](#)

使用傳入規則建立安全群組以進行 SSH 和 HTTP 存取

下列程式碼片段描述了使用 [AWS::EC2::SecurityGroup](#) 資源的兩個安全群組傳入規則。第一個輸入規則允許 SSH (連接埠 22) 從名為 `MyAdminSecurityGroup` 的現有安全群組進行存取，該安全群組由帳號為 AWS 的帳戶所擁有 1111-2222-3333。第二個傳入規則允許從稱為 `MySecurityGroupCreatedInCFN` 的不同安全群組存取 HTTP (通訊埠 80)，該安全群組在相同範本中建立。Ref 函數用於參考在相同範本中建立的安全群組的邏輯名稱。

在第一個傳入規則中，您必須為 `SourceSecurityGroupName` 和 `SourceSecurityGroupOwnerId` 屬性新增值。在第二個傳入規則中，`MySecurityGroupCreatedInCFNTemplate` 會參考在相同範本中建立的不同安全群組。確認邏輯名稱 `MySecurityGroupCreatedInCFNTemplate` 與您在較大範本中指定的安全群組資源的實際邏輯名稱相符。

如需安全群組的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Amazon EC2 執行個體的 Amazon EC2 安全群組](#)。

JSON

```
"SecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified source security group",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "SourceSecurityGroupName": "MyAdminSecurityGroup",
        "SourceSecurityGroupOwnerId": "1111-2222-3333"
      },
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "SourceSecurityGroupName": {
          "Ref": "MySecurityGroupCreatedInCFNTemplate"
        }
      }
    ]
  }
}
```

YAML

```
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified source security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        SourceSecurityGroupName: MyAdminSecurityGroup
        SourceSecurityGroupOwnerId: '1111-2222-3333'
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        SourceSecurityGroupName:
          Ref: MySecurityGroupCreatedInCFNTemplate
```

使用傳入規則建立安全群組，以從指定的 CIDR 範圍進行 HTTP 和 SSH 存取

下列程式碼片段會針對具有兩個傳入規則的 Amazon EC2 執行個體建立安全群組。傳入規則允許從指定 CIDR 範圍的指定連接埠上傳入 TCP 流量。[AWS::EC2::SecurityGroup](#) 資源用於指定規則。您必須針對每項規則指定協定。針對 TCP，您必須指定連接埠或連接埠範圍。如果您未指定來源安全群組或 CIDR 範圍，堆疊會成功啟動，但不會將規則套用至安全群組。

如需安全群組的詳細資訊，請參閱《Amazon EC2 使用者指南》中的 [Amazon EC2 執行個體的 Amazon EC2 安全群組](#)。

JSON

```
"ServerSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Allow connections from specified CIDR ranges",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "CidrIp": "192.0.2.0/24"
      },
      {
```

```

        "IpProtocol": "tcp",
        "FromPort": "22",
        "ToPort": "22",
        "CidrIp": "192.0.2.0/24"
    }
]
}
}

```

YAML

```

ServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Allow connections from specified CIDR ranges
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 192.0.2.0/24
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 192.0.2.0/24

```

使用傳入規則建立相互參照的安全群組

下列程式碼片段使用 [AWS::EC2::SecurityGroup](#) 資源來建立兩個 Amazon EC2 安全群組 (SGroup1 和 SGroup2)。允許兩個安全群組之間進行通訊的傳入規則使用 [AWS::EC2::SecurityGroupIngress](#) 資源建立。SGroup1Ingress 會建立 SGroup1 的傳入規則，以允許來源安全群組 SGroup2 連接埠 80 上的傳入 TCP 流量。SGroup2Ingress 會建立 SGroup2 的傳入規則，以允許來源安全群組 SGroup1 連接埠 80 上的傳入 TCP 流量。

JSON

```

"SGroup1": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "EC2 instance access"
  }
},
"SGroup2": {

```

```

    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "EC2 instance access"
    }
  },
  "SGroup1Ingress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupName": {
        "Ref": "SGroup1"
      },
      "IpProtocol": "tcp",
      "ToPort": "80",
      "FromPort": "80",
      "SourceSecurityGroupName": {
        "Ref": "SGroup2"
      }
    }
  },
  "SGroup2Ingress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupName": {
        "Ref": "SGroup2"
      },
      "IpProtocol": "tcp",
      "ToPort": "80",
      "FromPort": "80",
      "SourceSecurityGroupName": {
        "Ref": "SGroup1"
      }
    }
  }
}

```

YAML

```

SGroup1:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: EC2 Instance access
SGroup2:
  Type: AWS::EC2::SecurityGroup
  Properties:

```



```

    GroupDescription: EC2 Instance access
  SGroup1Ingress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupName: !Ref SGroup1
      IpProtocol: tcp
      ToPort: 80
      FromPort: 80
      SourceSecurityGroupName: !Ref SGroup2
  SGroup2Ingress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupName: !Ref SGroup2
      IpProtocol: tcp
      ToPort: 80
      FromPort: 80
      SourceSecurityGroupName: !Ref SGroup1

```

使用安全群組傳入規則建立 Elastic Load Balancer

下列範本會在指定的可用區域建立 [AWS::ElasticLoadBalancing::LoadBalancer](#) 資源。[AWS::ElasticLoadBalancing::LoadBalancer](#) 資源設定為在連接埠 80 上接聽 HTTP 流量，請求也會引導至連接埠 80 上的執行個體。Elastic Load Balancer 負責對執行個體之間的傳入 HTTP 流量進行負載平衡。

此外，此範本會產生與負載平衡器關聯的 [AWS::EC2::SecurityGroup](#) 資源。此安全群組使用單一傳入規則建立，描述為 ELB ingress group，這會允許連接埠 80 上的傳入 TCP 流量。此傳入規則的來源 Fn::GetAtt 函數定義，以從負載平衡器資源擷取屬性。SourceSecurityGroupOwnerId 使用 Fn::GetAtt 來取得負載平衡器來源安全群組的 OwnerAlias。SourceSecurityGroupName 使用 Fn::GetAtt 來取得 ELB 來源安全群組的 GroupName。

此設定可確保 ELB 與執行個體之間進行安全通訊。

如需負載平衡的詳細資訊，請參閱 [Elastic Load Balancing 使用者指南](#)。

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyELB": {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {

```

```

        "AvailabilityZones": [
            "aa-example-1a"
        ],
        "Listeners": [
            {
                "LoadBalancerPort": "80",
                "InstancePort": "80",
                "Protocol": "HTTP"
            }
        ]
    },
    "MyELBIngressGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "ELB ingress group",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "SourceSecurityGroupOwnerId": {
                        "Fn::GetAtt": [
                            "MyELB",
                            "SourceSecurityGroup.OwnerAlias"
                        ]
                    },
                    "SourceSecurityGroupName": {
                        "Fn::GetAtt": [
                            "MyELB",
                            "SourceSecurityGroup.GroupName"
                        ]
                    }
                }
            ]
        }
    }
}

```

YAML

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

```
Resources:
  MyELB:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
      AvailabilityZones:
        - aa-example-1a
      Listeners:
        - LoadBalancerPort: '80'
          InstancePort: '80'
          Protocol: HTTP
  MyELBIngressGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: ELB ingress group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '80'
          ToPort: '80'
          SourceSecurityGroupOwnerId:
            Fn::GetAtt:
              - MyELB
              - SourceSecurityGroup.OwnerAlias
      SourceSecurityGroupName:
        Fn::GetAtt:
          - MyELB
          - SourceSecurityGroup.GroupName
```

配置彈性 IP 位址並與 CloudFormation 產生關聯

下列範本程式碼片段是與 Amazon EC2 中的彈性 IP 地址 (EIP) 相關的範例。這些範例涵蓋您的執行個體的 EIP 配置、關聯和管理。

範例程式碼片段

- [配置彈性 IP 地址並與 Amazon EC2 執行個體建立關聯](#)
- [透過指定 IP 地址將彈性 IP 地址與 Amazon EC2 執行個體建立關聯](#)
- [透過指定 IP 地址的配置 ID 將彈性 IP 地址與 Amazon EC2 執行個體建立關聯](#)

配置彈性 IP 地址並與 Amazon EC2 執行個體建立關聯

下列程式碼片段配置 Amazon EC2 彈性 IP 地址，並使用 [AWS::EC2::EIP](#) 資源將其與 Amazon EC2 執行個體建立關聯。您可以從擁有的地址集區配置 EIP 地址，AWS 或從公有 IPv4 地址範圍建立的地址

集區配置 EIP 地址，AWS 以便使用[自有 IP 地址 \(BYOIP\)](#) 搭配 AWS 資源使用。在此範例中，EIP 是從擁有的地址集區配置 AWS。

如需有關彈性 IP 地址的詳細資訊，請參閱《Amazon EC2 使用者指南》中的[彈性 IP 地址](#)。

JSON

```
"ElasticIP": {
  "Type": "AWS::EC2::EIP",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    }
  }
}
```

YAML

```
ElasticIP:
  Type: AWS::EC2::EIP
  Properties:
    InstanceId: !Ref EC2Instance
```

透過指定 IP 地址將彈性 IP 地址與 Amazon EC2 執行個體建立關聯

下列程式碼片段使用 [AWS::EC2::EIPAssociation](#) 資源，將現有的 Amazon EC2 彈性 IP 地址與 EC2 執行個體建立關聯。您必須先配置彈性 IP 地址，才能在帳戶中使用。彈性 IP 地址可與單一執行個體建立關聯。

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "EIP": "192.0.2.0"
  }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    EIP: 192.0.2.0
```

透過指定 IP 地址的配置 ID 將彈性 IP 地址與 Amazon EC2 執行個體建立關聯

下列程式碼片段使用 [AWS::EC2::EIPAssociation](#) 資源，將現有的彈性 IP 地址與 Amazon EC2 執行個體建立關聯。在配置彈性 IP 地址時，配置 ID 會指派給彈性 IP 地址。

JSON

```
"IPAssoc": {
  "Type": "AWS::EC2::EIPAssociation",
  "Properties": {
    "InstanceId": {
      "Ref": "Ec2Instance"
    },
    "AllocationId": "eipalloc-1234567890abcdef0"
  }
}
```

YAML

```
IPAssoc:
  Type: AWS::EC2::EIPAssociation
  Properties:
    InstanceId: !Ref EC2Instance
    AllocationId: eipalloc-1234567890abcdef0
```

使用 CloudFormation 來設定 Amazon VPC 資源

本節提供使用 CloudFormation 設定 Amazon VPC 資源的範例。VPC 可讓您在 內建立虛擬網路 AWS，而這些程式碼片段會示範如何設定 VPCs 的各個層面，以符合您的聯網需求。

範例程式碼片段

- [在 VPC 中啟用僅限 IPv6 傳入網際網路存取](#)
- [彈性網路介面 \(ENI\) 範本程式碼片段](#)

在 VPC 中啟用僅限 IPv6 傳入網際網路存取

僅限傳入網際網路閘道允許 VPC 內的執行個體存取網際網路，並阻止網際網路上的資源與執行個體通訊。下列程式碼片段會從 VPC 內啟用僅限 IPv6 傳入網際網路存取。它會使用 [AWS::EC2::VPC](#) 資源建立具有 10.0.0/16 IPv4 地址範圍的 VPC。使用 [AWS::EC2::RouteTable](#) 資源，將路由表與此 VPC 資源建立關聯。路由表會管理 VPC 內執行個體的路由。[AWS::EC2::EgressOnlyInternetGateway](#) 用於建立僅限傳入網際網路閘道，以針對 VPC 內執行個體的傳出流量啟用 IPv6 通訊，同時阻止傳入流量。指定 [AWS::EC2::Route](#) 資源在路由表中建立 IPv6 路由，這會將所有傳出 IPv6 流量 (:::/0) 引導至僅限傳入網際網路閘道。

如需僅限傳出網際網路閘道的詳細資訊，請參閱 Amazon VPC 使用者指南中[使用僅限傳出網際網路閘道啟用傳出 IPv6 流量](#)。

JSON

```
"DefaultIpv6Route": {
  "Type": "AWS::EC2::Route",
  "Properties": {
    "DestinationIpv6CidrBlock": ":::/0",
    "EgressOnlyInternetGatewayId": {
      "Ref": "EgressOnlyInternetGateway"
    },
    "RouteTableId": {
      "Ref": "RouteTable"
    }
  }
},
"EgressOnlyInternetGateway": {
  "Type": "AWS::EC2::EgressOnlyInternetGateway",
  "Properties": {
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
"RouteTable": {
  "Type": "AWS::EC2::RouteTable",
  "Properties": {
    "VpcId": {
      "Ref": "VPC"
    }
  }
},
```

```
"VPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "CidrBlock": "10.0.0.0/16"
  }
}
```

YAML

```
DefaultIpv6Route:
  Type: AWS::EC2::Route
  Properties:
    DestinationIpv6CidrBlock: "::/0"
    EgressOnlyInternetGatewayId:
      Ref: "EgressOnlyInternetGateway"
    RouteTableId:
      Ref: "RouteTable"
EgressOnlyInternetGateway:
  Type: AWS::EC2::EgressOnlyInternetGateway
  Properties:
    VpcId:
      Ref: "VPC"
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: "VPC"
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: "10.0.0.0/16"
```

彈性網路介面 (ENI) 範本程式碼片段

使用附接的彈性網路介面 (ENI) 建立 Amazon EC2 執行個體

下列範例程式碼片段會使用 Amazon VPC 和子網路中指定的 [AWS::EC2::Instance](#) 資源建立 Amazon EC2 執行個體。它會將兩個網路介面 (ENI) 附接至執行個體，透過附接的 ENI 將彈性 IP 地址與執行個體建立關聯，以及設定安全群組進行 SSH 和 HTTP 存取。建立執行個體時，會將使用者資料做為啟動組態的一部分提供給執行個體。使用者資料包括以 base64 格式編碼的指令碼，以確保將其傳遞至執行個體。啟動執行個體後，指令碼會在引導程序中自動執行。它會安裝 ec2-net-utils、設定網路介面，以及啟動 HTTP 服務。

若要根據選取的區域確定適當的 Amazon Machine Image (AMI)，程式碼片段會使用可在 RegionMap 映射中查詢值的 `Fn::FindInMap` 函數。此映射必須在較大的範本中定義。使用 [AWS::EC2::NetworkInterface](#) 資源建立兩個網路介面。使用配置給 vpc 網域的 [AWS::EC2::EIP](#) 資源指定彈性 IP 位址。這些彈性 IP 位址與使用 [AWS::EC2::EIPAssociation](#) 資源的網路介面相關聯。

Outputs 區段定義您想要在建立堆疊之後存取的值或資源。在此程式碼片段中，定義的輸出為 InstancePublicIp，代表堆疊建立的 EC2 執行個體的公有 IP 地址。您可以在 CloudFormation 主控台的輸出索引標籤中擷取此輸出，或使用 [describe-stacks](#) 命令。

如需彈性網路介面的詳細資訊，請參閱[彈性網路介面](#)。

JSON

```
"Resources": {
  "ControlPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
      "Domain": "vpc"
    }
  },
  "AssociateControlPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
      "AllocationId": {
        "Fn::GetAtt": [
          "ControlPortAddress",
          "AllocationId"
        ]
      },
      "NetworkInterfaceId": {
        "Ref": "controlXface"
      }
    }
  },
  "WebPortAddress": {
    "Type": "AWS::EC2::EIP",
    "Properties": {
      "Domain": "vpc"
    }
  },
  "AssociateWebPort": {
    "Type": "AWS::EC2::EIPAssociation",
    "Properties": {
```



```

        "AllocationId": {
            "Fn::GetAtt": [
                "WebPortAddress",
                "AllocationId"
            ]
        },
        "NetworkInterfaceId": {
            "Ref": "webXface"
        }
    }
},
"SSHSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "VpcId": {
            "Ref": "VpcId"
        },
        "GroupDescription": "Enable SSH access via port 22",
        "SecurityGroupIngress": [
            {
                "CidrIp": "0.0.0.0/0",
                "FromPort": 22,
                "IpProtocol": "tcp",
                "ToPort": 22
            }
        ]
    }
},
"WebSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "VpcId": {
            "Ref": "VpcId"
        },
        "GroupDescription": "Enable HTTP access via user-defined port",
        "SecurityGroupIngress": [
            {
                "CidrIp": "0.0.0.0/0",
                "FromPort": 80,
                "IpProtocol": "tcp",
                "ToPort": 80
            }
        ]
    }
}

```

```
    },
    "controlXface": {
      "Type": "AWS::EC2::NetworkInterface",
      "Properties": {
        "SubnetId": {
          "Ref": "SubnetId"
        },
        "Description": "Interface for controlling traffic such as SSH",
        "GroupSet": [
          {
            "Fn::GetAtt": [
              "SSHSecurityGroup",
              "GroupId"
            ]
          }
        ],
        "SourceDestCheck": true,
        "Tags": [
          {
            "Key": "Network",
            "Value": "Control"
          }
        ]
      }
    },
    "webXface": {
      "Type": "AWS::EC2::NetworkInterface",
      "Properties": {
        "SubnetId": {
          "Ref": "SubnetId"
        },
        "Description": "Interface for web traffic",
        "GroupSet": [
          {
            "Fn::GetAtt": [
              "WebSecurityGroup",
              "GroupId"
            ]
          }
        ],
        "SourceDestCheck": true,
        "Tags": [
          {
            "Key": "Network",
```

```

        "Value": "Web"
      }
    ]
  },
  "Ec2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": {
        "Fn::FindInMap": [
          "RegionMap",
          {
            "Ref": "AWS::Region"
          },
          "AMI"
        ]
      },
      "KeyName": {
        "Ref": "KeyName"
      },
      "NetworkInterfaces": [
        {
          "NetworkInterfaceId": {
            "Ref": "controlXface"
          },
          "DeviceIndex": "0"
        },
        {
          "NetworkInterfaceId": {
            "Ref": "webXface"
          },
          "DeviceIndex": "1"
        }
      ],
      "Tags": [
        {
          "Key": "Role",
          "Value": "Test Instance"
        }
      ],
      "UserData": {
        "Fn::Base64": {
          "Fn::Sub": "#!/bin/bash -xe\nyum install ec2-net-utils -y\nec2ifup eth1\nservice httpd start\n"
        }
      }
    }
  }
}

```

```

    }
  }
}
},
"Outputs": {
  "InstancePublicIp": {
    "Description": "Public IP Address of the EC2 Instance",
    "Value": {
      "Fn::GetAtt": [
        "Ec2Instance",
        "PublicIp"
      ]
    }
  }
}
}

```

YAML

```

Resources:
  ControlPortAddress:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc
  AssociateControlPort:
    Type: AWS::EC2::EIPAssociation
    Properties:
      AllocationId:
        Fn::GetAtt:
          - ControlPortAddress
          - AllocationId
      NetworkInterfaceId:
        Ref: controlXface
  WebPortAddress:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc
  AssociateWebPort:
    Type: AWS::EC2::EIPAssociation
    Properties:
      AllocationId:
        Fn::GetAtt:
          - WebPortAddress

```

```
- AllocationId
NetworkInterfaceId:
  Ref: webXface
SSHSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 22
        IpProtocol: tcp
        ToPort: 22
WebSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VpcId
    GroupDescription: Enable HTTP access via user-defined port
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 80
        IpProtocol: tcp
        ToPort: 80
controlXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
    Description: Interface for controlling traffic such as SSH
    GroupSet:
      - Fn::GetAtt:
          - SSHSecurityGroup
          - GroupId
    SourceDestCheck: true
    Tags:
      - Key: Network
        Value: Control
webXface:
  Type: AWS::EC2::NetworkInterface
  Properties:
    SubnetId:
      Ref: SubnetId
```

```
Description: Interface for web traffic
GroupSet:
  - Fn::GetAtt:
    - WebSecurityGroup
    - GroupId
SourceDestCheck: true
Tags:
  - Key: Network
    Value: Web
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Fn::FindInMap:
        - RegionMap
        - Ref: AWS::Region
        - AMI
    KeyName:
      Ref: KeyName
    NetworkInterfaces:
      - NetworkInterfaceId:
          Ref: controlXface
          DeviceIndex: "0"
      - NetworkInterfaceId:
          Ref: webXface
          DeviceIndex: "1"
    Tags:
      - Key: Role
        Value: Test Instance
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash -xe
        yum install ec2-net-utils -y
        ec2ifup eth1
        service httpd start
Outputs:
  InstancePublicIp:
    Description: Public IP Address of the EC2 Instance
    Value:
      Fn::GetAtt:
        - Ec2Instance
        - PublicIp
```

Amazon Elastic Container Service 範例範本

Amazon Elastic Container Service (Amazon ECS) 是一種容器管理服務，可以在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體叢集上輕鬆執行、停止及管理 Docker 容器。

建立具有 AL2023 Amazon ECS-Optimized-AMI 的叢集

定義使用在 Amazon EC2 上啟動 AL2023 執行個體的容量提供者的叢集。

Important

如需取得最新的 AMI ID，請參閱《Amazon Elastic Container Service 開發人員指南》中的 [Amazon ECS 最佳化 AMI](#)。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.",
  "Parameters": {
    "InstanceType": {
      "Type": "String",
      "Description": "EC2 instance type",
      "Default": "t2.medium",
      "AllowedValues": [
        "t1.micro",
        "t2.2xlarge",
        "t2.large",
        "t2.medium",
        "t2.micro",
        "t2.nano",
        "t2.small",
        "t2.xlarge",
        "t3.2xlarge",
        "t3.large",
        "t3.medium",
        "t3.micro",
        "t3.nano",
        "t3.small",
```

```

        "t3.xlarge"
    ]
},
"DesiredCapacity": {
    "Type": "Number",
    "Default": "0",
    "Description": "Number of EC2 instances to launch in your ECS cluster."
},
"MaxSize": {
    "Type": "Number",
    "Default": "100",
    "Description": "Maximum number of EC2 instances that can be launched in your
ECS cluster."
},
"ECSAMI": {
    "Description": "The Amazon Machine Image ID used for the cluster",
    "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
    "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/
image_id"
},
"VpcId": {
    "Type": "AWS::EC2::VPC::Id",
    "Description": "VPC ID where the ECS cluster is launched",
    "Default": "vpc-1234567890abcdef0"
},
"SubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of subnet IDs where the EC2 instances will be launched",
    "Default": "subnet-021345abcdef67890"
}
},
"Resources": {
    "ECSCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {
            "ClusterSettings": [
                {
                    "Name": "containerInsights",
                    "Value": "enabled"
                }
            ]
        }
    }
},
"ECSAutoScalingGroup": {

```



```
"Type": "AWS::AutoScaling::AutoScalingGroup",
"DependsOn": [
    "ECSCluster",
    "EC2Role"
],
"Properties": {
    "VPCZoneIdentifier": {
        "Ref": "SubnetIds"
    },
    "LaunchTemplate": {
        "LaunchTemplateId": {
            "Ref": "ContainerInstances"
        },
        "Version": {
            "Fn::GetAtt": [
                "ContainerInstances",
                "LatestVersionNumber"
            ]
        }
    },
    "MinSize": 0,
    "MaxSize": {
        "Ref": "MaxSize"
    },
    "DesiredCapacity": {
        "Ref": "DesiredCapacity"
    },
    "NewInstancesProtectedFromScaleIn": true
},
"UpdatePolicy": {
    "AutoScalingReplacingUpdate": {
        "WillReplace": "true"
    }
},
"ContainerInstances": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
        "LaunchTemplateName": "asg-launch-template",
        "LaunchTemplateData": {
            "ImageId": {
                "Ref": "ECSAMI"
            },
            "InstanceType": {
```

```

        "Ref": "InstanceType"
    },
    "IamInstanceProfile": {
        "Name": {
            "Ref": "EC2InstanceProfile"
        }
    },
    "SecurityGroupIds": [
        {
            "Ref": "ContainerHostSecurityGroup"
        }
    ],
    "UserData": {
        "Fn::Base64": {
            "Fn::Sub": "#!/bin/bash -xe\n echo ECS_CLUSTER=${ECSCluster}
>> /etc/ecs/ecs.config\n yum install -y aws-cfn-bootstrap\n /opt/aws/bin/cfn-init -
v --stack ${AWS::StackId} --resource ContainerInstances --configsets full_install --
region ${AWS::Region} &\n"
        }
    },
    "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpTokens": "required"
    }
}
},
"EC2InstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [
            {
                "Ref": "EC2Role"
            }
        ]
    }
},
"CapacityProvider": {
    "Type": "AWS::ECS::CapacityProvider",
    "Properties": {
        "AutoScalingGroupProvider": {
            "AutoScalingGroupArn": {
                "Ref": "ECSAutoScalingGroup"
            }
        }
    }
}

```

```

        },
        "ManagedScaling": {
            "InstanceWarmupPeriod": 60,
            "MinimumScalingStepSize": 1,
            "MaximumScalingStepSize": 100,
            "Status": "ENABLED",
            "TargetCapacity": 100
        },
        "ManagedTerminationProtection": "ENABLED"
    }
},
"CapacityProviderAssociation": {
    "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
    "Properties": {
        "CapacityProviders": [
            {
                "Ref": "CapacityProvider"
            }
        ],
        "Cluster": {
            "Ref": "ECSCluster"
        },
        "DefaultCapacityProviderStrategy": [
            {
                "Base": 0,
                "CapacityProvider": {
                    "Ref": "CapacityProvider"
                },
                "Weight": 1
            }
        ]
    }
},
"ContainerHostSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the EC2 hosts that run containers",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"EC2Role": {

```

```

    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "ec2.amazonaws.com"
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ]
          }
        ]
      },
      "Path": "/",
      "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
      ]
    }
  },
  "ECSTaskExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": [
                "ecs-tasks.amazonaws.com"
              ]
            },
            "Action": [
              "sts:AssumeRole"
            ],
            "Condition": {
              "ArnLike": {
                "aws:SourceArn": {

```

```

        "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
    },
    "StringEquals": {
        "aws:SourceAccount": {
            "Fn::Sub": "${AWS::AccountId}"
        }
    }
},
"Path": "/",
"ManagedPolicyArns": [
    "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
]
}
},
"Outputs": {
    "ClusterName": {
        "Description": "The ECS cluster into which to launch resources",
        "Value": "ECSCluster"
    },
    "ECSTaskExecutionRole": {
        "Description": "The role used to start up a task",
        "Value": "ECSTaskExecutionRole"
    },
    "CapacityProvider": {
        "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task",
        "Value": "CapacityProvider"
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet.

```

An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.

Parameters:

InstanceType:

Type: String

Description: EC2 instance type

Default: "t2.medium"

AllowedValues:

- t1.micro
- t2.2xlarge
- t2.large
- t2.medium
- t2.micro
- t2.nano
- t2.small
- t2.xlarge
- t3.2xlarge
- t3.large
- t3.medium
- t3.micro
- t3.nano
- t3.small
- t3.xlarge

DesiredCapacity:

Type: Number

Default: "0"

Description: Number of EC2 instances to launch in your ECS cluster.

MaxSize:

Type: Number

Default: "100"

Description: Maximum number of EC2 instances that can be launched in your ECS cluster.

ECSAMI:

Description: The Amazon Machine Image ID used for the cluster

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id

VpcId:

Type: AWS::EC2::VPC::Id

Description: VPC ID where the ECS cluster is launched

Default: vpc-1234567890abcdef0

SubnetIds:

Type: List<AWS::EC2::Subnet::Id>

Description: List of subnet IDs where the EC2 instances will be launched

Default: "subnet-021345abcdef67890"

```
Resources:
# This is authorizes ECS to manage resources on your
# account on your behalf. This role is likely already created on your account
# ECSRole:
#   Type: AWS::IAM::ServiceLinkedRole
#   Properties:
#     AWSServiceName: 'ecs.amazonaws.com'

# ECS Resources
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterSettings:
      - Name: containerInsights
        Value: enabled

# Autoscaling group. This launches the actual EC2 instances that will register
# themselves as members of the cluster, and run the docker containers.
ECSAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    # This is to ensure that the ASG gets deleted first before these
    # resources, when it comes to stack teardown.
    - ECSCluster
    - EC2Role
  Properties:
    VPCZoneIdentifier:
      Ref: SubnetIds
    LaunchTemplate:
      LaunchTemplateId: !Ref ContainerInstances
      Version: !GetAtt ContainerInstances.LatestVersionNumber
    MinSize: 0
    MaxSize:
      Ref: MaxSize
    DesiredCapacity:
      Ref: DesiredCapacity
    NewInstancesProtectedFromScaleIn: true
    UpdatePolicy:
      AutoScalingReplacingUpdate:
        WillReplace: "true"
# The config for each instance that is added to the cluster
ContainerInstances:
  Type: AWS::EC2::LaunchTemplate
  Properties:
```

```

LaunchTemplateName: "asg-launch-template"
LaunchTemplateData:
  ImageId:
    Ref: ECSAMI
  InstanceType:
    Ref: InstanceType
  IamInstanceProfile:
    Name: !Ref EC2InstanceProfile
  SecurityGroupIds:
    - !Ref ContainerHostSecurityGroup
  # This injected configuration file is how the EC2 instance
  # knows which ECS cluster on your AWS account it should be joining
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash -xe
      echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
      yum install -y aws-cfn-bootstrap
      /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
ContainerInstances --configsets full_install --region ${AWS::Region} &
      # Disable IMDSv1, and require IMDSv2
  MetadataOptions:
    HttpEndpoint: enabled
    HttpTokens: required
EC2InstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref EC2Role
  # Create an ECS capacity provider to attach the ASG to the ECS cluster
  # so that it autoscales as we launch more containers
CapacityProvider:
  Type: AWS::ECS::CapacityProvider
  Properties:
    AutoScalingGroupProvider:
      AutoScalingGroupArn: !Ref ECSAutoScalingGroup
    ManagedScaling:
      InstanceWarmupPeriod: 60
      MinimumScalingStepSize: 1
      MaximumScalingStepSize: 100
      Status: ENABLED
      # Percentage of cluster reservation to try to maintain
      TargetCapacity: 100
    ManagedTerminationProtection: ENABLED

```



```

# Create a cluster capacity provider association so that the cluster
# will use the capacity provider
CapacityProviderAssociation:
  Type: AWS::ECS::ClusterCapacityProviderAssociations
  Properties:
    CapacityProviders:
      - !Ref CapacityProvider
    Cluster: !Ref ECSCluster
    DefaultCapacityProviderStrategy:
      - Base: 0
        CapacityProvider: !Ref CapacityProvider
        Weight: 1
# A security group for the EC2 hosts that will run the containers.
# This can be used to limit incoming traffic to or outgoing traffic
# from the container's host EC2 instance.
ContainerHostSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Access to the EC2 hosts that run containers
    VpcId:
      Ref: VpcId
# Role for the EC2 hosts. This allows the ECS agent on the EC2 hosts
# to communicate with the ECS control plane, as well as download the docker
# images from ECR to run on your host.
EC2Role:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: /
    ManagedPolicyArns:
      # See reference: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/
security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      # This managed policy allows us to connect to the instance using SSM
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
# This is a role which is used within Fargate to allow the Fargate agent
# to download images, and upload logs.

```

```

ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ecs-tasks.amazonaws.com
          Action:
            - sts:AssumeRole
          Condition:
            ArnLike:
              aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
            StringEquals:
              aws:SourceAccount: !Sub ${AWS::AccountId}
      Path: /
      # This role enables all features of ECS. See reference:
      # https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-iam-awsmanpol.html#security-iam-awsmanpol-AmazonECSTaskExecutionRolePolicy
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
Outputs:
  ClusterName:
    Description: The ECS cluster into which to launch resources
    Value: ECSCluster
  ECSTaskExecutionRole:
    Description: The role used to start up a task
    Value: ECSTaskExecutionRole
  CapacityProvider:
    Description: The cluster capacity provider that the service should use to
      request capacity when it wants to start up a task
    Value: CapacityProvider

```

部署服務

下列範本定義的服務使用容量提供者來請求 AL2023 容量以供執行。容器會在 AL2023 執行個體上線時啟動到這些執行個體上：

JSON

```
{
```

```
"AWSTemplateFormatVersion": "2010-09-09",
"Description": "An example service that deploys in AWS VPC networking mode on EC2
capacity. Service uses a capacity provider to request EC2 instances to run on. Service
runs with networking in private subnets, but still accessible to the internet via a
load balancer hosted in public subnets.",
"Parameters": {
  "VpcId": {
    "Type": "String",
    "Description": "The VPC that the service is running inside of"
  },
  "PublicSubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of public subnet ID's to put the load balancer in"
  },
  "PrivateSubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of private subnet ID's that the AWS VPC tasks are in"
  },
  "ClusterName": {
    "Type": "String",
    "Description": "The name of the ECS cluster into which to launch capacity."
  },
  "ECSTaskExecutionRole": {
    "Type": "String",
    "Description": "The role used to start up an ECS task"
  },
  "CapacityProvider": {
    "Type": "String",
    "Description": "The cluster capacity provider that the service should use to
request capacity when it wants to start up a task"
  },
  "ServiceName": {
    "Type": "String",
    "Default": "web",
    "Description": "A name for the service"
  },
  "ImageUrl": {
    "Type": "String",
    "Default": "public.ecr.aws/docker/library/nginx:latest",
    "Description": "The url of a docker image that contains the application
process that will handle the traffic for this service"
  },
  "ContainerCpu": {
    "Type": "Number",
```

```

        "Default": 256,
        "Description": "How much CPU to give the container. 1024 is 1 CPU"
    },
    "ContainerMemory": {
        "Type": "Number",
        "Default": 512,
        "Description": "How much memory in megabytes to give the container"
    },
    "ContainerPort": {
        "Type": "Number",
        "Default": 80,
        "Description": "What port that the application expects traffic on"
    },
    "DesiredCount": {
        "Type": "Number",
        "Default": 2,
        "Description": "How many copies of the service task to run"
    }
},
"Resources": {
    "TaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "Family": {
                "Ref": "ServiceName"
            },
            "Cpu": {
                "Ref": "ContainerCpu"
            },
            "Memory": {
                "Ref": "ContainerMemory"
            },
            "NetworkMode": "awsvpc",
            "RequiresCompatibilities": [
                "EC2"
            ],
            "ExecutionRoleArn": {
                "Ref": "ECSTaskExecutionRole"
            },
            "ContainerDefinitions": [
                {
                    "Name": {
                        "Ref": "ServiceName"
                    },

```

```

        "Cpu": {
            "Ref": "ContainerCpu"
        },
        "Memory": {
            "Ref": "ContainerMemory"
        },
        "Image": {
            "Ref": "ImageUrl"
        },
        "PortMappings": [
            {
                "ContainerPort": {
                    "Ref": "ContainerPort"
                },
                "HostPort": {
                    "Ref": "ContainerPort"
                }
            }
        ],
        "LogConfiguration": {
            "LogDriver": "awslogs",
            "Options": {
                "mode": "non-blocking",
                "max-buffer-size": "25m",
                "awslogs-group": {
                    "Ref": "LogGroup"
                },
                "awslogs-region": {
                    "Ref": "AWS::Region"
                },
                "awslogs-stream-prefix": {
                    "Ref": "ServiceName"
                }
            }
        }
    }
}

},
"Service": {
    "Type": "AWS::ECS::Service",
    "DependsOn": "PublicLoadBalancerListener",
    "Properties": {
        "ServiceName": {

```

```
        "Ref": "ServiceName"
    },
    "Cluster": {
        "Ref": "ClusterName"
    },
    "PlacementStrategies": [
        {
            "Field": "attribute:ecs.availability-zone",
            "Type": "spread"
        },
        {
            "Field": "cpu",
            "Type": "binpack"
        }
    ],
    "CapacityProviderStrategy": [
        {
            "Base": 0,
            "CapacityProvider": {
                "Ref": "CapacityProvider"
            },
            "Weight": 1
        }
    ],
    "NetworkConfiguration": {
        "AwsvpcConfiguration": {
            "SecurityGroups": [
                {
                    "Ref": "ServiceSecurityGroup"
                }
            ],
            "Subnets": {
                "Ref": "PrivateSubnetIds"
            }
        }
    },
    "DeploymentConfiguration": {
        "MaximumPercent": 200,
        "MinimumHealthyPercent": 75
    },
    "DesiredCount": {
        "Ref": "DesiredCount"
    },
    "TaskDefinition": {
```

```

        "Ref": "TaskDefinition"
    },
    "LoadBalancers": [
        {
            "ContainerName": {
                "Ref": "ServiceName"
            },
            "ContainerPort": {
                "Ref": "ContainerPort"
            },
            "TargetGroupArn": {
                "Ref": "ServiceTargetGroup"
            }
        }
    ]
},
"ServiceSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Security group for service",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"ServiceTargetGroup": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
        "HealthCheckIntervalSeconds": 6,
        "HealthCheckPath": "/",
        "HealthCheckProtocol": "HTTP",
        "HealthCheckTimeoutSeconds": 5,
        "HealthyThresholdCount": 2,
        "TargetType": "ip",
        "Port": {
            "Ref": "ContainerPort"
        },
        "Protocol": "HTTP",
        "UnhealthyThresholdCount": 10,
        "VpcId": {
            "Ref": "VpcId"
        },
        "TargetGroupAttributes": [

```

```

        {
            "Key": "deregistration_delay.timeout_seconds",
            "Value": 0
        }
    ]
}
},
"PublicLoadBalancerSG": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the public facing load balancer",
        "VpcId": {
            "Ref": "VpcId"
        },
        "SecurityGroupIngress": [
            {
                "CidrIp": "0.0.0.0/0",
                "IpProtocol": -1
            }
        ]
    }
},
"PublicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
        "Scheme": "internet-facing",
        "LoadBalancerAttributes": [
            {
                "Key": "idle_timeout.timeout_seconds",
                "Value": "30"
            }
        ],
        "Subnets": {
            "Ref": "PublicSubnetIds"
        },
        "SecurityGroups": [
            {
                "Ref": "PublicLoadBalancerSG"
            }
        ]
    }
},
"PublicLoadBalancerListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",

```



```

    "Properties": {
      "DefaultActions": [
        {
          "Type": "forward",
          "ForwardConfig": {
            "TargetGroups": [
              {
                "TargetGroupArn": {
                  "Ref": "ServiceTargetGroup"
                },
                "Weight": 100
              }
            ]
          }
        }
      ],
      "LoadBalancerArn": {
        "Ref": "PublicLoadBalancer"
      },
      "Port": 80,
      "Protocol": "HTTP"
    }
  },
  "ServiceIngressfromLoadBalancer": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "Description": "Ingress from the public ALB",
      "GroupId": {
        "Ref": "ServiceSecurityGroup"
      },
      "IpProtocol": -1,
      "SourceSecurityGroupId": {
        "Ref": "PublicLoadBalancerSG"
      }
    }
  },
  "LogGroup": {
    "Type": "AWS::Logs::LogGroup"
  }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: >-
  An example service that deploys in AWS VPC networking mode on EC2 capacity.
  Service uses a capacity provider to request EC2 instances to run on. Service
  runs with networking in private subnets, but still accessible to the internet
  via a load balancer hosted in public subnets.
Parameters:
  VpcId:
    Type: String
    Description: The VPC that the service is running inside of
  PublicSubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: List of public subnet ID's to put the load balancer in
  PrivateSubnetIds:
    Type: 'List<AWS::EC2::Subnet::Id>'
    Description: List of private subnet ID's that the AWS VPC tasks are in
  ClusterName:
    Type: String
    Description: The name of the ECS cluster into which to launch capacity.
  ECSTaskExecutionRole:
    Type: String
    Description: The role used to start up an ECS task
  CapacityProvider:
    Type: String
    Description: >-
      The cluster capacity provider that the service should use to request
      capacity when it wants to start up a task
  ServiceName:
    Type: String
    Default: web
    Description: A name for the service
  ImageUrl:
    Type: String
    Default: 'public.ecr.aws/docker/library/nginx:latest'
    Description: >-
      The url of a docker image that contains the application process that will
      handle the traffic for this service
  ContainerCpu:
    Type: Number
    Default: 256
    Description: How much CPU to give the container. 1024 is 1 CPU
  ContainerMemory:
```

```
Type: Number
Default: 512
Description: How much memory in megabytes to give the container
ContainerPort:
  Type: Number
  Default: 80
  Description: What port that the application expects traffic on
DesiredCount:
  Type: Number
  Default: 2
  Description: How many copies of the service task to run
Resources:
  TaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      Family: !Ref ServiceName
      Cpu: !Ref ContainerCpu
      Memory: !Ref ContainerMemory
      NetworkMode: awsvpc
      RequiresCompatibilities:
        - EC2
      ExecutionRoleArn: !Ref ECSTaskExecutionRole
      ContainerDefinitions:
        - Name: !Ref ServiceName
          Cpu: !Ref ContainerCpu
          Memory: !Ref ContainerMemory
          Image: !Ref ImageUrl
          PortMappings:
            - ContainerPort: !Ref ContainerPort
              HostPort: !Ref ContainerPort
          LogConfiguration:
            LogDriver: awslogs
            Options:
              mode: non-blocking
              max-buffer-size: 25m
              awslogs-group: !Ref LogGroup
              awslogs-region: !Ref AWS::Region
              awslogs-stream-prefix: !Ref ServiceName
  Service:
    Type: AWS::ECS::Service
    DependsOn: PublicLoadBalancerListener
    Properties:
      ServiceName: !Ref ServiceName
      Cluster: !Ref ClusterName
```

```
PlacementStrategies:
  - Field: 'attribute:ecs.availability-zone'
    Type: spread
  - Field: cpu
    Type: binpack
CapacityProviderStrategy:
  - Base: 0
    CapacityProvider: !Ref CapacityProvider
    Weight: 1
NetworkConfiguration:
  AwsVpcConfiguration:
    SecurityGroups:
      - !Ref ServiceSecurityGroup
    Subnets: !Ref PrivateSubnetIds
DeploymentConfiguration:
  MaximumPercent: 200
  MinimumHealthyPercent: 75
DesiredCount: !Ref DesiredCount
TaskDefinition: !Ref TaskDefinition
LoadBalancers:
  - ContainerName: !Ref ServiceName
    ContainerPort: !Ref ContainerPort
    TargetGroupArn: !Ref ServiceTargetGroup
ServiceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for service
    VpcId: !Ref VpcId
ServiceTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    HealthCheckIntervalSeconds: 6
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 2
    TargetType: ip
    Port: !Ref ContainerPort
    Protocol: HTTP
    UnhealthyThresholdCount: 10
    VpcId: !Ref VpcId
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: 0
```

```
PublicLoadBalancerSG:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Access to the public facing load balancer
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        IpProtocol: -1
PublicLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internet-facing
    LoadBalancerAttributes:
      - Key: idle_timeout.timeout_seconds
        Value: '30'
    Subnets: !Ref PublicSubnetIds
    SecurityGroups:
      - !Ref PublicLoadBalancerSG
PublicLoadBalancerListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref ServiceTargetGroup
              Weight: 100
    LoadBalancerArn: !Ref PublicLoadBalancer
    Port: 80
    Protocol: HTTP
ServiceIngressfromLoadBalancer:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    Description: Ingress from the public ALB
    GroupId: !Ref ServiceSecurityGroup
    IpProtocol: -1
    SourceSecurityGroupId: !Ref PublicLoadBalancerSG
LogGroup:
  Type: AWS::Logs::LogGroup
```

Amazon Elastic File System 範例範本

Amazon Elastic File System (Amazon EFS) 是適用於 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體的檔案儲存服務。透過 Amazon EFS，系統會隨著您新增和移除的檔案自動擴展及縮減儲存容量，讓應用程式可以隨時享有所需的儲存空間。

以下範例範本會部署與 Amazon EFS 檔案系統相關聯的 EC2 執行個體 (在 Auto Scaling 群組中)。為了將執行個體與檔案系統建立關聯，執行個體會執行 `cfn-init` 協助程式指令碼，進而下載並安裝 `nfs-utils` `yum` 套件、建立新目錄，接著透過檔案系統的 DNS 名稱在該目錄掛載檔案系統。而系統會在 Amazon EC2 執行個體可用區域中，將該檔案系統的 DNS 名稱解析為掛載目標的 IP 地址。如需有關 DNS 名稱結構的詳細資訊，請參閱 Amazon Elastic File System User Guide (《Amazon Elastic File System 使用者指南》) 中的 [Mounting File Systems](#) (掛載檔案系統)。

範本會加入自訂 Amazon CloudWatch 指標，藉此測量網路檔案系統的活動；範本亦會建立 VPC、子網路與安全群組。VPC 必須啟用 DNS，且掛載目標和 EC2 執行個體需位於子網路所指定的相同可用區域 (AZ)，才能讓執行個體與檔案系統互相通訊。

掛載目標的安全群組會啟用 TCP 連接埠 2049 的網路連線，此為 NFSv4 用戶端掛載檔案系統時的必要動作。如需有關 EC2 執行個體和掛載目標之安全群組的詳細資訊，請參閱 [Amazon Elastic File System User Guide](#) (《Amazon Elastic File System 使用者指南》) 中的 [Security](#) (安全性)。

Note

若更新掛載目標導致該目標遭取代，則使用相關檔案系統的執行個體或應用程式可能會中斷，這會造成未遞交的撰寫內容遺失。為了避免中斷，請在更新掛載目標時將所需容量設為零，進而停用執行個體。這項操作能讓執行個體先卸載檔案系統，隨後再刪除掛載目標。掛載更新完成後，請設定所需容量，即可在後續更新作業中啟動執行個體。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. **WARNING** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.",
  "Parameters": {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
```

```
"Type" : "String",
"Default" : "t2.small",
"AllowedValues" : [
  "t1.micro",
  "t2.nano",
  "t2.micro",
  "t2.small",
  "t2.medium",
  "t2.large",
  "m1.small",
  "m1.medium",
  "m1.large",
  "m1.xlarge",
  "m2.xlarge",
  "m2.2xlarge",
  "m2.4xlarge",
  "m3.medium",
  "m3.large",
  "m3.xlarge",
  "m3.2xlarge",
  "m4.large",
  "m4.xlarge",
  "m4.2xlarge",
  "m4.4xlarge",
  "m4.10xlarge",
  "c1.medium",
  "c1.xlarge",
  "c3.large",
  "c3.xlarge",
  "c3.2xlarge",
  "c3.4xlarge",
  "c3.8xlarge",
  "c4.large",
  "c4.xlarge",
  "c4.2xlarge",
  "c4.4xlarge",
  "c4.8xlarge",
  "g2.2xlarge",
  "g2.8xlarge",
  "r3.large",
  "r3.xlarge",
  "r3.2xlarge",
  "r3.4xlarge",
  "r3.8xlarge",
```

```

        "i2.xlarge",
        "i2.2xlarge",
        "i2.4xlarge",
        "i2.8xlarge",
        "d2.xlarge",
        "d2.2xlarge",
        "d2.4xlarge",
        "d2.8xlarge",
        "hi1.4xlarge",
        "hs1.8xlarge",
        "cr1.8xlarge",
        "cc2.8xlarge",
        "cg1.4xlarge"
    ],
    "ConstraintDescription" : "must be a valid EC2 instance type."
},
"KeyName": {
    "Type": "AWS::EC2::KeyPair::KeyName",
    "Description": "Name of an existing EC2 key pair to enable SSH access to the EC2
instances"
},
"AsgMaxSize": {
    "Type": "Number",
    "Description": "Maximum size and initial desired capacity of Auto Scaling Group",
    "Default": "2"
},
"SSHLocation" : {
    "Description" : "The IP address range that can be used to connect to the EC2
instances by using SSH",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
},
"VolumeName" : {
    "Description" : "The name to be used for the EFS volume",
    "Type": "String",
    "MinLength": "1",
    "Default": "myEFSvolume"
},
"MountPoint" : {
    "Description" : "The Linux mount point for the EFS volume",

```



```

    "Type": "String",
    "MinLength": "1",
    "Default": "myEFSvolume"
  }
},
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"      : { "Arch" : "HVM64" },
    "t2.nano"      : { "Arch" : "HVM64" },
    "t2.micro"      : { "Arch" : "HVM64" },
    "t2.small"     : { "Arch" : "HVM64" },
    "t2.medium"    : { "Arch" : "HVM64" },
    "t2.large"     : { "Arch" : "HVM64" },
    "m1.small"     : { "Arch" : "HVM64" },
    "m1.medium"    : { "Arch" : "HVM64" },
    "m1.large"     : { "Arch" : "HVM64" },
    "m1.xlarge"    : { "Arch" : "HVM64" },
    "m2.xlarge"    : { "Arch" : "HVM64" },
    "m2.2xlarge"   : { "Arch" : "HVM64" },
    "m2.4xlarge"   : { "Arch" : "HVM64" },
    "m3.medium"    : { "Arch" : "HVM64" },
    "m3.large"     : { "Arch" : "HVM64" },
    "m3.xlarge"    : { "Arch" : "HVM64" },
    "m3.2xlarge"   : { "Arch" : "HVM64" },
    "m4.large"     : { "Arch" : "HVM64" },
    "m4.xlarge"    : { "Arch" : "HVM64" },
    "m4.2xlarge"   : { "Arch" : "HVM64" },
    "m4.4xlarge"   : { "Arch" : "HVM64" },
    "m4.10xlarge"  : { "Arch" : "HVM64" },
    "c1.medium"    : { "Arch" : "HVM64" },
    "c1.xlarge"    : { "Arch" : "HVM64" },
    "c3.large"     : { "Arch" : "HVM64" },
    "c3.xlarge"    : { "Arch" : "HVM64" },
    "c3.2xlarge"   : { "Arch" : "HVM64" },
    "c3.4xlarge"   : { "Arch" : "HVM64" },
    "c3.8xlarge"   : { "Arch" : "HVM64" },
    "c4.large"     : { "Arch" : "HVM64" },
    "c4.xlarge"    : { "Arch" : "HVM64" },
    "c4.2xlarge"   : { "Arch" : "HVM64" },
    "c4.4xlarge"   : { "Arch" : "HVM64" },
    "c4.8xlarge"   : { "Arch" : "HVM64" },
    "g2.2xlarge"   : { "Arch" : "HVMG2" },
    "g2.8xlarge"   : { "Arch" : "HVMG2" },
    "r3.large"     : { "Arch" : "HVM64" },

```

```

    "r3.xlarge" : { "Arch" : "HVM64" },
    "r3.2xlarge" : { "Arch" : "HVM64" },
    "r3.4xlarge" : { "Arch" : "HVM64" },
    "r3.8xlarge" : { "Arch" : "HVM64" },
    "i2.xlarge" : { "Arch" : "HVM64" },
    "i2.2xlarge" : { "Arch" : "HVM64" },
    "i2.4xlarge" : { "Arch" : "HVM64" },
    "i2.8xlarge" : { "Arch" : "HVM64" },
    "d2.xlarge" : { "Arch" : "HVM64" },
    "d2.2xlarge" : { "Arch" : "HVM64" },
    "d2.4xlarge" : { "Arch" : "HVM64" },
    "d2.8xlarge" : { "Arch" : "HVM64" },
    "hi1.4xlarge" : { "Arch" : "HVM64" },
    "hs1.8xlarge" : { "Arch" : "HVM64" },
    "cr1.8xlarge" : { "Arch" : "HVM64" },
    "cc2.8xlarge" : { "Arch" : "HVM64" }
  },
  "AWSRegionArch2AMI" : {
    "us-east-1" : { "HVM64" : "ami-0ff8a91507f77f867", "HVMG2" :
"ami-0a584ac55a7631c0c"},
    "us-west-2" : { "HVM64" : "ami-a0cfeed8", "HVMG2" :
"ami-0e09505bc235aa82d"},
    "us-west-1" : { "HVM64" : "ami-0bdb828fd58c52235", "HVMG2" :
"ami-066ee5fd4a9ef77f1"},
    "eu-west-1" : { "HVM64" : "ami-047bb4163c506cd98", "HVMG2" :
"ami-0a7c483d527806435"},
    "eu-west-2" : { "HVM64" : "ami-f976839e", "HVMG2" : "NOT_SUPPORTED"},
    "eu-west-3" : { "HVM64" : "ami-0ebc281c20e89ba4b", "HVMG2" :
"NOT_SUPPORTED"},
    "eu-central-1" : { "HVM64" : "ami-0233214e13e500f77", "HVMG2" :
"ami-06223d46a6d0661c7"},
    "ap-northeast-1" : { "HVM64" : "ami-06cd52961ce9f0d85", "HVMG2" :
"ami-053cdd503598e4a9d"},
    "ap-northeast-2" : { "HVM64" : "ami-0a10b2721688ce9d2", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-northeast-3" : { "HVM64" : "ami-0d98120a9fb693f07", "HVMG2" :
"NOT_SUPPORTED"},
    "ap-southeast-1" : { "HVM64" : "ami-08569b978cc4dfa10", "HVMG2" :
"ami-0be9df32ae9f92309"},
    "ap-southeast-2" : { "HVM64" : "ami-09b42976632b27e9b", "HVMG2" :
"ami-0a9ce9fecc3d1daf8"},
    "ap-south-1" : { "HVM64" : "ami-0912f71e06545ad88", "HVMG2" :
"ami-097b15e89dbdcfcf4"},

```

```

        "us-east-2"           : {"HVM64" : "ami-0b59bfac6be064b78", "HVMG2" :
"NOT_SUPPORTED"},
        "ca-central-1"       : {"HVM64" : "ami-0b18956f", "HVMG2" : "NOT_SUPPORTED"},
        "sa-east-1"          : {"HVM64" : "ami-07b14488da8ea02a0", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-north-1"         : {"HVM64" : "ami-0a4eaf6c4454eda75", "HVMG2" :
"NOT_SUPPORTED"},
        "cn-northwest-1"     : {"HVM64" : "ami-6b6a7d09", "HVMG2" : "NOT_SUPPORTED"}
    }
},
"Resources": {
    "CloudWatchPutMetricsRole" : {
        "Type" : "AWS::IAM::Role",
        "Properties" : {
            "AssumeRolePolicyDocument" : {
                "Statement" : [ {
                    "Effect" : "Allow",
                    "Principal" : {
                        "Service" : [ "ec2.amazonaws.com" ]
                    },
                    "Action" : [ "sts:AssumeRole" ]
                } ]
            },
            "Path" : "/"
        }
    },
    "CloudWatchPutMetricsRolePolicy" : {
        "Type" : "AWS::IAM::Policy",
        "Properties" : {
            "PolicyName" : "CloudWatch_PutMetricData",
            "PolicyDocument" : {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "CloudWatchPutMetricData",
                        "Effect": "Allow",
                        "Action": ["cloudwatch:PutMetricData"],
                        "Resource": ["*"]
                    }
                ]
            },
            "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
        }
    }
},

```

```
"CloudWatchPutMetricsInstanceProfile" : {
  "Type" : "AWS::IAM::InstanceProfile",
  "Properties" : {
    "Path" : "/",
    "Roles" : [ { "Ref" : "CloudWatchPutMetricsRole" } ]
  }
},
"VPC": {
  "Type": "AWS::EC2::VPC",
  "Properties": {
    "EnableDnsSupport" : "true",
    "EnableDnsHostnames" : "true",
    "CidrBlock": "10.0.0.0/16",
    "Tags": [ {"Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
  }
},
"InternetGateway" : {
  "Type" : "AWS::EC2::InternetGateway",
  "Properties" : {
    "Tags" : [
      { "Key" : "Application", "Value" : { "Ref" : "AWS::StackName" } },
      { "Key" : "Network", "Value" : "Public" }
    ]
  }
},
"GatewayToInternet" : {
  "Type" : "AWS::EC2::VPCGatewayAttachment",
  "Properties" : {
    "VpcId" : { "Ref" : "VPC" },
    "InternetGatewayId" : { "Ref" : "InternetGateway" }
  }
},
"RouteTable":{
  "Type":"AWS::EC2::RouteTable",
  "Properties":{
    "VpcId": {"Ref":"VPC"}
  }
},
"SubnetRouteTableAssoc": {
  "Type" : "AWS::EC2::SubnetRouteTableAssociation",
  "Properties" : {
    "RouteTableId" : {"Ref":"RouteTable"},
    "SubnetId" : {"Ref":"Subnet"}
  }
}
```

```

},
"InternetGatewayRoute": {
  "Type": "AWS::EC2::Route",
  "Properties": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "RouteTableId": { "Ref": "RouteTable" },
    "GatewayId": { "Ref": "InternetGateway" }
  }
},
"Subnet": {
  "Type": "AWS::EC2::Subnet",
  "Properties": {
    "VpcId": { "Ref": "VPC" },
    "CidrBlock": "10.0.0.0/24",
    "Tags": [ { "Key": "Application", "Value": { "Ref": "AWS::StackId" } } ]
  }
},
"InstanceSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "VpcId": { "Ref": "VPC" },
    "GroupDescription": "Enable SSH access via port 22",
    "SecurityGroupIngress": [
      { "IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "CidrIp": { "Ref":
"SSHLocation" } },
      { "IpProtocol": "tcp", "FromPort": 80, "ToPort": 80, "CidrIp": "0.0.0.0/0" }
    ]
  }
},
"MountTargetSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "VpcId": { "Ref": "VPC" },
    "GroupDescription": "Security group for mount target",
    "SecurityGroupIngress": [
      {
        "IpProtocol": "tcp",
        "FromPort": 2049,
        "ToPort": 2049,
        "CidrIp": "0.0.0.0/0"
      }
    ]
  }
},
},

```

```

"FileSystem": {
  "Type": "AWS::EFS::FileSystem",
  "Properties": {
    "PerformanceMode": "generalPurpose",
    "FileSystemTags": [
      {
        "Key": "Name",
        "Value": { "Ref" : "VolumeName" }
      }
    ]
  }
},
"MountTarget": {
  "Type": "AWS::EFS::MountTarget",
  "Properties": {
    "FileSystemId": { "Ref": "FileSystem" },
    "SubnetId": { "Ref": "Subnet" },
    "SecurityGroups": [ { "Ref": "MountTargetSecurityGroup" } ]
  }
},
"LaunchConfiguration": {
  "Type": "AWS::AutoScaling::LaunchConfiguration",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "configSets" : {
        "MountConfig" : [ "setup", "mount" ]
      },
      "setup" : {
        "packages" : {
          "yum" : {
            "nfs-utils" : []
          }
        },
        "files" : {
          "/home/ec2-user/post_nfsstat" : {
            "content" : { "Fn::Join" : [ "", [
              "#!/bin/bash\n",
              "\n",
              "INPUT=\"$(cat)\"\n",
              "CW_JSON_OPEN='{ \"Namespace\": \"EFS\", \"MetricData\": [ '\n",
              "CW_JSON_CLOSE=' ] }'\n",
              "CW_JSON_METRIC=' '\n",
              "METRIC_COUNTER=0\n",
              "\n",

```

```

        "for COL in 1 2 3 4 5 6; do\n",
        "\n",
        "  COUNTER=0\n",
        "  METRIC_FIELD=$COL\n",
        "  DATA_FIELD=$(( $COL + ( $COL - 1 ) ) )\n",
        "\n",
        "  while read line; do\n",
        "    if [[ COUNTER -gt 0 ]]; then\n",
        "\n",
        "      LINE=`echo $line | tr -s ' ' `'\n",
        "      AWS_COMMAND=\"aws cloudwatch put-metric-data --region ",
{ "Ref": "AWS::Region" }, "\n",
        "      MOD=$(( $COUNTER % 2 ))\n",
        "\n",
        "      if [ $MOD -eq 1 ]; then\n",
        "        METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`\n",
\n",
        "        else\n",
        "          METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`\n",
        "          fi\n",
        "\n",
        "          if [[ -n \"$METRIC_NAME\" && -n \"$METRIC_VALUE\" ]]; then\n",
\n",
        "            INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-
data/instance-id)\n",
        "            CW_JSON_METRIC=\"${CW_JSON_METRIC} { \\\"MetricName\\\": \\\"
\\\"$METRIC_NAME\\\", \\\"Dimensions\\\": [{\\\"Name\\\": \\\"InstanceId\\\", \\\"Value\\\"
\\\": \\\"$INSTANCE_ID\\\"} ], \\\"Value\\\": $METRIC_VALUE },\n",
        "            unset METRIC_NAME\n",
        "            unset METRIC_VALUE\n",
        "\n",
        "            METRIC_COUNTER=$((METRIC_COUNTER+1))\n",
        "            if [ $METRIC_COUNTER -eq 20 ]; then\n",
        "              # 20 is max metric collection size, so we have to
submit here\n",
        "              aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\n",
        "\n",
        "              # reset\n",
        "              METRIC_COUNTER=0\n",
        "              CW_JSON_METRIC='' '\n",
        "              fi\n",
        "            fi\n",

```

```

        "\n",
        "\n",
        "\n",
        "    COUNTER=$((COUNTER+1))\n",
        "    fi\n",
        "\n",
        "    if [[ \"$line\" == \"Client nfs v4:\" ]]; then\n",
        "        # the next line is the good stuff\n",
        "        COUNTER=$((COUNTER+1))\n",
        "        fi\n",
        "    done <<< \"$INPUT\"\n",
        "done\n",
        "\n",
        "# submit whatever is left\n",
        "aws cloudwatch put-metric-data --region ", { "Ref":
"AWS::Region" }, " --cli-input-json \"`echo $CW_JSON_OPEN ${CW_JSON_METRIC%?}
$CW_JSON_CLOSE`\"
    ] ] },
    "mode": "000755",
    "owner": "ec2-user",
    "group": "ec2-user"
},
"/home/ec2-user/crontab" : {
    "content" : { "Fn::Join" : [ "", [
        "** * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
    ] ] },
    "owner": "ec2-user",
    "group": "ec2-user"
}
},
"commands" : {
    "01_createdir" : {
        "command" : { "Fn::Join" : [ "", [ "mkdir /", { "Ref" :
"MountPoint" } ] ] }
    }
},
"mount" : {
    "commands" : {
        "01_mount" : {
            "command" : { "Fn::Sub": "sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2 ${FileSystem}.efs.
${AWS::Region}.amazonaws.com:/ /${MountPoint}"
        }
    }
},

```



```

        "02_permissions" : {
            "command" : { "Fn::Join" : [ "", [ "chown ec2-user:ec2-user /",
{ "Ref" : "MountPoint" }]]] }
        }
    },
    "Properties": {
        "AssociatePublicIpAddress" : true,
        "ImageId": {
            "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" }, {
                "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" },
"Arch" ]
            } ]
        },
        "InstanceType": { "Ref": "InstanceType" },
        "KeyName": { "Ref": "KeyName" },
        "SecurityGroups": [ { "Ref": "InstanceSecurityGroup" } ],
        "IamInstanceProfile" : { "Ref" : "CloudWatchPutMetricsInstanceProfile" },
        "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
            "#!/bin/bash -xe\n",
            "yum install -y aws-cfn-bootstrap\n",

            "/opt/aws/bin/cfn-init -v ",
            "    --stack ", { "Ref" : "AWS::StackName" },
            "    --resource LaunchConfiguration ",
            "    --configsets MountConfig ",
            "    --region ", { "Ref" : "AWS::Region" }, "\n",

            "crontab /home/ec2-user/crontab\n",

            "/opt/aws/bin/cfn-signal -e $? ",
            "    --stack ", { "Ref" : "AWS::StackName" },
            "    --resource AutoScalingGroup ",
            "    --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]] }
    },
    "AutoScalingGroup": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "DependsOn": ["MountTarget", "GatewayToInternet"],
        "CreationPolicy" : {
            "ResourceSignal" : {

```

```

        "Timeout" : "PT15M",
        "Count"   : { "Ref": "AsgMaxSize" }
    }
},
"Properties": {
    "VPCZoneIdentifier": [ { "Ref": "Subnet" } ],
    "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
    "MinSize": "1",
    "MaxSize": { "Ref": "AsgMaxSize" },
    "DesiredCapacity": { "Ref": "AsgMaxSize" },
    "Tags": [ {
        "Key": "Name",
        "Value": "EFS FileSystem Mounted Instance",
        "PropagateAtLaunch": "true"
    } ]
}
}
},
"Outputs" : {
    "MountTargetID" : {
        "Description" : "Mount target ID",
        "Value" : { "Ref" : "MountTarget" }
    },
    "FileSystemID" : {
        "Description" : "File system ID",
        "Value" : { "Ref" : "FileSystem" }
    }
}
}
}

```

YAML

AWSTemplateFormatVersion: '2010-09-09'

Description: This template creates an Amazon EFS file system and mount target and associates it with Amazon EC2 instances in an Auto Scaling group. ****WARNING**** This template creates Amazon EC2 instances and related resources. You will be billed for the AWS resources used if you create a stack from this template.

Parameters:

InstanceType:

Description: WebServer EC2 instance type

Type: String

Default: t2.small

AllowedValues:

- t1.micro
- t2.nano
- t2.micro
- t2.small
- t2.medium
- t2.large
- m1.small
- m1.medium
- m1.large
- m1.xlarge
- m2.xlarge
- m2.2xlarge
- m2.4xlarge
- m3.medium
- m3.large
- m3.xlarge
- m3.2xlarge
- m4.large
- m4.xlarge
- m4.2xlarge
- m4.4xlarge
- m4.10xlarge
- c1.medium
- c1.xlarge
- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- c3.8xlarge
- c4.large
- c4.xlarge
- c4.2xlarge
- c4.4xlarge
- c4.8xlarge
- g2.2xlarge
- g2.8xlarge
- r3.large
- r3.xlarge
- r3.2xlarge
- r3.4xlarge
- r3.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge

- i2.8xlarge
- d2.xlarge
- d2.2xlarge
- d2.4xlarge
- d2.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- cr1.8xlarge
- cc2.8xlarge
- cg1.4xlarge

ConstraintDescription: must be a valid EC2 instance type.

KeyName:

Type: `AWS::EC2::KeyPair::KeyName`

Description: Name of an existing EC2 key pair to enable SSH access to the ECS instances

AsgMaxSize:

Type: Number

Description: Maximum size and initial desired capacity of Auto Scaling Group

Default: '2'

SSHLocation:

Description: The IP address range that can be used to connect to the EC2 instances by using SSH

Type: String

MinLength: '9'

MaxLength: '18'

Default: 0.0.0.0/0

AllowedPattern: "(\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})\\. (\\d{1,3})/(\\d{1,2})"

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.

VolumeName:

Description: The name to be used for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

MountPoint:

Description: The Linux mount point for the EFS volume

Type: String

MinLength: '1'

Default: myEFSvolume

Mappings:

AWSInstanceType2Arch:

```
t1.micro:
```

Arch: HVM64

t2.nano:

Arch: HVM64

```
t2.micro:
  Arch: HVM64
t2.small:
  Arch: HVM64
t2.medium:
  Arch: HVM64
t2.large:
  Arch: HVM64
m1.small:
  Arch: HVM64
m1.medium:
  Arch: HVM64
m1.large:
  Arch: HVM64
m1.xlarge:
  Arch: HVM64
m2.xlarge:
  Arch: HVM64
m2.2xlarge:
  Arch: HVM64
m2.4xlarge:
  Arch: HVM64
m3.medium:
  Arch: HVM64
m3.large:
  Arch: HVM64
m3.xlarge:
  Arch: HVM64
m3.2xlarge:
  Arch: HVM64
m4.large:
  Arch: HVM64
m4.xlarge:
  Arch: HVM64
m4.2xlarge:
  Arch: HVM64
m4.4xlarge:
  Arch: HVM64
m4.10xlarge:
  Arch: HVM64
c1.medium:
  Arch: HVM64
c1.xlarge:
  Arch: HVM64
```

```
c3.large:
  Arch: HVM64
c3.xlarge:
  Arch: HVM64
c3.2xlarge:
  Arch: HVM64
c3.4xlarge:
  Arch: HVM64
c3.8xlarge:
  Arch: HVM64
c4.large:
  Arch: HVM64
c4.xlarge:
  Arch: HVM64
c4.2xlarge:
  Arch: HVM64
c4.4xlarge:
  Arch: HVM64
c4.8xlarge:
  Arch: HVM64
g2.2xlarge:
  Arch: HVMG2
g2.8xlarge:
  Arch: HVMG2
r3.large:
  Arch: HVM64
r3.xlarge:
  Arch: HVM64
r3.2xlarge:
  Arch: HVM64
r3.4xlarge:
  Arch: HVM64
r3.8xlarge:
  Arch: HVM64
i2.xlarge:
  Arch: HVM64
i2.2xlarge:
  Arch: HVM64
i2.4xlarge:
  Arch: HVM64
i2.8xlarge:
  Arch: HVM64
d2.xlarge:
  Arch: HVM64
```

```
d2.2xlarge:
  Arch: HVM64
d2.4xlarge:
  Arch: HVM64
d2.8xlarge:
  Arch: HVM64
hi1.4xlarge:
  Arch: HVM64
hs1.8xlarge:
  Arch: HVM64
cr1.8xlarge:
  Arch: HVM64
cc2.8xlarge:
  Arch: HVM64
AWSRegionArch2AMI:
us-east-1:
  HVM64: ami-0ff8a91507f77f867
  HVMG2: ami-0a584ac55a7631c0c
us-west-2:
  HVM64: ami-a0cfeed8
  HVMG2: ami-0e09505bc235aa82d
us-west-1:
  HVM64: ami-0bdb828fd58c52235
  HVMG2: ami-066ee5fd4a9ef77f1
eu-west-1:
  HVM64: ami-047bb4163c506cd98
  HVMG2: ami-0a7c483d527806435
eu-west-2:
  HVM64: ami-f976839e
  HVMG2: NOT_SUPPORTED
eu-west-3:
  HVM64: ami-0ebc281c20e89ba4b
  HVMG2: NOT_SUPPORTED
eu-central-1:
  HVM64: ami-0233214e13e500f77
  HVMG2: ami-06223d46a6d0661c7
ap-northeast-1:
  HVM64: ami-06cd52961ce9f0d85
  HVMG2: ami-053cdd503598e4a9d
ap-northeast-2:
  HVM64: ami-0a10b2721688ce9d2
  HVMG2: NOT_SUPPORTED
ap-northeast-3:
  HVM64: ami-0d98120a9fb693f07
```

```
HVMG2: NOT_SUPPORTED
ap-southeast-1:
  HVM64: ami-08569b978cc4dfa10
  HVMG2: ami-0be9df32ae9f92309
ap-southeast-2:
  HVM64: ami-09b42976632b27e9b
  HVMG2: ami-0a9ce9fecc3d1daf8
ap-south-1:
  HVM64: ami-0912f71e06545ad88
  HVMG2: ami-097b15e89dbdcfcf4
us-east-2:
  HVM64: ami-0b59bfac6be064b78
  HVMG2: NOT_SUPPORTED
ca-central-1:
  HVM64: ami-0b18956f
  HVMG2: NOT_SUPPORTED
sa-east-1:
  HVM64: ami-07b14488da8ea02a0
  HVMG2: NOT_SUPPORTED
cn-north-1:
  HVM64: ami-0a4eaf6c4454eda75
  HVMG2: NOT_SUPPORTED
cn-northwest-1:
  HVM64: ami-6b6a7d09
  HVMG2: NOT_SUPPORTED
```

Resources:

CloudWatchPutMetricsRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Statement:

- Effect: Allow

Principal:

Service:

- ec2.amazonaws.com

Action:

- sts:AssumeRole

Path: "/"

CloudWatchPutMetricsRolePolicy:

Type: AWS::IAM::Policy

Properties:

PolicyName: CloudWatch_PutMetricData

PolicyDocument:

Version: '2012-10-17'


```
    Statement:
      - Sid: CloudWatchPutMetricData
        Effect: Allow
        Action:
          - cloudwatch:PutMetricData
        Resource:
          - "*"

    Roles:
      - Ref: CloudWatchPutMetricsRole
CloudWatchPutMetricsInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: "/"
    Roles:
      - Ref: CloudWatchPutMetricsRole
VPC:
  Type: AWS::EC2::VPC
  Properties:
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
    CidrBlock: 10.0.0.0/16
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackName
      - Key: Network
        Value: Public
GatewayToInternet:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId:
      Ref: VPC
    InternetGatewayId:
      Ref: InternetGateway
RouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
```

```
VpcId:
  Ref: VPC
SubnetRouteTableAssoc:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId:
      Ref: RouteTable
    SubnetId:
      Ref: Subnet
InternetGatewayRoute:
  Type: AWS::EC2::Route
  Properties:
    DestinationCidrBlock: 0.0.0.0/0
    RouteTableId:
      Ref: RouteTable
    GatewayId:
      Ref: InternetGateway
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    Tags:
      - Key: Application
        Value:
          Ref: AWS::StackId
InstanceSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription: Enable SSH access via port 22
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp:
          Ref: SSHLocation
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
MountTargetSecurityGroup:
```

```
Type: AWS::EC2::SecurityGroup
Properties:
  VpcId:
    Ref: VPC
  GroupDescription: Security group for mount target
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 2049
      ToPort: 2049
      CidrIp: 0.0.0.0/0
FileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    PerformanceMode: generalPurpose
    FileSystemTags:
      - Key: Name
        Value:
          Ref: VolumeName
MountTarget:
  Type: AWS::EFS::MountTarget
  Properties:
    FileSystemId:
      Ref: FileSystem
    SubnetId:
      Ref: Subnet
    SecurityGroups:
      - Ref: MountTargetSecurityGroup
LaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Metadata:
    AWS::CloudFormation::Init:
      configSets:
        MountConfig:
          - setup
          - mount
      setup:
        packages:
          yum:
            nfs-utils: []
        files:
          "/home/ec2-user/post_nfsstat":
            content: !Sub |
              #!/bin/bash
```

```

INPUT="$(cat)"
CW_JSON_OPEN='{ "Namespace": "EFS", "MetricData": [ '
CW_JSON_CLOSE=' ] }'
CW_JSON_METRIC=''
METRIC_COUNTER=0

for COL in 1 2 3 4 5 6; do

    COUNTER=0
    METRIC_FIELD=$COL
    DATA_FIELD=$(( $COL + ( $COL - 1 )) )

    while read line; do
        if [[ COUNTER -gt 0 ]]; then

            LINE=`echo $line | tr -s ' ' `
            AWS_COMMAND="aws cloudwatch put-metric-data --region
${AWS::Region}"
            MOD=$(( $COUNTER % 2 ))

            if [ $MOD -eq 1 ]; then
                METRIC_NAME=`echo $LINE | cut -d ' ' -f $METRIC_FIELD`
            else
                METRIC_VALUE=`echo $LINE | cut -d ' ' -f $DATA_FIELD`
            fi

            if [[ -n "$METRIC_NAME" && -n "$METRIC_VALUE" ]]; then
                INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/
instance-id)
                CW_JSON_METRIC="$CW_JSON_METRIC { \"MetricName\": \"$METRIC_NAME
\", \"Dimensions\": [{\"Name\": \"InstanceId\", \"Value\": \"$INSTANCE_ID\"} ], \"Value
\": $METRIC_VALUE },"
                unset METRIC_NAME
                unset METRIC_VALUE

                METRIC_COUNTER=$((METRIC_COUNTER+1))
                if [ $METRIC_COUNTER -eq 20 ]; then
                    # 20 is max metric collection size, so we have to submit here
                    aws cloudwatch put-metric-data --region ${AWS::Region} --cli-
input-json "`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"

                    # reset
                    METRIC_COUNTER=0
                    CW_JSON_METRIC=''

```

```

        fi
    fi

    COUNTER=$((COUNTER+1))
fi

if [[ "$line" == "Client nfs v4:" ]]; then
    # the next line is the good stuff
    COUNTER=$((COUNTER+1))
fi
done <<< "$INPUT"
done

# submit whatever is left
aws cloudwatch put-metric-data --region ${AWS::Region} --cli-input-json
"$`echo $CW_JSON_OPEN ${!CW_JSON_METRIC%?} $CW_JSON_CLOSE`"
mode: '000755'
owner: ec2-user
group: ec2-user
"/home/ec2-user/crontab":
content: "* * * * * /usr/sbin/nfsstat | /home/ec2-user/post_nfsstat\n"
owner: ec2-user
group: ec2-user
commands:
  01_createdir:
    command: !Sub "mkdir /${MountPoint}"
mount:
  commands:
    01_mount:
      command: !Sub >
        mount -t nfs4 -o nfsvers=4.1 ${FileSystem}.efs.
        ${AWS::Region}.amazonaws.com:/ /${MountPoint}
    02_permissions:
      command: !Sub "chown ec2-user:ec2-user /${MountPoint}"
Properties:
  AssociatePublicIpAddress: true
  ImageId:
    Fn::FindInMap:
      - AWSRegionArch2AMI
      - Ref: AWS::Region
      - Fn::FindInMap:
          - AWSInstanceType2Arch

```

```

    - Ref: InstanceType
    - Arch
  InstanceType:
    Ref: InstanceType
  KeyName:
    Ref: KeyName
  SecurityGroups:
    - Ref: InstanceSecurityGroup
  IamInstanceProfile:
    Ref: CloudWatchPutMetricsInstanceProfile
  UserData:
    Fn::Base64: !Sub |
      #!/bin/bash -xe
      yum install -y aws-cfn-bootstrap
      /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --resource
LaunchConfiguration --configsets MountConfig --region ${AWS::Region}
      crontab /home/ec2-user/crontab
      /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
AutoScalingGroup --region ${AWS::Region}
  AutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    DependsOn:
      - MountTarget
      - GatewayToInternet
    CreationPolicy:
      ResourceSignal:
        Timeout: PT15M
        Count:
          Ref: AsgMaxSize
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    LaunchConfigurationName:
      Ref: LaunchConfiguration
    MinSize: '1'
    MaxSize:
      Ref: AsgMaxSize
    DesiredCapacity:
      Ref: AsgMaxSize
    Tags:
      - Key: Name
        Value: EFS FileSystem Mounted Instance
        PropagateAtLaunch: 'true'
  Outputs:

```

```
MountTargetID:
  Description: Mount target ID
  Value:
    Ref: MountTarget
FileSystemID:
  Description: File system ID
  Value:
    Ref: FileSystem
```

Elastic Beanstalk 範本程式碼片段

使用 Elastic Beanstalk，您可以在 中快速部署和管理應用程式，AWS 而無需擔心執行這些應用程式的基礎設施。下列範例範本可協助您在 CloudFormation 範本中描述 Elastic Beanstalk 資源。

Elastic Beanstalk 範例 PHP

下列範例範本會部署存放在 Amazon S3 儲存貯體的範例 PHP Web 應用程式。此環境也是自動擴展的負載平衡環境，至少有兩個 Amazon EC2 執行個體，最多則有六個。此範例展示的 Elastic Beanstalk 環境使用了舊版啟動組態。有關改用啟動範本的資訊，請參閱《AWS Elastic Beanstalk 開發人員指南》中的[啟動範本](#)。

以解決方案堆疊名稱 (平台版本) 取代 *solution-stack*。如需可用解決方案堆疊的清單，請使用 AWS CLI 命令 `aws elasticbeanstalk list-available-solution-stacks`。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "sampleApplication": {
      "Type": "AWS::ElasticBeanstalk::Application",
      "Properties": {
        "Description": "AWS Elastic Beanstalk Sample Application"
      }
    },
    "sampleApplicationVersion": {
      "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
      "Properties": {
        "ApplicationName": {
          "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Application Version",
        "SourceBundle": {
```

```

        "S3Bucket": {
            "Fn::Sub": "elasticbeanstalk-samples-${AWS::Region}"
        },
        "S3Key": "php-newsample-app.zip"
    }
},
"sampleConfigurationTemplate": {
    "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
    "Properties": {
        "ApplicationName": {
            "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Configuration Template",
        "OptionSettings": [
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MinSize",
                "Value": "2"
            },
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MaxSize",
                "Value": "6"
            },
            {
                "Namespace": "aws:elasticbeanstalk:environment",
                "OptionName": "EnvironmentType",
                "Value": "LoadBalanced"
            },
            {
                "Namespace": "aws:autoscaling:launchconfiguration",
                "OptionName": "IamInstanceProfile",
                "Value": {
                    "Ref": "MyInstanceProfile"
                }
            }
        ],
        "SolutionStackName": "solution-stack"
    }
},
"sampleEnvironment": {
    "Type": "AWS::ElasticBeanstalk::Environment",
    "Properties": {

```



```

        "ApplicationName": {
            "Ref": "sampleApplication"
        },
        "Description": "AWS ElasticBeanstalk Sample Environment",
        "TemplateName": {
            "Ref": "sampleConfigurationTemplate"
        },
        "VersionLabel": {
            "Ref": "sampleApplicationVersion"
        }
    },
    "MyInstanceRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ec2.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ]
                    }
                ]
            },
            "Description": "Beanstalk EC2 role",
            "ManagedPolicyArns": [
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier",
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker",
                "arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier"
            ]
        }
    },
    "MyInstanceProfile": {
        "Type": "AWS::IAM::InstanceProfile",
        "Properties": {
            "Roles": [
                {

```

```

        "Ref": "MyInstanceRole"
      }
    ]
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  sampleApplication:
    Type: AWS::ElasticBeanstalk::Application
    Properties:
      Description: AWS Elastic Beanstalk Sample Application
  sampleApplicationVersion:
    Type: AWS::ElasticBeanstalk::ApplicationVersion
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Application Version
      SourceBundle:
        S3Bucket: !Sub "elasticbeanstalk-samples-${AWS::Region}"
        S3Key: php-newsample-app.zip
  sampleConfigurationTemplate:
    Type: AWS::ElasticBeanstalk::ConfigurationTemplate
    Properties:
      ApplicationName:
        Ref: sampleApplication
      Description: AWS ElasticBeanstalk Sample Configuration Template
      OptionSettings:
        - Namespace: aws:autoscaling:asg
          OptionName: MinSize
          Value: '2'
        - Namespace: aws:autoscaling:asg
          OptionName: MaxSize
          Value: '6'
        - Namespace: aws:elasticbeanstalk:environment
          OptionName: EnvironmentType
          Value: LoadBalanced
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile

```

```

    Value: !Ref MyInstanceProfile
    SolutionStackName: solution-stack
sampleEnvironment:
  Type: AWS::ElasticBeanstalk::Environment
  Properties:
    ApplicationName:
      Ref: sampleApplication
    Description: AWS ElasticBeanstalk Sample Environment
    TemplateName:
      Ref: sampleConfigurationTemplate
    VersionLabel:
      Ref: sampleApplicationVersion
MyInstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Description: Beanstalk EC2 role
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWebTier
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkMulticontainerDocker
      - arn:aws:iam::aws:policy/AWSElasticBeanstalkWorkerTier
MyInstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - !Ref MyInstanceRole

```

Elastic Load Balancing 範本程式碼片段

要建立 Application Load Balancer、Network Load Balancer 或 Gateway Load Balancer，請使用開頭為 `AWS::ElasticLoadBalancingV2` 的 V2 資源類型。要建立 Classic Load Balancer，請使用開頭為 `AWS::ElasticLoadBalancing` 的資源類型。

目錄

- [ELBv2 資源](#)
- [Classic Load Balancer 資源](#)

ELBv2 資源

此範例定義一個 Application Load Balancer，其具備 HTTP 接聽程式和將流量轉送至目標群組的預設動作。此負載平衡器使用預設的運作狀態檢查設定。該目標群組包含兩個已註冊的 EC2 執行個體。

YAML

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myHTTPListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup

  myTargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
      Name: "my-target-group"
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
```

```
- Id: !GetAtt Instance1.InstanceId
  Port: 80
- Id: !GetAtt Instance2.InstanceId
  Port: 80
```

JSON

```
{
  "Resources": {
    "myLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
        "Name": "my-alb",
        "Type": "application",
        "Scheme": "internal",
        "Subnets": [
          {
            "Ref": "subnet-AZ1"
          },
          {
            "Ref": "subnet-AZ2"
          }
        ],
        "SecurityGroups": [
          {
            "Ref": "mySecurityGroup"
          }
        ]
      }
    },
    "myHTTPListener": {
      "Type": "AWS::ElasticLoadBalancingV2::Listener",
      "Properties": {
        "LoadBalancerArn": {
          "Ref": "myLoadBalancer"
        },
        "Protocol": "HTTP",
        "Port": 80,
        "DefaultActions": [
          {
            "Type": "forward",
            "TargetGroupArn": {
              "Ref": "myTargetGroup"
            }
          }
        ]
      }
    }
  }
}
```

```

    }
  }
]
},
"myTargetGroup": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "Name": "my-target-group",
    "Protocol": "HTTP",
    "Port": 80,
    "TargetType": "instance",
    "VpcId": {
      "Ref": "myVPC"
    },
    "Targets": [
      {
        "Id": {
          "Fn::GetAtt": [
            "Instance1",
            "InstanceId"
          ]
        },
        "Port": 80
      },
      {
        "Id": {
          "Fn::GetAtt": [
            "Instance2",
            "InstanceId"
          ]
        },
        "Port": 80
      }
    ]
  }
}
}
}
}
}

```

Classic Load Balancer 資源

此範例定義了一個 Classic Load Balancer，其具備 HTTP 接聽程式但沒有已註冊 EC2 執行個體。此負載平衡器使用預設的運作狀態檢查設定。

YAML

```
myLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP
```

JSON

```
"myLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
}
```

此範例定義了一個 Classic Load Balancer，其具備 HTTP 接聽程式、兩個已註冊 EC2 執行個體和自訂運作狀態檢查設定。

YAML

```
myClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
```

```
Instances:
- Ref: Instance1
- Ref: Instance2
Listeners:
- LoadBalancerPort: '80'
  InstancePort: '80'
  Protocol: HTTP
HealthCheck:
  Target: HTTP:80/
  HealthyThreshold: '3'
  UnhealthyThreshold: '5'
  Interval: '30'
  Timeout: '5'
```

JSON

```
"myClassicLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Instances" : [
      { "Ref" : "Instance1" },
      { "Ref" : "Instance2" }
    ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
}
```


AWS Identity and Access Management 範本程式碼片段

本節包含 AWS Identity and Access Management 範本程式碼片段。

主題

- [宣告 IAM 使用者資源](#)
- [宣告一個 IAM 存取金鑰資源](#)
- [宣告 IAM 群組資源](#)
- [將使用者新增至群組](#)
- [宣告 IAM 政策](#)
- [宣告 Amazon S3 儲存貯體政策](#)
- [宣告 Amazon SNS 主題政策](#)
- [宣告 Amazon SQS 政策](#)
- [IAM 角色範本範例](#)

Important

當使用包含 IAM 資源的範本建立或更新堆疊時，您必須認可使用 IAM 功能。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。

宣告 IAM 使用者資源

此程式碼片段會示範如何宣告 [AWS::IAM::User](#) 資源來建立 IAM 使用者。使用者會使用路徑 ("/") 宣告，登入描述檔則會使用密碼 (myP@ssW0rd) 宣告。

名為 giveaccesstoqueueonly 的政策文件會給予使用者在 Amazon SQS 佇列資源 myqueue 上執行所有 Amazon SQS 動作的許可，並拒絕存取所有其他的 Amazon SQS 佇列資源。Fn::GetAtt 函數會取得 [AWS::SQS::Queue](#) 資源 myqueue 的 Arn 屬性。

名為 giveaccesstotopiconly 的政策文件會新增到使用者，給予使用者在 Amazon SNS 主題資源 mytopic 上執行所有 Amazon SNS 動作的許可，並拒絕存取所有其他的 Amazon SNS 資源。Ref 函數會取得 [AWS::SNS::Topic](#) 資源 mytopic 的 ARN。

JSON

```
"myuser" : {
```

```

    "Type" : "AWS::IAM::User",
    "Properties" : {
        "Path" : "/",
        "LoginProfile" : {
            "Password" : "myP@ssW0rd"
        },
        "Policies" : [ {
            "PolicyName" : "giveaccesstoqueueonly",
            "PolicyDocument" : {
                "Version": "2012-10-17",
                "Statement" : [ {
                    "Effect" : "Allow",
                    "Action" : [ "sqs:*" ],
                    "Resource" : [ {
                        "Fn::GetAtt" : [ "myqueue", "Arn" ]
                    } ]
                }, {
                    "Effect" : "Deny",
                    "Action" : [ "sqs:*" ],
                    "NotResource" : [ {
                        "Fn::GetAtt" : [ "myqueue", "Arn" ]
                    } ]
                }
            ]
        }, {
            "PolicyName" : "giveaccesstotopiconly",
            "PolicyDocument" : {
                "Version": "2012-10-17",
                "Statement" : [ {
                    "Effect" : "Allow",
                    "Action" : [ "sns:*" ],
                    "Resource" : [ { "Ref" : "mytopic" } ]
                }, {
                    "Effect" : "Deny",
                    "Action" : [ "sns:*" ],
                    "NotResource" : [ { "Ref" : "mytopic" } ]
                }
            ]
        }
    ]
}

```

YAML

```
myuser:
  Type: AWS::IAM::User
  Properties:
    Path: "/"
    LoginProfile:
      Password: myP@ssW0rd
    Policies:
      - PolicyName: giveaccesstoqueueonly
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - sqs:*
              Resource:
                - !GetAtt myqueue.Arn
            - Effect: Deny
              Action:
                - sqs:*
              NotResource:
                - !GetAtt myqueue.Arn
      - PolicyName: giveaccesstotopiconly
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - sns:*
              Resource:
                - !Ref mytopic
            - Effect: Deny
              Action:
                - sns:*
              NotResource:
                - !Ref mytopic
```

宣告一個 IAM 存取金鑰資源

此程式碼片段顯示一個 [AWS::IAM::AccessKey](#) 資源。myaccesskey 資源會建立存取金鑰，並將其指派給一個已在範本中宣告為 [AWS::IAM::User](#) 資源的 IAM 使用者。

JSON

```
"myaccesskey" : {
  "Type" : "AWS::IAM::AccessKey",
  "Properties" : {
    "UserName" : { "Ref" : "myuser" }
  }
}
```

YAML

```
myaccesskey:
  Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      !Ref myuser
```

您可以使用 `AWS::IAM::AccessKey` 函數取得 `Fn::GetAtt` 資源的秘密金鑰。擷取秘密金鑰的其中一個方法，便是將其放入 `Output` 值中。您可以使用 `Ref` 函數取得存取金鑰。以下 `Output` 值宣告會取得 `myaccesskey` 的存取金鑰和秘密金鑰。

JSON

```
"AccessKeyformyaccesskey" : {
  "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
  "Value" : {
    "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
  }
}
```

YAML

```
AccessKeyformyaccesskey:
  Value:
    !Ref myaccesskey
SecretKeyformyaccesskey:
  Value: !GetAtt myaccesskey.SecretAccessKey
```

您也可以將 AWS 存取金鑰和私密金鑰傳遞至範本中定義的 Amazon EC2 執行個體或 Auto Scaling 群組。下列 [AWS::EC2::Instance](#) 宣告使用 UserData 屬性傳遞 myaccesskey 資源的存取金鑰和秘密金鑰。

JSON

```
"myinstance" : {
  "Type" : "AWS::EC2::Instance",
  "Properties" : {
    "AvailabilityZone" : "us-east-1a",
    "ImageId" : "ami-0ff8a91507f77f867",
    "UserData" : {
      "Fn::Base64" : {
        "Fn::Join" : [
          "", [
            "ACCESS_KEY=", {
              "Ref" : "myaccesskey"
            },
            "&",
            "SECRET_KEY=",
            {
              "Fn::GetAtt" : [
                "myaccesskey",
                "SecretAccessKey"
              ]
            }
          ]
        ]
      }
    ]
  }
}
```

YAML

```
myinstance:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone: "us-east-1a"
    ImageId: ami-0ff8a91507f77f867
    UserData:
```

```
Fn::Base64: !Sub "ACCESS_KEY=${myaccesskey}&SECRET_KEY=
${myaccesskey.SecretAccessKey}"
```

宣告 IAM 群組資源

此程式碼片段顯示一個 [AWS::IAM::Group](#) 資源。群組具有路徑 ("/myapplication/"). 名為 myapppolicy 的政策文件會新增到群組，允許群組的使用者在 Amazon SQS 佇列資源 myqueue 上執行所有 Amazon SQS 動作，並拒絕存取除 myqueue 之外的所有其他 Amazon SQS 資源。

若要將政策指派給資源，IAM 需要資源的 Amazon Resource Name (ARN)。在程式碼片段中，Fn::GetAtt 函數會取得 [AWS::SQS::Queue](#) 資源佇列的 ARN。

JSON

```
"mygroup" : {
  "Type" : "AWS::IAM::Group",
  "Properties" : {
    "Path" : "/myapplication/",
    "Policies" : [ {
      "PolicyName" : "myapppolicy",
      "PolicyDocument" : {
        "Version": "2012-10-17",
        "Statement" : [ {
          "Effect" : "Allow",
          "Action" : [ "sqs:*" ],
          "Resource" : [ {
            "Fn::GetAtt" : [ "myqueue", "Arn" ]
          } ]
        },
        {
          "Effect" : "Deny",
          "Action" : [ "sqs:*" ],
          "NotResource" : [ { "Fn::GetAtt" : [ "myqueue", "Arn" ] } ]
        }
      ]
    } ]
  }
}
```

YAML

```
mygroup:
```

```
Type: AWS::IAM::Group
Properties:
  Path: "/myapplication/"
  Policies:
    - PolicyName: myapppolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - sqs:*
            Resource: !GetAtt myqueue.Arn
          - Effect: Deny
            Action:
              - sqs:*
            NotResource: !GetAtt myqueue.Arn
```

將使用者新增至群組

[AWS::IAM::UserToGroupAddition](#) 資源會將使用者新增至群組。在以下程式碼片段中，`addUserToGroup` 資源會將下列使用者新增到名為 `myexistinggroup2` 的現有群組：現有使用者 `existinguser1` 和在範本中已宣告為 [AWS::IAM::User](#) 資源的使用者 `myuser`。

JSON

```
"addUserToGroup" : {
  "Type" : "AWS::IAM::UserToGroupAddition",
  "Properties" : {
    "GroupName" : "myexistinggroup2",
    "Users" : [ "existinguser1", { "Ref" : "myuser" } ]
  }
}
```

YAML

```
addUserToGroup:
  Type: AWS::IAM::UserToGroupAddition
  Properties:
    GroupName: myexistinggroup2
    Users:
      - existinguser1
      - !Ref myuser
```

宣告 IAM 政策

此程式碼片段會示範如何使用名為 mypolicy 的 [AWS::IAM::Policy](#) 資源建立政策，並將其套用到多個群組。mypolicy 資源包含 PolicyDocument 屬性，允許在由 ARN GetObject 代表之 S3 儲存貯體中的物件上執行 PutObject、PutObjectAcl 和 arn:aws:s3:::myAWSBucket 動作。mypolicy 資源會將政策套用至名為 myexistinggroup1 的現有群組，以及已在範本中宣告為 [AWS::IAM::Group](#) 資源的 mygroup 群組。此範例會示範如何使用 Groups 屬性將政策套用至群組。但是，您也可以改為使用 Users 屬性，將政策文件新增至使用者清單。

JSON

```
"mypolicy" : {
  "Type" : "AWS::IAM::Policy",
  "Properties" : {
    "PolicyName" : "mygrouppolicy",
    "PolicyDocument" : {
      "Version": "2012-10-17",
      "Statement" : [ {
        "Effect" : "Allow",
        "Action" : [
          "s3:GetObject" , "s3:PutObject" , "s3:PutObjectAcl" ],
        "Resource" : "arn:aws:s3:::myAWSBucket/*"
      } ]
    },
    "Groups" : [ "myexistinggroup1", { "Ref" : "mygroup" } ]
  }
}
```

YAML

```
mypolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: mygrouppolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - s3:GetObject
            - s3:PutObject
            - s3:PutObjectAcl
```



```

    Resource: arn:aws:s3:::myAWSBucket/*
  Groups:
    - myexistinggroup1
    - !Ref mygroup

```

宣告 Amazon S3 儲存貯體政策

此程式碼片段示範如何使用 [AWS::S3::BucketPolicy](#) 資源建立政策，並將其套用到 Amazon S3 儲存貯體。mybucketpolicy 資源會宣告一個政策文件，允許 user1 IAM 使用者在套用此政策的 S3 儲存貯體中的所有物件上執行 GetObject 動作。在程式碼片段中，Fn::GetAtt 函數會取得 user1 資源的 ARN。mybucketpolicy 資源會將政策套用到 AWS::S3::BucketPolicy 資源 mybucket。Ref 函數會取得 mybucket 資源的儲存貯體名稱。

JSON

```

"mybucketpolicy" : {
  "Type" : "AWS::S3::BucketPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "ReadAccess",
        "Action" : [ "s3:GetObject" ],
        "Effect" : "Allow",
        "Resource" : { "Fn::Join" : [
          "", [ "arn:aws:s3:::", { "Ref" : "mybucket" } , "/*" ]
        ] },
        "Principal" : {
          "AWS" : { "Fn::GetAtt" : [ "user1", "Arn" ] }
        }
      ]
    },
    "Bucket" : { "Ref" : "mybucket" }
  }
}

```

YAML

```

mybucketpolicy:
  Type: AWS::S3::BucketPolicy

```

```
Properties:
  PolicyDocument:
    Id: MyPolicy
    Version: '2012-10-17'
    Statement:
      - Sid: ReadAccess
        Action:
          - s3:GetObject
        Effect: Allow
        Resource: !Sub "arn:aws:s3:::${mybucket}/*"
        Principal:
          AWS: !GetAtt user1.Arn
    Bucket: !Ref mybucket
```

宣告 Amazon SNS 主題政策

此程式碼片段示範如何使用 [AWS::SNS::TopicPolicy](#) 資源建立政策，並將其套用到 Amazon SNS 主題。mysnspolicy 資源包含 PolicyDocument 屬性，允許 [AWS::IAM::User](#) 資源 myuser 在 [AWS::SNS::Topic](#) 資源 mytopic 上執行 Publish 動作。在程式碼片段中，Fn::GetAtt 函數會取得 myuser 資源的 ARN，並且 Ref 函數會取得 mytopic 資源的 ARN。

JSON

```
"mysnspolicy" : {
  "Type" : "AWS::SNS::TopicPolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyTopicPolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "My-statement-id",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : { "Fn::GetAtt" : [ "myuser", "Arn" ] }
        },
        "Action" : "sns:Publish",
        "Resource" : "*"
      } ]
    },
    "Topics" : [ { "Ref" : "mytopic" } ]
  }
}
```

YAML

```
mysnspolicy:
  Type: AWS::SNS::TopicPolicy
  Properties:
    PolicyDocument:
      Id: MyTopicPolicy
      Version: '2012-10-17'
      Statement:
        - Sid: My-statement-id
          Effect: Allow
          Principal:
            AWS: !GetAtt myuser.Arn
          Action: sns:Publish
          Resource: "*"
    Topics:
      - !Ref mytopic
```

宣告 Amazon SQS 政策

此程式碼片段示範如何使用 [AWS::SQS::QueuePolicy](#) 資源建立政策，並將其套用到 Amazon SQS 佇列。PolicyDocument 屬性允許現有的使用者 myapp (以其 ARN 指定) 在以其 URL 指定的現有佇列，以及一個 [AWS::SQS::Queue](#) 資源 myqueue 上執行 SendMessage 動作。[Ref](#) 函數會取得 myqueue 資源的 URL。

JSON

```
"mysqspolicy" : {
  "Type" : "AWS::SQS::QueuePolicy",
  "Properties" : {
    "PolicyDocument" : {
      "Id" : "MyQueuePolicy",
      "Version": "2012-10-17",
      "Statement" : [ {
        "Sid" : "Allow-User-SendMessage",
        "Effect" : "Allow",
        "Principal" : {
          "AWS" : "arn:aws:iam::123456789012:user/myapp"
        },
        "Action" : [ "sqs:SendMessage" ],
        "Resource" : "*"
      } ]
    }
  },
},
```

```
"Queues" : [
  "https://sqs.us-east-2aws-region.amazonaws.com/123456789012/myexistingqueue",
  { "Ref" : "myqueue" }
]
}
```

YAML

```
mysqspolicy:
  Type: AWS::SQS::QueuePolicy
  Properties:
    PolicyDocument:
      Id: MyQueuePolicy
      Version: '2012-10-17'
      Statement:
        - Sid: Allow-User-SendMessage
          Effect: Allow
          Principal:
            AWS: arn:aws:iam::123456789012:user/myapp
          Action:
            - sqs:SendMessage
          Resource: "*"
    Queues:
      - https://sqs.aws-region.amazonaws.com/123456789012/myexistingqueue
      - !Ref myqueue
```

IAM 角色範本範例

本章節提供 EC2 執行個體 IAM 角色的 CloudFormation 範本範例。

如需詳細資訊，請參閱 Amazon EC2 User Guide 中的 [IAM roles for Amazon EC2](#)。

IAM 角色與 EC2

在此範例中，EC2 執行個體的 `IamInstanceProfile` 屬性會參考執行個體設定檔。執行個體政策和角色政策都會參考 [AWS::IAM::Role](#)。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
```

```
"myEC2Instance": {
  "Type": "AWS::EC2::Instance",
  "Version": "2009-05-15",
  "Properties": {
    "ImageId": "ami-0ff8a91507f77f867",
    "InstanceType": "m1.small",
    "Monitoring": "true",
    "DisableApiTermination": "false",
    "IamInstanceProfile": {
      "Ref": "RootInstanceProfile"
    }
  }
},
"RootRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [ {
        "Effect": "Allow",
        "Principal": {
          "Service": [ "ec2.amazonaws.com" ]
        },
        "Action": [ "sts:AssumeRole" ]
      } ]
    },
    "Path": "/"
  }
},
"RolePolicies": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyName": "root",
    "PolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [ {
        "Effect": "Allow",
        "Action": "*",
        "Resource": "*"
      } ]
    },
    "Roles": [ { "Ref": "RootRole" } ]
  }
},
```

```

    "RootInstanceProfile": {
      "Type": "AWS::IAM::InstanceProfile",
      "Properties": {
        "Path": "/",
        "Roles": [ { "Ref": "RootRole" } ]
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myEC2Instance:
    Type: AWS::EC2::Instance
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      Monitoring: 'true'
      DisableApiTermination: 'false'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'

```

```
    Statement:
      - Effect: Allow
        Action: "*"
        Resource: "*"
    Roles:
      - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - !Ref RootRole
```

IAM 角色與 Auto Scaling 群組

在此範例中，Amazon EC2 Auto Scaling 啟動組態的 `IamInstanceProfile` 屬性會參考執行個體設定檔。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "myLCOne": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Version": "2009-05-15",
      "Properties": {
        "ImageId": "ami-0ff8a91507f77f867",
        "InstanceType": "m1.small",
        "InstanceMonitoring": "true",
        "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
      }
    },
    "myASGrpOne": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Version": "2009-05-15",
      "Properties": {
        "AvailabilityZones": [ "us-east-1a" ],
        "LaunchConfigurationName": { "Ref": "myLCOne" },
        "MinSize": "0",
        "MaxSize": "0",
        "HealthCheckType": "EC2",
        "HealthCheckGracePeriod": "120"
```

```

    }
  },
  "RootRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [ {
          "Effect": "Allow",
          "Principal": {
            "Service": [ "ec2.amazonaws.com" ]
          },
          "Action": [ "sts:AssumeRole" ]
        } ]
      },
      "Path": "/"
    }
  },
  "RolePolicies": {
    "Type": "AWS::IAM::Policy",
    "Properties": {
      "PolicyName": "root",
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [ {
          "Effect": "Allow",
          "Action": "*",
          "Resource": "*"
        } ]
      },
      "Roles": [ { "Ref": "RootRole" } ]
    }
  },
  "RootInstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
      "Path": "/",
      "Roles": [ { "Ref": "RootRole" } ]
    }
  }
}

```


YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  myLCOne:
    Type: AWS::AutoScaling::LaunchConfiguration
    Version: '2009-05-15'
    Properties:
      ImageId: ami-0ff8a91507f77f867
      InstanceType: m1.small
      InstanceMonitoring: 'true'
      IamInstanceProfile:
        !Ref RootInstanceProfile
  myASGrpOne:
    Type: AWS::AutoScaling::AutoScalingGroup
    Version: '2009-05-15'
    Properties:
      AvailabilityZones:
        - "us-east-1a"
      LaunchConfigurationName:
        !Ref myLCOne
      MinSize: '0'
      MaxSize: '0'
      HealthCheckType: EC2
      HealthCheckGracePeriod: '120'
  RootRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: "/"
  RolePolicies:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: root
      PolicyDocument:
        Version: '2012-10-17'
```

```

    Statement:
      - Effect: Allow
        Action: "*"
        Resource: "*"
    Roles:
      - !Ref RootRole
  RootInstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: "/"
      Roles:
        - !Ref RootRole

```

AWS Lambda 範本

下列範本使用 AWS Lambda (Lambda) 函數和自訂資源，將新的安全群組附加至現有安全群組的清單。當您想動態建置安全群組清單，使該清單能夠同時包含全新與現有的安全群組時，此函數相當實用。例如，您可以將現有安全群組清單傳遞為參數值，並將新數值附加至該清單，接著在所有數值與 EC2 執行個體之間建立關聯。如需 Lambda 函式資源類型的詳細資訊，請參閱 [AWS::Lambda::Function](#)。

在此範例中，當 CloudFormation 建立 AllSecurityGroups 自訂資源時，CloudFormation 即會調用 AppendItemToListFunction Lambda 函式。接著，CloudFormation 會將現有安全群組清單與新的安全群組 (NewSecurityGroup) 傳遞至該函數，並透過此函數將新的安全群組附加至清單，然後傳回修改後的清單。而修改後的清單則可讓 CloudFormation 將所有安全群組與 MyEC2Instance 資源建立關聯。

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ExistingSecurityGroups": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>"
    },
    "ExistingVPC": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "The VPC ID that includes the security groups in the ExistingSecurityGroups parameter."
    },
    "InstanceType": {
      "Type": "String",

```

```

        "Default": "t2.micro",
        "AllowedValues": [
            "t2.micro",
            "t3.micro"
        ]
    },
    "Resources": {
        "SecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Allow HTTP traffic to the host",
                "VpcId": {
                    "Ref": "ExistingVPC"
                },
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "CidrIp": "0.0.0.0/0"
                    }
                ],
                "SecurityGroupEgress": [
                    {
                        "IpProtocol": "tcp",
                        "FromPort": 80,
                        "ToPort": 80,
                        "CidrIp": "0.0.0.0/0"
                    }
                ]
            }
        },
        "AllSecurityGroups": {
            "Type": "Custom::Split",
            "Properties": {
                "ServiceToken": {
                    "Fn::GetAtt": [
                        "AppendItemToListFunction",
                        "Arn"
                    ]
                },
                "List": {

```

```

        "Ref": "ExistingSecurityGroups"
      },
      "AppendedItem": {
        "Ref": "SecurityGroup"
      }
    }
  },
  "AppendItemToListFunction": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Role": {
        "Fn::GetAtt": [
          "LambdaExecutionRole",
          "Arn"
        ]
      },
      "Code": {
        "ZipFile": {
          "Fn::Join": [
            "",
            [
              "var response = require('cfn-response');",
              "exports.handler = function(event, context) {",
              "  var responseData = {Value:",
event.ResourceProperties.List};",
              "  ",
              "  responseData.Value.push(event.ResourceProperties.AppendedItem);",
              "  response.send(event, context, response.SUCCESS,",
              "  responseData);",
              "};"
            ]
          ]
        }
      },
      "Runtime": "nodejs20.x"
    }
  },
  "MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": "{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}",
      "SecurityGroupIds": {

```

```

        "Fn::GetAtt": [
            "AllSecurityGroups",
            "Value"
        ],
        "InstanceType": {
            "Ref": "InstanceType"
        }
    },
    "LambdaExecutionRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "lambda.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ]
                    }
                ]
            }
        },
        "Path": "/",
        "Policies": [
            {
                "PolicyName": "root",
                "PolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": [
                                "logs:*"
                            ],
                            "Resource": "arn:aws:logs:*:*:*"
                        }
                    ]
                }
            }
        ]
    }
}

```

```

    }
  }
]
}
},
"Outputs": {
  "AllSecurityGroups": {
    "Description": "Security Groups that are associated with the EC2 instance",
    "Value": {
      "Fn::Join": [
        ",",
        {
          "Fn::GetAtt": [
            "AllSecurityGroups",
            "Value"
          ]
        }
      ]
    }
  }
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  ExistingSecurityGroups:
    Type: List<AWS::EC2::SecurityGroup::Id>
  ExistingVPC:
    Type: AWS::EC2::VPC::Id
    Description: The VPC ID that includes the security groups in the
ExistingSecurityGroups parameter.
  InstanceType:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t3.micro
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup

```

```

Properties:
  GroupDescription: Allow HTTP traffic to the host
  VpcId: !Ref ExistingVPC
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
  SecurityGroupEgress:
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
AllSecurityGroups:
  Type: Custom::Split
  Properties:
    ServiceToken: !GetAtt AppendItemToListFunction.Arn
    List: !Ref ExistingSecurityGroups
    AppendedItem: !Ref SecurityGroup
AppendItemToListFunction:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile: !Join
        - ''
        - - var response = require('cfn-response');
          - exports.handler = function(event, context) {
            - '   var responseData = {Value: event.ResourceProperties.List};'
            - '   responseData.Value.push(event.ResourceProperties.AppendedItem);'
            - '   response.send(event, context, response.SUCCESS, responseData);'
            - '};'
    Runtime: nodejs20.x
MyEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: '{{resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2}}'
    SecurityGroupIds: !GetAtt AllSecurityGroups.Value
    InstanceType: !Ref InstanceType
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:

```

```
AssumeRolePolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
      Principal:
        Service:
          - lambda.amazonaws.com
      Action:
        - sts:AssumeRole
  Path: /
Policies:
  - PolicyName: root
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - logs:*
          Resource: arn:aws:logs:*:*:*
Outputs:
  AllSecurityGroups:
    Description: Security Groups that are associated with the EC2 instance
    Value: !Join
      - ', '
      - !GetAtt AllSecurityGroups.Value
```

Amazon Redshift 範本程式碼片段

Amazon Redshift 是一種在雲端中完全受管的 PB 級資料倉儲服務。您可以使用 CloudFormation 來佈建並管理 Amazon Redshift 叢集。

Amazon Redshift 叢集

系統建立堆疊時會指定參數值，而以下範例範本會根據該參數值來建立 Amazon Redshift 叢集。另外，與 Amazon Redshift 叢集相關的叢集參數群組則會啟用使用者活動記錄功能。此範本還會在自身定義的 Amazon VPC 中啟動 Amazon Redshift 叢集。該 VPC 將包含網際網路閘道，讓您可以從網際網路存取 Amazon Redshift 叢集。不過，您還是必須要透過路由表項目來啟用叢集和網際網路閘道之間的通訊。

 Note

範本會包含 `IsMultiNodeCluster` 條件，因此系統唯有在 `NumberOfNodes` 參數值設為 `ClusterType` 時，才會宣告 `multi-node` 參數。

範例定義的 `MysqlRootPassword` 參數已將其 `NoEcho` 屬性設為 `true`。若您將 `NoEcho` 屬性設為 `true`，CloudFormation 會將任何描述堆疊或堆疊事件呼叫所傳回的參數值以星號 (*****) 遮罩，但儲存在以下指定位置中的資訊除外。

 Important

使用 `NoEcho` 屬性不會遮罩任何儲存在下列資訊中的資訊：

- Metadata 範本區段。CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。若要取得更多資訊，請參閱[中繼資料](#)。
- Outputs 範本區段。如需詳細資訊，請參閱[輸出](#)。
- 資源定義的 Metadata 屬性。如需詳細資訊，請參閱[Metadata 屬性](#)。

我們強烈建議您不要使用這些機制來包含敏感資訊，例如密碼或秘密。

 Important

我們建議您不要直接在 CloudFormation 範本中嵌入敏感資訊，而是在堆疊範本中使用動態參數來參考在 CloudFormation 外部存放和管理的敏感資訊，例如在 AWS Systems Manager 參數存放區或中 AWS Secrets Manager。

如需詳細資訊，請參閱[請勿在範本中內嵌認證](#)的最佳實務。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the cluster is created",
```

```
"Type" : "String",
"Default" : "dev",
"AllowedPattern" : "([a-z]|[0-9])+"
},
"ClusterType" : {
  "Description" : "The type of cluster",
  "Type" : "String",
  "Default" : "single-node",
  "AllowedValues" : [ "single-node", "multi-node" ]
},
"NumberOfNodes" : {
  "Description" : "The number of compute nodes in the cluster. For multi-node clusters, the NumberOfNodes parameter must be greater than 1",
  "Type" : "Number",
  "Default" : "1"
},
"NodeType" : {
  "Description" : "The type of node to be provisioned",
  "Type" : "String",
  "Default" : "ds2.xlarge",
  "AllowedValues" : [ "ds2.xlarge", "ds2.8xlarge", "dc1.large", "dc1.8xlarge" ]
},
"MasterUsername" : {
  "Description" : "The user name that is associated with the master user account for the cluster that is being created",
  "Type" : "String",
  "Default" : "defaultuser",
  "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
},
"MasterUserPassword" : {
  "Description" : "The password that is associated with the master user account for the cluster that is being created.",
  "Type" : "String",
  "NoEcho" : "true"
},
"InboundTraffic" : {
  "Description" : "Allow inbound traffic to the cluster from this CIDR range.",
  "Type" : "String",
  "MinLength": "9",
  "MaxLength": "18",
  "Default" : "0.0.0.0/0",
  "AllowedPattern" : "(\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,3})\\.\\.\\.\\.((\\d{1,2})|)",
  "ConstraintDescription" : "must be a valid CIDR range of the form x.x.x.x/x."
```

```

    },
    "PortNumber" : {
        "Description" : "The port number on which the cluster accepts incoming
connections.",
        "Type" : "Number",
        "Default" : "5439"
    }
},
"Conditions" : {
    "IsMultiNodeCluster" : {
        "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
    }
},
"Resources" : {
    "RedshiftCluster" : {
        "Type" : "AWS::Redshift::Cluster",
        "DependsOn" : "AttachGateway",
        "Properties" : {
            "ClusterType" : { "Ref" : "ClusterType" },
            "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster", { "Ref" :
"NumberOfNodes" }, { "Ref" : "AWS::NoValue" } ]}],
            "NodeType" : { "Ref" : "NodeType" },
            "DBName" : { "Ref" : "DatabaseName" },
            "MasterUsername" : { "Ref" : "MasterUsername" },
            "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
            "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup" },
            "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
            "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
            "PubliclyAccessible" : "true",
            "Port" : { "Ref" : "PortNumber" }
        }
    },
    "RedshiftClusterParameterGroup" : {
        "Type" : "AWS::Redshift::ClusterParameterGroup",
        "Properties" : {
            "Description" : "Cluster parameter group",
            "ParameterGroupFamily" : "redshift-1.0",
            "Parameters" : [{
                "ParameterName" : "enable_user_activity_logging",
                "ParameterValue" : "true"
            }]
        }
    },
    "RedshiftClusterSubnetGroup" : {

```

```

    "Type" : "AWS::Redshift::ClusterSubnetGroup",
    "Properties" : {
        "Description" : "Cluster subnet group",
        "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
    }
},
"VPC" : {
    "Type" : "AWS::EC2::VPC",
    "Properties" : {
        "CidrBlock" : "10.0.0.0/16"
    }
},
"PublicSubnet" : {
    "Type" : "AWS::EC2::Subnet",
    "Properties" : {
        "CidrBlock" : "10.0.0.0/24",
        "VpcId" : { "Ref" : "VPC" }
    }
},
"SecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Security group",
        "SecurityGroupIngress" : [ {
            "CidrIp" : { "Ref" : "InboundTraffic" },
            "FromPort" : { "Ref" : "PortNumber" },
            "ToPort" : { "Ref" : "PortNumber" },
            "IpProtocol" : "tcp"
        } ],
        "VpcId" : { "Ref" : "VPC" }
    }
},
"myInternetGateway" : {
    "Type" : "AWS::EC2::InternetGateway"
},
"AttachGateway" : {
    "Type" : "AWS::EC2::VPCGatewayAttachment",
    "Properties" : {
        "VpcId" : { "Ref" : "VPC" },
        "InternetGatewayId" : { "Ref" : "myInternetGateway" }
    }
},
"PublicRouteTable" : {
    "Type" : "AWS::EC2::RouteTable",

```

```

    "Properties" : {
      "VpcId" : {
        "Ref" : "VPC"
      }
    },
    "PublicRoute" : {
      "Type" : "AWS::EC2::Route",
      "DependsOn" : "AttachGateway",
      "Properties" : {
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        },
        "DestinationCidrBlock" : "0.0.0.0/0",
        "GatewayId" : {
          "Ref" : "myInternetGateway"
        }
      }
    },
    "PublicSubnetRouteTableAssociation" : {
      "Type" : "AWS::EC2::SubnetRouteTableAssociation",
      "Properties" : {
        "SubnetId" : {
          "Ref" : "PublicSubnet"
        },
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        }
      }
    },
    "Outputs" : {
      "ClusterEndpoint" : {
        "Description" : "Cluster endpoint",
        "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Port" ] } ] ] }
      },
      "ClusterName" : {
        "Description" : "Name of cluster",
        "Value" : { "Ref" : "RedshiftCluster" }
      },
      "ParameterGroupName" : {
        "Description" : "Name of parameter group",

```

```

    "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
  },
  "RedshiftClusterSubnetGroupName" : {
    "Description" : "Name of cluster subnet group",
    "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
  },
  "RedshiftClusterSecurityGroupName" : {
    "Description" : "Name of cluster security group",
    "Value" : { "Ref" : "SecurityGroup" }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Parameters:
  DatabaseName:
    Description: The name of the first database to be created when the cluster is
      created
    Type: String
    Default: dev
    AllowedPattern: "([a-z]|[0-9])+\"
  ClusterType:
    Description: The type of cluster
    Type: String
    Default: single-node
    AllowedValues:
      - single-node
      - multi-node
  NumberOfNodes:
    Description: The number of compute nodes in the cluster. For multi-node clusters,
      the NumberOfNodes parameter must be greater than 1
    Type: Number
    Default: '1'
  NodeType:
    Description: The type of node to be provisioned
    Type: String
    Default: ds2.xlarge
    AllowedValues:
      - ds2.xlarge
      - ds2.8xlarge
      - dc1.large

```

```
- dc1.8xlarge
```

```
MasterUsername:  
  Description: The user name that is associated with the master user account for  
    the cluster that is being created  
  Type: String  
  Default: defaultuser  
  AllowedPattern: "([a-z])([a-z]|[0-9])*"
```

```
MasterUserPassword:  
  Description: The password that is associated with the master user account for  
    the cluster that is being created.  
  Type: String  
  NoEcho: 'true'
```

```
InboundTraffic:  
  Description: Allow inbound traffic to the cluster from this CIDR range.  
  Type: String  
  MinLength: '9'  
  MaxLength: '18'  
  Default: 0.0.0.0/0  
  AllowedPattern: "(\\d{1,3})\\.((\\d{1,3})|\\.((\\d{1,3})|\\.((\\d{1,3})/(\\d{1,2})))"  
  ConstraintDescription: must be a valid CIDR range of the form x.x.x.x/x.
```

```
PortNumber:  
  Description: The port number on which the cluster accepts incoming connections.  
  Type: Number  
  Default: '5439'
```

```
Conditions:
```

```
IsMultiNodeCluster:  
  Fn::Equals:  
    - Ref: ClusterType  
    - multi-node
```

```
Resources:
```

```
RedshiftCluster:  
  Type: AWS::Redshift::Cluster  
  DependsOn: AttachGateway  
  Properties:  
    ClusterType:  
      Ref: ClusterType  
    NumberOfNodes:  
      Fn::If:  
        - IsMultiNodeCluster  
        - Ref: NumberOfNodes  
        - Ref: AWS::NoValue  
    NodeType:  
      Ref: NodeType  
    DBName:
```

```
    Ref: DatabaseName
  MasterUsername:
    Ref: MasterUsername
  MasterUserPassword:
    Ref: MasterUserPassword
  ClusterParameterGroupName:
    Ref: RedshiftClusterParameterGroup
  VpcSecurityGroupIds:
    - Ref: SecurityGroup
  ClusterSubnetGroupName:
    Ref: RedshiftClusterSubnetGroup
  PubliclyAccessible: 'true'
  Port:
    Ref: PortNumber
RedshiftClusterParameterGroup:
  Type: AWS::Redshift::ClusterParameterGroup
  Properties:
    Description: Cluster parameter group
    ParameterGroupFamily: redshift-1.0
    Parameters:
      - ParameterName: enable_user_activity_logging
        ParameterValue: 'true'
RedshiftClusterSubnetGroup:
  Type: AWS::Redshift::ClusterSubnetGroup
  Properties:
    Description: Cluster subnet group
    SubnetIds:
      - Ref: PublicSubnet
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
PublicSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock: 10.0.0.0/24
    VpcId:
      Ref: VPC
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group
    SecurityGroupIngress:
      - CidrIp:
```



```

        Ref: InboundTraffic
    FromPort:
        Ref: PortNumber
    ToPort:
        Ref: PortNumber
    IpProtocol: tcp
    VpcId:
        Ref: VPC
myInternetGateway:
    Type: AWS::EC2::InternetGateway
AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
        VpcId:
            Ref: VPC
        InternetGatewayId:
            Ref: myInternetGateway
PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
        VpcId:
            Ref: VPC
PublicRoute:
    Type: AWS::EC2::Route
    DependsOn: AttachGateway
    Properties:
        RouteTableId:
            Ref: PublicRouteTable
        DestinationCidrBlock: 0.0.0.0/0
        GatewayId:
            Ref: myInternetGateway
PublicSubnetRouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
        SubnetId:
            Ref: PublicSubnet
        RouteTableId:
            Ref: PublicRouteTable
Outputs:
    ClusterEndpoint:
        Description: Cluster endpoint
        Value: !Sub "${RedshiftCluster.Endpoint.Address}:${RedshiftCluster.Endpoint.Port}"
    ClusterName:
        Description: Name of cluster

```

```
Value:
  Ref: RedshiftCluster
ParameterGroupName:
  Description: Name of parameter group
  Value:
    Ref: RedshiftClusterParameterGroup
RedshiftClusterSubnetGroupName:
  Description: Name of cluster subnet group
  Value:
    Ref: RedshiftClusterSubnetGroup
RedshiftClusterSecurityGroupName:
  Description: Name of cluster security group
  Value:
    Ref: SecurityGroup
```

另請參閱

[AWS::Redshift::Cluster](#)

Amazon RDS 範本程式碼片段

主題

- [Amazon RDS 資料庫執行個體資源](#)
- [Amazon RDS Oracle Database 資料庫執行個體資源](#)
- [適用於 CIDR 範圍的 Amazon RDS DBSecurityGroup 資源](#)
- [具有 Amazon EC2 安全群組的 Amazon RDS DBSecurityGroup](#)
- [多個 VPC 安全群組](#)
- [VPC 安全群組中的 Amazon RDS 資料庫執行個體](#)

Amazon RDS 資料庫執行個體資源

此範例顯示具有受管主要使用者密碼的 Amazon RDS 資料庫執行個體資源。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的[使用 AWS Secrets Manager進行密碼管理](#)和《Aurora 使用者指南》中的[使用 AWS Secrets Manager進行密碼管理](#)。由於未指定選用的 EngineVersion 屬性，此資料庫執行個體將使用預設的引擎版本。如需有關預設引擎版本和其他預設設定的詳細資訊，請參閱 [CreateDBInstance](#)。DBSecurityGroups 屬性授權網路輸入至名為 MyDbSecurityByEC2SecurityGroup 和 MyDbSecurityByCIDRIPGroup 的 AWS::RDS::DBSecurityGroup 資源。如需詳細資訊，請參閱[AWS::RDS::DBInstance](#)。資料庫執行

個體資源也有設定為 Snapshot 的 DeletionPolicy 屬性。使用 SnapshotDeletionPolicy 集，CloudFormation 會先取得此資料庫執行個體的快照，再於堆疊刪除期間將其刪除。

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "MySQL",
    "MasterUsername" : "MyName",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
    Engine: MySQL
    MasterUsername: MyName
    ManageMasterUserPassword: true
    MasterUserSecret:
      KmsKeyId: !Ref KMSKey
  DeletionPolicy: Snapshot
```

Amazon RDS Oracle Database 資料庫執行個體資源

此範例建立具有受管主要使用者密碼的 Oracle Database 資料庫執行個體資源。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的[使用 AWS Secrets Manager 進行密碼管理](#)。此範例指定 Engine 做為 oracle-ee 並使用自有授權的授權模式。如需有關 Oracle Database 資料庫執行個體的詳細資訊，請參閱 [CreateDBInstance](#)。DBSecurityGroups 屬性授權網路輸入至名為 MyDbSecurityByEC2SecurityGroup 和 MyDbSecurityByCIDRIPGroup 的 AWS::RDS::DBSecurityGroup 資源。如需詳細資訊，請參閱[AWS::RDS::DBInstance](#)。資料庫執行個體資源也有設定為 Snapshot 的 DeletionPolicy 屬性。使用 SnapshotDeletionPolicy 集，CloudFormation 會先取得此資料庫執行個體的快照，再於堆疊刪除期間將其刪除。

JSON

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "DBSecurityGroups" : [
      {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" :
"MyDbSecurityByCIDRIPGroup"} ],
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.t2.small",
    "Engine" : "oracle-ee",
    "LicenseModel" : "bring-your-own-license",
    "MasterUsername" : "master",
    "ManageMasterUserPassword" : true,
    "MasterUserSecret" : {
      "KmsKeyId" : {"Ref" : "KMSKey"}
    }
  },
  "DeletionPolicy" : "Snapshot"
}
```

YAML

```
MyDB:
  Type: AWS::RDS::DBInstance
  Properties:
    DBSecurityGroups:
      - Ref: MyDbSecurityByEC2SecurityGroup
      - Ref: MyDbSecurityByCIDRIPGroup
    AllocatedStorage: '5'
    DBInstanceClass: db.t2.small
```

```
Engine: oracle-ee
LicenseModel: bring-your-own-license
MasterUsername: master
ManageMasterUserPassword: true
MasterUserSecret:
  KmsKeyId: !Ref KMSKey
DeletionPolicy: Snapshot
```

適用於 CIDR 範圍的 Amazon RDS DBSecurityGroup 資源

此範例顯示 Amazon RDS DBSecurityGroup 資源與具有 ddd.ddd.ddd.ddd/dd 格式之指定 CIDR 範圍的輸入授權。如需詳細資訊，請參閱 [AWS::RDS::DBSecurityGroup](#) 和 [Ingress](#)。

JSON

```
"MyDbSecurityByCIDRIPGroup" : {
  "Type" : "AWS::RDS::DBSecurityGroup",
  "Properties" : {
    "GroupDescription" : "Ingress for CIDRIP",
    "DBSecurityGroupIngress" : {
      "CIDRIP" : "192.168.0.0/32"
    }
  }
}
```

YAML

```
MyDbSecurityByCIDRIPGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    GroupDescription: Ingress for CIDRIP
    DBSecurityGroupIngress:
      CIDRIP: "192.168.0.0/32"
```

具有 Amazon EC2 安全群組的 Amazon RDS DBSecurityGroup

此範例顯示 [AWS::RDS::DBSecurityGroup](#) 資源與來自 MyEc2SecurityGroup 參考之 Amazon EC2 安全群組的輸入授權。

若要這樣做，請您定義 EC2 安全群組，然後使用內建 Ref 函數，以參照您的 DBSecurityGroup 內的 EC2 安全群組。

JSON

```

"DBInstance" : {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
    "DBName"          : { "Ref" : "DBName" },
    "Engine"          : "MySQL",
    "MasterUsername"   : { "Ref" : "DBUsername" },
    "DBInstanceClass"  : { "Ref" : "DBClass" },
    "DBSecurityGroups" : [ { "Ref" : "DBSecurityGroup" } ],
    "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
    "MasterUserPassword": { "Ref" : "DBPassword" }
  }
},

"DBSecurityGroup": {
  "Type": "AWS::RDS::DBSecurityGroup",
  "Properties": {
    "DBSecurityGroupIngress": {
      "EC2SecurityGroupName": {
        "Fn::GetAtt": ["WebServerSecurityGroup", "GroupName"]
      }
    },
    "GroupDescription" : "Frontend Access"
  }
},

"WebServerSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
    "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
    "SecurityGroupIngress" : [
      { "IpProtocol" : "tcp", "FromPort" : 80, "ToPort" : 80, "CidrIp" :
"0.0.0.0/0"},
      { "IpProtocol" : "tcp", "FromPort" : 22, "ToPort" : 22, "CidrIp" : "0.0.0.0/0"}
    ]
  }
}

```

YAML

此範例擷取自以下完整範例：[Drupal_Single_Instance_With_RDS.template](#)

```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MasterUsername:
      Ref: DBUsername
    DBInstanceClass:
      Ref: DBClass
    DBSecurityGroups:
      - Ref: DBSecurityGroup
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword
DBSecurityGroup:
  Type: AWS::RDS::DBSecurityGroup
  Properties:
    DBSecurityGroupIngress:
      EC2SecurityGroupName:
        Ref: WebServerSecurityGroup
      GroupDescription: Frontend Access
WebServerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Enable HTTP access via port 80 and SSH access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
```

多個 VPC 安全群組

此範例顯示 [AWS::RDS::DBSecurityGroup](#) 資源與 [AWS::RDS::DBSecurityGroupIngress](#) 中多個 Amazon EC2 VPC 安全群組的輸入授權。

JSON

```
{
  "Resources" : {
    "DBInstance" : {
      "Type" : "AWS::RDS::DBInstance",
      "Properties" : {
        "AllocatedStorage" : "5",
        "DBInstanceClass" : "db.t2.small",
        "DBName" : { "Ref" : "MyDBName" },
        "DBSecurityGroups" : [ { "Ref" : "DbSecurityByEC2SecurityGroup" } ],
        "DBSubnetGroupName" : { "Ref" : "MyDBSubnetGroup" },
        "Engine" : "MySQL",
        "MasterUserPassword": { "Ref" : "MyDBPassword" },
        "MasterUsername" : { "Ref" : "MyDBUsername" }
      },
      "DeletionPolicy" : "Snapshot"
    },
    "DbSecurityByEC2SecurityGroup" : {
      "Type" : "AWS::RDS::DBSecurityGroup",
      "Properties" : {
        "GroupDescription" : "Ingress for Amazon EC2 security group",
        "EC2VpcId" : { "Ref" : "MyVPC" },
        "DBSecurityGroupIngress" : [ {
          "EC2SecurityGroupId" : "sg-b0ff1111",
          "EC2SecurityGroupOwnerId" : "111122223333"
        }, {
          "EC2SecurityGroupId" : "sg-ffd72222",
          "EC2SecurityGroupOwnerId" : "111122223333"
        } ]
      }
    }
  }
}
```

YAML

```
Resources:
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      AllocatedStorage: '5'
      DBInstanceClass: db.t2.small
```



```

    DBName:
      Ref: MyDBName
    DBSecurityGroups:
      - Ref: DbSecurityByEC2SecurityGroup
    DBSubnetGroupName:
      Ref: MyDBSubnetGroup
    Engine: MySQL
    MasterUserPassword:
      Ref: MyDBPassword
    MasterUsername:
      Ref: MyDBUsername
    DeletionPolicy: Snapshot
  DbSecurityByEC2SecurityGroup:
    Type: AWS::RDS::DBSecurityGroup
    Properties:
      GroupDescription: Ingress for Amazon EC2 security group
      EC2VpcId:
        Ref: MyVPC
      DBSecurityGroupIngress:
        - EC2SecurityGroupId: sg-b0ff1111
          EC2SecurityGroupOwnerId: '111122223333'
        - EC2SecurityGroupId: sg-ffd72222
          EC2SecurityGroupOwnerId: '111122223333'

```

VPC 安全群組中的 Amazon RDS 資料庫執行個體

此範例顯示與 Amazon EC2 VPC 安全群組關聯的 Amazon RDS 資料庫執行個體。

JSON

```

{
  "DBEC2SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription": "Open database for access",
      "SecurityGroupIngress" : [{
        "IpProtocol" : "tcp",
        "FromPort" : 3306,
        "ToPort" : 3306,
        "SourceSecurityGroupName" : { "Ref" : "WebServerSecurityGroup" }
      }]
    }
  },

```

```

"DBInstance" : {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
    "DBName"          : { "Ref" : "DBName" },
    "Engine"          : "MySQL",
    "MultiAZ"         : { "Ref": "MultiAZDatabase" },
    "MasterUsername"  : { "Ref" : "DBUser" },
    "DBInstanceClass" : { "Ref" : "DBClass" },
    "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
    "MasterUserPassword": { "Ref" : "DBPassword" },
    "VPCSecurityGroups" : [ { "Fn::GetAtt": [ "DBEC2SecurityGroup", "GroupId" ] } ]
  }
}

```

YAML

```

DBEC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Open database for access
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 3306
        ToPort: 3306
        SourceSecurityGroupName:
          Ref: WebServerSecurityGroup
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    DBName:
      Ref: DBName
    Engine: MySQL
    MultiAZ:
      Ref: MultiAZDatabase
    MasterUsername:
      Ref: DBUser
    DBInstanceClass:
      Ref: DBClass
    AllocatedStorage:
      Ref: DBAllocatedStorage
    MasterUserPassword:
      Ref: DBPassword

```

```
VPCSecurityGroups:
- !GetAtt DBEC2SecurityGroup.GroupId
```

Route 53 範本程式碼片段

主題

- [使用託管區域名稱或 ID 的 Amazon Route 53 資源紀錄集](#)
- [使用 RecordSetGroup 設定加權資源紀錄集](#)
- [使用 RecordSetGroup 設定別名資源紀錄集](#)
- [CloudFront 分佈的別名資源紀錄集](#)

使用託管區域名稱或 ID 的 Amazon Route 53 資源紀錄集

當您建立 Amazon Route 53 資源紀錄集時，您必須指定您希望新增它的託管區域。CloudFormation 提供指定託管區域的兩種方式：

- 您可以使用 HostedZoneId 屬性來明確指定託管區域。
- 您可以使用 HostedZoneName 屬性讓 CloudFormation 尋找託管區域。若您使用 HostedZoneName 屬性，而且有多個相同名稱的託管區域，則 CloudFormation 不會建立堆疊。

使用 HostedZoneId 新增 RecordSet

此範例會新增包含使用 HostedZoneId 屬性指定託管區域之網域名稱 mysite.example.com SPF 記錄的 Amazon Route 53 資源紀錄集，

JSON

```
"myDNSRecord" : {
  "Type" : "AWS::Route53::RecordSet",
  "Properties" :
  {
    "HostedZoneId" : "Z3DG6IL3SJCGPX",
    "Name" : "mysite.example.com.",
    "Type" : "SPF",
    "TTL" : "900",
    "ResourceRecords" : [ "\"v=spf1 ip4:192.168.0.1/16 -all\"" ]
  }
}
```

```
}
```

YAML

```
myDNSRecord:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneId: Z3DG6IL3SJCGPX
    Name: mysite.example.com.
    Type: SPF
    TTL: '900'
    ResourceRecords:
      - '"v=spf1 ip4:192.168.0.1/16 -all"'
```

使用 HostedZoneName 新增 RecordSet

此範例使用 HostedZoneName 屬性來指定託管區域，為網域名稱 "mysite.example.com" 新增 Amazon Route 53 資源紀錄集。

JSON

```
"myDNSRecord2" : {
    "Type" : "AWS::Route53::RecordSet",
    "Properties" : {
        "HostedZoneName" : "example.com.",
        "Name" : "mysite.example.com.",
        "Type" : "A",
        "TTL" : "900",
        "ResourceRecords" : [
            "192.168.0.1",
            "192.168.0.2"
        ]
    }
}
```

YAML

```
myDNSRecord2:
  Type: AWS::Route53::RecordSet
  Properties:
    HostedZoneName: example.com.
```

```
Name: mysite.example.com.  
Type: A  
TTL: '900'  
ResourceRecords:  
- 192.168.0.1  
- 192.168.0.2
```

使用 RecordSetGroup 設定加權資源紀錄集

此範例使用 [AWS::Route53::RecordSetGroup](#) 設定 "example.com" 的兩個 CNAME 記錄。託管區域。RecordSets 屬性包含 "mysite.example.com" DNS 名稱的 CNAME 記錄集。每一個記錄集都包含識別碼 (SetIdentifier) 和權重 (Weight)。路由至資源的網際網路流量比例是根據下列計算：

- Frontend One: $140 / (140 + 60) = 140 / 200 = 70\%$
- Frontend Two: $60 / (140 + 60) = 60 / 200 = 30\%$

如需有關加權資源記錄集的詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的 [加權路由](#)。

JSON

```
"myDNSOne" : {  
  "Type" : "AWS::Route53::RecordSetGroup",  
  "Properties" : {  
    "HostedZoneName" : "example.com.",  
    "Comment" : "Weighted RR for my frontends.",  
    "RecordSets" : [  
      {  
        "Name" : "mysite.example.com.",  
        "Type" : "CNAME",  
        "TTL" : "900",  
        "SetIdentifier" : "Frontend One",  
        "Weight" : "140",  
        "ResourceRecords" : ["example-ec2.amazonaws.com"]  
      },  
      {  
        "Name" : "mysite.example.com.",  
        "Type" : "CNAME",  
        "TTL" : "900",  
        "SetIdentifier" : "Frontend Two",  
        "Weight" : "60",  
        "ResourceRecords" : ["example-ec2-larger.amazonaws.com"]  
      }  
    ]  
  }  
}
```

```

    ]
  }
}
```

YAML

```

myDNSOne:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Weighted RR for my frontends.
    RecordSets:
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend One
        Weight: '140'
        ResourceRecords:
          - example-ec2.amazonaws.com
      - Name: mysite.example.com.
        Type: CNAME
        TTL: '900'
        SetIdentifier: Frontend Two
        Weight: '60'
        ResourceRecords:
          - example-ec2-larger.amazonaws.com
```

使用 RecordSetGroup 設定別名資源紀錄集

下列範例使用 [AWS::Route53::RecordSetGroup](#) 來設定名為 example.com 的別名資源紀錄集，以將流量路由至 ELB 第 1 版 負載平衡器 (Classic Load Balancer) 以及第 2 版負載平衡器 (Application Load Balancer 或 Network Load Balancer)。 [AliasTarget](#) 屬性使用 GetAtt 內建函數，以指定 myELB LoadBalancer 的託管區域 ID 和 DNS 名稱。GetAtt 會擷取 myELB 資源的不同屬性，取決於您將流量路由至第 1 版還是第 2 版負載平衡器：

- 第 1 版負載平衡器：CanonicalHostedZoneNameID 和 DNSName
- 第 2 版負載平衡器：CanonicalHostedZoneID 和 DNSName

如需有關別名資源紀錄集的詳細資訊，請參閱《Route 53 開發人員指南》中的 [選擇別名或非別名記錄](#)。

適用於第 1 版負載平衡器的 JSON

```

"myELB" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : [ "us-east-1a" ],
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ]
  }
},
"myDNS" : {
  "Type" : "AWS::Route53::RecordSetGroup",
  "Properties" : {
    "HostedZoneName" : "example.com.",
    "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
    "RecordSets" : [
      {
        "Name" : "example.com.",
        "Type" : "A",
        "AliasTarget" : {
          "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
"CanonicalHostedZoneNameID"] },
          "DNSName" : { "Fn::GetAtt" : ["myELB","DNSName"] }
        }
      }
    ]
  }
}

```

適用於第 1 版負載平衡器的 YAML

```

myELB:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    AvailabilityZones:
      - "us-east-1a"
    Listeners:
      - LoadBalancerPort: '80'
        InstancePort: '80'
        Protocol: HTTP

```

```

myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
      - Name: example.com.
        Type: A
        AliasTarget:
          HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneNameID'
          DNSName: !GetAtt 'myELB.DNSName'

```

適用於第 2 版負載平衡器的 JSON

```

{
  "myELB" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
      "Subnets" : [
        { "Ref": "SubnetAZ1" },
        { "Ref": "SubnetAZ2" }
      ]
    }
  },
  "myDNS" : {
    "Type" : "AWS::Route53::RecordSetGroup",
    "Properties" : {
      "HostedZoneName" : "example.com.",
      "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
      "RecordSets" : [
        {
          "Name" : "example.com.",
          "Type" : "A",
          "AliasTarget" : {
            "HostedZoneId" : { "Fn::GetAtt" : ["myELB",
              "CanonicalHostedZoneID"] },
            "DNSName" : { "Fn::GetAtt" : ["myELB", "DNSName"] }
          }
        }
      ]
    }
  }
}

```


適用於第 2 版負載平衡器的 YAML

```
myELB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Subnets:
      - Ref: SubnetAZ1
      - Ref: SubnetAZ2
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: example.com.
    Comment: Zone apex alias targeted to myELB LoadBalancer.
    RecordSets:
      - Name: example.com.
        Type: A
        AliasTarget:
          HostedZoneId: !GetAtt 'myELB.CanonicalHostedZoneID'
          DNSName: !GetAtt 'myELB.DNSName'
```

CloudFront 分佈的別名資源紀錄集

下列範例建立一個別名 A 記錄，將自訂網域名稱指向現有的 CloudFront 發行版本。myHostedZoneID 預設為參考同一範本中實際的 AWS::Route53::HostedZone 資源，或為一個參數。myCloudFrontDistribution 參考同一範本中的 AWS::CloudFront::Distribution 資源。此別名記錄使用標準的 CloudFront 託管區域 ID (Z2FDTNDATAQYW2)，並透過 Fn::GetAtt 自動解析發行版本的網域名稱。此設定可讓網路流量從自訂網域路由至 CloudFront 發行版本，無需使用 IP 位址。

Note

當您建立別名資源紀錄集時，您必須為 HostedZoneId 屬性指定 Z2FDTNDATAQYW2。CloudFront 的別名資源紀錄集無法在私有區域中建立。

JSON

```
{
  "myDNS": {
    "Type": "AWS::Route53::RecordSetGroup",
```

```

    "Properties": {
      "HostedZoneId": {
        "Ref": "myHostedZoneID"
      },
      "RecordSets": [
        {
          "Name": {
            "Ref": "myRecordSetDomainName"
          },
          "Type": "A",
          "AliasTarget": {
            "HostedZoneId": "Z2FDTNDATAQYW2",
            "DNSName": {
              "Fn::GetAtt": [
                "myCloudFrontDistribution",
                "DomainName"
              ]
            },
            "EvaluateTargetHealth": false
          }
        }
      ]
    }
  }
}

```

YAML

```

myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneId: !Ref myHostedZoneID
    RecordSets:
      - Name: !Ref myRecordSetDomainName
        Type: A
        AliasTarget:
          HostedZoneId: Z2FDTNDATAQYW2
          DNSName: !GetAtt
            - myCloudFrontDistribution
            - DomainName
        EvaluateTargetHealth: false

```

Amazon S3 範本程式碼片段

使用這些 Amazon S3 範例範本，以協助透過 CloudFormation 描述 Amazon S3 儲存貯體。如需更多範例，請參閱 `AWS::S3::Bucket` 中的[範例](#)區段。

主題

- [以預設值建立 Amazon S3 儲存貯體](#)
- [建立用於網站託管並具有 `DeletionPolicy` 的 Amazon S3 儲存貯體](#)
- [使用自訂網域建立靜態網站](#)

以預設值建立 Amazon S3 儲存貯體

此範例使用 `AWS::S3::Bucket` 以預設設定建立儲存貯體。

JSON

```
"myS3Bucket" : {
    "Type" : "AWS::S3::Bucket"
}
```

YAML

```
MyS3Bucket:
  Type: AWS::S3::Bucket
```

建立用於網站託管並具有 **DeletionPolicy** 的 Amazon S3 儲存貯體

本範例會建立一個做為網站的儲存貯體，並停用公開存取封鎖功能 (設定用於網站託管的儲存貯體需具備公開讀取許可)。隨後會向該儲存貯體新增公有儲存貯體政策。由於此儲存貯體資源的 `DeletionPolicy` 屬性設為 `Retain`，因此 CloudFormation 在刪除堆疊時不會刪除此儲存貯體。Output 區段使用 `Fn::GetAtt` 來擷取 `S3Bucket` 資源的 `WebsiteURL` 屬性和 `DomainName` 屬性。

Note

下列範例假設已在帳戶層級停用 `BlockPublicPolicy` 和 `RestrictPublicBuckets` 封鎖公用存取設定。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "S3Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "PublicAccessBlockConfiguration": {
          "BlockPublicAcls": false,
          "BlockPublicPolicy": false,
          "IgnorePublicAcls": false,
          "RestrictPublicBuckets": false
        },
        "WebsiteConfiguration": {
          "IndexDocument": "index.html",
          "ErrorDocument": "error.html"
        }
      },
      "DeletionPolicy": "Retain",
      "UpdateReplacePolicy": "Retain"
    },
    "BucketPolicy": {
      "Type": "AWS::S3::BucketPolicy",
      "Properties": {
        "PolicyDocument": {
          "Id": "MyPolicy",
          "Version": "2012-10-17",
          "Statement": [
            {
              "Sid": "PublicReadForGetBucketObjects",
              "Effect": "Allow",
              "Principal": "*",
              "Action": "s3:GetObject",
              "Resource": {
                "Fn::Join": [
                  "",
                  [
                    "arn:aws:s3:::",
                    {
                      "Ref": "S3Bucket"
                    },
                    "/"
                  ]
                ]
              }
            }
          ]
        }
      }
    }
  }
}
```

```

    ]
  }
}

},
"Bucket": {
  "Ref": "S3Bucket"
}
}
},
"Outputs": {
  "WebsiteURL": {
    "Value": {
      "Fn::GetAtt": [
        "S3Bucket",
        "WebsiteURL"
      ]
    },
    "Description": "URL for website hosted on S3"
  },
  "S3BucketSecureURL": {
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "S3Bucket",
              "DomainName"
            ]
          }
        ]
      ]
    },
    "Description": "Name of S3 bucket to hold website content"
  }
}
}
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: error.html
      DeletionPolicy: Retain
      UpdateReplacePolicy: Retain
  BucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      PolicyDocument:
        Id: MyPolicy
        Version: 2012-10-17
        Statement:
          - Sid: PublicReadForGetBucketObjects
            Effect: Allow
            Principal: '*'
            Action: 's3:GetObject'
            Resource: !Join
              - ''
              - - 'arn:aws:s3:::'
                - !Ref S3Bucket
                - /*
            Bucket: !Ref S3Bucket
Outputs:
  WebsiteURL:
    Value: !GetAtt
      - S3Bucket
      - WebsiteURL
    Description: URL for website hosted on S3
  S3BucketSecureURL:
    Value: !Join
      - ''
      - - 'https://'
```

- !GetAtt
- S3Bucket
- DomainName

Description: Name of S3 bucket to hold website content

使用自訂網域建立靜態網站

您可以將 Route 53 用於已註冊的網域。下列範例假設您已在 Route 53 中為您的網域建立託管區域。此範例會建立兩個用於網站託管的儲存貯體。根儲存貯體會託管內容，而另一個儲存貯體則會將 `www.domainname.com` 請求重新導向至根儲存貯體。記錄集會將您的網域名稱映射至 Amazon S3 端點。

您也需要新增儲存貯體政策，如上述範例所示。

有關使用自訂網域的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[教學課程：使用在 Route 53 中註冊的自訂網域設定靜態網站](#)。

Note

下列範例假設已在帳戶層級停用 BlockPublicPolicy 和 RestrictPublicBuckets 封鎖公用存取設定。

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "S3hostedzoneID" : "Z3AQBSTGFYJSTF", "websiteendpoint" :
"s3-website-us-east-1.amazonaws.com" },
      "us-west-1" : { "S3hostedzoneID" : "Z2F56UZL2M1ACD", "websiteendpoint" :
"s3-website-us-west-1.amazonaws.com" },
      "us-west-2" : { "S3hostedzoneID" : "Z3BJ6K6RIION7M", "websiteendpoint" :
"s3-website-us-west-2.amazonaws.com" },
      "eu-west-1" : { "S3hostedzoneID" : "Z1BKCTXD74EZPE", "websiteendpoint" :
"s3-website-eu-west-1.amazonaws.com" },
      "ap-southeast-1" : { "S3hostedzoneID" : "Z300J2DXBE1FTB",
"websiteendpoint" : "s3-website-ap-southeast-1.amazonaws.com" },
      "ap-southeast-2" : { "S3hostedzoneID" : "Z1WCIGYICN2BYD",
"websiteendpoint" : "s3-website-ap-southeast-2.amazonaws.com" },

```

```

        "ap-northeast-1" : { "S3hostedzoneID" : "Z2M4EHUR26P7ZW",
"websiteendpoint" : "s3-website-ap-northeast-1.amazonaws.com" },
        "sa-east-1" : { "S3hostedzoneID" : "Z31GFT0UA1I2HV", "websiteendpoint" :
"s3-website-sa-east-1.amazonaws.com" }
    }
},
"Parameters": {
    "RootDomainName": {
        "Description": "Domain name for your website (example.com)",
        "Type": "String"
    }
},
"Resources": {
    "RootBucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "BucketName" : {"Ref":"RootDomainName"},
            "PublicAccessBlockConfiguration": {
                "BlockPublicAcls": false,
                "BlockPublicPolicy": false,
                "IgnorePublicAcls": false,
                "RestrictPublicBuckets": false
            },
            "WebsiteConfiguration": {
                "IndexDocument":"index.html",
                "ErrorDocument":"404.html"
            }
        }
    },
    "WWWBucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "BucketName": {
                "Fn::Join": [ "", [ "www.", {"Ref":"RootDomainName"} ] ]
            },
            "AccessControl": "BucketOwnerFullControl",
            "WebsiteConfiguration": {
                "RedirectAllRequestsTo": {
                    "HostName": {"Ref": "RootBucket"}
                }
            }
        }
    },
    "myDNS": {

```



```

    "Type": "AWS::Route53::RecordSetGroup",
    "Properties": {
      "HostedZoneName": {
        "Fn::Join": ["", [{"Ref": "RootDomainName"}, "."]]
      },
      "Comment": "Zone apex alias.",
      "RecordSets": [
        {
          "Name": {"Ref": "RootDomainName"},
          "Type": "A",
          "AliasTarget": {
            "HostedZoneId": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "S3hostedzoneID"]},
            "DNSName": {"Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "websiteendpoint"]}
          }
        },
        {
          "Name": {
            "Fn::Join": ["", [{"Ref": "RootDomainName"}, "www."]]
          },
          "Type": "CNAME",
          "TTL" : "900",
          "ResourceRecords" : [
            {"Fn::GetAtt":["WWWBucket", "DomainName"]}
          ]
        }
      ]
    }
  },
  "Outputs": {
    "WebsiteURL": {
      "Value": {"Fn::GetAtt": ["RootBucket", "WebsiteURL"]},
      "Description": "URL for website hosted on S3"
    }
  }
}

```

YAML

```

Parameters:
  RootDomainName:

```

```
    Description: Domain name for your website (example.com)
    Type: String
Mappings:
  RegionMap:
    us-east-1:
      S3hostedzoneID: Z3AQBSTGFYJSTF
      websiteendpoint: s3-website-us-east-1.amazonaws.com
    us-west-1:
      S3hostedzoneID: Z2F56UZL2M1ACD
      websiteendpoint: s3-website-us-west-1.amazonaws.com
    us-west-2:
      S3hostedzoneID: Z3BJ6K6RII0N7M
      websiteendpoint: s3-website-us-west-2.amazonaws.com
    eu-west-1:
      S3hostedzoneID: Z1BKCTXD74EZPE
      websiteendpoint: s3-website-eu-west-1.amazonaws.com
    ap-southeast-1:
      S3hostedzoneID: Z300J2DXBE1FTB
      websiteendpoint: s3-website-ap-southeast-1.amazonaws.com
    ap-southeast-2:
      S3hostedzoneID: Z1WCIGYICN2BYD
      websiteendpoint: s3-website-ap-southeast-2.amazonaws.com
    ap-northeast-1:
      S3hostedzoneID: Z2M4EHUR26P7ZW
      websiteendpoint: s3-website-ap-northeast-1.amazonaws.com
    sa-east-1:
      S3hostedzoneID: Z31GFT0UA1I2HV
      websiteendpoint: s3-website-sa-east-1.amazonaws.com
Resources:
  RootBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref RootDomainName
      PublicAccessBlockConfiguration:
        BlockPublicAcls: false
        BlockPublicPolicy: false
        IgnorePublicAcls: false
        RestrictPublicBuckets: false
      WebsiteConfiguration:
        IndexDocument: index.html
        ErrorDocument: 404.html
  WWWBucket:
    Type: AWS::S3::Bucket
    Properties:
```

```

    BucketName: !Sub
      - www.${Domain}
      - Domain: !Ref RootDomainName
    AccessControl: BucketOwnerFullControl
    WebsiteConfiguration:
      RedirectAllRequestsTo:
        HostName: !Ref RootBucket
myDNS:
  Type: AWS::Route53::RecordSetGroup
  Properties:
    HostedZoneName: !Sub
      - ${Domain}.
      - Domain: !Ref RootDomainName
    Comment: Zone apex alias.
    RecordSets:
      - Name: !Ref RootDomainName
        Type: A
        AliasTarget:
          HostedZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', S3hostedzoneID]
          DNSName: !FindInMap [ RegionMap, !Ref 'AWS::Region', websiteendpoint]
      - Name: !Sub
          - www.${Domain}
          - Domain: !Ref RootDomainName
        Type: CNAME
        TTL: 900
        ResourceRecords:
          - !GetAtt WWWBucket.DomainName
  Outputs:
    WebsiteURL:
      Value: !GetAtt RootBucket.WebsiteURL
      Description: URL for website hosted on S3

```

Amazon SNS 範本程式碼片段

此範例顯示 Amazon SNS 主題資源。它需要一個有效的電子郵件地址。

JSON

```

"MySNSTopic" : {
  "Type" : "AWS::SNS::Topic",
  "Properties" : {
    "Subscription" : [ {
      "Endpoint" : "add valid email address",

```

```
        "Protocol" : "email"
      } ]
    }
  }
```

YAML

```
MySNSTopic:
  Type: AWS::SNS::Topic
  Properties:
    Subscription:
      - Endpoint: "add valid email address"
        Protocol: email
```

Amazon SQS 範本程式碼片段

此範例顯示 Amazon SQS 佇列。

JSON

```
"MyQueue" : {
  "Type" : "AWS::SQS::Queue",
  "Properties" : {
    "VisibilityTimeout" : "value"
  }
}
```

YAML

```
MyQueue:
  Type: AWS::SQS::Queue
  Properties:
    VisibilityTimeout: value
```

Amazon Timestream 範本程式碼片段

Amazon Timestream for InfluxDB 可讓應用程式開發人員和 DevOps 團隊輕鬆地在 上執行全受管 InfluxDB 資料庫，AWS 以便使用開放原始碼 APIs 進行即時時間序列應用程式。您可以快速建立能夠處理高要求時間序列工作負載的 InfluxDB 資料庫。只要幾個簡單的 API 呼叫，您就可以 AWS 透過自動化軟體修補、備份和復原，在 上設定、遷移、操作和擴展 InfluxDB 資料庫。您也能在 GitHub 上的

awslabs/amazon-timestream-tools/tree/mainline/integrations/cloudformation/timestream-influxdb 找到這些範例。

主題

- [使用預設值的最小範例](#)
- [包含參數的更完整範例](#)

這些 CloudFormation 範本會建立下列資源，以便成功建立、連線和監控 Amazon Timestream for InfluxDB 執行個體：

Amazon VPC

- VPC
- 一個或多個 Subnet
- InternetGateway
- RouteTable
- SecurityGroup

Amazon S3

- Bucket

Amazon Timestream

- InfluxDBInstance

使用預設值的最小範例

此範例在可能的情況下使用預設值，部署多可用區且可公開存取的執行個體。

JSON

```
{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
```

```

        "Parameters": [
            "DbInstanceName",
            "InfluxDBPassword"
        ]
    },
    "ParameterLabels": {
        "VPCCIDR": {"default": "VPC CIDR"}
    }
},
"Parameters": {
    "DbInstanceName": {
        "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
        "Type": "String",
        "Default": "mydbinstance",
        "MinLength": 3,
        "MaxLength": 40,
        "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
    },
    "InfluxDBPassword": {
        "Description": "The password of the initial admin user created in InfluxDB. This
password will allow you to access the InfluxDB UI to perform various administrative
tasks and also use the InfluxDB CLI to create an operator token. These attributes will
be stored in a Secret created in AWS Secrets Manager in your account.",
        "Type": "String",
        "NoEcho": true,
        "MinLength": 8,
        "MaxLength": 64,
        "AllowedPattern": "^[a-zA-Z0-9]+$"
    }
},
"Resources": {
    "VPC": {
        "Type": "AWS::EC2::VPC",
        "Properties": {"CidrBlock": "10.0.0.0/16"}
    },
    "InternetGateway": {"Type": "AWS::EC2::InternetGateway"},
    "InternetGatewayAttachment": {
        "Type": "AWS::EC2::VPCGatewayAttachment",
        "Properties": {

```

```
    "InternetGatewayId": {"Ref": "InternetGateway"},
    "VpcId": {"Ref": "VPC"}
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          {"Fn::GetAZs": ""}
        ]
      },
      "CidrBlock": {
        "Fn::Select": [
          0,
          {
            "Fn::Cidr": [
              {
                "Fn::GetAtt": [
                  "VPC",
                  "CidrBlock"
                ]
              },
              2,
              12
            ]
          }
        ]
      },
      "MapPublicIpOnLaunch": true
    }
  },
  "Subnet2": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          1,
          {"Fn::GetAZs": ""}
        ]
      }
    }
  },
```

```

    "CidrBlock": {
      "Fn::Select": [
        1,
        {
          "Fn::Cidr": [
            {
              "Fn::GetAtt": [
                "VPC",
                "CidrBlock"
              ]
            },
            2,
            12
          ]
        }
      ],
      "MapPublicIpOnLaunch": true
    },
    "RouteTable": {
      "Type": "AWS::EC2::RouteTable",
      "Properties": {
        "VpcId": {"Ref": "VPC"}
      }
    },
    "DefaultRoute": {
      "Type": "AWS::EC2::Route",
      "DependsOn": "InternetGatewayAttachment",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
      }
    },
    "Subnet1RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
      }
    },
    "Subnet2RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",

```



```

    "Properties": {
      "RouteTableId": {"Ref": "RouteTable"},
      "SubnetId": {"Ref": "Subnet2"}
    }
  },
  "InfluxDBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupName": "influxdb-sg",
      "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
      "VpcId": {"Ref": "VPC"}
    }
  },
  "InfluxDBSecurityGroupIngress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupId": {"Ref": "InfluxDBSecurityGroup"},
      "IpProtocol": "tcp",
      "CidrIp": "0.0.0.0/0",
      "FromPort": 8086,
      "ToPort": 8086
    }
  },
  "InfluxDBLogsS3Bucket": {
    "Type": "AWS::S3::Bucket",
    "DeletionPolicy": "Retain"
  },
  "InfluxDBLogsS3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/"
            *"},
            "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
          }
        ],
        {
          "Action": "s3:*",
          "Effect": "Deny",

```

```

        "Resource": [
            {
                "Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*",
                "Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"
            },
            {
                "Principal": "*",
                "Condition": {
                    "Bool": {"aws:SecureTransport": false}
                }
            }
        ],
    },
    "DbInstance": {
        "Type": "AWS::Timestream::InfluxDBInstance",
        "DependsOn": "InfluxDBLogsS3BucketPolicy",
        "Properties": {
            "AllocatedStorage": 20,
            "DbInstanceType": "db.influx.medium",
            "Name": {"Ref": "DbInstanceName"},
            "Password": {"Ref": "InfluxDBPassword"},
            "PubliclyAccessible": true,
            "DeploymentType": "WITH_MULTIAZ_STANDBY",
            "VpcSecurityGroupIds": [
                {"Ref": "InfluxDBSecurityGroup"}
            ],
            "VpcSubnetIds": [
                {"Ref": "Subnet1"},
                {"Ref": "Subnet2"}
            ],
            "LogDeliveryConfiguration": {
                "S3Configuration": {
                    "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
                    "Enabled": true
                }
            }
        }
    },
    "Outputs": {
        "VPC": {
            "Description": "A reference to the VPC used to create network resources",
            "Value": {"Ref": "VPC"}
        }
    },

```

```

"Subnets": {
  "Description": "A list of the subnets created",
  "Value": {
    "Fn::Join": [
      ",",
      [
        {"Ref": "Subnet1"},
        {"Ref": "Subnet2"}
      ]
    ]
  }
},
"Subnet1": {
  "Description": "A reference to the subnet in the 1st Availability Zone",
  "Value": {"Ref": "Subnet1"}
},
"Subnet2": {
  "Description": "A reference to the subnet in the 2nd Availability Zone",
  "Value": {"Ref": "Subnet2"}
},
"InfluxDBSecurityGroup": {
  "Description": "Security group with port 8086 ingress rule",
  "Value": {"Ref": "InfluxDBSecurityGroup"}
},
"InfluxDBLogsS3Bucket": {
  "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
  "Value": {"Ref": "InfluxDBLogsS3Bucket"}
},
"DbInstance": {
  "Description": "A reference to the Timestream for InfluxDB DB instance",
  "Value": {"Ref": "DbInstance"}
},
"InfluxAuthParametersSecretArn": {
  "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
  "Value": {
    "Fn::GetAtt": [
      "DbInstance",
      "InfluxAuthParametersSecretArn"
    ]
  }
},

```

```

"Endpoint": {
  "Description": "The endpoint URL to connect to InfluxDB",
  "Value": {
    "Fn::Join": [
      "",
      [
        "https://",
        {
          "Fn::GetAtt": [
            "DbInstance",
            "Endpoint"
          ]
        },
        ":8086"
      ]
    ]
  }
}
}
}
}

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBPassword
    ParameterLabels:
      VPCCIDR:
        default: VPC CIDR

Parameters:
  DbInstanceName:
    Description: The name that uniquely identifies the DB instance when interacting
with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a
prefix included in the endpoint. DB instance names must be unique per customer and per
Region.
    Type: String

```

Default: mydbinstance

MinLength: 3

MaxLength: 40

AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*\$

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

Type: String

NoEcho: true

MinLength: 8

MaxLength: 64

AllowedPattern: ^[a-zA-Z0-9]+\$

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/16

InternetGateway:

Type: AWS::EC2::InternetGateway

InternetGatewayAttachment:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref VPC

Subnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [0, !GetAZs '']

CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12]]

MapPublicIpOnLaunch: true

Subnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [1, !GetAZs '']

CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12]]

MapPublicIpOnLaunch: true

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

```
VpcId: !Ref VPC
DefaultRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
Subnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !Ref InfluxDBSecurityGroup
    IpProtocol: tcp
    CidrIp: 0.0.0.0/0
    FromPort: 8086
    ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
```

```

    Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
    Principal:
      Service: timestream-influxdb.amazonaws.com
  - Action: "s3:*"
    Effect: Deny
    Resource:
      - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
      - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
    Principal: "*"
    Condition:
      Bool:
        aws:SecureTransport: false

```

DbInstance:

```

Type: AWS::Timestream::InfluxDBInstance
DependsOn: InfluxDBLogsS3BucketPolicy
Properties:
  AllocatedStorage: 20
  DbInstanceType: db.influx.medium
  Name: !Ref DbInstanceName
  Password: !Ref InfluxDBPassword
  PubliclyAccessible: true
  DeploymentType: WITH_MULTIAZ_STANDBY
  VpcSecurityGroupIds:
    - !Ref InfluxDBSecurityGroup
  VpcSubnetIds:
    - !Ref Subnet1
    - !Ref Subnet2
  LogDeliveryConfiguration:
    S3Configuration:
      BucketName: !Ref InfluxDBLogsS3Bucket
      Enabled: true

```

Outputs:

```
# Network Resources
```

VPC:

```

Description: A reference to the VPC used to create network resources
Value: !Ref VPC

```

Subnets:

```

Description: A list of the subnets created
Value: !Join ["", [!Ref Subnet1, !Ref Subnet2]]

```

Subnet1:

```

Description: A reference to the subnet in the 1st Availability Zone
Value: !Ref Subnet1

```

Subnet2:

```

    Description: A reference to the subnet in the 2nd Availability Zone
    Value: !Ref Subnet2
InfluxDBSecurityGroup:
    Description: Security group with port 8086 ingress rule
    Value: !Ref InfluxDBSecurityGroup

# Timestream for InfluxDB Resources
InfluxDBLogsS3Bucket:
    Description: S3 Bucket containing InfluxDB logs from the DB instance
    Value: !Ref InfluxDBLogsS3Bucket
DbInstance:
    Description: A reference to the Timestream for InfluxDB DB instance
    Value: !Ref DbInstance
InfluxAuthParametersSecretArn:
    Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
    containing the initial InfluxDB authorization parameters. The secret value is a JSON
    formatted key-value pair holding InfluxDB authorization values: organization, bucket,
    username, and password."
    Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
Endpoint:
    Description: The endpoint URL to connect to InfluxDB
    Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]

```

包含參數的更完整範例

此範例範本會根據提供的參數動態調整網路資源。參數包括 `PubliclyAccessible` 與 `DeploymentType`。

JSON

```

{
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterGroups": [
        {
          "Label": {"default": "Network Configuration"},
          "Parameters": ["VPCCIDR"]
        },
        {
          "Label": {"default": "Amazon Timestream for InfluxDB Configuration"},
          "Parameters": [
            "DbInstanceName",

```



```

        "InfluxDBUsername",
        "InfluxDBPassword",
        "InfluxDBOrganization",
        "InfluxDBBucket",
        "DbInstanceType",
        "DbStorageType",
        "AllocatedStorage",
        "PubliclyAccessible",
        "DeploymentType"
    ]
}
],
"ParameterLabels": {
    "VPCCIDR": {"default": "VPC CIDR"}
}
},
"Parameters": {
    "VPCCIDR": {
        "Description": "Please enter the IP range (CIDR notation) for the new VPC",
        "Type": "String",
        "Default": "10.0.0.0/16"
    },
    "DbInstanceName": {
        "Description": "The name that uniquely identifies the DB instance when
interacting with the Amazon Timestream for InfluxDB API and CLI commands. This name
will also be a prefix included in the endpoint. DB instance names must be unique per
customer and per Region.",
        "Type": "String",
        "Default": "mydbinstance",
        "MinLength": 3,
        "MaxLength": 40,
        "AllowedPattern": "^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$"
    },
    "InfluxDBUsername": {
        "Description": "The username of the initial admin user created in InfluxDB. Must
start with a letter and can't end with a hyphen or contain two consecutive hyphens.
For example, my-user1. This username will allow you to access the InfluxDB UI to
perform various administrative tasks and also use the InfluxDB CLI to create an
operator token. These attributes will be stored in a Secret created in AWS Secrets
Manager in your account.",
        "Type": "String",
        "Default": "admin",
        "MinLength": 1,

```

```

    "MaxLength": 64
  },
  "InfluxDBPassword": {
    "Description": "The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.",
    "Type": "String",
    "NoEcho": true,
    "MinLength": 8,
    "MaxLength": 64,
    "AllowedPattern": "^[a-zA-Z0-9]+$"
  },
  "InfluxDBOrganization": {
    "Description": "The name of the initial organization for the initial admin user in InfluxDB. An InfluxDB organization is a workspace for a group of users.",
    "Type": "String",
    "Default": "org",
    "MinLength": 1,
    "MaxLength": 64
  },
  "InfluxDBBucket": {
    "Description": "The name of the initial InfluxDB bucket. All InfluxDB data is stored in a bucket. A bucket combines the concept of a database and a retention period (the duration of time that each data point persists). A bucket belongs to an organization.",
    "Type": "String",
    "Default": "bucket",
    "MinLength": 2,
    "MaxLength": 64,
    "AllowedPattern": "^[^_\\\\""][^\\\\""]*$"
  },
  "DeploymentType": {
    "Description": "Specifies whether the Timestream for InfluxDB is deployed as Single-AZ or with a MultiAZ Standby for High availability",
    "Type": "String",
    "Default": "WITH_MULTIAZ_STANDBY",
    "AllowedValues": [
      "SINGLE_AZ",
      "WITH_MULTIAZ_STANDBY"
    ]
  },
  "AllocatedStorage": {

```

```

    "Description": "The amount of storage to allocate for your DB storage type in GiB
(gibibytes).",
    "Type": "Number",
    "Default": 400,
    "MinValue": 20,
    "MaxValue": 16384
  },
  "DbInstanceType": {
    "Description": "The Timestream for InfluxDB DB instance type to run InfluxDB
on.",
    "Type": "String",
    "Default": "db.influx.medium",
    "AllowedValues": [
      "db.influx.medium",
      "db.influx.large",
      "db.influx.xlarge",
      "db.influx.2xlarge",
      "db.influx.4xlarge",
      "db.influx.8xlarge",
      "db.influx.12xlarge",
      "db.influx.16xlarge"
    ]
  },
  "DbStorageType": {
    "Description": "The Timestream for InfluxDB DB storage type to read and write
InfluxDB data.",
    "Type": "String",
    "Default": "InfluxIOIncludedT1",
    "AllowedValues": [
      "InfluxIOIncludedT1",
      "InfluxIOIncludedT2",
      "InfluxIOIncludedT3"
    ]
  },
  "PubliclyAccessible": {
    "Description": "Configures the DB instance with a public IP to facilitate
access.",
    "Type": "String",
    "Default": true,
    "AllowedValues": [
      true,
      false
    ]
  }
}

```

```

},
"Conditions": {
  "IsMultiAZ": {
    "Fn::Equals": [
      {"Ref": "DeploymentType"},
      "WITH_MULTIAZ_STANDBY"
    ]
  },
  "IsPublic": {
    "Fn::Equals": [
      {"Ref": "PubliclyAccessible"},
      true
    ]
  }
},
"Resources": {
  "VPC": {
    "Type": "AWS::EC2::VPC",
    "Properties": {
      "CidrBlock": {"Ref": "VPCCIDR"}
    }
  },
  "InternetGateway": {
    "Type": "AWS::EC2::InternetGateway",
    "Condition": "IsPublic"
  },
  "InternetGatewayAttachment": {
    "Type": "AWS::EC2::VPCGatewayAttachment",
    "Condition": "IsPublic",
    "Properties": {
      "InternetGatewayId": {"Ref": "InternetGateway"},
      "VpcId": {"Ref": "VPC"}
    }
  },
  "Subnet1": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          {"Fn::GetAZs": ""}
        ]
      }
    }
  },
},

```

```

    "CidrBlock": {
      "Fn::Select": [
        0,
        {
          "Fn::Cidr": [
            {
              "Fn::GetAtt": [
                "VPC",
                "CidrBlock"
              ]
            },
            2,
            12
          ]
        }
      ],
    },
    "MapPublicIpOnLaunch": {
      "Fn::If": [
        "IsPublic",
        true,
        false
      ]
    }
  ],
  "Subnet2": {
    "Type": "AWS::EC2::Subnet",
    "Condition": "IsMultiAZ",
    "Properties": {
      "VpcId": {"Ref": "VPC"},
      "AvailabilityZone": {
        "Fn::Select": [
          1,
          {"Fn::GetAZs": ""}
        ]
      },
    },
    "CidrBlock": {
      "Fn::Select": [
        1,
        {
          "Fn::Cidr": [
            {
              "Fn::GetAtt": [

```

```

        "VPC",
        "CidrBlock"
    ]
},
2,
12
]
}
]
},
"MapPublicIpOnLaunch": {
    "Fn::If": [
        "IsPublic",
        true,
        false
    ]
}
},
"RouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {"Ref": "VPC"}
    }
},
"DefaultRoute": {
    "Type": "AWS::EC2::Route",
    "Condition": "IsPublic",
    "DependsOn": "InternetGatewayAttachment",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {"Ref": "InternetGateway"}
    }
},
"Subnet1RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
        "RouteTableId": {"Ref": "RouteTable"},
        "SubnetId": {"Ref": "Subnet1"}
    }
},
"Subnet2RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",

```

```

    "Condition": "IsMultiAZ",
    "Properties": {
      "RouteTableId": {"Ref": "RouteTable"},
      "SubnetId": {"Ref": "Subnet2"}
    }
  },
  "InfluxDBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupName": "influxdb-sg",
      "GroupDescription": "Security group allowing port 8086 ingress for InfluxDB",
      "VpcId": {"Ref": "VPC"}
    }
  },
  "InfluxDBSecurityGroupIngress": {
    "Type": "AWS::EC2::SecurityGroupIngress",
    "Properties": {
      "GroupId": {"Ref": "InfluxDBSecurityGroup"},
      "IpProtocol": "tcp",
      "CidrIp": "0.0.0.0/0",
      "FromPort": 8086,
      "ToPort": 8086
    }
  },
  "InfluxDBLogsS3Bucket": {
    "Type": "AWS::S3::Bucket",
    "DeletionPolicy": "Retain"
  },
  "InfluxDBLogsS3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {"Ref": "InfluxDBLogsS3Bucket"},
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/
*"}},
            "Principal": {"Service": "timestream-influxdb.amazonaws.com"}
          },
          {
            "Action": "s3:*",

```

```

        "Effect": "Deny",
        "Resource": [
            {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}/*"},
            {"Fn::Sub": "arn:aws:s3:::${InfluxDBLogsS3Bucket}"}
        ],
        "Principal": "*",
        "Condition": {
            "Bool": {"aws:SecureTransport": false}
        }
    }
]
}
},
"DbInstance": {
    "Type": "AWS::Timestream::InfluxDBInstance",
    "DependsOn": "InfluxDBLogsS3BucketPolicy",
    "Properties": {
        "DbStorageType": {"Ref": "DbStorageType"},
        "AllocatedStorage": {"Ref": "AllocatedStorage"},
        "DbInstanceType": {"Ref": "DbInstanceType"},
        "Name": {"Ref": "DbInstanceName"},
        "Username": {"Ref": "InfluxDBUsername"},
        "Password": {"Ref": "InfluxDBPassword"},
        "Organization": {"Ref": "InfluxDBOrganization"},
        "Bucket": {"Ref": "InfluxDBBucket"},
        "PubliclyAccessible": {
            "Fn::If": [
                "IsPublic",
                true,
                false
            ]
        },
    },
    "DeploymentType": {"Ref": "DeploymentType"},
    "VpcSecurityGroupIds": [
        {"Ref": "InfluxDBSecurityGroup"}
    ],
    "VpcSubnetIds": {
        "Fn::If": [
            "IsMultiAZ",
            [
                {"Ref": "Subnet1"},
                {"Ref": "Subnet2"}
            ]
        },
    },

```



```

        [
            {"Ref": "Subnet1"}
        ]
    ],
    "LogDeliveryConfiguration": {
        "S3Configuration": {
            "BucketName": {"Ref": "InfluxDBLogsS3Bucket"},
            "Enabled": true
        }
    }
}
},
"Outputs": {
    "VPC": {
        "Description": "A reference to the VPC used to create network resources",
        "Value": {"Ref": "VPC"}
    },
    "Subnets": {
        "Description": "A list of the subnets created",
        "Value": {
            "Fn::If": [
                "IsMultiAZ",
                {
                    "Fn::Join": [
                        ",",
                        [
                            {"Ref": "Subnet1"},
                            {"Ref": "Subnet2"}
                        ]
                    ]
                },
                {"Ref": "Subnet1"}
            ]
        }
    },
    "Subnet1": {
        "Description": "A reference to the subnet in the 1st Availability Zone",
        "Value": {"Ref": "Subnet1"}
    },
    "Subnet2": {
        "Condition": "IsMultiAZ",
        "Description": "A reference to the subnet in the 2nd Availability Zone",

```

```

    "Value": {"Ref": "Subnet2"}
  },
  "InfluxDBSecurityGroup": {
    "Description": "Security group with port 8086 ingress rule",
    "Value": {"Ref": "InfluxDBSecurityGroup"}
  },
  "InfluxDBLogsS3Bucket": {
    "Description": "S3 Bucket containing InfluxDB logs from the DB instance",
    "Value": {"Ref": "InfluxDBLogsS3Bucket"}
  },
  "DbInstance": {
    "Description": "A reference to the Timestream for InfluxDB DB instance",
    "Value": {"Ref": "DbInstance"}
  },
  "InfluxAuthParametersSecretArn": {
    "Description": "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password.",
    "Value": {
      "Fn::GetAtt": [
        "DbInstance",
        "InfluxAuthParametersSecretArn"
      ]
    }
  },
  "Endpoint": {
    "Description": "The endpoint URL to connect to InfluxDB",
    "Value": {
      "Fn::Join": [
        "",
        [
          "https://",
          {
            "Fn::GetAtt": [
              "DbInstance",
              "Endpoint"
            ]
          },
          ":8086"
        ]
      ]
    }
  }
}

```

```

    }
  }

```

YAML

```

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      -
        Label:
          default: "Network Configuration"
        Parameters:
          - VPCCIDR
      -
        Label:
          default: "Amazon Timestream for InfluxDB Configuration"
        Parameters:
          - DbInstanceName
          - InfluxDBUsername
          - InfluxDBPassword
          - InfluxDBOrganization
          - InfluxDBBucket
          - DbInstanceType
          - DbStorageType
          - AllocatedStorage
          - PubliclyAccessible
          - DeploymentType
    ParameterLabels:
      VPCCIDR:
        default: VPC CIDR

Parameters:
  # Network Configuration
  VPCCIDR:
    Description: Please enter the IP range (CIDR notation) for the new VPC
    Type: String
    Default: 10.0.0.0/16
  # Timestream for InfluxDB Configuration
  DbInstanceName:
    Description: The name that uniquely identifies the DB instance when interacting
    with the Amazon Timestream for InfluxDB API and CLI commands. This name will also be a
    prefix included in the endpoint. DB instance names must be unique per customer and per
    Region.

```

```
Type: String
Default: mydbinstance
MinLength: 3
MaxLength: 40
AllowedPattern: ^[a-zA-z][a-zA-Z0-9]*(-[a-zA-Z0-9]+)*$
```

InfluxDB initial user configurations

InfluxDBUsername:

Description: The username of the initial admin user created in InfluxDB. Must start with a letter and can't end with a hyphen or contain two consecutive hyphens. For example, my-user1. This username will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS Secrets Manager in your account.

```
Type: String
Default: admin
MinLength: 1
MaxLength: 64
```

InfluxDBPassword:

Description: The password of the initial admin user created in InfluxDB. This password will allow you to access the InfluxDB UI to perform various administrative tasks and also use the InfluxDB CLI to create an operator token. These attributes will be stored in a Secret created in AWS in your account.

```
Type: String
NoEcho: true
MinLength: 8
MaxLength: 64
AllowedPattern: ^[a-zA-Z0-9]+$
```

InfluxDBOrganization:

Description: The name of the initial organization for the initial admin user in InfluxDB. An InfluxDB organization is a workspace for a group of users.

```
Type: String
Default: org
MinLength: 1
MaxLength: 64
```

InfluxDBBucket:

Description: The name of the initial InfluxDB bucket. All InfluxDB data is stored in a bucket. A bucket combines the concept of a database and a retention period (the duration of time that each data point persists). A bucket belongs to an organization.

```
Type: String
Default: bucket
MinLength: 2
MaxLength: 64
AllowedPattern: ^[^\_"][^\"]*$
```

DeploymentType:

Description: Specifies whether the Timestream for InfluxDB is deployed as Single-AZ or with a MultiAZ Standby for High availability

Type: String

Default: WITH_MULTIAZ_STANDBY

AllowedValues:

- SINGLE_AZ
- WITH_MULTIAZ_STANDBY

AllocatedStorage:

Description: The amount of storage to allocate for your DB storage type in GiB (gibibytes).

Type: Number

Default: 400

MinValue: 20

MaxValue: 16384

DbInstanceType:

Description: The Timestream for InfluxDB DB instance type to run InfluxDB on.

Type: String

Default: db.influx.medium

AllowedValues:

- db.influx.medium
- db.influx.large
- db.influx.xlarge
- db.influx.2xlarge
- db.influx.4xlarge
- db.influx.8xlarge
- db.influx.12xlarge
- db.influx.16xlarge

DbStorageType:

Description: The Timestream for InfluxDB DB storage type to read and write InfluxDB data.

Type: String

Default: InfluxIOIncludedT1

AllowedValues:

- InfluxIOIncludedT1
- InfluxIOIncludedT2
- InfluxIOIncludedT3

PubliclyAccessible:

Description: Configures the DB instance with a public IP to facilitate access.

Type: String

Default: true

AllowedValues:

- true
- false

Conditions:

```
IsMultiAZ: !Equals [!Ref DeploymentType, WITH_MULTIAZ_STANDBY]
IsPublic: !Equals [!Ref PubliclyAccessible, true]
```

Resources:**VPC:**

```
Type: AWS::EC2::VPC
Properties:
  CidrBlock: !Ref VPCCIDR
```

InternetGateway:

```
Type: AWS::EC2::InternetGateway
Condition: IsPublic
```

InternetGatewayAttachment:

```
Type: AWS::EC2::VPCGatewayAttachment
Condition: IsPublic
Properties:
  InternetGatewayId: !Ref InternetGateway
  VpcId: !Ref VPC
```

Subnet1:

```
Type: AWS::EC2::Subnet
Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [0, !GetAZs '']
  CidrBlock: !Select [0, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
  MapPublicIpOnLaunch: !If [IsPublic, true, false]
```

Subnet2:

```
Type: AWS::EC2::Subnet
Condition: IsMultiAZ
Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [1, !GetAZs '']
  CidrBlock: !Select [1, !Cidr [!GetAtt VPC.CidrBlock, 2, 12 ]]
  MapPublicIpOnLaunch: !If [IsPublic, true, false]
```

RouteTable:

```
Type: AWS::EC2::RouteTable
Properties:
  VpcId: !Ref VPC
```

DefaultRoute:

```
Type: AWS::EC2::Route
Condition: IsPublic
DependsOn: InternetGatewayAttachment
Properties:
  RouteTableId: !Ref RouteTable
  DestinationCidrBlock: 0.0.0.0/0
```

```
    GatewayId: !Ref InternetGateway
Subnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet1
Subnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Condition: IsMultiAZ
  Properties:
    RouteTableId: !Ref RouteTable
    SubnetId: !Ref Subnet2
InfluxDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: "influxdb-sg"
    GroupDescription: "Security group allowing port 8086 ingress for InfluxDB"
    VpcId: !Ref VPC
InfluxDBSecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !Ref InfluxDBSecurityGroup
    IpProtocol: tcp
    CidrIp: 0.0.0.0/0
    FromPort: 8086
    ToPort: 8086
InfluxDBLogsS3Bucket:
  Type: AWS::S3::Bucket
  DeletionPolicy: Retain
InfluxDBLogsS3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref InfluxDBLogsS3Bucket
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Action: "s3:PutObject"
          Effect: Allow
          Resource: !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/InfluxLogs/*
          Principal:
            Service: timestream-influxdb.amazonaws.com
        - Action: "s3:*"
          Effect: Deny
          Resource:
```

```

        - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}/*
        - !Sub arn:aws:s3:::${InfluxDBLogsS3Bucket}
    Principal: "*"
    Condition:
        Bool:
            aws:SecureTransport: false
DbInstance:
    Type: AWS::Timestream::InfluxDBInstance
    DependsOn: InfluxDBLogsS3BucketPolicy
    Properties:
        DbStorageType: !Ref DbStorageType
        AllocatedStorage: !Ref AllocatedStorage
        DbInstanceType: !Ref DbInstanceType
        Name: !Ref DbInstanceName
        Username: !Ref InfluxDBUsername
        Password: !Ref InfluxDBPassword
        Organization: !Ref InfluxDBOrganization
        Bucket: !Ref InfluxDBBucket
        PubliclyAccessible: !If [IsPublic, true, false]
        DeploymentType: !Ref DeploymentType
        VpcSecurityGroupIds:
            - !Ref InfluxDBSecurityGroup
        VpcSubnetIds: !If
            - IsMultiAZ
            -
                - !Ref Subnet1
                - !Ref Subnet2
            -
                - !Ref Subnet1
        LogDeliveryConfiguration:
            S3Configuration:
                BucketName: !Ref InfluxDBLogsS3Bucket
                Enabled: true

Outputs:
    # Network Resources
    VPC:
        Description: A reference to the VPC used to create network resources
        Value: !Ref VPC
    Subnets:
        Description: A list of the subnets created
        Value: !If
            - IsMultiAZ
            - !Join [",", [!Ref Subnet1, !Ref Subnet2]]

```



```

    - !Ref Subnet1
Subnet1:
  Description: A reference to the subnet in the 1st Availability Zone
  Value: !Ref Subnet1
Subnet2:
  Condition: IsMultiAZ
  Description: A reference to the subnet in the 2nd Availability Zone
  Value: !Ref Subnet2
InfluxDBSecurityGroup:
  Description: Security group with port 8086 ingress rule
  Value: !Ref InfluxDBSecurityGroup

# Timestream for InfluxDB Resources
InfluxDBLogsS3Bucket:
  Description: S3 Bucket containing InfluxDB logs from the DB instance
  Value: !Ref InfluxDBLogsS3Bucket
DbInstance:
  Description: A reference to the Timestream for InfluxDB DB instance
  Value: !Ref DbInstance
InfluxAuthParametersSecretArn:
  Description: "The Amazon Resource Name (ARN) of the AWS Secrets Manager secret
containing the initial InfluxDB authorization parameters. The secret value is a JSON
formatted key-value pair holding InfluxDB authorization values: organization, bucket,
username, and password."
  Value: !GetAtt DbInstance.InfluxAuthParametersSecretArn
Endpoint:
  Description: The endpoint URL to connect to InfluxDB
  Value: !Join ["", ["https://", !GetAtt DbInstance.Endpoint, ":8086"]]

```

使用 CloudFormation 部署 Windows 型堆疊

此頁面提供 Windows 型部署中常用 CloudFormation 資源的技術參考文件連結。

CloudFormation 透過 Infrastructure as Code (IaC) 支援部署和管理 Microsoft Windows 堆疊。您可以使用 CloudFormation 來自動佈建 Windows 型 EC2 執行個體、Amazon RDS SQL Server 和 Microsoft Active Directory 透過 Directory Service。

AWS 提供專為 Windows 平台設計的預先設定 Amazon Machine Image (AMIs)，可協助您在 Amazon EC2 上快速部署應用程式。這些 AMIs 包含 Microsoft 預設設定和 AWS 特定自訂。使用 CloudFormation，可以選擇適當的 AMI，啟動執行個體，並使用遠端桌面連線存取它，就像使用任何其他 Windows Server 一樣。AMIs 包含必要的軟體元件，包括 EC2Launch (版本因 Windows Server 版本而異)、AWS Tools for PowerShell、AWS Systems Manager CloudFormation 和各種

網路、儲存和圖形驅動程式，以確保最佳的效能和 AWS 與服務的相容性。如需詳細資訊，請參閱 [AWS Windows AMI 參考](#)。

CloudFormation 也支援軟體組態工具，例如 UserData 指令碼，可在 EC2 執行個體首次開機時執行 PowerShell 或批次命令。它還提供協助程式指令碼 (cfn-init、cfn-signal、cfn-get-metadata 和 cfn-hup)，並支援 `AWS::CloudFormation::Init` 中繼資料，用於管理 Windows 執行個體上的套件、檔案和服務。

對於企業環境，CloudFormation 會啟用網域加入，透過 EC2 Windows 授權模型進行授權管理，以及使用 AWS Secrets Manager 進行安全憑證處理。CloudFormation 結合版本控制範本和可重複的部署，可協助組織在多個 AWS 區域和帳戶中維持一致、安全且可擴展 Windows 的環境。

如需 Windows 型部署中常用 CloudFormation 資源的詳細資訊，請參閱下列技術參考主題。

Resource Type (資源類型)	Description
AWS::EC2::Instance	用於啟動 Windows EC2 執行個體。
AWS::EC2::SecurityGroup	定義 Windows 工作負載的防火牆規則。
AWS::AutoScaling::AutoScalingGroup	用於擴展 Windows EC2 執行個體。
AWS::EC2::LaunchTemplate	
AWS::DirectoryService::MicrosoftAD	用於部署 Microsoft Active Directory。
AWS::FSx::FileSystem	用於部署 FSx for Windows File Server。
AWS::RDS::DBInstance	用於在 Amazon RDS 上佈建 SQL Server。
AWS::CloudFormation::Init	在 EC2 中繼資料內用於設定執行個體。 如需詳細資訊，請參閱 引導 Windows 型 CloudFormation 堆疊 。
AWS::SecretsManager::Secret	用於安全管理憑證和 Windows 密碼。
AWS::SSM::Parameter	用於安全地存放組態值。
AWS::IAM::InstanceProfile	用於為 EC2 執行個體上執行的應用程式授予許可。

Resource Type (資源類型)	Description
AWS::IAM::Role	

引導 Windows 型 CloudFormation 堆疊

本主題說明如何引導 Windows 堆疊和疑難排解堆疊建立問題。

主題

- [EC2 執行個體中的使用者資料](#)
- [CloudFormation 協助程式指令碼](#)
- [引導 Windows 堆疊的範例](#)
- [Windows 檔案路徑中的轉義反斜線](#)
- [管理 Windows 服務](#)
- [疑難排解堆疊建立問題](#)

EC2 執行個體中的使用者資料

使用者資料是一種 Amazon EC2 功能，可讓您在啟動時將指令碼或組態資訊傳遞給 EC2 執行個體。

對於 Windows EC2 執行個體：

- 可以使用批次指令碼 (使用 `<script>` 標籤) 或 PowerShell 指令碼 (使用 `<powershell>` 標籤)。
- 由 EC2Launch 處理指令碼執行。

Important

如果要建立自己的 Windows AMI 以搭配 CloudFormation 使用，請確保已正確設定 EC2Launch v2。CloudFormation 引導工具需要 EC2Launch v2，才能在堆疊建立期間正確初始化和設定 Windows 執行個體。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的[在 EC2 Windows 執行個體啟動期間使用 EC2Launch v2 代理程式來執行任務](#)。如需 AMIs 的相關資訊 AWS Windows，請參閱 [AWS Windows AMI 參考](#)。

CloudFormation 協助程式指令碼

協助程式指令碼是在引導程序期間設定執行個體的公用程式。與 Amazon EC2 使用者資料搭配使用，其可提供強大的組態選項。

CloudFormation 提供下列 Python 協助程式指令碼，您可以使用這些指令碼在當做堆疊一部分建立的 Amazon EC2 執行個體上安裝軟體及啟動服務：

- `cfn-init` – 用來擷取及解釋資源中繼資料、安裝套件、建立檔案及啟動服務。
- `cfn-signal` – 用來透過 `CreationPolicy` 發出訊號，讓您可以在先決條件資源或應用程式就緒時，同步堆疊中的其他資源。
- `cfn-get-metadata` – 用來擷取資源的中繼資料或特定金鑰的路徑。
- `cfn-hup` – 用來檢查是否有中繼資料更新，並在偵測到變更時執行自訂勾點。

您可以直接從範本呼叫指令碼。指令碼會搭配同一個範本中定義的資源中繼資料使用。指令碼會在堆疊建立程序期間於 Amazon EC2 執行個體上執行。

如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [CloudFormation 協助程式指令碼參考](#)。

引導 Windows 堆疊的範例

讓我們檢查執行下列動作的 Windows Server 範本中的範例程式碼片段：

- 從 Windows Server 2022 AMI 中啟動名為 `TestInstance` 的 EC2 執行個體。
- 建立簡單的測試檔案以驗證 `cfn-init` 是否正常運作。
- 設定 `cfn-hup` 以進行持續的組態管理。
- 使用 `CreationPolicy` 來確保執行個體發出成功完成的訊號。

`cfn-init` 協助程式指令碼用於根據範本中 `AWS::CloudFormation::Init` 資源中的資訊來執行這些動作中的每個動作。

`AWS::CloudFormation::Init` 區段名為 `TestInstance`，開頭為下列宣告。

```
TestInstance:
  Type: AWS::EC2::Instance
  Metadata:
```

```
AWS::CloudFormation::Init:
  configSets:
    default:
      - create_files
      - start_services
```

在此之後，將宣告 `AWS::CloudFormation::Init` 的 `files` 區段：

```
create_files:
  files:
    c:\cfn\test.txt:
      content: !Sub |
        Hello from ${AWS::StackName}
    c:\cfn\cfn-hup.conf:
      content: !Sub |
        [main]
        stack=${AWS::StackName}
        region=${AWS::Region}
        interval=2
    c:\cfn\hooks.d\cfn-auto-reloader.conf:
      content: !Sub |
        [cfn-auto-reloader-hook]
        triggers=post.update
        path=Resources.TestInstance.Metadata.AWS::CloudFormation::Init
        action=cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --
region ${AWS::Region}
```

在此建立三個檔案，並放在伺服器執行個體的 `C:\cfn` 目錄中：

- `test.txt`，一種簡單的測試檔案，可驗證 `cfn-init` 是否正常運作，並可建立具有動態內容的檔案。
- `cfn-hup.conf`，`cfn-hup` 的組態檔案，具有 2 分鐘檢查間隔。
- `cfn-auto-reloader.conf`，`cfn-hup` 使用的掛接組態檔案，當 `AWS::CloudFormation::Init` 中的中繼資料變更時，用於啟動更新 (呼叫 `cfn-init`)。

接下來是 `start_services` 區段，它可設定 Windows 服務。

```
start_services:
  services:
    windows:
      cfn-hup:
```

```
enabled: true
ensureRunning: true
files:
  - c:\cfn\cfn-hup.conf
  - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

本節可確保 cfn-hup 服務已啟動，並在修改組態檔案時自動重新啟動。服務會監控 CloudFormation 中繼資料的變更，並在偵測到更新時重新執行 cfn-init。

接下來是 Properties 區段。

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Metadata:
    AWS::CloudFormation::Init:
      # ... metadata configuration ...
  Properties:
    InstanceType: t2.large
    ImageId: '{{resolve:ssm:/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base}}'
    SecurityGroupIds:
      - !Ref InstanceSecurityGroup
    KeyName: !Ref KeyPairName
    UserData:
      Fn::Base64: !Sub |
        <powershell>
          cfn-init.exe -v -s ${AWS::StackName} -r TestInstance -c default --region
${AWS::Region}
          cfn-signal.exe -e $lastexitcode --stack ${AWS::StackName} --resource
TestInstance --region ${AWS::Region}
        </powershell>
```

在此區段中，UserData 屬性包含 EC2Launch 將執行的 PowerShell 指令碼，被包圍在 <powershell> 標籤中。指令碼使用 default configSet 來執行 cfn-init，然後使用 cfn-signal 向 CloudFormation 回報結束碼。CreationPolicy 用於確保在堆疊建立視為完成之前正確設定執行個體。

ImageId 屬性會使用 Systems Manager Parameter Store 公有參數自動擷取最新的 Windows Server 2022 AMI ID。此方法不需要區域特定的 AMI 映射，並確保您始終取得最新的 AMI。針對 AMI ID

使用 Systems Manager 參數是維護目前 AMI 參考的最佳實務。如果打算連線到執行個體，請確保 SecurityGroupIds 屬性可參考允許 RDP 存取的安全群組。

CreationPolicy 會宣告為資源屬性的一部分，並指定逾時期間。執行個體組態完成時，使用者資料中的 cfn-signal 命令會發出訊號：

```
TestInstance:
  Type: AWS::EC2::Instance
  CreationPolicy:
    ResourceSignal:
      Timeout: PT20M
  Properties:
    # ... other properties ...
```

由於引導程序很小，而且只會建立檔案並啟動服務，因此 CreationPolicy 會在逾時前等待 20 分鐘 (PT20M)。使用 ISO 8601 持續時間格式來指定逾時。請注意，Windows 執行個體啟動所需的時間通常比 Linux 執行個體長，因此請進行徹底測試，以確定滿足您需求的最佳逾時值。

如果一切順利，CreationPolicy 會成功完成，而且您可以使用執行個體的公有 IP 位址來存取 Windows Server 執行個體。堆疊建立完成後，執行個體 ID 和公有 IP 位址將顯示在 CloudFormation 主控台的輸出標籤中。

```
Outputs:
  InstanceId:
    Value: !Ref TestInstance
    Description: Instance ID of the Windows Server
  PublicIP:
    Value: !GetAtt TestInstance.PublicIp
    Description: Public IP address of the Windows Server
```

也可以手動驗證引導是否正常運作，方法是透過 RDP 連線至執行個體，並檢查檔案 C:\cfn\test.txt 是否存在並包含預期的內容。如需有關連線至 Windows 執行個體的詳細資訊，請參閱《Amazon EC2 使用者指南》中的[使用 RDP 連線至 Windows 執行個體](#)。

Windows 檔案路徑中的轉義反斜線

在 CloudFormation 範本中參考 Windows 路徑時，請務必記得根據您使用的範本格式正確轉義反斜線 (\)。

- 對於 JSON 範本，必須在 Windows 檔案路徑中使用雙反斜線，因為 JSON 會將反斜線視為轉義字元。第一個反斜線會轉義第二個斜線，導致解譯單一常值反斜線。

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\\\SharePoint\\\\SharePointFoundation2010.exe /extract:C:\\\\SharePoint
\\\\SPF2010 /quiet /log:C:\\\\SharePoint\\\\SharePointFoundation2010-extract.log"
  }
}
```

- 對於 YAML 範本，單一反斜線通常就足夠。

```
commands:
  1-extract:
    command: C:\SharePoint\SharePointFoundation2010.exe /extract:C:\SharePoint
\SPF2010 /quiet /log:C:\SharePoint\SharePointFoundation2010-extract.log
```

管理 Windows 服務

您管理 Windows 服務的方式與管理 Linux 服務的方式相同，但使用的是 windows 鍵而非 sysvinit。下列範例會啟動 cfn-hup 服務，並將其設為「自動」，如果 cfn-init 修改 c:\cfn\cfn-hup.conf 或 c:\cfn\hooks.d\cfn-auto-reloader.conf 組態檔案，則會重新啟動服務。

```
services:
  windows:
    cfn-hup:
      enabled: true
      ensureRunning: true
      files:
        - c:\cfn\cfn-hup.conf
        - c:\cfn\hooks.d\cfn-auto-reloader.conf
```

可以使用該名稱而非顯示名稱來參考服務，以相同的方式管理其他 Windows 服務。

疑難排解堆疊建立問題

如果堆疊在建立期間失敗，預設行為是復原故障。雖然這樣的預設行為一般而言是好的，因其可避免不必要的額外費用，但卻會對堆疊建立失敗原因進行除錯時造成困難。

若要在使用 CloudFormation 主控台建立或更新堆疊時關閉此行為，請選擇堆疊失敗選項下的保留成功佈建的資源選項。如需詳細資訊，請參閱[選擇佈建資源時的失敗處理方式](#)。這可讓您登入執行個體並檢視日誌檔案，以找出執行啟動指令碼時發生的問題。

要查看的重要日誌如下：

- EC2 組態日誌，位於 %ProgramData%\Amazon\EC2Launch\log\agent.log
- C:\cfn\log\cfn-init.log 中的 cfn-init 日誌 (檢查特定故障點的結束代碼和錯誤訊息)

如需更多日誌，請參閱《Amazon EC2 使用者指南》中的下列主題：

- [EC2Launch 目錄結構](#)
- [EC2Launch v2 目錄結構](#)

如需有關對引導問題進行疑難排解的詳細資訊，請參閱[如何使用 Windows 執行個體對 CloudFormation 堆疊中不會引導的協助程式指令碼進行疑難排解？](#)。

使用 CloudFormation 提供的資源類型來擴展範本的功能

CloudFormation 提供數種可在堆疊範本中使用的資源類型來擴展其功能，從而超越簡單的堆疊範本功能。

這些資源類型包括：

Resource Type (資源類型)	Description	文件
自訂資源	AWS::CloudFormation::Custom Resource 資源類型可讓您建立自訂資源，以執行特定佈建任務，或包含無法作為 CloudFormation 資源類型使用的資源。	自訂資源
巨集	AWS::CloudFormation::Macro 資源類型會定義可重複使用的程式碼片段，可在 CloudFormation 範本上執行自訂處理。巨集可以在堆疊建立或更新期間修改範本、產生其他資源或執行其他自訂操作。	範本巨集
巢狀堆疊	AWS::CloudFormation::Stack 資源類型可讓您在 CloudFormation 範本內建立巢狀堆疊，以取得更加模組化且可重複使用的堆疊架構。	巢狀堆疊

Resource Type (資源類型)	Description	文件
StackSet	AWS::CloudFormation::StackSet 資源類型會建立或更新 CloudFormation StackSet，這是堆疊的容器，可跨多個 AWS 帳戶和區域部署。	透過 StackSets 管理堆疊
等待條件	AWS::CloudFormation::WaitCondition 資源類型會暫停堆疊建立或更新，直到符合特定條件為止，例如成功完成長時間執行的程序或外部資源的可用性。	等待條件
等待條件控點	AWS::CloudFormation::WaitConditionHandle 資源類型可與 AWS::CloudFormation::WaitCondition 資源類型搭配使用。它提供預先簽章的 URL，用於傳送訊號，指出已滿足特定條件。這些訊號可讓堆疊建立或更新程序繼續進行。	等待條件

使用自訂資源建立自訂佈建邏輯

自訂資源可讓您能在 CloudFormation 範本中撰寫自訂佈建邏輯，讓 CloudFormation 可在您建立、更新 (若您變更自訂資源) 或刪除堆疊時隨時執行。當您的佈建需求涉及無法以 CloudFormation 內建資源類型表示的複雜邏輯或工作流程時，這會很實用。

例如，您可能會希望包含無法做為 CloudFormation 資源類型使用的資源。您可以使用自訂資源包含那些資源。以此方式，您仍然可以在單一堆疊中管理所有相關資源。

在 CloudFormation 範本中，您可以使用 `AWS::CloudFormation::CustomResource` 或 `Custom::MyCustomResourceTypeName` 資源類型來定義自訂資源。自訂資源需要一個屬性：服務字符，指定 CloudFormation 傳送請求的目的地，例如 Amazon SNS 主題或 Lambda 函式。

下列主題提供有關如何使用自訂資源的資訊：

主題

- [自訂資源的運作方式](#)
- [回應逾時](#)

- [CloudFormation 自訂資源請求和回應參考](#)
- [Amazon SNS 支援的自訂資源](#)
- [Lambda 後端自訂資源](#)

Note

CloudFormation 登錄檔與自訂資源各有其優點。自訂資源提供下列優點：

- 您無需註冊資源。
- 您可在範本中包含整個資源，而無需註冊。
- 支援 Create、Update 和 Delete 操作

登錄檔型資源所提供的優點包括如下：

- 支援建模、佈建和管理第三方應用程式資源
- 支援 Create、Read、Update Delete、和 List (CRUDL) 操作
- 支援私有和第三方資源類型的偏移偵測

與自訂資源不同，以登錄為基礎的資源不需要關聯 Amazon SNS 主題或 Lambda 函數來執行 CRUDL 操作。如需詳細資訊，請參閱[使用 CloudFormation 登錄檔管理擴充功能](#)。

自訂資源的運作方式

設定新自訂資源的一般程序包括下列步驟。這些步驟涉及兩個角色：擁有自訂資源的自訂資源提供者，以及建立包含自訂資源類型範本的開發人員。這可以是同一個人，但若不是，自訂資源提供者應與範本開發人員協作。

1. 自訂資源提供者會撰寫邏輯，用於決定如何處理來自 CloudFormation 的請求，以及對自訂資源執行相應操作。
2. 自訂資源提供者會建立 Amazon SNS 主題或 Lambda 函式，供 CloudFormation 傳送請求。Amazon SNS 主題或 Lambda 函式必須位於即將建立堆疊的同一區域。
3. 自訂資源提供者會將 Amazon SNS 主題 ARN 或 Lambda 函式 ARN 提供給範本開發人員。

4. 範本開發人員在 CloudFormation 範本中定義自訂資源。其中包含服務字符和任何輸入資料參數。輸入資料的服務字符和結構由自訂資源提供者定義。服務字符會指定 Amazon SNS 主題 ARN 或 Lambda 函式 ARN，且為必要項目，但輸入資料是否需要則取決於自訂資源。

現在，每當任何人使用範本來建立、更新或刪除自訂資源時，CloudFormation 都會向指定的服務字符傳送請求，並等待回應後再繼續執行堆疊操作。

以下是從範本建立堆疊的流程總結：

1. CloudFormation 向指定的服務字符傳送請求。請求包含請求類型和預先簽章的 Amazon S3 儲存貯體 URL 等資訊，自訂資源會將回應傳送到其中。如需請求中包含項目的詳細資訊，請參閱 [CloudFormation 自訂資源請求和回應參考](#)。

以下範例資料顯示 CloudFormation 在 Create 請求中包含的項目：在此範例中，ResourceProperties 可讓 CloudFormation 建立自訂承載，以傳送至 Lambda 函式。

```
{
  "RequestType" : "Create",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::TestResource",
  "LogicalResourceId" : "MyTestResource",
  "ResourceProperties" : {
    "Name" : "Value",
    "List" : [ "1", "2", "3" ]
  }
}
```

2. 自訂資源提供者會處理 CloudFormation 請求，並將 SUCCESS 或 FAILED 的回應傳送至預先簽章的 URL。custom resource provider 會以 JSON 格式檔案提供回應，並上傳至預先簽章的 S3 URL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [使用預先簽章的 URL 上傳物件](#)。

在回應中，自訂資源提供者也可包含範本開發人員可存取的名稱值對。例如，若請求成功，回應可包含輸出資料，或是若請求失敗，則包含錯誤訊息。如需回應的詳細資訊，請參閱 [CloudFormation 自訂資源請求和回應參考](#)。

 Important

如果名稱值對包含敏感資訊，您應該使用 NoEcho 欄位來遮罩自訂資源的輸出。否則，這些值可透過表示屬性值 (例如 DescribeStackEvents) 的 API 可見。

如需使用 NoEcho 遮罩敏感資訊的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

custom resource provider 負責接聽和回應請求。例如，針對 Amazon SNS 通知，自訂資源提供者必須接聽和回應傳送到特定主題 ARN 的通知。CloudFormation 會在預先簽章的 URL 位置等待並接聽回應。

以下範例資料顯示回應中可能包含的自訂資源：

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique id for this create request",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MyTestResource",
  "PhysicalResourceId" : "TestResource1",
  "Data" : {
    "OutputName1" : "Value1",
    "OutputName2" : "Value2",
  }
}
```

- 在取得 SUCCESS 回應後，CloudFormation 便會繼續堆疊操作。若傳回 FAILED 回應或沒有任何回應，則操作便會失敗。任何來自自訂資源的輸出資料都會存放在預先簽章的 URL 位置。範本開發人員可使用 [Fn::GetAtt](#) 函數擷取該資料。

 Note

如果您使用 AWS PrivateLink，VPC 中的自訂資源必須能夠存取 CloudFormation 特定的 S3 儲存貯體。自訂資源必須傳送回應至預先簽章的 Amazon S3 URL。若無法將回應傳送至 Amazon S3，CloudFormation 便不會收到回應，而堆疊操作則會失敗。如需詳細資訊，請參閱 [CloudFormation 使用界面端點存取 \(AWS PrivateLink\)](#)。

回應逾時

自訂資源的預設逾時為 3600 秒 (1 小時)。如果在這段時間內沒有收到任何回應，堆疊操作就會失敗。

您可根據預期自訂資源傳回回應的時間，調整逾時數值。例如，若佈建的自訂資源會調用預期在五分鐘內回應的 Lambda 函式，您可在堆疊範本中透過指定 `ServiceTimeout` 屬性，將逾時時間設定為五分鐘。如需詳細資訊，請參閱[CloudFormation 自訂資源請求和回應參考](#)。如此一來，若 Lambda 函式發生錯誤導致卡住，CloudFormation 會在五分鐘後使堆疊操作失敗，而非等待完整的一小時。

但請注意，勿將逾時數值設定過低。為避免非預期的逾時，請確保您的自訂資源有足夠時間執行必要動作並傳回回應。

CloudFormation 自訂資源請求和回應參考

CloudFormation 透過與您的自訂資源提供者通訊的請求回應通訊協定來管理自訂資源。每個請求都包含請求類型 (Create、或 Delete)Update，並遵循此高階工作流程：

1. 範本開發人員在範本 `ServiceTimeout` 中定義具有 `ServiceToken` 和 的自訂資源，並啟動堆疊操作。
2. CloudFormation 透過 SNS 或 Lambda 將 JSON 請求傳送至自訂資源提供者。
3. 自訂資源提供者會處理請求，並在逾時期間到期之前，將 JSON 回應傳回預先簽章的 Amazon S3 儲存貯體 URL。
4. CloudFormation 會讀取回應並繼續堆疊操作。如果在逾時期間結束前未收到任何回應，請求會被視為失敗，且堆疊操作會失敗。

如需詳細資訊，請參閱[自訂資源的運作方式](#)。

本節說明每個請求類型的結構、參數和預期回應。

Note

回應內文的總大小不能超過 4096 個位元組。

範本設定

在範本中定義自訂資源時，範本開發人員會使用 [AWS::CloudFormation::CustomResource](#) 搭配下列屬性：

ServiceToken

Amazon SNS 主題 ARN 或 Lambda 函數 ARN，來自與堆疊相同的區域。

必要：是

類型：字串

ServiceTimeout

自訂資源操作逾時之前的時間上限，以秒為單位。它必須是介於 1 到 3600 之間的值。預設：3600 秒 (1 小時)。

必要：否

類型：字串

支援其他資源屬性。資源屬性將做為 包含在請求ResourceProperties中。自訂資源提供者必須判斷哪些屬性有效及其可接受的值。

請求物件

Create

當範本開發人員建立包含自訂資源的堆疊時，CloudFormation 會傳送將 RequestType 設定為 的請求Create。

建立包含以下欄位的請求：

RequestType

Create.

必要：是

類型：字串

RequestId

請求的唯一 ID。

結合 StackId 與 RequestId 形成值，該值可用於唯一識別特定自訂資源上的請求。

必要：是

類型：字串

StackId

識別包含自訂資源之堆疊的 Amazon Resource Name (ARN)。

結合 StackId 與 RequestId 形成值，該值可用於唯一識別特定自訂資源上的請求。

必要：是

類型：字串

ResponseURL

回應 URL 會識別預先簽章的 S3 儲存貯體，該儲存貯體會從自訂資源提供者接收對 CloudFormation 的回應。

必要：是

類型：字串

ResourceType

CloudFormation 範本內由範本開發人員選擇之自訂資源的資源類型。自訂資源類型名稱的長度上限為 60 個字元，且可包含英數字元及以下字元：_@-。

必要：是

類型：字串

LogicalResourceId

CloudFormation 範本中自訂資源的範本開發人員選擇名稱（邏輯 ID）。

必要：是

類型：字串

ResourceProperties

此欄位包含由範本開發人員傳送的 Properties 物件內容。其內容由自訂資源提供者定義。

必要：否

類型：JSON 物件

範例

```
{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-create-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Update

當範本開發人員變更範本中自訂資源的屬性並更新堆疊時，CloudFormation 會將請求傳送至自訂資源提供者，並將 `RequestType` 設定為 `Update`。這表示您的自訂資源程式碼不需要偵測資源的變更，因為它知道其屬性在請求類型為 `Update` 時已變更。

更新包含以下欄位的請求：

RequestType

Update.

必要：是

類型：字串

RequestId

請求的唯一 ID。

結合 `StackId` 與 `RequestId` 形成值，該值可用於唯一識別特定自訂資源上的請求。

必要：是

類型：字串

StackId

識別包含自訂資源之堆疊的 Amazon Resource Name (ARN)。

結合 StackId 與 RequestId 形成值，該值可用於唯一識別特定自訂資源上的請求。

必要：是

類型：字串

ResponseURL

回應 URL 會識別預先簽章的 S3 儲存貯體，該儲存貯體會從自訂資源提供者接收對 CloudFormation 的回應。

必要：是

類型：字串

ResourceType

CloudFormation 範本內由範本開發人員選擇之自訂資源的資源類型。自訂資源類型名稱的長度上限為 60 個字元，且可包含英數字元及以下字元：_@-。您無法於更新期間變更類型。

必要：是

類型：字串

LogicalResourceId

CloudFormation 範本中自訂資源的範本開發人員選擇名稱（邏輯 ID）。

必要：是

類型：字串

PhysicalResourceId

自訂資源提供者定義的實體 ID，對於該提供者是唯一的。

必要：是

類型：字串

ResourceProperties

此欄位包含由範本開發人員傳送的 Properties 物件內容。其內容由自訂資源提供者定義。

必要：否

類型：JSON 物件

OldResourceProperties

僅用於 Update 請求。範本開發人員之前在 CloudFormation 範本中宣告的資源屬性值。

必要：是

類型：JSON 物件

範例

```
{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-update-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "new-string",
    "key2" : [ "new-list" ],
    "key3" : { "key4" : "new-map" }
  },
  "OldResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

Delete

當範本開發人員刪除堆疊或從範本中移除自訂資源，然後更新堆疊時，CloudFormation 會傳送將 RequestType 設定為 的請求Delete。

刪除包含以下欄位的請求：

RequestType

Delete.

必要：是

類型：字串

RequestId

請求的唯一 ID。

必要：是

類型：字串

StackId

識別包含自訂資源之堆疊的 Amazon Resource Name (ARN)。

必要：是

類型：字串

ResponseURL

回應 URL 會識別預先簽章的 S3 儲存貯體，該儲存貯體會從自訂資源提供者接收對 CloudFormation 的回應。

必要：是

類型：字串

ResourceType

CloudFormation 範本內由範本開發人員選擇之自訂資源的資源類型。自訂資源類型名稱的長度上限為 60 個字元，且可包含英數字元及以下字元：_@-。

必要：是

類型：字串

LogicalResourceId

CloudFormation 範本中自訂資源的範本開發人員選擇名稱（邏輯 ID）。

必要：是

類型：字串

PhysicalResourceId

自訂資源提供者定義的實體 ID，對於該提供者是唯一的。

必要：是

類型：字串

ResourceProperties

此欄位包含由範本開發人員傳送的 Properties 物件內容。其內容由自訂資源提供者定義。

必要：否

類型：JSON 物件

範例

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/id",
  "ResponseURL" : "pre-signed-url-for-delete-response",
  "ResourceType" : "Custom::MyCustomResourceType",
  "LogicalResourceId" : "resource-logical-id",
  "PhysicalResourceId" : "provider-defined-physical-id",
  "ResourceProperties" : {
    "key1" : "string",
    "key2" : [ "list" ],
    "key3" : { "key4" : "map" }
  }
}
```

回應物件

自訂資源提供者會將回應傳送至所有請求類型的預先簽章 URL。如果自訂資源提供者未傳送回應，CloudFormation 會等到操作逾時。

回應必須是具下列欄位的 JSON 物件：

Status

必須為 SUCCESS 或 FAILED。

必要：是

類型：字串

RequestId

請求的唯一 ID。複製此值與請求中顯示的完全相同。

必要：是

類型：字串

StackId

識別包含自訂資源之堆疊的 Amazon Resource Name (ARN)。複製此值與請求中顯示的完全相同。

必要：是

類型：字串

LogicalResourceId

CloudFormation 範本中自訂資源的範本開發人員選擇名稱（邏輯 ID）。複製此值與請求中顯示的完全相同。

必要：是

類型：字串

PhysicalResourceId

此值應為自訂資源廠商的唯一識別碼，且大小上限為 1 KB。此值必須是非空白字串，並且對於相同資源的所有回應必須完全相同。

更新自訂資源時，傳回的值會PhysicalResourceId決定更新行為。如果值保持不變，CloudFormation 會將其視為正常更新。如果值變更，CloudFormation 會將更新解譯為取代，並將刪除請求傳送至舊資源。如需詳細資訊，請參閱[AWS::CloudFormation::CustomResource](#)。

必要：是

類型：字串

Reason

描述失敗回應的原因。

在 Status 為 FAILED 時需要。否則為選用。

必要：有條件

類型：字串

NoEcho

指示使用 `Fn::GetAtt` 函數擷取自訂資源時是否要遮罩其輸出。如果設定為 `true`，則所有傳回的值都會以星號 (`*****`) 遮罩，但儲存在範本 *Metadata* 區段中的值除外。CloudFormation 不會轉換、修改或標記您在 *Metadata* 區段中包含的任何資訊。預設值為 `false`。

如需使用 NoEcho 遮罩敏感資訊的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

僅適用於 Create 和 Update 回應。不支援 Delete 回應。

必要：否

類型：布林值

Data

要與回應一起傳送的自訂資源提供者定義之名稱值對。您可以在模板中透過 `Fn::GetAtt` 依名稱存取此處提供的值。

僅適用於 Create 和 Update 回應。不支援 Delete 回應。

Important

如果名稱值對包含敏感資訊，您應該使用 NoEcho 欄位來遮罩自訂資源的輸出。否則，這些值可透過表示屬性值 (例如 `DescribeStackEvents`) 的 API 可見。

必要：否

類型：JSON 物件

成功回應範例

Create 和 Update 回應

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "NoEcho": true,
```

```
"Data": {
  "key1": "value1",
  "key2": "value2"
}
```

Delete 回應

```
{
  "Status": "SUCCESS",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id"
}
```

失敗的回應範例

```
{
  "Status": "FAILED",
  "RequestId": "unique-request-id",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/name/id",
  "LogicalResourceId": "resource-logical-id",
  "PhysicalResourceId": "provider-defined-physical-id",
  "Reason": "Required failure reason string"
}
```

Amazon SNS 支援的自訂資源

本主題說明如何為自訂資源設定服務字符，該服務字符會指定 CloudFormation 傳送請求的 Amazon SNS 主題。您也會了解自訂資源堆疊建立、更新和刪除所產生的事件順序，以及傳送和接收的訊息。

您可以使用自訂資源和 Amazon SNS 來啟用案例，例如將新資源新增至堆疊，並將動態資料插入堆疊。例如，當您建立堆疊時，CloudFormation 可以將 Create 請求傳送到應用程式 (執行於 Amazon EC2 執行個體上) 監控的主題。Amazon SNS 通知會觸發應用程式執行額外的佈建任務，例如擷取列入允許清單的彈性 IP 地址集區。完成之後，應用程式會傳送回應 (以及任何輸出資料)，以通知 CloudFormation 繼續堆疊操作。

若您指定 Amazon SNS 主題作為自訂資源的目標，CloudFormation 會在涉及該自訂資源的堆疊操作期間，向指定的 SNS 主題傳送訊息。若要處理這些訊息並執行必要的操作，您必須有支援的端點訂閱該 SNS 主題。

有關自訂資源的簡介及其運作方式，請參閱 [自訂資源的運作方式](#)。如需 Amazon SNS 及其運作方式的資訊，請參閱 [Amazon Simple Notification Service 開發人員指南](#)。

使用 Amazon SNS 建立自訂資源

主題

- [步驟 1：建立堆疊](#)
- [步驟 2：更新堆疊](#)
- [步驟 3：刪除堆疊](#)

步驟 1：建立堆疊

1. 範本開發人員會建立包含自訂資源的 CloudFormation 堆疊。

下列範例範本中，我們為邏輯 ID 為 *MySeleniumTest* 的自訂資源使用自訂資源類型名稱 *Custom::SeleniumTester*。自訂資源類型名稱必須是英數字元，長度上限為 60 個字元。

自訂資源類型的宣告是使用服務字符、選用的提供者特定屬性，以及選用的 [Fn::GetAtt](#) 屬性 (由自訂資源提供者定義)。這些屬性可用來將資訊從範本開發人員傳遞到自訂資源提供者 (反之亦然)。服務字符會指定資源提供者已設定的 Amazon SNS 主題。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
      "Type": "Custom::SeleniumTester",
      "Version" : "1.0",
      "Properties" : {
        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
          "http://search.mysite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
      }
    }
  },
  "Outputs" : {
    "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    },
    "numRespondents" : {
```

```

        "Value" : { "Fn::GetAtt" : [ "MySeleniumTest", "lastUpdate" ] }
    }
}

```

Note

在提供者回應 CloudFormation 期間，自訂資源提供者會傳回以 Fn::GetAtt 存取的資料名稱和值。如果 custom resource provider 是第三方，則範本開發人員必須從自訂資源提供者取得這些傳回值的名稱。

2. CloudFormation 會使用 "RequestType" : "Create" 將 Amazon SNS 通知傳送給資源提供者，其中包含堆疊的相關資訊、堆疊範本的自訂資源屬性，以及回應的 S3 URL。

ServiceToken 屬性中的範本內嵌了用來傳送通知的 SNS 主題。為避免使用硬式編碼的值，範本開發人員可以使用範本參數，在啟動堆疊時輸入值。

以下範例顯示自訂資源 Create 請求，其中包括使用 Custom::SeleniumTester 的 LogicalResourceId 所建立的自訂資源類型名稱 MySeleniumTester：

```

{
  "RequestType" : "Create",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}

```

如需 Create 請求之請求物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

3. 自訂資源提供者會處理範本開發人員傳送的資料，並判斷 Create 請求是否成功。接著，資源提供者會使用 CloudFormation 傳送的 S3 URL 來傳送 SUCCESS 或 FAILED 的回應。

CloudFormation 預期的回應欄位根據回應類型而異。如需特定請求類型的回應欄位資訊，請參閱 [請求和回應參考](#) 區段中該請求類型的文件。

在回應建立或更新請求時，自訂資源提供者可傳回回應 Data 欄位中的資料元素。這些是名稱值對；名稱對應至 Fn::GetAtt 屬性，其與堆疊範本中的自訂資源搭配使用。值是範本開發人員以屬性名稱在資源上呼叫 Fn::GetAtt 時傳回的資料。

以下為自訂資源回應的範例：

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "Data" : {
    "resultsPage" : "http://www.myexampledomain/test-results/guid",
    "lastUpdate" : "2012-11-14T03:30Z"
  }
}
```

如需 Create 請求之回應物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

StackId、RequestId 以及 LogicalResourceId 欄位必須從請求逐字複製。

4. CloudFormation 會將堆疊狀態宣告為 CREATE_COMPLETE 或 CREATE_FAILED。如果已成功建立堆疊，範本開發人員可透過 [Fn::GetAtt](#) 來存取利用所建立自訂資源的輸出值。

例如，示範的自訂資源範本會使用 Fn::GetAtt 將資源輸出複製到堆疊輸出中：

```
"Outputs" : {
  "topItem" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
  },
  "numRespondents" : {
    "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
  }
}
```

步驟 2：更新堆疊

若要更新現有堆疊，您必須提交範本以指定堆疊中的資源屬性更新，如以下範例所示。CloudFormation 只會更新含範本中指定變更的資源。如需詳細資訊，請參閱[了解更新堆疊資源的行為](#)。

您可以更新需要替換基礎實體資源的自訂資源。當您更新 CloudFormation 範本中的自訂資源時，CloudFormation 會將更新請求傳送給該自訂資源。若需替換自訂資源，新的自訂資源必須以新的實體 ID 傳送回應。CloudFormation 接收回應時，會比較新舊自訂資源的 `PhysicalResourceId`。如果不同，CloudFormation 會將更新視為取代，並將刪除請求傳送到舊資源，如[步驟 3：刪除堆疊](#)所示。

Note

如果您未變更自訂資源，CloudFormation 就不會在堆疊更新期間傳送請求給自訂資源。

1. 範本開發人員可啟動包含自訂資源的堆疊更新。在更新期間，範本開發人員可以指定堆疊範本中的新 Properties (屬性)。

以下是使用自訂資源類型的堆疊範本 Update 範例：

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "MySeleniumTest" : {
      "Type": "Custom::SeleniumTester",
      "Version" : "1.0",
      "Properties" : {
        "ServiceToken": "arn:aws:sns:us-west-2:123456789012:CRTest",
        "seleniumTester" : "SeleniumTest()",
        "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
          "http://search.mysite.com",
          "http://mynewsite.com" ],
        "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
      }
    }
  },
  "Outputs" : {
    "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
    }
  }
}
```

```

    },
    "numRespondents" : {
        "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
    }
}
}

```

- CloudFormation 會使用 "RequestType" : "Update" 將 Amazon SNS 通知傳送給資源提供者，其中包含類似於 Create 呼叫的資訊，差別在於 OldResourceProperties 欄位包含舊資源屬性，而 ResourceProperties 包含更新的資源屬性 (如果有)。

以下是 Update 請求範例：

```

{
  "RequestType" : "Update",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com",
    "http://mynewsite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
  },
  "OldResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
    "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com" ],
    "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
  }
}

```

如需 Update 請求之請求物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

- 自訂資源提供者會處理 CloudFormation 傳送的資料。自訂資源會執行更新並傳送 SUCCESS 或 FAILED 的回應給 S3 URL。然後，CloudFormation 會比較新舊自訂資源的 PhysicalResourceIDs。若其不同，CloudFormation 會將該更新視為需要替換，並傳送刪除請求給舊資源。以下範例示範 custom resource provider 對 Update 請求的回應。

```
{
  "Status" : "SUCCESS",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester2"
}
```

如需 Update 請求之回應物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

StackId、RequestId 以及 LogicalResourceId 欄位必須從請求逐字複製。

4. CloudFormation 會將堆疊狀態宣告為 UPDATE_COMPLETE 或 UPDATE_FAILED。如果更新失敗，即會復原堆疊。如果已成功更新堆疊，範本開發人員即可使用 Fn::GetAtt 來存取已建立之自訂資源的任何新輸出值。

步驟 3：刪除堆疊

1. 範本開發人員可刪除包含自訂資源的堆疊。CloudFormation 可取得堆疊範本中指定的目前屬性與 SNS 主題，並準備向自訂資源提供者發出請求。
2. CloudFormation 會使用 "RequestType" : "Delete" 將 Amazon SNS 通知傳送給資源提供者，其中包含堆疊的目前資訊、堆疊範本的自訂資源屬性，以及回應的 S3 URL。

每當您刪除堆疊，或更新堆疊以移除或取代自訂資源時，CloudFormation 會比較新舊自訂資源的 PhysicalResourceId。如果不同，CloudFormation 會將更新視為取代，並針對舊資源 (OldPhysicalResource) 傳送刪除請求，如下列 Delete 請求範例所示。

```
{
  "RequestType" : "Delete",
  "RequestId" : "unique-request-id",
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",
  "ResponseURL" : "http://pre-signed-S3-url-for-response",
  "ResourceType" : "Custom::SeleniumTester",
  "LogicalResourceId" : "MySeleniumTester",
  "PhysicalResourceId" : "Tester1",
  "ResourceProperties" : {
    "seleniumTester" : "SeleniumTest()",
  }
}
```

```
"endpoints" : [ "http://mysite.com", "http://myecommercesite.com/", "http://search.mysite.com",  
  "http://mynewsite.com" ],  
  "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]  
}
```

如需 Delete 請求之請求物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

DescribeStackResource、DescribeStackResources 以及 ListStackResources 會顯示使用者定義的名稱 (如有指定)。

3. 自訂資源提供者會處理 CloudFormation 傳送的資料，並判斷 Delete 請求是否成功。接著，資源提供者會使用 CloudFormation 傳送的 S3 URL 來傳送 SUCCESS 或 FAILED 的回應。若要成功刪除含自訂資源的堆疊，custom resource provider 必須成功回應刪除請求。

以下為 custom resource provider 對 Delete 請求的回應範例：

```
{  
  "Status" : "SUCCESS",  
  "RequestId" : "unique-request-id",  
  "StackId" : "arn:aws:cloudformation:us-west-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10",  
  "LogicalResourceId" : "MySeleniumTester",  
  "PhysicalResourceId" : "Tester1"  
}
```

如需 Delete 請求之回應物件的詳細資訊，請參閱 [請求和回應參考](#) 主題。

StackId、RequestId 以及 LogicalResourceId 欄位必須從請求逐字複製。

4. CloudFormation 會將堆疊狀態宣告為 DELETE_COMPLETE 或 DELETE_FAILED。

Lambda 後端自訂資源

當您將 Lambda 函式與自訂資源建立關聯時，便會在建立、更新或刪除自訂資源時調用函數。CloudFormation 會呼叫 Lambda API 以調用函數，並將所有請求資料 (例如請求類型和資源屬性) 傳遞給函數。Lambda 函式與 CloudFormation 結合使用所帶來的功能和自訂性，可帶來各種使用案例，例如在堆疊建立時動態尋找 AMI ID，或是實作及使用公用程式函數 (例如字串反轉函數)。

有關自訂資源的簡介及其運作方式，請參閱 [自訂資源的運作方式](#)。

主題

- [逐步解說：使用 Lambda 支援的自訂資源建立延遲機制](#)
- [cfn-response 模組](#)

逐步解說：使用 Lambda 支援的自訂資源建立延遲機制

此逐步解說展示如何使用範例 CloudFormation 範本來設定和啟動 Lambda 支援的自訂資源。此範本會建立延遲機制，可暫停堆疊部署一段指定時間。當您需要在資源佈建期間刻意加入延遲時，例如等待資源穩定後再建立相依資源，這項功能相當實用。

Note

雖然先前建議 Lambda 後端自訂資源用於擷取 AMI IDs，但現在建議使用 AWS Systems Manager 參數。此方法可讓您的範本更具可重複使用性，且更易於維護。如需詳細資訊，請參閱[從 Systems Manager Parameter Store 取得純文字值](#)。

主題

- [概觀](#)
- [範例範本](#)
- [範例範本逐步解說](#)
- [先決條件](#)
- [啟動堆疊](#)
- [清除資源](#)
- [相關資訊](#)

概觀

本操作指南使用的堆疊範例範本，會建立一個 Lambda 支援的自訂資源。此自訂資源會在堆疊建立期間加入可設定的延遲 (預設為 60 秒)。僅當自訂資源的屬性被修改時，堆疊更新期間才會發生延遲。

此範本會佈建以下資源：

- 自訂資源，
- Lambda 函式，以及
- 讓 Lambda 將日誌寫入 CloudWatch 的 IAM 角色。

它還定義了兩個輸出：

- 函數實際等待的時間。
- 每次執行 Lambda 函式時產生的唯一識別碼。

Note

CloudFormation 是免費服務，但 Lambda 會根據您函數的請求次數和程式碼執行時間收費。如需 Lambda 定價的詳細資訊，請參閱 [AWS Lambda 定價](#)。

範例範本

以下是具備延遲機制的 Lambda 支援自訂資源範例範本：

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "LambdaExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Principal": { "Service": ["lambda.amazonaws.com"] },
            "Action": ["sts:AssumeRole"]
          }]
        },
        "Path": "/",
        "Policies": [{
          "PolicyName": "AllowLogs",
          "PolicyDocument": {
            "Statement": [{
              "Effect": "Allow",
              "Action": ["logs:*"],
              "Resource": "*"
            }]
          }
        }]
      }
    }
  }
}
```

```

    }
  },
  "CFNWaiter": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Runtime": "python3.9",
      "Timeout": 900,
      "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
      "Code": {
        "ZipFile": { "Fn::Join": ["\n", [
          "from time import sleep",
          "import json",
          "import cfnresponse",
          "import uuid",
          "",
          "def handler(event, context):",
          "    wait_seconds = 0",
          "    id = str(uuid.uuid1())",
          "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
          "        wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout\", 0))",
          "        sleep(wait_seconds)",
          "        response = {",
          "            \"TimeWaited\": wait_seconds,",
          "            \"Id\": id ",
          "        }",
          "        cfnresponse.send(event, context, cfnresponse.SUCCESS, response,",
          "\"Waiter-\"+id)",
          ""
        ]]}
      }
    }
  },
  "CFNWaiterCustomResource": {
    "Type": "AWS::CloudFormation::CustomResource",
    "Properties": {
      "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
      "ServiceTimeout": 60
    }
  }
},
"Outputs": {
  "TimeWaited": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },

```

```

    "Export": { "Name": "TimeWaited" }
  },
  "WaiterId": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
    "Export": { "Name": "WaiterId" }
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: "2010-09-09"
Resources:
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "lambda.amazonaws.com"
            Action:
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        - PolicyName: "AllowLogs"
          PolicyDocument:
            Statement:
              - Effect: "Allow"
                Action:
                  - "logs:*"
                Resource: "*"
  CFNWaiter:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.handler
      Runtime: python3.9
      Timeout: 900
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile:
          !Sub |

```

```
from time import sleep
import json
import cfnresponse
import uuid

def handler(event, context):
    wait_seconds = 0
    id = str(uuid.uuid1())
    if event["RequestType"] in ["Create", "Update"]:
        wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
        sleep(wait_seconds)
    response = {
        "TimeWaited": wait_seconds,
        "Id": id
    }
    cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn
    ServiceTimeout: 60
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
```

範例範本逐步解說

以下程式碼片段將說明範例範本的相關部分，協助您了解 Lambda 函式如何與自訂資源相關聯，並了解其輸出。

[AWS::Lambda::Function](#) 資源 CFNWaiter

`AWS::Lambda::Function` 資源會指定函式的原始程式碼、處理常式名稱、執行時期環境，以及執行角色的 Amazon Resource Name (ARN)。

由於使用 Python 原始程式碼，因此 Handler 屬性設定為 `index.handler`。如需有關使用內嵌函數原始程式碼時可接受處理常式識別碼的詳細資訊，請參閱 [AWS::Lambda::Function Code](#)。

由於原始檔是 Python 程式碼，Runtime 指定為 `python3.9`。

Timeout 設定為 900 秒。

Role 屬性會使用 `Fn::GetAtt` 函數來擷取範本中 `AWS::IAM::Role` 資源所宣告的 `LambdaExecutionRole` 執行角色 ARN。

Code 屬性會使用 Python 函數來內嵌定義函數程式碼。範例範本中的 Python 函式會執行下列動作：

- 使用 UUID 建立唯一 ID
- 檢查請求是建立請求還是更新請求
- 在 Create 或 Update 請求期間為 `ServiceTimeout` 指定的持續時間休眠
- 傳回等待時間及唯一 ID

JSON

```
...
  "CFNWaiter": {
    "Type": "AWS::Lambda::Function",
    "Properties": {
      "Handler": "index.handler",
      "Runtime": "python3.9",
      "Timeout": 900,
      "Role": { "Fn::GetAtt": ["LambdaExecutionRole", "Arn"] },
      "Code": {
        "ZipFile": { "Fn::Join": ["\n", [
          "from time import sleep",
          "import json",
          "import cfnresponse",
          "import uuid",
          "",
          "def handler(event, context):",
          "    wait_seconds = 0",
          "    id = str(uuid.uuid1())",
          "    if event[\"RequestType\"] in [\"Create\", \"Update\"]:",
          "        wait_seconds = int(event[\"ResourceProperties\"].get(\"ServiceTimeout\", 0))",
          "        sleep(wait_seconds)",
```

```

        " response = {",
        "   \"TimeWaited\": wait_seconds,",
        "   \"Id\": id ",
        " }",
        " cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
\"Waiter-\"+id)"
    ]}]
    }
}
},
...

```

YAML

```

...
CFNWaiter:
  Type: AWS::Lambda::Function
  Properties:
    Handler: index.handler
    Runtime: python3.9
    Timeout: 900
    Role: !GetAtt LambdaExecutionRole.Arn
    Code:
      ZipFile:
        !Sub |
        from time import sleep
        import json
        import cfnresponse
        import uuid

        def handler(event, context):
            wait_seconds = 0
            id = str(uuid.uuid1())
            if event["RequestType"] in ["Create", "Update"]:
                wait_seconds = int(event["ResourceProperties"].get("ServiceTimeout", 0))
                sleep(wait_seconds)
            response = {
                "TimeWaited": wait_seconds,
                "Id": id
            }
            cfnresponse.send(event, context, cfnresponse.SUCCESS, response,
"Waiter-"+id)
...

```

[AWS::IAM::Role](#) 資源 LambdaExecutionRole

`AWS::IAM::Role` 資源會為 Lambda 函式建立執行角色，其中包含允許 Lambda 使用該角色的擔任角色政策。此外，它還包含一個允許存取 CloudWatch Logs 的政策。

JSON

```
...
  "LambdaExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [{
          "Effect": "Allow",
          "Principal": { "Service": ["lambda.amazonaws.com"] },
          "Action": ["sts:AssumeRole"]
        }]
      },
      "Path": "/",
      "Policies": [{
        "PolicyName": "AllowLogs",
        "PolicyDocument": {
          "Statement": [{
            "Effect": "Allow",
            "Action": ["logs:*"],
            "Resource": "*"
          }]
        }
      ]
    }
  },
  ...
```

YAML

```
...
LambdaExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: "Allow"
```

```

Principal:
  Service:
    - "lambda.amazonaws.com"
  Action:
    - "sts:AssumeRole"
Path: "/"
Policies:
  - PolicyName: "AllowLogs"
    PolicyDocument:
      Statement:
        - Effect: "Allow"
          Action:
            - "logs:*"
          Resource: "*"
...

```

[AWS::CloudFormation::CustomResource](#) 資源 CFNWaiterCustomResource

自訂資源會透過 `!GetAtt CFNWaiter.Arn`，使用 Lambda 函式的 ARN 與其建立關聯。根據 `ServiceTimeout` 中的設定，它會為建立和更新操作實作 60 秒的等待時間。僅當屬性被修改時，才會在更新操作中調用此資源。

JSON

```

...
"CFNWaiterCustomResource": {
  "Type": "AWS::CloudFormation::CustomResource",
  "Properties": {
    "ServiceToken": { "Fn::GetAtt": ["CFNWaiter", "Arn"] },
    "ServiceTimeout": 60
  }
},
...

```

YAML

```

...
CFNWaiterCustomResource:
  Type: AWS::CloudFormation::CustomResource
  Properties:
    ServiceToken: !GetAtt CFNWaiter.Arn

```



```
ServiceTimeout: 60
```

```
...
```

Outputs

此範本的 Outputs 為 TimeWaited 和 WaiterId。TimeWaited 數值會使用 Fn::GetAtt 函數，提供等待程式資源實際等待的時間。WaiterId 會使用 Fn::GetAtt 函數來提供產生並與執行相關聯的唯一 ID。

JSON

```
...
"Outputs": {
  "TimeWaited": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "TimeWaited"] },
    "Export": { "Name": "TimeWaited" }
  },
  "WaiterId": {
    "Value": { "Fn::GetAtt": ["CFNWaiterCustomResource", "Id"] },
    "Export": { "Name": "WaiterId" }
  }
}
}
```

YAML

```
...
Outputs:
  TimeWaited:
    Value: !GetAtt CFNWaiterCustomResource.TimeWaited
    Export:
      Name: TimeWaited
  WaiterId:
    Value: !GetAtt CFNWaiterCustomResource.Id
    Export:
      Name: WaiterId
...
```

先決條件

您必須具備使用所有對應服務 (如 Lambda 和 CloudFormation) 的 IAM 許可。

啟動堆疊

建立堆疊

1. 從 [範例範本](#) 區段尋找您偏好設定的範本 (YAML 或 JSON)，並將其儲存到名為 `samplelambdabackedcustomresource.template` 的機器。
2. 開啟位在 <https://console.aws.amazon.com/cloudformation/> 的 CloudFormation 主控台。
3. 從堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。
4. 對於先決條件 - 準備範本，選取選擇現有範本。
5. 對於指定範本下，選擇上傳範本檔案，然後選擇選擇檔案。
6. 選取您先前儲存的 `samplelambdabackedcustomresource.template` 範本檔案。
7. 選擇下一步。
8. 對於秘密名稱，輸入 **SampleCustomResourceStack**，然後選擇下一步。
9. 在此逐步解說中，您不需要新增標籤或指定進階設定，因此請選擇 Next (下一步)。
10. 確定堆疊名稱正確，然後選擇更新。

CloudFormation 可能需要幾分鐘的時間來建立您的堆疊。若要監控進度，請檢視堆疊事件。如需詳細資訊，請參閱[在 CloudFormation 主控台中檢視堆疊資訊](#)。

若堆疊建立成功，則系統會一併建立堆疊中所有資源 (如 Lambda 函式和自訂資源)。您已成功使用 Lambda 函式和自訂資源。

若 Lambda 函式回傳錯誤訊息，則可以在 CloudWatch Logs 的[主控台](#)檢視該函數的日誌。日誌串流名稱即為自訂資源的實體 ID，您能夠檢視堆疊資源以查詢該名稱。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的[檢視日誌資料](#)。

清除資源

您可以刪除堆疊以清除建立的所有堆疊資源，便無需為不必要的資源支付費用。

刪除堆疊

1. 從 CloudFormation 主控台選擇 SampleCustomResourceStack 堆疊。

2. 選擇 Actions (動作)，然後選擇 Delete Stack (刪除堆疊)。
3. 在確認訊息中，選擇 Yes, Delete (是，刪除)。

系統將刪除您建立的所有資源。

現在，您已經了解如何建立並使用 Lambda 支援的自訂資源，即可善用本逐步說明的範例範本和程式碼來建置其他堆疊及函數。

相關資訊

- [CloudFormation 自訂資源參考](#)
- [AWS::CloudFormation::CustomResource](#)

cfn-response 模組

在 CloudFormation 範本中，您可以將 Lambda 函式指定為自訂資源的目標。使用 ZipFile 屬性來指定[函數的](#)原始程式碼時，可以載入 cfn-response 模組，以將回應從 Lambda 函式傳送至自訂資源。cfn-response 模組是一個程式庫，可簡化將回應傳送至調用 Lambda 函式的自訂資源。此模組具有 send 方法，它可透過 Amazon S3 預先簽章的 URL (ResponseURL) 將[回應物件](#)傳送給自訂資源。

只有在您使用 cfn-response 屬性撰寫來源碼時，才能使用 ZipFile 模組。這不適用於 Amazon S3 儲存貯體中所存放的來源碼。針對儲存貯體中的程式碼，您必須撰寫自己的函數來傳送回應。

Note

執行 send 方法之後，Lambda 函數即會終止，因此會忽略您在該方法之後撰寫的任何內容。

載入 cfn-response 模組

對於 Node.js 函數，使用 require() 函數來載入 cfn-response 模組。例如，下列程式碼範例會建立名為 cfn-response 的 response 物件：

```
var response = require('cfn-response');
```

對於 Python，使用 import 陳述式載入 cfnresponse 模組，如下列範例所示：

Note

使用此確切 import 陳述式。如果您使用 import 陳述式的其他變體，則 CloudFormation 不會包括回應模組。

```
import cfnresponse
```

send 方法參數

您可以搭配使用下列參數與 send 方法。

event

[自訂資源請求](#)中的欄位。

context

Lambda 函數特有的物件，可用來指定函數和任何回呼完成執行的時機，或存取 Lambda 執行環境內的資訊。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的[使用 Node.js 建立 Lambda 函式](#)。

responseStatus

函數是否成功完成。使用 cfnresponse 模組常數指定狀態：SUCCESS 表示執行成功，而 FAILED 表示執行失敗。

responseData

自訂資源[回應物件](#)的 Data 欄位。資料是名稱/值組清單。

physicalResourceId

選用。已呼叫函數之自訂資源的唯一識別碼。根據預設，模組會使用與 Lambda 函數建立關聯之 Amazon CloudWatch Logs 日誌串流的名稱。

傳回的 PhysicalResourceId 值可以變更自訂資源更新操作。如果傳回的值相同，則視為正常更新。若傳回的值不同，CloudFormation 會將該更新視為取代，並傳送刪除請求給舊資源。如需詳細資訊，請參閱[AWS::CloudFormation::CustomResource](#)。

noEcho

選用。指示在使用 `Fn::GetAtt` 函數擷取自訂資源時是否要遮蔽其輸出。如果設定為 `true`，則所有傳回的值都會以星號 (`*****`) 遮罩，但儲存在下列指定位置的資訊除外。根據預設，此值為 `false`。

Important

使用 `NoEcho` 屬性不會遮罩任何儲存在下列資訊中的資訊：

- Metadata 範本區段。CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。若要取得更多資訊，請參閱[中繼資料](#)。
 - Outputs 範本區段。如需詳細資訊，請參閱[輸出](#)。
 - 資源定義的 Metadata 屬性。如需詳細資訊，請參閱[Metadata 屬性](#)。
- 我們強烈建議您不要使用這些機制來包含敏感資訊，例如密碼或秘密。

如需使用 `NoEcho` 遮罩敏感資訊的詳細資訊，請參閱[請勿在您的範本中內嵌憑證](#) 最佳實務。

範例

Node.js

在下列 Node.js 範例中，內嵌 Lambda 函數採用輸入值，並將它乘以 5。內嵌函數特別適用於較小的函數，因為它們可讓您直接在範本中指定來源碼，而不是建立套件並將它上傳至 Amazon S3 儲存貯體。此函數使用 `cfn-response send` 方法，將結果送回給呼叫它的自訂資源。

JSON

```
"ZipFile": { "Fn::Join": ["", [
  "var response = require('cfn-response');",
  "exports.handler = function(event, context) {",
  "  var input = parseInt(event.ResourceProperties.Input);",
  "  var responseData = {Value: input * 5};",
  "  response.send(event, context, response.SUCCESS, responseData);",
  "};",
]] }
```

YAML

```
ZipFile: >
```

```
var response = require('cfn-response');
exports.handler = function(event, context) {
    var input = parseInt(event.ResourceProperties.Input);
    var responseData = {Value: input * 5};
    response.send(event, context, response.SUCCESS, responseData);
};
```

Python

在下列 Python 範例中，內嵌 Lambda 函數採用整數值，並將它乘以 5。

JSON

```
"ZipFile" : { "Fn::Join" : ["\n", [
    "import json",
    "import cfnresponse",
    "def handler(event, context):",
    "    responseValue = int(event['ResourceProperties']['Input']) * 5",
    "    responseData = {}",
    "    responseData['Data'] = responseValue",
    "    cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,",
    "\"CustomResourcePhysicalID\"")
]]}
```

YAML

```
ZipFile: |
    import json
    import cfnresponse
    def handler(event, context):
        responseValue = int(event['ResourceProperties']['Input']) * 5
        responseData = {}
        responseData['Data'] = responseValue
        cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
            "CustomResourcePhysicalID")
```

模組來源碼

主題

- [非同步 Node.js 原始程式碼](#)
- [Node.js 原始程式碼](#)
- [Python 原始碼](#)

非同步 Node.js 原始程式碼

如果處理常式為異步，以下是 Node.js 函數的回應模組原始程式碼。請檢閱它以了解模組功能，並協助您實作您自己的回應函數。

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    return new Promise((resolve, reject) => {
        var responseBody = JSON.stringify({
            Status: responseStatus,
            Reason: "See the details in CloudWatch Log Stream: " +
context.logStreamName,
            PhysicalResourceId: physicalResourceId || context.logStreamName,
            StackId: event.StackId,
            RequestId: event.RequestId,
            LogicalResourceId: event.LogicalResourceId,
            NoEcho: noEcho || false,
            Data: responseData
        });

        console.log("Response body:\n", responseBody);

        var https = require("https");
        var url = require("url");

        var parsedUrl = url.parse(event.ResponseURL);
        var options = {
            hostname: parsedUrl.hostname,
            port: 443,
            path: parsedUrl.path,
            method: "PUT",
            headers: {
                "content-type": "",
                "content-length": responseBody.length
            }
        };

    });
```

```

    var request = https.request(options, function(response) {
        console.log("Status code: " + parseInt(response.statusCode));
        resolve(context.done());
    });

    request.on("error", function(error) {
        console.log("send(..) failed executing https.request(..): " +
maskCredentialsAndSignature(error));
        reject(context.done(error));
    });

    request.write(responseBody);
    request.end();
  })
}

function maskCredentialsAndSignature(message) {
    return message.replace(/X-Amz-Credential=[^\s]+/i, 'X-Amz-Credential=*****')
        .replace(/X-Amz-Signature=[^\s]+/i, 'X-Amz-Signature=*****');
}

```

Node.js 原始程式碼

如果處理常式不是異步，以下是 Node.js 函數的回應模組原始程式碼。請檢閱它以了解模組功能，並協助您實作您自己的回應函數。

```

// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: MIT-0

exports.SUCCESS = "SUCCESS";
exports.FAILED = "FAILED";

exports.send = function(event, context, responseStatus, responseData,
    physicalResourceId, noEcho) {

    var responseBody = JSON.stringify({
        Status: responseStatus,
        Reason: "See the details in CloudWatch Log Stream: " + context.logStreamName,
        PhysicalResourceId: physicalResourceId || context.logStreamName,
        StackId: event.StackId,
        RequestId: event.RequestId,
        LogicalResourceId: event.LogicalResourceId,
        NoEcho: noEcho || false,
    });
}

```



```
        Data: responseData
    });

    console.log("Response body:\n", responseBody);

    var https = require("https");
    var url = require("url");

    var parsedUrl = url.parse(event.ResponseURL);
    var options = {
        hostname: parsedUrl.hostname,
        port: 443,
        path: parsedUrl.path,
        method: "PUT",
        headers: {
            "content-type": "",
            "content-length": responseBody.length
        }
    };

    var request = https.request(options, function(response) {
        console.log("Status code: " + parseInt(response.statusCode));
        context.done();
    });

    request.on("error", function(error) {
        console.log("send(..) failed executing https.request(..): " +
            maskCredentialsAndSignature(error));
        context.done();
    });

    request.write(responseBody);
    request.end();
}
```

Python 原始碼

以下是 Python 函數的回應模組來源程式碼：

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0

from __future__ import print_function
import urllib3
```

```
import json
import re

SUCCESS = "SUCCESS"
FAILED = "FAILED"

http = urllib3.PoolManager()

def send(event, context, responseStatus, responseData, physicalResourceId=None,
noEcho=False, reason=None):
    responseUrl = event['ResponseURL']

    responseBody = {
        'Status' : responseStatus,
        'Reason' : reason or "See the details in CloudWatch Log Stream:
{}".format(context.log_stream_name),
        'PhysicalResourceId' : physicalResourceId or context.log_stream_name,
        'StackId' : event['StackId'],
        'RequestId' : event['RequestId'],
        'LogicalResourceId' : event['LogicalResourceId'],
        'NoEcho' : noEcho,
        'Data' : responseData
    }

    json_responseBody = json.dumps(responseBody)

    print("Response body:")
    print(json_responseBody)

    headers = {
        'content-type' : '',
        'content-length' : str(len(json_responseBody))
    }

    try:
        response = http.request('PUT', responseUrl, headers=headers,
body=json_responseBody)
        print("Status code:", response.status)

    except Exception as e:
```

```
print("send(..) failed executing http.request(..):",
      mask_credentials_and_signature(e))

def mask_credentials_and_signature(message):
    message = re.sub(r'X-Amz-Credential=[^\s]+' , 'X-Amz-Credential=*****', message,
                     flags=re.IGNORECASE)
    return re.sub(r'X-Amz-Signature=[^\s]+' , 'X-Amz-Signature=*****', message,
                  flags=re.IGNORECASE)
```

使用範本巨集在 CloudFormation 範本上執行自訂處理

透過巨集，讓可您在範本上執行自訂處理，例如尋找和取代操作這種簡單動作，乃至於全部範本的大規模轉換。

若要了解可能性的廣度，請考慮使用 `AWS::Include` 和 `AWS::Serverless` 轉換 (由 CloudFormation 託管的巨集)：

- [AWS::Include 轉換](#) 可讓您將樣板式範本程式碼片段插入範本中。
- [AWS::Serverless 轉換](#) 採用以無 AWS 伺服器應用程式模型 (AWS SAM) 語法編寫的整個範本，並將其轉換並擴展為合規的 CloudFormation 範本。如需無伺服器應用程式的詳細資訊 AWS SAM，請參閱 [AWS Serverless Application Model 開發人員指南](#)。

主題

- [帳單](#)
- [巨集範例](#)
- [相關資源](#)
- [CloudFormation 巨集概觀](#)
- [建立 CloudFormation 巨集定義](#)
- [簡單字串取代巨集範](#)
- [疑難排解處理過的範本](#)

帳單

當巨集執行時，Lambda 函數的擁有者需要支付執行該函數的任何相關費用。

`AWS::Include` 和 `AWS::Serverless` 轉換是由 CloudFormation 託管的巨集。可免費使用它們。

巨集範例

除了本區段中的此範例外，您還可以在我們的 [GitHub 程式碼庫](#) 中找到包含原始程式碼和範本的巨集範例。這些範例「按現狀」提供作為教學用途。

相關資源

- [AWS::CloudFormation::Macro](#)
- [CloudFormation 範本 Transform 區段](#)
- [Fn::Transform](#)
- [AWS::Serverless 轉換](#)
- [AWS::Include 轉換](#)

CloudFormation 巨集概觀

使用巨集處理範本有兩個主要步驟：建立巨集本身，然後使用巨集在您的範本上執行處理。

若要建立巨集定義，您必須建立下列項目：

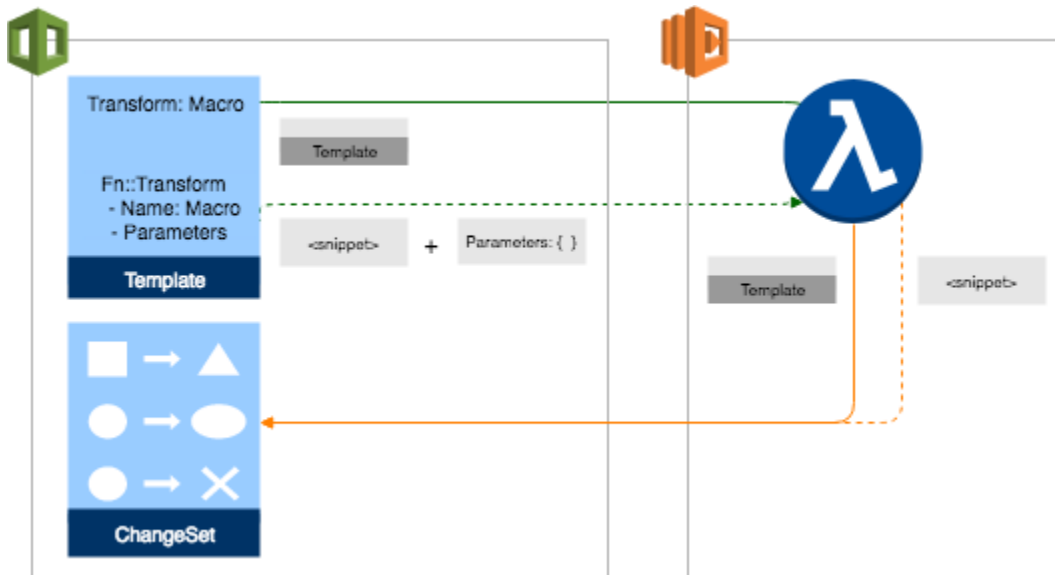
- 用來執行範本處理的 Lambda 函式。此 Lambda 函數接受程式碼片段或整個範本，以及您定義的任何其他參數。它會傳回已處理的範本程式碼片段或整個範本作為回應。
- [AWS::CloudFormation::Macro](#) 類型的資源，可讓使用者從 CloudFormation 範本內呼叫 Lambda 函式。此資源指定針對此巨集叫用的 Lambda 函數的 ARN，以及可協助除錯的其他選用屬性。若要在帳戶中建立此資源，請撰寫包含 `AWS::CloudFormation::Macro` 資源的範本，然後從範本建立具有自我管理許可的堆疊或堆疊集。CloudFormation StackSets 目前不支援從參考巨集的範本建立或更新具有服務管理許可的堆疊集。

若要使用巨集，請在範本中參考巨集：

- 若要處理範本的區段或部分，請在 `Fn::Transform` 函數 (位置相對於您想要轉換的範本內容) 中參考巨集。當您使用 `Fn::Transform` 時，您也可以傳遞它需要的任何指定參數。
- 若要處理整個範本，請在範本的 [Transform](#) 區段中參考巨集。

接著，您通常會建立變更集，然後加以執行。(處理巨集可能會新增您不了解的多項資源。為了確保您知道巨集所產生的所有變更，我們強烈建議您使用變更集。) CloudFormation 會將指定的範本內容以及任何額外的指定參數，傳遞到巨集資源中指定的 Lambda 函式。Lambda 函數會傳回已處理的範本內容，不論是程式碼片段或整個範本。

所有巨集之後，CloudFormation 會產生已稱為在範本中變更設定，包含處理範本的內容。檢閱變更集之後，請執行它來套用變更。



如何直接建立堆疊

若要使用參考巨集的範本來建立或更新堆疊，您通常會建立變更集，然後執行它。變更集說明 CloudFormation 將根據處理過的範本所採取的動作。處理巨集可能會新增您不了解的多個資源。為了確保您知道巨集所產生的所有變更，我們強烈建議您使用變更集。檢閱變更集之後，您可以執行它來確實套用變更。

巨集可以將 IAM 資源新增至您的範本。針對這些資源，CloudFormation 會要求您[確認其功能](#)。因為 CloudFormation 無法知道在處理範本之前新增哪些資源，所以您建立變更集時可能需要確認 IAM 功能 (視參考的巨集是否包含 IAM 資源而定)。因此，當您執行變更集時，CloudFormation 具有必要功能來建立 IAM 資源。

要透過處理範本建立或更新堆疊，無需先在變更集中審核建議的變更，可在 `CreateStack` 或 `UpdateStack` 請求期間指定 `CAPABILITY_AUTO_EXPAND` 功能。如果您知道巨集執行哪些處理，您應只直接透過包含巨集的堆疊範本來建立堆疊。您不能將變更集與堆疊集巨集一起使用；您必須直接更新堆疊集。

如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateStack](#) 或 [UpdateStack](#)。

Important

如果您的堆疊集範本參考了一或多個巨集，您必須直接從處理的範本建立堆疊集，而不必先檢視變更集中產生的變更。處理巨集可能會新增您不了解的多個資源。從直接參考巨集的範本建立或更新堆疊集之前，請確定您知道巨集會執行哪些處理操作。

若要減少從參考巨集的範本啟動堆疊的步驟數目，您可以使用 `package` 和 `deploy` AWS CLI 命令。如需詳細資訊，請參閱[使用 將本機成品上傳至 S3 儲存貯體 AWS CLI](#)及[建立包含轉換的堆疊](#)。

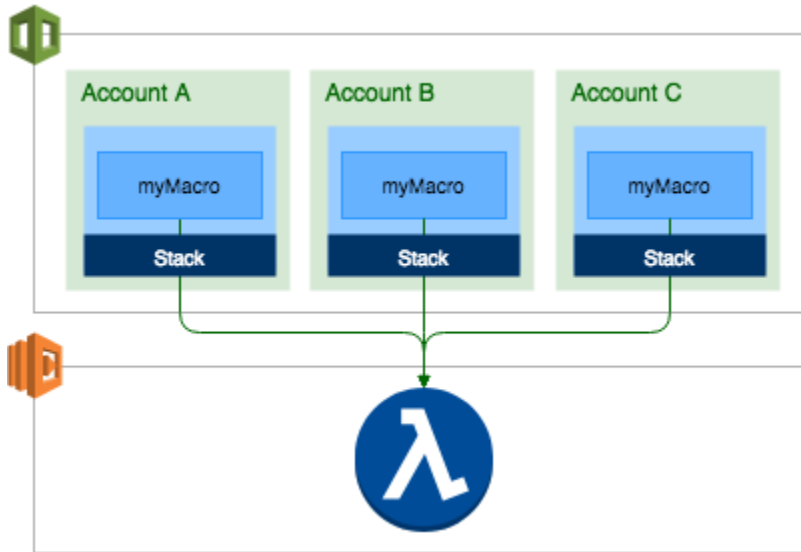
考量事項

配合使用巨集時，請記住下列考量事項和限制：

- 只有提供 Lambda AWS 區域 的 才支援巨集。如需可使用 Lambda 的區域清單，請參閱 [AWS Lambda 端點與配額](#)。
- 任何處理的範本程式碼片段都必須是有效的 JSON。
- 任何處理的範本程式碼片段都必須通過建立堆疊、更新堆疊、建立堆疊集或更新堆疊集操作的驗證檢查。
- CloudFormation 會先解析巨集，再處理範本。產生的範本必須是有效的 JSON，且不能超過範本大小上限。
- 由於 CloudFormation 處理範本中元素的順序，巨集無法將模組包含在傳回至 CloudFormation 的處理範本內容中。如需詳細資訊，請參閱[巨集評估順序](#)。
- CloudFormation 會採用原始範本的副本來執行更新復原功能。即使加入的程式碼片段有所變更，它也會復原至原始範本。
- 在巨集內包含巨集沒有作用，因為我們不會處理以遞迴方式處理巨集。
- 巨集內目前不支援 `Fn::ImportValue` 內部函數。
- 範本包含的內部函數是在任何巨集之後評估。因此，巨集傳回的已處理範本內容可能包含內部函數的呼叫，一樣照常評估。
- StackSets 目前不支援從參考 CloudFormation 巨集的範本建立或更新具有服務受管許可的 StackSets。

巨集帳戶範圍和許可

巨集只能在它們已建立為資源的帳戶中使用。巨集的名稱在給定帳戶內必須是唯一的。不過，您可以在基礎 Lambda 函數上啟用跨帳戶存取，然後建立巨集定義在多個帳戶中參考該函數，就能讓多個帳戶使用相同的功能。在下列範例中，三個帳戶包含巨集定義，各指向相同的 Lambda 函數。



為了建立巨集定義，使用者必須有許可，才能在指定的帳戶內建立堆疊或堆疊集。

為了讓 CloudFormation 成功執行範本中包含的巨集，使用者必須有基礎 Lambda 函式的 Invoke 許可。為避免潛在提升許可，CloudFormation 會在執行巨集時模擬使用者。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的[在 AWS Lambda 中管理許可](#)，以及《服務授權參考》中的[AWS Lambda 的動作、資源和條件索引鍵](#)。

建立 CloudFormation 巨集定義

當您建立巨集定義時，巨集定義會在指定的帳戶中提供基礎 Lambda 函式，供 CloudFormation 調用以處理範本。

事件映射

當 CloudFormation 調用巨集的 Lambda 函式時，會傳送 JSON 格式的請求，其結構如下：

```
{
  "region" : "us-east-1",
  "accountId" : "$ACCOUNT_ID",
  "fragment" : { ... },
  "transformId" : "$TRANSFORM_ID",
  "params" : { ... },
```

```
"requestId" : "$REQUEST_ID",  
"templateParameterValues" : { ... }  
}
```

- **region**

巨集所在的區域。

- **accountId**

供巨集用來叫用 Lambda 函數的帳戶的帳戶 ID。

- **fragment**

可供自訂處理的範本內容 (JSON 格式)。

- 對於 Transform 範本區段包含的巨集，這是整個範本 (Transform 區段除外)。
- 對於 Fn::Transform 內部函數呼叫包含的巨集，這包括以內部函數 (Fn::Transform 函數除外) 在範本內的位置為基礎的所有同級節點 (及其子系)。如需詳細資訊，請參閱[巨集的範本範圍](#)。

- **transformId**

叫用此函數的巨集的名稱。

- **params**

若為 Fn::Transform 函數呼叫，則為該函數指定的任何參數。CloudFormation 不會在將這些參數傳遞至 函數之前對其進行評估。

對於 Transform 範本區段包含的巨集，這個區段是空的。

- **requestId**

叫用此函數的請求的 ID。

- **templateParameterValues**

範本 [Parameters](#) 區段中指定的任何參數。CloudFormation 會在將這些參數傳遞至函數之前對其進行評估。

回應格式

CloudFormation 期望 Lambda 函式傳回的回應是下列 JSON 格式：

```
{
```



```
"requestId" : "$REQUEST_ID",  
"status" : "$STATUS",  
"fragment" : { ... },  
"errorMessage": "optional error message for failures"  
}
```

- **requestId**

叫用此函數的請求的 ID。這必須符合 CloudFormation 在調用函數時提供的請求 ID。

- **status**

請求的狀態 (不區分大小寫)。應設為 `success`。CloudFormation 會將任何其他回應視為失敗。

- **fragment**

供 CloudFormation 納入處理後範本的已處理範本內容，含同級項目。CloudFormation 會將傳遞給 Lambda 函式的範本內容取代為在 Lambda 回應中收到的範本片段。

處理過的範本內容必須是有效的 JSON，而納入所處理的範本中之後必須形成有效的範本。

如果您的函數不實際變更 CloudFormation 傳來的範本內容，但您仍然需要將該內容納入所處理的範本中，則函數必須在其回應中將該範本內容傳回給 CloudFormation。

- **errorMessage**

說明轉換失敗原因的錯誤訊息。CloudFormation 會在堆疊 Stack details (堆疊詳細資訊) 頁面的 Events (事件) 窗格中顯示此錯誤訊息。

例如：

```
Error creating change set: Transform  
    AWS ## account  
    number::macro name failed with:  
    error message string.
```

建立巨集定義

建立 CloudFormation 巨集定義

1. [建置 Lambda 函式](#)，用於處理範本內容的處理。它可以處理範本的任何部分，甚至整個範本。

2. 建立包含 `AWS::CloudFormation::Macro` 資源類型的 CloudFormation 範本，並指定 `Name` 和 `FunctionName` 屬性。`FunctionName` 屬性必須包含當 CloudFormation 執行巨集時調用的 Lambda 函式的 ARN。
3. (選用) 為了協助除錯，當您為巨集建立 `AWS::CloudFormation::Macro` 資源類型時，您也可以指定 `LogGroupName` 和 `LogRoleArn` 屬性。這些屬性可讓您指定 CloudWatch Logs 日誌群組，供 CloudFormation 在調用巨集的基礎 Lambda 函式時傳送錯誤日誌記錄資訊，也可指定 CloudFormation 將日誌項目傳送到這些日誌時應該擔任的角色。
4. 在您要使用它的帳戶中，使用具有巨集的範本[建立堆疊](#)。或者，在管理員帳戶中使用包含該巨集的範本，[建立具有自我管理許可的堆疊集](#)，然後在目標帳戶中建立堆疊執行個體。
5. 當 CloudFormation 成功建立包含巨集定義的堆疊之後，即可在這些帳戶內使用巨集。在範本中，請在您要處理的範本內容的相對適當位置上參考巨集，以此來使用巨集。

巨集的範本範圍

範本的 Transform 區段中參考的巨集可以處理該範本的整個內容。

`Fn::Transform` 函數中參考的巨集可以處理該 `Fn::Transform` 函數在範本中的任何同級元素 (包括子系) 的內容。

例如，在以下範本範例中，`AWS::Include` 可以根據其所在 `Fn::Transform` 函數的位置，而處理所有 `MyBucket` 屬性。`MyMacro` 可以處理整個範本的內容，因為它已納入 Transform 區段中。

```
# Start of processable content for MyMacro
AWSTemplateFormatVersion: 2010-09-09
Transform: [MyMacro]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    # Start of processable content for AWS::Include
    Properties:
      BucketName: amzn-s3-demo-bucket1
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: 'AWS::Include'
          Parameters:
            Location: s3://amzn-s3-demo-bucket2/MyFileName.yaml
      CorsConfiguration: []
```

```
# End of processable content for AWS::Include
MyEc2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageID: ami-1234567890abcdef0
# End of processable content for MyMacro
```

巨集評估順序

您可以在給定的範本中參考多個巨集，包括由 CloudFormation 託管的轉換，例如 [AWS::Include](#) 和 [AWS::Serverless](#)。

巨集是根據在範本中的位置而依序評估，從最深巢狀往外到最通用。範本中同一位置的巨集是根據它們所列順序而按順序評估。

在動作順序和範圍方面，轉換 (例如 AWS::Include 和 AWS::Transform) 與任何其他巨集視為相同。

例如，在以下範本範例中，CloudFormation 會先評估 PolicyAdder 巨集，因為它是範本中最深層巢狀的巨集。然後，CloudFormation 會評估 MyMacro，再評估 AWS::Serverless，因為它在 Transform 區段中列在 AWS::Serverless 前面。

```
AWS::TemplateFormatVersion: 2010-09-09
Transform: [MyMacro, AWS::Serverless]
Resources:
  WaitCondition:
    Type: AWS::CloudFormation::WaitCondition
  MyBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: amzn-s3-demo-bucket
      Tags: [{"key": "value"}]
      'Fn::Transform':
        - Name: PolicyAdder
      CorsConfiguration: []
  MyEc2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageID: ami-1234567890abcdef0
```

簡單字串取代巨集範

以下範例逐步介紹如何使用巨集，包括在範本中定義巨集、為巨集建立 Lambda 函式，然後在範本中使用巨集。

在這個範例中，我們會建立簡單巨集，在處理的範本中插入指定的字串，以取代指定的目標內容。然後，我們會使用它在處理的範本中的指定位置插入空白 `WaitHandleCondition`。

建立巨集

使用巨集之前，我們必須先做兩件事：建立 Lambda 函數來執行所需的範本處理，然後建立巨集定義，讓該 Lambda 函數可供 CloudFormation 使用。

以下範例範本包含範例巨集的定義。若要讓巨集可在特定 中使用 AWS 帳戶，請從範本建立堆疊。巨集定義會指定巨集名稱、簡短描述，並於範本中使用此巨集時參考 CloudFormation 叫用的 Lambda 函數的 ARN。(我們未包含用於錯誤記錄的 `LogGroupName` 或 `LogRoleARN` 屬性。)

在這個範例中，假設從這個範本建立的堆疊名為 `JavaMacroFunc`。由於巨集 `Name` 屬性設為堆疊名稱，產生的巨集也名為 `JavaMacroFunc`。

```
AWS::CloudFormation::Macro
Resources:
  Macro:
    Type: AWS::CloudFormation::Macro
    Properties:
      Name: !Sub '${AWS::StackName}'
      Description: Adds a blank WaitConditionHandle named WaitHandle
      FunctionName: 'arn:aws:lambda:us-east-1:012345678910:function:JavaMacroFunc'
```

使用巨集

為了使用我們的巨集，我們將使用 `Fn::Transform` 內建函數將它納入範本中。

當我們使用以下範本建立堆疊時，CloudFormation 會呼叫我們的範例巨集。基礎 Lambda 函數會將一個指定的字串替換成另一個指定的字串。在此案例中，結果是在所處理的範本中插入空白 `AWS::CloudFormation::WaitConditionHandle`。

```
Parameters:
  ExampleParameter:
    Type: String
    Default: 'SampleMacro'
```

```
Resources:
  2a:
    Fn::Transform:
      Name: "JavaMacroFunc"
      Parameters:
        replacement: 'AWS::CloudFormation::WaitConditionHandle'
        target: '$$REPLACEMENT$$'
      Type: '$$REPLACEMENT$$'
```

- 要調用的巨集指定為 JavaMacroFunc，這是來自上一個巨集定義範例。
- 有兩個參數傳遞給巨集：target 和 replacement，代表目標字串及其所需的替換值。
- 巨集可以處理 Type 節點的內容，因為 Type 是參考巨集的 Fn::Transform 函數的同級項目。
- 產生的 AWS::CloudFormation::WaitConditionHandle 名為 2a。
- 範本還包含巨集也可存取的範本參數：ExampleParameter (但在這個案例中未使用)。

Lambda 輸入資料

當 CloudFormation 在堆疊建立期間處理我們的範例範本時，它會將下列事件映射傳遞到 JavaMacroFunc 巨集定義中參考的 Lambda 函式。

- region: us-east-1
- accountId: 012345678910
- fragment:

```
{
  "Type": "$$REPLACEMENT$$"
}
```

- transformId: 012345678910::JavaMacroFunc
- params:

```
{
  "replacement": "AWS::CloudFormation::WaitConditionHandle",
  "target": "$$REPLACEMENT$$"
}
```

- requestId: 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- templateParameterValues:

```
{
  "ExampleParameter": "SampleMacro"
}
```

請注意，fragment 包含的 JSON 代表巨集可處理的範本片段。此片段由 `Fn::Transform` 函數呼叫的同級項目組成，而不是函數呼叫本身。此外，params 包含代表巨集參數的 JSON。在此案例中是 replacement 和 target。同樣地，templateParameterValues 包含的 JSON 代表針對整個範本指定的參數。

Lambda 函式程式碼

以下是 JavaMacroFunc 範例巨集的基礎 Lambda 函式的實際程式碼。它會逐一查看回應 (不論是字串、清單或映射格式) 包含的範本片段，尋找指定的目標字串。如果找到指定的目標字串，Lambda 函數會以指定的替換字串取代目標字串。如果找不到，該函數會將範本片段保持不變。然後，該函數會將預期屬性的映射 (以下詳細討論) 傳回給 CloudFormation。

```
package com.macroexample.lambda.demo;

import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

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

public class LambdaFunctionHandler implements RequestHandler<Map<String, Object>,
    Map<String, Object>> {

    private static final String REPLACEMENT = "replacement";
    private static final String TARGET = "target";
    private static final String PARAMS = "params";
    private static final String FRAGMENT = "fragment";
    private static final String REQUESTID = "requestId";
    private static final String STATUS = "status";
    private static final String SUCCESS = "SUCCESS";
    private static final String FAILURE = "FAILURE";

    @Override
    public Map<String, Object> handleRequest(Map<String, Object> event, Context
context) {
```

```
// TODO: implement your handler
final Map<String, Object> responseMap = new HashMap<String, Object>();
responseMap.put(REQUESTID, event.get(REQUESTID));
responseMap.put(STATUS, FAILURE);
try {
    if (!event.containsKey(PARAMS)) {
        throw new RuntimeException("Params are required");
    }

    final Map<String, Object> params = (Map<String, Object>) event.get(PARAMS);
    if (!params.containsKey(REPLACEMENT) || !params.containsKey(TARGET)) {
        throw new RuntimeException("replacement or target under Params are
required");
    }

    final String replacement = (String) params.get(REPLACEMENT);
    final String target = (String) params.get(TARGET);
    final Object fragment = event.getOrDefault(FRAGMENT, new HashMap<String,
Object>());
    final Object retFragment;
    if (fragment instanceof String) {
        retFragment = iterateAndReplace(replacement, target, (String) fragment);
    } else if (fragment instanceof List) {
        retFragment = iterateAndReplace(replacement, target, (List<Object>) fragment);
    } else if (fragment instanceof Map) {
        retFragment = iterateAndReplace(replacement, target, (Map<String, Object>)
fragment);
    } else {
        retFragment = fragment;
    }
    responseMap.put(STATUS, SUCCESS);
    responseMap.put(FRAGMENT, retFragment);
    return responseMap;
} catch (Exception e) {
    e.printStackTrace();
    context.getLogger().log(e.getMessage());
    return responseMap;
}

private Map<String, Object> iterateAndReplace(final String replacement, final
String target, final Map<String, Object> fragment) {
    final Map<String, Object> retFragment = new HashMap<String, Object>();
    final List<String> replacementKeys = new ArrayList<>();
```

```

        fragment.forEach((k, v) -> {
            if (v instanceof String) {
                retFragment.put(k, iterateAndReplace(replacement, target, (String)v));
            } else if (v instanceof List) {
                retFragment.put(k, iterateAndReplace(replacement, target, (List<Object>)v));
            } else if (v instanceof Map ) {
                retFragment.put(k, iterateAndReplace(replacement, target, (Map<String, Object>)
v));
            } else {
                retFragment.put(k, v);
            }
        });
        return retFragment;
    }

    private List<Object> iterateAndReplace(final String replacement, final String
target, final List<Object> fragment) {
        final List<Object> retFragment = new ArrayList<>();
        fragment.forEach(o -> {
            if (o instanceof String) {
                retFragment.add(iterateAndReplace(replacement, target, (String) o));
            } else if (o instanceof List) {
                retFragment.add(iterateAndReplace(replacement, target, (List<Object>) o));
            } else if (o instanceof Map) {
                retFragment.add(iterateAndReplace(replacement, target, (Map<String, Object>)
o));
            } else {
                retFragment.add(o);
            }
        });
        return retFragment;
    }

    private String iterateAndReplace(final String replacement, final String target,
final String fragment) {
        System.out.println(replacement + " == " + target + " == " + fragment );
        if (fragment != null AND_AND fragment.equals(target))
            return replacement;
        return fragment;
    }
}

```


Lambda 函式回應

以下是 Lambda 函式傳回給 CloudFormation 來處理的映射。

- `requestId` : 5dba79b5-f117-4de0-9ce4-d40363bfb6ab
- `status` : SUCCESS
- `fragment` :

```
{
  "Type": "AWS::CloudFormation::WaitConditionHandle"
}
```

`requestId` 符合 CloudFormation 傳來的值，而 `status` 值 SUCCESS 表示 Lambda 函數成功處理請求中包含的範本片段。在這個回應中，`fragment` 包含的 JSON 代表在所處理的範本中插入的內容，用以取代原始範本程式碼片段。

產生的已處理範本

CloudFormation 收到 Lambda 函數的成功回應之後，就會將傳回的範本片段插入所處理的範本中。

以下是我們的範例產生的處理過範本。已不包含參考 `JavaMacroFunc` 巨集的 `Fn::Transform` 內建函數呼叫。Lambda 函數傳回的範本片段會納入適當的位置中，結果是內容 `"Type": "$REPLACEMENT$"` 已替換成 `"Type": "AWS::CloudFormation::WaitConditionHandle"`。

```
{
  "Parameters": {
    "ExampleParameter": {
      "Default": "SampleMacro",
      "Type": "String"
    }
  },
  "Resources": {
    "2a": {
      "Type": "AWS::CloudFormation::WaitConditionHandle"
    }
  }
}
```

疑難排解處理過的範本

使用巨集時，可在 CloudFormation 主控台中找到處理過的範本。

範本的階段會標示其處理狀態：

- **Original**：使用者最初提交以建立或更新堆疊或堆疊集的範本。
- **Processed**：由 CloudFormation 在處理任何參考的巨集後用來建立或更新堆疊或堆疊集的範本。即使原始範本是 YAML 格式，處理過的範本會變成 JSON 格式。

疑難排解時，請使用處理過的範本。如果範本未參考巨集，則原始範本和處理過的範本會相同。

如需詳細資訊，請參閱[在 CloudFormation 主控台中檢視堆疊資訊](#)。

若要使用 AWS CLI 取得已處理的範本，請使用 [get-template](#) 命令。

大小限制

已處理堆疊範本直接傳遞到 `CreateStack`、`UpdateStack` 或 `ValidateTemplate` 請求時的大小上限是 51,200 位元組，當使用 Amazon S3 範本的 URL 作為 S3 物件傳遞時的大小上限是 1 MB。不過，CloudFormation 在處理期間會更新範本的臨時狀態，因為它會按順序處理範本中包含的巨集。因此，處理期間的範本大小可能暫時超過完全處理範本允許的大小。CloudFormation 允許為這些處理中範本提供一些緩衝區。不過，您在設計範本和巨集時應謹記處理堆疊範本的允許大小上限。

如果 CloudFormation 傳回 `Transformation data limit exceeded` 錯誤，同時處理您的範本，則範本超過 CloudFormation 在處理期間允許的範本大小上限。

若要解決這個問題，請考量下列解決方案：

- 將範本重建為多個範本，以避免超過處理中範本的大小上限。例如：
 - 使用巢狀堆疊範本，來封裝部分範本。如需詳細資訊，請參閱[運用巢狀堆疊，將範本分割成可重複使用的部分](#)。
 - 建立多個堆疊，並使用跨堆疊參考來在這些堆疊之間交換資訊。如需詳細資訊，請參閱[參考在另一個 CloudFormation 堆疊中的資源輸出](#)。
- 減少特定巨集傳回的範本片段大小。CloudFormation 不會竄改巨集傳回的片段內容。

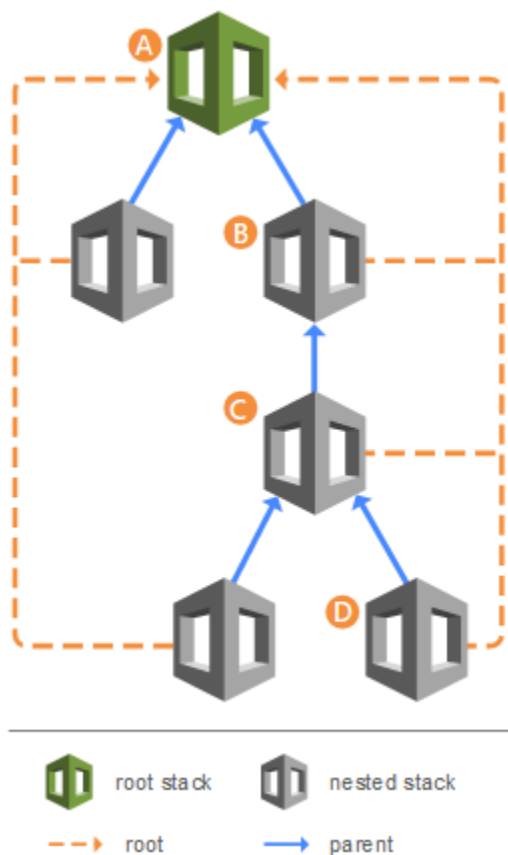
運用巢狀堆疊，將範本分割成可重複使用的部分

隨著基礎設施擴充，您可能會在多個範本中重複建立相同的資源組態。為避免這種重複性，您可以將這些通用組態拆分至專用範本中。然後，在其他範本中使用 [AWS::CloudFormation::Stack](#) 資源參考這些專用範本，建立巢狀堆疊。

例如，假設您有一個用於您大多數堆疊的負載平衡器組態。相較於複製相同的組態並在您的範本中貼上，您可以為負載平衡器建立專用的範本。之後，任何需要相同負載平衡器組態的其他範本，都可以參考此範本。

巢狀堆疊本身也能包含其他巢狀堆疊，形成堆疊的階層，如下圖所示。根堆疊為最上層的堆疊，即為所有巢狀堆疊最終所屬的堆疊。每一個巢狀堆疊都有一個立即的「父系堆疊」。針對第一層巢狀堆疊，根堆疊也是其父系堆疊。

- 堆疊 A 是所有其他、巢狀、階層中堆疊的根堆疊。
- 針對堆疊 B，堆疊 A 同時是其父系堆疊及根堆疊。
- 針對堆疊 D，堆疊 C 是父系堆疊；針對堆疊 C，堆疊 B 為其父系堆疊。



主題

- [範本拆分的前後範例](#)
- [巢狀堆疊架構範例](#)
- [對巢狀堆疊執行堆疊操作](#)
- [相關資訊](#)

範本拆分的前後範例

本範例展示如何將單一大型 CloudFormation 範本，透過巢狀範本，重組為更具結構化且可重複使用的設計。一開始，「巢狀堆疊前」的範本將所有資源定義在同一個檔案中。隨著資源數量增加，這種方式會變得雜亂且難以管理。「巢狀堆疊後」的範本將資源拆分至多個較小的獨立範本中。每個巢狀堆疊負責處理一組特定的相關資源，使整體結構更具條理且易於維護。

巢狀堆疊前	巢狀堆疊後
<pre> AWSTemplateFormatVersion: 2010-09-09 Parameters: InstanceType: Type: String Default: t2.micro Description: The EC2 instance type Environment: Type: String Default: Production Description: The deployment environment Resources: MyEC2Instance: Type: AWS::EC2::Instance Properties: ImageId: ami-1234567890abcdef0 InstanceType: !Ref InstanceType MyS3Bucket: Type: AWS::S3::Bucket </pre>	<pre> AWSTemplateFormatVersion: 2010-09-09 Resources: MyFirstNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ <i>amzn-s3-demo-bucket</i> /first-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed InstanceType: t3.micro MySecondNestedStack: Type: AWS::CloudFormation::Stack Properties: TemplateURL: https://s3.amazonaws.com/ <i>amzn-s3-demo-bucket</i> /second-nested-stack.yaml Parameters: # Pass parameters to the nested stack if needed Environment: Testing DependsOn: MyFirstNestedStack </pre>

巢狀堆疊架構範例

本節展示一個巢狀堆疊架構，包含一個參考巢狀堆疊的最上層堆疊。該巢狀堆疊會部署 Node.js Lambda 函式，從最上層堆疊接收參數數值，並傳回透過最上層堆疊公開的輸出。

主題

- [步驟 1：在本機系統建立巢狀堆疊的範本](#)
- [步驟 2：在本機系統建立最上層堆疊的範本](#)
- [步驟 3：封裝並部署範本](#)

步驟 1：在本機系統建立巢狀堆疊的範本

下列範例示範巢狀堆疊範本的格式。

YAML

```
AWS::CloudFormation::Template
AWSTemplateFormatVersion: 2010-09-09
Description: Nested stack template for Lambda function deployment
Parameters:
  MemorySize:
    Type: Number
    Default: 128
    MinValue: 128
    MaxValue: 10240
    Description: Lambda function memory allocation (128-10240 MB)
Resources:
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Sub "${AWS::StackName}-Function"
      Runtime: nodejs18.x
      Handler: index.handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          exports.handler = async (event) => {
            return {
              statusCode: 200,
              body: JSON.stringify('Hello from Lambda!')
            };
          };

```

```
    MemorySize: !Ref MemorySize
  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: sts:AssumeRole
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole'
  Outputs:
    LambdaArn:
      Description: ARN of the created Lambda function
      Value: !GetAtt LambdaFunction.Arn
```

步驟 2：在本機系統建立最上層堆疊的範本

下列範例展示最上層堆疊範本的格式，以及參考您在上一步建立之堆疊的 [AWS::CloudFormation::Stack](#) 資源。

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: Top-level stack template that deploys a nested stack
Resources:
  NestedStack:
    Type: AWS::CloudFormation::Stack
    Properties:
      TemplateURL: /path_to_template/nested-template.yaml
      Parameters:
        MemorySize: 256
  Outputs:
    NestedStackLambdaArn:
      Description: ARN of the Lambda function from nested stack
      Value: !GetAtt NestedStack.Outputs.LambdaArn
```

步驟 3：封裝並部署範本

Note

在本機使用範本時，AWS CLI `package` 命令可協助您準備部署的範本。它會自動處理本機成品上傳至 Amazon S3 (包括 `TemplateURL`)，並產生新的範本檔案，其中包含這些 S3 位置的更新參考。如需詳細資訊，請參閱[使用 將本機成品上傳至 S3 儲存貯體 AWS CLI](#)。

接著，您可以使用 `package` 命令將巢狀範本上傳至 Amazon S3 儲存貯體。

```
aws cloudformation package \  
  --s3-bucket amzn-s3-demo-bucket \  
  --template /path_to_template/top-level-template.yaml \  
  --output-template-file packaged-template.yaml \  
  --output json
```

命令會在 `--output-template-file` 指定的路徑上產生新的範本。如下所示，它會將 `TemplateURL` 參考取代為 Amazon S3 位置。

產生範本

```
AWSTemplateFormatVersion: 2010-09-09  
Description: Top-level stack template that deploys a nested stack  
Resources:  
  NestedStack:  
    Type: AWS::CloudFormation::Stack  
    Properties:  
      TemplateURL: https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket/8b3bb7aa7abfc6e37e2d06b869484bed.template  
      Parameters:  
        MemorySize: 256  
Outputs:  
  NestedStackLambdaArn:  
    Description: ARN of the Lambda function from nested stack  
    Value:  
      Fn::GetAtt:  
        - NestedStack  
        - Outputs.LambdaArn
```

執行 `package` 命令後，您可以使用 `deploy` 命令部署已處理的範本。對於包含 IAM 資源巢狀堆疊的 IAM 功能，您必須透過包含 `--capabilities` 選項確認 IAM 功能。

```
aws cloudformation deploy \
  --template-file packaged-template.yaml \
  --stack-name stack-name \
  --capabilities CAPABILITY_NAMED_IAM
```

對巢狀堆疊執行堆疊操作

處理巢狀堆疊時，執行操作時需謹慎對待。某些堆疊操作 (例如堆疊更新) 應從根堆疊啟動，而非直接在巢狀堆疊上執行。更新根堆疊時，僅有範本發生變更的巢狀堆疊會被更新。

此外，巢狀堆疊的存在可能會影響對根堆疊的操作。例如，若某個巢狀堆疊卡在 `UPDATE_ROLLBACK_IN_PROGRESS` 狀態，根堆疊會等待該巢狀堆疊完成復原後再繼續。執行更新操作前，請確保您具備取消堆疊更新的 IAM 許可，以防其復原。如需詳細資訊，請參閱[使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

透過下列步驟尋找根堆疊和巢狀堆疊。

檢視巢狀堆疊的根堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在堆疊頁面上，選擇要檢視根堆疊的巢狀堆疊名稱。

巢狀堆疊會在其堆疊名稱上方顯示巢狀。

3. 在堆疊資訊索引標籤的概觀區段中，選擇列為根堆疊的堆疊名稱。

檢視屬於根堆疊的巢狀堆疊

1. 從您要檢視其巢狀堆疊的根堆疊，選擇資源索引標籤。
2. 在類型欄中，尋找 `AWS::CloudFormation::Stack` 類型的資源。

相關資訊

- [巢狀現有堆疊](#)
- [了解更新堆疊資源的行為](#)

- [從失敗的巢狀堆疊更新中繼續復原](#)
- [巢狀堆疊復原失敗](#)

在 CloudFormation 範本中建立等待條件

此主題解釋了如何在範本中建立等待條件可協調堆疊資源的建立，或追蹤組態程序的進度。例如，您可以在應用程式組態完成一部分後，開始建立另一個資源，或是在安裝和組態程序期間傳送訊號來追蹤其進度。

在 CloudFormation 建立包含等待條件的堆疊時：

- 它像任何其他資源一樣建立等待條件，並將等待條件的狀態設定為 `CREATE_IN_PROGRESS`。
- CloudFormation 會等待，直到收到所需數量的成功訊號，或等待條件的逾時時間到期。
- 若在逾時時間到期前收到所需數量的成功訊號：
 - 等待條件狀態變更為 `CREATE_COMPLETE`
 - 堆疊建立繼續
- 如果逾時過期，或收到失敗訊號：
 - 等待條件狀態變更為 `CREATE_FAILED`
 - 堆疊復原

Important

針對 Amazon EC2 和 Auto Scaling 資源，我們建議您使用 `CreationPolicy` 屬性，而非等待條件。請將 `CreationPolicy` 屬性新增至這類資源，並使用 `cfn-signal` 的 helper 指令碼在執行個體建立流程成功完成後發出訊號。

如需詳細資訊，請參閱 [CreationPolicy 屬性](#)。

Note

如果您使用 AWS PrivateLink，VPC 中回應等待條件的資源必須能夠存取 CloudFormation 特定的 Amazon Simple Storage Service (Amazon S3) 儲存貯體。資源必須傳送等待條件回應至預先簽章的 Amazon S3 URL。若無法將回應傳送至 Amazon S3，CloudFormation 便不會收到回應，而堆疊操作則會失敗。如需詳細資訊，請參閱 [CloudFormation 使用界面端點存取 \(AWS PrivateLink\)](#) 和 [使用儲存貯體政策控制 VPC 端點的存取](#)。

主題

- [在您的範本中建立等待條件](#)
- [等待條件訊號語法](#)
- [存取訊號資料](#)

在您的範本中建立等待條件

1. 等待條件控點

首先在堆疊的範本中定義 [AWS::CloudFormation::WaitConditionHandle](#) 資源。此資源會產生傳送訊號所需的預先簽章 URL。這可讓您傳送訊號，無需提供您的 AWS 憑證。例如：

```
Resources:
  MyWaitHandle:
    Type: AWS::CloudFormation::WaitConditionHandle
```

2. 等待條件

接下來，您可在堆疊的範本中定義 [AWS::CloudFormation::WaitCondition](#) 資源。AWS::CloudFormation::WaitCondition 的基本結構如下所示：

```
MyWaitCondition:
  Type: AWS::CloudFormation::WaitCondition
  Properties:
    Handle: String
    Timeout: String
    Count: Integer
```

AWS::CloudFormation::WaitCondition 資源有兩個必要屬性和一個選用屬性。

- Handle (必要) – 範本中所宣告 WaitConditionHandle 的參考。
- Timeout (必要) – CloudFormation 等待收到必要訊號數的秒數。Timeout 是最小限制屬性，表示逾時不會早於您指定的時間，但會在之後不久發生。您可以指定的時間上限是 43200 秒 (12 小時)。
- Count (選用) – CloudFormation 在將等待條件的狀態設定為 CREATE_COMPLETE 之前，必須接收的成功訊號數量，並繼續建立堆疊。如果未指定，預設值為 1。

通常，您會希望等待條件在特定資源建立完成後立即開始。您可以將 `DependsOn` 屬性新增至等待條件來執行此操作。向等待條件新增 `DependsOn` 屬性後，CloudFormation 會先建立 `DependsOn` 屬性中指定的資源，再建立等待條件。如需詳細資訊，請參閱 [DependsOn 屬性](#)。

下列範例示範的等待條件如下：

- 在成功建立 `MyEC2Instance` 資源後開始
- 將 `MyWaitHandle` 資源用作 `WaitConditionHandle`
- 逾時時間為 4500 秒
- 預設 `Count` 為 1 (因為未指定 `Count` 屬性)

MyWaitCondition:

```
Type: AWS::CloudFormation::WaitCondition
DependsOn: MyEC2Instance
Properties:
  Handle: !Ref MyWaitHandle
  Timeout: '4500'
```

3. 傳送訊號

若要向 CloudFormation 傳出成功或失敗訊號，您通常會執行一些程式碼或指令碼。例如，在 EC2 執行個體上執行的應用程式可能會執行一些額外的組態工作，然後向 CloudFormation 傳送訊號以表示完成。

訊號必須傳送至等待條件控制代碼產生的預先簽章 URL。使用預先簽章的 URL 傳送成功或失敗訊號。

傳送訊號

1. 要擷取範本中的預先簽章 URL，請使用 `Ref` 內建函數搭配等待條件控點的邏輯名稱。

如以下範例所示，您的範本可以宣告 Amazon EC2 執行個體，並透過 Amazon EC2 `UserData` 屬性將預先簽章的 URL 傳遞至 Amazon EC2 執行個體。這讓在這些執行個體上執行的指令碼或應用程式能向 CloudFormation 傳送成功或失敗訊號。

MyEC2Instance:

```
Type: AWS::EC2::Instance
Properties:
  InstanceType: t2.micro # Example instance type
  ImageId: ami-055e3d4f0bbeb5878 # Change this as needed (Amazon Linux 2023 in us-west-2)
```

```

UserData:
  Fn::Base64:
    Fn::Join:
      - ""
      - - "SignalURL="
        - { "Ref": "MyWaitHandle" }

```

這導致 UserData 輸出結果類似以下內容：

```
SignalURL=https://amzn-s3-demo-bucket.s3.amazonaws.com/....
```

注意：在 AWS 管理主控台 和 命令列工具中，預先簽章的 URL 會顯示為等待條件處理資源的實體 ID。

2. (選用) 若要偵測堆疊何時進入等待條件，您可以使用下列其中一種方法：
 - 如果您建立堆疊並啟用通知，CloudFormation 會針對每個堆疊事件發佈通知到指定的主題。如果您或您的應用程式訂閱該主題，您可以監控等待條件控點建立事件的通知，並從通知訊息擷取預先簽章的 URL。
 - 您也可以使用 AWS 管理主控台、AWS CLI 或 SDK 來監控堆疊的事件。
3. 若要傳送訊號，您可以使用預先簽章的 URL 傳送 HTTP 請求訊息。請求方法必須是 PUT，而且 Content-Type 標頭必須是空白字串或予以省略。請求訊息必須是 [等待條件訊號語法](#) 中指定格式的 JSON 結構。

您必須傳送 Count 屬性指定的成功訊號數目，CloudFormation 才能繼續建立堆疊。如果您的 Count 大於 1，在傳送至特定等待條件的所有訊號之間，每個訊號的 UniqueId 值必須都是不重複的。UniqueId 是任意英數字串。

curl 命令是傳送訊號的一種方式。下列範例示範將成功訊號傳送至等待條件的 curl 命令列。

```

$ curl -T /tmp/a \
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-
west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-
WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?
Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo
%3D"

```

其中 `/tmp/a` 檔案包含下列 JSON 結構：

```
{
```

```
"Status" : "SUCCESS",  
"Reason" : "Configuration Complete",  
"UniqueId" : "ID1234",  
"Data" : "Application has completed configuration."  
}
```

此範例示範傳送同一個成功訊號的 curl 命令列，差別在於在命令列中以參數的形式傳送 JSON 結構。

```
$ curl -X PUT \  
  -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" : "Configuration  
  Complete","UniqueId" : "ID1234","Data" : "Application has completed  
  configuration."}' \  
  "https://amzn-s3-demo-bucket.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-  
  west-2%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-  
  WaitCondition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitConditionHandle?  
  Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wy0oRejVo  
  %3D"
```

等待條件訊號語法

當您向等候條件控點所產生的 URL 傳送訊號時，您必須使用下列 JSON 格式：

```
{  
  "Status" : "StatusValue",  
  "UniqueId" : "Some UniqueId",  
  "Data" : "Some Data",  
  "Reason" : "Some Reason"  
}
```

Properties

Status 欄位可以是以下其中一個值：

- SUCCESS
- FAILURE

UniqueId 欄位用於識別 CloudFormation 的訊號。如果等待條件的 Count 屬性大於 1，UniqueId 值在傳送給特定等待條件的所有訊號之間必須是唯一的；否則，CloudFormation 會認為是使用先前傳送過 UniqueId 相同之訊號重新傳輸的訊號，而忽略該訊號。

Data 欄位可包含您想要隨訊號傳回的任何資訊。在範本內使用 [Fn::GetAtt](#) 函數就可存取 Data 值。

Reason 欄位是字串，內容除了需要符合 JSON 規定以外，沒有其他限制。

存取訊號資料

要存取有效訊號傳送的資料，您可在 CloudFormation 範本中建立等待條件的輸出值。例如：

Outputs:

WaitConditionData:

Description: *The data passed back as part of signalling the WaitCondition*

Value: *!GetAtt MyWaitCondition.Data*

然後，您可使用 [describe-stacks](#) 命令或 CloudFormation 主控台的輸出索引標籤來檢視此資料。

Fn::GetAtt 函數會在 JSON 結構內以名稱/值對的形式傳回 UniqueId 和 Data。例如：

```
{"Signal1": "Application has completed configuration."}
```

透過 CloudFormation 模組建立可跨範本納入的可重複使用資源組態

模組可讓您以透明、易於管理且可重複的方式，來封裝可包含在堆疊範本之間的資源組態。模組可透過可自訂的模組化建置區塊形式，來封裝通用服務組態和最佳實務，以便您將其納入堆疊範本中。模組可讓您納入資源組態，這些資源組態會將最佳實務、專業領域知識和公認指導方針 (適用於安全性、合規性、控管和產業法規等領域) 納入範本，無須深入了解資源實作的複雜細節。

例如，聯網的領域專家可以建立模組，其中包含內建安全性群組和遵守安全性指導方針的輸入/輸出規則。然後，您可以將該模組納入範本中，以便在堆疊中佈建安全的聯網基礎設施，而不需花時間了解 VPC、子網路、安全性群組和閘道的運作方式。由於模組是由版本控制的，如果安全性指導方針會隨著時間而改變，則模組作者可以建立新版的模組，以納入那些變更。

在範本中使用模組的特性包括：

- 可預測性 – 模組必須遵守它在 CloudFormation 登錄中註冊的結構描述，以便讓您了解在將模組包含在範本時，該模組可以解析哪些資源。

- 可重複使用性 - 您可以跨多個範本和帳戶使用相同的模組。
- 可追蹤性 - CloudFormation 會保留堆疊中哪些資源是透過模組佈建而得的知識，讓您輕鬆了解資源變更的來源。
- 可管理性 - 在您註冊模組後，就可以透過 CloudFormation 登錄管理模組，包括版本控制、帳戶和區域可用性。

模組可以包含：

- 要透過模組佈建的一或多個資源，以及輸出或條件等任何相關資料。
- 任何模組參數，可讓您在使用的模組時指定自訂值。

如需開發模組的資訊，請參閱《CloudFormation CLI 使用者指南》中的[開發模組](#)。

主題

- [使用模組時的考量](#)
- [了解模組版本控制](#)
- [使用 CloudFormation 私有登錄檔的模組](#)
- [使用參數來指定模組值](#)
- [在 CloudFormation 範本中參考模組資源](#)

使用模組時的考量

- 使用模組無需額外收費。您只需為那些模組在堆疊中解析的資源付費。
- CloudFormation 配額 (例如堆疊中允許的資源數目上限或範本內文的大小上限) 適用於已處理的範本，無論該範本中包含的資源是否來自模組。如需詳細資訊，請參閱[了解 CloudFormation 配額](#)。
- 您在堆疊層級指定的標籤會指派給自模組衍生的個別資源。
- CloudFormation 處理範本時，在模組層級指定的協助程式指令碼不會傳播到模組中包含的個別資源。
- 模組中指定的輸出會傳播到範本層級的輸出。

每個輸出將分配到一個邏輯 ID，此 ID 是模組邏輯名稱和在模組中定義之輸出名稱的串連。如需詳細資訊，請參閱[從部署的 CloudFormation 堆疊取得匯出的輸出](#)。

- 模組中指定的參數不會傳播到範本層級的參數。

不過，您可以建立參考模組層級參數的範本層級參數。如需詳細資訊，請參閱[使用參數來指定模組值](#)。

了解模組版本控制

CloudFormation 登錄檔做為儲存庫，您可以在其中註冊和管理模組，以便在 AWS 帳戶和區域中使用。您可以在您的帳戶和區域中註冊來自各種來源的模組 AWS，包括第三方發佈者和您自己的自訂擴充功能。如需詳細資訊，請參閱[使用 CloudFormation 登錄檔管理擴充功能](#)。

模組可以有不同版本，因此您可以指定要使用的模組版本。當您需要更新或修改模組而不中斷依賴它的現有堆疊時，此版本控制功能特別實用。

使用多個版本的模組時，請注意以下考量事項：

- 在堆疊操作期間，CloudFormation 會使用目前註冊為執行堆疊操作之 AWS 帳戶和區域中預設版本的任何模組版本。這包括在其他模組中巢狀堆疊的模組。

因此，請注意，如果您在不同帳戶或區域中將相同模組的不同版本註冊為預設版本，則使用相同的範本可能會產生不同的結果。

- 在堆疊操作期間，CloudFormation 會使用目前註冊為執行堆疊操作之 AWS 帳戶和區域中預設版本的任何資源版本。其中包括內含模組所產生的資源。
- 變更模組的預設版本不會起始任何堆疊更新操作。不過，在您下次使用任何包含該模組的範本 (例如堆疊更新) 執行堆疊操作時，CloudFormation 將在操作中使用新的預設版本。

如下所述，唯一的例外是，使用指定的 `use previous template` (使用先前範本) 選項執行堆疊更新時。

- 對於堆疊更新操作，如果您指定 `use previous template` (使用先前範本) 選項，CloudFormation 會使用先前處理的範本進行堆疊更新，而且不會針對您可能對其所做的任何變更重新處理模組。
- 為了保證結果一致，如果您將模組包含在堆疊範本中以供堆疊集使用，則應確保在您計劃部署堆疊執行個體的所有帳戶和區域中，將相同的模組版本設定為預設版本。這包括在其他模組巢狀堆疊中的模組。如需詳細資訊，請參閱[透過 StackSets 跨帳戶和區域管理堆疊](#)。

啟用第三方公有模組的需求

若要成功啟用帳戶和區域中的第三方公有模組，模組中包含的每個第三方公有擴充功能 (資源或模組) 必須符合下列條件：

- 擴充功能啟用 – 擴充功能必須在您要使用它的帳戶及區域中啟用。如需詳細資訊，請參閱[從 CloudFormation 登錄檔使用第三方公有擴充功能](#)。
- 別名註冊 - 如果模組中的擴充功能使用類型名稱別名，則必須使用相同類型名稱別名在帳戶和區域中註冊擴充功能。如需詳細資訊，請參閱[使用別名來稱呼擴充功能](#)。
- 版本相容性 - 目前已啟用擴充功能版本必須是模組中指定之該擴充功能的主要支援版本之一。

如果您沒有啟用正確的第三方公有擴充功能和擴充功能版本，CloudFormation 將導致操作失敗並顯示錯誤，此錯誤會列出成功啟用模組之前需要啟用的擴充功能和版本。

使用 CloudFormation 私有登錄檔的模組

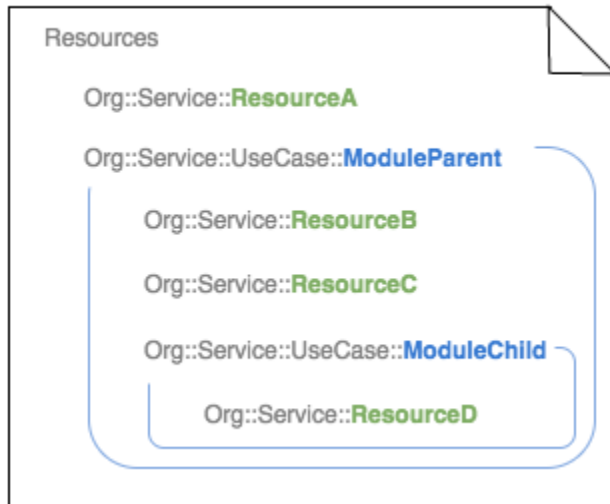
本主題說明了如何在 CloudFormation 範本中使用模組。可將模組視為預先製作的資源套件，您可以將其新增至範本中。

使用模組的步驟如下：

- 註冊模組 – 您可以在 CloudFormation 登錄檔中將模組註冊為私有延伸項目。請確定它已在您正在使用的 AWS 帳戶 和 區域中註冊。如需詳細資訊，請參閱[CloudFormation 登錄檔概念](#)。
- 將其包含在範本中 – 如同處理其他資源一般，將模組新增至 CloudFormation 範本的 [Resources](#) 區段。您還需提供該模組所需的所有必要屬性。
- 建立或更新堆疊 – 當您起始堆疊操作時，CloudFormation 會產生處理過的範本，將任何包含的模組解析為適當的資源。
- 預覽變更 – 進行變更前，您可使用變更集查看即將新增或變更的資源。如需詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

請考量下面的範例：你擁有同時內含資源和模組的範本。該範本包含一個個別的資源 ResourceA，以及模組 ModuleParent。該模組包含兩個資源 ResourceB 與 ResourceC，以及一個巢狀模組 ModuleChild，而 ModuleChild 包含單一資源 ResourceD。如果您透過此範本建立堆疊，CloudFormation 會處理此範本，並將模組解析為適當的資源。最終的堆疊包含四個資源：ResourceA、ResourceB、ResourceC 和 ResourceD。

Template



Processed Template



CloudFormation 會追蹤堆疊中的哪些資源是透過模組而建立。您可以在指定堆疊的 Events (事件)、Resources (資源) 和 Drifts (漂移) 索引標籤上檢視此資訊，變更集預覽中也會包含此資訊。

模組與範本中的資源有所不同，因為它們遵循的是以下四段命名慣例，而不是資源使用的典型三段慣例：

```
organization::service::use-case::MODULE
```

使用參數來指定模組值

在 CloudFormation 中，可以透過在堆疊建立或更新期間提供輸入值，使用範本參數來自訂堆疊。這些參數可讓您根據需求變更堆疊的某些層面。如需定義範本參數的詳細資訊，請參閱 [CloudFormation 範本 Parameters 語法](#)。

同樣地，模組也可擁有參數。這些模組參數可讓您從使用它的範本 (或其他模組) 將自訂值輸入模組。然後，此模組可以使用這些自訂值，來設定其包含的屬性值。

您還可以定義範本參數，進而設定模組屬性，以便您可以輸入在堆疊操作時傳遞給模組的值。

如果模組包含的巢狀模組擁有自己的模組參數，您可以選擇以下任一方式：

- 直接在父模組中指定巢狀模組參數的值。
- 在父模組中定義對應的模組參數，使包含父模組的範本 (或模組) 可設定巢狀模組的參數。

使用範本參數指定模組參數值

下列範例顯示如何定義將值傳遞給模組的範本參數。

包含 `My::S3::SampleBucket::MODULE` 的範本會定義範本參數 `BucketName`，該參數可讓使用者在堆疊操作過程中指定 S3 儲存貯體名稱。

```
# Template containing My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucket::MODULE'
    Properties:
      BucketName: !Ref BucketName
```

透過父模組在子模組中指定資源的屬性

下列範例說明如何在另一個模組中巢狀堆疊的模組中指定參數值。

第一個模組 `My::S3::SampleBucketPrivate::MODULE` 將是子模組。其定義了兩個參數：`BucketName` 和 `AccessControl`。為這些參數指定的值會用來指定此模組包含之 `AWS::S3::Bucket` 資源的 `BucketName` 和 `AccessControl` 屬性。以下是 `My::S3::SampleBucketPrivate::MODULE` 的範本片段。

```
# My::S3::SampleBucketPrivate::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket with Versioning and DeletionPolicy.
Parameters:
  BucketName:
    Description: Name for the bucket
    Type: String
  AccessControl:
    Description: AccessControl for the bucket
    Type: String
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Ref BucketName
```

```
AccessControl: !Ref AccessControl
DeletionPolicy: Retain
VersioningConfiguration:
  Status: Enabled
```

接下來，前一個模組會在父模組 `My::S3::SampleBucket::MODULE` 中巢狀堆疊。此父模組 `My::S3::SampleBucket::MODULE` 會以下列方式設定子模組參數：

- 它會將 `My::S3::SampleBucketPrivate::MODULE` 的 `AccessControl` 參數設定為 `Private`。
- 若是 `BucketName`，則會定義模組參數，此參數將允許在包含 `My::S3::SampleBucket::MODULE` 的範本 (或模組) 中指定儲存貯體名稱。

```
# My::S3::SampleBucket::MODULE
AWSTemplateFormatVersion: 2010-09-09
Description: A sample S3 Bucket. With Private AccessControl.
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: 'My::S3::SampleBucketPrivate::MODULE'
    Properties:
      BucketName: !Ref BucketName
      AccessControl: Private
```

指定模組參數的限制條件

模組參數不支援限制條件強制實施。若要對模組參數執行限制條件檢查，請建立具有所需限制條件的範本參數。然後，在您的模組參數中參考該範本參數。如需定義範本參數的詳細資訊，請參閱 [CloudFormation 範本 Parameters 語法](#)。

在 CloudFormation 範本中參考模組資源

在 CloudFormation 範本中，您通常需要根據另一個資源的名稱或屬性，設定某個資源的屬性。如需詳細資訊，請參閱 [參考資源](#)。

若要在 CloudFormation 範本中參考模組內含的資源，您必須合併兩個邏輯名稱：

- 當您將其引入範本時，為模組本身設定的邏輯名稱。
- 該模組內特定資源的邏輯名稱。

您可以在這兩個邏輯名稱之間加點 (.) 或不加點，合併形成完整名稱。例如，若模組的邏輯名稱是 MyModule，而資源的邏輯名稱是 MyBucket，您可將該資源稱為 MyModule.MyBucket 或 MyModuleMyBucket。

若要查詢模組內資源的邏輯名稱，可以參閱模組的結構描述，該結構描述可在 CloudFormation 登錄檔或使用 [DescribeType](#) 操作取得。該結構描述會列出屬於模組的所有資源及其邏輯名稱。

取得完整的邏輯名稱後，您可以使用 GetAtt 和 Ref 等 CloudFormation 函數，存取模組資源上的屬性值。

例如，您有一個 My::S3::SampleBucket::MODULE 模組，其中包含邏輯名為 S3Bucket 的 AWS::S3::Bucket 資源。要使用 Ref 函數參考此儲存貯體的名稱，可以將範本 (MyBucket) 中的模組名稱與模組 (S3Bucket) 中的資源邏輯名稱合併。完整邏輯名稱可為 MyBucket.S3Bucket 或 MyBucketS3Bucket。

範例 範本

下列範例範本會使用 My::S3::SampleBucket::MODULE 模組建立一個 S3 儲存貯體。它還會建立 Amazon SQS 佇列，並將其名稱設定為與模組中的儲存貯體名稱相同。此外，該範本會輸出建立的 S3 儲存貯體的 Amazon Resource Name (ARN)。

```
# Template that uses My::S3::SampleBucket::MODULE
Parameters:
  BucketName:
    Description: Name for your sample bucket
    Type: String
Resources:
  MyBucket:
    Type: My::S3::SampleBucket::MODULE
    Properties:
      BucketName: !Ref BucketName
  exampleQueue:
    Type: AWS::SQS::Queue
    Properties:
      QueueName: !Ref MyBucket.S3Bucket
Outputs:
  BucketArn:
```

Value: **!GetAtt MyBucket.S3Bucket.Arn**

使用 CloudFormation 堆疊以單一單位管理 AWS 資源

堆疊是您可以單一單位管理的一組 AWS 資源。換言之，您可以建立、更新及刪除堆疊，來建立、更新及刪除資源集合。

建立堆疊需部署 CloudFormation 範本，範本中會指定資源及其組態，其後由 CloudFormation 佈建並設定這些資源。

更新堆疊需變更範本或參數。CloudFormation 會將您提交的變更與您堆疊目前的狀態進行比較，並只更新變更的資源。CloudFormation 可能會中斷資源或取代更新的資源，取決於您更新的屬性。如需資源更新行為的詳細資訊，請參閱 [了解更新堆疊資源的行為](#)。

CloudFormation 提供更新堆疊的兩種方式：

- **變更集** – 使用變更集，您可以預覽 CloudFormation 對您堆疊進行的變更，然後決定是否要套用那些變更。變更集為 JSON 格式的文件，摘要 CloudFormation 對堆疊進行的變更。當您希望確保 CloudFormation 不會進行任何無意變更，或是您希望考慮幾個選項時，請使用變更集。例如，您可以使用變更集確認 CloudFormation 不會在更新期間取代您堆疊的資料庫執行個體。
- **直接更新** – 當您直接更新堆疊時，您會提交變更，CloudFormation 也會立即部署他們。當您希望快速部署更新時，請使用直接更新。

刪除堆疊會刪除與之相關聯的資源。舉例來說，堆疊可包含所有需要用來執行 Web 應用程式的資源，例如 Web 伺服器、資料庫和網路規則。若您不再需要該 Web 應用程式，您只需要刪除堆疊，即可刪除所有相關資源。

Note

您必須為運作堆疊資源的期間支付費用 (即使您立即刪除堆疊)。

CloudFormation 確保所有堆疊資源都會以適當方式建立或刪除。因為 CloudFormation 將堆疊資源視為單一單位，他們必須全部成功建立或成功刪除，才能使堆疊成功建立或刪除。若無法建立其中一項資源，CloudFormation 會復原堆疊並自動刪除任何已建立的資源。若無法刪除其中一項資源，任何剩餘的資源都會受到保留，直到可成功刪除堆疊為止。

主題

- [管理堆疊的介面](#)

- [從 CloudFormation 主控台建立堆疊](#)
- [在 CloudFormation 主控台中檢視堆疊資訊](#)
- [更新您的堆疊範本](#)
- [了解更新堆疊資源的行為](#)
- [透過變更集更新 CloudFormation 堆疊](#)
- [驗證堆疊部署](#)
- [直接更新堆疊](#)
- [取消堆疊更新](#)
- [從 CloudFormation 主控台中刪除堆疊](#)
- [監控堆疊進度](#)
- [透過復原觸發條件，在警示違規時復原您的 CloudFormation 堆疊](#)
- [透過漂移偵測功能，偵測堆疊和資源的未受管組態變更](#)
- [將 AWS 資源匯入 CloudFormation 堆疊](#)
- [堆疊重構](#)
- [資源類型支援](#)
- [使用快速建立連結來建立 CloudFormation 堆疊](#)
- [AWS CLI 和 PowerShell 的 CloudFormation 堆疊操作命令範例](#)

管理堆疊的介面

您可以透過下列介面管理 CloudFormation 堆疊：

- CloudFormation 主控台 – 提供 Web 介面，您可使用此介面來存取堆疊。您可以使用導覽列上的 AWS 管理主控台搜尋方塊來搜尋 CloudFormation，然後從搜尋結果中選擇 CloudFormation，以存取 CloudFormation 主控台。
- AWS Command Line Interface – 提供包括 CloudFormation 在內的大量 AWS 服務命令，Windows、Mac 和 Linux 都支援。如需有關 CloudFormation 命令的資訊，請參閱《AWS CLI 命令參考》中的 [cloudformation](#)。
- AWS Tools for PowerShell - 一組建立在適用於 .NET 的 SDK公開的功能之上的 PowerShell 模組。Tools for PowerShell 可讓您從PowerShell命令列編寫 AWS 資源操作的指令碼。您可以在 [AWS Tools for PowerShell Cmdlet 參考](#)中找到適用於 CloudFormation 的 cmdlets。

- Query API – 提供您可以使用 HTTPS 請求呼叫的低層級 API 動作。若您在應用程式中進行 API 呼叫，必須撰寫程式碼處理低階詳細資訊，例如產生用於請求簽名的雜湊值。關於 CloudFormation 的 API 動作的詳細資訊，請參閱《AWS CloudFormation API 參考》中的[動作](#)。
- AWS SDKs – 提供語言特定的 APIs，並負責許多連線詳細資訊，例如計算簽章、處理請求重試和錯誤處理。如需詳細資訊，請參閱在 [AWS](#)。
- AWS Cloud Development Kit (AWS CDK) – AWS CDK 是一種開放原始碼軟體開發架構，可讓您使用熟悉的程式設計語言定義 AWS 基礎設施，例如 TypeScript、Python、Java 和 .NET。透過 CDK，您可以建立應用程式資源的模型，然後直接從整合式開發環境 (IDE) 使用 CloudFormation 佈建這些資源。如需詳細資訊，請參閱[AWS Cloud Development Kit \(AWS CDK\)](#)。

從 CloudFormation 主控台建立堆疊

可以建立堆疊範本，然後使用它透過 CloudFormation 主控台或命令列工具建立堆疊。主控台提供精靈驅動的介面與預先定義的選項，可簡化堆疊建立程序。

主題

- [建立堆疊](#)
- [設定堆疊選項](#)
- [預覽堆疊的組態](#)

建立堆疊

請依照本節中的步驟來部署範本並建立堆疊。

先決條件

- 您必須已經建立堆疊範本。如需詳細資訊，請參閱[使用 CloudFormation 範本](#)。

若要建立堆疊 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇要在其中 AWS 區域 建立堆疊的。
3. 在堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。

或者，您可以選擇使用現有資源 (匯入資源) 選項，以匯入範本中描述的現有 AWS 資源。如需有關此選項的詳細資訊，請參閱 [將 AWS 資源匯入 CloudFormation 堆疊](#)。

4. 在建立堆疊頁面中，執行下列動作：

- 若要使用現有範本，請在先決條件 - 準備範本中選取選擇現有範本。然後，在指定範本下，根據範本的位置選擇 Amazon S3 URL 或上傳範本檔案。
- 如果選擇 Amazon S3 URL，請將 URL 提供給 S3 儲存貯體中的範本檔案。

如果範本包含巢狀堆疊 (例如，子目錄中其他範本文件中所述的堆疊)，請確保 S3 儲存貯體包含必要的檔案和目錄。

如果您擁有已啟用版本控制的儲存貯體中的範本，則可以指定範本的特定版本，方法是將 `versionId=version-id` 附加到 URL。如需有關已啟用版本控制的儲存貯體詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用已啟用版本控制之儲存貯體中的物件](#)。

URL 必須指向大小上限為 1 MB 的範本，該範本存放在您具有讀取許可的 S3 儲存貯體內。URL 長度上限為 1024 個字元。部分資源可能要求在堆疊所在的相同區域內有儲存貯體。

- 如果您選擇上傳範本檔案，請選取選擇檔案以從本機電腦中選擇範本檔案。範本檔案大小必須為 1 MB 或以下。

一旦選擇範本，CloudFormation 就會上傳該檔案，並顯示 S3 URL。CloudFormation 會將其上傳至 AWS 帳戶中的 Amazon S3 儲存貯體。如果您已有 CloudFormation 在中建立的 S3 儲存貯體 AWS 帳戶，CloudFormation 會將範本新增至該儲存貯體。如果您還沒有現有的 CloudFormation 建立的 S3 儲存貯體，它會針對您上傳範本檔案的每個區域建立唯一儲存貯體。

以下是使用 CloudFormation 建立的 S3 儲存貯體時的考量：

- 擁有 Amazon S3 許可的任何人都可以存取儲存貯體 AWS 帳戶。
- CloudFormation 建立儲存貯體時，伺服器端加密預設為啟用，藉此加密所有儲存貯體中存放的物件。

您可以直接管理 CloudFormation 已建立的儲存貯體之加密選項，例如，使用位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台或使用 AWS CLI。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[設定 Amazon S3 儲存貯體的預設伺服器端加密行為](#)。

- 您可以使用自己的儲存貯體，並手動上傳範本至 Amazon S3 以管理其許可。當您建立或更新堆疊時，請指定範本檔的 Amazon S3 URL。

- 如果尚未準備好範本，可以選擇從 Infrastructure Composer 建置，以使用 Infrastructure Composer 建立範本。如需詳細資訊，請參閱[Infrastructure Composer](#)。

5. 選擇下一步以繼續並驗證範本。

在繼續之前，CloudFormation 會驗證您的範本以捕捉語法和語意錯誤，例如循環參考。在驗證期間，CloudFormation 會先檢查範本是否為有效的 JSON。如果不是，CloudFormation 會檢查範本是否為有效的 YAML。如果兩種檢查都失敗，則 CloudFormation 會傳回範本驗證錯誤。

6. 在 Specify stack details (指定堆疊詳細資訊) 頁面的 Stack name (堆疊名稱) 方塊中，輸入堆疊名稱。

堆疊名稱是一個識別碼，可協助您從堆疊清單中找到特定堆疊。堆疊名稱只能包含英數字元 (區分大小寫) 和連字號。必須以字母字元開頭，且長度不可超過 128 個字元。

7. 在參數區段中，指定範本中定義的參數值。

8. 請選擇下一步，繼續建立堆疊。

9. (選用) 在配置堆疊選項頁面中，變更預設堆疊選項。如需詳細資訊，請參閱[設定堆疊選項](#)。

10. 如果範本包含 IAM 資源，在功能中選擇我知道此範本可建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。

11. 選擇 Next (下一步) 繼續。

12. 在檢閱並建立頁面中，檢閱堆疊詳細資訊。

若要在啟動堆疊之前變更任何值，請在具有要變更之設定的區段中選擇編輯。

13. (選用) 您可以建立變更集，以在建立之前預覽堆疊的組態。在檢閱並建立頁面中，選擇建立變更集並遵循指示。如需詳細資訊，請參閱[預覽堆疊的組態](#)。

14. 選擇提交以啟動堆疊。

CloudFormation 接著會繼續建立範本中定義的所有資源。

可以在新堆疊的事件索引標籤中監控堆疊建立的進度和狀態。如需詳細資訊，請參閱[監控堆疊進度](#)。

若要使用命令列來建立堆疊

您可以使用下列其中一項命令：

- [create-stack](#) (AWS CLI)
- [New-CFNStack](#) (AWS Tools for Windows PowerShell)

如需使用命令列建立堆疊的範例，請參閱 [AWS CLI 和 PowerShell 的 CloudFormation 堆疊操作命令範例](#)。

設定堆疊選項

在設定堆疊選項頁面中，可以為 CloudFormation 堆疊設定選項，例如標籤、堆疊事件通知或堆疊政策。

您可以設定下列堆疊選項：

Tags (標籤)

最多可以將 50 個標籤金鑰對新增至堆疊以及 CloudFormation 支援標記的任何資源。標籤是客戶定義的金鑰和值，可以指派給 AWS 資源，用於成本追蹤等目的。

Key (金鑰) 會包含任何英數字元或空格。而標籤金鑰的長度最多可達 127 個字元。

Value (數值) 會包含任何英數字元或空格。而標籤值的長度最多可達 255 個字元。

建立堆疊後，新增、更新或移除堆疊層級標籤會啟動堆疊更新。支援堆疊層級標籤傳播的所有資源都會相應更新。

許可

CloudFormation 可以擔任的現有 IAM 服務角色。CloudFormation 會使用該角色的憑證來建立堆疊，而非採用您帳戶的憑證。如需詳細資訊，請參閱 [CloudFormation 服務角色](#)。

堆疊失敗選項

為所有堆疊部署和變更集操作指定佈建失敗選項。如需詳細資訊，請參閱 [選擇佈建資源時的失敗處理方式](#)。

堆疊狀態為 CREATE_FAILED 或 UPDATE_FAILED 時，Roll back all stack resources (復原所有堆疊資源) 選項將復原範本中指定的所有資源。

針對建立操作，Preserve successfully provisioned resources (保留已成功佈建的資源) 選項可保留成功資源的狀態，而失敗的資源會保持失敗狀態，直到執行下一次更新操作為止。

針對更新與變更集操作，Preserve successfully provisioned resources (保留已成功佈建的資源) 選項可保留成功資源的狀態，同時將失敗資源復原到上一個已知穩定狀態。失敗的資源將處於 UPDATE_FAILED 狀態。沒有上一個已知穩定狀態的資源將在下次堆疊操作時遭刪除。

您也可以針對堆疊建立設定以下進階選項：

Stack policy (堆疊政策)

可藉由此選項來定義堆疊更新期間，您欲避免意外更新的資源。在預設情況下，系統可以在堆疊更新期間將所有資源進行更新。

您可以直接輸入堆疊政策作為 JSON，或上傳包含堆疊政策的 JSON 檔案。如需詳細資訊，請參閱[避免更新堆疊資源](#)。

復原組態

可讓 CloudFormation 監控堆疊在堆疊建立和更新期間的狀態，如果堆疊超過您指定的任何警示閾值，也可復原該操作。指定 CloudFormation 應該監控的 Cloudwatch 警示。如有任何警示在堆疊操作或監控期間進入 ALARM 狀態，CloudFormation 將會復原整個堆疊操作。如需詳細資訊，請參閱[透過復原觸發條件，在警示違規時復原您的 CloudFormation 堆疊](#)。

通知選項

您可指定新的或現有 Amazon Simple Notification Service 主題，其為系統傳送堆疊事件通知的目標位置。

建立 Amazon SNS 主題時，您必須指定名稱與電子郵件地址，以便接收堆疊事件通知。

堆疊建立選項

堆疊建立包含以下選項，但不提供為堆疊更新的一部分。

Timeout (逾時)

指定在堆疊建立操作逾時之前 CloudFormation 應分配的時間量 (以分鐘為單位)。如果 CloudFormation 無法在分配的時間內建立整個堆疊，操作便會因為逾時而失敗，並復原堆疊。

依預設，堆疊建立沒有逾時。不過，個別資源可能會因其實作的服務性質而有自己的逾時。例如，如果堆疊中的個別資源逾時，堆疊建立也會逾時，即使尚未達到您指定的堆疊建立逾時。

終止保護

此選項可防止堆疊遭意外刪除。如果使用者嘗試刪除啟用終止保護的堆疊，則刪除會失敗，且堆疊及其狀態保持不變。如需詳細資訊，請參閱[防止 CloudFormation 堆疊遭到刪除](#)。

根據預設，終止保護為 Disabled (已停用)。

預覽堆疊的組態

如需在建立堆疊前，預覽 CloudFormation 堆疊的設定方式，則請建立變更集。本功能可以讓您在執行變更集前檢查各種組態，並針對堆疊進行修正與更改。如需變更集的詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

建立新堆疊的變更集

若要建立新堆疊的變更集，請選取您的堆疊範本並指定堆疊組態，就像建立新堆疊一樣，然後選擇建立新的變更集，而不是建立新的堆疊。

若要建立新堆疊的變更集

1. 在檢閱並建立頁面中，選取建立變更集。
2. 在建立變更集對話方塊中，輸入變更集的名稱，並視需要輸入描述。選擇 Create change set (建立變更集)。

當您建立新堆疊的變更集時，CloudFormation 會執行下列動作：

- 啟動狀態為 REVIEW_IN_PROGRESS (正在檢閱) 的新堆疊。
- 建立新堆疊的變更集，此堆疊會反映您在前列步驟中指定的堆疊組態。

CloudFormation 會顯示提議堆疊的 Change sets (變更集) 頁面。當 CloudFormation 建立變更集時，其狀態為 CREATE_IN_PROGRESS (正在建立)，執行狀態為 UNAVAILABLE (無法使用)。CloudFormation 成功建立變更集時，其會將變更集狀態設定為 CREATE_COMPLETE，並將執行狀態設定為 AVAILABLE。堆疊狀態更新為 REVIEW_IN_PROGRESS (正在檢閱)。此時，您可以執行變更集來完成建立新堆疊。

在 Changes (變更) 窗格中，CloudFormation 會顯示堆疊的建議組態。

如果 CloudFormation 無法建立變更集，它會將變更集狀態設為 CREATE_FAILED (建立失敗)。修正 Status reason (狀態原因) 欄位中顯示的錯誤，然後建立新的變更集。在此階段，您可以嘗試在執行下一個變更集前檢查各種組態，並針對堆疊進行修正與更改。

3. 若要根據變更集完成建立新堆疊，請選擇 Execute (執行)，指定復原組態，然後選擇 Execute change set (執行變更集)。

在 CloudFormation 主控台中檢視堆疊資訊

建立 CloudFormation 堆疊之後，您可以使用 AWS 管理主控台 來檢視其資料和資源。

檢視 CloudFormation 堆疊的相關資訊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
3. 在堆疊頁面中，選擇堆疊名稱。CloudFormation 會顯示所選取堆疊的堆疊詳細資訊。

Note

如果已刪除堆疊，可以使用篩選狀態選項找到它。如需詳細資訊，請參閱[在 CloudFormation 主控台中檢視已刪除的堆疊](#)。

您能夠檢視下列堆疊資訊：

堆疊資訊

顯示堆疊及其組態的一般資訊，包括：

概觀

顯示堆疊名稱、堆疊 ID、根堆疊和 [IAM 角色](#) 以及狀態資訊，例如堆疊狀態、偏離狀態和終止保護。

Tags (標籤)

會顯示與堆疊相關聯的任何標籤。

Stack policy (堆疊政策)

會說明能防止堆疊更新的堆疊資源。您必須在堆疊更新期間明確指定允許的資源，才可以更新這些資源。如需詳細資訊，請參閱[避免更新堆疊資源](#)。

復原組態

顯示 CloudFormation 在堆疊操作期間或指定監控期間應該監控的任何指定 CloudWatch 警示。如有任何警示在堆疊操作或監控期間進入 ALARM 狀態，CloudFormation 將會復原整個堆疊操作。如需詳細資訊，請參閱[透過復原觸發條件，在警示違規時復原您的 CloudFormation 堆疊](#)。

通知選項

顯示傳送堆疊事件通知的 Amazon Simple Notification Service 主題 (如指定)。

事件

會顯示建立、更新或刪除堆疊時所追蹤的操作。如需詳細資訊，請參閱[監控堆疊進度](#)。

針對特定堆疊操作觸發的所有事件，系統會指派相同的用戶端請求字符，您即可透過該字符追蹤操作。為了讓您能輕鬆地辨識堆疊操作，從主控台啟動的堆疊操作皆會採用 Console-StackOperation-ID 字符格式。舉例來說，若您使用主控台建立堆疊，則系統將指派相同字符給每個產生的堆疊事件，該字符格式如下所示：Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002。

Resources

會顯示堆疊內的資源。

輸出

會顯示堆疊範本中所宣告的輸出。如需詳細資訊，請參閱[從部署的 CloudFormation 堆疊取得匯出的輸出](#)。

Parameters

會顯示堆疊的參數及其值。

對於包含 Systems Manager 參數的堆疊，解析的值資料欄會顯示 Systems Manager 參數的堆疊定義中所使用的值。如需詳細資訊，請參閱[使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)。

範本

會顯示堆疊的範本。

針對包含巨集的堆疊，請選擇 View original template (檢視原始範本) 以檢視使用者提交的範本，或選擇 View processed template (檢視已處理範本) 以檢視 CloudFormation 處理參考巨集後的範本。CloudFormation 會使用已處理範本來建立或更新堆疊。

變更集

顯示堆疊的變更集。

如需詳細資訊，請參閱[檢視 CloudFormation 堆疊的變更集](#)。

Git 同步

顯示堆疊的 Git 同步儀表板。

如需詳細資訊，請參閱[Git 同步狀態儀表板](#)。

更新您的堆疊範本

要修改 CloudFormation 堆疊中的資源或屬性，您必須更新堆疊的範本。以該堆疊的現有範本為基礎，對其進行變更。如果您要在原始程式碼控制系統中管理範本，則請將該範本的複本做為起點。不然，您亦可從 CloudFormation 取得範本的複本。

若您僅希望變更堆疊的參數或設定 (例如：堆疊的 Amazon SNS 主題)，可以無需複本，重複使用現有堆疊範本。

您可以使用文字編輯器或 [Infrastructure Composer](#) 來更新 CloudFormation 堆疊範本。

透過 Infrastructure Composer 更新現有堆疊範本

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在堆疊頁面中，選擇要更新的堆疊名稱。
3. 選擇範本索引標籤，再選擇 Infrastructure Composer 中的檢視。

CloudFormation 在 Infrastructure Composer 中開啟範本。

4. 使用下列方法之一更新您的範本：
 - 畫布介面：在這裡，您可以從資源面板中拖放。按兩下卡片以開啟資源屬性面板來設定資源。根據需要連接資源。如需使用畫布介面的詳細說明，請參閱[如何在 AWS Infrastructure Composer 中編寫](#)。
 - 範本介面：從畫布切換到範本介面。對範本程式碼進行內嵌更新。根據需要在 JSON 和 YAML 格式之間切換。
5. 選擇驗證，檢查範本中的語法錯誤。
6. 準備好將變更匯出至 CloudFormation 時，選擇更新範本。

使用 更新現有的堆疊範本 AWS CLI

1. 如需取得要更新的堆疊範本，請採用 [get-template](#) CLI 命令。
2. 複製範本並將其貼至文字檔案，進行修改後加以儲存。請務必僅複製範本。該命令會以引號括住範本，請勿複製括住該範本的引號。範本本身會以左大括號開始，並以右大括號結束。您可以在此檔案中指定要進行的堆疊資源變更。

變更範本時，請注意下列事項：

- 在資源不支援更新的情況下，您將無法新增、修改或刪除該資源所使用的參數。
- 對於大多數資源而言，變更資源的邏輯名稱相當於刪除該資源，且會用新資源取而代之。如果任何其他資源與重新命名的資源之間有依存關係，則需一併進行更新，亦可能會遭取代。為了啟動更新作業，您必須更新其他資源的屬性，而不是只變更邏輯名稱。
- 部分資源對特定屬性的可設定數值有約束條件。例如，變更 RDS 資料庫執行個體的 `AllocatedStorage` 屬性時，新數值必須大於目前數值。若您的更新違反這些規則，該部分操作將會失敗。
- 更新一項資源也可能影響參考它的其他資源。若您使用 [Ref 函數](#) 或 [Fn::GetAtt 函數](#) 等函數，根據另一項資源設定某個屬性，則當被參考的資源變更時，CloudFormation 也會更新參考它的資源。
- 如需有關更新特定資源屬性效果的資訊，請參閱 [AWS 資源和屬性類型參考](#)。更新各個屬性將產生下列其中一種效果：
 - 更新要求：[不中斷](#)
 - 更新要求：[部分中斷](#)
 - 更新要求：[替換](#)
- 您可以使用 [validate-template](#) CLI 命令，或在主控台上指定範本，驗證範本的 JSON 或 YAML 語法。主控台會自動執行驗證。但這些方法僅驗證您範本的語法，而不會驗證您為資源指定的屬性數值是否對該資源有效。如需更複雜的驗證或檢查最佳實務，也可以使用其他工具，如 [CloudFormation Linter \(cfn-lint\)](#) 和 [CloudFormation Rain \(rain fmt\)](#)。

Note

有時 CloudFormation 會禁止您嘗試進行的某些變更，並告知您該變更不被允許。然而在預設情況下，CloudFormation 會採隨機順序建立與更新資源，因此系統可能以非同步方式顯示該訊息。

了解更新堆疊資源的行為

當您提交更新時，會根據您提交的內容與堆疊目前範本之間的差異來 CloudFormation 更新資源。尚未變更的資源會在更新程序期間執行，無需中斷。針對更新的資源，CloudFormation 會使用下列其中一個更新行為：

更新時不會中斷

CloudFormation 會更新資源，而不會中斷該資源的操作，也不會變更資源的實體 ID。例如，如果您更新 [AWS::CloudTrail::Trail](#) 資源上的特定屬性，會 CloudFormation 更新追蹤而不會中斷。

某些中斷的更新

CloudFormation 會在部分中斷時更新資源。例如，如果您更新 [AWS::EC2::Instance](#) 資源上的特定屬性，執行個體可能會在 CloudFormation 和 Amazon EC2 重新設定執行個體時受到一些中斷。

取代

CloudFormation 在更新期間重新建立資源，這也會產生新的實體 ID。CloudFormation 通常會先建立替代資源、變更其他相依資源的參考以指向替代資源，然後刪除舊資源。例如，如果您更新 [AWS::EC2::Instance](#) 資源類型的 AvailabilityZone 屬性，會 CloudFormation 建立新的資源，並將目前的 EC2 執行個體資源取代為新的資源。

如果您要新增或移除需要取代的屬性，亦會觸發更新。即使屬性的實際值沒有改變，也會進行更新。

方法 CloudFormation 使用 取決於您為指定資源類型更新的屬性。[AWS 資源和屬性類型參考](#) 中將說明每個屬性的更新行為。

視更新行為而定，您可以決定何時修改資源，來降低這些變更對您的應用程式造成的影響。特別是，您可以規劃何時必須在更新期間「取代」資源。例如，如果您更新 [AWS::RDS::DBInstance](#) 資源類型的 Port 屬性，會使用更新後的連接埠設定建立新的資料庫執行個體來 CloudFormation 取代資料庫執行個體，並刪除舊的資料庫執行個體。更新之前，您可能會規劃執行下列作業以準備取代資料庫：

- 擷取目前資料庫的快照。
- 準備策略，讓使用該資料庫執行個體的應用程式可以處理資料庫執行個體被取代期間的中斷情況。
- 確保使用該資料庫執行個體的應用程式將更新的連接埠設定及您所做的任何其他更新列入考量。
- 使用資料庫快照在新的資料庫執行個體上還原資料庫。

此範例並不完整，旨在提供您想法來規劃何時在更新期間取代資源。

 Note

如果範本包含一或多個[巢狀堆疊](#)，CloudFormation 也會為每個巢狀堆疊啟動更新。這對於判斷巢狀堆疊是否已修改是必要的。只會 CloudFormation 更新巢狀堆疊中具有對應範本中指定變更的那些資源。

透過變更集更新 CloudFormation 堆疊

當您需要更新堆疊時，可以先掌握變更對執行中資源的影響，再進行實作；如此一來，即可安心地更新堆疊。變更集可讓您預覽對堆疊的擬議變更可能對執行中資源造成的影響，包括對資源屬性和屬性值的影響。唯有在確定要執行變更集時，CloudFormation 才會進行堆疊變更，讓您能藉此決定要繼續執行請求的變更，或是建立另一個變更集以探索其他變更。您可以使用 CloudFormation 主控台 AWS CLI 或 CloudFormation API 來建立和管理變更集。

主題

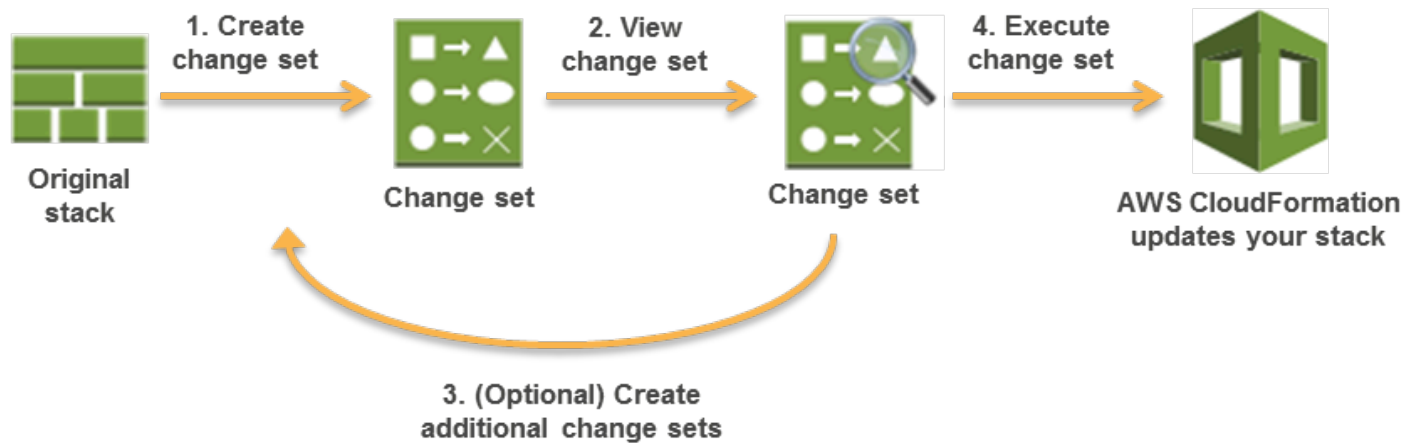
- [建立 CloudFormation 堆疊的變更集](#)
- [檢視 CloudFormation 堆疊的變更集](#)
- [使用偏離感知變更集](#)
- [執行 CloudFormation 堆疊的變更集](#)
- [刪除 CloudFormation 堆疊的變更集](#)
- [CloudFormation 堆疊的變更集範例](#)
- [巢狀堆疊變更集](#)

 Important

變更集不會指示 CloudFormation 是否能成功更新堆疊。例如，變更集並不會檢查是否超過帳戶配額、更新中的資源是否不支援更新，或者具備的許可是否不足以修改資源，上述所有情況都可能導致堆疊更新失敗。若更新失敗，CloudFormation 即會嘗試將資源復原回原始狀態。

變更集概觀

下圖將摘要說明利用變更集更新堆疊的方式：



1. 為欲更新的堆疊提交變更，進而建立變更集；您可以提交修改後的堆疊範本或修改後的輸入參數值。CloudFormation 會將堆疊與提交的變更進行比較，以便產生變更集。此時，系統還不會變更堆疊。
2. 檢視變更集，即可掌握會變更的堆疊設定與資源。例如，您可以查看 CloudFormation 要新增、修改或刪除的資源。此外，您還可以檢視 CloudFormation 即將修改的資源屬性和屬性值 (如標籤) 的前後對比。
3. 選用：如果您在確定要進行的變更之前，還需要考慮其他變更，則請建立額外的變更集。建立多個變更集將有助於您了解並評估不同變更對資源和屬性造成的影響。且您可以視需要建立任意數量的變更集。
4. 執行變更集，該變更集內含要套用至堆疊的變更。CloudFormation 將利用這些變更來更新堆疊。

Note

一旦執行變更，CloudFormation 即會移除所有與堆疊相關聯的變更集，因為這些變更集不適用於更新後的堆疊。

您亦可自行刪除變更集，以防執行不應套用的變更集。

建立 CloudFormation 堆疊的變更集

若要建立執行中堆疊的變更集，則可以提供修改後的範本、新的輸入參數值，或同時提供兩者，藉此提交要進行的變更。CloudFormation 會比較堆疊與提交的變更，進而產生變更集。

您可以在建立變更集之前或在建立變更集期間修改範本。

Create a change set (console)

建立變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選擇您要建立變更集的執行中堆疊。
4. 在「堆疊詳細資訊」窗格中，選擇堆疊動作，然後選擇建立變更集。
5. 在 Create change set for **stack-name** (建立堆疊名稱的變更集) 頁面上，執行以下其中一項操作，修改輸入參數值、指定更新範本的位置或修改範本：

任務	Action
修改輸入參數值	選擇使用目前的範本，然後選擇下一步，繼續輸入或修改輸入參數值。
指定更新的範本位置	如已修改範本，請選擇取代現有的範本，然後執行以下其中一項操作： • 如需存放在 Amazon S3 儲存貯體的範本，請選擇 Amazon S3 URL。輸入或貼上該範本的 URL，然後選擇 Next (下一步)。 如果您擁有已啟用版本控制的儲存貯體中的範本，則可以指定範本的特定版本，方法是將 <code>?versionId= version-id</code> 附加到 URL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用已啟用版本控制之儲存貯體中的物件。 • 針對存放在本機電腦上的範本，選擇 Upload a template file (上傳範本檔案)。選擇 Choose File (選擇檔案) 導覽至該檔案並選取它，然後選擇 Next (下一步)。
修改範本	如果您尚未修改範本，請在選擇 Infrastructure Composer 中編輯範本，然後選擇在 Infrastructure Composer 中選擇編輯。您會重新導向到 AWS Infrastructure Composer。修改範本後，請選擇建立變更集，然後選擇確認並繼續前往 CloudFormation 以返回 建立 stack-name 變更集頁面，然後選擇下一步。

6. 在指定堆疊詳細資訊頁面，在概觀區段指定變更集名稱以及選擇性指定變更集描述，以識別其用途。如果您的範本包含參數，請在 Specify stack details (指定堆疊詳細資訊) 頁面上，輸入或修改合適的輸入參數值，然後選擇 Next (下一步)。

若重複使用堆疊的範本，CloudFormation 會將堆疊中目前的值填入各參數 (不包含 NoEcho 屬性所宣告的參數)。選取 Use existing value (使用現有的值)，即可使用現有的參數值。

如需使用 NoEcho 遮罩敏感資訊，以及使用動態參數管理密碼的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

7. 在設定堆疊選項頁面上，更新堆疊標籤、IAM 服務角色、堆疊原則、復原組態、Amazon SNS 通知主題 (如果適用) 或變更集。

 Note

巢狀堆疊的變更集預設為 Enabled (啟用)，這將會為範本中指定的所有巢狀堆疊建立變更集。若要僅為目前堆疊建立變更集，請選擇 Disabled (停用)。若要取得有關巢狀堆疊變更集的更多資訊，請參閱 [巢狀堆疊變更集](#)。

8. 如果範本包括 IAM 資源，請在功能中選擇我確認 CloudFormation 可能會建立 IAM 資源。IAM 資源可修改 AWS 帳戶中的許可；請檢閱這些資源，確保系統僅能執行所需的動作。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。
9. 選擇 Next (下一步) 繼續。
10. 在 Review **stack-name** (檢閱堆疊名稱) 頁面上，檢閱此變更集的變更。
11. 選擇提交。

您會被重新導向到變更集詳細資訊頁面的 Changes (變更) 標籤。CloudFormation 產生變更集時，該變更集會顯示 CREATE_PENDING 狀態。變更集建立完成後，CloudFormation 即會將狀態設為 CREATE_COMPLETE。在 Changes (變更) 區段中，CloudFormation 會列出將對堆疊進行的所有變更。如需詳細資訊，請參閱[檢視 CloudFormation 堆疊的變更集](#)。

選擇屬性層級變更欄中的檢視詳細資訊，以檢視在屬性層級所做的變更。

若 CloudFormation 無法建立變更集 (回報 FAILED 狀態)，請修復 Status (狀態) 欄位中顯示的錯誤，接著重新建立變更集。

12. 確認變更看起來正確後，選擇執行變更集

Create a change set for nested stacks (console)

建立巢狀堆疊的變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選取您要建立變更集的執行中堆疊。
4. 在「堆疊詳細資訊」窗格中，選擇堆疊動作，然後選擇建立變更集。
5. 在 Create change set for **stack-name** (建立堆疊名稱的變更集) 頁面上，執行以下其中一項操作，修改輸入參數值、指定更新範本的位置或修改範本：

任務	Action
修改輸入參數值	選擇使用目前的範本，然後選擇下一步，繼續輸入或修改輸入參數值。
指定更新的範本位置	如已修改範本，請選擇取代現有的範本，然後執行以下其中一項操作： • 如需存放在 Amazon S3 儲存貯體的範本，請選擇 Amazon S3 URL。輸入或貼上該範本的 URL，然後選擇 Next (下一步)。 如果您擁有已啟用版本控制的儲存貯體中的範本，則可以指定範本的特定版本，方法是將 <code>?versionId= version-id</code> 附加到 URL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用已啟用版本控制之儲存貯體中的物件。 • 針對存放在本機電腦上的範本，選擇 Upload a template file (上傳範本檔案)。選擇 Choose File (選擇檔案) 導覽至該檔案並選取它，然後選擇 Next (下一步)。
修改範本	如果您尚未修改範本，請在選擇 Infrastructure Composer 中編輯範本，然後選擇在 Infrastructure Composer 中選擇編輯。您會重新導向到 AWS Infrastructure Composer。修改範本後，請選擇建立變更集，然後選擇確認並繼續前往 CloudFormation 以返回 建立 stack-name 變更集頁面，然後選擇下一步。

- 在指定堆疊詳細資訊頁面，在概觀區段指定變更集名稱以及選擇性指定變更集描述，以識別其用途。如果您的範本包含參數，請在 Specify stack details (指定堆疊詳細資訊) 頁面上，輸入或修改合適的輸入參數值，然後選擇 Next (下一步)。

若重複使用堆疊的範本，CloudFormation 會將堆疊中目前的值填入各參數 (不包含 NoEcho 屬性所宣告的參數)。選取 Use existing value (使用現有的值)，即可使用現有的參數值。

如需使用 NoEcho 遮罩敏感資訊，以及使用動態參數管理密碼的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

- 在設定堆疊選項頁面上，更新堆疊標籤、IAM 服務角色、堆疊原則、復原組態、Amazon SNS 通知主題 (如果適用) 或變更集。如需詳細資訊，請參閱 [設定堆疊選項](#)。

 Note

巢狀堆疊的變更集預設為 Enabled (啟用)，這將會為範本中指定的所有巢狀堆疊建立變更集。若要取得有關巢狀堆疊變更集的更多資訊，請參閱 [巢狀堆疊變更集](#)。

- 如果範本包括 IAM 資源，請在功能中選擇我確認 CloudFormation 可能會建立 IAM 資源。IAM 資源可修改 AWS 帳戶中的許可；請檢閱這些資源，確保系統僅能執行所需的動作。如需詳細資訊，請參閱 [認可 CloudFormation 範本中的 IAM 資源](#)。
- 選擇 Next (下一步) 繼續。
- 在 Review **stack-name** (檢閱堆疊名稱) 頁面上，檢閱此變更集的變更。
- 選擇提交。

 Note

當您為巢狀堆疊建立變更集時，CloudFormation 屬性層級變更集不會解析跨堆疊參考。若變更集參考父堆疊的輸出，且父堆疊已修改，則變更集可以標記子堆疊中的資源以進行條件式取代。

您會被重新導向到變更集詳細資訊頁面的 Changes (變更) 標籤。CloudFormation 產生變更集時，該變更集會顯示 CREATE_PENDING 狀態。變更集建立完成後，CloudFormation 即會將狀態設為 CREATE_COMPLETE。在 Changes (變更) 區段中，CloudFormation 會列出將對堆疊進行的所有變更。如需詳細資訊，請參閱 [檢視 CloudFormation 堆疊的變更集](#)。

若 CloudFormation 無法建立變更集 (回報 FAILED 狀態)，請修復 Status (狀態) 欄位中顯示的錯誤，接著重新建立變更集。

12. 確認變更看起來正確後，選擇執行變更集

建立變更集 (AWS CLI)

- 使用 [create-change-set](#) 命令。

您能夠以命令選項的形式來提交變更，亦能指定新的參數值、修改後的範本，或同時指定兩者。例如，以下命令將為 SampleChangeSet 堆疊建立 MyStack 變更集。該變更集會使用目前的堆疊範本，但 Purpose 參數將具備不同值：

```
aws cloudformation create-change-set --stack-name MyStack \  
  --change-set-name SampleChangeSet --use-previous-template \  
  --parameters \  
    ParameterKey="InstanceType",UsePreviousValue=true  
    ParameterKey="KeyPairName",UsePreviousValue=true  
    ParameterKey="Purpose",ParameterValue="production"
```

檢視 CloudFormation 堆疊的變更集

建立變更集之後，即可先檢視請求的變更，再加以執行。您可以使用 CloudFormation 主控台 AWS CLI 或 CloudFormation API 來檢視變更集。CloudFormation 主控台會提供變更摘要，並以 JSON 格式詳列變更清單；AWS CLI 和 AWS CloudFormation API 會傳回 JSON 格式變更的詳細清單。

View a change set (console)

檢視變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選擇包含想要檢視變更集的堆疊名稱。
4. 在導覽窗格中，選擇 Change Sets (變更集) 以檢視堆疊的變更集清單。
5. 選擇要檢視的變更集名稱。

CloudFormation 主控台會將您導向至變更集的詳細資訊頁面，讓您可以查看變更集的建立時間、狀態、用來產生變更集的輸入，以及變更的摘要。

在 Changes (變更) 區段中，每一列代表 CloudFormation 將新增、修改或移除的資源。

- Add (新增) – 當您將資源新增至堆疊範本時，CloudFormation 會建立資源。
- Modify (修改) – 當您變更堆疊範本中的資源屬性時，CloudFormation 會修改資源。
- Remove (移除) – 當您從堆疊範本中刪除資源時，CloudFormation 即會刪除該資源。

 Note

修改作業可能會造成系統終止或替換 (重新建立) 資源。如需資源更新行為的詳細資訊，請參閱 [了解更新堆疊資源的行為](#)。

您可以使用篩選條件檢視，以便著重特定的變更。舉例來說，篩選特定的資料類型，例如 `AWS::EC2::Instance`。若要篩選特定資源，請指定該資源的邏輯或實體 ID，例如：`myWebServer` 或 `i-123abcd4`。

6. 在變更區段中，選擇屬性層級變更欄中的檢視詳細資訊，即可檢視對資源所做的屬性值變更。
7. CloudFormation 主控台會引導您前往資源的屬性層級變更頁面，您可以在其中查看執行變更集前的資源範本組態，以及範本組態在執行變更集之後的外觀。

屬性層級變更區段表格顯示受影響屬性的路徑、變更類型、之前值及之後值。在表格中，選擇您要在範本之前和之後檢視中反白顯示的每個變更的核取方塊，以查看屬性層級將進行的變更。

- 新增 – 新增的屬性會以綠色反白顯示。
- 修改 – 修改的屬性會以藍色反白顯示。
- 移除 – 移除的屬性會以紅色反白顯示。

View a change set for nested stack (console)

檢視巢狀堆疊的變更集 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選擇包含想要檢視變更集的堆疊名稱。
4. 在導覽窗格中，選擇變更集以檢視堆疊的變更集清單。

5. 選擇要檢視的變更集名稱。

CloudFormation 主控台會將您導向至變更集的詳細資訊頁面，讓您可以查看變更集的建立時間、狀態、用來產生變更集的輸入，以及變更的摘要。

在 Changes (變更) 區段中，每一列代表 CloudFormation 將新增、修改、移除或顯示動態狀態的資源。

- Add (新增) – 當您將資源新增至堆疊範本時，CloudFormation 會建立資源。
- Modify (修改) – 當您變更堆疊範本中的資源屬性時，CloudFormation 會修改資源。
- Remove (移除) – 當您從堆疊範本中刪除資源時，CloudFormation 即會刪除該資源。
- Dynamic (動態) – CloudFormation 無法確定巢狀堆疊範本的確切資源變更動作。

Note

修改作業可能會造成系統終止或替換 (重新建立) 資源。如需資源更新行為的詳細資訊，請參閱 [了解更新堆疊資源的行為](#)。

您可以使用篩選條件檢視，以便著重特定的變更。舉例來說，篩選特定的資料類型，例如 **AWS::CloudFormation::Stack**。若要篩選特定資源，請指定該資源的邏輯或實體 ID，例如：**DeadLetterQueue** 或 **NestedStack**。

6. 在 Changes (變更) 區段中，選擇您要檢視之巢狀變更集的 View nested change set (檢視巢狀變更集)。

CloudFormation 主控台會將您導向巢狀變更集的詳細資訊頁面。您可以選擇 Go to root change set (跳至根變更集) 來檢視根變更集，或者選擇 View parent change set (檢視父項變更集) 來檢視父項變更集。如需詳細資訊，請參閱 [巢狀堆疊變更集](#)。

Note

當您為巢狀堆疊建立變更集時，CloudFormation 屬性層級變更集不會解析跨堆疊參考。若變更集參考父堆疊的輸出，且父堆疊已修改，則變更集可以標記子堆疊中的資源以進行條件式取代。

檢視變更集 (AWS CLI)

1. 執行 [change-sets](#) 命令，即可取得變更集的 ID。

指定堆疊的名稱，該堆疊內含您要檢視的變更集；如以下範例所示：

```
aws cloudformation list-change-sets --stack-name MyStack
```

CloudFormation 會傳回變更集清單，與下方內容相似：

```
{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
      "Status": "CREATE_COMPLETE",
      "ChangeSetName": "SampleChangeSet",
      "CreationTime": "2020-11-18T20:44:05.889Z",
      "StackName": "MyStack",
      "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
    },
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
      "Status": "CREATE_COMPLETE",
      "ChangeSetName": "SampleChangeSet-conditional",
      "CreationTime": "2020-11-18T21:15:56.398Z",
      "StackName": "MyStack",
      "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet-conditional/1a2345b6-0000-00a0-a123-00abc0abc000"
    },
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
      "Status": "CREATE_COMPLETE",
      "ChangeSetName": "SampleChangeSet-replacement",
      "CreationTime": "2020-11-18T21:03:37.706Z",
      "StackName": "MyStack",
      "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet-replacement/1a2345b6-0000-00a0-a123-00abc0abc000"
    }
  ]
}
```

```
    }
  ]
}
```

2. 執行 [describe-change-set](#) 命令，藉此指定要檢視的變更集 ID。例如：

```
aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

CloudFormation 會傳回特定變更集的相關資訊。

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "ellioty-useast1",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Static",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          }
        ]
      }
    }
  ]
}
```

```

        }
      },
      "Action": "Modify",
      "Scope": [
        "Tags"
      ],
      "LogicalResourceId": "MyEC2Instance",
      "Replacement": "False"
    },
    "Type": "Resource"
  }
],
"CreationTime": "2020-11-18T23:35:25.813Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/9edde307-960d-4e6e-ad66-b09ea2f20255"
}

```

使用 `--include-property-values` 搭配 `describe-change-set`，以列出屬性層級變更。

Changes 金鑰會列出系統對資源進行的變更。CloudFormation 會更新 `i-1abc23d4` EC2 執行個體的標籤，讓您得以執行此變更集。如需各欄位的說明，請參閱《AWS CloudFormation API 參考》中的 [Change 資料類型](#)。

如需變更集的其他範例，請參閱 [CloudFormation 堆疊的變更集範例](#)。

檢視變更集中的屬性層級變更 (AWS CLI)

- 下列命令會列出與 `AWS::EC2::NetworkInterface` 資源變更集相關的屬性層級變更，該變更集會移除 `Ipv4Prefixes` 屬性、修改資源的 `Description`，並新增 `Tag`：

```
aws cloudformation describe-change-set --include-property-values \
  --change-set-name arn:aws:cloudformation:us-east-1:123456789012:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841
```

以下為範例輸出。

```
"ChangeSetName": "ExampleChangeSet",
```

```

    "ChangeSetId": "arn:aws:cloudformation:us-east-1:803642222207:changeSet/
ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
    "StackId": "arn:aws:cloudformation:us-east-1:803642222207:stack/ExampleStack/
ab664180-f686-11ee-9e29-12cd92393671",
    "StackName": "ExampleStack",
    "Description": null,
    "Parameters": null,
    "CreationTime": "2024-04-09T18:04:59.935000+00:00",
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE",
    "StatusReason": null,
    "NotificationARNs": [],
    "RollbackConfiguration": {
        "RollbackTriggers": []
    },
    "Capabilities": [],
    "Tags": null,
    "ParentChangeSetId": null,
    "IncludeNestedStacks": true,
    "RootChangeSetId": null,
    "OnStackFailure": null,
  {
    "Changes": [
      {
        "Type": "Resource",
        "ResourceChange": {
          "Action": "Modify",
          "LogicalResourceId":
"EC2NetworkInterface00eni067fd35b649a05b7100Tpyls",
          "PhysicalResourceId": "eni-067fd35b649a05b71",
          "ResourceType": "AWS::EC2::NetworkInterface",
          "Replacement": "False",
          "Scope": [
            "Properties",
            "Tags"
          ],
          "Details": [
            {
              "Target": {
                "Attribute": "Properties",
                "Name": "Ipv4Prefixes",
                "RequiresRecreation": "Never",
                "Path": "/Properties/Ipv4Prefixes",
                "BeforeValue": "[]",

```



```

        "AttributeChangeType": "Remove"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
},
{
    "Target": {
        "Attribute": "Properties",
        "Name": "Description",
        "RequiresRecreation": "Never",
        "Path": "/Properties/Description",
        "BeforeValue": "",
        "AfterValue": "Description",
        "AttributeChangeType": "Modify"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
},
{
    "Target": {
        "Attribute": "Tags",
        "RequiresRecreation": "Never",
        "Path": "/Properties/Tags/0",
        "AfterValue": "{\"Key\":\"Test\",\"Value\":\"Test\"}",
        "AttributeChangeType": "Add"
    },
    "Evaluation": "Static",
    "ChangeSource": "DirectModification"
}
],
    "BeforeContext": "{\"Properties\":{\"Description\":\"\",
    \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount\": \"0\",
    \"Ipv6PrefixCount\": \"0\", \"Ipv4Prefixes\": [], \"Ipv4PrefixCount\": \"0\", \"GroupSet
    \": [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": []},
    \"UpdateReplacePolicy\": \"Retain\", \"DeletionPolicy\": \"Retain\"}",
    "AfterContext": "{\"Properties\":{\"Description\":\"Description\",
    \"PrivateIpAddress\":\"172.31.76.2\", \"PrivateIpAddresses\": [{\"PrivateIpAddress
    \": \"172.31.76.2\", \"Primary\": \"true\"}], \"SecondaryPrivateIpAddressCount
    \": \"0\", \"Ipv6PrefixCount\": \"0\", \"Ipv4PrefixCount\": \"0\", \"GroupSet\":
    [\"sg-05a45689b1059e82d\"], \"Ipv6Prefixes\": [], \"SubnetId\": \"subnet-455e8969\",
    \"SourceDestCheck\": \"true\", \"InterfaceType\": \"interface\", \"Tags\": [{\"Value\":

```

```
\\"Test\\",\\"Key\\":\\"Test\\"}}],\\"UpdateReplacePolicy\\":\\"Retain\\",\\"DeletionPolicy\\":\\"Retain\\"}"
    }
  ],
  "ChangeSetName": "ExampleChangeSet",
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/ExampleChangeSet/9f7b541b-126b-44f7-998e-932174557841",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/ExampleStack/ab664180-f686-11ee-9e29-12cd92393671",
```

使用偏離感知變更集

偏離感知變更集是增強型 CloudFormation 變更集，可讓您以安全的方式識別和管理堆疊偏離。當堆疊資源在 CloudFormation 外部變更、使用基礎服務的或 SDK 時 AWS 管理主控台 AWS CLI，堆疊偏離預期的範本組態。偏離感知變更集會將範本與堆疊資源的實際狀態進行比較，並使偏離的資源與其範本定義保持一致。更新資源的範本定義以符合其實際狀態時，偏離感知變更集會重設資源的偏離狀態，而不會修改資源。

使用偏離感知變更集的優點包括：

- 預覽覆寫偏離 – 偵測部署是否會還原事件回應期間所做的重大out-of-band變更。
- 系統性偏離對帳 – 透過使用範本值覆寫偏離屬性或更新範本以符合實際資源狀態，將偏離的資源與合規範本定義安全地對齊。
- 增強的復原功能 – 如果部署失敗，請將資源還原至其實際的部署前狀態，而非先前的範本狀態。
- 完整可見性 – 在進行變更之前，請確實查看您的部署將如何影響實際的資源組態。

傳統變更集提供新範本與堆疊先前範本的比較，但不考慮堆疊偏離。偏離感知變更集透過提供以下項目之間的三向比較來解決此問題：

- 實際狀態 – 資源的即時組態。CloudFormation 將在透過基礎服務 APIs 建立變更集時讀取您帳戶中的資源組態。
- 先前的部署狀態 – 您上次 CloudFormation 部署的範本中定義的組態。
- 所需狀態 – 新範本中定義的組態。

即使範本中未明確變更資源，偏離感知變更集也會更新所有堆疊資源的實際狀態，以符合所需狀態。

考量事項

- 資源類型支援 – 偏離感知變更集支援將所需狀態與數百種資源類型的實際狀態進行比較。對於不支援的資源類型，偏離感知變更集會回復為比較先前的部署狀態與所需的狀態。如需詳細資訊，請參閱 [資源類型](#) 一節。
- 唯讀屬性 – 對於包含敏感資料（密碼、秘密）的屬性，偏離感知變更集會與先前的部署值比較，而非實際值。
- AWS受管屬性 – 偏離感知變更集會保留 AWS 服務對受管屬性所做的out-of-band變更，例如 Auto Scaling 群組所需的容量。如需詳細資訊，請參閱 [AWS受管屬性](#) 一節。
- 外部標籤金鑰 – 偏離感知變更集不會移除或修改範本中未指定的標籤金鑰，以防止與屬性型存取控制 (ABAC) 系統發生衝突。
- 取代漂移資源 – 漂移感知變更集不支援不可變屬性的漂移對帳。
- 跨堆疊連接 – 有些資源類型，例如 `AWS::IAM::Policy`，可能會影響其他資源類型的實際狀態，例如 `AWS::IAM::Role`。偏離感知變更集會處理堆疊內資源的連接。如果透過從不同堆疊連接次要資源來修改資源，偏離感知變更集會將修改偵測為偏離，並且可以還原連接。熱門的可連接資源包括 `AWS::IAM::Policy`、`AWS::EC2::SecurityGroupIngress`、`AWS::IAM::ManagedPolicy` 和 `AWS::EC2::SecurityGroupEgress`。

AWS受管屬性

您可以設定作用中 AWS 管理的特定資源屬性。例如，您可以允許 Amazon Relational Database Service (Amazon RDS) 自動升級 Amazon RDS 資料表的次要引擎版本。這些變更可能會在 CloudFormation 中顯示為堆疊偏離。偏離感知變更集會辨識 AWS受管屬性的預期偏離，如果您尚未在範本中修改屬性，則讓其實際值保持不變。AWS受管屬性的熱門範例如下：

- 啟用 Amazon RDS 資料表的 `AutoMinorVersionUpgrade` 屬性，以允許引擎版本的自動更新。
- 使用 `AWS::ApplicationAutoScaling::ScalableTarget` 資源為 Amazon DynamoDB 資料表的讀取/寫入容量單位和 Amazon Elastic Container Service 叢集的所需計數等屬性啟用自動擴展。
- 使用 `AWS::AutoScaling::ScalingPolicy` for Auto Scaling 群組。

偏離感知變更集會釐清識別為 AWS受管的屬性。如需詳細資訊，請參閱 [AWS CLI](#) 一節。

使用偏離感知變更集（主控台）

您可以使用與傳統變更集相同的工作流程，以及部署模式的其他選項，透過 CloudFormation 主控台建立和管理偏離感知變更集。

建立偏離感知變更集

建立偏離感知變更集（主控台）

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在堆疊頁面上，選取您要更新的堆疊。
3. 選擇堆疊動作，然後選擇建立目前堆疊的變更集。
4. 在建立變更集頁面上，針對變更集類型，選擇偏離感知變更集。
5. 選擇 Create change set (建立變更集)。
6. 在變更集詳細資訊頁面上，檢閱顯示每個受影響資源的實際、先前部署和所需狀態的三向比較。
7. 如果您對變更感到滿意，請選擇執行變更集。

檢閱偏離感知變更集

當您在主控台中檢視偏離感知變更集時，您會看到與傳統變更集相比的增強型資訊：

- 堆疊偏離狀態 – 指出堆疊是否已偏離其上次部署。
- 屬性比較 – 顯示受影響資源的實際和所需狀態之間的 JSON 差異。
- 偏離指標 – 明確標記已偏離的 JSON diff 內的屬性。按一下檢視偏離以查看屬性的先前部署值。
- 值來源指標 – 顯示屬性值之前的值是來自實際狀態還是先前的部署狀態。

使用偏離感知變更集 (AWS CLI)

您可以將 `--deployment-mode REVERT_DRIFT` 參數新增至 `create-change-set` 命令，AWS CLI 以使用 建立和管理偏離感知變更集。

建立偏離感知變更集

建立偏離感知變更集

使用 [create-change-set](#) 命令搭配 `--deployment-mode REVERT_DRIFT` 參數來建立 drift-aware 變更集。

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name my-drift-aware-changeset \  
  --template-body file://updated-template.yaml \  
  --deployment-mode REVERT_DRIFT
```

```
--deployment-mode REVERT_DRIFT \  
--capabilities CAPABILITY_IAM
```

檢閱偏離感知變更集

若要檢閱偏離感知變更集的詳細資訊

使用 [describe-change-set](#) 命令來檢閱 drift-aware 變更集的詳細資訊：

```
aws cloudformation describe-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

回應包含漂移感知變更集的增強資訊：

- StackDriftStatus – 顯示堆疊是否已偏離 (DRIFTED、NOT_CHECKED、IN_SYNC或 UNKNOWN)。
- ResourceDriftStatus – 顯示每個資源 (DELETED、IN_SYNC、MODIFIED或) 的偏離狀態NOT_CHECKED。
- BeforeValueFrom – 指出屬性的之前值是來自 ACTUAL_STATE或 PREVIOUS_DEPLOYMENT_STATE。
- Drift – 包含屬性的偏離詳細資訊，包括 PreviousValue、ActualValue和 DriftDetectionTimestamp。
- ResourceDriftIgnoredProperties – 包含變更集不會還原偏離的資源屬性，以及忽略偏離的原因。

如需詳細資訊，請參閱 AWS CloudFormation API 參考中的 [DescribeChangeSet](#)。

執行偏離感知變更集

執行偏離感知變更集

檢閱變更集後，請使用 [execute-change-set](#) 命令來套用變更：

```
aws cloudformation execute-change-set \  
  --change-set-name my-drift-aware-changeset \  
  --stack-name my-stack
```

如果部署失敗，CloudFormation 會將資源復原至其實際的部署前狀態，並保留out-of-band變更。

支援的資源類型

偏離感知變更集支援所有支援偏離偵測[之資源類型的](#)實際狀態、先前部署狀態和所需狀態的三向比較，但下列資源類型除外：

- `AWS::SageMaker::MlflowTrackingServer`
- `AWS::Route53Resolver::FirewallRuleGroup`
- `AWS::MediaLive::Multiplexprogram`
- `AWS::VpcLattice::ResourceConfiguration`
- `AWS::S3::StorageLensGroup`
- `AWS::Bedrock::AgentAlias`
- `AWS::MSK::Cluster`
- `AWS::RDS::DBProxy`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::QBusiness::Index`
- `AWS::NetworkManager::CoreNetwork`
- `AWS::IAM::OIDCProvider`
- `AWS::Organizations::ResourcePolicy`
- `AWS::SNS::TopicInlinePolicy`
- `AWS::Route53::KeySigningKey`
- `AWS::DataZone::PolicyGrant`
- `AWS::Transfer::Certificate`
- `AWS::SageMaker::ImageVersion`
- `AWS::Neptune::DBParameterGroup`
- `AWS::ODB::CloudVmCluster`
- `AWS::RolesAnywhere::TrustAnchor`
- `AWS::Detective::Graph`
- `AWS::Maester::DocumentType`
- `AWS::SageMaker::ModelPackageGroup`
- `AWS::S3Express::BucketPolicy`
- `AWS::Panorama::PackageVersion`
- `AWS::S3Tables::TableBucketPolicy`

對於不支援三向比較的資源，偏離感知變更集會回復到先前部署狀態和所需狀態的比較。

執行 CloudFormation 堆疊的變更集

只需執行變更集，即可對堆疊進行變更集中所述的變更。

Important

執行變更集後，CloudFormation 就會刪除所有與堆疊相關聯的額外變更集，因為這些變更集對更新後的堆疊不再有效。如果更新失敗，則需要建立新的變更集。

堆疊政策與執行變更集

若您在具有相關聯堆疊政策的堆疊上執行變更集，CloudFormation 即會在更新堆疊時強制執行該政策。此外，您無法在執行變更集時，指定會覆寫現有堆疊政策的暫時政策。您必須更新堆疊政策或使用直接更新法，才能更新受保護的資源。如需詳細資訊，請參閱[直接更新堆疊](#)。

Execute a change set (console)

執行變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面中，選擇您要更新的堆疊名稱。
4. 在導覽窗格中，選擇變更集以檢視堆疊的變更集清單。
5. 選擇您要執行的變更集名稱。
6. 在變更集的詳細資訊頁面上，選擇執行變更集。

CloudFormation 將立即開始更新堆疊；CloudFormation 主控台會將您導向事件索引標籤，您可在這監控堆疊更新的進度。如需詳細資訊，請參閱[監控堆疊進度](#)。

Execute a change set for nested stacks (console)

執行巢狀堆疊的變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。

3. 在堆疊頁面中，選擇您要更新的堆疊名稱。您必須選擇與根變更集相關聯的堆疊名稱。
4. 在導覽窗格中，選擇變更集以檢視堆疊的變更集清單。
5. 選擇您要執行的根變更集名稱。
6. 在變更集的詳細資訊頁面上，選擇執行變更集。

 Note

如果在 [建立 CloudFormation 堆疊的變更集](#) 過程中為巢狀堆疊的變更集選取 Enabled (啟用)，則 CloudFormation 會執行根變更集和巢狀變更集中描述的變更。

CloudFormation 將立即開始更新堆疊；CloudFormation 主控台會將您導向事件索引標籤，您可在該處監控堆疊更新的進度。如需詳細資訊，請參閱[監控堆疊進度](#)。

執行變更集 (AWS CLI)

- 執行 [execute-change-set](#) 命令。

指定要執行的變更集 ID，如以下範例所示：

```
aws cloudformation execute-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

範例中的命令會執行 ID 為 `arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000` 的變更集。

執行該命令之後，CloudFormation 即會開始更新堆疊。若要檢視堆疊的進度，請善用 [describe-stacks](#) 命令。

刪除 CloudFormation 堆疊的變更集

一旦刪除變更集，系統便會從堆疊的變更集清單中將之移除。您可以刪除變更集，以防自己或其他使用者意外執行不應套用的變更集。除非您刪除變更集，否則 CloudFormation 在更新堆疊之前會保留所有變更集。

Delete a change set

刪除變更集 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選擇包含要刪除變更集的堆疊名稱。
4. 在導覽窗格中，選擇變更集以檢視堆疊的變更集清單。
5. 選取您要刪除的變更集名稱。
6. 在變更集的詳細資訊頁面上，選擇刪除變更集。

CloudFormation 會立即開始刪除變更集堆疊清單中的變更集，而您會被重新導向到 Stacks (堆疊) 頁面。

Delete a change set for nested stacks (console)

刪除巢狀堆疊的變更集

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選取與根變更集相關聯的堆疊名稱。
4. 在導覽窗格中，選擇變更集以檢視堆疊的變更集清單。
5. 選取您要刪除的變更集名稱。
6. 在變更集的詳細資訊頁面上，選擇 Delete (刪除)。選擇刪除變更集，您將刪除巢狀變更集的整個階層。

Note

巢狀堆疊變更集的刪除操作是非同步的作業，會顯示 DELETE_PENDING 狀態，後面再顯示 DELETE_IN_PROGRESS 狀態。刪除變更集操作完成後，變更集將從清單中移除。如果在建立變更集期間建立 REVIEW_IN_PROGRESS 狀態中的巢狀堆疊，則該巢狀堆疊也會被刪除。

CloudFormation 會立即開始從堆疊的變更集清單中刪除變更集。

Note

若您有停滯在進行中操作的巢狀堆疊，請參閱[巢狀堆疊復原失敗](#)中的故障診斷錯誤。

刪除變更集 (AWS CLI)

- 執行 [delete-change-set](#) 命令，藉此指定要刪除的變更集 ID，如以下範例所示：

```
aws cloudformation delete-change-set \  
  --change-set-name \  
    arn:aws:cloudformation:us-east-1:123456789012:changeSet/  
SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000
```

CloudFormation 堆疊的變更集範例

本節所提供的變更集範例能讓 CloudFormation 用來建立一般堆疊變更。這些範例將說明如何直接編輯範本、修改單一輸入參數、規劃資源重新建立 (替換) 作業，以免未備份的資料遺失，或是堆疊中執行的應用程式中斷；透過這些範例，您亦能掌握新增與移除資源的方法。為了示範變更集的運作方式，我們將逐步解說您所提交的變更，並探討隨後產生的變更集。本節的每個範例皆是以先前的範例為建立基礎，且會假設您已掌握之前的釋例，因此建議您按照順序詳讀。如需變更集各欄位的說明，請參閱《AWS CloudFormation API 參考》中的 [Change](#) (變更) 資料類型。

您可以使用[主控台](#)、AWS CLI或 CloudFormation [DescribeChangeSet](#) API 操作來檢視變更集詳細資訊。

為了從堆疊中產生下述各個變更集，此處會採用以下[範例範本](#)：

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Description" : "A sample EC2 instance template for testing change sets.",  
  "Parameters" : {  
    "Purpose" : {  
      "Type" : "String",  
      "Default" : "testing",  
      "AllowedValues" : ["testing", "production"],  
      "Description" : "The purpose of this instance."  
    },  
    "KeyPairName" : {
```

```

    "Type": "AWS::EC2::KeyPair::KeyName",
    "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
instance"
  },
  "InstanceType" : {
    "Type" : "String",
    "Default" : "t2.micro",
    "AllowedValues" : ["t2.micro", "t2.small", "t2.medium"],
    "Description" : "The EC2 instance type."
  }
},
"Resources" : {
  "MyEC2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "KeyName" : { "Ref" : "KeyPairName" },
      "InstanceType" : { "Ref" : "InstanceType" },
      "ImageId" : "ami-8fcee4e5",
      "Tags" : [
        {
          "Key" : "Purpose",
          "Value" : { "Ref" : "Purpose" }
        }
      ]
    }
  }
}
}
}
}

```

直接編輯範本

當您為產生變更集而在堆疊範本中直接修改資源時，CloudFormation 即會將變更分類成直接修改作業，而不是由更新後參數值所啟動的變更。以下變更集便是直接修改的範例，其會將新標籤新增至 i-1abc23d4 執行個體。由於要著重探討 Changes 結構，參數值和功能等所有其他輸入值皆會保持不變。

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-direct",
  "Parameters": [
    {

```

```
        "ParameterValue": "testing",
        "ParameterKey": "Purpose"
    },
    {
        "ParameterValue": "MyKeyName",
        "ParameterKey": "KeyPairName"
    },
    {
        "ParameterValue": "t2.micro",
        "ParameterKey": "InstanceType"
    }
],
"Changes": [
    {
        "ResourceChange": {
            "ResourceType": "AWS::EC2::Instance",
            "PhysicalResourceId": "i-1abc23d4",
            "Details": [
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Static",
                    "Target": {
                        "Attribute": "Tags",
                        "RequiresRecreation": "Never"
                    }
                }
            ],
            "Action": "Modify",
            "Scope": [
                "Tags"
            ],
            "LogicalResourceId": "MyEC2Instance",
            "Replacement": "False"
        },
        "Type": "Resource"
    }
],
"CreationTime": "2020-11-18T23:35:25.813Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-direct/1a2345b6-0000-00a0-a123-00abc0abc000"
```

```
}
```

在 `Changes` 結構中，僅有一個 `ResourceChange` 結構。透過此結構的說明，即可了解 CloudFormation 將變更的資源類型、CloudFormation 會採取的動作、資源 ID、變更範圍等資訊，並知曉該變更是否需要執行替換作業 (CloudFormation 會建立新資源，然後刪除舊資源)。在本範例中，變更集會指示 CloudFormation 要修改的 `i-1abc23d4` EC2 執行個體 `Tags` 屬性，且不需要替換執行個體。

在 `Details` 結構中，CloudFormation 會將此變更標記為直接修改，因此絕不用重新建立或替換執行個體。明白 CloudFormation 不會替換執行個體後，您即可安心地執行此變更。

CloudFormation 會將此變更顯示為 `Static` 評估。`Static` 評估表示 CloudFormation 可以先判斷標籤值，再執行變更集。在某些情況下，唯有在執行變更集後，CloudFormation 才能判斷標籤值。而 CloudFormation 會將這類變更標記為 `Dynamic` 評估。假若更新後資源的參考資料將在滿足條件時進行替換作業，則 CloudFormation 便無法判斷更新後資源的參考資料是否會改變。

修改輸入參數值

當您修改輸入參數值時，CloudFormation 即會採用更新後的參數值，藉此為各個資源產生兩項變更。在本範例中，我們會詳細說明這些變更的效果，以及應注重的資訊。系統在產生本範例時，僅有變更 `Purpose` 輸入參數的值。

`Purpose` 參數會指定 EC2 執行個體的標籤金鑰值。本範例會將 `testing` 參數值變更為 `production`。而 `Parameters` 結構即會顯示新的值。

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet",
  "Parameters": [
    {
      "ParameterValue": "production",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ]
}
```

```

    }
  ],
  "Changes": [
    {
      "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [
          {
            "ChangeSource": "DirectModification",
            "Evaluation": "Dynamic",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          },
          {
            "CausingEntity": "Purpose",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
              "Attribute": "Tags",
              "RequiresRecreation": "Never"
            }
          }
        ],
        "Action": "Modify",
        "Scope": [
          "Tags"
        ],
        "LogicalResourceId": "MyEC2Instance",
        "Replacement": "False"
      },
      "Type": "Resource"
    }
  ],
  "CreationTime": "2020-11-18T23:59:18.447Z",
  "Capabilities": [],
  "StackName": "MyStack",
  "NotificationARNs": [],
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/SampleChangeSet/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

此處 Changes 結構的運作方式，將與其在 [直接編輯範本](#) 範例中的運作方式類似。意即該結構僅有一個 ResourceChange 結構，且會說明 Tags EC2 執行個體 i-1abc23d4 屬性的變更。

不過，在 Details 結構中，即便僅變更一個參數值，變更集仍會針對 Tags 屬性顯示兩項變更。若資源是將變更後的參數值作為參考資料 (使用 Ref 內部函數)，一律會產生兩項變更：一項是 Dynamic 評估，另一項則是 Static 評估。如需查看這類型的變更，請檢視下列欄位：

- 若變更屬於 Static 評估，請檢視 ChangeSource 欄位。在本範例中，ChangeSource 欄位等於 ParameterReference，這表示此變更是更新後參數參考值所產生的結果；該變更集必須包含類似的 Dynamic 評估變更。
- 透過比較兩項變更的 Dynamic 結構，即可找到相符的 Target 評估變更，當中將包含相同資訊。在本範例中，兩項變更的 Target 結構皆涵蓋相同的 Attribute 值和 RequireRecreation 欄位。

查看這類型的變更時，應該著重於 Static 評估，因為該評估所提供的變更資訊最為詳細。在本範例中，Static 評估顯示的變更屬於參數參考值 (ParameterReference) 的更改結果；CauseEntity 欄位 (Purpose 參數) 則會確切指示已變更的參數。

判斷 Replacement 欄位的值

ResourceChange 結構中的 Replacement 欄位會指示 CloudFormation 是否應重新建立資源。只需規劃資源的重新建立或替換作業，即可避免未備份的資料遺失，或是堆疊中執行的應用程式中斷。

Replacement 欄位中的值會取決於是否需要替換變更項目，這會視變更 RequiresRecreation 結構中的 Target 欄位指示而定。舉例來說，若 RequiresRecreation 欄位為 Never，則 Replacement 欄位的值將為 False。但若單一資源上有多項變更，且各變更在 RequiresRecreation 欄位中的值皆不相同，CloudFormation 即會透過最具侵入性的行為來更新資源。換而言之，即使多項變更中僅有其中一項需要替換，CloudFormation 仍必須取代整個資源；因此，請將 Replacement 欄位設為 True。

系統會變更每個 Purpose、InstanceType 和 KeyPairName 參數的值，藉此產生下列變更集，而這些參數皆可供 EC2 執行個體使用。一旦執行這些變更，Replacement 欄位將等於 True，所以 CloudFormation 需要替換執行個體。

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-multiple",
  "Parameters": [
    {
```

```

        "ParameterValue": "production",
        "ParameterKey": "Purpose"
    },
    {
        "ParameterValue": "MyNewKeyName",
        "ParameterKey": "KeyPairName"
    },
    {
        "ParameterValue": "t2.small",
        "ParameterKey": "InstanceType"
    }
],
"Changes": [
    {
        "ResourceChange": {
            "ResourceType": "AWS::EC2::Instance",
            "PhysicalResourceId": "i-7bef86f8",
            "Details": [
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Properties",
                        "Name": "KeyName",
                        "RequiresRecreation": "Always"
                    }
                },
                {
                    "ChangeSource": "DirectModification",
                    "Evaluation": "Dynamic",
                    "Target": {
                        "Attribute": "Properties",
                        "Name": "InstanceType",
                        "RequiresRecreation": "Conditionally"
                    }
                }
            ],
            {
                "ChangeSource": "DirectModification",
                "Evaluation": "Dynamic",
                "Target": {
                    "Attribute": "Tags",
                    "RequiresRecreation": "Never"
                }
            }
        },

```



```

        {
            "CausingEntity": "KeyPairName",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Properties",
                "Name": "KeyName",
                "RequiresRecreation": "Always"
            }
        },
        {
            "CausingEntity": "InstanceType",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Properties",
                "Name": "InstanceType",
                "RequiresRecreation": "Conditionally"
            }
        },
        {
            "CausingEntity": "Purpose",
            "ChangeSource": "ParameterReference",
            "Evaluation": "Static",
            "Target": {
                "Attribute": "Tags",
                "RequiresRecreation": "Never"
            }
        }
    ],
    "Action": "Modify",
    "Scope": [
        "Tags",
        "Properties"
    ],
    "LogicalResourceId": "MyEC2Instance",
    "Replacement": "True"
},
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T00:39:35.974Z",
"Capabilities": [],
"StackName": "MyStack",

```

```

    "NotificationARNs": [],
    "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-multiple/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

您可以檢視各項變更 (意即為 Details 結構中的 Static 評估)，藉此找出需替換資源的變更。在本範例中，儘管每項變更在 RequireRecreation 欄位中的值皆不相同，但一律要執行重新建立操作，因為系統在變更 KeyName 屬性時會採取最具侵入性的更新行為。由於金鑰名稱有所改變，CloudFormation 將替換執行個體。

若金鑰名稱保持不變，則系統變更 InstanceType 屬性時會採取最具侵入性的更新行為 (Conditionally)；如此一來，Replacement 欄位的值將為 Conditionally。若要尋找 CloudFormation 替換執行個體的條件，請檢視 [AWS::EC2::Instance](#) 資源高達的 InstanceType 屬性的更新行為。

新增與移除資源

系統將提交修改後的範本，進而產生以下範例；該範本會移除 EC2 執行個體並新增 Auto Scaling 群組與啟動組態。

```

{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
MyStack/1a2345b6-0000-00a0-a123-00abc0abc000",
  "Status": "CREATE_COMPLETE",
  "ChangeSetName": "SampleChangeSet-addremove",
  "Parameters": [
    {
      "ParameterValue": "testing",
      "ParameterKey": "Purpose"
    },
    {
      "ParameterValue": "MyKeyName",
      "ParameterKey": "KeyPairName"
    },
    {
      "ParameterValue": "t2.micro",
      "ParameterKey": "InstanceType"
    }
  ],
  "Changes": [
    {
      "ResourceChange": {

```

```

        "Action": "Add",
        "ResourceType": "AWS::AutoScaling::AutoScalingGroup",
        "Scope": [],
        "Details": [],
        "LogicalResourceId": "AutoScalingGroup"
    },
    "Type": "Resource"
},
{
    "ResourceChange": {
        "Action": "Add",
        "ResourceType": "AWS::AutoScaling::LaunchConfiguration",
        "Scope": [],
        "Details": [],
        "LogicalResourceId": "LaunchConfig"
    },
    "Type": "Resource"
},
{
    "ResourceChange": {
        "ResourceType": "AWS::EC2::Instance",
        "PhysicalResourceId": "i-1abc23d4",
        "Details": [],
        "Action": "Remove",
        "Scope": [],
        "LogicalResourceId": "MyEC2Instance"
    },
    "Type": "Resource"
}
],
"CreationTime": "2020-11-18T01:44:08.444Z",
"Capabilities": [],
"StackName": "MyStack",
"NotificationARNs": [],
"ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet-addremove/1a2345b6-0000-00a0-a123-00abc0abc000"
}

```

Changes 結構中有三種 ResourceChange 結構，每個資源各有一種結構。每個資源的 Action 欄位均會指示 CloudFormation 應新增或移除資源。Scope 和 Details 欄位則沒有內容，因為這些欄位僅適用於修改後的資源。

使用新資源時，您必須執行變更集，否則 CloudFormation 將無法判斷部分欄位的值。例如，由於 Auto Scaling 群組與啟動組態尚不存在，CloudFormation 無法提供這兩者的實體 ID。當您執行變更集時，CloudFormation 即會建立新資源。

檢視屬性層級變更

下列範例顯示 Amazon EC2 執行個體屬性的 Tag 屬性層級變更。標籤 Value 和 Key 將變更為 Test。

```
"ChangeSetName": "SampleChangeSet",
  "ChangeSetId": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/
SampleChangeSet/38d91d27-798d-4736-9bf1-fb7c46207807",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
SampleEc2Template/68edcdc0-f6b6-11ee-966c-126d572cdd11",
  "StackName": "SampleEc2Template",
  "Description": "A sample EC2 instance template for testing change sets.",
  "Parameters": [
    {
      "ParameterKey": "KeyPairName",
      "ParameterValue": "BatchTest"
    },
    {
      "ParameterKey": "Purpose",
      "ParameterValue": "testing"
    },
    {
      "ParameterKey": "InstanceType",
      "ParameterValue": "t2.micro"
    }
  ],
  "CreationTime": "2024-04-09T21:29:10.759000+00:00",
  "ExecutionStatus": "AVAILABLE",
  "Status": "CREATE_COMPLETE",
  "StatusReason": null,
  "NotificationARNs": [],
  "RollbackConfiguration": {
    :...skipping...
  },
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
```

```

        "LogicalResourceId": "MyEC2Instance",
        "PhysicalResourceId": "i-0cc7856a36315e62b",
        "ResourceType": "AWS::EC2::Instance",
        "Replacement": "False",
        "Scope": [
            "Tags"
        ],
        "Details": [
            {
                "Target": {
                    "Attribute": "Tags",
                    "RequiresRecreation": "Never",
                    "Path": "/Properties/Tags/0/Value",
                    "BeforeValue": "testing",
                    "AfterValue": "Test",
                    "AttributeChangeType": "Modify"
                },
                "Evaluation": "Static",
                "ChangeSource": "DirectModification"
            },
            {
                "Target": {
                    "Attribute": "Tags",
                    "RequiresRecreation": "Never",
                    "Path": "/Properties/Tags/0/Key",
                    "BeforeValue": "Purpose",
                    "AfterValue": "Test",
                    "AttributeChangeType": "Modify"
                },
                "Evaluation": "Static",
                "ChangeSource": "DirectModification"
            }
        ],
        "BeforeContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"testing\",\"Key\":\"Purpose\"}]}}",
        "AfterContext": "{\"Properties\":{\"KeyName\":\"BatchTest\",\"ImageId\":\"ami-8fcee4e5\",\"InstanceType\":\"t2.micro\",\"Tags\":[{\"Value\":\"Test\",\"Key\":\"Test\"}]}}"
    }
}
]

```

此 Details 結構會顯示執行變更集前 Key 和 Value 的值，以及執行變更集後其預期數值。

巢狀堆疊變更集

透過巢狀堆疊變更集，您可以預覽整個巢狀堆疊階層中對應用程式和基礎設施資源所做的變更，並在確認所有變更都如預期時繼續進行更新。

如需有關巢狀堆疊變更集的詳細資訊，請參閱下列章節：

主題

- [變更集和巢狀堆疊的概觀](#)
- [使用巢狀堆疊的變更集 \(主控台\)](#)
- [使用巢狀堆疊的變更集 \(AWS CLI\)](#)

變更集和巢狀堆疊的概觀

巢狀堆疊的變更集合將下列功能結合在一起，以將預覽變更的範圍擴展至整個堆疊階層：

- 變更集是一種 CloudFormation 功能，可預覽對堆疊的提議變更將如何影響現有或新建立的資源。建立變更集時，透過將堆疊與您提交之資源的變更相比較，CloudFormation 會提供提議變更的清單。如需變更集的詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。
- 巢狀堆疊是建立為另一個堆疊一部分的堆疊。例如，您可能在一個巢狀堆疊中擁有聯網和安全性相關資源，而在另一個巢狀堆疊中擁有應用程式資源。透過這種方式來分割應用程式模型，有助於提升程式碼的可維護性和重複使用性。如需巢狀堆疊的詳細資訊，請參閱[運用巢狀堆疊，將範本分割成可重複使用的部分](#)。

使用巢狀堆疊的變更集 (主控台)

- 建立變更集 – 透過從堆疊階層的任何層級提交變更來建立變更集。您可以提交修改後的堆疊範本或修改後的輸入參數值，而 CloudFormation 會將巢狀堆疊與提交的變更進行比較，以產生變更集。在 CloudFormation 主控台中預設會啟用巢狀堆疊的變更集。如需詳細資訊，請參閱[建立 CloudFormation 堆疊的變更集](#)。

▼ Change sets

Change sets for nested stacks

Creates change sets for all nested stacks specified in your template. [Learn more](#) 

☒ **Enabled**

Creates a change set for all nested stacks specified in the template.

☐ **Disabled**

Create a change set only on the target stack.

Note

根變更集是與建立整個變更集階層的堆疊相關聯的變更集。您必須從根變更集中執行或刪除巢狀堆疊的變更集。如需詳細資訊，請參閱[對巢狀堆疊執行堆疊操作](#)。

- 檢視變更集 – 在執行巢狀堆疊之前，以視覺方式呈現對巢狀堆疊內資源的變更。您可以在變更集的 Changes (變更) 區段中檢視提議的變更，方法是瀏覽目前的堆疊及其巢狀變更集。如需詳細資訊，請參閱[檢視 CloudFormation 堆疊的變更集](#)。
- 執行變更集 – 執行變更集中所述的變更，這些變更都與目前的堆疊及其子代相關。必須透過根變更集進行執行操作。如需詳細資訊，請參閱[執行 CloudFormation 堆疊的變更集](#)。
- 刪除變更集 – 將變更集從目前堆疊移除。刪除變更集可防止自己或其他使用者意外起始不應套用的變更集。必須透過根變更集進行刪除操作。如需詳細資訊，請參閱[刪除 CloudFormation 堆疊的變更集](#)。

使用巢狀堆疊的變更集 (AWS CLI)

create-change-set

- [create-change-set](#) – 根據預設，不會為 AWS CLI 啟用巢狀堆疊的變更集。若要為整個堆疊階層建立變更集，請指定 `--include-nested-stacks` 選項。如需詳細資訊，請參閱[建立 CloudFormation 堆疊的變更集](#)。

下列 AWS CLI 範例會為指定的根堆疊建立變更集。

```
aws cloudformation create-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM \  
  --include-nested-stacks
```

以下為範例輸出。

```
{
  "Id": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:Stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

描述變更集

- [describe-change-set](#) – 傳回如果您執行變更集，CloudFormation 將進行的變更清單。如果指定的變更集包含屬於巢狀堆疊的子變更集，則 ChangeSetId 會傳回與該變更集相關的資訊。如需詳細資訊，請參閱[檢視 CloudFormation 堆疊的變更集](#)。

下列 AWS CLI 範例說明指定根堆疊的變更集。

```
aws cloudformation describe-change-set \
  --change-set-name my-root-stack-change-set \
  --stack-name my-root-stack
```

以下為範例輸出。

```
{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "ChildStack",
        "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
        "ResourceType": "AWS::CloudFormation::Stack",
        "Replacement": "False",
        "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
        "Scope": [
          "Properties"
        ],
        "Details": [
```



```

        {
            "Target": {
                "Attribute": "Properties",
                "RequiresRecreation": "Never"
            },
            "Evaluation": "Dynamic",
            "ChangeSource": "Automatic"
        }
    ]
}

},
"ChangeSetName": "my-root-stack-change-set",
"ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
"StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-root-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
"StackName": "my-root-stack",
"IncludeNestedStacks": true,
"ParentChangeSetId": null,
"RootChangeSetId": null,
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "AVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": null,
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
    "CAPABILITY_IAM"
],
"Tags": null
}

```

下列 AWS CLI 範例說明指定巢狀堆疊的變更集。

```

aws cloudformation describe-change-set \
  --change-set-name my-nested-stack-change-set \

```

```
--stack-name my-nested-stack
```

以下為範例輸出。

```
{
  "Changes": [
    {
      "Type": "Resource",
      "ResourceChange": {
        "Action": "Modify",
        "LogicalResourceId": "function",
        "PhysicalResourceId": "my-function",
        "ResourceType": "AWS::Lambda::Function",
        "Replacement": "False",
        "ChangeSetId": null,
        "Scope": [
          "Properties"
        ],
        "Details": [
          {
            "Target": {
              "Attribute": "Properties",
              "Name": "Timeout",
              "RequiresRecreation": "Never"
            },
            "Evaluation": "Static",
            "ChangeSource": "DirectModification"
          }
        ]
      }
    }
  ],
  "ChangeSetName": "my-nested-stack-change-set",
  "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-nested-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-nested-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99205",
  "ParentChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "RootChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-root-stack-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0",
  "IncludeNestedStacks": true,
  "StackName": "my-nested-stack",
```

```
"Description": null,
"Parameters": null,
"CreationTime": "2020-11-18T05:20:56.651Z",
"ExecutionStatus": "UNAVAILABLE",
"Status": "CREATE_COMPLETE",
"StatusReason": "Executable from root change set",
"NotificationARNs": [

],
"RollbackConfiguration": {

},
"Capabilities": [
    "CAPABILITY_IAM"
],
"Tags": null
}
```

執行變更集

- [execute-change-set](#) – 使用在建立特定變更集時所提供的輸入資訊，建立或更新堆疊。若要為整個堆疊階層建立變更集，您必須在 `create-change-set` 操作期間指定 `--include-nested-stacks` 選項。如需詳細資訊，請參閱[執行 CloudFormation 堆疊的變更集](#)。

Note

`execute-change-set` 的執行必須透過根變更集進行，並將變更集套用至整個堆疊階層。

下列 AWS CLI 範例會針對指定的根堆疊執行變更集。

```
aws cloudformation execute-change-set \
  --stack-name my-root-stack \
  --change-set-name my-root-stack-change-set
```

刪除變更集

- [delete-change-set](#) – 刪除指定的變更集。刪除變更集可確保不會使用到錯誤的變更集。對於使用 `--include-nested-stacks` 選項建立的變更集，變更集的刪除不是同時發生的。如需詳細資訊，請參閱[刪除 CloudFormation 堆疊的變更集](#)。

 Note

delete-change-set 的執行必須透過根變更集進行，並將會刪除整個變更集的階層。如果在 create-change-set 操作期間建立巢狀堆疊，也會刪除狀態為 REVIEW_IN_PROGRESS 的巢狀堆疊。

下列 AWS CLI 範例會刪除指定根堆疊的變更集。

```
aws cloudformation delete-change-set \  
  --stack-name my-root-stack \  
  --change-set-name my-root-stack-change-set
```

驗證堆疊部署

透過部署前驗證，您可以在執行 CloudFormation 變更集之前識別和解決潛在的部署問題。此功能會根據常見的失敗案例驗證您的範本，協助您在開發週期的早期發現問題。

主題

- [部署前驗證的運作方式](#)
- [考量事項](#)
- [先決條件](#)
- [驗證堆疊部署（主控台）](#)
- [驗證堆疊部署 \(AWS CLI\)](#)
- [驗證類型](#)
- [資源限制](#)

部署前驗證的運作方式

部署前驗證涉及下列階段：

1. 建立變更集 – 像平常一樣產生 CloudFormation 堆疊更新的變更集。建立變更集時，預設會啟用部署前驗證。

2. 驗證執行 – CloudFormation 會針對您的範本和目標環境執行多個驗證檢查。目前支援 3 種類型的驗證：針對資源結構描述的屬性語法驗證、與現有資源的資源名稱衝突偵測，以及用於刪除操作的 S3 儲存貯體清空驗證。
3. 檢閱驗證結果 – CloudFormation 會針對發現的任何問題提供詳細意見回饋，包括精確找出範本中問題位置的路徑、消除手動範本偵錯。
4. 解決問題 – 在繼續部署之前，透過更新您的範本或解決衝突來解決已識別的問題。
5. 自信地執行 - 在知道常見失敗案例已預先驗證的情況下部署您的變更集。

考量事項

當您使用部署前驗證時，請記住下列事項：

- 部署前驗證著重於三種常見的部署失敗案例。它不保證您的部署會成功，但可以降低常見故障的可能性。
- 驗證模式的行為不同：
 - 當驗證偵測到錯誤時，失敗模式可防止變更集執行，確保有問題的範本無法繼續部署。這適用於屬性語法錯誤和資源命名衝突。
 - WARN 模式允許在驗證失敗時成功建立變更集，提供警告，讓開發人員可以在執行之前檢閱和解決此問題。這適用於限制條件違規，例如可透過手動介入解決的 S3 儲存貯體清空。
- 驗證結果會繫結至特定變更集。如果您修改範本，則需要建立新的變更集，以取得更新的驗證結果。
- S3 儲存貯體驗證只會檢查物件是否存在，而不是儲存貯體政策或其他可能防止刪除的限制。

先決條件

若要使用部署前驗證，您必須具有：

- 在帳戶中建立變更集和讀取資源所需的 IAM 許可。對於 S3 儲存貯體清空檢查，您需要 `s3:ListBucket` 許可。
- 存取 AWS 區域 部署堆疊的。
- 您要在部署之前驗證的 CloudFormation 範本。

驗證堆疊部署（主控台）

使用下列程序，使用 主控台驗證您的堆疊部署。

在部署之前驗證範本

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
3. 在堆疊頁面上，選擇您要建立變更集的執行中堆疊。
4. 在堆疊詳細資訊窗格中，選擇更新堆疊，然後選擇建立變更集。
5. 在建立 ##### 的變更集頁面上，上傳已更新的範本或指定範本來源。
6. 選擇下一步，繼續進行剩餘的變更集組態步驟。
7. 如果範本包括 IAM 資源，請在功能中選擇我確認 CloudFormation 可能會建立 IAM 資源。IAM 資源可修改 AWS 帳戶中的許可；請檢閱這些資源，確保系統僅能執行所需的動作。如需詳細資訊，請參閱 [CloudFormation 範本中的確認 IAM 資源](#)。
8. 在檢閱頁面上，選擇建立變更集。
9. CloudFormation 將建立變更集並執行驗證檢查。在部署驗證索引標籤中檢閱驗證結果。
10. 如果驗證通過或您對警告感到滿意，請選擇執行變更集來部署變更。
11. 如果驗證失敗，請修正問題並建立新的變更集，以重新驗證您的部署。

驗證堆疊部署 (AWS CLI)

部署前驗證的 AWS CLI 命令包括：

- [create-change-set](#) 會在建立變更集期間自動驗證。
- [describe-change-set](#) 以驗證變更集狀態
- [describe-events](#) 來檢閱驗證結果。

使用下列程序來驗證使用的堆疊部署 AWS CLI。

在部署之前驗證範本

1. 使用 [create-change-set](#) 命令：

```
aws cloudformation create-change-set \  
  --stack-name MyStack \  
  --change-set-name MyChangeSet \  
  --change-set-type "CREATE" \  
  --template-url https://s3.amazonaws.com/...
```

```
--template-body file://updated-template.yaml
```

命令會同時傳回變更集 ARN 和堆疊 ARN。

2. 使用 [describe-events](#) 命令搭配變更集 ARN 或變更集名稱，以檢閱驗證狀態和結果。

```
aws cloudformation describe-events \  
  --change-set-name "arn:aws:cloudformation:us-east-1:123456789012:changeSet/MyChangeSet/94498df5-1afb-43b1-9869-9f82b2d877ac"
```

驗證錯誤的範例輸出：

```
{  
  "OperationEvents": [  
    {  
      "EventId": "9b5c9a29-4704-4ad0-8082-afb49418d55b",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",  
      "OperationStatus": "FAILED",  
      "EventType": "STACK_EVENT",  
      "Timestamp": "2025-10-27T17:10:02.923Z",  
      "StartTime": "2025-10-27T17:09:57.537Z",  
      "EndTime": "2025-10-27T17:10:02.923Z"  
    },  
    {  
      "EventId": "2d8c3262-3468-4283-82fb-6e780e9e4f1d",  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/  
c3908380-b357-11f0-a97f-0ad08f35df65",  
      "OperationId": "f558b823-e1e3-4de3-a222-e6b930ddcad4",  
      "OperationType": "CREATE_CHANGESET",  
      "EventType": "VALIDATION_ERROR",  
      "LogicalResourceId": "NotificationBucket",  
      "PhysicalResourceId": "",  
      "ResourceType": "AWS::S3::Bucket",  
      "Timestamp": "2025-10-27T17:10:02.461Z",  
      "ValidationFailureMode": "FAIL",  
      "ValidationName": "PROPERTY_VALIDATION",  
      "ValidationStatus": "FAILED",  
      "ValidationStatusReason": "#/NotificationConfiguration/  
QueueConfigurations/0: required key [Event] not found",  
    }  
  ]  
}
```

```
        "ValidationPath": "/Resources/NotificationBucket/Properties/NotificationConfiguration/QueueConfigurations/0"
      }
    ]
  }
}
```

3. 更新範本以解決任何驗證錯誤，然後建立新的變更集。
4. 驗證通過後，請執行變更集：

```
aws cloudformation execute-change-set \  
  --change-set-name MyChangeSet \  
  --stack-name MyStack
```

驗證類型

部署前驗證包括下列類型的檢查：

- 屬性語法驗證 – 根據資源結構描述驗證 AWS 資源屬性。它會檢查所需的屬性和有效的屬性值，並識別已棄用或不支援的屬性組合。
- 資源名稱衝突偵測 – 檢查命名是否與現有 AWS 資源衝突。它會驗證資源名稱是否符合 AWS 命名要求，並在部署嘗試之前識別潛在的衝突。
- S3 儲存貯體清空驗證 – 嘗試刪除包含物件的 S3 儲存貯體時發出警告。它提供物件計數，以協助評估刪除影響，並協助防止常見的 S3 刪除失敗。

每個驗證類型都會在範本中提供特定的錯誤訊息和錯誤位置，以協助您快速解決問題。

資源限制

部署前驗證不支援下列資源類型：

- `AWS::ApiGatewayV2::ApiGatewayManagedOverrides`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AppMesh::GatewayRoute`
- `AWS::AppMesh::Mesh`
- `AWS::AppMesh::Route`
- `AWS::AppMesh::VirtualGateway`

- `AWS::AppMesh::VirtualNode`
- `AWS::AppMesh::VirtualRouter`
- `AWS::AppMesh::VirtualService`
- `AWS::AppStream::Fleet`
- `AWS::AppStream::Stack`
- `AWS::AppStream::StackFleetAssociation`
- `AWS::AppStream::StackUserAssociation`
- `AWS::AppStream::User`
- `AWS::AppSync::ApiCache`
- `AWS::AppSync::ApiKey`
- `AWS::AppSync::GraphQLSchema`
- `AWS::AutoScalingPlans::ScalingPlan`
- `AWS::Budgets::Budget`
- `AWS::CertificateManager::Certificate`
- `AWS::Cloud9::EnvironmentEC2`
- `AWS::CloudFormation::CustomResource`
- `AWS::CloudFormation::Macro`
- `AWS::CloudFormation::WaitCondition`
- `AWS::CloudFormation::WaitConditionHandle`
- `AWS::CloudFront::StreamingDistribution`
- `AWS::CloudWatch::AnomalyDetector`
- `AWS::CloudWatch::InsightRule`
- `AWS::CodeBuild::Project`
- `AWS::CodeBuild::ReportGroup`
- `AWS::CodeBuild::SourceCredential`
- `AWS::CodeCommit::Repository`
- `AWS::CodeDeploy::DeploymentGroup`
- `AWS::CodeStar::GitHubRepository`

- `AWS::Config::ConfigurationRecorder`
- `AWS::Config::DeliveryChannel`
- `AWS::Config::OrganizationConfigRule`
- `AWS::Config::RemediationConfiguration`
- `AWS::DAX::Cluster`
- `AWS::DAX::ParameterGroup`
- `AWS::DAX::SubnetGroup`
- `AWS::DirectoryService::MicrosoftAD`
- `AWS::DLM::LifecyclePolicy`
- `AWS::DMS::Certificate`
- `AWS::DMS::Endpoint`
- `AWS::DMS::EventSubscription`
- `AWS::DMS::ReplicationInstance`
- `AWS::DMS::ReplicationSubnetGroup`
- `AWS::DMS::ReplicationTask`
- `AWS::DocDB::DBCluster`
- `AWS::DocDB::DBClusterParameterGroup`
- `AWS::DocDB::DBInstance`
- `AWS::DocDB::DBSubnetGroup`
- `AWS::DocDB::EventSubscription`
- `AWS::EC2::ClientVpnAuthorizationRule`
- `AWS::EC2::ClientVpnEndpoint`
- `AWS::EC2::ClientVpnRoute`
- `AWS::EC2::ClientVpnTargetNetworkAssociation`
- `AWS::EC2::NetworkInterfacePermission`
- `AWS::ElastiCache::CacheCluster`
- `AWS::ElastiCache::ReplicationGroup`
- `AWS::ElastiCache::SecurityGroup`

- `AWS::ElasticCache::SecurityGroupIngress`
- `AWS::ElasticLoadBalancing::LoadBalancer`
- `AWS::ElasticLoadBalancingV2::ListenerCertificate`
- `AWS::Elasticsearch::Domain`
- `AWS::EMR::Cluster`
- `AWS::EMR::InstanceFleetConfig`
- `AWS::EMR::InstanceGroupConfig`
- `AWS::FSx::FileSystem`
- `AWS::FSx::Snapshot`
- `AWS::FSx::StorageVirtualMachine`
- `AWS::FSx::Volume`
- `AWS::Glue::Classifier`
- `AWS::Glue::Connection`
- `AWS::Glue::CustomEntityType`
- `AWS::Glue::DataCatalogEncryptionSettings`
- `AWS::Glue::DataQualityRuleset`
- `AWS::Glue::DevEndpoint`
- `AWS::Glue::MLTransform`
- `AWS::Glue::Partition`
- `AWS::Glue::SecurityConfiguration`
- `AWS::Glue::Table`
- `AWS::Glue::TableOptimizer`
- `AWS::Glue::Workflow`
- `AWS::Greengrass::ConnectorDefinition`
- `AWS::Greengrass::ConnectorDefinitionVersion`
- `AWS::Greengrass::CoreDefinition`
- `AWS::Greengrass::CoreDefinitionVersion`
- `AWS::Greengrass::DeviceDefinition`

- `AWS::Greengrass::DeviceDefinitionVersion`
- `AWS::Greengrass::FunctionDefinition`
- `AWS::Greengrass::FunctionDefinitionVersion`
- `AWS::Greengrass::Group`
- `AWS::Greengrass::GroupVersion`
- `AWS::Greengrass::LoggerDefinition`
- `AWS::Greengrass::LoggerDefinitionVersion`
- `AWS::Greengrass::ResourceDefinition`
- `AWS::Greengrass::ResourceDefinitionVersion`
- `AWS::Greengrass::SubscriptionDefinition`
- `AWS::Greengrass::SubscriptionDefinitionVersion`
- `AWS::IAM::AccessKey`
- `AWS::IAM::UserToGroupAddition`
- `AWS::IoT::PolicyPrincipalAttachment`
- `AWS::IoT::ThingPrincipalAttachment`
- `AWS::IoTThingsGraph::FlowTemplate`
- `AWS::KinesisAnalytics::Application`
- `AWS::KinesisAnalytics::ApplicationOutput`
- `AWS::KinesisAnalytics::ApplicationReferenceDataSource`
- `AWS::KinesisAnalyticsV2::ApplicationCloudWatchLoggingOption`
- `AWS::KinesisAnalyticsV2::ApplicationOutput`
- `AWS::KinesisAnalyticsV2::ApplicationReferenceDataSource`
- `AWS::LakeFormation::DataLakeSettings`
- `AWS::LakeFormation::Permissions`
- `AWS::LakeFormation::Resource`
- `AWS::ManagedBlockchain::Member`
- `AWS::ManagedBlockchain::Node`
- `AWS::MediaConvert::JobTemplate`

- `AWS::MediaConvert::Preset`
- `AWS::MediaConvert::Queue`
- `AWS::MediaLive::Channel`
- `AWS::MediaLive::Input`
- `AWS::MediaLive::InputSecurityGroup`
- `AWS::MediaStore::Container`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::Pinpoint::ADMChannel`
- `AWS::Pinpoint::APNSChannel`
- `AWS::Pinpoint::APNSSandboxChannel`
- `AWS::Pinpoint::APNSVoipChannel`
- `AWS::Pinpoint::APNSVoipSandboxChannel`
- `AWS::Pinpoint::App`
- `AWS::Pinpoint::ApplicationSettings`
- `AWS::Pinpoint::BaiduChannel`
- `AWS::Pinpoint::Campaign`
- `AWS::Pinpoint::EmailChannel`
- `AWS::Pinpoint::EmailTemplate`
- `AWS::Pinpoint::EventStream`
- `AWS::Pinpoint::GCMChannel`
- `AWS::Pinpoint::PushTemplate`
- `AWS::Pinpoint::Segment`

- `AWS::Pinpoint::SMSChannel`
- `AWS::Pinpoint::SmsTemplate`
- `AWS::Pinpoint::VoiceChannel`
- `AWS::PinpointEmail::ConfigurationSet`
- `AWS::PinpointEmail::ConfigurationSetEventDestination`
- `AWS::PinpointEmail::DedicatedIpPool`
- `AWS::PinpointEmail::Identity`
- `AWS::QLDB::Ledger`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SageMaker::CodeRepository`
- `AWS::SageMaker::EndpointConfig`
- `AWS::SageMaker::Model`
- `AWS::SageMaker::NotebookInstance`
- `AWS::SageMaker::NotebookInstanceLifecycleConfig`
- `AWS::SageMaker::Workteam`
- `AWS::SDB::Domain`
- `AWS::ServiceCatalog::AcceptedPortfolioShare`
- `AWS::ServiceCatalog::LaunchRoleConstraint`
- `AWS::ServiceCatalog::Portfolio`
- `AWS::ServiceCatalog::StackSetConstraint`
- `AWS::ServiceDiscovery::HttpNamespace`
- `AWS::ServiceDiscovery::Instance`
- `AWS::ServiceDiscovery::PrivateDnsNamespace`

- `AWS::ServiceDiscovery::PublicDnsNamespace`
- `AWS::ServiceDiscovery::Service`
- `AWS::SES::ReceiptFilter`
- `AWS::SES::ReceiptRule`
- `AWS::SES::ReceiptRuleSet`
- `AWS::SSM::MaintenanceWindow`
- `AWS::SSM::MaintenanceWindowTarget`
- `AWS::SSM::MaintenanceWindowTask`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFRegional::ByteMatchSet`
- `AWS::WAFRegional::GeoMatchSet`
- `AWS::WAFRegional::IPSet`
- `AWS::WAFRegional::RateBasedRule`
- `AWS::WAFRegional::RegexPatternSet`
- `AWS::WAFRegional::Rule`
- `AWS::WAFRegional::SizeConstraintSet`
- `AWS::WAFRegional::SqlInjectionMatchSet`
- `AWS::WAFRegional::WebACL`
- `AWS::WAFRegional::WebACLAssociation`
- `AWS::WAFRegional::XssMatchSet`
- `AWS::WorkSpaces::Workspace`
- `AWS::AmazonMQ::ConfigurationAssociation`

- `AWS::ApiGateway::DomainNameAccessAssociation`
- `AWS::AppConfig::ExtensionAssociation`
- `AWS::AppStream::ApplicationEntitlementAssociation`
- `AWS::AppStream::ApplicationFleetAssociation`
- `AWS::AppSync::DomainNameApiAssociation`
- `AWS::AppSync::SourceApiAssociation`
- `AWS::CleanRooms::ConfiguredTableAssociation`
- `AWS::CleanRooms::IdNamespaceAssociation`
- `AWS::CodeGuruReviewer::RepositoryAssociation`
- `AWS::Cognito::IdentityPoolRoleAttachment`
- `AWS::Cognito::UserPoolRiskConfigurationAttachment`
- `AWS::Cognito::UserPoolUICustomizationAttachment`
- `AWS::Cognito::UserPoolUserToGroupAttachment`
- `AWS::Connect::IntegrationAssociation`
- `AWS::Deadline::QueueFleetAssociation`
- `AWS::Deadline::QueueLimitAssociation`
- `AWS::EC2::EIPAssociation`
- `AWS::EC2::EnclaveCertificateIamRoleAssociation`
- `AWS::EC2::GatewayRouteTableAssociation`
- `AWS::EC2::IPAMResourceDiscoveryAssociation`
- `AWS::EC2::IpPoolRouteTableAssociation`
- `AWS::EC2::LocalGatewayRouteTableVPCAssociation`
- `AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation`
- `AWS::EC2::NetworkInterfaceAttachment`
- `AWS::EC2::RouteServerAssociation`
- `AWS::EC2::SecurityGroupVpcAssociation`
- `AWS::EC2::SubnetNetworkAclAssociation`
- `AWS::EC2::SubnetRouteTableAssociation`

- `AWS::EC2::TransitGatewayAttachment`
- `AWS::EC2::TransitGatewayMulticastDomainAssociation`
- `AWS::EC2::TransitGatewayPeeringAttachment`
- `AWS::EC2::TransitGatewayRouteTableAssociation`
- `AWS::EC2::TransitGatewayVpcAttachment`
- `AWS::EC2::VPCDHCPOptionsAssociation`
- `AWS::EC2::VPCGatewayAttachment`
- `AWS::EC2::VolumeAttachment`
- `AWS::ECS::ClusterCapacityProviderAssociations`
- `AWS::EKS::PodIdentityAssociation`
- `AWS::FSx::DataRepositoryAssociation`
- `AWS::FSx::S3AccessPointAttachment`
- `AWS::GlobalAccelerator::CrossAccountAttachment`
- `AWS::LakeFormation::TagAssociation`
- `AWS::NetworkFirewall::VpcEndpointAssociation`
- `AWS::NetworkManager::ConnectAttachment`
- `AWS::NetworkManager::CustomerGatewayAssociation`
- `AWS::NetworkManager::DirectConnectGatewayAttachment`
- `AWS::NetworkManager::LinkAssociation`
- `AWS::NetworkManager::SiteToSiteVpnAttachment`
- `AWS::NetworkManager::TransitGatewayRouteTableAttachment`
- `AWS::NetworkManager::VpcAttachment`
- `AWS::Notifications::ChannelAssociation`
- `AWS::Notifications::ManagedNotificationAccountContactAssociation`
- `AWS::Notifications::ManagedNotificationAdditionalChannelAssociation`
- `AWS::Notifications::OrganizationalUnitAssociation`
- `AWS::ResourceExplorer2::DefaultViewAssociation`
- `AWS::Route53Profiles::ProfileAssociation`

- `AWS::Route53Profiles::ProfileResourceAssociation`
- `AWS::Route53Resolver::FirewallRuleGroupAssociation`
- `AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation`
- `AWS::Route53Resolver::ResolverRuleAssociation`
- `AWS::SSM::Association`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecurityHub::PolicyAssociation`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::ServiceActionAssociation`
- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkResourceAssociation`
- `AWS::VpcLattice::ServiceNetworkServiceAssociation`
- `AWS::VpcLattice::ServiceNetworkVpcAssociation`
- `AWS::WAFv2::WebACLAssociation`
- `AWS::Wisdom::AssistantAssociation`
- `AWS::WorkspacesInstances::VolumeAssociation`
- `AWS::IAM::Policy`
- `AWS::SNS::TopicPolicy`
- `AWS::SQS::QueuePolicy`
- `AWS::EC2::NetworkAclEntry`
- `AWS::EC2::VPNGatewayRoutePropagation`
- `AWS::CloudFormation::Stack`
- `AWS::CloudWatch::MetricStream`
- `AWS::WorkSpaces::ConnectionAlias`
- `AWS::IoT::ProvisioningTemplate`

- `AWS::MediaPackage::Channel`
- `AWS::CloudFront::OriginRequestPolicy`
- `AWS::Route53Resolver::ResolverQueryLoggingConfig`
- `AWS::NetworkManager::TransitGatewayRegistration`
- `AWS::ImageBuilder::Image`
- `AWS::Config::ConformancePack`
- `AWS::S3::AccessPoint`
- `AWS::CodeStarConnections::Connection`
- `AWS::CloudFront::CachePolicy`
- `AWS::FMS::NotificationChannel`
- `AWS::ImageBuilder::InfrastructureConfiguration`
- `AWS::Detective::Graph`
- `AWS::EC2::CarrierGateway`
- `AWS::CloudWatch::CompositeAlarm`
- `AWS::CodeArtifact::Repository`
- `AWS::GroundStation::DataflowEndpointGroup`
- `AWS::ElasticLoadBalancingV2::Listener`
- `AWS::ImageBuilder::ImageRecipe`
- `AWS::NetworkManager::Device`
- `AWS::Kendra::DataSource`
- `AWS::Timestream::Database`
- `AWS::CodeGuruProfiler::ProfilingGroup`
- `AWS::Lambda::EventSourceMapping`
- `AWS::ECR::Repository`
- `AWS::WAFv2::IPSet`
- `AWS::GameLift::Alias`
- `AWS::IoTSiteWise::Asset`
- `AWS::OpsWorksCM::Server`

- `AWS::IoT::Authorizer`
- `AWS::WAFv2::RuleGroup`
- `AWS::NetworkManager::Site`
- `AWS::ResourceGroups::Group`
- `AWS::MediaPackage::PackagingConfiguration`
- `AWS::ImageBuilder::ImagePipeline`
- `AWS::ECS::TaskDefinition`
- `AWS::Macie::CustomDataIdentifier`
- `AWS::MediaPackage::OriginEndpoint`
- `AWS::Logs::LogGroup`
- `AWS::CodeArtifact::Domain`
- `AWS::Kendra::Faq`
- `AWS::ECS::TaskSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::ECS::Cluster`
- `AWS::SSO::Assignment`
- `AWS::GlobalAccelerator::Listener`
- `AWS::ServiceCatalog::CloudFormationProvisionedProduct`
- `AWS::RDS::DBProxy`
- `AWS::EC2::FlowLog`
- `AWS::ImageBuilder::Component`
- `AWS::CloudFront::RealtimeLogConfig`
- `AWS::NetworkManager::GlobalNetwork`
- `AWS::RDS::DBProxyTargetGroup`
- `AWS::WAFv2::WebACL`
- `AWS::IVS::StreamKey`
- `AWS::IVS::PlaybackKeyPair`
- `AWS::Macie::Session`

- `AWS::Route53::HealthCheck`
- `AWS::Synthetics::Canary`
- `AWS::Lambda::CodeSigningConfig`
- `AWS::EFS::AccessPoint`
- `AWS::Timestream::Table`
- `AWS::MediaPackage::PackagingGroup`
- `AWS::ECS::PrimaryTaskSet`
- `AWS::Config::ConfigurationAggregator`
- `AWS::GroundStation::Config`
- `AWS::IoTSiteWise::AssetModel`
- `AWS::SES::ConfigurationSet`
- `AWS::ImageBuilder::DistributionConfiguration`
- `AWS::Config::OrganizationConformancePack`
- `AWS::EC2::LocalGatewayRoute`
- `AWS::KMS::Key`
- `AWS::Detective::MemberInvitation`
- `AWS::EKS::FargateProfile`
- `AWS::MediaPackage::Asset`
- `AWS::GlobalAccelerator::EndpointGroup`
- `AWS::Macie::FindingsFilter`
- `AWS::IoT::Certificate`
- `AWS::SageMaker::MonitoringSchedule`
- `AWS::IVS::Channel`
- `AWS::Kendra::Index`
- `AWS::EventSchemas::RegistryPolicy`
- `AWS::KinesisFirehose::DeliveryStream`
- `AWS::GlobalAccelerator::Accelerator`
- `AWS::EC2::PrefixList`
- `AWS::GameLift::GameServerGroup`

- `AWS::NetworkManager::Link`
- `AWS::EFS::FileSystem`
- `AWS::Route53::HostedZone`
- `AWS::GroundStation::MissionProfile`
- `AWS::KMS::Alias`
- `AWS::FMS::Policy`
- `AWS::SSO::PermissionSet`
- `AWS::StepFunctions::StateMachine`
- `AWS::QLDB::Stream`
- `AWS::IoTSiteWise::Gateway`
- `AWS::ECS::Service`
- `AWS::ECS::CapacityProvider`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::SecurityGroupIngress`
- `AWS::EC2::SecurityGroupEgress`
- `AWS::EC2::EC2Fleet`
- `AWS::IAM::Group`
- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::ApiGateway::GatewayResponse`
- `AWS::S3::BucketPolicy`
- `AWS::SNS::Topic`
- `AWS::SNS::Subscription`
- `AWS::RDS::DBInstance`
- `AWS::RDS::DBParameterGroup`
- `AWS::RDS::DBCluster`
- `AWS::RDS::DBClusterParameterGroup`
- `AWS::RDS::DBSubnetGroup`
- `AWS::RDS::EventSubscription`

- `AWS::RDS::GlobalCluster`
- `AWS::RDS::OptionGroup`
- `AWS::Neptune::DBInstance`
- `AWS::Neptune::DBParameterGroup`
- `AWS::Neptune::DBCluster`
- `AWS::Neptune::DBClusterParameterGroup`
- `AWS::Neptune::DBSubnetGroup`
- `AWS::Redshift::Cluster`
- `AWS::Redshift::ClusterParameterGroup`
- `AWS::Redshift::ClusterSubnetGroup`
- `AWS::Redshift::EndpointAccess`
- `AWS::Redshift::EndpointAuthorization`
- `AWS::Redshift::EventSubscription`
- `AWS::Redshift::ScheduledAction`
- `AWS::ElastiCache::SubnetGroup`

直接更新堆疊

想要快速部署以更新堆疊時，請您執行直接更新。使用直接更新，您將提交或輸入更新堆疊中資源的範本或參數，CloudFormation 並會立即部署這些更新。如果您想要使用範本進行更新，可以修改目前的範本，並將其存放在本機或 Amazon S3 儲存貯體。

如果是不支援更新的資源屬性，您必須保留目前的值。若要在進行更新堆疊之前，預覽 CloudFormation 將對堆疊進行的變更，請使用變更集。如需詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

當更新堆疊時，CloudFormation 可能會中斷資源或取代更新的資源，取決於您更新的屬性。如需資源更新行為的詳細資訊，請參閱[了解更新堆疊資源的行為](#)。

更新 堆疊 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。

2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在堆疊頁面上，選取您要更新的執行中堆疊。
4. 在 stack details (堆疊詳細資訊) 窗格中，選擇 Update (更新)。
5. 如尚未修改堆疊範本，請選擇使用現有的範本，然後選擇下一步。

如已修改範本，請選取取代現有的範本，然後在指定範本區段中指定更新的範本位置：

- 如果是存放在本機電腦上的範本，請選取 Upload a template file (上傳範本檔案)。選擇 Choose File (選擇檔案) 導覽至該檔案並選取它，然後選擇 Next (下一步)。

Note

如果您上傳本機範本檔案，CloudFormation 會將其上傳至 中的 Amazon Simple Storage Service (Amazon S3) 儲存貯體 AWS 帳戶。如果您尚未擁有 CloudFormation 建立的 S3 儲存貯體，&CFN; 會針對您已上傳範本檔案至其中的每個區域建立唯一儲存貯體。如果您的 AWS 帳戶中已經擁有 CloudFormation 建立的 S3 儲存貯體，CloudFormation 會將範本新增到該儲存貯體。

針對 CloudFormation 所建立的 S3 儲存貯體，需謹記的事項

- AWS 帳戶中擁有 Amazon S3 許可的任何人皆可存取這些儲存貯體。
- CloudFormation 建立儲存貯體時，伺服器端加密預設為啟用，藉此加密所有儲存貯體中存放的物件。

您可以直接管理 CloudFormation 已建立的儲存貯體之加密選項；例如，使用位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台或使用 AWS CLI。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [設定 Amazon S3 儲存貯體的預設伺服器端加密行為](#)。

- 您可以使用自己的儲存貯體，並手動上傳範本至 Amazon S3 以管理其許可。當您建立或更新堆疊時，請指定範本檔的 Amazon S3 URL。

- 如需存放在 Amazon S3 儲存貯體的範本，請選擇 Amazon S3 URL。輸入或貼上該範本的 URL，然後選擇 Next (下一步)。

如果您擁有已啟用版本控制的儲存貯體中的範本，則可以指定範本的特定版本，方法是將 `versionId=version-id` 附加到 URL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [使用已啟用版本控制之儲存貯體中的物件](#)。

若偵測到任何語法問題，主控台會提供錯誤訊息，以協助您更正範本。

6. 如果您的範本包含參數，請在 Specify stack details (指定堆疊詳細資訊) 頁面上輸入或修改參數值，然後選擇 Next (下一步)。

CloudFormation 會將堆疊中目前設定的值填入各個參數，但不包括以 NoEcho 屬性宣告的參數；不過，您仍可勾選 Use existing value (使用現有的值) 繼續使用目前的值。

如需使用 NoEcho 遮罩敏感資訊，以及使用動態參數管理密碼的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

7. 在 Configure stack options (設定堆疊選項) 頁面上，您可以更新套用到堆疊的標籤和許可，以及修改進階選項，例如堆疊政策、復原組態，或更新 Amazon SNS 通知主題。如需關於這些選項的詳細資訊，請參閱 [設定堆疊選項](#)。
8. 如果範本包含 IAM 資源，在功能中選擇我知道此範本可建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱 [認可 CloudFormation 範本中的 IAM 資源](#)。
9. 選擇 Next (下一步) 繼續。
10. 檢閱堆疊資訊和您提交的變更。

檢查是否提交了正確的資訊，例如正確的參數值或範本 URL。

在變更集預覽區段中，檢查 CloudFormation 是否會進行您預期的所有變更。例如，您可以檢查 CloudFormation 是否會新增、移除及修改您要新增、移除或修改的資源。CloudFormation 透過建立針對堆疊設定的變更，產生此預覽。如需詳細資訊，請參閱 [透過變更集更新 CloudFormation 堆疊](#)。

11. 當您完成變更後，請選擇 Update stack (更新堆疊)。

Note

此時，您也可以選擇檢視變更集，更仔細地檢閱您提出的更新。若要執行此操作，請選擇 View change set (檢視變更集) 而不是 Update stack (更新堆疊)。CloudFormation 會顯示根據更新所產生的變更集。當您準備好執行堆疊更新時，請選擇 Execute (執行)。

在選取 Events (事件) 窗格的情況下，CloudFormation 會顯示您堆疊的 stack details (堆疊詳細資訊) 頁面。您的堆疊目前的狀態為 UPDATE_IN_PROGRESS。CloudFormation 成功更新堆疊之後，就會將堆疊狀態設為 UPDATE_COMPLETE。

如果堆疊更新失敗，CloudFormation 會自動復原變更，並將堆疊狀態設為 UPDATE_ROLLBACK_COMPLETE。

 Note

您可以在處於 UPDATE_IN_PROGRESS 狀態時取消更新。如需詳細資訊，請參閱[取消堆疊更新](#)。

使用命令列更新堆疊

您可以使用下列其中一項命令：

- [update-stack](#) (AWS CLI)
- [Update-CFNStack](#) (AWS Tools for Windows PowerShell)

如需使用命令列來更新堆疊的範例，請參閱 [AWS CLI 和 PowerShell 的 CloudFormation 堆疊操作命令範例](#)。

取消堆疊更新

堆疊更新開始之後，如有堆疊仍在 UPDATE_IN_PROGRESS 狀態，您可以取消堆疊更新。更新完成之後，您便無法將其取消。不過，您可以使用任何之前的設定重新更新堆疊。

如果您取消堆疊更新，堆疊會復原至啟動堆疊更新之前存在的堆疊組態。

主題

- [更新堆疊更新 \(主控台\)](#)
- [取消堆疊更新 \(AWS CLI\)](#)

更新堆疊更新 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
3. 在堆疊頁面上，選擇正在更新中的堆疊。其狀態必須為 UPDATE_IN_PROGRESS。
4. 選擇 Stack actions (堆疊動作)，然後 Cancel update stack (取消更新堆疊)。
5. 若要繼續取消更新，請選擇 Cancel update (取消更新)。否則，請選擇 Cancel (取消) 繼續更新。

堆疊繼續進行到 UPDATE_ROLLBACK_IN_PROGRESS (正在刪除) 狀態。更新取消完成之後，堆疊會設為 UPDATE_ROLLBACK_COMPLETE。

取消堆疊更新 (AWS CLI)

使用命令 [cancel-update-stack](#) 以取消更新。如需詳細資訊，請參閱[取消堆疊更新](#)。

從 CloudFormation 主控台中刪除堆疊

如果不再需要堆疊中的資源，可以刪除整個堆疊。

刪除堆疊時，除非您使用刪除政策來保留特定資源，否則 CloudFormation 會刪除該堆疊中的所有資源。如需詳細資訊，請參閱 [DeletionPolicy 屬性](#)。

若要刪除堆疊 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
3. 在堆疊頁面中，選取要刪除的堆疊。此堆疊目前必須正在執行。
4. 選擇 Delete (刪除)。
5. 出現確認提示時，請選擇 Delete (刪除)。

Note

堆疊刪除操作開始後將無法停止。堆疊繼續進行到 DELETE_IN_PROGRESS (正在刪除) 狀態。

在堆疊刪除完成之後，堆疊即處於 DELETE_COMPLETE (刪除完成) 狀態。根據預設，CloudFormation 主控台不會顯示狀態為 DELETE_COMPLETE (刪除完成) 的堆疊。若要顯示已刪除的堆疊，您必須按照[在 CloudFormation 主控台中檢視已刪除的堆疊](#)所述變更堆疊檢視的篩選條件。

若要強制刪除堆疊 (主控台)

堆疊刪除可能會失敗，因為堆疊中的資源無法刪除。例如，CloudFormation 將無法刪除另一個堆疊也依賴的資源。任何尚未刪除的資源皆會保持不變，直到您可以成功刪除堆疊為止。如果刪除失敗並傳回 DELETE_FAILED 狀態，您可以選擇使用兩種方法之一重試。

1. 在 CloudFormation 主控台的堆疊頁面中，選取您要強制刪除的堆疊。
2. 在堆疊詳細資訊窗格中，選擇重試刪除。
3. 可以在以下選項中選擇：
 - 刪除此堆疊但保留資源：此選項可讓您選取最初無法刪除但您想在強制堆疊刪除期間保留的特定資源。
 - 強制刪除整個堆疊：此選項會保留無法刪除的所有資源，並保留這些資源的相依性。
4. 選擇刪除，使用您的選擇開始強制刪除程序。

若要檢閱保留的資源 (主控台)

刪除堆疊後，可以檢視主控台中保留的資源。

1. 在堆疊清單中，選擇篩選狀態，然後選取已刪除。
2. 選擇已刪除的堆疊。
3. 選擇 Resources (資源) 標籤。
4. 所有保留的資源都會顯示 DELETE_SKIPPED 狀態。
5. 選擇您要檢閱的保留資源。

若要使用命令列來刪除 VPC

您可以使用下列其中一項命令：

- [delete-stack](#) (AWS CLI)
- [Remove-CFNStack](#) (AWS Tools for Windows PowerShell)

如需使用命令列來刪除堆疊的範例，請參閱 [AWS CLI 和 PowerShell 的 CloudFormation 堆疊操作命令範例](#)。

相關資源

如需針對堆疊刪除錯誤進行疑難排解的說明，請參閱 [刪除堆疊失敗](#) 疑難排解主題。

如需如何防止堆疊遭意外刪除的資訊，請參閱[防止 CloudFormation 堆疊遭到刪除](#)。

在 CloudFormation 主控台中檢視已刪除的堆疊

根據預設，CloudFormation 主控台不會顯示狀態為 DELETE_COMPLETE 的堆疊。若希望主控台顯示已刪除堆疊的相關資訊，則需變更堆疊檢視。

檢視已刪除的堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇已刪除堆疊 AWS 區域 所在的。
3. 在堆疊頁面中，從篩選狀態下拉式清單中選擇已刪除。

CloudFormation 會列出狀態為 DELETE_COMPLETE 的所有堆疊。

另請參閱

- [從 CloudFormation 主控台中刪除堆疊](#)
- [在 CloudFormation 主控台中檢視堆疊資訊](#)

監控堆疊進度

本節說明如何監控正在進行中的堆疊部署。CloudFormation 會提供按時間順序排列的詳細部署事件清單，顯示部署過程中的進度及遇到的任何問題。

主題

- [檢視 CloudFormation 堆疊事件](#)
- [依操作檢視堆疊事件](#)
- [檢視 CloudFormation 堆疊部署時間軸](#)
- [了解 CloudFormation 堆疊建立事件](#)
- [監控堆疊更新的進度](#)
- [繼續復原更新](#)
- [確定堆疊失敗的原因](#)
- [選擇佈建資源時的失敗處理方式](#)

檢視 CloudFormation 堆疊事件

您可以檢視堆疊事件，以監控堆疊與資源在堆疊中的進度和狀態。堆疊事件可協助您了解建立、更新或刪除資源的時間，以及堆疊部署是否如預期進行。

主題

- [檢視堆疊事件 \(主控台\)](#)
- [檢視堆疊事件 \(AWS CLI\)](#)
- [堆疊狀態碼](#)

檢視堆疊事件 (主控台)

檢視堆疊事件

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立堆疊所在的 AWS 區域。
3. 在 CloudFormation 主控台的 Stacks (堆疊) 頁面上，選取堆疊名稱。CloudFormation 會顯示所選取堆疊的堆疊詳細資訊。
4. 選擇事件索引標籤以檢視 CloudFormation 為堆疊產生的堆疊事件。

CloudFormation 每分鐘會自動重新整理堆疊事件。此外，CloudFormation 會在新的堆疊事件發生時顯示有新事件可用徽章。選擇重新整理圖示，以將這些事件載入清單中。透過檢視堆疊建立事件，您即可了解建立堆疊前的事件順序，或者了解對堆疊進行除錯失敗前的事件順序。

正在建立的堆疊會列在堆疊頁面中，狀態為 `CREATE_IN_PROGRESS`。堆疊建立成功後，其狀態將變更為 `CREATE_COMPLETE`。

如需詳細資訊，請參閱[了解 CloudFormation 堆疊建立事件](#)及[監控堆疊更新的進度](#)。

檢視堆疊事件 (AWS CLI)

或者，您可以使用 [describe-stack-events](#) 命令，同時建立堆疊以檢視報告的事件。

以下 `describe-stack-events` 命令說明 *my-stack* 堆疊事件。

```
aws cloudformation describe-stack-events --stack-name my-stack
```

以下是回應範例。

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "7b755820-7edf-11f0-ab15-0673b09f3847",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2025-08-21T22:37:56.243000+00:00",
      "ResourceStatus": "CREATE_COMPLETE",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServer-CREATE_COMPLETE-2025-08-21T22:37:54.356Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServer",
      "PhysicalResourceId": "i-099df76cb31b866a9",
      "ResourceType": "AWS::EC2::Instance",
      "Timestamp": "2025-08-21T22:37:54.356000+00:00",
      "ResourceStatus": "CREATE_COMPLETE",
      "ResourceProperties": "{\"UserData\": \"\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I\\\"\", \"ImageId\": \"ami-0bbc328167dee8f3c\", \"InstanceType\": \"t2.micro\", \"SecurityGroupIds\": [\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\"], \"Tags\": [{\"Value\": \"Bootstrap Tutorial Web Server\", \"Key\": \"Name\"}]}\",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:31.226Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServer",
      "PhysicalResourceId": "i-099df76cb31b866a9",
      "ResourceType": "AWS::EC2::Instance",
      "Timestamp": "2025-08-21T22:37:31.226000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",

```

```

        "ResourceStatusReason": "Resource creation Initiated",
        "ResourceProperties": "{ \"UserData\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\", \\\"ImageId\\\": \\\"ami-0bbc328167dee8f3c\\\", \\\"InstanceType\\\": \\\"t2.micro\\\",
\\\"SecurityGroupIds\\\": [\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"], \\\"Tags\\\":
[{ \\\"Value\\\": \\\"Bootstrap Tutorial Web Server\\\", \\\"Key\\\": \\\"Name\\\" } ] }\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServer-CREATE_IN_PROGRESS-2025-08-21T22:37:29.210Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServer\",
        \"PhysicalResourceId\": \"\",
        \"ResourceType\": \"AWS::EC2::Instance\",
        \"Timestamp\": \"2025-08-21T22:37:29.210000+00:00\",
        \"ResourceStatus\": \"CREATE_IN_PROGRESS\",
        \"ResourceProperties\": \"{ \\\"UserData\\\":
\\\"IyEvYmluL2Jhc2gKeXVtIGluc3RhbGwgLXkgYXdzLWNmbi1ib290c3RyYXAKL29wdC9hd3MvYmluL2Nmbi1pbml0IC12I
\\\", \\\"ImageId\\\": \\\"ami-0bbc328167dee8f3c\\\", \\\"InstanceType\\\": \\\"t2.micro\\\",
\\\"SecurityGroupIds\\\": [\\\"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\\\"], \\\"Tags\\\":
[{ \\\"Value\\\": \\\"Bootstrap Tutorial Web Server\\\", \\\"Key\\\": \\\"Name\\\" } ] }\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",
        \"EventId\": \"WebServerSecurityGroup-
CREATE_COMPLETE-2025-08-21T22:37:28.803Z\",
        \"StackName\": \"my-stack\",
        \"LogicalResourceId\": \"WebServerSecurityGroup\",
        \"PhysicalResourceId\": \"my-stack-WebServerSecurityGroup-n8A43bQT1ty2\",
        \"ResourceType\": \"AWS::EC2::SecurityGroup\",
        \"Timestamp\": \"2025-08-21T22:37:28.803000+00:00\",
        \"ResourceStatus\": \"CREATE_COMPLETE\",
        \"ResourceProperties\": \"{ \\\"GroupDescription\\\": \\\"Allow HTTP access from my IP
address\\\", \\\"SecurityGroupIngress\\\": [{ \\\"CidrIp\\\": \\\"0.0.0.0/0\\\", \\\"Description\\\": \\\"HTTP
\\\", \\\"FromPort\\\": \\\"80\\\", \\\"ToPort\\\": \\\"80\\\", \\\"IpProtocol\\\": \\\"tcp\\\" } ] }\",
        \"ClientRequestToken\": \"token\"
    },
    {
        \"StackId\": \"arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325\",

```



```

      "EventId": "WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:22.626Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "my-stack-WebServerSecurityGroup-n8A43bQT1ty2",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:22.626000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceStatusReason": "Resource creation Initiated",
      "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP
address\", \"SecurityGroupIngress\": [{\"CidrIp\":\"0.0.0.0/0\", \"Description\":\"HTTP
\", \"FromPort\":\"80\", \"ToPort\":\"80\", \"IpProtocol\":\"tcp\"}]}",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "WebServerSecurityGroup-
CREATE_IN_PROGRESS-2025-08-21T22:37:20.186Z",
      "StackName": "my-stack",
      "LogicalResourceId": "WebServerSecurityGroup",
      "PhysicalResourceId": "",
      "ResourceType": "AWS::EC2::SecurityGroup",
      "Timestamp": "2025-08-21T22:37:20.186000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceProperties": "{\"GroupDescription\":\"Allow HTTP access from my IP
address\", \"SecurityGroupIngress\": [{\"CidrIp\":\"0.0.0.0/0\", \"Description\":\"HTTP
\", \"FromPort\":\"80\", \"ToPort\":\"80\", \"IpProtocol\":\"tcp\"}]}",
      "ClientRequestToken": "token"
    },
    {
      "StackId": "arn:aws:cloudformation:aws-region:123456789012:stack/my-
stack/64726230-7edf-11f0-8a36-06453a64f325",
      "EventId": "64740fe0-7edf-11f0-8a36-06453a64f325",
      "StackName": "my-stack",
      "LogicalResourceId": "my-stack",
      "PhysicalResourceId": "arn:aws:cloudformation:aws-
region:123456789012:stack/my-stack/64726230-7edf-11f0-8a36-06453a64f325",
      "ResourceType": "AWS::CloudFormation::Stack",
      "Timestamp": "2025-08-21T22:37:17.819000+00:00",
      "ResourceStatus": "CREATE_IN_PROGRESS",
      "ResourceStatusReason": "User Initiated",
      "ClientRequestToken": "token"
    }
  }

```

```
    ]
}
```

最近的事件會先報告。下表描述 `describe-stack-events` 命令傳回的欄位：

欄位	Description
EventId	事件識別碼。
StackName	事件對應之堆疊的名稱。
StackId	事件對應之堆疊的識別碼。
LogicalResourceId	資源的邏輯識別碼。
PhysicalResourceId	資源的實體識別碼。
ResourceProperties	資源的屬性。
ResourceType	資源的類型。
Timestamp	事件發生時的時間。
ResourceStatus	資源狀態可為下列其中一個狀態碼：CREATE_COMPLETE CREATE_FAILED CREATE_IN_PROGRESS DELETE_COMPLETE DELETE_FAILED DELETE_IN_PROGRESS DELETE_SKIPPED IMPORT_COMPLETE IMPORT_IN_PROGRESS IMPORT_ROLLBACK_COMPLETE IMPORT_ROLLBACK_FAILED IMPORT_ROLLBACK_IN_PROGRESS REVIEW_IN_PROGRESS ROLLBACK_COMPLETE ROLLBACK_FAILED ROLLBACK_IN_PROGRESS UPDATE_COMPLETE UPDATE_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_FAILED UPDATE_IN_PROGRESS UPDATE_ROLLBACK_COMPLETE UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS UPDATE_ROLLBACK_FAILED UPDATE_ROLLBACK_IN_PROGRESS DELETE_SKIPPED 狀態套用至具有保留刪除政策屬性的資源。

欄位	Description
DetailedStatus	堆疊的詳細狀態。若 CONFIGURATION_COMPLETE 存在，則堆疊資源組態階段已完成，且資源穩定化作業正在進行中。
ResourceStatusReason	狀態的詳細資訊。

堆疊狀態碼

下表會說明堆疊的狀態碼：

堆疊狀態和選用的詳細狀態	Description
CREATE_COMPLETE	一或多個堆疊建立成功。
CREATE_IN_PROGRESS	一或多個堆疊建立中。
CREATE_FAILED	一或多個堆疊建立失敗。檢視堆疊事件，查看任何相關聯的錯誤訊息。建立失敗的可能原因包括權限不足，無法使用堆疊中的所有資源、AWS 服務拒絕的參數值，或在資源建立期間逾時。
DELETE_COMPLETE	一或多個堆疊成功刪除。刪除的堆疊仍保留，90 天內仍可檢視。
DELETE_FAILED	一或多個堆疊刪除失敗。因為刪除失敗，所以您可能有一些資源仍在執行，但是您無法使用或更新堆疊。再次刪除堆疊，或檢視堆疊事件，查看任何相關聯的錯誤訊息。
DELETE_IN_PROGRESS	一或多個堆疊移除中。
REVIEW_IN_PROGRESS	一或多個堆疊建立中，這些堆疊具有預期的 StackId，但沒有任何範本或資源。

Important

針對[最大可能堆疊數](#)計數有此狀態碼的堆疊。

堆疊狀態和選用的詳細狀態	Description
ROLLBACK_COMPLETE	在堆疊建立失敗或明確取消建立堆疊之後，成功移除一或多個堆疊。堆疊返回到之前的工作狀態。在建立堆疊操作期間建立的所有資源都予以刪除。 此狀態只會在堆疊建立失敗後出現。此狀態表示來自所建立之部分堆疊的所有操作均已適當清除。在此狀態時，只能執行刪除操作。
ROLLBACK_FAILED	在堆疊建立失敗或明確取消建立堆疊之後，移除一或多個堆疊失敗。刪除堆疊，或檢視堆疊事件，查看任何相關聯的錯誤訊息。
ROLLBACK_IN_PROGRESS	在堆疊建立失敗或明確取消建立堆疊之後，持續移除一或多個堆疊。
UPDATE_COMPLETE	一或多個堆疊更新成功。
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	成功更新堆疊之後，持續移除一或多個堆疊的舊資源。針對需要替換資源的堆疊更新，CloudFormation 會先建立新資源，然後刪除舊資源，藉以減少任何堆疊中斷。在此狀態下，堆疊已更新且可用，但 CloudFormation 仍會刪除舊資源。
UPDATE_FAILED	一或多個堆疊更新不成功。檢視堆疊事件，查看任何相關聯的錯誤訊息。
UPDATE_IN_PROGRESS	持續更新一或多個堆疊。
UPDATE_ROLLBACK_COMPLETE	堆疊更新失敗之後，成功將一或多個堆疊返回到之前的運作狀態。
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	更新堆疊失敗之後，持續移除一或多個堆疊的新資源。在此狀態下，堆疊已復原到之前的運作狀態且可使用，但 CloudFormation 仍將刪除在堆疊更新期間建立的任何新資源。

堆疊狀態和選用的詳細狀態	Description
UPDATE_ROLLBACK_FAILED	堆疊更新失敗之後，將一或多個堆疊返回到之前的運作狀態失敗。在此狀態時，您可以刪除堆疊或 繼續復原 。您可能需要先修正錯誤，您的堆疊才能回到運作狀態。或者，您可以聯絡 將堆疊 支援 還原為可用狀態 。
UPDATE_ROLLBACK_IN_PROGRESS	堆疊更新失敗之後，持續將一或多個堆疊返回到之前的運作狀態。
IMPORT_IN_PROGRESS	匯入操作正在進行。
IMPORT_COMPLETE	已針對支援 resource import 堆疊中的所有資源順利完成匯入操作。
IMPORT_ROLLBACK_IN_PROGRESS	匯入將復原回先前的範本組態。
IMPORT_ROLLBACK_FAILED	堆疊中至少一個資源的匯入復原操作失敗。成功匯入資源 CloudFormation 的結果將可提供使用。
IMPORT_ROLLBACK_COMPLETE	匯入已成功復原回先前的範本組態。

依操作檢視堆疊事件

您可以檢視依操作分組的堆疊事件，以進一步了解對堆疊所做的變更順序和範圍。操作型分組可協助您一起追蹤相關事件，讓您更輕鬆地監控進度，並在問題發生時進行疑難排解。

每個堆疊操作（建立、更新、刪除、轉返）都會指派一個唯一的操作 ID，將所有相關事件分組。這可讓您專注於特定操作，並快速識別失敗的根本原因。

主題

- [先決條件](#)
- [依操作檢視堆疊事件（主控台）](#)
- [依操作檢視堆疊事件 \(AWS CLI\)](#)
- [堆疊狀態碼](#)

先決條件

若要使用 DescribeEvents API，您必須擁有必要的 IAM 許可才能：DescribeEvents。

依操作檢視堆疊事件（主控台）

檢視依操作分組的堆疊事件

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 您在其中建立堆疊的。
3. 在堆疊頁面上，選取堆疊名稱。CloudFormation 會顯示所選取堆疊的堆疊詳細資訊。
4. 選擇事件索引標籤以檢視 CloudFormation 為堆疊產生的堆疊事件。
5. 事件會依操作 ID 自動分組。每個操作都會顯示為可擴展區段，顯示操作類型、狀態和時間戳記。
6. 按一下操作 ID 以開啟詳細檢視，僅顯示與該特定操作相關的事件。
7. 在操作詳細資訊檢視中，選取僅顯示失敗核取方塊，僅顯示失敗的事件以進行根本原因分析。

CloudFormation 每分鐘會自動重新整理堆疊事件。新堆疊事件發生時，會顯示新事件可用徽章。選擇重新整理圖示，以將這些事件載入清單中。

透過檢視依操作分組的堆疊事件，您可以了解每個操作的事件順序，並快速識別哪些特定操作導致問題（如果您要偵錯堆疊）。

當您的堆疊操作正在執行時，會列出狀態為 CREATE_IN_PROGRESS、UPDATE_IN_PROGRESS 或的堆疊操作DELETE_IN_PROGRESS。操作成功完成後，其狀態會變更為 CREATE_COMPLETE、UPDATE_COMPLETE或 DELETE_COMPLETE。

如需詳細資訊，請參閱[了解 CloudFormation 堆疊建立事件](#)及[監控堆疊更新的進度](#)。

依操作檢視堆疊事件 (AWS CLI)

您可以使用 describe-events 命令搭配操作 ID 篩選，以檢視特定操作的事件。

取得最後一個操作 IDs

可透過 describe-stacks API 取得的堆疊描述現在包含 LastOperations 資訊，顯示最近的操作 IDs 及其類型。這可讓您快速識別發生的操作及其目前狀態，而無需剖析事件日誌。

```
aws cloudformation describe-stacks --stack-name MyStack
```

以下是範例回應，顯示最後一個操作是在更新操作失敗後轉返。

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/07580010-bb79-11f0-8f6c-0289bb5c804f",
      "StackName": "MyStack",
      "Description": "A simple CloudFormation template to create an S3 bucket.",

      "CreationTime": "2025-11-07T01:28:13.778000+00:00",
      "LastUpdatedTime": "2025-11-07T01:43:39.838000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "UPDATE_ROLLBACK_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Tags": [],
      "EnableTerminationProtection": false,
      "DriftInformation": {
        "StackDriftStatus": "NOT_CHECKED"
      },
      "LastOperations": [
        {
          "OperationType": "ROLLBACK",
          "OperationId": "d0f12313-7bdb-414d-a879-828a99b36f29"
        },
        {
          "OperationType": "UPDATE_STACK",
          "OperationId": "1c211b5a-4538-4dc9-bfed-e07734371e57"
        }
      ]
    }
  ]
}
```

依操作 ID 篩選事件

下列describe-events命令說明特定操作 ID 的事件：

```
aws cloudformation describe-events \
```

```
--operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57
```

若要僅檢視故障診斷的失敗事件，請使用 `--filter FailedEvents=true` 參數：

```
aws cloudformation describe-events \  
  --operation-id 1c211b5a-4538-4dc9-bfed-e07734371e57 \  
  --filter FailedEvents=true
```

新的操作 ID 篩選功能可讓您專注於特定操作及其相關事件。這對於下列項目特別有用：

- 針對特定故障進行故障診斷：將事件與故障操作隔離，以了解發生了什麼問題。
- 監控長時間執行的操作：追蹤複雜更新或大型堆疊部署的進度。
- 稽核變更：檢閱與特定更新操作相關聯的所有事件。
- 根本原因分析：使用失敗篩選條件快速識別部署問題的來源。

堆疊狀態碼

下表說明使用操作 ID 篩選時 `describe-events` 命令傳回的欄位：

欄位	Description
EventId	事件識別碼。
OperationId	產生此事件之操作的唯一識別符。
StackName	事件對應之堆疊的名稱。
StackId	事件對應之堆疊的識別碼。
LogicalResourceId	資源的邏輯識別碼。
PhysicalResourceId	資源的實體識別碼。
ResourceProperties	資源的屬性。
ResourceType	資源的類型。

欄位	Description
Timestamp	事件發生時的時間。
ResourceStatus	資源的狀態 (CREATE_COMPLETE 、 UPDATE_FAILED 等) 。
DetailedStatus	堆疊的詳細狀態。若 CONFIGURATION_COMPLETE 存在，則堆疊資源組態階段已完成，且資源穩定化作業正在進行中。
ResourceStatusReason	狀態的詳細資訊。

檢視 CloudFormation 堆疊部署時間軸

堆疊部署時間軸圖表以視覺化方式呈現堆疊部署時間軸。此檢視會顯示堆疊及其每個資源的部署狀態，及每個狀態變更的時間。堆疊部署狀態以對應的顏色表示。

主題

- [了解堆疊部署時間軸圖表](#)
- [檢視堆疊部署時間軸圖表 \(主控台\)](#)

了解堆疊部署時間軸圖表

下圖顯示由於 Amazon EC2 執行個體資源啟動失敗而導致部署失敗的堆疊部署時間軸圖表。

Table view

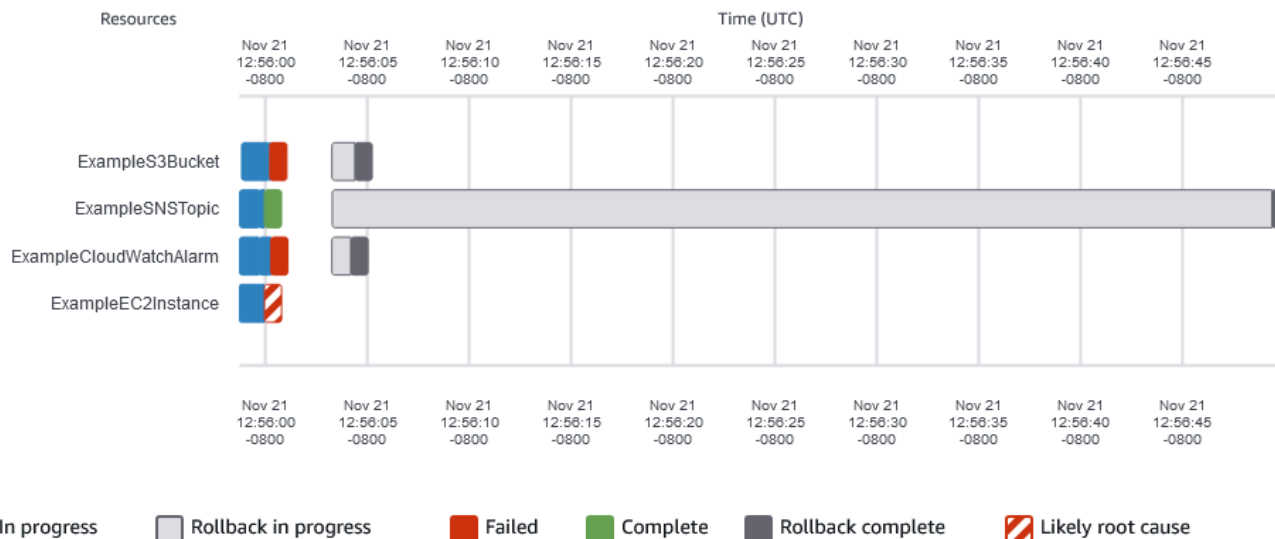
Timeline view - new

Latest deployment timeline - new

View root cause



This is a timeline view of your latest stack deployment. If you start a new deployment, this view will reset.

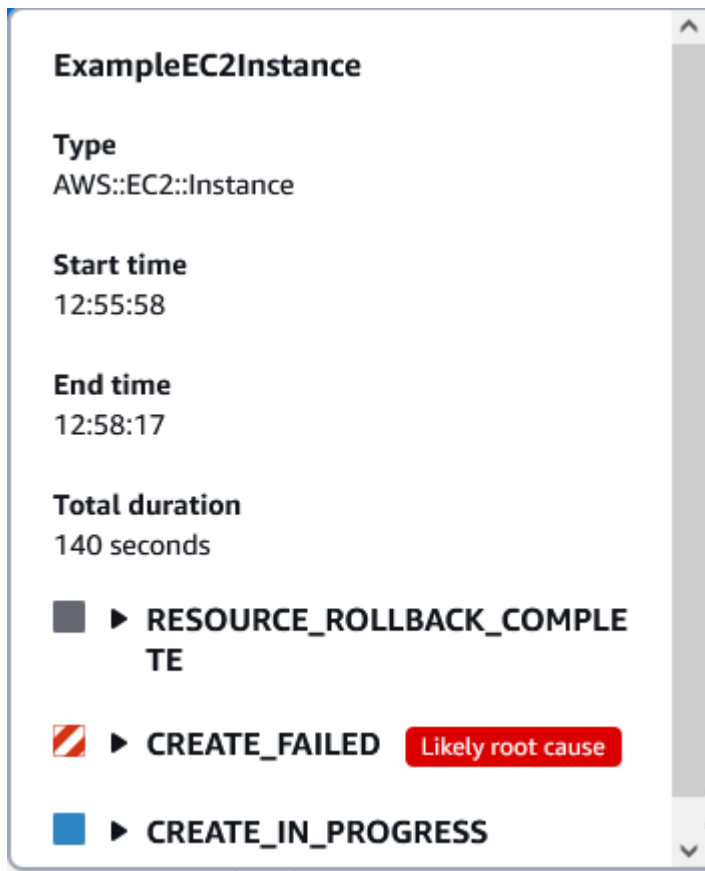


堆疊資源的名稱位於圖表左側，與部署時間相關的日期和時間位於圖表頂部。

每個資源一開始均處於進行中狀態。對於每個成功部署，狀態列會變更為完成。當資源部署失敗時，狀態列會變更為失敗。當資源部署失敗且堆疊部署也失敗時，導致堆疊部署失敗的資源會獲得可能為根本原因失敗狀態。

堆疊部署操作失敗後，已成功部署的資源開始復原，並變更為復原進行中狀態。在資源完成復原後，狀態會變更為復原完成。

選擇每個資源可提供部署時間軸的更精細的詳細資訊：



選擇資源會顯示部署的類型、部署開始時間、部署結束時間及總持續時間。您還可以在下方的下拉式選單中找到每個部署狀態的開始時間、結束時間及持續時間。如果資源部署失敗，將提供失敗原因。

如需堆疊狀態的詳細資訊，請參閱 [堆疊狀態碼](#)。

檢視堆疊部署時間軸圖表 (主控台)

檢視堆疊部署時間軸圖表：

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 您在其中建立堆疊的。
3. 在 CloudFormation 主控台的堆疊頁面上，選擇堆疊名稱。CloudFormation 會顯示所選取堆疊的堆疊詳細資訊。
4. 選擇事件索引標籤以檢視 CloudFormation 為堆疊產生的堆疊事件。
5. 選擇時間軸圖表按鈕，以檢視堆疊的時間軸圖表。

了解 CloudFormation 堆疊建立事件

在堆疊部署期間，會發生多個事件以建立、設定和驗證堆疊範本中定義的資源。了解這些事件能協助您最佳化堆疊建立程序，並簡化部署作業。

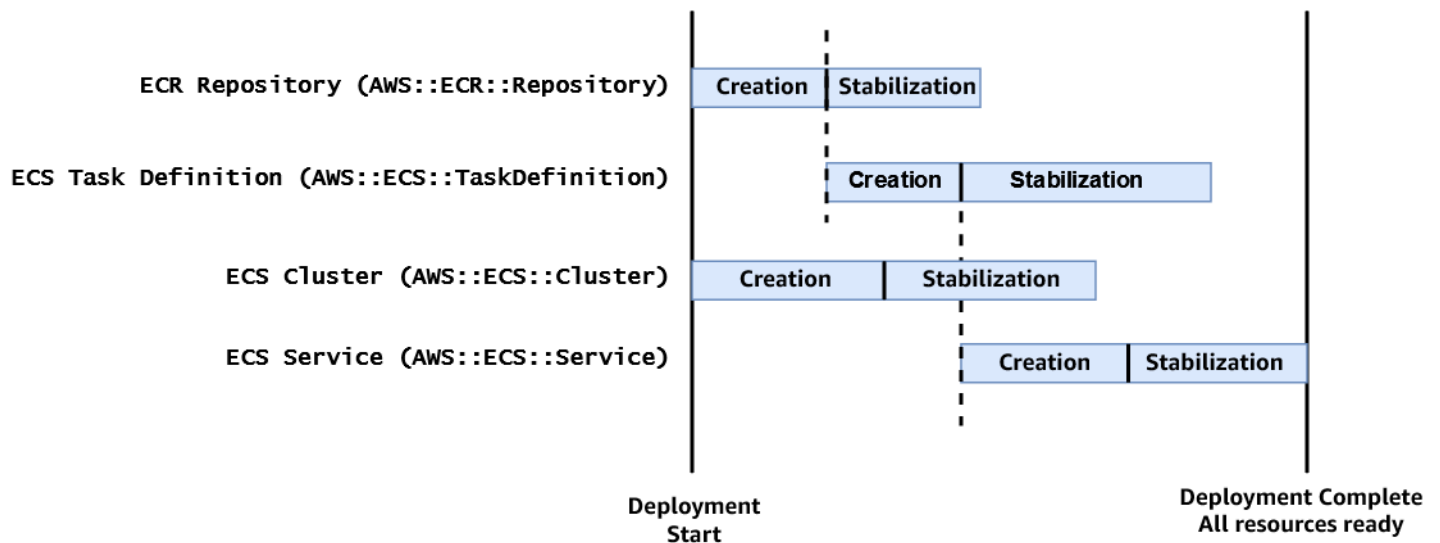
- 資源建立事件 – 當每個資源開始建立程序時，會設定 `CREATE_IN_PROGRESS` 事件狀態。此事件表示資源正在佈建中。
- 最終一致性檢查 – 堆疊建立時間的相當大部分用於對堆疊建立的資源執行最終一致性檢查。在此階段，服務會執行內部一致性檢查，確保資源可完全運作，並符合每個資源定義的服務穩定標準 AWS 服務。
- 組態完成事件 – 每個資源完成佈建的最終一致性檢查階段時，會設定詳細狀態為 `CONFIGURATION_COMPLETE` 的事件。
- 資源建立完成事件 – 資源依照指定建立和設定完成，且組態與範本中指定的內容一致後，會設定狀態為 `CREATE_COMPLETE` 的事件。

在不需要資源最終一致性檢查的場景中，例如驗證預生產環境堆疊組態或跨堆疊佈建，您可以利用 `CONFIGURATION_COMPLETE` 事件簡化堆疊建立程序。您可以透過多種方式使用此事件。例如，您可以將其做為視覺化信號，跳過等待資源或堆疊一致性檢查完成的步驟。或者，您可以使用它來建立使用持續整合和持續交付 (CI/CD) 建立自動化機制，以觸發額外動作。

Important

雖然利用 `CONFIGURATION_COMPLETE` 事件能加快堆疊建立速度，但您需了解其權衡利弊。首先，它僅支援支援漂移偵測的部分資源類型。如需支援漂移偵測的資源類型清單，請參閱 [資源類型支援](#)。此方法可能不適用於所有案例，尤其是在資源需要徹底的最終一致性檢查，以確保整個雲端環境 (例如在生產環境中) 的完整營運準備度的情況。建議您仔細評估您的部署需求以及每個資源一致性檢查的重要性。利用 `CONFIGURATION_COMPLETE` 事件最佳化部署速度，同時不影響基礎設施的完整性和可靠性。

由於無法保證一定會設定 `CONFIGURATION_COMPLETE` 事件，因此所有使用此事件的場景都應準備好在未設定 `CREATE_COMPLETE` 事件時，處理 `CONFIGURATION_COMPLETE` 事件。



堆疊部署開始時，AWS::ECR::Repository 和 AWS::ECS::Cluster 資源都會啟動建立程序 (ResourceStatus = CREATE_IN_PROGRESS)。當 AWS::ECR::Repository 資源類型開始執行最終一致性檢查 (DetailedStatus = CONFIGURATION_COMPLETE) 後，AWS::ECS::TaskDefinition 資源即可啟動建立程序。同樣地，一旦 AWS::ECS::TaskDefinition 資源開始最終一致性檢查，AWS::ECS::Service 資源就會開始建立程序。

CREATE_IN_PROGRESS 和 CREATE_COMPLETE 事件

- [堆疊] : CREATE_IN_PROGRESS
- [資源] : ECR 儲存庫 CREATE_IN_PROGRESS
- [資源] : ECS 叢集 CREATE_IN_PROGRESS
- [資源] : ECR 儲存庫 CREATE_IN_PROGRESS、CONFIGURATION_COMPLETE
- [資源] : ECS 任務定義 CREATE_IN_PROGRESS
- [資源] : ECS 叢集 CREATE_IN_PROGRESS、CONFIGURATION_COMPLETE
- [資源] : ECS 任務定義 CREATE_IN_PROGRESS、CONFIGURATION_COMPLETE
- [資源] : ECS 服務 CREATE_IN_PROGRESS
- [資源] : ECR 儲存庫 CREATE_COMPLETE
- [資源] : ECS 叢集 CREATE_COMPLETE
- [資源] : ECS 服務 CREATE_IN_PROGRESS、CONFIGURATION_COMPLETE
- [堆疊] : CREATE_IN_PROGRESS、CONFIGURATION_COMPLETE
- [資源] : ECS 任務定義 CREATE_COMPLETE

- [資源] : ECS 服務 CREATE_COMPLETE
- [堆疊] : CREATE_COMPLETE

監控堆疊更新的進度

透過檢視堆疊事件，即可監控該堆疊更新的進度。堆疊的事件索引標籤會顯示堆疊建立與更新過程中的每個主要步驟，並依照各事件的時間進行排序，而最新的事件會置於最上方。如需詳細資訊，請參閱[監控堆疊進度](#)。

主題

- [成功堆疊更新期間產生的事件](#)
- [資源更新失敗時產生的事件](#)

成功堆疊更新期間產生的事件

系統會在堆疊更新程序的起始處，將堆疊標上 UPDATE_IN_PROGRESS 事件：

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

接下來，若更新後範本中的各資源有所變更，則系統會在開始和完成更新時標記下述事件。例如，更新名為 MyDB 的 [AWS::RDS::DBInstance](#) 資源時，會出現下列項目：

```
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

當 CloudFormation 報告已開始更新資源時，會記錄 UPDATE_IN_PROGRESS 事件。當資源成功建立時，會記錄 UPDATE_COMPLETE 事件。

當 CloudFormation 成功更新堆疊時，您會看到以下事件：

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

Important

在堆疊更新操作期間，如果 CloudFormation 需要替換現有資源，首先會建立一個新的資源，然後刪除舊的資源。但是，在某些情況下，CloudFormation 可能無法刪除舊資源（例如，如果使用者沒有許可刪除給定類型的資源）。

CloudFormation 會嘗試刪除舊資源三次。如果 CloudFormation 無法刪除舊資源，即會從堆疊中移除舊資源並繼續更新堆疊。堆疊更新完成時，CloudFormation 會發出 UPDATE_COMPLETE 堆疊事件，但內含 StatusReason，其會指出無法刪除一或多個資源。CloudFormation 也會針對特定資源發出 DELETE_FAILED 事件，內含相應的 StatusReason，其會說明 CloudFormation 為何無法刪除此資源的更多詳細資訊。

舊資源仍存在，並且會繼續產生費用，但無法再透過 CloudFormation 存取。若要刪除舊資源，請直接使用主控台或基礎服務的 API 來存取舊資源。

這也適用於已從堆疊範本移除的資源，該資源將會在堆疊更新期間從堆疊中刪除。

資源更新失敗時產生的事件

如果資源更新失敗，CloudFormation 即會回報 UPDATE_FAILED 事件，並包含失敗的原因。舉例來說，如果更新範本指定該資源不支援的屬性變更 (如縮減 [AWS::RDS::DBInstance](#) 資源的 AllocatedStorage 規模)，則出現的事件將如下所示：

```
2011-09-30 09:36 PDT AWS::RDS::DBInstance MyDB UPDATE_FAILED Size cannot be less than
current size; requested: 5; current: 10
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

如果資源更新失敗，CloudFormation 即會將升級期間所更新的任何資源復原為原始組態，隨後再進行更新。此處範例將顯示更新復原期間會出現的事件：

```
2011-09-30 09:38 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_COMPLETE
2011-09-30 09:38 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:37 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
2011-09-30 09:37 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_IN_PROGRESS The
following resource(s) failed to update: [MyDB]
```

繼續復原更新

有時候，當 CloudFormation 嘗試復原堆疊更新時，無法回復更新過程中所做的所有變更。這種情況稱為 `UPDATE_ROLLBACK_FAILED` 狀態。例如，您可能會有一個堆疊，其開始復原回已在 CloudFormation 之外刪除的舊資料庫執行個體。因為 CloudFormation 不知道資料庫已遭刪除，它會假設資料庫執行個體仍然存在，並嘗試復原回該資料庫執行個體，造成更新復原失敗。

`UPDATE_ROLLBACK_FAILED` 狀態的堆疊無法更新，但可以復原為工作狀態 (`UPDATE_ROLLBACK_COMPLETE`)。使堆疊回到其原始設定後，您可以嘗試再次更新。

在大多數情況下，您必須修正造成更新復原失敗的錯誤，才能繼續復原您的堆疊。在其他情況下，您可以繼續復原更新，而無須進行任何變更 (例如：當堆疊操作逾時時)。

Note

若您使用巢狀堆疊，復原父系堆疊也會嘗試復原所有子堆疊。

繼續復原更新 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
3. 在堆疊頁面上，選擇要更新的堆疊，選擇堆疊動作，然後選擇繼續更新復原。

若 [故障診斷錯誤](#) 中的任何解決方案都無效，您可以使用進階選項，跳過 CloudFormation 無法成功復原的資源。您必須 [尋找](#) 並輸入您希望跳過的資源邏輯 ID。請只指定在 UpdateRollback 期間進入 `UPDATE_FAILED` 狀態的資源，而非向前更新期間進入該狀態的資源。

Warning

CloudFormation 會將指定資源的狀態設為 `UPDATE_COMPLETE` 並繼續復原堆疊。在復原完成後，跳過資源的狀態會和堆疊範本中資源的狀態不一致。在執行下一次堆疊更新前，您必須更新堆疊或資源，使其和彼此保持一致。若您未執行此作業，後續的堆疊更新可能會失敗，堆疊可能會變得無法復原。

指定成功復原您堆疊所需要的最小資源數。例如，失敗的資源更新可能會導致依存資源失敗。在這種情況下，您可能不需要跳過依存資源。

若要跳過屬巢狀堆疊的部分資源，請使用下列格式：

式：*NestedStackName.ResourceLogicalID*。若您希望在 Type:

AWS::CloudFormation::Stack 清單中指定堆疊資源 (ResourcesToSkip) 的邏輯 ID，則其對應的內嵌堆疊必須處於下列任一狀態：DELETE_IN_PROGRESS、DELETE_COMPLETE，或 DELETE_FAILED。

繼續復原更新 (AWS CLI)

- 使用 [continue-update-rollback](#) 命令搭配 --stack-name 選項，指定您希望繼續復原的堆疊 ID。

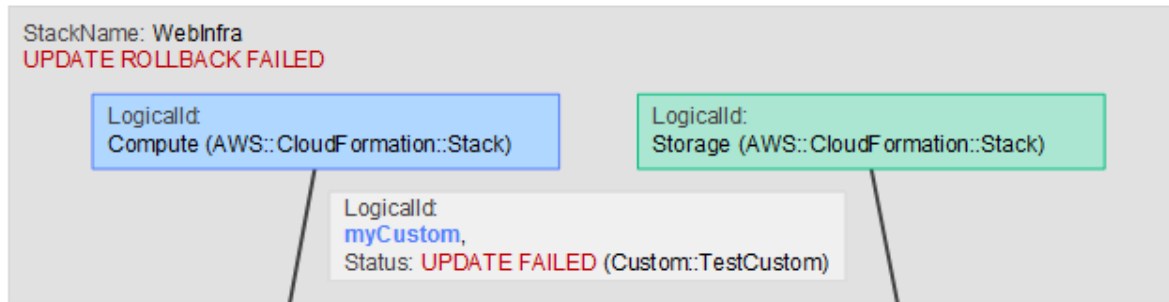
從失敗的巢狀堆疊更新中繼續復原

當您有多個彼此巢狀的堆疊時，可能需要跳過多個嵌套層級的資源，才能讓全堆疊階層回到可運作狀態。

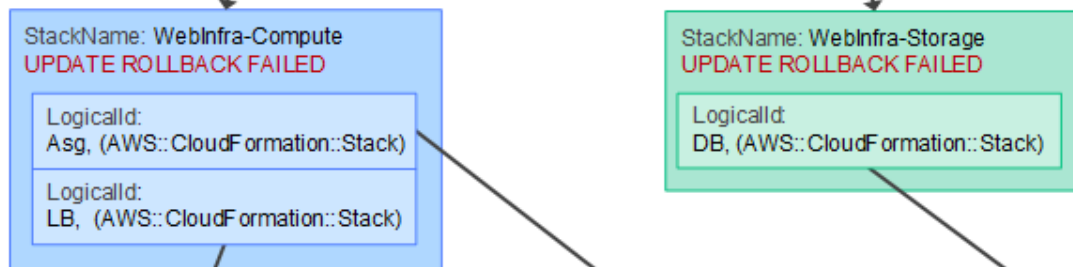
例如，您有一個名為 WebInfra 的根堆疊，其中包含兩個較小的堆疊：WebInfra-Compute 和 WebInfra-Storage。這兩個堆疊本身也包含各自的巢狀堆疊。

若更新過程中發生問題導致更新失敗，整個堆疊階層可能會進入 UPDATE_ROLLBACK_FAILED 狀態，如下圖所示。

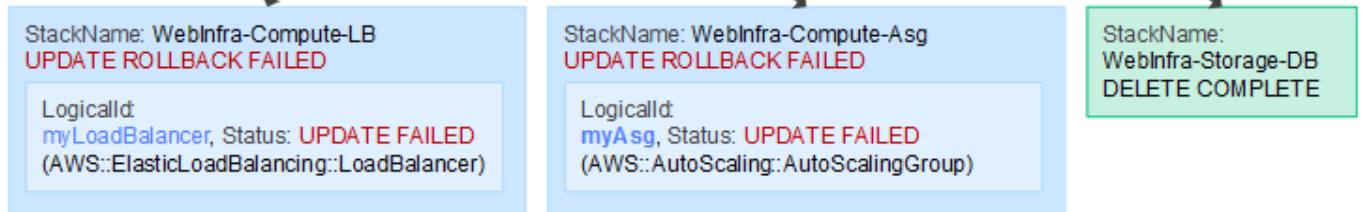
Root



Level 1



Level 2

**Note**

為求簡潔，此範例中的堆疊名稱已進行截斷。子堆疊名稱通常會由 CloudFormation 產生，並且包含唯一的隨機字串，因此實際的名稱可能不方便使用。

若要使用 `continue-update-rollback` 命令讓根堆疊成功進入可運作狀態，您必須使用 `--resources-to-skip` 選項跳過復原失敗的資源。

下列 `continue-update-rollback` 範例從先前失敗的堆疊更新處繼續復原操作。在此範例中，`--resources-to-skip` 選項會包含下列項目：

- *myCustom*
- *WebInfra-Compute-Asg.myAsg*

- *WebInfra-Compute-LB.myLoadBalancer*
- *WebInfra-Storage.DB*

對於根堆疊的資源，只需提供邏輯 ID，例如 *myCustom*。但對於巢狀堆疊中包含的資源，必須同時提供巢狀堆疊名稱和資源邏輯 ID，中間以句點分隔。例如 *WebInfra-Compute-Asg.myAsg*。

```
aws cloudformation continue-update-rollback --stack-name WebInfra \  
  --resources-to-skip myCustom WebInfra-Compute-Asg.myAsg WebInfra-Compute-  
  LB.myLoadBalancer WebInfra-Storage.DB
```

尋找巢狀堆疊的堆疊名稱

您可以在子堆疊的堆疊 ID 或 Amazon Resource Name (ARN) 中找到它。

下列 ARN 範例對應的堆疊名稱為 *WebInfra-Storage-Z2VKC706XKXT*。

```
arn:aws:cloudformation:us-east-1:123456789012:stack/WebInfra-Storage-Z2VKC706XKXT/  
ea9e7f90-54f7-11e6-a032-028f3d2330bd
```

尋找巢狀堆疊的邏輯 ID

您可以在子堆疊的父系範本定義中找到其邏輯 ID。在圖表中，*WebInfra-Storage-DB* 子堆疊的 *LogicalId* 為位於其父系 *WebInfra-Storage* 中的 *DB*。

在 CloudFormation 主控台中，您也可以 Resources (資源) 索引標籤或 Events (事件) 索引標籤上，在堆疊資源的 Logical ID (邏輯 ID) 資料行中找到邏輯 ID。如需詳細資訊，請參閱 [在 CloudFormation 主控台中檢視堆疊資訊](#)。

確定堆疊失敗的原因

如果堆疊建立失敗，則 CloudFormation 可協助您確定可能是堆疊故障之根本原因的事件。視案例和您的許可而定，AWS CloudTrail 事件可能可以提供有關根本原因的進一步詳細資訊，以防事件中提供的狀態原因不清楚。

若要判斷堆疊失敗的根本原因

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在堆疊頁面中，選擇失敗的堆疊。
3. 選擇事件索引標籤。

4. 選擇偵測根本原因。透過將可能的根本原因標籤新增至特定事件狀態，CloudFormation 將對故障進行分析，並指出可能是故障根本原因的事件。如需 CloudFormation 主控台中狀態的進一步說明，請參閱狀態原因。
5. 使用可能的根本原因標籤選擇失敗的狀態，以進一步了解失敗原因。根據案例和您的許可，您可以檢閱詳細的 CloudTrail 事件。以下是選擇狀態的潛在結果
 - 與此問題相關的 CloudTrail 事件可供使用，可能有助於解決問題。檢視 CloudTrail 事件。
 - 我們找不到與此問題相關的有助於解決問題的任何 CloudTrail 事件。
 - 您目前的許可不允許檢視 CloudTrail 事件的存取權。進一步了解。
 - 在檢查是否有可用的 CloudTrail 事件的過程中，請在幾分鐘後回來查看。
 - 擷取 CloudTrail 事件時發生錯誤。如需手動檢查，請造訪 CloudTrail 主控台。
6. 如果狀態原因中提供的原因不清楚，且根本原因顯示 CloudTrail 主控台的連結，請開啟連結以檢視事件，以尋找詳細的根本原因。

如需 CloudTrail 事件的詳細資訊，請參閱[了解 CloudTrail 事件](#)和 [CloudTrail 記錄內容](#)。

如需 CloudTrail 事件歷史記錄的詳細資訊，請參閱[使用 CloudTrail 事件歷史記錄](#)。

Note

巢狀堆疊不支援偵測根本原因。

選擇佈建資源時的失敗處理方式

若您的堆疊操作失敗，則不必每次都復原已成功佈建的資源，並從頭開始。反之，您可以對處於 `CREATE_FAILED` 或 `UPDATE_FAILED` 狀態的資源疑難排解，然後從問題發生點繼續佈建。

要這樣做，您必須啟用保留成功佈建的資源選項。此選項適用於所有堆疊部署和變更集操作。

- 建立堆疊時，若選擇保留成功佈建資源選項，CloudFormation 會保留成功建立之資源的狀態，並將失敗的資源保持在失敗狀態，直到執行下次更新操作為止。
- 在更新與變更集操作期間，保留已成功佈建的資源選項可保留成功資源的狀態，同時將失敗資源復原到上一個已知穩定狀態。失敗的資源將處於 `UPDATE_FAILED` 狀態。沒有上一個已知穩定狀態的資源將在下一次堆疊操作時遭刪除。

主題

- [堆疊失敗選項概觀](#)
- [暫停堆疊復原的必要條件](#)
- [保留已成功佈建的資源 \(主控台\)](#)
- [保留已成功佈建的資源 \(AWS CLI\)](#)

堆疊失敗選項概觀

從 CloudFormation 主控台、API 或 發出 操作之前 AWS CLI，請指定佈建資源失敗的行為。然後，繼續進行資源的部署過程，而無需進行任何其他修改。在操作失敗的情況下，CloudFormation 會在每個獨立佈建路徑中的第一次失敗時停止。CloudFormation 可識別資源之間的相依性，以平行化獨立佈建動作。然後，它會繼續在每個獨立佈建路徑上佈建資源，直到遇到失敗為止。一個路徑中的失敗不會影響其他佈建路徑。CloudFormation 將繼續佈建資源，直到完成或在遇到其他失敗時停止。

修復任何問題以繼續部署程序。在早些時候無法成功佈建的資源上重試佈建動作之前，CloudFormation 會先執行必要的更新。您可以透過提交 Retry (重試)、Update (更新) 或 Roll back (復原) 操作來修復問題。例如，如果您要佈建 Amazon EC2 執行個體，且 EC2 執行個體在建立操作期間失敗，您可能需要調查錯誤，而不是立即復原失敗的資源。您可以檢閱系統狀態檢查和執行個體狀態檢查，一旦問題得到解決，選取 Retry (重試) 操作。

當堆疊操作失敗，並且您已經從 Stack failure options (堆疊失敗選項) 選單指定 Preserve successfully provisioned resources (保留已成功佈建的資源) 時，您可以選取下列選項。

- Retry (重試) – 在失敗的資源上重試佈建操作，並繼續佈建範本，直到成功完成堆疊操作或下一次失敗為止。如果資源因為不需要修改範本的問題而無法佈建，例如 AWS Identity and Access Management (IAM) 許可，請選取此選項。
- Update (更新) – 已佈建的資源會在範本更新時更新。將重試建立或更新失敗的資源。如果資源因範本錯誤而無法佈建，且您已修改範本，請選取此選項。當您更新狀態為 FAILED 的堆疊時，您必須針對 Stack failure options (堆疊失敗選項) 選取 Preserve successfully provisioned resources (保留已成功佈建的資源)，以繼續更新您的堆疊。
- Roll back (復原) – CloudFormation 會將堆疊復原到上一個已知穩定狀態。

暫停堆疊復原的必要條件

要防止 CloudFormation 自動復原和刪除已成功建立的資源，必須符合下列條件。

1. 建立或更新堆疊時，您必須選擇保留成功佈建資源的選項。這會告知 CloudFormation，即使整體堆疊操作失敗，也不要刪除已成功建立的資源。

2. 堆疊操作必須已失敗，即堆疊狀態為 CREATE_FAILED 或 UPDATE_FAILED。

Note

不支援不可變更新類型。

保留已成功佈建的資源 (主控台)

Create stack

在建立堆疊操作期間保留已成功佈建的資源

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從堆疊頁面的右上角，選擇建立堆疊，並選擇使用新資源 (標準)。
3. 對於先決條件 - 準備範本，選取選擇現有範本。
4. 在 Specify template (指定範本) 下，選擇指定堆疊範本所在 S3 儲存貯體的 URL，或上傳堆疊範本檔案。然後選擇下一步。
5. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，在 Stack name (堆疊名稱) 方塊中輸入堆疊名稱。
6. 在 Parameters (參數) 部分中，指定堆疊範本中定義的參數。

您可以用預設值來使用或變更任何參數。

7. 當您滿意參數值時，請選擇 Next (下一步)。
8. 在 Configure stack options (設定堆疊選項) 頁面上，您可以設定堆疊的其他選項。
9. 對於 Stack failure options (堆疊失敗選項)，請選取 Preserve successfully provisioned resources (保留已成功佈建的資源)。
10. 當您滿意堆疊選項時，請選擇 Next (下一步)。
11. 在 Review (檢閱) 頁面上檢閱您的堆疊並選取 Create stack (建立堆疊)。

結果：建立失敗的堆疊會將堆疊狀態轉換為 CREATE_FAILED，以防止堆疊在堆疊操作遭遇失敗時復原。已成功佈建的資源處於 CREATE_COMPLETE 狀態。您可以在 Stack events (堆疊事件) 索引標籤中監控堆疊。

Update stack

在更新堆疊操作期間保留已成功佈建的資源

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 選取您希望更新的堆疊，然後選擇 Update (更新)。
3. 在 Update stack (更新堆疊) 頁面上，使用下列其中一個選項選擇堆疊範本：
 - 使用現有範本
 - Replace current template (取代目前範本)
 - 在 Infrastructure Composer 中編輯範本

接受您的設定並選取 Next (下一步)。

4. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，指定堆疊範本中所定義的參數。

您可以用預設值來使用或變更任何參數。

5. 當您滿意參數值時，請選擇 Next (下一步)。
6. 在 Configure stack options (設定堆疊選項) 頁面上，您可以設定堆疊的其他選項。
7. 對於 Behavior on provisioning failure (佈建失敗時的行為)，選取 Preserve successfully provisioned resources (保留已成功佈建的資源)。
8. 當您滿意堆疊選項時，請選擇 Next (下一步)。
9. 在 Review (檢閱) 頁面上檢閱您的堆疊並選取 Update stack (更新堆疊)。

結果：更新失敗的堆疊會將狀態轉換為 UPDATE_FAILED，並復原到上一個已知穩定狀態。沒有上一個已知穩定狀態的資源將在下一次堆疊操作時由 CloudFormation 刪除。已成功佈建的資源處於 CREATE_COMPLETE 或 UPDATE_COMPLETE 狀態。您可以在 Stack events (堆疊事件) 索引標籤中監控堆疊。

Change set

Note

您可以為狀態為 CREATE_FAILED 或 UPDATE_FAILED 的堆疊啟動變更集，但不適用於狀態 UPDATE_ROLLBACK_FAILED。

在變更集操作期間保留已成功佈建的資源

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 選取包含您要啟動的變更集的堆疊，然後選擇 Change sets (變更集) 索引標籤。
3. 選取變更集，然後選擇 Execute (執行)。
4. 對於 Execute change set (執行變更集)，選取 Preserve successfully provisioned resources (保留已成功佈建的資源) 選項。
5. 選取 Execute change set (執行變更集)。

結果：更新失敗的堆疊會將狀態轉換為 UPDATE_FAILED，並復原到上一個已知穩定狀態。沒有上一個已知穩定狀態的資源將在下一次堆疊操作時由 CloudFormation 刪除。已成功佈建的資源處於 CREATE_COMPLETE 或 UPDATE_COMPLETE 狀態。您可以在 Stack events (堆疊事件) 索引標籤中監控堆疊。

保留已成功佈建的資源 (AWS CLI)

Create stack

在堆疊建立操作期間保留已成功佈建的資源

指定 [create-stack](#) 操作期間的 `--disable-rollback` 選項或 `on-failure DO_NOTHING` 列舉。

1. 使用 `--disable-rollback` 選項向 `create-stack` 命令提供堆疊名稱和範本。

```
aws cloudformation create-stack --stack-name myteststack \  
  --template-body file://template.yaml \  
  --disable-rollback
```

命令會傳回下列輸出：

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

2. 使用 `describe-stacks` 命令描述堆疊的狀態。


```
aws cloudformation describe-stacks --stack-name myteststack
```

命令會傳回下列輸出：

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "StackStatusReason": "The following resource(s) failed to create:
[MyBucket]",
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "CREATE_FAILED",
      "DisableRollback": true
    }
  ]
}
```

Update stack

在更新堆疊操作期間保留已成功佈建的資源

1. 使用 `--disable-rollback` 選項向 `update-stack` 命令提供現有堆疊名稱和範本。

```
aws cloudformation update-stack --stack-name myteststack \
  --template-url https://s3.amazonaws.com/amzn-s3-demo-bucket/updated.template
  --disable-rollback
```

命令會傳回下列輸出：

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. 使用 `describe-stacks` 或 `describe-stack-events` 選項描述堆疊的狀態。

```
aws cloudformation describe-stacks --stack-name myteststack
```

命令會傳回下列輸出：

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "UPDATE_COMPLETE",
      "DisableRollback": true
    }
  ]
}
```

Change set

Note

您可以為狀態為 `CREATE_FAILED` 或 `UPDATE_FAILED` 的堆疊啟動變更集，但不適用於狀態 `UPDATE_ROLLBACK_FAILED`。

在變更集操作期間保留已成功佈建的資源

指定 [execute-change-set](#) 操作期間的 `--disable-rollback` 選項。

1. 使用 `--disable-rollback` 選項向 `execute-change-set` 命令提供堆疊名稱和範本。

```
aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --template-body file://template.yaml
```

命令會傳回下列輸出：

```
{
  "Id": "arn:aws:cloudformation:us-east-1:123456789012:changeSet/my-change-set/
bc9555ba-a949-xmpl-bfb8-f41d04ec5784",
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

2. 使用 `--disable-rollback` 選項啟動變更集。

```
aws cloudformation execute-change-set --stack-name myteststack \
  --change-set-name my-change-set --disable-rollback
```

3. 使用 `describe-stacks` 或 `describe-stack-events` 選項判斷堆疊的狀態。

```
aws cloudformation describe-stack-events --stack-name myteststack
```

命令會傳回下列輸出：

```
{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "EventId": "49c966a0-7b74-11ea-8071-024244bb0672",
      "StackName": "myteststack",
      "LogicalResourceId": " MyBucket",
      "PhysicalResourceId": "myteststack-MyBucket-abcdefghijkl1",
      "ResourceType": "AWS::S3::Bucket",
      "Timestamp": "2020-04-10T21:43:17.015Z",
      "ResourceStatus": "UPDATE_FAILED"
      "ResourceStatusReason": "User XYZ is not allowed to perform
S3::UpdateBucket on MyBucket"
    }
  ]
}
```

4. 修正許可錯誤，然後重試操作。

```
aws cloudformation update-stack --stack-name myteststack \
  --use-previous-template --disable-rollback
```

命令會傳回下列輸出：

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

5. 使用 `describe-stacks` 或 `describe-stack-events` 選項描述堆疊的狀態。

```
aws cloudformation describe-stacks --stack-name myteststack
```

命令會傳回下列輸出：

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
      "Description": "AWS CloudFormation Sample Template",
      "Tags": [],
      "Outputs": [],
      "CreationTime": "2013-08-23T01:02:15.422Z",
      "Capabilities": [],
      "StackName": "myteststack",
      "StackStatus": "UPDATE_COMPLETE",
      "DisableRollback": true
    }
  ]
}
```

復原堆疊

您可使用 [rollback-stack](#) 命令，將具有 `CREATE_FAILED` 或 `UPDATE_FAILED` 堆疊狀態的堆疊復原至其最後穩定狀態。

下列 `rollback-stack` 命令會復原指定的堆疊。

```
aws cloudformation rollback-stack --stack-name myteststack
```

命令會傳回下列輸出：

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Note

如果堆疊不包含上一個已知穩定狀態，rollback-stack 操作將刪除堆疊。

透過復原觸發條件，在警示違規時復原您的 CloudFormation 堆疊

透過轉返觸發條件，您可以在堆疊建立和更新期間 CloudFormation 監控應用程式的狀態，並在應用程式超過您指定的警示閾值時轉返該操作。對於建立的每個復原觸發條件，您都要指定 CloudFormation 應監控的 CloudWatch 警示。CloudFormation 會在堆疊建立或更新操作期間，以及所有資源部署後一段指定的時間量內，監控指定的警示。如有任何警示在堆疊操作或監控期間進入 ALARM 狀態，CloudFormation 將會復原整個堆疊操作。

您可以設定監控時間，從預設的 0 分鐘到最長 180 分鐘。在這段期間，CloudFormation 會在堆疊建立或更新操作部署所有必要的資源之後，監控所有復原觸發條件。如有任何警示在此堆疊操作或監控期間進入 ALARM 狀態，CloudFormation 將會復原整個堆疊操作。接著，以更新操作而言，如果監控期間過期，而且沒有任何警示進入 ALARM 狀態，CloudFormation 會繼續照常處置舊資源。如果您設定監控時間，但未指定任何復原觸發條件，CloudFormation 仍會等待指定的期間，然後再清理更新操作的舊資源。您可以使用此監控期間來執行任何手動堆疊驗證，以及視需要手動取消堆疊建立或更新。如果您設定 0 分鐘的監控時間，CloudFormation 仍會在堆疊建立和更新操作期間監控復原觸發條件；如有警示進入 ALARM 狀態，將會復原操作。接著，對於未發生警示的更新操作，將會在操作完成後立即開始處置舊資源。

在預設情況下，如果警示進入 ALARM 狀態 (而非 INSUFFICIENT_DATA 狀態)，CloudFormation 只會復原堆疊操作。若要在警示進入 INSUFFICIENT_DATA 狀態時，也讓 CloudFormation 復原堆疊操作，請編輯 CloudWatch 警示，將缺少資料視為 breaching。如需詳細資訊，請參閱「Amazon CloudWatch 使用者指南」中的[設定 CloudWatch 警示如何處理遺失資料](#)。

CloudFormation 在更新操作期間復原堆疊時，不會監控復原觸發條件。

您最多可以新增 5 個復原觸發條件。若要新增復原觸發條件，請指定 CloudWatch 警示的 Amazon Resource Name (ARN)。目前，AWS::CloudWatch::Alarm 和 AWS::CloudWatch::CompositeAlarm 類型可用作復原觸發條件。如需有關 CloudWatch 警示的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的[使用 Amazon CloudWatch 警示](#)。

如果找不到指定的 Cloudwatch 警示，整個堆疊操作將會失敗並復原。

請注意，存取 CloudWatch 需要憑證。這些登入資料必須具有存取 AWS 資源的許可，例如擷取有關資源的 CloudWatch 指標資料。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南中的 [CloudWatch 身分驗證和存取控制](#)。

新增堆疊建立或更新期間的復原觸發條件

新增堆疊建立或更新期間的復原觸發條件 (主控台)

1. 在建立或更新堆疊期間，在設定堆疊選項頁面的進階選項下，展開復原組態區段。
2. 請輸入介於 0 – 180 分鐘的監控時間。預設值為 0。
3. 指定您想要用作復原觸發條件的 Cloudwatch 警示或複合警示 ARN，然後選擇新增 CloudWatch 警示 ARN。

例如，以下是 CloudWatch 警示或複合警示的 ARN：`arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`。

4. 選擇下一步 並檢閱堆疊詳細資訊。
5. 準備就緒時，請選擇提交，以建立或更新堆疊。

新增堆疊建立或更新期間的復原觸發條件 (AWS CLI)

使用 [create-stack](#) 或 [update-stack](#) 命令搭配 `--rollback-configuration` 選項。

例如，下列 `update-stack` 命令會將 `MyCompositeAlarm` 設定為復原觸發條件，監控時間為 5 分鐘：

```
aws cloudformation update-stack --stack-name MyStack \  
  --use-previous-template \  
  --rollback-configuration \  
    "RollbackTriggers=[{Arn=arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyCompositeAlarm,Type=AWS::CloudWatch::CompositeAlarm}],MonitoringTime=5"
```

將復原觸發條件新增至變更集

將復原觸發條件新增至變更集 (主控台)

1. 在建立變更集時，在設定堆疊選項頁面的進階選項下，展開復原組態區段。
2. 請輸入介於 0 – 180 分鐘的監控時間。預設值為 5。

3. 指定您想要用作復原觸發條件的 Cloudwatch 警示或複合警示 ARN，然後選擇新增 CloudWatch 警示 ARN。

例如，以下是 CloudWatch 警示或複合警示的 ARN：`arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName`。

4. 選擇下一步，並檢閱變更集的詳細資訊。
5. 準備就緒時，請選擇建立變更集，以建立變更集。

將復原觸發條件新增至變更集 (AWS CLI)

使用 [create-change-set](#) 命令搭配 `--rollback-configuration` 選項。

檢視堆疊的復原觸發條件

若要檢視堆疊的復原觸發條件，請參閱 Rollback configuration (復原組態) 區段。

1. 在堆疊頁面上，從左側清單選擇您希望檢視的堆疊。
2. 在堆疊資訊索引標籤上，展開復原組態區段以檢視復原觸發條件。

檢視變更集的復原觸發條件

若要檢視變更集的復原觸發條件，請參閱復原組態區段。

1. 在堆疊頁面上，從左側清單選擇您希望檢視的堆疊。
2. 選擇變更集索引標籤，然後選擇要檢視的變更集。
3. 選擇輸入索引標籤，然後檢視復原組態區段。

透過漂移偵測功能，偵測堆疊和資源的未受管組態變更

正當您透過 CloudFormation 管理資源時，使用者可以在 CloudFormation 的外部變更這些資源。使用者可以使用建立資源的基礎服務來直接編輯資源。例如，您可以使用 Amazon EC2 主控台來更新已建立成為 CloudFormation 堆疊一部分的伺服器執行個體。有些變更可能是意外，有些則是為了回應時間急迫性運作事件而刻意為之。無論如何，在 CloudFormation 外部所做的變更會使堆疊更新或刪除操作變得複雜。您可以使用漂移偵測來識別已在 CloudFormation 管理之外發生組態變更的堆疊資源。然後，您可以採取矯正動作，讓堆疊資源與堆疊範本中的定義再次同步，例如直接更新偏離的資源，以符合其範本定義。解決偏離有助於確保組態一致性和成功的堆疊操作。

主題

- [什麼是偏離？](#)
- [漂移偵測狀態碼](#)
- [偵測偏離時的考量](#)
- [在整個 CloudFormation 堆疊上偵測偏離](#)
- [在個別堆疊資源上偵測偏離](#)
- [透過匯入作業解決漂移](#)

什麼是偏離？

漂移偵測可讓您偵測堆疊的實際組態是否不同於 (或已漂移) 其預期組態。使用 CloudFormation 在整個堆疊或堆疊內的個別資源上偵測偏離。如果資源的任何實際屬性值不同於預期的屬性值，資源就視為已偏離。這包括是否已刪除屬性或資源。如果堆疊的一個或多個資源已偏離，堆疊就視為已偏離。

為了判斷資源是否已漂移，CloudFormation 會決定堆疊範本中定義的預期資源屬性值，以及指定為範本參數的任何值。CloudFormation 接著會比較目前存在於堆疊中的資源屬性的預期值與實際值。如果資源的一個或多個屬性已遭刪除，或值已變更，資源就視為已偏離。

CloudFormation 會針對堆疊中已偏離的每個資源產生詳細資訊。

CloudFormation 會偵測那些支援偏離偵測 AWS 的資源上的偏離。不支援資源漂移偵測的資源會指派為 NOT_CHECKED 的漂移狀態。如需支援偏離偵測 AWS 的資源清單，請參閱 [資源類型支援](#)。

此外，CloudFormation 還支援可佈建之私有資源類型的漂移偵測；也就是說，佈建類型為 FULLY_MUTABLE 或 IMMUTABLE。若要對私人資源類型的資源執行漂移偵測，您在帳號中註冊之資源類型的預設版本必須是可佈建的資源。如需有關資源佈建類型的詳細資訊，請參閱《AWS CloudFormation API 參考》中 [DescribeType](#) 動作的 ProvisioningType，與《AWS CLI 命令參考》中 [DescribeType](#) 命令的該參數。如需私有資源的詳細資訊，請參閱 [使用 CloudFormation 登錄檔管理擴充功能](#)。

您可以在處於下列狀態的堆疊上執行漂移偵

測：CREATE_COMPLETE、UPDATE_COMPLETE、UPDATE_ROLLBACK_COMPLETE 和 UPDATE_ROLLBACK_FAILED。

在堆疊上偵測偏離時，CloudFormation 不會在任何屬於該堆疊的巢狀堆疊上偵測偏離。如需詳細資訊，請參閱[運用巢狀堆疊，將範本分割成可重複使用的部分](#)。反而是，您可以直接在巢狀堆疊上啟動漂移偵測操作。

 Note

CloudFormation 只會針對透過堆疊範本或經由指定範本參數而明確設定的屬性值，判斷是否偏離。這不包含資源屬性的預設值。如果是為了判斷偏離而讓 CloudFormation 追蹤資源屬性，請明確設定屬性值 (即使設為預設值)。

漂移偵測狀態碼

本節中的表格描述用於漂移偵測的各種狀態類型：

- 漂移偵測操作狀態描述漂移操作的目前狀態。
- Drift status (偏離狀態)

針對「堆疊集合」，這會描述整個堆疊集合的偏離狀態，而此狀態是以屬於該堆疊集合堆疊執行個體的偏離狀態為基礎。

針對「堆疊執行個體」，這會描述堆疊執行個體的偏離狀態，而此狀態是以其相關聯堆疊的偏離狀態為基礎。

針對「堆疊」，這會描述整個堆疊的偏離狀態，而此狀態是以其資源的偏離狀態為基礎。

- 資源偏離狀態描述個別資源的偏離狀態。

下表列出 CloudFormation 指派給堆疊漂移偵測操作的狀態代碼。

漂移偵測操作狀態	Description
DETECTION_COMPLETE	對於堆疊中支援漂移偵測的所有資源，堆疊漂移偵測操作已成功完成。
DETECTION_FAILED	堆疊中至少一個資源的堆疊漂移偵測操作已失敗。Cloud Formation 成功完成漂移偵測的資源會有結果。
DETECTION_IN_PROGRESS	堆疊漂移偵測操作目前進行中。

下表列出 CloudFormation 指派給堆疊的偏離狀態代碼。

Drift status (偏離狀態)	Description
DRIFTED	針對堆疊：堆疊不同於或已「偏離」其預期範本組態。如果堆疊的一個或多個資源已偏離，堆疊就視為已偏離。 針對堆疊執行個體：如果與其相關聯的堆疊已發生偏離，則會將堆疊執行個體視為已偏離。 針對堆疊集合：如果有一或多個堆疊執行個體已發生偏離，則會將堆疊集合視為已偏離。
NOT_CHECKED	CloudFormation 尚未檢查堆疊、堆疊集合或堆疊執行個體是否與其預期範本組態不同。
IN_SYNC	每個受支援資源的目前組態符合其預期範本組態。沒有支援漂移偵測資源的堆疊、堆疊集合，或堆疊執行個體，其狀態也會是 IN_SYNC。

下表列出 CloudFormation 指派給堆疊資源的偏離狀態代碼。

資源偏離狀態	Description
DELETED	資源不同於其預期範本組態，因為已刪除資源。
MODIFIED	資源不同於其預期範本組態。
NOT_CHECKED	CloudFormation 未檢查資源是否不同於其預期範本組態。
IN_SYNC	資源的目前組態符合其預期範本組態。

針對不同於預期範本組態的資源屬性，下表列出 CloudFormation 指派這些資源屬性的差異類型狀態代碼。

屬性差異類型	Description
ADD	作為陣列或清單資料類型的資源屬性已新增一個值。

屬性差異類型	Description
REMOVE	已從目前的資源組態中移除屬性。
NOT_EQUAL	目前的屬性值不同於堆疊範本中定義的預期值。

偵測偏離時的考量

為了在堆疊上成功執行漂移偵測，使用者必須有以下許可：

- 對於堆疊包含的每個支援漂移偵測的資源，需要有讀取許可。例如，如果堆疊包含 `AWS::EC2::Instance` 資源，則您必須有 `ec2:DescribeInstances` 許可，才能在堆疊上執行漂移偵測。
- `cloudformation:DetectStackDrift`
- `cloudformation:DetectStackResourceDrift`
- `cloudformation:BatchDescribeTypeConfigurations`

如需有關 CloudFormation 中的設定許可的詳細資訊，請參閱 [使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

在某些極端案例中，CloudFormation 可能無法總是傳回準確的偏離結果。您應該注意這些極端案例，以適當解讀您的漂移偵測結果。

- 在某些情況下，屬性陣列包含的物件會回報為偏離，而事實上，它們是由負責資源的基礎服務提供給屬性的預設值。
- 某些資源與相關資源之間存在附加關係，以致於一個資源實際上可能連接或移除其他資源 (在同一個或另一個範本中定義) 的屬性值。例如，`AWS::EC2::SecurityGroupIngress` 和 `AWS::EC2::SecurityGroupEgress` 資源可用來連接和移除 `AWS::EC2::SecurityGroup` 資源的值。在這些情況下，CloudFormation 在執行偏離比較之前會分析堆疊範本找出附加項。不過，CloudFormation 無法跨堆疊執行此分析，因此當附加的資源位於不同的堆疊時，可能不會傳回準確的偏離結果。

支援漂移偵測且允許或需要從其他資源附加的資源包括：

Resource Type (資源類型)	附加資源類型
AWS::SNS::Topic	AWS::SNS::Subscription
AWS::IAM::User	AWS::IAM::UserToGroupAddition
AWS::IAM::Group	AWS::IAM::ManagedPolicy
AWS::IAM::Role	AWS::IAM::Policy
AWS::IAM::User	AWS::IAM::RolePolicy
AWS::ElasticLoadBalancingV2::Listener	AWS::ElasticLoadBalancingV2::ListenerCertificate
AWS::EC2::SecurityGroup	AWS::EC2::SecurityGroupEgress AWS::EC2::SecurityGroupIngress

- CloudFormation 不會在任何資源的 `KMSKeyId` 屬性上執行漂移偵測。由於多個別名可以參考 AWS KMS 金鑰，CloudFormation 無法保證此屬性的一致準確偏離結果。
- 您在堆疊範本中可指定的某些資源屬性，在本質上，無法供 CloudFormation 與產生的堆疊資源中的屬性相互比較。因此，這些屬性無法納入漂移偵測結果中。這些屬性分為兩大類：
 - CloudFormation 無法映射的屬性值 (無法往回映射到它們在堆疊範本中的初始資源屬性值)。

例如，CloudFormation 無法將 Lambda 函式的原始程式碼往回映射到 [AWS::Lambda::Function](#) 資源的 `Code` 屬性類型，因此，CloudFormation 無法將它納入漂移偵測結果中。

- 負責資源的服務不會傳回的屬性值。

某些屬性值原本就不可能由資源所屬的服務傳回。這些通常包含機密資訊，例如，不應公開的密碼或其他敏感資料。例如，IAM 服務絕不會傳回 [AWS::IAM::User LoginProfile](#) 屬性類型的 `Password` 屬性值，因此，CloudFormation 無法將它納入漂移偵測結果中。

- 陣列中的物件實際上可能是服務預設，而不是手動新增的漂移。
- 如果您發現任何誤報，請使用 CloudFormation 主控台的意見回饋連結來傳送您的意見，或透過 [AWS re:Post](#) 聯絡我們。
- 部分屬性的輸入值可能會相等但不相同。為避免誤報，您應確保預期組態與實際組態相符。

- 例如，資源屬性的預期組態可能是 1024 MB，而相同資源屬性的實際組態可能是 1 GB。1024 MB 和 1 GB 相等，但不相同。

如果在此資源屬性上執行漂移偵測，漂移偵測會發出漂移結果訊號。

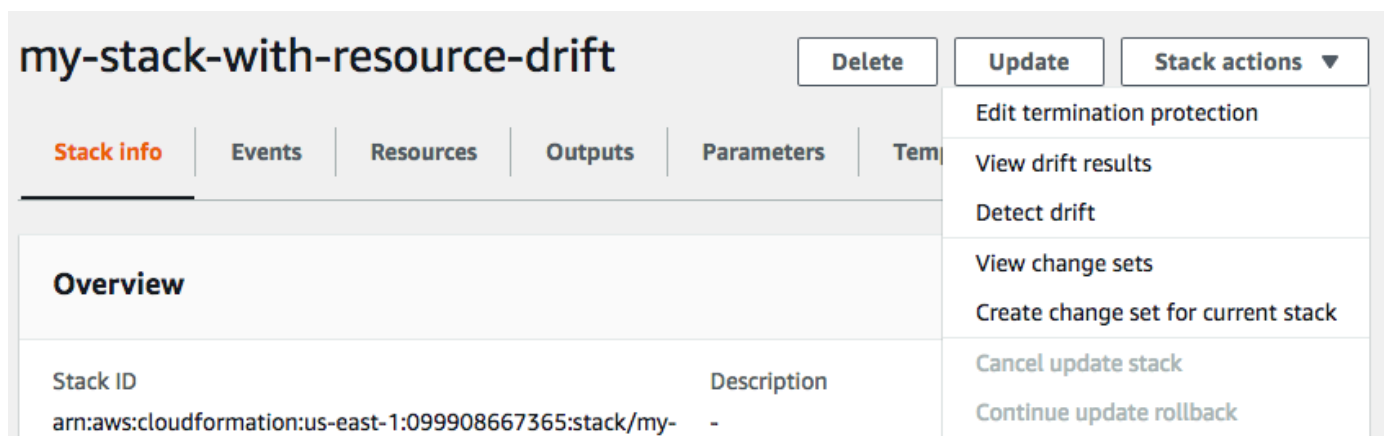
若要避免此誤報，請將資源屬性的預期組態變更為 1024 MB，然後執行漂移偵測。

在整個 CloudFormation 堆疊上偵測偏離

在堆疊上執行漂移偵測操作會判斷堆疊是否已漂移其預期範本組態，然後針對堆疊中每個支援漂移偵測的資源，傳回其漂移狀態的詳細資訊。

使用 偵測整個堆疊上的偏離 AWS 管理主控台

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從堆疊清單中選取要執行漂移偵測的堆疊。在 stack details (堆疊詳細資訊) 窗格中，選擇 Stack actions (堆疊動作)，然後選擇 Detect drift (偵測偏移)。



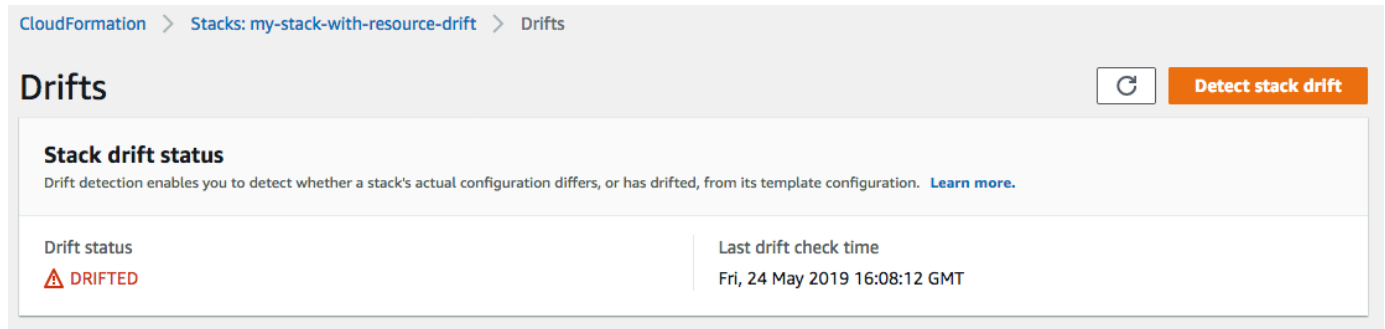
CloudFormation 會顯示資訊列，說明已啟動所選堆疊的漂移偵測。

3. 等待 CloudFormation 完成漂移偵測操作。當漂移偵測操作完成後，CloudFormation 會更新堆疊的 Drift status (漂移狀態) 和 Last drift check time (上次檢查漂移的時間)。這些欄位會列在堆疊詳細資訊頁面之 Stack info (堆疊資訊) 窗格的 Overview (概觀) 區段。

漂移偵測操作可能需要幾分鐘的時間，這取決於堆疊包含的資源數量。您在任一堆疊上同時間只能執行單一漂移偵測操作。即使關閉資訊列，CloudFormation 仍會繼續執行漂移偵測操作。

4. 檢閱堆疊及其資源的漂移偵測結果。在選取堆疊的情況下，從 Stack actions (堆疊動作) 選單中選取 View drift results (檢視偏移結果)。

CloudFormation 會列出堆疊的整體漂移狀態，以及上次在堆疊或其任何個別資源上啟動漂移偵測的時間。如果堆疊的一個或多個資源已偏離，堆疊就視為已偏離。



在 Resource drift status (資源漂移狀態) 區段中，CloudFormation 會列出每個堆疊資源、其漂移狀態，以及上次在資源上啟動漂移偵測的時間。每個資源會顯示邏輯 ID 和實體 ID，以協助您識別它們。此外，對於狀態為 MODIFIED 的資源，CloudFormation 會顯示資源偏離詳細資訊。

您可以根據資源的偏移狀態，使用 Drift status (偏離狀態) 欄排序資源。

- 檢視已修改過的資源的詳細資訊。
- 在選取已修改資源的情況下，選取 View drift details (檢視偏離詳細資訊)。

CloudFormation 會顯示該資源的偏離詳細資訊頁面。此頁面會顯示資源的預期和目前屬性值，以及兩者間的任何差異。

若要強調差異，請在 Differences (差異) 區段選取屬性名稱。

- Details (詳細資訊) 區段的 Current (目前) 欄位會以綠色反白顯示新增的屬性。
- Details (詳細資訊) 區段的 Expected (預期) 欄位會以紅色反白顯示刪除的屬性。
- 值已變更的屬性在 Expected (預期) 和 Current (目前) 欄中以黃色醒目提示。

Differences (4)					< 1 >
☒	Property	Change	Expected value	Current value	
☒	AttributeDefinitions.1	ADD	-	{ "AttributeName": "stuff", "AttributeType": "S" }	
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6	
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4	
☒	Tags	REMOVE	[{ "Key": "test", "Value": "test" }]	-	

Details

Expected

Actual

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}
```

```
{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}
```

使用 偵測整個堆疊上的偏離 AWS CLI

 Important

檢閱堆疊的上次偏離檢查時間，並確認其早於資源偏離結果中顯示的時間戳記，以防止使用過時資料。

若要使用 偵測整個堆疊上的偏離 AWS CLI，請使用下列 AWS CLI 命令：

- `detect-stack-drift`：在堆疊上啟動漂移偵測操作。
- `describe-stack-drift-detection-status`：監控堆疊漂移偵測操作的狀態。
- `describe-stack-resource-drifts`：檢閱堆疊漂移偵測操作的詳細資訊。

1. 使用 `detect-stack-drift` 在整個堆疊上偵測偏離。指定堆疊名稱或 ARN。您也可以指定任何特定資源的邏輯 ID，作為此漂移偵測操作的篩選條件。

```
aws cloudformation detect-stack-drift --stack-name my-stack-with-resource-drift
```

輸出：

```
{
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample"
}
```

2. 由於堆疊漂移偵測操作可能會長時間執行，請使用 `describe-stack-drift-detection-status` 監控漂移操作的狀態。此命令接受 `detect-stack-drift` 命令傳回的堆疊漂移偵測 ID。

在以下範例中，我們接受上述 `detect-stack-drift` 範例所傳回的堆疊漂移偵測 ID，並當作參數傳遞到 `describe-stack-drift-detection-status`。此參數傳回操作詳細資訊，其中指出漂移偵測操作已完成、單一堆疊資源已漂移，也因此整個堆疊視為已漂移。

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

輸出：

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
  "StackDriftDetectionId": "624af370-311a-11e8-b6b7-500cexample",
  "StackDriftStatus": "DRIFTED",
  "Timestamp": "2018-03-26T17:23:22.279Z",
  "DetectionStatus": "DETECTION_COMPLETE",
  "DriftedStackResourceCount": 1
}
```

3. 當堆疊漂移偵測操作完成後，請使用 `describe-stack-resource-drifts` 命令來檢視結果，包括已漂移的資源的實際和預期屬性值。

以下範例使用 `--stack-resource-drift-status-filters` 選項，以請求已修改或已刪除的資源的堆疊偏離資訊。該請求傳回已修改的一項資源的相關資訊，包括值已變更的兩個屬性的詳細資訊。未刪除任何資源。

```
aws cloudformation describe-stack-resource-drifts --stack-name my-stack-with-resource-drift --stack-resource-drift-status-filters MODIFIED DELETED
```

輸出：

```
{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ActualProperties": "{\n\"ReceiveMessageWaitTimeSeconds\":0,\n\"DelaySeconds\":120,\n\"RedrivePolicy\":{\n\"deadLetterTargetArn\":\n\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",\n\"maxReceiveCount\":12},\n\"MessageRetentionPeriod\":345600,\n\"MaximumMessageSize\":262144,\n\"VisibilityTimeout\":60,\n\"QueueName\":\n\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}",
      "ResourceType": "AWS::SQS::Queue",
      "Timestamp": "2018-03-26T17:23:34.489Z",
      "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-Queue-494PBHC076H4",
      "StackResourceDriftStatus": "MODIFIED",
      "ExpectedProperties": "{\n\"ReceiveMessageWaitTimeSeconds\":0,\n\"DelaySeconds\":20,\n\"RedrivePolicy\":{\n\"deadLetterTargetArn\":\n\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\",\n\"maxReceiveCount\":10},\n\"MessageRetentionPeriod\":345600,\n\"MaximumMessageSize\":262144,\n\"VisibilityTimeout\":60,\n\"QueueName\":\n\"my-stack-with-resource-drift-Queue-494PBHC076H4\"}",
      "PropertyDifferences": [
        {
          "PropertyPath": "/DelaySeconds",
          "ActualValue": "120",
          "ExpectedValue": "20",
          "DifferenceType": "NOT_EQUAL"
        },
        {
          "PropertyPath": "/RedrivePolicy/maxReceiveCount",
```

```
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
      }
    ],
    "LogicalResourceId": "Queue"
  }
}
```

在個別堆疊資源上偵測偏離

您可以在堆疊內的特定資源上偵測偏離，而非整個堆疊。當您只需要判斷特定資源現在是否再次符合其預期範本組態時，這特別有用。

在資源上執行漂移偵測時，CloudFormation 也會更新整體堆疊漂移狀態和 Last drift check time (上次漂移檢查時間) (如適用)。例如，假設堆疊的偏離狀態為 IN_SYNC。您讓 CloudFormation 在該堆疊包含的一個或多個資源上執行漂移偵測，CloudFormation 會偵測到其中一個或多個資源已漂移。CloudFormation 將堆疊漂移狀態更新為 DRIFTED。反之，假設堆疊因為單一偏離的資源而處於偏離狀態 DRIFTED。如果您將該資源設回其預期的屬性值，然後在資源再次偵測偏離，CloudFormation 就會將資源偏離狀態和堆疊偏離狀態都更新為 IN_SYNC，您也就不需要在整個堆疊上再次偵測偏離。

使用 偵測個別資源上的偏離 AWS 管理主控台

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從堆疊清單中，選取包含資源的堆疊。CloudFormation 會顯示該堆疊的堆疊詳細資訊。
3. 在左側瀏覽窗格的Stacks (堆疊) 下，選擇 Stack actions (堆疊動作)，然後選擇 Detect drift (偵測偏離)。
4. 在 Resource drift status (資源偏離狀態) 下，選擇資源，然後選取Detect drift for resource (偵測資源的偏離)。

CloudFormation 在所選資源上執行漂移偵測。如果成功，CloudFormation 會更新資源的漂移狀態，以及整體堆疊漂移狀態 (如有必要)。CloudFormation 還會更新上次在資源和整個堆疊上執行漂移偵測的時間戳記。如果已修改資源，CloudFormation 會顯示資源的預期和目前屬性值的詳細偏離資訊。

5. 檢閱資源的漂移偵測結果。
 - 檢視已修改過的資源的詳細資訊。

- 在選取已修改資源的情況下，選取 View drift details (檢視偏離詳細資訊)。

CloudFormation 會顯示該資源的偏離詳細資訊，包括資源的預期和目前屬性值，以及兩者間的任何差異。

若要強調差異，請在 Differences (差異) 區段選取屬性名稱。

- Details (詳細資訊) 區段的 Current (目前) 欄位會以綠色反白顯示新增的屬性。
- Details (詳細資訊) 區段的 Expected (預期) 欄位會以紅色反白顯示刪除的屬性。
- 值已變更的屬性在 Expected (預期) 和 Current (目前) 欄中以黃色醒目提示。

Differences (4)
< 1 >

☒	Property	Change	Expected value	Current value
☒	AttributeDefinitions.1	ADD	-	{"AttributeName": "stuff", "AttributeType": "S"}
☒	ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	6
☒	ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	5	4
☒	Tags	REMOVE	[{"Key": "test", "Value": "test"}]	-

Details

Expected

```

{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 5,
    "WriteCapacityUnits": 5
  },
  "Tags": [
    {
      "Key": "test",
      "Value": "test"
    }
  ]
}

```

Actual

```

{
  "AttributeDefinitions": [
    {
      "AttributeName": "Name",
      "AttributeType": "S"
    },
    {
      "AttributeName": "stuff",
      "AttributeType": "S"
    }
  ],
  "KeySchema": [
    {
      "AttributeName": "Name",
      "KeyType": "HASH"
    }
  ],
  "ProvisionedThroughput": {
    "ReadCapacityUnits": 6,
    "WriteCapacityUnits": 4
  }
}

```

使用 偵測個別資源上的偏離 AWS CLI

Important

檢閱堆疊資源的上次偏離檢查時間，並確認其早於資源偏離結果中顯示的時間戳記，以防止使用過時資料。

若要使用 偵測個別資源上的偏離 AWS CLI，請使用 `detect-stack-resource-drift` 命令。指定資源的邏輯 ID，以及資源所在的堆疊。

以下範例在特定堆疊資源 `my-drifted-resource` 上執行偏離偵測操作。回應傳回的資訊可確認資源已修改，包括值已變更的兩個屬性的詳細資訊。

```
aws cloudformation detect-stack-resource-drift \
  --stack-name my-stack-with-resource-drift \
  --logical-resource-id my-drifted-resource
```

輸出：

```
{
  "StackResourceDrift": {
    "StackId": "arn:aws:cloudformation:us-east-1:099908667365:stack/my-stack-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
    "ActualProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds\":120,\"RedrivePolicy\":{\"\"deadLetterTargetArn\":\":\\\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\\\",\\\"maxReceiveCount\\\":12},\\\"MessageRetentionPeriod\\\":345600,\"MaximumMessageSize\":262144,\"VisibilityTimeout\":60,\"QueueName\":\\\"my-stack-with-resource-drift-Queue-494PBHC076H4\\\"}\",
    "ResourceType": "AWS::SQS::Queue",
    "Timestamp": "2018-03-26T18:54:28.462Z",
    "PhysicalResourceId": "https://sqs.us-east-1.amazonaws.com/099908667365/my-stack-with-resource-drift-Queue-494PBHC076H4",
    "StackResourceDriftStatus": "MODIFIED",
    "ExpectedProperties": "{\"ReceiveMessageWaitTimeSeconds\":0,\"DelaySeconds\":20,\"RedrivePolicy\":{\"\"deadLetterTargetArn\":\":\\\"arn:aws:sqs:us-east-1:099908667365:my-stack-with-resource-drift-DLQ-1BCY7HHD5QIM3\\\",\\\"maxReceiveCount\\\":10},\\\"MessageRetentionPeriod\\\":345600,\"MaximumMessageSize\":262144,\"VisibilityTimeout\":60,\"QueueName\":\\\"my-stack-with-resource-drift-Queue-494PBHC076H4\\\"}\",
```

```
    "PropertyDifferences": [
      {
        "PropertyPath": "/DelaySeconds",
        "ActualValue": "120",
        "ExpectedValue": "20",
        "DifferenceType": "NOT_EQUAL"
      },
      {
        "PropertyPath": "/RedrivePolicy/maxReceiveCount",
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
      }
    ],
    "LogicalResourceId": "my-drifted-resource"
  }
}
```

透過匯入作業解決漂移

在某些情況下，資源的組態已經從其預期組態中漂移，並且您想要接受新的組態作為預期組態。在大多數情況下，您會透過使用新組態更新堆疊範本中的資源來定義，然後執行堆疊更新來解決漂移結果。不過，如果新組態更新需要取代的資源屬性，則會在堆疊更新期間重新建立資源。如果您想要保留現有資源，可以使用資源匯入功能來更新資源並解決漂移結果，而不會導致資源被取代。

透過匯入作業解決資源的漂移包含下列基本步驟：

- [將 `DeletionPolicy` 屬性 \(設為「保留」\) 新增至資源](#)。這可確保現有資源在從堆疊中移除時保留，而不是刪除它。
- [從範本中移除資源，並執行堆疊更新作業](#)。這會從堆疊中移除資源，但不會刪除它。
- [在堆疊範本中描述資源的實際狀態，然後再將現有資源匯入堆疊](#)。這會將資源新增回堆疊，並解決造成漂移結果的屬性差異。

如需資源匯入的詳細資訊，請參閱 [手動將 AWS 資源匯入 CloudFormation 堆疊](#)。如需支援匯入的資源清單，請參閱 [資源類型支援](#)。

在此範例中，我們使用下列範本，名為 `templateToImport.json`。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}
```

```
    ],
    "BillingMode": "PROVISIONED",
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  GamesTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Games
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5

```

WriteCapacityUnits: 1

在此範例中，讓我們假設使用者變更了 CloudFormation 外部的資源。執行漂移檢測後，我們發現 GamesTable 已將 BillingMode 修改為 PAY_PER_REQUEST。如需關於漂移偵測的詳細資訊，請參閱 [透過漂移偵測功能，偵測堆疊和資源的未受管組態變更](#)。

Differences (3)				
Property	Change	Expected value	Current value	
BillingMode	NOT_EQUAL	PROVISIONED	PAY_PER_REQUEST	
ProvisionedThroughput.ReadCapacityUnits	NOT_EQUAL	5	0	
ProvisionedThroughput.WriteCapacityUnits	NOT_EQUAL	1	0	

Details	
Expected	Actual
<pre>{ "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PROVISIONED", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 5, "WriteCapacityUnits": 1 }, "TableName": "Games" }</pre>	<pre>{ "AttributeDefinitions": [{ "AttributeName": "key", "AttributeType": "S" }], "BillingMode": "PAY_PER_REQUEST", "KeySchema": [{ "AttributeName": "key", "KeyType": "HASH" }], "ProvisionedThroughput": { "ReadCapacityUnits": 0, "WriteCapacityUnits": 0 }, "TableName": "Games" }</pre>

我們的堆疊現已過時，而我們的資源卻是即時的，但我們還是希望能保留預期的資源組態。我們可以透過導入操作解決漂移而不中斷服務來做到這一點。

使用 CloudFormation 主控台透過匯入作業解決漂移

步驟 1. 使用保留刪除政策更新堆疊

使用具有 **Retain** 選項的 **DeletionPolicy** 屬性來更新堆疊

- 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
- 在堆疊頁面上，選擇已漂移的堆疊。
- 選擇更新，然後從堆疊詳細資料窗格中選擇取代目前的範本。
- 在指定範本頁面上，使用下列其中一種方法，透過 Retain 選項提供包含 DeletionPolicy 屬性的更新範本：
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。

然後選擇下一步。

- 複查指定堆疊詳細資料頁面，然後選擇下一步。

6. 檢閱配置堆疊選項頁面，並選擇下一步。
7. 在 檢閱 **stack-name** 頁面上，選擇更新堆疊。

結果：在堆疊的事件頁面上，狀態為 UPDATE_COMPLETE。

若要解決匯入作業的漂移，而不會中斷服務，請為您要從堆疊中移除的資源指定 Retain [DeletionPolicy](#)。在下列範例中，我們已將 [DeletionPolicy](#) 屬性 (設為 Retain) 新增至 GamesTable 資源。

Example JSON

```
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
```

Example YAML

```
GamesTable:
  Type: AWS::DynamoDB::Table
  DeletionPolicy: Retain
  Properties:
    TableName: Games
```

步驟 2. 移除漂移的資源、相關參數和輸出

移除漂移資源、相關參數和輸出

1. 選擇更新，然後從堆疊詳細資料窗格中選擇取代目前的範本。
2. 在指定範本頁面上，使用下列其中一種方法，提供已更新範本的資源、相關參數和輸出：
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。

然後選擇下一步。

3. 複查指定堆疊詳細資料頁面，然後選擇下一步。
4. 檢閱配置堆疊選項頁面，並選擇下一步。

5. 在檢閱 *stack-name* 頁面上，選擇更新堆疊。

結果：邏輯 ID GamesTable 在堆疊的事件頁面上的狀態為 DELETE_SKIPPED。

等待直到 CloudFormation 完成堆疊更新作業。堆疊更新作業完成後，請從堆疊範本移除資源、相關參數和輸出。然後，匯入更新的範本。完成這些動作之後，範例範本現在如下所示。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "BillingMode": "PROVISIONED",
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}
```

Example YAML

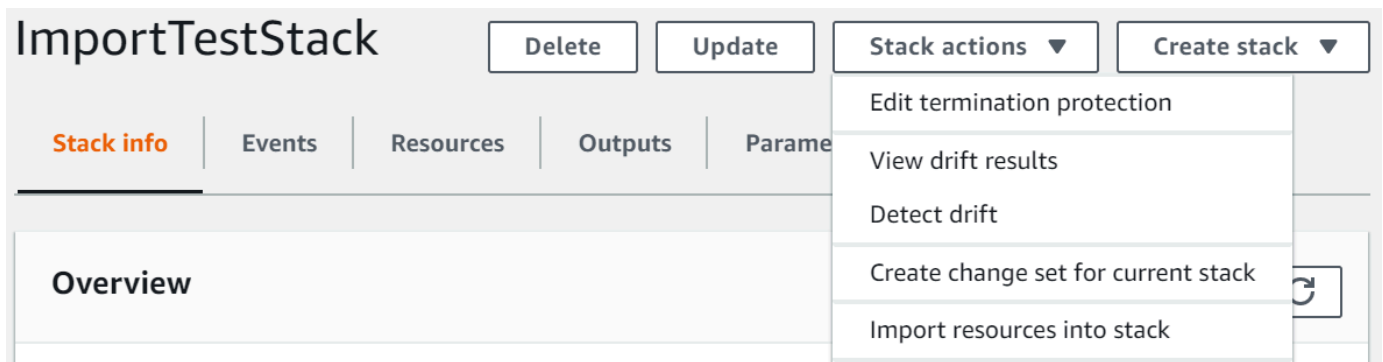
```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: Import test
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      BillingMode: PROVISIONED
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
```

步驟 3. 更新範本以符合資源的即時狀態

更新範本以符合資源的即時狀態

1. 若要匯入更新的範本，請選擇堆疊動作，然後選擇將資源匯入堆疊。



2. 請閱讀匯入概觀頁面，以取得您在此操作期間必須提供的項目清單，接著選擇下一步。
3. 在指定範本頁面上，使用以下其中一種方法提供您更新後的範本。
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。

然後選擇下一步。

4. 在 Identify resources (識別資源) 頁面上，識別每個目標資源。如需詳細資訊，請參閱[資源識別碼](#)。
 - a. 在 Identifier property (識別碼屬性) 下方，選擇資源識別碼類型。例如，TableName 屬性會識別 AWS::DynamoDB::Table 資源。
 - b. 在識別碼值下方，輸入實際的屬性值。在範例範本中，用於 GamesTable 資源的 TableName 是 Games。
 - c. 選擇下一步。
5. 檢閱指定堆疊詳細資料頁面，然後選擇下一步。
6. 在匯入概觀頁面上，檢閱要匯入的資源，然後選擇匯入資源。這會將 AWS::DynamoDB::Table 資源類型導回堆疊中。

結果：在此範例中，我們透過匯入作業解決資源漂移，而不會中斷服務。您可以在 Events (事件) 索引標籤的 CloudFormation 主控台中檢查匯入動作的進度。匯入的資源會有一個 IMPORT_COMPLETE 狀態，後接資源匯入完成的 CREATE_COMPLETE 狀態作為狀態原因。

等待直到 CloudFormation 完成堆疊更新作業。堆疊更新作業完成後，請更新範本，以符合資源的實際漂移狀態。例如，BillingMode 將設為 PAY_PER_REQUEST 和 ReadCapacityUnits, WriteCapacityUnits 將設為 0。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
```

```

        "KeyType": "HASH"
      }
    ],
    "BillingMode": "PROVISIONED",
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
},
"GamesTable": {
  "Type": "AWS::DynamoDB::Table",
  "DeletionPolicy": "Retain",
  "Properties": {
    "TableName": "Games",
    "AttributeDefinitions": [
      {
        "AttributeName": "key",
        "AttributeType": "S"
      }
    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "BillingMode": "PAY_PER_REQUEST",
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 0,
      "WriteCapacityUnits": 0
    }
  }
}
}
}
}
}

```

Example YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Import test
Resources:
  ServiceTable:

```

```
Type: AWS::DynamoDB::Table
Properties:
  TableName: Service
  AttributeDefinitions:
    - AttributeName: key
      AttributeType: S
  KeySchema:
    - AttributeName: key
      KeyType: HASH
  BillingMode: PROVISIONED
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 1
GamesTable:
  Type: AWS::DynamoDB::Table
  DeletionPolicy: Retain
  Properties:
    TableName: Games
    AttributeDefinitions:
      - AttributeName: key
        AttributeType: S
    KeySchema:
      - AttributeName: key
        KeyType: HASH
    BillingMode: PAY_PER_REQUEST
    ProvisionedThroughput:
      ReadCapacityUnits: 0
      WriteCapacityUnits: 0
```

將 AWS 資源匯入 CloudFormation 堆疊

您可將現有資源匯入 CloudFormation 堆疊。若您希望開始使用 CloudFormation 管理在其外部建立的資源，且無需刪除並重新建立這些資源，此功能會非常實用。

CloudFormation 提供以下選項，可將現有資源匯入堆疊：

- [IaC 產生器](#)是一種可自動掃描現有資源，並根據其目前狀態產生 CloudFormation 範本的工具。隨後可使用此範本將這些資源匯入堆疊。
- [資源匯入](#)是一種手動程序，您需在 CloudFormation 範本中描述現有資源，然後將其匯入堆疊。此方法需要您在範本中手動指定資源屬性及組態。

- [自動匯入](#)是一種自動程序，您在 CloudFormation 範本中描述現有資源後，CloudFormation 會將自訂名稱匹配的資源匯入堆疊。
- [堆疊重構](#)是一項可簡化 CloudFormation 堆疊中的資源重組，同時仍保留現有資源屬性和資料的功能。透過堆疊重構，您可在堆疊之間移動資源、將單一堆疊拆分為較小的元件，或將多個堆疊合併為一個。

除了將現有資源納入 CloudFormation 管理外，資源匯入功能在下列場景中也十分實用：

- 在堆疊之間移動資源 – 您可以將資源從一個堆疊匯入另一個，以便根據需要重組基礎設施。
- 巢狀現有堆疊 – 您可以將現有堆疊做為巢狀堆疊，匯入另一個堆疊，實現模組化且可重複使用的基礎設施設計。

CloudFormation 支援匯入各種類型的資源。如需詳細資訊，請參閱[資源類型支援](#)。

主題

- [手動將 AWS 資源匯入 CloudFormation 堆疊](#)
- [自動將 AWS 資源匯入 CloudFormation 堆疊](#)
- [還原匯入操作](#)

手動將 AWS 資源匯入 CloudFormation 堆疊

透過資源匯入，您可以將現有 AWS 資源匯入新的或現有的 CloudFormation 堆疊。在匯入操作期間，您會建立一個變更集合，將您的現有資源匯入堆疊，或是從您現有的資源建立新堆疊。您可以在匯入期間提供以下資訊。

- 描述整個堆疊的範本，包括原始堆疊資源和您正在匯入的資源。每個要匯入的資源都必須具備[DeletionPolicy 屬性](#)。
- 您要匯入的 CloudFormation 可用於映射範本中邏輯 ID 與現有資源的資源識別碼。

Note

CloudFormation 只支援使用資源匯入進行一個層級的巢狀。這表示您無法將堆疊匯入子堆疊，或是匯入具有子系的堆疊。

主題

- [資源識別碼](#)
- [資源匯入驗證](#)
- [資源匯入狀態碼](#)
- [匯入操作期間的考量](#)
- [其他資源](#)
- [從現有資源建立堆疊](#)
- [將現有資源匯入堆疊](#)
- [在堆疊之間移動資源](#)
- [巢狀現有堆疊](#)

資源識別碼

您可以提供兩個值來識別您匯入的每個資源。

- 識別碼屬性。這是可用來識別每個資源類型的資源屬性。例如，您可以使用 `AWS::S3::Bucket` 資源的 `BucketName` 來識別該資源。

您用於識別要匯入資源的資源屬性，會因資源類型而異。您可以在 CloudFormation 主控台中找到該資源屬性。建立包含要匯入資源的範本後，即可啟動匯入程序，並在其中找到要匯入資源的識別碼屬性。對於部分資源類型，可能有多種識別方式，您可在下拉清單中選取要使用的屬性。

或者，您可以呼叫 [get-template-summary](#) CLI 命令，並指定堆疊範本的 S3 URL 做為 `--template-url` 選項的值，以取得要匯入資源的識別碼屬性。

- 識別碼的值。這是資源的實際屬性值。例如，`BucketName` 屬性的實際值可能是 `MyS3Bucket`。

您可以從該資源的服務主控台中，取得識別碼屬性的數值。

資源匯入驗證

在匯入操作期間，CloudFormation 會執行下列驗證。

- 存在要匯入的資源。
- 每個要匯入資源的屬性和組態值都符合資源類型結構描述，該結構描述定義了其接受的屬性、必要屬性，以及支援的屬性值。

- 必要屬性會在範本中指定。每個資源類型的必要屬性都會在 [AWS 資源和屬性類型參考](#) 中說明。
- 要匯入的資源不屬於相同區域中的另一個堆疊。

CloudFormation 不會檢查範本組態是否與資源屬性的實際組態相符。

Important

驗證範本中定義的資源及其屬性是否符合資源匯入的預期組態，以避免發生未預期的變更。

資源匯入狀態碼

此表格說明搭配資源匯入功能使用的各種狀態類型。

匯入作業狀態	Description
IMPORT_IN_PROGRESS	匯入操作正在進行。
IMPORT_COMPLETE	已針對堆疊中的所有資源完成匯入操作。
IMPORT_ROLLBACK_IN_PROGRESS	復原匯入操作正在復原先前的範本組態。
IMPORT_ROLLBACK_FAILED	匯入復原操作失敗。
IMPORT_ROLLBACK_COMPLETE	匯入已復原先前的範本組態。

匯入操作期間的考量

- 在匯入完成且在執行後續的堆疊操作前，我們建議在匯入的資源上執行漂移偵測。漂移偵測可確保範本組態與實際組態相符。如需詳細資訊，請參閱[在整個 CloudFormation 堆疊上偵測偏離](#)。
- 匯入操作不允許建立新資源、刪除資源，或是對屬性組態進行變更。
- 要匯入的每個資源都必須具備 DeletionPolicy 屬性，匯入操作才能成功。DeletionPolicy 可設為任何可能值。僅有您要匯入的資源需要 DeletionPolicy。已經是堆疊一部分的資源不需要 DeletionPolicy。
- 您無法將相同的資源匯入多個堆疊。

- 您可以使用 `cloudformation:ImportResourceTypes` IAM 政策條件來控制使用者在匯入操作期間可使用的資源類型。如需詳細資訊，請參閱 [CloudFormation 的政策條件索引鍵](#)。
- 匯入資源時會套用 CloudFormation 堆疊限制。如需限制的詳細資訊，請參閱 [了解 CloudFormation 配額](#)。

其他資源

如需透過資源匯入解決堆疊漂移問題，請參閱 [透過匯入作業解決漂移](#)。

從現有資源建立堆疊

本主題說明如何透過在範本中描述現有 AWS 資源來建立堆疊。若要改為掃描現有資源並自動產生範本，以用於將現有資源匯入 CloudFormation 或在新帳戶中複製資源，請參閱 [使用 IaC 產生器從現有資源中產生範本](#)。

先決條件

開始之前，您必須準備好以下項目：

- 描述您在新堆疊中所需全部資源的範本。在本機或 Amazon S3 儲存貯體中儲存範本。
- 對於每個您要匯入的資源，需包含下列項目：
 - 定義資源目前組態的屬性與屬性數值。
 - 資源的唯一識別碼，如資源名稱。如需詳細資訊，請參閱 [資源識別碼](#)。
 - [DeletionPolicy](#) 屬性。

主題

- [範例 範本](#)
- [使用 從現有資源建立堆疊 AWS 管理主控台](#)
- [使用 從現有資源建立堆疊 AWS CLI](#)

範例 範本

在此逐步說明中，假設您使用下列名為 `TemplateToImport.json` 的範例範本，指定在 CloudFormation 外部建立的兩個 DynamoDB 資料表。ServiceTable 和 GamesTable 是匯入的目標。

 Note

此範本僅供範例參考。若要用於自身測試，請將範例資源替換為您帳戶中的資源。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ]
      }
    }
  }
}
```

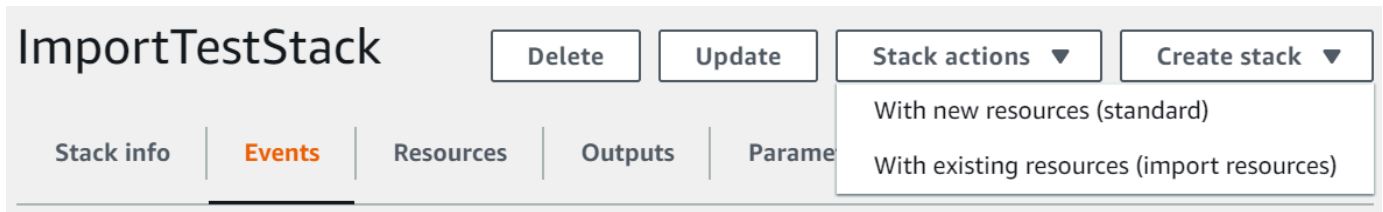
```

    ],
    "KeySchema": [
      {
        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  }
}
}
}

```

使用 從現有資源建立堆疊 AWS 管理主控台

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在 Stacks (堆疊) 頁面上，選擇 Create stack (建立堆疊)，然後選擇 With existing resources (import resources) (使用現有資源 (匯入資源))。



3. 請閱讀 Import overview (匯入概觀) 頁面，以取得您在此操作期間必須提供的項目清單。然後選擇下一步。
4. 在 Specify template (指定範本) 頁面上，使用以下其中一種方法提供您的範本，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。
5. 在 Identify resources (識別資源) 頁面上，識別每個目標資源。如需詳細資訊，請參閱 [資源識別碼](#)。
 - a. 在 Identifier property (識別碼屬性) 下方，選擇資源識別碼類型。例如，AWS::DynamoDB::Table 資源可以使用 TableName 屬性進行識別。

- b. 在 Identifier value (識別碼值) 下方，輸入實際的屬性值。例如，範例範本中 GamesTable 資源的 TableName 是 *Games*。
 - c. 選擇下一步。
6. 在 Specify stack details (識別堆疊詳細資訊) 頁面上，修改任何參數，然後選擇 Next (下一步)。這會自動建立變更集合。

Important

如果您修改了啟動建立、更新或刪除操作的現有參數，匯入操作便會失敗。

7. 在 Review *stack-name* (檢閱 stack-name) 頁面上，確認您正在匯入正確的資源，然後選擇 Import resources (匯入資源)。這會自動執行在最後一個步驟中建立的變更集合。

隨即會顯示新堆疊 Stack details (堆疊詳細資訊) 頁面的 Events (事件) 窗格。

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

8. (選用) 在 堆疊上執行漂移偵測，確認範本和匯入資源的實際組態相符。如需關於偵測漂移的詳細資訊，請參閱 [在整個 CloudFormation 堆疊上偵測偏離](#)。
9. (選用) 如果您匯入的資源與其預期的範本組態不相符，請修正範本組態，或是直接更新資源。在本演練中，我們會修正範本組態，使其與資源的實際組態相符。
 - a. 針對受影響的資源 [還原匯入操作](#)。
 - b. 再次將匯入目標新增至您的範本，確認範本組態與實際組態相符。
 - c. 使用修改後的範本重複步驟 2 到 8，再次匯入資源。

使用 從現有資源建立堆疊 AWS CLI

1. 若要了解範本中哪些屬性用於識別每種資源類型，請執行 `get-template-summary` 命令，並指定範本的 S3 網址。例如，`AWS::DynamoDB::Table` 資源可以使用 `TableName` 屬性進行識別。針對範例範本中的 `GamesTable` 資源，`TableName` 的值是 `Games`。在下一個步驟中，您將需要這項資訊。

```
aws cloudformation get-template-summary \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/  
TemplateToImport.json
```

如需詳細資訊，請參閱[資源識別碼](#)。

2. 以下列 JSON 字串格式編寫您範本中實際資源的清單，以及這些資源的唯一識別碼。

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}},  
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifier":{"TableName":"Service"}}]
```

您也可以組態檔案中指定 JSON 格式參數。

例如，若要匯入 `ServiceTable` 和 `GamesTable`，您可以建立名為 *ResourcesToImport.txt* 的檔案，並包含下列組態。

```
[  
  {  
    "ResourceType":"AWS::DynamoDB::Table",  
    "LogicalResourceId":"GamesTable",  
    "ResourceIdentifier":{  
      "TableName":"Games"  
    }  
  },  
  {  
    "ResourceType":"AWS::DynamoDB::Table",  
    "LogicalResourceId":"ServiceTable",  
    "ResourceIdentifier":{  
      "TableName":"Service"  
    }  
  }  
]
```

3. 要建立變更集，請使用下列 `create-change-set` 命令並取代預留位置文字。對於 `--change-set-type` 選項，請指定值為 **IMPORT**。對於 `--resources-to-import` 選項，請將範例 JSON 字串取代為您剛建立的實際 JSON 字串。

```
aws cloudformation create-change-set \  
  --stack-name TargetStack --change-set-name ImportChangeSet \  
  --change-set-type IMPORT \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/  
TemplateToImport.json \  
  --resources-to-  
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifi  
{"TableName":"Games"}},  
{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"ServiceTable","ResourceIdentifi  
{"TableName":"Service"}}]'
```

Note

`--resources-to-import` 不支援內嵌 YAML。JSON 字串中轉義引號的請求有所不同，這取決於您的終端。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[在字串內使用引號](#)。

或者，您可以如下列範例所示，使用檔案 URL 做為 `--resources-to-import` 選項的輸入。

```
--resources-to-import file:///ResourcesToImport.txt
```

4. 檢閱變更集合，確認您將匯入正確的資源。

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. 若要啟動變更集並匯入資源，請使用下列 `execute-change-set` 命令，並取代預留位置文字。成功完成操作 (IMPORT_COMPLETE) 後，資源便已順利匯入。

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (選用) 在 IMPORT_COMPLETE 堆疊上執行漂移偵測，確認範本和匯入資源的實際組態相符。如需關於偵測漂移的詳細資訊，請參閱[在個別堆疊資源上偵測偏離](#)。

- a. 在指定的堆疊上執行漂移偵測。

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

如果成功，此命令傳回的下列範例輸出。

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. 檢視指定堆疊漂移偵測 ID 的漂移偵測操作進度。

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. 檢視特定堆疊中已檢查是否漂移的資源的漂移資訊。

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (選用) 如果您匯入的資源與其預期的範本組態不相符，請修正範本組態，或是直接更新資源。在本演練中，我們會修正範本組態，使其與資源的實際組態相符。

- a. 針對受影響的資源[還原匯入操作](#)。
- b. 再次將匯入目標新增至您的範本，確認範本組態與實際組態相符。
- c. 使用修改後的範本重複步驟 3 – 6，以再次匯入資源。

將現有資源匯入堆疊

本主題說明如何透過在範本中描述現有資源，將現有 AWS 資源匯入現有堆疊。若要改為掃描現有資源並自動產生範本，以用於將現有資源匯入 CloudFormation 或在新帳戶中複製資源，請參閱 [使用 IaC 產生器從現有資源中產生範本](#)。

先決條件

開始之前，您必須準備好以下項目：

- 描述整個堆疊的範本，包括已是堆疊一部分的資源，以及要匯入的資源。在本機或 Amazon S3 儲存貯體中儲存範本。

取得執行中堆疊範本的副本

1. 開啟位在 <https://console.aws.amazon.com/cloudformation/> 的 CloudFormation 主控台。
 2. 從堆疊清單中，選擇要從中擷取範本的堆疊。
 3. 在堆疊詳細資訊窗格中，選擇範本索引標籤，再選擇複製到剪貼簿。
 4. 將程式碼貼到文字編輯器中，以開始將其他資源新增至範本。
- 對於每個您要匯入的資源，需包含下列項目：
 - 定義資源目前組態的屬性與屬性數值。
 - 資源的唯一識別碼，如資源名稱。如需詳細資訊，請參閱[資源識別碼](#)。
 - [DeletionPolicy](#) 屬性。

主題

- [範例 範本](#)
- [使用 將現有資源匯入堆疊 AWS 管理主控台](#)
- [使用 將現有資源匯入堆疊 AWS CLI](#)

範例 範本

在此逐步說明中，我們假設您使用下列名為 `TemplateToImport.json` 的範例範本，指定兩個 DynamoDB 資料表。GamesTable 目前是堆疊的一部分，而 ServiceTable 是您要匯入的資料表。

Note

此範本僅供範例參考。若要用於自身測試，請將範例資源替換為您帳戶中的資源。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ]
      }
    }
  }
}
```

```

        }
    ],
    "KeySchema": [
        {
            "AttributeName": "key",
            "KeyType": "HASH"
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
    }
},
"GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}
}
}
}
}

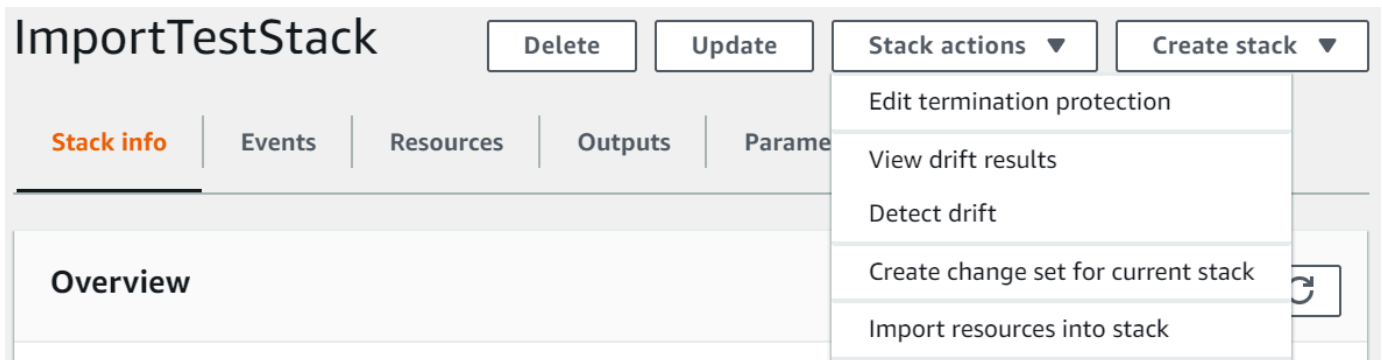
```

使用 將現有資源匯入堆疊 AWS 管理主控台

Note

CloudFormation 主控台不支援在匯入資源時使用內建函數 `Fn::Transform`。您可以使用 AWS CLI 匯入使用 `Fn::Transform` 函數的資源。

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在 Stacks (堆疊) 頁面上，選擇您要匯入資源的堆疊。
3. 選擇 Stack actions (堆疊動作)，然後選擇 Import resources into stack (將資源匯入堆疊)。



4. 檢閱 Import overview (匯入概觀) 頁面，然後選擇 Next (下一步)。
5. 在 Specify template (指定範本) 頁面上，使用以下其中一種方法提供您更新後的範本，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。
6. 在 Identify resources (識別資源) 頁面上，識別每個目標資源。如需詳細資訊，請參閱[資源識別碼](#)。
 - a. 在 Identifier property (識別碼屬性) 下方，選擇資源識別碼類型。例如，`AWS::DynamoDB::Table` 資源可以使用 `TableName` 屬性進行識別。
 - b. 在 Identifier value (識別碼值) 下方，輸入實際的屬性值。例如，範例範本中 `GamesTable` 資源的 `TableName` 是 `Games`。
 - c. 選擇下一步。
7. 在 Specify stack details (識別堆疊詳細資訊) 頁面上，更新任何參數，然後選擇 Next (下一步)。這會自動建立變更集合。

Note

如果您修改了啟動建立、更新或刪除操作的現有參數，匯入操作便會失敗。

- 在 Review **stack-name** (檢閱 stack-name) 頁面上，檢閱要匯入的資源，然後選擇 Import resources (匯入資源)。這會自動執行在最後一個步驟中建立的變更集合。此時會將任何堆疊層級標籤套用到匯入的資源。如需詳細資訊，請參閱[設定堆疊選項](#)。

隨即會顯示堆疊的 Events (事件) 頁面。

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

- (選用) 在 堆疊上執行漂移偵測，確認範本和匯入資源的實際組態相符。如需關於偵測漂移的詳細資訊，請參閱 [在整個 CloudFormation 堆疊上偵測偏離](#)。
- (選用) 如果您匯入的資源與其預期的範本組態不相符，請修正範本組態，或是直接更新資源。如需有關匯入漂移資源的詳細資訊，請參閱 [透過匯入作業解決漂移](#)。

使用 將現有資源匯入堆疊 AWS CLI

- 若要了解範本中哪些屬性用於識別每種資源類型，請執行 get-template-summary 命令，並指定範本的 S3 網址。例如，AWS::DynamoDB::Table 資源可以使用 TableName 屬性進行識別。針對範例範本中的 GamesTable 資源，TableName 的值是 Games。在下一個步驟中，您將需要這項資訊。

```
aws cloudformation get-template-summary \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
  TemplateToImport.json
```

如需詳細資訊，請參閱[資源識別碼](#)。

2. 以下列 JSON 字串格式編寫要匯入的實際資源清單，以及這些資源的唯一識別碼。

```
[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]
```

您也可以在此組態檔案中指定 JSON 格式參數。

例如，若要匯入 GamesTable，您可以建立名為 *ResourcesToImport.txt* 的檔案，並包含下列組態。

```
[
  {
    "ResourceType":"AWS::DynamoDB::Table",
    "LogicalResourceId":"GamesTable",
    "ResourceIdentifier": {
      "TableName":"Games"
    }
  }
]
```

3. 要建立變更集，請使用下列 create-change-set 命令並取代預留位置文字。對於 --change-set-type 選項，請指定值為 **IMPORT**。對於 --resources-to-import 選項，請將範例 JSON 字串取代為您剛建立的實際 JSON 字串。

```
aws cloudformation create-change-set \
  --stack-name TargetStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/
TemplateToImport.json \
  --resources-to-
import '[{"ResourceType":"AWS::DynamoDB::Table","LogicalResourceId":"GamesTable","ResourceIdentifier":{"TableName":"Games"}}]'
```

 Note

`--resources-to-import` 不支援內嵌 YAML。JSON 字串中轉義引號的請求有所不同，這取決於您的終端。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[在字串內使用引號](#)。

或者，您可以如下列範例所示，使用檔案 URL 做為 `--resources-to-import` 選項的輸入。

```
--resources-to-import file:///ResourcesToImport.txt
```

4. 檢閱變更集合，確認您將匯入正確的資源。

```
aws cloudformation describe-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

5. 若要啟動變更集並匯入資源，請使用下列 `execute-change-set` 命令，並取代預留位置文字。此時會將任何堆疊層級標籤套用到匯入的資源。如需詳細資訊，請參閱[設定堆疊選項](#)。成功完成操作 (IMPORT_COMPLETE) 後，資源便已順利匯入。

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStack
```

6. (選用) 在 IMPORT_COMPLETE 堆疊上執行漂移偵測，確認範本和匯入資源的實際組態相符。如需關於偵測漂移的詳細資訊，請參閱[在整個 CloudFormation 堆疊上偵測偏離](#)。

- a. 在指定的堆疊上執行漂移偵測。

```
aws cloudformation detect-stack-drift --stack-name TargetStack
```

如果成功，此命令傳回的下列範例輸出。

```
{ "Stack-Drift-Detection-Id" : "624af370-311a-11e8-b6b7-500cexample" }
```

- b. 檢視指定堆疊漂移偵測 ID 的漂移偵測操作進度。

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 624af370-311a-11e8-b6b7-500cexample
```

- c. 檢視特定堆疊中已檢查是否漂移的資源的漂移資訊。

```
aws cloudformation describe-stack-resource-drifts --stack-name TargetStack
```

7. (選用) 如果您匯入的資源與其預期的範本組態不相符，請修正範本組態，或是直接更新資源。如需有關匯入漂移資源的詳細資訊，請參閱 [透過匯入作業解決漂移](#)。

在堆疊之間移動資源

使用 `resource import` 功能，您可以在資源之間移動 (或是「重構」) 堆疊。您需要先將 Retain 刪除政策新增到您要移動的資源，以確保當您從來源堆疊移除該資源，並匯入到目標堆疊時，仍會保留該資源。

若您是初次匯入，建議您先檢閱 [將 AWS 資源匯入 CloudFormation 堆疊](#) 主題中的簡介資訊。

Important

並非所有資源都支援匯入操作。從堆疊移除資源之前，請參閱 [支援匯入操作的資源](#)。如果您從堆疊移除不支援匯入操作的資源，則無法將資源匯入另一個堆疊或放回來源堆疊。

使用 重構堆疊 AWS 管理主控台

1. 在來源範本中，為您希望移動的資源指定 Retain [DeletionPolicy](#)。

在以下範例來源範本中，Games 是此重構的目標。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
```

```

        "AttributeType": "S"
    },
    ],
    "KeySchema": [
        {
            "AttributeName": "key",
            "KeyType": "HASH"
        }
    ],
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
    }
}

},
"GamesTable": {
    "Type": "AWS::DynamoDB::Table",
    "DeletionPolicy": "Retain",
    "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}

}

}

```

2. 開啟 CloudFormation 主控台來執行堆疊更新，以套用刪除政策。
 - a. 在堆疊頁面上，已選擇堆疊的情況下選擇更新。

- b. 在 Prepare template (準備範本) 下方，選擇 Replace current template (取代目前範本)。
 - c. 在 Specify template (指定範本) 下方，提供在 GamesTable 上具備 DeletionPolicy 屬性的更新來源範本，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中指定指向更新後來源範本的 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽更新後的來源範本檔案。
 - d. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，不需要進行任何變更。選擇下一步。
 - e. 在 Configure stack options (設定堆疊選項) 頁面上，不需要進行任何變更。選擇下一步。
 - f. 在 檢閱 **SourceStackName** 頁面上，檢閱您的變更。如果您的範本包含 IAM 資源，請選取 I acknowledge that this template may create IAM resources (我知道此範本可能會建立 IAM 資源)，以指定您要使用此範本中的 IAM 資源。如需有關使用範本中 IAM 資源的詳細資訊，請參閱[使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。然後，透過建立變更集合來更新來源堆疊，或直接更新來源堆疊。
3. 從來源範本移除資源、相關參數和輸出，然後將這些項目新增到目標範本。

來源範本現在看起來會如下。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
```

```
        "ReadCapacityUnits":5,  
        "WriteCapacityUnits":1  
      }  
    }  
  }  
}
```

以下範例目標範本目前具備 PlayersTable 資源，現在也包含了 GamesTable。

Example JSON

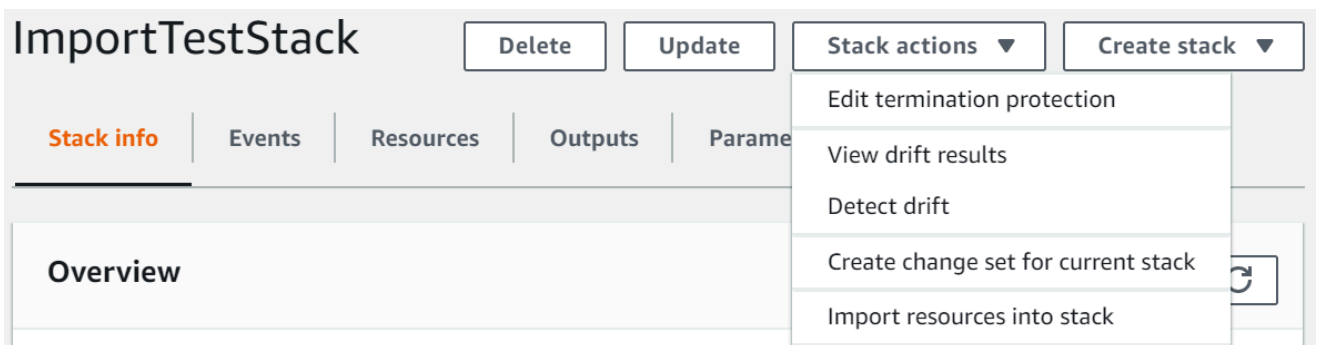
```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "Import test",  
  "Resources": {  
    "PlayersTable": {  
      "Type": "AWS::DynamoDB::Table",  
      "Properties": {  
        "TableName": "Players",  
        "AttributeDefinitions": [  
          {  
            "AttributeName": "key",  
            "AttributeType": "S"  
          }  
        ],  
        "KeySchema": [  
          {  
            "AttributeName": "key",  
            "KeyType": "HASH"  
          }  
        ],  
        "ProvisionedThroughput": {  
          "ReadCapacityUnits": 5,  
          "WriteCapacityUnits": 1  
        }  
      }  
    },  
    "GamesTable": {  
      "Type": "AWS::DynamoDB::Table",  
      "DeletionPolicy": "Retain",  
      "Properties": {  
        "TableName": "Games",
```

```

        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    }
}
}
}

```

4. 重複步驟 2 到 3 以再次更新來源堆疊，這一次請從堆疊刪除目標資源。
5. 執行匯入操作，將 GamesTable 新增到目標堆疊。
 - a. 在 Stacks (堆疊) 頁面上，在選取父堆疊的情況下，選擇 Stack actions (堆疊動作)，然後選擇 Import resources into stack (將資源匯入堆疊)。



- b. 請閱讀 Import overview (匯入概觀) 頁面，以取得您在此操作期間必須提供的項目清單。然後選擇下一步。
- c. 在 Specify template (指定範本) 頁面上，完成以下其中一項操作，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽要上傳的檔案。

- d. 在 Identify resources (識別資源) 頁面上，識別您正在移動的資源 (在此範例中為 GamesTable)。如需詳細資訊，請參閱[資源識別碼](#)。
 1. 在 Identifier property (識別碼屬性) 下方，選擇資源識別碼類型。例如，AWS::DynamoDB::Table 資源可以使用 TableName 屬性進行識別。
 2. 在 Identifier value (識別碼值) 下方，輸入實際的屬性值。例如 *GamesTables*。
 3. 選擇下一步。
- e. 在 Specify stack details (識別堆疊詳細資訊) 頁面上，修改任何參數，然後選擇 Next (下一步)。這會自動建立變更集合。

Important

如果您修改了啟動建立、更新或刪除操作的現有參數，匯入操作便會失敗。

- f. 在檢閱 *TargetStackName* 頁面上，確認您正在匯入正確的資源，然後選擇匯入資源。這會自動啟動在最後一個步驟中建立的變更集。此時會將任何[堆疊層級標籤](#)套用到匯入的資源。
- g. 隨即會顯示父堆疊 Stack details (堆疊詳細資訊) 頁面的 Events (事件) 窗格。

ImportTestStack Delete Update Stack actions ▼ Create stack ▼

Stack info **Events** Resources Outputs Parameters Template Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User initiated

Note

在此匯入操作後不需要在父堆疊上執行漂移偵測，因為 AWS::CloudFormation::Stack 資源已由 CloudFormation 進行管理。

使用 重構堆疊 AWS CLI

1. 在來源範本中，為您希望移動的資源指定 Retain [DeletionPolicy](#)。

在以下範例來源範本中，GamesTable 是此重構的目標。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ]
      }
    }
  }
}
```

```

        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}

```

2. 更新來源堆疊，以將刪除政策套用到資源。

```
aws cloudformation update-stack --stack-name SourceStackName
```

3. 從來源範本移除資源、相關參數和輸出，然後將這些項目新增到目標範本。

來源範本現在看起來會如下。

Example JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {

```



```

    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}

```

4. 更新來源堆疊，以從堆疊刪除 GamesTable 資源及其相關參數和輸出。

```
aws cloudformation update-stack --stack-name SourceStackName
```

5. 以下列 JSON 字串格式編寫要匯入的實際資源清單，以及這些資源的唯一識別碼。如需詳細資訊，請參閱[資源識別碼](#)。

```
[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": "GamesTable", "Properties": {"TableName": "Games"}}
```

您也可以在此檔案中指定 JSON 格式參數。

例如，若要匯入 GamesTable，您可以建立名為 *ResourcesToImport.txt* 的檔案，並包含下列組態。

```
[
```



```
{
  "ResourceType": "AWS::DynamoDB::Table",
  "LogicalResourceId": "GamesTable",
  "ResourceIdentifier": {
    "TableName": "Games"
  }
}
```

6. 要建立變更集，請使用下列 `create-change-set` 命令並取代預留位置文字。對於 `--change-set-type` 選項，請指定值為 **IMPORT**。對於 `--resources-to-import` 選項，請將範例 JSON 字串取代為您剛建立的實際 JSON 字串。

```
aws cloudformation create-change-set \
  --stack-name TargetStackName --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-import
  "'[{"ResourceType": "AWS::DynamoDB::Table", "LogicalResourceId": "GamesTable", "ResourceIdentifier": {"TableName": "Games"}}]'"
```

Note

`--resources-to-import` 不支援內嵌 YAML。JSON 字串中轉義引號的請求有所不同，這取決於您的終端。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[在字串內使用引號](#)。

或者，您可以如下列範例所示，使用檔案 URL 做為 `--resources-to-import` 選項的輸入。

```
--resources-to-import file://ResourcesToImport.txt
```

7. 檢閱變更集合，確認您正在將正確的資源匯入目標堆疊。

```
aws cloudformation describe-change-set \
  --change-set-name ImportChangeSet
```

8. 若要啟動變更集並匯入資源，請使用下列 `execute-change-set` 命令，並取代預留位置文字。此時會將任何堆疊層級標籤套用到匯入的資源。成功完成操作 (IMPORT_COMPLETE) 後，資源便已順利匯入。

```
aws cloudformation execute-change-set \  
  --change-set-name ImportChangeSet --stack-name TargetStackName
```

Note

在此匯入操作後不需要在父堆疊上執行漂移偵測，因為資源已由 CloudFormation 進行管理。

巢狀現有堆疊

使用 `resource import` 功能來在另一個現有堆疊中巢狀現有堆疊。巢狀堆疊是您進行宣告，並從其他範本參考的常見元件。透過這種方式，您可以避免在您的範本中複製和貼上相同的組態，簡化堆疊更新。如果您有一個常見元件的範本，您可以使用 `AWS::CloudFormation::Stack` 資源來從另一個範本中參考此範本。如需巢狀堆疊的詳細資訊，請參閱 [運用巢狀堆疊，將範本分割成可重複使用的部分](#)。

CloudFormation 僅支援使用的一個巢狀層級 `resource import`。這表示您無法將堆疊匯入子堆疊，或是匯入具有子系的堆疊。

若您是初次匯入，建議您先檢閱 [手動將 AWS 資源匯入 CloudFormation 堆疊](#) 主題中的簡介資訊。

巢狀堆疊匯入驗證

在巢狀堆疊匯入操作期間，CloudFormation 會執行下列驗證。

- 父堆疊範本中的巢狀 `AWS::CloudFormation::Stack` 定義與實際的巢狀堆疊範本相符。
- 父堆疊範本中巢狀 `AWS::CloudFormation::Stack` 定義的標籤與實際的巢狀堆疊資源相符。

使用 巢狀現有堆疊 AWS 管理主控台

- 將 `AWS::CloudFormation::Stack` 資源新增到具備 Retain [DeletionPolicy](#) 的父堆疊範本。在以下範例父堆疊範本中，`MyNestedStack` 是匯入目標。

JSON

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Resources" : {
```

```

    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "KeySchema":[
          {
            "AttributeName":"key",
            "KeyType":"HASH"
          }
        ],
        "ProvisionedThroughput":{
          "ReadCapacityUnits":5,
          "WriteCapacityUnits":1
        }
      }
    },
    "MyNestedStack" : {
      "Type" : "AWS::CloudFormation::Stack",
      "DeletionPolicy": "Retain",
      "Properties" : {
        "TemplateURL" : ""https://s3.amazonaws.com/cloudformation-templates-us-east-2/EC2ChooseAMI.template"",
        ""Parameters" : {
          ""InstanceType" : "t1.micro",
          ""KeyName" : "mykey"
        }
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table

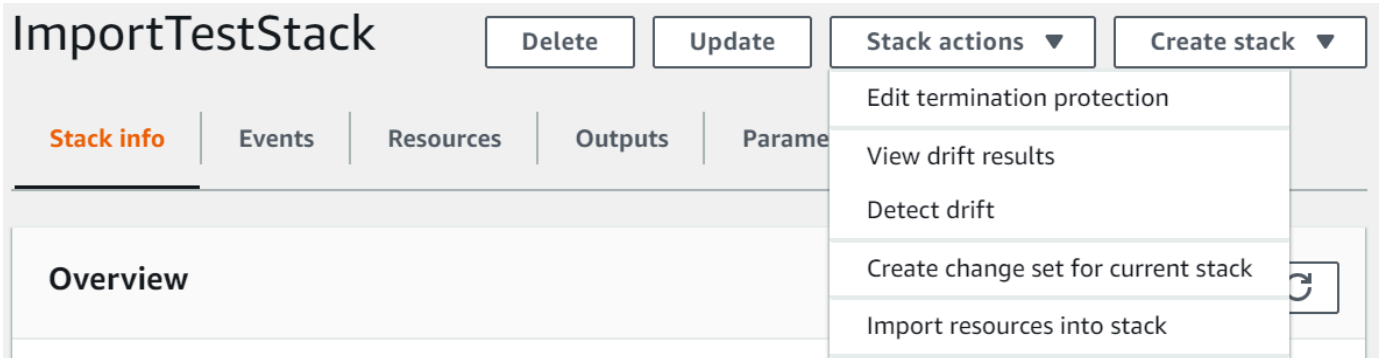
```

```

Properties:
  TableName: Service
  AttributeDefinitions:
    - AttributeName: key
      AttributeType: S
  KeySchema:
    - AttributeName: key
      KeyType: HASH
  ProvisionedThroughput:
    ReadCapacityUnits: 5
    WriteCapacityUnits: 1
MyNestedStack:
  Type: AWS::CloudFormation::Stack
  DeletionPolicy: Retain
  Properties:
    TemplateURL: >-
      https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template
    Parameters:
      InstanceType: t1.micro
      KeyName: mykey

```

2. 開啟 CloudFormation 主控台。
3. 在 Stacks (堆疊) 頁面上，在選取父堆疊的情況下，選擇 Stack actions (堆疊動作)，然後選擇 Import resources into stack (將資源匯入堆疊)。



4. 請閱讀 Import overview (匯入概觀) 頁面，以取得您在此操作期間必須提供的項目清單。然後選擇下一步。
5. 在 Specify template (指定範本) 頁面上，使用以下其中一種方法提供更新後的父範本，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。

6. 在 Identify resources (識別資源) 頁面上，識別 `AWS::CloudFormation::Stack` 資源。
 - a. 在 Identifier property (識別碼屬性) 下方，選擇資源識別碼類型。例如，`AWS::CloudFormation::Stack` 資源可以使用 `StackId` 屬性進行識別。
 - b. 在識別碼值下方，輸入要匯入之堆疊的 ARN。例如 `arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10`。

Resources to import (1)

The following logical IDs are new to your template. Provide the identifier values for the resource you are importing.

NestedStack
AWS::CloudFormation::Stack

Identifier property	Identifier value
StackId	Enter StackId

- c. 選擇下一步。
7. 在 Specify stack details (識別堆疊詳細資訊) 頁面上，修改任何參數，然後選擇 Next (下一步)。這會自動建立變更集合。

 Important

如果您修改了啟動建立、更新或刪除操作的現有參數，匯入操作便會失敗。

8. 在 檢閱 **MyParentStack** 頁面上，確認您正在匯入正確的資源，然後選擇匯入資源。這會自動執行在最後一個步驟中建立的變更集合。此時會將任何堆疊層級標籤套用到匯入的資源。
9. 隨即會顯示父堆疊 Stack details (堆疊詳細資訊) 頁面的 Events (事件) 窗格。

ImportTestStack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events

Q Search events

Timestamp	Logical ID	Status	Status reason
2019-10-24 15:52:49 UTC-0700	ImportTestStack	IMPORT_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_COMPLETE	-
2019-10-24 15:52:47 UTC-0700	SampleIamUser	UPDATE_IN_PROGRESS	Apply stack-level tags to imported resource if applicable.
2019-10-24 15:52:46 UTC-0700	SampleIamUser	IMPORT_COMPLETE	Resource import completed.
2019-10-24 15:52:45 UTC-0700	SampleIamUser	IMPORT_IN_PROGRESS	Resource import started.
2019-10-24 15:52:43 UTC-0700	ImportTestStack	IMPORT_IN_PROGRESS	User Initiated
2019-10-24 15:52:01 UTC-0700	ImportTestStack	REVIEW_IN_PROGRESS	User Initiated

Note

在此匯入操作後不需要在父堆疊上執行漂移偵測，因為
AWS::CloudFormation::Stack 資源已由 CloudFormation 進行管理。

使用 巢狀現有堆疊 AWS CLI

- 將 AWS::CloudFormation::Stack 資源新增到具備 Retain [DeletionPolicy](#) 的父堆疊範本。在以下範例父範本中，MyNestedStack 是匯入目標。

JSON

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "ServiceTable":{
      "Type":"AWS::DynamoDB::Table",
      "Properties":{
        "TableName":"Service",
        "AttributeDefinitions":[
          {
            "AttributeName":"key",
            "AttributeType":"S"
          }
        ],
        "KeySchema":[
          {

```

```

        "AttributeName": "key",
        "KeyType": "HASH"
      }
    ],
    "ProvisionedThroughput": {
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 1
    }
  },
  "MyNestedStack" : {
    "Type" : "AWS::CloudFormation::Stack",
    "DeletionPolicy": "Retain",
    "Properties" : {
      "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-us-east-2/
EC2ChooseAMI.template",
      "Parameters" : {
        "InstanceType" : "t1.micro",
        "KeyName" : "mykey"
      }
    }
  }
}
}
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Resources:
  ServiceTable:
    Type: AWS::DynamoDB::Table
    Properties:
      TableName: Service
      AttributeDefinitions:
        - AttributeName: key
          AttributeType: S
      KeySchema:
        - AttributeName: key
          KeyType: HASH
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 1
  MyNestedStack:

```

```
Type: AWS::CloudFormation::Stack
DeletionPolicy: Retain
Properties:
  TemplateURL: >-
    https://s3.amazonaws.com/cloudformation-templates-us-east-2/
    EC2ChooseAMI.template
  Parameters:
    InstanceType: t1.micro
    KeyName: mykey
```

2. 按照下列範例撰寫 JSON 字串，並進行以下修改：

- 將 *MyNestedStack* 取代為範本中所指定的目標資源邏輯 ID。
- 將 *arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10* 取代為您要匯入之堆疊的 ARN。

```
[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack","ResourceIdentifier":{"StackId":"arn:aws:cloudformation:us-east-2:123456789012:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"}}]
```

您也可以在此組態檔案中指定參數。

例如，若要匯入 *MyNestedStack*，您可以建立名為 *ResourcesToImport.txt* 的檔案，並包含下列組態。

JSON

```
[
  {
    "ResourceType":"AWS::CloudFormation::Stack",
    "LogicalResourceId":"MyNestedStack",
    "ResourceIdentifier": {
      "StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-cd98-11ea-90d5-0a9cd3354c10"
    }
  }
]
```

YAML


```

ResourceType: AWS::CloudFormation::Stack
LogicalResourceId: MyNestedStack
ResourceIdentifier:
  StackId: >-
    arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10

```

3. 要建立變更集，請使用下列 `create-change-set` 命令並取代預留位置文字。對於 `--change-set-type` 選項，請指定值為 **IMPORT**。對於 `--resources-to-import` 選項，請將範例 JSON 字串取代為您剛建立的實際 JSON 字串。

```

aws cloudformation create-change-set \
  --stack-name MyParentStack --change-set-name ImportChangeSet \
  --change-set-type IMPORT \
  --template-body file://TemplateToImport.json \
  --resources-to-import '[{"ResourceType":"AWS::CloudFormation::Stack","LogicalResourceId":"MyNestedStack",
{"StackId":"arn:aws:cloudformation:us-west-2:12345678910:stack/mystack/5b918d10-
cd98-11ea-90d5-0a9cd3354c10"}}]'

```

Note

`--resources-to-import` 不支援內嵌 YAML。JSON 字串中轉義引號的請求有所不同，這取決於您的終端。如需詳細資訊，請參閱《AWS Command Line Interface 使用者指南》中的[在字串內使用引號](#)。

或者，您可以如下列範例所示，使用檔案 URL 做為 `--resources-to-import` 選項的輸入。

```
--resources-to-import file://ResourcesToImport.txt
```

如果成功，此命令傳回的下列範例輸出。

```

{
  "Id": "arn:aws:cloudformation:us-west-2:12345678910:changeSet/
ImportChangeSet/8ad75b3f-665f-46f6-a200-0b4727a9442e",
  "StackId": "arn:aws:cloudformation:us-west-2:12345678910:stack/
MyParentStack/4e345b70-1281-11ef-b027-027366d8e82b"
}

```

4. 檢閱變更集合，確認您正在匯入正確的堆疊。

```
aws cloudformation describe-change-set --change-set-name ImportChangeSet
```

5. 若要啟動變更集並將堆疊匯入來源父堆疊，請使用下列 `execute-change-set` 命令，並取代預留位置文字。此時會將任何[堆疊層級標籤](#)套用到匯入的資源。成功完成匯入操作 (IMPORT_COMPLETE) 後，堆疊便已順利巢狀。

```
aws cloudformation execute-change-set --change-set-name ImportChangeSet
```

Note

在此匯入操作後不需要在父堆疊上執行漂移偵測，因為 `AWS::CloudFormation::Stack` 資源已由 CloudFormation 進行管理。

自動將 AWS 資源匯入 CloudFormation 堆疊

現在建立或更新 CloudFormation 堆疊時，您可以自動匯入具名資源。具名資源指具有自訂名稱的資源。如需詳細資訊，請參閱 CloudFormation 範本參考中的[名稱類型](#)。

啟動自動匯入後，CloudFormation 會檢查與您的範本匹配的現有資源，並在部署期間將其匯入。對於巢狀堆疊，請從根堆疊建立變更集。

在匯入完成且在執行後續的堆疊操作前，我們建議在匯入的資源上執行漂移偵測。漂移偵測可確保範本組態與實際組態相符。如需詳細資訊，請參閱[在整個 CloudFormation 堆疊上偵測偏離](#)。

若要匯入資源，需符合下列要求：

- 資源必須在您的範本中定義靜態自訂名稱。目前不支援動態名稱 (使用 `!Ref` 或其他函數建立)。
- 資源必須具有 `Retain` 或 `RetainExceptOnCreate` 的 `DeletionPolicy`。
- 資源不得已屬於另一個 CloudFormation 堆疊。
- 該資源類型必須支援 CloudFormation 匯入操作。如需詳細資訊，請參閱[資源類型支援](#)。
- 資源類型的主要 ID 必須包含在範本中。具有唯讀屬性的主要 ID 不受支援。若要查詢某類型的主要 ID，請在資源結構描述中查找 `primaryIdentifier` 屬性。如需該屬性的詳細資訊，請參閱[primaryIdentifier](#)。

Example 自動匯入範例

下列範例使用變更集 `CreateChangeSet`，根據範本檔案 `template.yaml` 建立名為 `my-stack` 的堆疊，並會自動匯入相符的資源。

```
aws cloudformation create-change-set \  
  --stack-name my-stack \  
  --change-set-name CreateChangeSet \  
  --change-set-type CREATE \  
  --template-body file://template.yaml \  
  --import-existing-resources
```

疑難排解

若自動匯入失敗，請執行下列步驟進行疑難排解：

- 確認範本中的資源名稱與實際資源名稱是否完全一致
- 確認該資源未被另一個堆疊管理
- 確定資源類型支援匯入操作
- 確認您的範本包含資源類型所需的全部必要屬性

還原匯入操作

若要還原匯入操作，請為您要從範本移除的資源指定 `Retain` 刪除政策，確保在您從堆疊刪除該資源時，仍會保留該資源。

使用 還原匯入操作 AWS 管理主控台

1. 為您希望從堆疊移除的資源指定 `Retain` [DeletionPolicy](#)。在以下範例範本中，`GameTable` 是此還原操作的目標。

Example JSON

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "Import test",  
  "Resources": {  
    "ServiceTable": {  
      "Type": "AWS::DynamoDB::Table",  
      "Properties": {
```

```
        "TableName": "Service",
        "AttributeDefinitions": [
            {
                "AttributeName": "key",
                "AttributeType": "S"
            }
        ],
        "KeySchema": [
            {
                "AttributeName": "key",
                "KeyType": "HASH"
            }
        ],
        "ProvisionedThroughput": {
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 1
        }
    },
    "GamesTable": {
        "Type": "AWS::DynamoDB::Table",
        "DeletionPolicy": "Retain",
        "Properties": {
            "TableName": "Games",
            "AttributeDefinitions": [
                {
                    "AttributeName": "key",
                    "AttributeType": "S"
                }
            ],
            "KeySchema": [
                {
                    "AttributeName": "key",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 5,
                "WriteCapacityUnits": 1
            }
        }
    }
}
```

```
}
```

2. 開啟 CloudFormation 主控台來執行堆疊更新，以套用刪除政策。
 - a. 在 Stacks (堆疊) 頁面上，選取堆疊、選擇 Update (更新)，然後選擇 Update stack (standard) (更新堆疊 (標準))。
 - b. 在 Prepare template (準備範本) 下方，選擇 Replace current template (取代目前範本)。
 - c. 在 Specify template (指定範本) 下方，提供在 GamesTable 上具備 DeletionPolicy 屬性的更新來源範本，然後選擇 Next (下一步)。
 - 選擇 Amazon S3 URL，然後在文字方塊中指定指向更新後來源範本的 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽更新後的來源範本檔案。
 - d. 在 Specify stack details (指定堆疊詳細資訊) 頁面上，不需要進行任何變更。選擇下一步。
 - e. 在 Configure stack options (設定堆疊選項) 頁面上，不需要進行任何變更。選擇下一步。
 - f. 在檢閱 **stack_name** 頁面上，檢閱您的變更。如果您的範本包含 IAM 資源，請選取 I acknowledge that this template may create IAM resources (我知道此範本可能會建立 IAM 資源)，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。然後，透過建立變更集合來更新來源堆疊，或直接更新來源堆疊。
3. 從堆疊範本中移除資源、相關參數和輸出。在此範例中，範本現在看起來如下。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ]
      }
    }
  }
}
```

```
}  
],  
  "ProvisionedThroughput": {  
    "ReadCapacityUnits": 5,  
    "WriteCapacityUnits": 1  
  }  
}  
}  
}
```

4. 重複步驟 2，以從堆疊刪除資源 (GamesTable) 及其相關參數和輸出。

使用 還原匯入操作 AWS CLI

1. 為您希望從堆疊移除的資源指定 Retain [DeletionPolicy](#)。在以下範例範本中，GamesTable 是此還原操作的目標。

Example JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "TableName": "Service",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
```

```

        "WriteCapacityUnits":1
      }
    },
    "GamesTable": {
      "Type": "AWS::DynamoDB::Table",
      "DeletionPolicy": "Retain",
      "Properties": {
        "TableName": "Games",
        "AttributeDefinitions": [
          {
            "AttributeName": "key",
            "AttributeType": "S"
          }
        ],
        "KeySchema": [
          {
            "AttributeName": "key",
            "KeyType": "HASH"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 5,
          "WriteCapacityUnits": 1
        }
      }
    }
  }
}

```

2. 更新堆疊，以將刪除政策套用到資源。

```
aws cloudformation update-stack --stack-name MyStack
```

3. 從堆疊範本中移除資源、相關參數和輸出。在此範例中，範本現在看起來如下。

Example JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Import test",
  "Resources": {
    "ServiceTable":{

```

```
    "Type": "AWS::DynamoDB::Table",
    "Properties": {
      "TableName": "Service",
      "AttributeDefinitions": [
        {
          "AttributeName": "key",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "key",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 5,
        "WriteCapacityUnits": 1
      }
    }
  }
}
```

4. 更新堆疊以從堆疊刪除資源 (GamesTable) 及其相關參數和輸出。

```
aws cloudformation update-stack --stack-name MyStack
```

堆疊重構

透過堆疊重構，您可以重整 CloudFormation 堆疊中的資源，同時保留現有資源的屬性與資料。您可以在堆疊之間移動資源、將大型堆疊分割為較小的堆疊，或將多個堆疊合併為一個堆疊。

主題

- [堆疊重構的運作方式](#)
- [堆疊重構考量事項](#)
- [先決條件](#)
- [重構堆疊（主控台）](#)
- [重構堆疊 \(AWS CLI\)](#)

- [資源限制](#)

堆疊重構的運作方式

堆疊重構包含下列階段：

1. 評估目前的基礎設施 – 檢閱您現有的 CloudFormation 堆疊與資源，找出可進行堆疊重構的機會。
2. 規劃重構 – 定義資源的組織方式。考量資源相依性、命名慣例及運作限制。這些因素會影響後續的 CloudFormation 驗證。
3. 決定目的地堆疊 – 確定您要將資源重構至哪些堆疊。您可以在至少 2 個堆疊（使用 主控台）和最多 5 個堆疊（使用 `AWS CLI`）之間移動資源。資源可在巢狀堆疊之間移動。
4. 更新範本 – 變更 CloudFormation 範本，以反映規劃的變更，例如在範本間移動資源定義。您可在程序期間重新命名邏輯 ID。
5. 建立堆疊重構 – 提供您要重構的堆疊名稱與範本清單。
6. 檢閱重構影響並解決任何衝突 – CloudFormation 會驗證您提供的範本，並檢查跨堆疊相依性、標籤更新有問題的資源類型，以及資源邏輯 ID 衝突。

若驗證成功，CloudFormation 會產生執行期間即將執行的重構動作預覽。

若驗證失敗，請解決偵測到的問題後重試。若有衝突，請提供資源邏輯 ID 映射，以顯示衝突資源的來源和目的地。

7. 執行重構 – 確認變更符合重構目標後，完成堆疊重構。
8. 監控 – 追蹤執行狀態，確保作業順利完成。

堆疊重構考量事項

重構堆疊時，請注意下列事項：

- 堆疊重構僅限於重整現有資源。您無法在重構期間建立或刪除資源、修改資源組態，或變更或新增參數、條件或映射。若要進行這些變更，請先更新您的堆疊，然後執行堆疊重構。
- 您無法將相同的資源重構入多個堆疊。
- 若資源參考的虛擬參數 (例如 `AWS::StackName`) 在來源與目標堆疊中的數值不同，則無法對該資源進行重構。

- CloudFormation 不支援空白堆疊。若重構會導致某個堆疊沒有資源，您必須先向該堆疊新增至少一項資源，再執行 [create-stack-refactor](#)。可新增簡單資源，例如 `AWS::SNS::Topic` 或 `AWS::CloudFormation::WaitCondition`。例如：

```
Resources:
  MySimpleSNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: MySimpleTopic
```

- 堆疊重構不支援已附加堆疊政策的堆疊，不論該政策允許或拒絕哪些操作。

先決條件

若要重構堆疊，您必須已建立修訂後的範本。

使用 [get-template](#) 命令，擷取您要重構的堆疊對應 CloudFormation 範本。

```
aws cloudformation get-template --stack-name Stack1
```

取得範本後，使用您偏好的整合式開發環境 (IDE) 更新範本，使其符合預期的結構與資源組織方式。

重構堆疊（主控台）

使用下列程序來重構使用主控台的堆疊。

重構堆疊

- 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
- 在畫面頂端的導覽列上，選擇 AWS 區域 堆疊所在的。
- 在左側導覽窗格中，選擇堆疊重構。
- 在堆疊重構頁面上，選擇開始堆疊重構。
- 針對描述，提供描述以協助您識別堆疊重構。然後選擇下一步。
- 對於堆疊 1，請執行下列動作：
 - 選擇更新現有堆疊的範本或建立新堆疊。

如果您選擇更新現有堆疊的範本，請從清單中選擇現有的堆疊。或者，選擇輸入堆疊 ARN 以輸入現有堆疊的 ARN。

如果您選擇建立新堆疊，請針對堆疊名稱提供新堆疊的名稱。

- b. 在將現有範本取代為重構範本下，選擇 Amazon S3 URL 或上傳範本檔案，以上傳 Stack 1 所需的範本。
- c. 選擇下一步。

7. 對於 Stack 2，請執行下列動作：

- a. 選擇更新現有堆疊的範本或建立新堆疊。

如果您選擇更新現有堆疊的範本，請從清單中選擇現有的堆疊。或者，選擇輸入堆疊 ARN 以輸入現有堆疊的 ARN。

如果您選擇建立新堆疊，請針對堆疊名稱提供新堆疊的名稱。

- b. 在將現有範本取代為重構範本下，選擇 Amazon S3 URL 或上傳範本檔案，以上傳 Stack 2 所需的範本。
- c. 選擇下一步。

8. 在指定邏輯資源 ID 重新命名頁面上，確定 CloudFormation 知道如何將顯示的任何資源映射至其正確的邏輯 IDs 來重構堆疊。作為堆疊重構的一部分，如果任何資源的邏輯 IDs 已變更，您需要提供來源堆疊名稱、原始邏輯 ID、目的地堆疊名稱和重新命名的邏輯 ID 來指定其重新命名方式。在某些情況下，CloudFormation 主控台可能會自動偵測資源映射，您可以直接驗證預先填入的資源映射是否正確，然後再繼續。

9. 選擇下一步。
10. 在檢閱和執行頁面上，檢閱先前步驟中的所有選擇，並確認一切設定正確。
11. 當您準備好重構堆疊時，請選擇執行堆疊重構。

重構堆疊 (AWS CLI)

用於堆疊重構的 AWS CLI 命令包括：

- [create-stack-refactor](#) 用於驗證和產生規劃變更的預覽。
- [describe-stack-refactor](#) 用於擷取堆疊重構操作的狀態與詳細資訊。
- [execute-stack-refactor](#) 用於完成經驗證的堆疊重構操作。
- [list-stack-refactors](#) 用於列出您帳戶中所有堆疊重構操作及其目前狀態和基本資訊。

- [list-stack-refactor-actions](#) 用於顯示 CloudFormation 在重構執行期間將對每個堆疊和資源執行的特定動作的預覽。

利用以下程序使用 AWS CLI 重構堆疊。

重構堆疊

1. 使用 [create-stack-refactor](#) 命令，並提供要重構的堆疊名稱與更新後的範本。加入 `--enable-stack-creation` 選項，讓 CloudFormation 可在新堆疊不存在時建立。

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation
```

命令會傳回 `StackRefactorId`，後續步驟將會用到。

```
{  
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841"  
}
```

若範本驗證期間偵測到衝突 (可在下一個步驟確認)，請搭配 `--resource-mappings` 選項使用 [create-stack-refactor](#) 命令。

```
aws cloudformation create-stack-refactor \  
  --stack-definitions \  
    StackName=Stack1,TemplateBody@=file://template1-updated.yaml \  
    StackName=Stack2,TemplateBody@=file://template2-updated.yaml \  
  --enable-stack-creation \  
  --resource-mappings file://resource-mapping.json
```

以下是範例 `resource-mapping.json` 檔案。

```
[  
  {  
    "Source": {  
      "StackName": "Stack1",  
      "LogicalResourceId": "MySNSTopic"  
    },  
  },  
]
```

```
        "Destination": {
            "StackName": "Stack2",
            "LogicalResourceId": "MyLambdaSNSTopic"
        }
    }
}
```

2. 使用 [describe-stack-refactor](#) 命令來確定 Status 為 CREATE_COMPLETE。這表示驗證已完成。

```
aws cloudformation describe-stack-refactor \
    --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

輸出範例：

```
{
    "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
    "StackIds": [
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "arn:aws:cloudformation:us-east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
    ],
    "ExecutionStatus": "AVAILABLE",
    "Status": "CREATE_COMPLETE"
}
```

3. 使用 [list-stack-refactor-actions](#) 命令，預覽即將執行的特定動作。

```
aws cloudformation list-stack-refactor-actions \
    --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841
```

輸出範例：

```
{
    "StackRefactorActions": [
        {
            "Action": "MOVE",
            "Entity": "RESOURCE",
            "PhysicalResourceId": "MyTestLambdaRole",
            "Description": "No configuration changes detected.",
            "Detection": "AUTO",
            "TagResources": [],

```

```

        "UntagResources": [],
        "ResourceMapping": {
            "Source": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
                "LogicalResourceId": "MyLambdaRole"
            },
            "Destination": {
                "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
                "LogicalResourceId": "MyLambdaRole"
            }
        },
        {
            "Action": "MOVE",
            "Entity": "RESOURCE",
            "PhysicalResourceId": "MyTestFunction",
            "Description": "Resource configuration changes will be validated during
refactor execution.",
            "Detection": "AUTO",
            "TagResources": [
                {
                    "Key": "aws:cloudformation:stack-name",
                    "Value": "Stack2"
                },
                {
                    "Key": "aws:cloudformation:logical-id",
                    "Value": "MyFunction"
                },
                {
                    "Key": "aws:cloudformation:stack-id",
                    "Value": "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
                }
            ],
            "UntagResources": [
                "aws:cloudformation:stack-name",
                "aws:cloudformation:logical-id",
                "aws:cloudformation:stack-id"
            ],
            "ResourceMapping": {
                "Source": {

```

```

        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
        "LogicalResourceId": "MyFunction"
    },
    "Destination": {
        "StackName": "arn:aws:cloudformation:us-
east-1:123456789012:stack/Stack2/5da91650-94b1-11f0-81cf-0a23500e151b",
        "LogicalResourceId": "MyFunction"
    }
}
]
}

```

4. 檢閱並確認變更後，使用 [execute-stack-refactor](#) 命令來完成堆疊重構操作。

```

aws cloudformation execute-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

5. 使用 [describe-stack-refactor](#) 命令監控執行狀態。

```

aws cloudformation describe-stack-refactor \
  --stack-refactor-id 9c384f70-4e07-4ed7-a65d-fee5eb430841

```

輸出範例：

```

{
  "StackRefactorId": "9c384f70-4e07-4ed7-a65d-fee5eb430841",
  "StackIds": [
    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack1/3e6a1ff0-94b1-11f0-aa6f-0a88d2e03acf",
    "arn:aws:cloudformation:us-east-1:123456789012:stack/
Stack2/5da91650-94b1-11f0-81cf-0a23500e151b"
  ],
  "ExecutionStatus": "SUCCEEDED",
  "Status": "COMPLETE"
}

```

資源限制

- 堆疊重構僅支援具有 provisioningType 為 FULLY_MUTABLE 的資源類型，您可以使用 [describe-type](#) 命令確認。
- 建立重構時，CloudFormation 會驗證資源適用性，並在 [describe-stack-refactor](#) 命令的輸出中回報所有不支援的資源。
- 下列資源不支援堆疊重構：
 - AWS::ACMPCA::Certificate
 - AWS::ACMPCA::CertificateAuthority
 - AWS::ACMPCA::CertificateAuthorityActivation
 - AWS::ApiGateway::BasePathMapping
 - AWS::ApiGateway::Method
 - AWS::AppConfig::ConfigurationProfile
 - AWS::AppConfig::Deployment
 - AWS::AppConfig::Environment
 - AWS::AppConfig::Extension
 - AWS::AppConfig::ExtensionAssociation
 - AWS::AppStream::DirectoryConfig
 - AWS::AppStream::StackFleetAssociation
 - AWS::AppStream::StackUserAssociation
 - AWS::AppStream::User
 - AWS::BackupGateway::Hypervisor
 - AWS::CertificateManager::Certificate
 - AWS::CloudFormation::CustomResource
 - AWS::CloudFormation::Macro
 - AWS::CloudFormation::WaitCondition
 - AWS::CloudFormation::WaitConditionHandle
 - AWS::CodeDeploy::DeploymentGroup
 - AWS::CodePipeline::CustomActionType
 - AWS::Cognito::UserPoolRiskConfigurationAttachment
 - AWS::Cognito::UserPoolUICustomizationAttachment

- AWS::Cognito::UserPoolUserToGroupAttachment
- AWS::Config::ConfigRule
- AWS::Config::ConfigurationRecorder
- AWS::Config::DeliveryChannel
- AWS::DataBrew::Dataset
- AWS::DataBrew::Job
- AWS::DataBrew::Project
- AWS::DataBrew::Recipe
- AWS::DataBrew::Ruleset
- AWS::DataBrew::Schedule
- AWS::DataZone::DataSource
- AWS::DataZone::Environment
- AWS::DataZone::EnvironmentBlueprintConfiguration
- AWS::DataZone::EnvironmentProfile
- AWS::DataZone::Project
- AWS::DataZone::SubscriptionTarget
- AWS::DirectoryService::MicrosoftAD
- AWS::DynamoDB::GlobalTable
- AWS::EC2::CustomerGateway
- AWS::EC2::EIP
- AWS::EC2::LaunchTemplate
- AWS::EC2::NetworkInterfacePermission
- AWS::EC2::PlacementGroup
- AWS::EC2::SpotFleet
- AWS::EC2::VPCDHCPOptionsAssociation
- AWS::EC2::VolumeAttachment
- AWS::EMR::Cluster
- AWS::EMR::InstanceFleetConfig
- AWS::EMR::InstanceGroupConfig
- AWS::ElastiCache::CacheCluster

- AWS::ElastiCache::ReplicationGroup
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache::SecurityGroupIngress
- AWS::ElasticBeanstalk::ConfigurationTemplate
- AWS::ElasticLoadBalancing::LoadBalancer
- AWS::ElasticLoadBalancingV2::ListenerCertificate
- AWS::Elasticsearch::Domain
- AWS::FIS::ExperimentTemplate
- AWS::Glue::Schema
- AWS::GuardDuty::IPSet
- AWS::GuardDuty::PublishingDestination
- AWS::GuardDuty::ThreatIntelSet
- AWS::IAM::AccessKey
- AWS::IAM::UserToGroupAddition
- AWS::ImageBuilder::Component
- AWS::IoT::PolicyPrincipalAttachment
- AWS::IoT::ThingPrincipalAttachment
- AWS::IoTFleetWise::Campaign
- AWS::IoTWireless::WirelessDeviceImportTask
- AWS::Lambda::EventInvokeConfig
- AWS::Lex::BotVersion
- AWS::M2::Application
- AWS::MSK::Configuration
- AWS::MSK::ServerlessCluster
- AWS::Maester::DocumentType
- AWS::MediaTailor::Channel
- AWS::NeptuneGraph::PrivateGraphEndpoint
- AWS::Omics::AnnotationStore
- AWS::Omics::ReferenceStore
- AWS::Omics::SequenceStore

- `AWS::OpenSearchServerless::Collection`
- `AWS::OpsWorks::App`
- `AWS::OpsWorks::ElasticLoadBalancerAttachment`
- `AWS::OpsWorks::Instance`
- `AWS::OpsWorks::Layer`
- `AWS::OpsWorks::Stack`
- `AWS::OpsWorks::UserProfile`
- `AWS::OpsWorks::Volume`
- `AWS::PCAConnectorAD::Connector`
- `AWS::PCAConnectorAD::DirectoryRegistration`
- `AWS::PCAConnectorAD::Template`
- `AWS::PCAConnectorAD::TemplateGroupAccessControlEntry`
- `AWS::Panorama::PackageVersion`
- `AWS::QuickSight::Theme`
- `AWS::RDS::DBSecurityGroup`
- `AWS::RDS::DBSecurityGroupIngress`
- `AWS::Redshift::ClusterSecurityGroup`
- `AWS::Redshift::ClusterSecurityGroupIngress`
- `AWS::RefactorSpaces::Environment`
- `AWS::RefactorSpaces::Route`
- `AWS::RefactorSpaces::Service`
- `AWS::RoboMaker::RobotApplication`
- `AWS::RoboMaker::SimulationApplication`
- `AWS::Route53::RecordSet`
- `AWS::Route53::RecordSetGroup`
- `AWS::SDB::Domain`
- `AWS::SageMaker::InferenceComponent`
- `AWS::ServiceCatalog::PortfolioPrincipalAssociation`
- `AWS::ServiceCatalog::PortfolioProductAssociation`
- `AWS::ServiceCatalog::PortfolioShare`

- `AWS::ServiceCatalog::TagOptionAssociation`
- `AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation`
- `AWS::ServiceCatalogAppRegistry::ResourceAssociation`
- `AWS::StepFunctions::StateMachineVersion`
- `AWS::Synthetics::Canary`
- `AWS::VoiceID::Domain`
- `AWS::WAF::ByteMatchSet`
- `AWS::WAF::IPSet`
- `AWS::WAF::Rule`
- `AWS::WAF::SizeConstraintSet`
- `AWS::WAF::SqlInjectionMatchSet`
- `AWS::WAF::WebACL`
- `AWS::WAF::XssMatchSet`
- `AWS::WAFv2::IPSet`
- `AWS::WAFv2::RegexPatternSet`
- `AWS::WAFv2::RuleGroup`
- `AWS::WAFv2::WebACL`
- `AWS::WorkSpaces::Workspace`

資源類型支援

下表列出目前支援匯入、偏離偵測和基礎設施即程式碼 (IaC) 產生器操作 AWS 的資源類型。每個資源類型名稱都會連結至 [CloudFormation 範本參考指南](#) 中對應的參考主題。

支援資源匯入的資源也可支援自動匯入。如需詳細資訊，請參閱[將 AWS 資源匯入 CloudFormation 堆疊](#)。

這不是詳盡 AWS 的資源清單。若特定資源類型未列出，可能表示它無法透過 AWS 雲端控制 API 存取。如需詳細資訊，請參閱《Cloud Control API 使用者指南》中的[支援 Cloud Control API 的資源類型](#)。每個 AWS 服務都會獨立決定要透過 Cloud Control API 存取哪些資源類型。

CloudFormation 還支援可佈建私有資源類型的匯入和漂移偵測作業 (佈建類型為 `FULLY_MUTABLE` 或 `IMMUTABLE`)。若要匯入或執行私人資源類型的漂移偵測，您必須先在帳戶中註冊該資源類型的預設版本，並確保其可佈建。如需詳細資訊，請參閱[使用已與您共用的第三方私有擴充功能](#)。

請注意，IaC 產生器僅支援與您區域中 Cloud Control API 相容的 AWS 資源。

若要以程式設計方式存取公有和私有可佈建資源類型的相關資訊，您可使用 AWS 雲端控制 API。如需詳細資訊，請參閱《Cloud Control API 使用者指南》中的[判斷資源類型是否支援 Cloud Control API](#)。

若要開始使用匯入、漂移偵測或 IaC 產生器，以下是一些值得檢閱的實用主題：

- [將 AWS 資源匯入 CloudFormation 堆疊](#)
- [透過漂移偵測功能，偵測堆疊和資源的未受管組態變更](#)
- [使用 IaC 產生器從現有資源中產生範本](#)

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::ACMPCA::Certificate	 是	 否	 否
AWS::ACMPCA::CertificateAuthority	 是	 是	 是
AWS::ACMPCA::CertificateAuthorityActivation	 是	 否	 否
AWS::ACMPCA::Permission	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::AIOps::InvestigationGroup</u>	 是	 是	 否
<u>AWS::APS::AnomalyDetector</u>	 是	 是	 否
<u>AWS::APS::ResourcePolicy</u>	 是	 是	 否
<u>AWS::APS::RuleGroupsNamespace</u>	 是	 是	 是
<u>AWS::APS::Scraper</u>	 是	 是	 否
<u>AWS::APS::Workspace</u>	 是	 是	 是
<u>AWS::ARCRegionSwitch::Plan</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ARCZonalShift::AutoshiftObserveNotificationStatus</u>	 是	 是	 否
<u>AWS::ARCZonalShift::ZonalAutoshiftConfiguration</u>	 是	 是	 否
<u>AWS::AccessAnalyzer::Analyzer</u>	 是	 是	 是
<u>AWS::AmazonMQ::Broker</u>	 是	 是	 否
<u>AWS::AmazonMQ::Configuration</u>	 是	 是	 否
<u>AWS::Amplify::App</u>	 是	 是	 是
<u>AWS::Amplify::Branch</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Amplify::Domain</u>	 是	 是	 是
<u>AWS::AmplifyUIBuilder::Component</u>	 是	 是	 是
<u>AWS::AmplifyUIBuilder::Form</u>	 是	 是	 是
<u>AWS::AmplifyUIBuilder::Theme</u>	 是	 是	 是
<u>AWS::ApiGateway::Account</u>	 是	 是	 否
<u>AWS::ApiGateway::ApiKey</u>	 是	 是	 是
<u>AWS::ApiGateway::Authorizer</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::ApiGateway::BasePathMapping</u>	 是	 是	 否
<u>AWS::ApiGateway::BasePathMappingV2</u>	 是	 是	 否
<u>AWS::ApiGateway::ClientCertificate</u>	 是	 是	 是
<u>AWS::ApiGateway::Deployment</u>	 是	 是	 是
<u>AWS::ApiGateway::DocumentationPart</u>	 是	 是	 否
<u>AWS::ApiGateway::DocumentationVersion</u>	 是	 是	 是
<u>AWS::ApiGateway::DomainName</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ApiGateway::DomainNameAccessAssociation</u>	 是	 是	 否
<u>AWS::ApiGateway::DomainNameV2</u>	 是	 是	 否
<u>AWS::ApiGateway::GatewayResponse</u>	 是	 否	 否
<u>AWS::ApiGateway::Method</u>	 是	 是	 否
<u>AWS::ApiGateway::Model</u>	 是	 是	 否
<u>AWS::ApiGateway::RequestValidator</u>	 是	 是	 否
<u>AWS::ApiGateway::Resource</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::ApiGateway::RestApi	 是	 是	 是
AWS::ApiGateway::Stage	 是	 是	 是
AWS::ApiGateway::UsagePlan	 是	 是	 是
AWS::ApiGateway::UsagePlanKey	 是	 否	 是
AWS::ApiGateway::VpcLink	 是	 是	 否
AWS::ApiGatewayV2::Api	 是	 是	 是
AWS::ApiGatewayV2::ApiMapping	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ApiGatewayV2::Authorizer</u>	 是	 是	 否
<u>AWS::ApiGatewayV2::Deployment</u>	 是	 是	 是
<u>AWS::ApiGatewayV2::DomainName</u>	 是	 是	 是
<u>AWS::ApiGatewayV2::Integration</u>	 是	 否	 否
<u>AWS::ApiGatewayV2::IntegrationResponse</u>	 是	 是	 否
<u>AWS::ApiGatewayV2::Model</u>	 是	 是	 否
<u>AWS::ApiGatewayV2::Route</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ApiGatewayV2::RouteResponse</u>	 是	 是	 否
<u>AWS::ApiGatewayV2::RoutingRule</u>	 是	 是	 否
<u>AWS::ApiGatewayV2::VpcLink</u>	 是	 是	 是
<u>AWS::AppConfig::Application</u>	 是	 是	 是
<u>AWS::AppConfig::ConfigurationProfile</u>	 是	 是	 否
<u>AWS::AppConfig::Deployment</u>	 是	 是	 否
<u>AWS::AppConfig::DeploymentStrategy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AppConfig::Environment</u>	 是	 是	 否
<u>AWS::AppConfig::Extension</u>	 是	 是	 否
<u>AWS::AppConfig::ExtensionAssociation</u>	 是	 是	 否
<u>AWS::AppConfig::HostedConfigurationVersion</u>	 是	 是	 否
<u>AWS::AppFlow::Connector</u>	 是	 是	 是
<u>AWS::AppFlow::ConnectorProfile</u>	 是	 是	 是
<u>AWS::AppFlow::Flow</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AppIntegrations::Application</u>	 是	 是	 否
<u>AWS::AppIntegrations::DataIntegration</u>	 是	 是	 是
<u>AWS::AppIntegrations::EventIntegration</u>	 是	 是	 是
<u>AWS::AppRunner::AutoScalingConfiguration</u>	 是	 是	 否
<u>AWS::AppRunner::ObservabilityConfiguration</u>	 是	 是	 否
<u>AWS::AppRunner::Service</u>	 是	 是	 否
<u>AWS::AppRunner::VpcConnector</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AppRunner::VpcIngressConnection</u>	 是	 是	 否
<u>AWS::AppStream::AppBlock</u>	 是	 是	 否
<u>AWS::AppStream::AppBlockBuilder</u>	 是	 是	 否
<u>AWS::AppStream::Application</u>	 是	 是	 否
<u>AWS::AppStream::ApplicationEntitlementAssociation</u>	 是	 是	 否
<u>AWS::AppStream::ApplicationFleetAssociation</u>	 是	 是	 否
<u>AWS::AppStream::DirectoryConfig</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AppStream::Entitlement</u>	 是	 是	 否
<u>AWS::AppStream::ImageBuilder</u>	 是	 是	 是
<u>AWS::AppSync::Api</u>	 是	 是	 否
<u>AWS::AppSync::ChannelNamespace</u>	 是	 是	 否
<u>AWS::AppSync::DataSource</u>	 是	 是	 否
<u>AWS::AppSync::DomainName</u>	 是	 是	 是
<u>AWS::AppSync::DomainNameApiAssociation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AppSync::FunctionConfiguration</u>	 是	 是	 否
<u>AWS::AppSync::GraphQLApi</u>	 否	 是	 否
<u>AWS::AppSync::Resolver</u>	 是	 是	 否
<u>AWS::AppSync::SourceApiAssociation</u>	 是	 是	 是
<u>AWS::AppTest::TestCase</u>	 是	 是	 否
<u>AWS::ApplicationAutoScaling::ScalableTarget</u>	 是	 是	 是
<u>AWS::ApplicationAutoScaling::ScalingPolicy</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ApplicationInsights::Application</u>	 是	 是	 否
<u>AWS::ApplicationSignals::Discovery</u>	 是	 是	 否
<u>AWS::ApplicationSignals::GroupingConfiguration</u>	 是	 是	 否
<u>AWS::ApplicationSignals::ServiceLevelObjective</u>	 是	 是	 否
<u>AWS::Athena::CapacityReservation</u>	 是	 是	 是
<u>AWS::Athena::DataCatalog</u>	 是	 是	 是
<u>AWS::Athena::NamedQuery</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Athena::PreparedStatement</u>	 是	 是	 是
<u>AWS::Athena::WorkGroup</u>	 是	 否	 是
<u>AWS::AuditManager::Assessment</u>	 是	 否	 否
<u>AWS::AutoScaling::AutoScalingGroup</u>	 是	 是	 是
<u>AWS::AutoScaling::LaunchConfiguration</u>	 是	 是	 是
<u>AWS::AutoScaling::LifecycleHook</u>	 是	 是	 是
<u>AWS::AutoScaling::ScalingPolicy</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::AutoScaling::ScheduledAction</u>	 是	 是	 是
<u>AWS::AutoScaling::WarmPool</u>	 是	 是	 否
<u>AWS::B2BI::Capability</u>	 是	 是	 否
<u>AWS::B2BI::Partnership</u>	 是	 是	 否
<u>AWS::B2BI::Profile</u>	 是	 是	 否
<u>AWS::B2BI::Transformer</u>	 是	 是	 否
<u>AWS::BCMDDataExports::Export</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Backup::BackupPlan</u>	 是	 是	 否
<u>AWS::Backup::BackupSelection</u>	 是	 否	 是
<u>AWS::Backup::BackupVault</u>	 是	 是	 否
<u>AWS::Backup::Framework</u>	 是	 是	 是
<u>AWS::Backup::LogicallyAirGappedBackupVault</u>	 是	 是	 否
<u>AWS::Backup::ReportPlan</u>	 是	 是	 是
<u>AWS::Backup::RestoreTestingPlan</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Backup::RestoreTestingSelection</u>	 是	 是	 否
<u>AWS::Backup::TieringConfiguration</u>	 是	 是	 否
<u>AWS::BackupGateway::Hypervisor</u>	 是	 是	 否
<u>AWS::Batch::ComputeEnvironment</u>	 是	 是	 是
<u>AWS::Batch::ConsumableResource</u>	 是	 是	 否
<u>AWS::Batch::JobDefinition</u>	 是	 是	 否
<u>AWS::Batch::JobQueue</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Batch::SchedulingPolicy</u>	 是	 是	 是
<u>AWS::Batch::ServiceEnvironment</u>	 是	 是	 否
<u>AWS::Bedrock::Agent</u>	 是	 是	 是
<u>AWS::Bedrock::AgentAlias</u>	 是	 是	 是
<u>AWS::Bedrock::ApplicationInferenceProfile</u>	 是	 是	 是
<u>AWS::Bedrock::AutomatedReasoningPolicy</u>	 是	 是	 否
<u>AWS::Bedrock::AutomatedReasoningPolicyVersion</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::Bedrock::Blueprint	 是	 是	 是
AWS::Bedrock::DataAutomationProject	 是	 是	 是
AWS::Bedrock::DataSource	 是	 是	 是
AWS::Bedrock::Flow	 是	 是	 是
AWS::Bedrock::FlowAlias	 是	 是	 是
AWS::Bedrock::FlowVersion	 是	 是	 是
AWS::Bedrock::Guardrail	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Bedrock::GuardrailVersion</u>	 是	 是	 否
<u>AWS::Bedrock::IntelligentPromptRoute</u>	 是	 是	 是
<u>AWS::Bedrock::KnowledgeBase</u>	 是	 否	 是
<u>AWS::Bedrock::Prompt</u>	 是	 是	 是
<u>AWS::Bedrock::PromptVersion</u>	 是	 是	 是
<u>AWS::BedrockAgentCore::BrowserCustom</u>	 是	 是	 否
<u>AWS::BedrockAgentCore::BrowserProfile</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::BedrockAgentCore::CodeInterpreterCustom	 是	 是	 否
AWS::BedrockAgentCore::Evaluator	 是	 是	 否
AWS::BedrockAgentCore::Gateway	 是	 是	 否
AWS::BedrockAgentCore::GatewayTarget	 是	 是	 否
AWS::BedrockAgentCore::Memory	 是	 是	 否
AWS::BedrockAgentCore::OnlineEvaluationConfig	 是	 是	 否
AWS::BedrockAgentCore::Policy	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::BedrockAgentCore::PolicyEngine</u>	 是	 是	 否
<u>AWS::BedrockAgentCore::Runtime</u>	 是	 是	 否
<u>AWS::BedrockAgentCore::RuntimeEndpoint</u>	 是	 是	 否
<u>AWS::BedrockAgentCore::WorkloadIdentity</u>	 是	 是	 否
<u>AWS::BedrockMantle::Project</u>	 是	 是	 否
<u>AWS::Billing::BillingView</u>	 是	 是	 否
<u>AWS::BillingConductor::BillingGroup</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::BillingConductor::CustomLineItem</u>	 是	 是	 是
<u>AWS::BillingConductor::PricingPlan</u>	 是	 是	 是
<u>AWS::BillingConductor::PricingRule</u>	 是	 是	 是
<u>AWS::Budgets::BudgetsAction</u>	 是	 是	 是
<u>AWS::CE::AnomalyMonitor</u>	 是	 是	 否
<u>AWS::CE::AnomalySubscription</u>	 是	 是	 否
<u>AWS::CE::CostCategory</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::CUR::ReportDefinition</u>	 是	 是	 否
<u>AWS::Cases::CaseRule</u>	 是	 是	 否
<u>AWS::Cases::Domain</u>	 是	 是	 否
<u>AWS::Cases::Field</u>	 是	 是	 否
<u>AWS::Cases::Layout</u>	 是	 是	 否
<u>AWS::Cases::Template</u>	 是	 是	 否
<u>AWS::Cassandra::Keyspace</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Cassandra::Table</u>	 是	 是	 是
<u>AWS::Cassandra::Type</u>	 是	 是	 否
<u>AWS::CertificateManager::Account</u>	 是	 是	 否
<u>AWS::Chatbot::CustomAction</u>	 是	 是	 否
<u>AWS::Chatbot::MicrosoftTeamsChannelConfiguration</u>	 是	 是	 是
<u>AWS::Chatbot::SlackChannelConfiguration</u>	 是	 是	 是
<u>AWS::CleanRooms::AnalysisTemplate</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::CleanRooms::Collaboration</u>	 是	 是	 是
<u>AWS::CleanRooms::ConfiguredTable</u>	 是	 是	 是
<u>AWS::CleanRooms::ConfiguredTableAssociation</u>	 是	 是	 是
<u>AWS::CleanRooms::IdMappingTable</u>	 是	 是	 否
<u>AWS::CleanRooms::IdNamespaceAssociation</u>	 是	 是	 否
<u>AWS::CleanRooms::Membership</u>	 是	 是	 是
<u>AWS::CleanRooms::PrivacyBudgetTemplate</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CleanRoomsML::ConfiguredModelAlgorithm</u>	 是	 是	 否
<u>AWS::CleanRoomsML::ConfiguredModelAlgorithmAssociation</u>	 是	 是	 否
<u>AWS::CleanRoomsML::TrainingDataset</u>	 是	 是	 否
<u>AWS::CloudFormation::GuardHook</u>	 是	 是	 否
<u>AWS::CloudFormation::HookDefaultVersion</u>	 是	 是	 是
<u>AWS::CloudFormation::HookTypeConfig</u>	 是	 是	 是
<u>AWS::CloudFormation::HookVersion</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::CloudFormation::LambdaHook</u>	 是	 是	 否
<u>AWS::CloudFormation::ModuleDefaultVersion</u>	 是	 是	 否
<u>AWS::CloudFormation::ModuleVersion</u>	 是	 是	 否
<u>AWS::CloudFormation::PublicTypeVersion</u>	 是	 是	 否
<u>AWS::CloudFormation::Publisher</u>	 是	 是	 否
<u>AWS::CloudFormation::ResourceDefaultVersion</u>	 是	 是	 否
<u>AWS::CloudFormation::ResourceVersion</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CloudFormation::Stack</u>	 是	 否	 否
<u>AWS::CloudFormation::StackSet</u>	 是	 是	 否
<u>AWS::CloudFormation::TypeActivation</u>	 是	 是	 否
<u>AWS::CloudFront::AnycastIpList</u>	 是	 是	 否
<u>AWS::CloudFront::CachePolicy</u>	 是	 是	 是
<u>AWS::CloudFront::CloudFrontOriginAccessIdentity</u>	 是	 是	 是
<u>AWS::CloudFront::ConnectionFunction</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CloudFront::ConnectionGroup</u>	 是	 是	 否
<u>AWS::CloudFront::ContinuousDeploymentPolicy</u>	 是	 是	 是
<u>AWS::CloudFront::Distribution</u>	 是	 是	 是
<u>AWS::CloudFront::DistributionTenant</u>	 是	 是	 否
<u>AWS::CloudFront::Function</u>	 是	 是	 否
<u>AWS::CloudFront::KeyGroup</u>	 是	 是	 是
<u>AWS::CloudFront::KeyValueStore</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CloudFront::MonitoringSubscription</u>	 是	 是	 否
<u>AWS::CloudFront::OriginAccessControl</u>	 是	 是	 是
<u>AWS::CloudFront::OriginRequestPolicy</u>	 是	 是	 是
<u>AWS::CloudFront::PublicKey</u>	 是	 是	 是
<u>AWS::CloudFront::RealtimeLogConfig</u>	 是	 是	 是
<u>AWS::CloudFront::ResponseHeadersPolicy</u>	 是	 是	 否
<u>AWS::CloudFront::TrustStore</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CloudFront::VpcOrigin</u>	 是	 是	 否
<u>AWS::CloudTrail::Channel</u>	 是	 是	 否
<u>AWS::CloudTrail::Dashboard</u>	 是	 是	 否
<u>AWS::CloudTrail::EventDataStore</u>	 是	 是	 是
<u>AWS::CloudTrail::ResourcePolicy</u>	 是	 是	 否
<u>AWS::CloudTrail::Trail</u>	 是	 是	 是
<u>AWS::CloudWatch::Alarm</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CloudWatch::AlarmMuteRule</u>	 是	 是	 否
<u>AWS::CloudWatch::CompositeAlarm</u>	 是	 是	 是
<u>AWS::CloudWatch::Dashboard</u>	 是	 是	 否
<u>AWS::CloudWatch::MetricStream</u>	 是	 是	 否
<u>AWS::CodeArtifact::Domain</u>	 是	 是	 是
<u>AWS::CodeArtifact::PackageGroup</u>	 是	 是	 否
<u>AWS::CodeArtifact::Repository</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CodeBuild::Fleet</u>	 是	 是	 否
<u>AWS::CodeConnections::Connection</u>	 是	 是	 否
<u>AWS::CodeDeploy::Application</u>	 是	 是	 是
<u>AWS::CodeDeploy::DeploymentConfig</u>	 是	 是	 否
<u>AWS::CodeDeploy::DeploymentGroup</u>	 是	 是	 否
<u>AWS::CodeGuruProfiler::ProfilingGroup</u>	 是	 是	 是
<u>AWS::CodeGuruReviewer::RepositoryAssociation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::CodePipeline::CustomActionType</u>	 是	 是	 否
<u>AWS::CodePipeline::Pipeline</u>	 是	 是	 否
<u>AWS::CodePipeline::Webhook</u>	 是	 是	 否
<u>AWS::CodeStarConnections::Connection</u>	 是	 是	 是
<u>AWS::CodeStarConnections::Repository Link</u>	 是	 是	 否
<u>AWS::CodeStarConnections::SyncConfiguration</u>	 是	 是	 否
<u>AWS::CodeStarNotifications::NotificationRule</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Cognito::IdentityPool</u>	 是	 是	 否
<u>AWS::Cognito::IdentityPoolPrincipalTag</u>	 是	 是	 是
<u>AWS::Cognito::IdentityPoolRoleAttachment</u>	 是	 否	 否
<u>AWS::Cognito::LogDeliveryConfiguration</u>	 是	 是	 否
<u>AWS::Cognito::ManagedLoginBranding</u>	 是	 是	 否
<u>AWS::Cognito::Terms</u>	 是	 是	 否
<u>AWS::Cognito::UserPool</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Cognito::UserPoolClient</u>	 是	 是	 否
<u>AWS::Cognito::UserPoolDomain</u>	 是	 是	 否
<u>AWS::Cognito::UserPoolGroup</u>	 是	 是	 是
<u>AWS::Cognito::UserPoolIdentityProvider</u>	 是	 是	 否
<u>AWS::Cognito::UserPoolResourceServer</u>	 是	 是	 否
<u>AWS::Cognito::UserPoolRiskConfigurationAttachment</u>	 是	 是	 否
<u>AWS::Cognito::UserPoolUICustomizationAttachment</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Cognito::UserPoolUser</u>	 是	 否	 否
<u>AWS::Cognito::UserPoolUserToGroupAttachment</u>	 是	 是	 否
<u>AWS::Comprehend::DocumentClassifier</u>	 是	 是	 是
<u>AWS::Comprehend::Flywheel</u>	 是	 是	 是
<u>AWS::ComputeOptimizer::AutomationRule</u>	 是	 是	 否
<u>AWS::Config::AggregationAuthorization</u>	 是	 是	 是
<u>AWS::Config::ConfigRule</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Config::ConfigurationAggregator</u>	 是	 是	 是
<u>AWS::Config::ConformancePack</u>	 是	 是	 是
<u>AWS::Config::OrganizationConformancePack</u>	 是	 是	 是
<u>AWS::Config::StoredQuery</u>	 是	 是	 是
<u>AWS::Connect::AgentStatus</u>	 是	 是	 是
<u>AWS::Connect::ApprovedOrigin</u>	 是	 是	 是
<u>AWS::Connect::ContactFlow</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Connect::ContactFlowModule</u>	 是	 是	 是
<u>AWS::Connect::ContactFlowModuleAlias</u>	 是	 是	 否
<u>AWS::Connect::ContactFlowModuleVersion</u>	 是	 是	 否
<u>AWS::Connect::ContactFlowVersion</u>	 是	 是	 是
<u>AWS::Connect::DataTable</u>	 是	 是	 否
<u>AWS::Connect::DataTableAttribute</u>	 是	 是	 否
<u>AWS::Connect::DataTableRecord</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Connect::EmailAddress</u>	 是	 是	 是
<u>AWS::Connect::EvaluationForm</u>	 是	 是	 是
<u>AWS::Connect::HoursOfOperation</u>	 是	 是	 是
<u>AWS::Connect::Instance</u>	 是	 是	 是
<u>AWS::Connect::InstanceStorageConfig</u>	 是	 是	 是
<u>AWS::Connect::IntegrationAssociation</u>	 是	 是	 是
<u>AWS::Connect::Notification</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Connect::PhoneNumber</u>	 是	 是	 是
<u>AWS::Connect::PredefinedAttribute</u>	 是	 是	 是
<u>AWS::Connect::Prompt</u>	 是	 是	 是
<u>AWS::Connect::Queue</u>	 是	 是	 是
<u>AWS::Connect::QuickConnect</u>	 是	 是	 是
<u>AWS::Connect::RoutingProfile</u>	 是	 是	 是
<u>AWS::Connect::Rule</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Connect::SecurityKey</u>	 是	 是	 是
<u>AWS::Connect::SecurityProfile</u>	 是	 是	 是
<u>AWS::Connect::TaskTemplate</u>	 是	 是	 是
<u>AWS::Connect::TrafficDistributionGroup</u>	 是	 是	 是
<u>AWS::Connect::User</u>	 是	 是	 是
<u>AWS::Connect::UserHierarchyGroup</u>	 是	 是	 是
<u>AWS::Connect::UserHierarchyStructure</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Connect::View</u>	 是	 是	 是
<u>AWS::Connect::ViewVersion</u>	 是	 是	 是
<u>AWS::Connect::Workspace</u>	 是	 是	 否
<u>AWS::ConnectCampaigns::Campaign</u>	 是	 否	 是
<u>AWS::ConnectCampaignsV2::Campaign</u>	 是	 是	 是
<u>AWS::ControlTower::EnabledBaseline</u>	 是	 是	 否
<u>AWS::ControlTower::EnabledControl</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ControlTower::LandingZone</u>	 是	 是	 否
<u>AWS::CustomerProfiles::CalculatedAttributeDefinition</u>	 是	 是	 是
<u>AWS::CustomerProfiles::Domain</u>	 是	 是	 是
<u>AWS::CustomerProfiles::EventStream</u>	 是	 是	 是
<u>AWS::CustomerProfiles::EventTrigger</u>	 是	 是	 否
<u>AWS::CustomerProfiles::Integration</u>	 是	 是	 是
<u>AWS::CustomerProfiles::ObjectType</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::CustomerProfiles::SegmentDefinition</u>	 是	 是	 否
<u>AWS::DMS::DataMigration</u>	 是	 是	 否
<u>AWS::DMS::DataProvider</u>	 是	 是	 否
<u>AWS::DMS::InstanceProfile</u>	 是	 是	 否
<u>AWS::DMS::MigrationProject</u>	 是	 是	 否
<u>AWS::DMS::ReplicationConfig</u>	 是	 是	 是
<u>AWS::DSQL::Cluster</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::DataBrew::Dataset</u>	 是	 是	 是
<u>AWS::DataBrew::Job</u>	 是	 是	 是
<u>AWS::DataBrew::Project</u>	 是	 是	 是
<u>AWS::DataBrew::Recipe</u>	 是	 是	 是
<u>AWS::DataBrew::Ruleset</u>	 是	 是	 是
<u>AWS::DataBrew::Schedule</u>	 是	 是	 是
<u>AWS::DataPipeline::Pipeline</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::DataSync::Agent</u>	 是	 是	 是
<u>AWS::DataSync::LocationAzureBlob</u>	 是	 是	 是
<u>AWS::DataSync::LocationEFS</u>	 是	 是	 是
<u>AWS::DataSync::LocationFSxLustre</u>	 是	 是	 是
<u>AWS::DataSync::LocationFSxONTAP</u>	 是	 是	 是
<u>AWS::DataSync::LocationFSxOpenZFS</u>	 是	 是	 是
<u>AWS::DataSync::LocationFSxWindows</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::DataSync::LocationHDFS</u>	 是	 是	 是
<u>AWS::DataSync::LocationNFS</u>	 是	 是	 是
<u>AWS::DataSync::LocationObjectStorage</u>	 是	 是	 是
<u>AWS::DataSync::LocationS3</u>	 是	 是	 是
<u>AWS::DataSync::LocationSMB</u>	 是	 是	 是
<u>AWS::DataSync::Task</u>	 是	 是	 是
<u>AWS::DataZone::Connection</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::DataZone::DataSource</u>	 是	 是	 否
<u>AWS::DataZone::Domain</u>	 是	 是	 否
<u>AWS::DataZone::DomainUnit</u>	 是	 是	 否
<u>AWS::DataZone::Environment</u>	 是	 是	 否
<u>AWS::DataZone::EnvironmentActions</u>	 是	 是	 否
<u>AWS::DataZone::EnvironmentBlueprintConfiguration</u>	 是	 是	 否
<u>AWS::DataZone::EnvironmentProfile</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::DataZone::FormType</u>	 是	 是	 否
<u>AWS::DataZone::GroupProfile</u>	 是	 是	 否
<u>AWS::DataZone::Owner</u>	 是	 是	 否
<u>AWS::DataZone::PolicyGrant</u>	 是	 是	 否
<u>AWS::DataZone::Project</u>	 是	 是	 否
<u>AWS::DataZone::ProjectMembership</u>	 是	 是	 否
<u>AWS::DataZone::ProjectProfile</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::DataZone::SubscriptionTarget</u>	 是	 是	 否
<u>AWS::DataZone::UserProfile</u>	 是	 是	 否
<u>AWS::Deadline::Farm</u>	 是	 是	 是
<u>AWS::Deadline::Fleet</u>	 是	 是	 是
<u>AWS::Deadline::LicenseEndpoint</u>	 是	 是	 是
<u>AWS::Deadline::Limit</u>	 是	 是	 是
<u>AWS::Deadline::MeteredProduct</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Deadline::Monitor</u>	 是	 是	 是
<u>AWS::Deadline::Queue</u>	 是	 是	 是
<u>AWS::Deadline::QueueEnvironment</u>	 是	 是	 否
<u>AWS::Deadline::QueueFleetAssociation</u>	 是	 是	 否
<u>AWS::Deadline::QueueLimitAssociation</u>	 是	 是	 否
<u>AWS::Deadline::StorageProfile</u>	 是	 是	 是
<u>AWS::Detective::Graph</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Detective::MemberInvitation</u>	 是	 是	 是
<u>AWS::Detective::OrganizationAdmin</u>	 是	 是	 是
<u>AWS::DevOpsAgent::AgentSpace</u>	 是	 是	 否
<u>AWS::DevOpsAgent::Association</u>	 是	 是	 否
<u>AWS::DevOpsAgent::Service</u>	 是	 是	 否
<u>AWS::DevOpsGuru::LogAnomalyDetectionIntegration</u>	 是	 是	 是
<u>AWS::DevOpsGuru::NotificationChannel</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::DevOpsGuru::ResourceCollection	 是	 是	 是
AWS::DeviceFarm::DevicePool	 是	 是	 否
AWS::DeviceFarm::InstanceProfile	 是	 是	 否
AWS::DeviceFarm::NetworkProfile	 是	 是	 否
AWS::DeviceFarm::Project	 是	 是	 否
AWS::DeviceFarm::TestGridProject	 是	 是	 否
AWS::DeviceFarm::VPCEConfiguration	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::DirectConnect::Connection</u>	 是	 是	 否
<u>AWS::DirectConnect::DirectConnectGateway</u>	 是	 是	 否
<u>AWS::DirectConnect::DirectConnectGatewayAssociation</u>	 是	 是	 否
<u>AWS::DirectConnect::Lag</u>	 是	 是	 否
<u>AWS::DirectConnect::PrivateVirtualInterface</u>	 是	 是	 否
<u>AWS::DirectConnect::PublicVirtualInterface</u>	 是	 是	 否
<u>AWS::DirectConnect::TransitVirtualInterface</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::DirectoryService::SimpleAD</u>	 是	 是	 是
<u>AWS::DocDB::GlobalCluster</u>	 是	 是	 否
<u>AWS::DocDBElastic::Cluster</u>	 是	 是	 是
<u>AWS::DynamoDB::GlobalTable</u>	 是	 是	 否
<u>AWS::DynamoDB::Table</u>	 是	 是	 是
<u>AWS::EC2::CapacityManagerDataExport</u>	 是	 是	 否
<u>AWS::EC2::CapacityReservation</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::EC2::CapacityReservationFleet	 是	 是	 是
AWS::EC2::CarrierGateway	 是	 是	 是
AWS::EC2::CustomerGateway	 是	 是	 是
AWS::EC2::DHCPOptions	 是	 是	 是
AWS::EC2::EC2Fleet	 是	 是	 否
AWS::EC2::EIP	 是	 是	 是
AWS::EC2::EIPAssociation	 是	 否	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::EgressOnlyInternetGateway</u>	 是	 是	 是
<u>AWS::EC2::EnclaveCertificateIamRoleAssociation</u>	 是	 是	 是
<u>AWS::EC2::FlowLog</u>	 是	 是	 是
<u>AWS::EC2::GatewayRouteTableAssociation</u>	 是	 是	 否
<u>AWS::EC2::Host</u>	 是	 是	 是
<u>AWS::EC2::IPAM</u>	 是	 是	 是
<u>AWS::EC2::IPAMAllocation</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::IPAMPool</u>	 是	 是	 是
<u>AWS::EC2::IPAMPoolCidr</u>	 是	 是	 是
<u>AWS::EC2::IPAMPrefixListResolver</u>	 是	 是	 否
<u>AWS::EC2::IPAMPrefixListResolverTarget</u>	 是	 是	 否
<u>AWS::EC2::IPAMResourceDiscovery</u>	 是	 是	 是
<u>AWS::EC2::IPAMResourceDiscoveryAssociation</u>	 是	 是	 是
<u>AWS::EC2::IPAMScope</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::EC2::Instance	 是	 是	 是
AWS::EC2::InstanceConnectEndpoint	 是	 是	 是
AWS::EC2::InternetGateway	 是	 是	 是
AWS::EC2::IpPoolRouteTableAssociation	 是	 是	 否
AWS::EC2::KeyPair	 是	 是	 是
AWS::EC2::LaunchTemplate	 是	 是	 否
AWS::EC2::LocalGatewayRoute	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::LocalGatewayRouteTable</u>	 是	 是	 是
<u>AWS::EC2::LocalGatewayRouteTableVPCAssociation</u>	 是	 是	 是
<u>AWS::EC2::LocalGatewayRouteTableVirtualInterfaceGroupAssociation</u>	 是	 是	 是
<u>AWS::EC2::LocalGatewayVirtualInterface</u>	 是	 是	 否
<u>AWS::EC2::LocalGatewayVirtualInterfaceGroup</u>	 是	 是	 否
<u>AWS::EC2::NatGateway</u>	 是	 是	 是
<u>AWS::EC2::NetworkAcl</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::NetworkInsightsAccessScope</u>	 是	 是	 否
<u>AWS::EC2::NetworkInsightsAccessScopeAnalysis</u>	 是	 是	 是
<u>AWS::EC2::NetworkInsightsAnalysis</u>	 是	 是	 否
<u>AWS::EC2::NetworkInsightsPath</u>	 是	 是	 是
<u>AWS::EC2::NetworkInterface</u>	 是	 是	 是
<u>AWS::EC2::NetworkInterfaceAttachment</u>	 是	 是	 是
<u>AWS::EC2::NetworkPerformanceMetricSubscription</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::EC2::PlacementGroup	 是	 是	 是
AWS::EC2::PrefixList	 是	 是	 否
AWS::EC2::Route	 是	 是	 是
AWS::EC2::RouteServer	 是	 是	 否
AWS::EC2::RouteServerAssociation	 是	 是	 否
AWS::EC2::RouteServerEndpoint	 是	 是	 否
AWS::EC2::RouteServerPeer	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::RouteServerPropagation</u>	 是	 是	 否
<u>AWS::EC2::RouteTable</u>	 是	 是	 是
<u>AWS::EC2::SecurityGroup</u>	 是	 是	 是
<u>AWS::EC2::SecurityGroupEgress</u>	 是	 否	 否
<u>AWS::EC2::SecurityGroupIngress</u>	 是	 否	 否
<u>AWS::EC2::SecurityGroupVpcAssociation</u>	 是	 是	 否
<u>AWS::EC2::SnapshotBlockPublicAccess</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::SpotFleet</u>	 是	 是	 否
<u>AWS::EC2::Subnet</u>	 是	 是	 是
<u>AWS::EC2::SubnetCidrBlock</u>	 是	 否	 是
<u>AWS::EC2::SubnetNetworkAclAssociation</u>	 是	 是	 是
<u>AWS::EC2::SubnetRouteTableAssociation</u>	 是	 否	 是
<u>AWS::EC2::TrafficMirrorFilter</u>	 是	 是	 否
<u>AWS::EC2::TrafficMirrorFilterRule</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::TrafficMirrorSession</u>	 是	 是	 否
<u>AWS::EC2::TrafficMirrorTarget</u>	 是	 是	 否
<u>AWS::EC2::TransitGateway</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayAttachment</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayConnect</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayConnectPeer</u>	 是	 是	 否
<u>AWS::EC2::TransitGatewayMeteringPolicy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::TransitGatewayMeteringPolicyEntry</u>	 是	 是	 否
<u>AWS::EC2::TransitGatewayMulticastDomain</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayMulticastDomainAssociation</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayMulticastGroupMember</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayMulticastGroupSource</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayPeeringAttachment</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayRoute</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::TransitGatewayRouteTable</u>	 是	 是	 是
<u>AWS::EC2::TransitGatewayRouteTableAssociation</u>	 是	 是	 否
<u>AWS::EC2::TransitGatewayRouteTablePropagation</u>	 是	 是	 否
<u>AWS::EC2::TransitGatewayVpcAttachment</u>	 是	 是	 是
<u>AWS::EC2::VPC</u>	 是	 是	 是
<u>AWS::EC2::VPCBlockPublicAccessExclusion</u>	 是	 是	 否
<u>AWS::EC2::VPCBlockPublicAccessOptions</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::VPCCidrBlock</u>	 是	 是	 否
<u>AWS::EC2::VPCDHCPOptionsAssociation</u>	 是	 是	 是
<u>AWS::EC2::VPCEncryptionControl</u>	 是	 是	 否
<u>AWS::EC2::VPCEndpoint</u>	 是	 否	 是
<u>AWS::EC2::VPCEndpointConnectionNotification</u>	 是	 是	 是
<u>AWS::EC2::VPCEndpointService</u>	 是	 是	 是
<u>AWS::EC2::VPCEndpointServicePermissions</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::EC2::VPCGatewayAttachment	 是	 否	 否
AWS::EC2::VPCPeeringConnection	 是	 是	 是
AWS::EC2::VPNConcentrator	 是	 是	 否
AWS::EC2::VPNConnection	 是	 是	 是
AWS::EC2::VPNConnectionRoute	 是	 是	 是
AWS::EC2::VPNGateway	 是	 是	 是
AWS::EC2::VerifiedAccessEndpoint	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EC2::VerifiedAccessGroup</u>	 是	 是	 是
<u>AWS::EC2::VerifiedAccessInstance</u>	 是	 是	 是
<u>AWS::EC2::VerifiedAccessTrustProvider</u>	 是	 是	 是
<u>AWS::EC2::Volume</u>	 是	 是	 是
<u>AWS::EC2::VolumeAttachment</u>	 是	 是	 是
<u>AWS::ECR::PublicRepository</u>	 是	 是	 是
<u>AWS::ECR::PullThroughCacheRule</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ECR::PullTimeUpdateExclusion</u>	 是	 是	 否
<u>AWS::ECR::RegistryPolicy</u>	 是	 是	 是
<u>AWS::ECR::RegistryScanningConfiguration</u>	 是	 是	 否
<u>AWS::ECR::ReplicationConfiguration</u>	 是	 是	 是
<u>AWS::ECR::Repository</u>	 是	 是	 是
<u>AWS::ECR::RepositoryCreationTemplate</u>	 是	 是	 否
<u>AWS::ECR::SigningConfiguration</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ECS::CapacityProvider</u>	 是	 是	 否
<u>AWS::ECS::Cluster</u>	 是	 是	 是
<u>AWS::ECS::ClusterCapacityProviderAssociations</u>	 是	 否	 是
<u>AWS::ECS::ExpressGatewayService</u>	 是	 是	 否
<u>AWS::ECS::PrimaryTaskSet</u>	 是	 是	 否
<u>AWS::ECS::Service</u>	 是	 是	 是
<u>AWS::ECS::TaskDefinition</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::ECS::TaskSet	 是	 是	 否
AWS::EFS::AccessPoint	 是	 是	 是
AWS::EFS::FileSystem	 是	 是	 是
AWS::EFS::MountTarget	 是	 是	 是
AWS::EKS::AccessEntry	 是	 是	 否
AWS::EKS::Addon	 是	 是	 是
AWS::EKS::Capability	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EKS::Cluster</u>	 是	 否	 是
<u>AWS::EKS::FargateProfile</u>	 是	 是	 是
<u>AWS::EKS::IdentityProviderConfig</u>	 是	 是	 是
<u>AWS::EKS::Nodegroup</u>	 是	 否	 是
<u>AWS::EKS::PodIdentityAssociation</u>	 是	 是	 否
<u>AWS::EMR::SecurityConfiguration</u>	 是	 是	 是
<u>AWS::EMR::Step</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EMR::Studio</u>	 是	 是	 是
<u>AWS::EMR::StudioSessionMapping</u>	 是	 是	 是
<u>AWS::EMR::WALWorkspace</u>	 是	 是	 否
<u>AWS::EMRContainers::Endpoint</u>	 是	 是	 否
<u>AWS::EMRContainers::SecurityConfiguration</u>	 是	 是	 否
<u>AWS::EMRContainers::VirtualCluster</u>	 是	 是	 否
<u>AWS::EMRServerless::Application</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::EVS::Environment</u>	 是	 是	 否
<u>AWS::ElastiCache::GlobalReplicationGroup</u>	 是	 是	 否
<u>AWS::ElastiCache::ParameterGroup</u>	 是	 是	 否
<u>AWS::ElastiCache::ReplicationGroup</u>	 是	 是	 否
<u>AWS::ElastiCache::ServerlessCache</u>	 是	 是	 否
<u>AWS::ElastiCache::SubnetGroup</u>	 是	 是	 是
<u>AWS::ElastiCache::User</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ElastiCache::UserGroup</u>	 是	 是	 是
<u>AWS::ElasticBeanstalk::Application</u>	 是	 是	 是
<u>AWS::ElasticBeanstalk::ApplicationVersion</u>	 是	 是	 是
<u>AWS::ElasticBeanstalk::ConfigurationTemplate</u>	 是	 是	 是
<u>AWS::ElasticBeanstalk::Environment</u>	 是	 是	 否
<u>AWS::ElasticLoadBalancing::LoadBalancer</u>	 是	 是	 否
<u>AWS::ElasticLoadBalancingV2::Listener</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ElasticLoadBalancingV2::ListenerRule</u>	 是	 是	 是
<u>AWS::ElasticLoadBalancingV2::LoadBalancer</u>	 是	 是	 是
<u>AWS::ElasticLoadBalancingV2::TargetGroup</u>	 是	 是	 是
<u>AWS::ElasticLoadBalancingV2::TrustStore</u>	 是	 是	 否
<u>AWS::ElasticLoadBalancingV2::TrustStoreRevocation</u>	 是	 是	 否
<u>AWS::ElementalInference::Feed</u>	 是	 是	 否
<u>AWS::EntityResolution::IdMappingWorkflow</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EntityResolution::IdNamespace</u>	 是	 是	 否
<u>AWS::EntityResolution::MatchingWorkflow</u>	 是	 是	 是
<u>AWS::EntityResolution::PolicyStatement</u>	 是	 是	 否
<u>AWS::EntityResolution::SchemaMapping</u>	 是	 是	 是
<u>AWS::EventSchemas::Discoverer</u>	 是	 是	 否
<u>AWS::EventSchemas::Registry</u>	 是	 是	 否
<u>AWS::EventSchemas::RegistryPolicy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::EventSchemas::Schema</u>	 是	 是	 否
<u>AWS::Events::ApiDestination</u>	 是	 是	 是
<u>AWS::Events::Archive</u>	 是	 是	 是
<u>AWS::Events::Connection</u>	 是	 是	 否
<u>AWS::Events::Endpoint</u>	 是	 是	 是
<u>AWS::Events::EventBus</u>	 是	 是	 是
<u>AWS::Events::EventBusPolicy</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Events::Rule</u>	 是	 是	 是
<u>AWS::Evidently::Experiment</u>	 是	 是	 否
<u>AWS::Evidently::Feature</u>	 是	 否	 否
<u>AWS::Evidently::Launch</u>	 是	 是	 否
<u>AWS::Evidently::Project</u>	 是	 是	 否
<u>AWS::Evidently::Segment</u>	 是	 是	 否
<u>AWS::FIS::ExperimentTemplate</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::FIS::TargetAccountConfiguration</u>	 是	 是	 否
<u>AWS::FMS::NotificationChannel</u>	 是	 是	 否
<u>AWS::FMS::Policy</u>	 是	 否	 否
<u>AWS::FMS::ResourceSet</u>	 是	 是	 否
<u>AWS::FSx::DataRepositoryAssociation</u>	 是	 是	 是
<u>AWS::FSx::S3AccessPointAttachment</u>	 是	 是	 否
<u>AWS::FinSpace::Environment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Forecast::Dataset</u>	 是	 是	 否
<u>AWS::Forecast::DatasetGroup</u>	 是	 是	 否
<u>AWS::FraudDetector::Detector</u>	 是	 否	 是
<u>AWS::FraudDetector::EntityType</u>	 是	 是	 是
<u>AWS::FraudDetector::EventType</u>	 是	 是	 是
<u>AWS::FraudDetector::Label</u>	 是	 是	 是
<u>AWS::FraudDetector::List</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::FraudDetector::Outcome</u>	 是	 是	 是
<u>AWS::FraudDetector::Variable</u>	 是	 是	 是
<u>AWS::GameLift::Alias</u>	 是	 是	 是
<u>AWS::GameLift::Build</u>	 是	 是	 是
<u>AWS::GameLift::ContainerFleet</u>	 是	 是	 否
<u>AWS::GameLift::ContainerGroupDefinition</u>	 是	 是	 否
<u>AWS::GameLift::Fleet</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::GameLift::GameServerGroup</u>	 是	 是	 否
<u>AWS::GameLift::GameSessionQueue</u>	 是	 是	 否
<u>AWS::GameLift::Location</u>	 是	 是	 否
<u>AWS::GameLift::MatchmakingConfiguration</u>	 是	 是	 否
<u>AWS::GameLift::MatchmakingRuleSet</u>	 是	 是	 否
<u>AWS::GameLift::Script</u>	 是	 是	 否
<u>AWS::GameLiftStreams::Application</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::GameLiftStreams::StreamGroup</u>	 是	 是	 否
<u>AWS::GlobalAccelerator::Accelerator</u>	 是	 是	 是
<u>AWS::GlobalAccelerator::CrossAccountAttachment</u>	 是	 是	 否
<u>AWS::GlobalAccelerator::EndpointGroup</u>	 是	 是	 是
<u>AWS::GlobalAccelerator::Listener</u>	 是	 是	 是
<u>AWS::Glue::Catalog</u>	 是	 是	 否
<u>AWS::Glue::Crawler</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Glue::Database</u>	 是	 是	 否
<u>AWS::Glue::IdentityCenterConfiguration</u>	 是	 是	 否
<u>AWS::Glue::Integration</u>	 是	 是	 否
<u>AWS::Glue::IntegrationResourceProperty</u>	 是	 是	 否
<u>AWS::Glue::Job</u>	 是	 是	 否
<u>AWS::Glue::Registry</u>	 是	 是	 是
<u>AWS::Glue::Schema</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Glue::SchemaVersion</u>	 是	 是	 是
<u>AWS::Glue::SchemaVersionMetadata</u>	 是	 是	 是
<u>AWS::Glue::Trigger</u>	 是	 是	 否
<u>AWS::Glue::UsageProfile</u>	 是	 是	 否
<u>AWS::Grafana::Workspace</u>	 是	 是	 是
<u>AWS::GreengrassV2::ComponentVersion</u>	 是	 是	 否
<u>AWS::GreengrassV2::Deployment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::GroundStation::Config</u>	 是	 是	 是
<u>AWS::GroundStation::DataflowEndpointGroup</u>	 是	 是	 是
<u>AWS::GroundStation::DataflowEndpointGroupV2</u>	 是	 是	 否
<u>AWS::GroundStation::MissionProfile</u>	 是	 是	 是
<u>AWS::GuardDuty::Detector</u>	 是	 是	 是
<u>AWS::GuardDuty::Filter</u>	 是	 是	 否
<u>AWS::GuardDuty::IPSet</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::GuardDuty::MalwareProtectionPlan</u>	 是	 是	 否
<u>AWS::GuardDuty::Master</u>	 是	 是	 否
<u>AWS::GuardDuty::Member</u>	 是	 是	 否
<u>AWS::GuardDuty::PublishingDestination</u>	 是	 是	 否
<u>AWS::GuardDuty::ThreatEntitySet</u>	 是	 是	 否
<u>AWS::GuardDuty::ThreatIntelSet</u>	 是	 是	 否
<u>AWS::GuardDuty::TrustedEntitySet</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::HealthImaging::Datastore	 是	 是	 是
AWS::HealthLake::FHIRDatastore	 是	 是	 是
AWS::IAM::Group	 是	 是	 是
AWS::IAM::GroupPolicy	 是	 是	 否
AWS::IAM::InstanceProfile	 是	 是	 是
AWS::IAM::ManagedPolicy	 是	 是	 是
AWS::IAM::OIDCProvider	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::IAM::Role	 是	 是	 是
AWS::IAM::RolePolicy	 是	 是	 否
AWS::IAM::SAMLProvider	 是	 是	 是
AWS::IAM::ServerCertificate	 是	 是	 是
AWS::IAM::ServiceLinkedRole	 是	 是	 否
AWS::IAM::User	 是	 是	 是
AWS::IAM::UserPolicy	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IAM::VirtualMFADevice</u>	 是	 是	 否
<u>AWS::IVS::Channel</u>	 是	 是	 是
<u>AWS::IVS::EncoderConfiguration</u>	 是	 是	 否
<u>AWS::IVS::IngestConfiguration</u>	 是	 是	 否
<u>AWS::IVS::PlaybackKeyPair</u>	 是	 是	 是
<u>AWS::IVS::PlaybackRestrictionPolicy</u>	 是	 是	 否
<u>AWS::IVS::PublicKey</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IVS::RecordingConfiguration</u>	 是	 是	 是
<u>AWS::IVS::Stage</u>	 是	 是	 否
<u>AWS::IVS::StorageConfiguration</u>	 是	 是	 否
<u>AWS::IVS::StreamKey</u>	 是	 是	 否
<u>AWS::IVSChat::LoggingConfiguration</u>	 是	 是	 是
<u>AWS::IVSChat::Room</u>	 是	 是	 是
<u>AWS::IdentityStore::Group</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::IdentityStore::GroupMembership	 是	 是	 是
AWS::ImageBuilder::Component	 是	 是	 是
AWS::ImageBuilder::ContainerRecipe	 是	 否	 否
AWS::ImageBuilder::DistributionConfiguration	 是	 是	 是
AWS::ImageBuilder::Image	 是	 是	 是
AWS::ImageBuilder::ImagePipeline	 是	 是	 是
AWS::ImageBuilder::ImageRecipe	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ImageBuilder::InfrastructureConfiguration</u>	 是	 是	 是
<u>AWS::ImageBuilder::LifecyclePolicy</u>	 是	 是	 否
<u>AWS::ImageBuilder::Workflow</u>	 是	 是	 否
<u>AWS::Inspector::AssessmentTarget</u>	 是	 是	 是
<u>AWS::Inspector::AssessmentTemplate</u>	 是	 是	 是
<u>AWS::Inspector::ResourceGroup</u>	 是	 是	 否
<u>AWS::InspectorV2::CisScanConfiguration</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::InspectorV2::CodeSecurityIntegration</u>	 是	 是	 否
<u>AWS::InspectorV2::CodeSecurityScanConfiguration</u>	 是	 是	 否
<u>AWS::InspectorV2::Filter</u>	 是	 是	 否
<u>AWS::InternetMonitor::Monitor</u>	 是	 是	 是
<u>AWS::Invoicing::InvoiceUnit</u>	 是	 是	 否
<u>AWS::IoT::AccountAuditConfiguration</u>	 是	 是	 是
<u>AWS::IoT::Authorizer</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoT::BillingGroup</u>	 是	 是	 是
<u>AWS::IoT::CACertificate</u>	 是	 是	 是
<u>AWS::IoT::Certificate</u>	 是	 是	 是
<u>AWS::IoT::CertificateProvider</u>	 是	 是	 否
<u>AWS::IoT::Command</u>	 是	 是	 否
<u>AWS::IoT::CustomMetric</u>	 是	 是	 是
<u>AWS::IoT::Dimension</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoT::DomainConfiguration</u>	 是	 是	 否
<u>AWS::IoT::EncryptionConfiguration</u>	 是	 是	 否
<u>AWS::IoT::FleetMetric</u>	 是	 是	 否
<u>AWS::IoT::JobTemplate</u>	 是	 是	 否
<u>AWS::IoT::Logging</u>	 是	 是	 是
<u>AWS::IoT::MitigationAction</u>	 是	 是	 是
<u>AWS::IoT::Policy</u>	 是	 否	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoT::ProvisioningTemplate</u>	 是	 是	 是
<u>AWS::IoT::ResourceSpecificLogging</u>	 是	 是	 是
<u>AWS::IoT::RoleAlias</u>	 是	 是	 是
<u>AWS::IoT::ScheduledAudit</u>	 是	 是	 是
<u>AWS::IoT::SecurityProfile</u>	 是	 否	 是
<u>AWS::IoT::SoftwarePackage</u>	 是	 是	 是
<u>AWS::IoT::SoftwarePackageVersion</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoT::Thing</u>	 是	 是	 是
<u>AWS::IoT::ThingGroup</u>	 是	 是	 是
<u>AWS::IoT::ThingType</u>	 是	 是	 否
<u>AWS::IoT::TopicRule</u>	 是	 是	 是
<u>AWS::IoT::TopicRuleDestination</u>	 是	 是	 是
<u>AWS::IoTAnalytics::Channel</u>	 是	 是	 是
<u>AWS::IoTAnalytics::Dataset</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoTAnalytics::Datastore</u>	 是	 是	 是
<u>AWS::IoTAnalytics::Pipeline</u>	 是	 是	 是
<u>AWS::IoTCoreDeviceAdvisor::SuiteDefinition</u>	 是	 是	 是
<u>AWS::IoTEvents::AlarmModel</u>	 是	 是	 是
<u>AWS::IoTEvents::DetectorModel</u>	 是	 是	 是
<u>AWS::IoTEvents::Input</u>	 是	 是	 是
<u>AWS::IoT FleetWise::Campaign</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoT FleetWise::DecoderManifest</u>	 是	 是	 是
<u>AWS::IoT FleetWise::Fleet</u>	 是	 是	 是
<u>AWS::IoT FleetWise::ModelManifest</u>	 是	 否	 是
<u>AWS::IoT FleetWise::SignalCatalog</u>	 是	 否	 是
<u>AWS::IoT FleetWise::StateTemplate</u>	 是	 是	 否
<u>AWS::IoT FleetWise::Vehicle</u>	 是	 是	 是
<u>AWS::IoT SiteWise::AccessPolicy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoTSiteWise::Asset</u>	 是	 是	 是
<u>AWS::IoTSiteWise::AssetModel</u>	 是	 是	 是
<u>AWS::IoTSiteWise::ComputationModel</u>	 是	 是	 否
<u>AWS::IoTSiteWise::Dashboard</u>	 是	 是	 否
<u>AWS::IoTSiteWise::Dataset</u>	 是	 是	 否
<u>AWS::IoTSiteWise::Gateway</u>	 是	 是	 是
<u>AWS::IoTSiteWise::Portal</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::IoTSiteWise::Project	 是	 是	 否
AWS::IoTTwinMaker::ComponentType	 是	 是	 是
AWS::IoTTwinMaker::Entity	 是	 是	 是
AWS::IoTTwinMaker::Scene	 是	 是	 是
AWS::IoTTwinMaker::SyncJob	 是	 是	 是
AWS::IoTTwinMaker::Workspace	 是	 是	 是
AWS::IoTWireless::Destination	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoTWireless::DeviceProfile</u>	 是	 是	 是
<u>AWS::IoTWireless::FwotaTask</u>	 是	 是	 是
<u>AWS::IoTWireless::MulticastGroup</u>	 是	 是	 是
<u>AWS::IoTWireless::NetworkAnalyzerConfiguration</u>	 是	 是	 是
<u>AWS::IoTWireless::PartnerAccount</u>	 是	 是	 是
<u>AWS::IoTWireless::ServiceProfile</u>	 是	 是	 是
<u>AWS::IoTWireless::TaskDefinition</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::IoTWireless::WirelessDevice</u>	 是	 是	 是
<u>AWS::IoTWireless::WirelessDeviceImportTask</u>	 是	 是	 是
<u>AWS::IoTWireless::WirelessGateway</u>	 是	 是	 是
<u>AWS::KMS::Alias</u>	 是	 否	 是
<u>AWS::KMS::Key</u>	 是	 是	 是
<u>AWS::KMS::ReplicaKey</u>	 是	 否	 是
<u>AWS::KafkaConnect::Connector</u>	 是	 否	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::KafkaConnect::CustomPlugin</u>	 是	 是	 否
<u>AWS::KafkaConnect::WorkerConfiguration</u>	 是	 是	 否
<u>AWS::Kendra::DataSource</u>	 是	 是	 是
<u>AWS::Kendra::FAQ</u>	 是	 是	 是
<u>AWS::Kendra::Index</u>	 是	 是	 是
<u>AWS::KendraRanking::ExecutionPlan</u>	 是	 是	 是
<u>AWS::Kinesis::ResourcePolicy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Kinesis::Stream</u>	 是	 是	 是
<u>AWS::Kinesis::StreamConsumer</u>	 是	 是	 否
<u>AWS::KinesisAnalyticsV2::Application</u>	 是	 是	 否
<u>AWS::KinesisFirehose::DeliveryStream</u>	 是	 否	 否
<u>AWS::KinesisVideo::SignalingChannel</u>	 是	 是	 否
<u>AWS::KinesisVideo::Stream</u>	 是	 是	 否
<u>AWS::LakeFormation::DataCellsFilter</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::LakeFormation::PrincipalPermissions	 是	 是	 否
AWS::LakeFormation::Tag	 是	 是	 否
AWS::LakeFormation::TagAssociation	 是	 是	 否
AWS::Lambda::Alias	 是	 是	 否
AWS::Lambda::CapacityProvider	 是	 是	 否
AWS::Lambda::CodeSigningConfig	 是	 是	 是
AWS::Lambda::EventInvokeConfig	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Lambda::EventSourceMapping</u>	 是	 否	 是
<u>AWS::Lambda::Function</u>	 是	 是	 是
<u>AWS::Lambda::LayerVersion</u>	 是	 是	 否
<u>AWS::Lambda::LayerVersionPermission</u>	 是	 是	 是
<u>AWS::Lambda::Permission</u>	 是	 是	 是
<u>AWS::Lambda::Url</u>	 是	 是	 是
<u>AWS::Lambda::Version</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::LaunchWizard::Deployment</u>	 是	 是	 否
<u>AWS::Lex::Bot</u>	 是	 是	 否
<u>AWS::Lex::BotAlias</u>	 是	 是	 否
<u>AWS::Lex::BotVersion</u>	 是	 是	 否
<u>AWS::Lex::ResourcePolicy</u>	 是	 是	 否
<u>AWS::LicenseManager::Grant</u>	 是	 是	 否
<u>AWS::LicenseManager::License</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::Lightsail::Alarm	 是	 是	 是
AWS::Lightsail::Bucket	 是	 是	 是
AWS::Lightsail::Certificate	 是	 是	 是
AWS::Lightsail::Container	 是	 是	 是
AWS::Lightsail::Database	 是	 是	 否
AWS::Lightsail::DatabaseSnapshot	 是	 是	 否
AWS::Lightsail::Disk	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Lightsail::DiskSnapshot</u>	 是	 是	 否
<u>AWS::Lightsail::Distribution</u>	 是	 是	 是
<u>AWS::Lightsail::Domain</u>	 是	 是	 否
<u>AWS::Lightsail::Instance</u>	 是	 是	 是
<u>AWS::Lightsail::InstanceSnapshot</u>	 是	 是	 否
<u>AWS::Lightsail::LoadBalancer</u>	 是	 是	 是
<u>AWS::Lightsail::LoadBalancerTlsCertificate</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Lightsail::StaticIp</u>	 是	 是	 是
<u>AWS::Location::APIKey</u>	 是	 是	 否
<u>AWS::Location::GeofenceCollection</u>	 是	 是	 是
<u>AWS::Location::Map</u>	 是	 是	 是
<u>AWS::Location::PlaceIndex</u>	 是	 是	 是
<u>AWS::Location::RouteCalculator</u>	 是	 是	 是
<u>AWS::Location::Tracker</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Location::TrackerConsumer</u>	 是	 是	 是
<u>AWS::Logs::AccountPolicy</u>	 是	 是	 是
<u>AWS::Logs::Delivery</u>	 是	 是	 否
<u>AWS::Logs::DeliveryDestination</u>	 是	 是	 否
<u>AWS::Logs::DeliverySource</u>	 是	 是	 否
<u>AWS::Logs::Destination</u>	 是	 是	 是
<u>AWS::Logs::Integration</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Logs::LogAnomalyDetector</u>	 是	 是	 否
<u>AWS::Logs::LogGroup</u>	 是	 是	 是
<u>AWS::Logs::LogStream</u>	 是	 是	 是
<u>AWS::Logs::MetricFilter</u>	 是	 是	 是
<u>AWS::Logs::QueryDefinition</u>	 是	 是	 否
<u>AWS::Logs::ResourcePolicy</u>	 是	 是	 是
<u>AWS::Logs::ScheduledQuery</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Logs::SubscriptionFilter</u>	 是	 是	 是
<u>AWS::Logs::Transformer</u>	 是	 是	 否
<u>AWS::LookoutEquipment::InferenceScheduler</u>	 是	 是	 否
<u>AWS::LookoutVision::Project</u>	 是	 是	 是
<u>AWS::M2::Application</u>	 是	 是	 否
<u>AWS::M2::Deployment</u>	 是	 是	 否
<u>AWS::M2::Environment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MPA::ApprovalTeam</u>	 是	 是	 否
<u>AWS::MPA::IdentitySource</u>	 是	 是	 否
<u>AWS::MSK::BatchScramSecret</u>	 是	 是	 是
<u>AWS::MSK::Cluster</u>	 是	 是	 是
<u>AWS::MSK::ClusterPolicy</u>	 是	 是	 是
<u>AWS::MSK::Configuration</u>	 是	 是	 是
<u>AWS::MSK::Replicator</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MSK::ServerlessCluster</u>	 是	 是	 是
<u>AWS::MSK::Topic</u>	 是	 是	 否
<u>AWS::MSK::VpcConnection</u>	 是	 是	 是
<u>AWS::MWAA::Environment</u>	 是	 是	 否
<u>AWS::MWAAServerless::Workflow</u>	 是	 是	 否
<u>AWS::Macie::AllowList</u>	 是	 是	 否
<u>AWS::Macie::CustomDataIdentifier</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Macie::FindingsFilter</u>	 是	 是	 否
<u>AWS::Macie::Session</u>	 是	 是	 是
<u>AWS::ManagedBlockchain::Accessor</u>	 是	 是	 是
<u>AWS::MediaConnect::Bridge</u>	 是	 是	 是
<u>AWS::MediaConnect::BridgeOutput</u>	 是	 是	 否
<u>AWS::MediaConnect::BridgeSource</u>	 是	 是	 否
<u>AWS::MediaConnect::Flow</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MediaConnect::FlowEntitlement</u>	 是	 是	 否
<u>AWS::MediaConnect::FlowOutput</u>	 是	 是	 否
<u>AWS::MediaConnect::FlowSource</u>	 是	 是	 否
<u>AWS::MediaConnect::FlowVpcInterface</u>	 是	 是	 是
<u>AWS::MediaConnect::Gateway</u>	 是	 是	 是
<u>AWS::MediaConnect::RouterInput</u>	 是	 是	 否
<u>AWS::MediaConnect::RouterNetworkInterface</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::MediaConnect::RouterOutput	 是	 是	 否
AWS::MediaLive::ChannelPlacementGroup	 是	 是	 否
AWS::MediaLive::CloudWatchAlarmTemplate	 是	 是	 否
AWS::MediaLive::CloudWatchAlarmTemplateGroup	 是	 是	 否
AWS::MediaLive::Cluster	 是	 是	 否
AWS::MediaLive::EventBridgeRuleTemplate	 是	 是	 否
AWS::MediaLive::EventBridgeRuleTemplateGroup	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MediaLive::Multiplex</u>	 是	 是	 否
<u>AWS::MediaLive::Multiplexprogram</u>	 是	 是	 否
<u>AWS::MediaLive::Network</u>	 是	 是	 否
<u>AWS::MediaLive::SdiSource</u>	 是	 是	 否
<u>AWS::MediaLive::SignalMap</u>	 是	 是	 否
<u>AWS::MediaPackage::Asset</u>	 是	 是	 是
<u>AWS::MediaPackage::Channel</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MediaPackage::OriginEndpoint</u>	 是	 是	 是
<u>AWS::MediaPackage::PackagingConfiguration</u>	 是	 是	 是
<u>AWS::MediaPackage::PackagingGroup</u>	 是	 是	 是
<u>AWS::MediaPackageV2::Channel</u>	 是	 是	 是
<u>AWS::MediaPackageV2::ChannelGroup</u>	 是	 是	 是
<u>AWS::MediaPackageV2::ChannelPolicy</u>	 是	 是	 否
<u>AWS::MediaPackageV2::OriginEndpoint</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MediaPackageV2::OriginEndpointPolicy</u>	 是	 是	 否
<u>AWS::MediaTailor::Channel</u>	 是	 是	 否
<u>AWS::MediaTailor::ChannelPolicy</u>	 是	 是	 否
<u>AWS::MediaTailor::LiveSource</u>	 是	 是	 是
<u>AWS::MediaTailor::PlaybackConfiguration</u>	 是	 是	 否
<u>AWS::MediaTailor::SourceLocation</u>	 是	 是	 是
<u>AWS::MediaTailor::VodSource</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::MemoryDB::ACL</u>	 是	 是	 是
<u>AWS::MemoryDB::Cluster</u>	 是	 是	 是
<u>AWS::MemoryDB::MultiRegionCluster</u>	 是	 是	 否
<u>AWS::MemoryDB::ParameterGroup</u>	 是	 是	 是
<u>AWS::MemoryDB::SubnetGroup</u>	 是	 是	 是
<u>AWS::MemoryDB::User</u>	 是	 是	 是
<u>AWS::Neptune::DBCluster</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Neptune::DBClusterParameterGroup</u>	 是	 是	 否
<u>AWS::Neptune::DBInstance</u>	 是	 是	 否
<u>AWS::Neptune::DBParameterGroup</u>	 是	 是	 否
<u>AWS::Neptune::DBSubnetGroup</u>	 是	 是	 否
<u>AWS::Neptune::EventSubscription</u>	 是	 是	 否
<u>AWS::NeptuneGraph::Graph</u>	 是	 是	 否
<u>AWS::NeptuneGraph::PrivateGraphEndpoint</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::NetworkFirewall::Firewall</u>	 是	 是	 是
<u>AWS::NetworkFirewall::FirewallPolicy</u>	 是	 是	 是
<u>AWS::NetworkFirewall::LoggingConfiguration</u>	 是	 是	 否
<u>AWS::NetworkFirewall::RuleGroup</u>	 是	 是	 是
<u>AWS::NetworkFirewall::TLSInspectionConfiguration</u>	 是	 是	 否
<u>AWS::NetworkFirewall::VpcEndpointAssociation</u>	 是	 是	 否
<u>AWS::NetworkManager::ConnectAttachment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::NetworkManager::ConnectPeer</u>	 是	 是	 是
<u>AWS::NetworkManager::CoreNetwork</u>	 是	 是	 是
<u>AWS::NetworkManager::CoreNetworkPrefixListAssociation</u>	 是	 是	 否
<u>AWS::NetworkManager::CustomerGatewayAssociation</u>	 是	 是	 是
<u>AWS::NetworkManager::Device</u>	 是	 是	 是
<u>AWS::NetworkManager::DirectConnectGatewayAttachment</u>	 是	 是	 否
<u>AWS::NetworkManager::GlobalNetwork</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::NetworkManager::Link</u>	 是	 是	 是
<u>AWS::NetworkManager::LinkAssociation</u>	 是	 是	 是
<u>AWS::NetworkManager::Site</u>	 是	 是	 是
<u>AWS::NetworkManager::SiteToSiteVpnAttachment</u>	 是	 是	 是
<u>AWS::NetworkManager::TransitGatewayPeering</u>	 是	 是	 是
<u>AWS::NetworkManager::TransitGatewayRegistration</u>	 是	 是	 是
<u>AWS::NetworkManager::TransitGatewayRouteTableAttachment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::NetworkManager::VpcAttachment</u>	 是	 是	 是
<u>AWS::Notifications::ChannelAssociation</u>	 是	 是	 否
<u>AWS::Notifications::EventRule</u>	 是	 是	 否
<u>AWS::Notifications::ManagedNotificationAccountContactAssociation</u>	 是	 是	 否
<u>AWS::Notifications::ManagedNotificationAdditionalChannelAssociation</u>	 是	 是	 否
<u>AWS::Notifications::NotificationConfiguration</u>	 是	 是	 否
<u>AWS::Notifications::NotificationHub</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Notifications::OrganizationalUnitAssociation</u>	 是	 是	 否
<u>AWS::NotificationsContacts::EmailContact</u>	 是	 是	 否
<u>AWS::ODB::CloudAutonomousVmCluster</u>	 是	 是	 否
<u>AWS::ODB::CloudExadataInfrastructure</u>	 是	 是	 否
<u>AWS::ODB::CloudVmCluster</u>	 是	 是	 否
<u>AWS::ODB::OdbNetwork</u>	 是	 是	 否
<u>AWS::ODB::OdbPeeringConnection</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::OSIS::Pipeline</u>	 是	 是	 是
<u>AWS::Oam::Link</u>	 是	 是	 是
<u>AWS::Oam::Sink</u>	 是	 是	 是
<u>AWS::ObservabilityAdmin::OrganizationCentralizationRule</u>	 是	 是	 否
<u>AWS::ObservabilityAdmin::OrganizationTelemetryRule</u>	 是	 是	 否
<u>AWS::ObservabilityAdmin::S3TableIntegration</u>	 是	 是	 否
<u>AWS::ObservabilityAdmin::TelemetryEnrichment</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ObservabilityAdmin::TelemetryPipelines</u>	 是	 是	 否
<u>AWS::ObservabilityAdmin::TelemetryRule</u>	 是	 是	 否
<u>AWS::Omics::AnnotationStore</u>	 是	 是	 是
<u>AWS::Omics::ReferenceStore</u>	 是	 是	 是
<u>AWS::Omics::RunGroup</u>	 是	 是	 是
<u>AWS::Omics::SequenceStore</u>	 是	 是	 是
<u>AWS::Omics::VariantStore</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Omics::Workflow</u>	 是	 是	 是
<u>AWS::Omics::WorkflowVersion</u>	 是	 是	 否
<u>AWS::OpenSearchServerless::AccessPolicy</u>	 是	 是	 是
<u>AWS::OpenSearchServerless::Collection</u>	 是	 是	 否
<u>AWS::OpenSearchServerless::CollectionGroup</u>	 是	 是	 否
<u>AWS::OpenSearchServerless::Index</u>	 是	 是	 否
<u>AWS::OpenSearchServerless::LifecyclePolicy</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::OpenSearchServerless::SecurityConfig</u>	 是	 是	 是
<u>AWS::OpenSearchServerless::SecurityPolicy</u>	 是	 是	 是
<u>AWS::OpenSearchServerless::VpcEndpoint</u>	 是	 是	 是
<u>AWS::OpenSearchService::Application</u>	 是	 是	 否
<u>AWS::OpenSearchService::Domain</u>	 是	 是	 否
<u>AWS::Organizations::Account</u>	 是	 是	 是
<u>AWS::Organizations::Organization</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Organizations::OrganizationalUnit</u>	 是	 是	 是
<u>AWS::Organizations::Policy</u>	 是	 是	 是
<u>AWS::Organizations::ResourcePolicy</u>	 是	 是	 是
<u>AWS::PCAConnectorAD::Connector</u>	 是	 是	 是
<u>AWS::PCAConnectorAD::DirectoryRegistration</u>	 是	 是	 是
<u>AWS::PCAConnectorAD::ServicePrincipalName</u>	 是	 是	 是
<u>AWS::PCAConnectorAD::Template</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::PCAConnectorAD::TemplateGroupAccessControlEntry</u>	 是	 是	 是
<u>AWS::PCAConnectorSCEP::Challenge</u>	 是	 是	 否
<u>AWS::PCAConnectorSCEP::Connector</u>	 是	 是	 否
<u>AWS::PCS::Cluster</u>	 是	 是	 否
<u>AWS::PCS::ComputeNodeGroup</u>	 是	 是	 否
<u>AWS::PCS::Queue</u>	 是	 是	 否
<u>AWS::Panorama::ApplicationInstance</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::Panorama::Package	 是	 是	 是
AWS::Panorama::PackageVersion	 是	 是	 否
AWS::PaymentCryptography::Alias	 是	 是	 否
AWS::PaymentCryptography::Key	 是	 是	 否
AWS::Personalize::Dataset	 是	 是	 是
AWS::Personalize::DatasetGroup	 是	 是	 是
AWS::Personalize::Schema	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Personalize::Solution</u>	 是	 是	 否
<u>AWS::Pinpoint::InAppTemplate</u>	 是	 是	 是
<u>AWS::Pipes::Pipe</u>	 是	 是	 是
<u>AWS::Proton::EnvironmentAccountConnection</u>	 是	 是	 是
<u>AWS::Proton::EnvironmentTemplate</u>	 是	 是	 是
<u>AWS::Proton::ServiceTemplate</u>	 是	 是	 是
<u>AWS::QBusiness::Application</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::QBusiness::DataAccessor</u>	 是	 是	 否
<u>AWS::QBusiness::DataSource</u>	 是	 是	 否
<u>AWS::QBusiness::Index</u>	 是	 是	 否
<u>AWS::QBusiness::Permission</u>	 是	 是	 否
<u>AWS::QBusiness::Plugin</u>	 是	 是	 否
<u>AWS::QBusiness::Retriever</u>	 是	 是	 否
<u>AWS::QBusiness::WebExperience</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::QLDB::Stream</u>	 是	 是	 是
<u>AWS::QuickSight::ActionConnector</u>	 是	 是	 否
<u>AWS::QuickSight::Analysis</u>	 是	 是	 否
<u>AWS::QuickSight::CustomPermissions</u>	 是	 是	 否
<u>AWS::QuickSight::Dashboard</u>	 是	 是	 否
<u>AWS::QuickSight::DataSet</u>	 是	 是	 是
<u>AWS::QuickSight::DataSource</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::QuickSight::Folder</u>	 是	 是	 否
<u>AWS::QuickSight::RefreshSchedule</u>	 是	 是	 是
<u>AWS::QuickSight::Template</u>	 是	 是	 否
<u>AWS::QuickSight::Theme</u>	 是	 是	 否
<u>AWS::QuickSight::Topic</u>	 是	 是	 是
<u>AWS::QuickSight::VPCConnection</u>	 是	 是	 是
<u>AWS::RAM::Permission</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::RAM::ResourceShare</u>	 是	 是	 否
<u>AWS::RDS::CustomDBEngineVersion</u>	 是	 是	 否
<u>AWS::RDS::DBCluster</u>	 是	 是	 是
<u>AWS::RDS::DBClusterParameterGroup</u>	 是	 是	 否
<u>AWS::RDS::DBInstance</u>	 是	 是	 是
<u>AWS::RDS::DBParameterGroup</u>	 是	 是	 否
<u>AWS::RDS::DBProxy</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::RDS::DBProxyEndpoint</u>	 是	 是	 是
<u>AWS::RDS::DBProxyTargetGroup</u>	 是	 是	 是
<u>AWS::RDS::DBShardGroup</u>	 是	 是	 否
<u>AWS::RDS::DBSubnetGroup</u>	 是	 是	 是
<u>AWS::RDS::EventSubscription</u>	 是	 是	 是
<u>AWS::RDS::GlobalCluster</u>	 是	 是	 是
<u>AWS::RDS::Integration</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::RDS::OptionGroup</u>	 是	 是	 是
<u>AWS::RTBFabric::InboundExternalLink</u>	 是	 是	 否
<u>AWS::RTBFabric::Link</u>	 是	 是	 否
<u>AWS::RTBFabric::OutboundExternalLink</u>	 是	 是	 否
<u>AWS::RTBFabric::RequesterGateway</u>	 是	 是	 否
<u>AWS::RTBFabric::ResponderGateway</u>	 是	 是	 否
<u>AWS::RUM::AppMonitor</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Rbin::Rule</u>	 是	 是	 否
<u>AWS::Redshift::Cluster</u>	 是	 是	 是
<u>AWS::Redshift::ClusterParameterGroup</u>	 是	 是	 否
<u>AWS::Redshift::ClusterSubnetGroup</u>	 是	 是	 否
<u>AWS::Redshift::EndpointAccess</u>	 是	 是	 是
<u>AWS::Redshift::EndpointAuthorization</u>	 是	 是	 是
<u>AWS::Redshift::EventSubscription</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Redshift::Integration</u>	 是	 是	 否
<u>AWS::Redshift::ScheduledAction</u>	 是	 是	 是
<u>AWS::RedshiftServerless::Namespace</u>	 是	 是	 否
<u>AWS::RedshiftServerless::Snapshot</u>	 是	 是	 否
<u>AWS::RedshiftServerless::Workgroup</u>	 是	 是	 否
<u>AWS::RefactorSpaces::Application</u>	 是	 是	 是
<u>AWS::RefactorSpaces::Environment</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::RefactorSpaces::Route</u>	 是	 是	 是
<u>AWS::RefactorSpaces::Service</u>	 是	 是	 是
<u>AWS::Rekognition::Collection</u>	 是	 是	 是
<u>AWS::Rekognition::Project</u>	 是	 是	 是
<u>AWS::Rekognition::StreamProcessor</u>	 是	 是	 是
<u>AWS::ResilienceHub::App</u>	 是	 是	 是
<u>AWS::ResilienceHub::ResiliencyPolicy</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ResourceExplorer2::DefaultViewAssociation</u>	 是	 是	 否
<u>AWS::ResourceExplorer2::Index</u>	 是	 是	 是
<u>AWS::ResourceExplorer2::View</u>	 是	 是	 是
<u>AWS::ResourceGroups::Group</u>	 是	 是	 是
<u>AWS::ResourceGroups::TagSyncTask</u>	 是	 是	 否
<u>AWS::RolesAnywhere::CRL</u>	 是	 是	 是
<u>AWS::RolesAnywhere::Profile</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::RolesAnywhere::TrustAnchor</u>	 是	 是	 否
<u>AWS::Route53::CidrCollection</u>	 是	 是	 是
<u>AWS::Route53::DNSSEC</u>	 是	 是	 是
<u>AWS::Route53::HealthCheck</u>	 是	 是	 是
<u>AWS::Route53::HostedZone</u>	 是	 是	 是
<u>AWS::Route53::KeySigningKey</u>	 是	 是	 是
<u>AWS::Route53GlobalResolver::AccessSource</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Route53GlobalResolver::AccessTo</u> <u>ken</u>	 是	 是	 否
<u>AWS::Route53GlobalResolver::DnsView</u>	 是	 是	 否
<u>AWS::Route53GlobalResolver::Firewall</u> <u>DomainList</u>	 是	 是	 否
<u>AWS::Route53GlobalResolver::Firewall</u> <u>Rule</u>	 是	 是	 否
<u>AWS::Route53GlobalResolver::GlobalRe</u> <u>solver</u>	 是	 是	 否
<u>AWS::Route53GlobalResolver::HostedZo</u> <u>neAssociation</u>	 是	 是	 否
<u>AWS::Route53Profiles::Profile</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Route53Profiles::ProfileAssociation</u>	 是	 是	 否
<u>AWS::Route53Profiles::ProfileResourceAssociation</u>	 是	 是	 否
<u>AWS::Route53RecoveryControl::Cluster</u>	 是	 是	 是
<u>AWS::Route53RecoveryControl::ControlPanel</u>	 是	 是	 是
<u>AWS::Route53RecoveryControl::RoutingControl</u>	 是	 是	 是
<u>AWS::Route53RecoveryControl::SafetyRule</u>	 是	 是	 是
<u>AWS::Route53RecoveryReadiness::Cell</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::Route53RecoveryReadiness::ReadinessCheck	 是	 是	 是
AWS::Route53RecoveryReadiness::RecoveryGroup	 是	 是	 是
AWS::Route53RecoveryReadiness::ResourceSet	 是	 是	 是
AWS::Route53Resolver::FirewallDomainList	 是	 是	 否
AWS::Route53Resolver::FirewallRuleGroup	 是	 是	 是
AWS::Route53Resolver::FirewallRuleGroupAssociation	 是	 是	 是
AWS::Route53Resolver::OutpostResolver	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Route53Resolver::ResolverConfig</u>	 是	 是	 是
<u>AWS::Route53Resolver::ResolverDNSSECConfig</u>	 是	 是	 是
<u>AWS::Route53Resolver::ResolverEndpoint</u>	 是	 是	 否
<u>AWS::Route53Resolver::ResolverQueryLoggingConfig</u>	 是	 是	 是
<u>AWS::Route53Resolver::ResolverQueryLoggingConfigAssociation</u>	 是	 是	 是
<u>AWS::Route53Resolver::ResolverRule</u>	 是	 是	 否
<u>AWS::Route53Resolver::ResolverRuleAssociation</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::S3::AccessGrant</u>	 是	 是	 否
<u>AWS::S3::AccessGrantsInstance</u>	 是	 是	 否
<u>AWS::S3::AccessGrantsLocation</u>	 是	 是	 否
<u>AWS::S3::AccessPoint</u>	 是	 是	 是
<u>AWS::S3::Bucket</u>	 是	 是	 是
<u>AWS::S3::BucketPolicy</u>	 是	 否	 是
<u>AWS::S3::MultiRegionAccessPoint</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::S3::MultiRegionAccessPointPolicy</u>	 是	 否	 否
<u>AWS::S3::StorageLens</u>	 是	 是	 是
<u>AWS::S3::StorageLensGroup</u>	 是	 是	 否
<u>AWS::S3Express::AccessPoint</u>	 是	 是	 否
<u>AWS::S3Express::BucketPolicy</u>	 是	 是	 否
<u>AWS::S3Express::DirectoryBucket</u>	 是	 是	 否
<u>AWS::S3ObjectLambda::AccessPoint</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::S3ObjectLambda::AccessPointPolicy	 是	 是	 否
AWS::S3Outposts::AccessPoint	 是	 是	 是
AWS::S3Outposts::Bucket	 是	 是	 是
AWS::S3Outposts::BucketPolicy	 是	 是	 否
AWS::S3Outposts::Endpoint	 是	 是	 是
AWS::S3Tables::Namespace	 是	 是	 否
AWS::S3Tables::Table	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::S3Tables::TableBucket</u>	 是	 是	 否
<u>AWS::S3Tables::TableBucketPolicy</u>	 是	 是	 否
<u>AWS::S3Tables::TablePolicy</u>	 是	 是	 否
<u>AWS::S3Vectors::Index</u>	 是	 是	 否
<u>AWS::S3Vectors::VectorBucket</u>	 是	 是	 否
<u>AWS::S3Vectors::VectorBucketPolicy</u>	 是	 是	 否
<u>AWS::SES::ConfigurationSet</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SES::ConfigurationSetEventDestination</u>	 否	 是	 否
<u>AWS::SES::ContactList</u>	 是	 是	 否
<u>AWS::SES::CustomVerificationEmailTemplate</u>	 是	 是	 否
<u>AWS::SES::DedicatedIpPool</u>	 是	 是	 是
<u>AWS::SES::EmailIdentity</u>	 是	 是	 是
<u>AWS::SES::MailManagerAddonInstance</u>	 是	 是	 否
<u>AWS::SES::MailManagerAddonSubscription</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SES::MailManagerAddressList</u>	 是	 是	 否
<u>AWS::SES::MailManagerArchive</u>	 是	 是	 否
<u>AWS::SES::MailManagerIngressPoint</u>	 是	 是	 否
<u>AWS::SES::MailManagerRelay</u>	 是	 是	 否
<u>AWS::SES::MailManagerRuleSet</u>	 是	 是	 否
<u>AWS::SES::MailManagerTrafficPolicy</u>	 是	 是	 否
<u>AWS::SES::MultiRegionEndpoint</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::SES::Template	 是	 是	 否
AWS::SES::Tenant	 是	 是	 否
AWS::SES::VdmAttributes	 是	 是	 否
AWS::SMSVOICE::ConfigurationSet	 是	 是	 否
AWS::SMSVOICE::OptOutList	 是	 是	 否
AWS::SMSVOICE::PhoneNumber	 是	 是	 否
AWS::SMSVOICE::Pool	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SMSVOICE::ProtectConfiguration</u>	 是	 是	 否
<u>AWS::SMSVOICE::ResourcePolicy</u>	 是	 是	 否
<u>AWS::SMSVOICE::SenderId</u>	 是	 是	 否
<u>AWS::SNS::Subscription</u>	 是	 否	 否
<u>AWS::SNS::Topic</u>	 是	 是	 是
<u>AWS::SNS::TopicInlinePolicy</u>	 是	 是	 否
<u>AWS::SQS::Queue</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::SQS::QueueInlinePolicy	 是	 是	 否
AWS::SSM::Association	 是	 是	 是
AWS::SSM::Document	 是	 否	 否
AWS::SSM::MaintenanceWindow	 是	 是	 否
AWS::SSM::MaintenanceWindowTarget	 是	 是	 否
AWS::SSM::MaintenanceWindowTask	 是	 是	 否
AWS::SSM::Parameter	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SSM::PatchBaseline</u>	 是	 是	 否
<u>AWS::SSM::ResourceDataSync</u>	 是	 是	 否
<u>AWS::SSM::ResourcePolicy</u>	 是	 是	 是
<u>AWS::SSMContacts::Contact</u>	 是	 是	 是
<u>AWS::SSMContacts::ContactChannel</u>	 是	 是	 是
<u>AWS::SSMContacts::Plan</u>	 是	 是	 否
<u>AWS::SSMContacts::Rotation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::SSMGuiConnect::Preferences</u>	 是	 是	 否
<u>AWS::SSMIncidents::ReplicationSet</u>	 是	 是	 是
<u>AWS::SSMIncidents::ResponsePlan</u>	 是	 是	 是
<u>AWS::SSMQuickSetup::ConfigurationManager</u>	 是	 是	 否
<u>AWS::SSMQuickSetup::LifecycleAutomation</u>	 是	 是	 否
<u>AWS::SSO::Application</u>	 是	 是	 否
<u>AWS::SSO::ApplicationAssignment</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SSO::Assignment</u>	 是	 是	 否
<u>AWS::SSO::Instance</u>	 是	 是	 否
<u>AWS::SSO::InstanceAccessControlAttributeConfiguration</u>	 是	 是	 是
<u>AWS::SageMaker::App</u>	 是	 是	 否
<u>AWS::SageMaker::AppImageConfig</u>	 是	 是	 否
<u>AWS::SageMaker::Cluster</u>	 是	 是	 否
<u>AWS::SageMaker::DataQualityJobDefinition</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SageMaker::Device</u>	 是	 是	 否
<u>AWS::SageMaker::DeviceFleet</u>	 是	 是	 否
<u>AWS::SageMaker::Domain</u>	 是	 是	 否
<u>AWS::SageMaker::Endpoint</u>	 否	 是	 否
<u>AWS::SageMaker::FeatureGroup</u>	 是	 是	 是
<u>AWS::SageMaker::Image</u>	 是	 是	 是
<u>AWS::SageMaker::ImageVersion</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::SageMaker::InferenceComponent</u>	 是	 是	 否
<u>AWS::SageMaker::InferenceExperiment</u>	 是	 是	 是
<u>AWS::SageMaker::MlflowTrackingServer</u>	 是	 是	 否
<u>AWS::SageMaker::ModelBiasJobDefinition</u>	 是	 是	 否
<u>AWS::SageMaker::ModelCard</u>	 是	 是	 是
<u>AWS::SageMaker::ModelExplainabilityJobDefinition</u>	 是	 是	 否
<u>AWS::SageMaker::ModelPackage</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SageMaker::ModelPackageGroup</u>	 是	 是	 是
<u>AWS::SageMaker::ModelQualityJobDefinition</u>	 是	 是	 否
<u>AWS::SageMaker::MonitoringSchedule</u>	 是	 否	 否
<u>AWS::SageMaker::PartnerApp</u>	 是	 是	 否
<u>AWS::SageMaker::Pipeline</u>	 是	 是	 是
<u>AWS::SageMaker::ProcessingJob</u>	 是	 是	 否
<u>AWS::SageMaker::Project</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SageMaker::Space</u>	 是	 是	 否
<u>AWS::SageMaker::StudioLifecycleConfiguration</u>	 是	 是	 否
<u>AWS::SageMaker::UserProfile</u>	 是	 是	 否
<u>AWS::Scheduler::Schedule</u>	 是	 是	 是
<u>AWS::Scheduler::ScheduleGroup</u>	 是	 是	 否
<u>AWS::SecretsManager::ResourcePolicy</u>	 是	 否	 否
<u>AWS::SecretsManager::RotationSchedule</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SecretsManager::Secret</u>	 是	 是	 是
<u>AWS::SecretsManager::SecretTargetAttachment</u>	 是	 否	 否
<u>AWS::SecurityHub::AggregatorV2</u>	 是	 是	 否
<u>AWS::SecurityHub::AutomationRule</u>	 是	 是	 否
<u>AWS::SecurityHub::AutomationRuleV2</u>	 是	 是	 否
<u>AWS::SecurityHub::ConfigurationPolicy</u>	 是	 是	 否
<u>AWS::SecurityHub::ConnectorV2</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::SecurityHub::DelegatedAdmin</u>	 是	 是	 否
<u>AWS::SecurityHub::FindingAggregator</u>	 是	 是	 否
<u>AWS::SecurityHub::Hub</u>	 是	 是	 否
<u>AWS::SecurityHub::HubV2</u>	 是	 是	 否
<u>AWS::SecurityHub::Insight</u>	 是	 是	 否
<u>AWS::SecurityHub::OrganizationConfiguration</u>	 是	 是	 否
<u>AWS::SecurityHub::PolicyAssociation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SecurityHub::ProductSubscription</u>	 是	 是	 否
<u>AWS::SecurityHub::SecurityControl</u>	 是	 是	 否
<u>AWS::SecurityHub::Standard</u>	 是	 是	 否
<u>AWS::SecurityLake::AwsLogSource</u>	 是	 是	 否
<u>AWS::SecurityLake::DataLake</u>	 是	 是	 否
<u>AWS::SecurityLake::Subscriber</u>	 是	 是	 否
<u>AWS::SecurityLake::SubscriberNotification</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ServiceCatalog::CloudFormationProvisionedProduct</u>	 是	 否	 否
<u>AWS::ServiceCatalog::LaunchNotificationConstraint</u>	 是	 是	 否
<u>AWS::ServiceCatalog::LaunchRoleConstraint</u>	 是	 是	 否
<u>AWS::ServiceCatalog::LaunchTemplateConstraint</u>	 是	 是	 否
<u>AWS::ServiceCatalog::Portfolio</u>	 是	 是	 否
<u>AWS::ServiceCatalog::PortfolioPrincipalAssociation</u>	 是	 是	 否
<u>AWS::ServiceCatalog::PortfolioProductAssociation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ServiceCatalog::PortfolioShare</u>	 是	 否	 否
<u>AWS::ServiceCatalog::ResourceUpdateConstraint</u>	 是	 是	 否
<u>AWS::ServiceCatalog::ServiceAction</u>	 是	 是	 是
<u>AWS::ServiceCatalog::ServiceActionAssociation</u>	 是	 是	 否
<u>AWS::ServiceCatalog::StackSetConstraint</u>	 是	 是	 否
<u>AWS::ServiceCatalog::TagOption</u>	 是	 是	 否
<u>AWS::ServiceCatalog::TagOptionAssociation</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::ServiceCatalogAppRegistry::Application</u>	 是	 是	 是
<u>AWS::ServiceCatalogAppRegistry::AttributeGroup</u>	 是	 是	 是
<u>AWS::ServiceCatalogAppRegistry::AttributeGroupAssociation</u>	 是	 否	 否
<u>AWS::ServiceCatalogAppRegistry::ResourceAssociation</u>	 是	 否	 否
<u>AWS::ServiceDiscovery::Service</u>	 是	 是	 否
<u>AWS::Shield::DRTAccess</u>	 是	 是	 否
<u>AWS::Shield::ProactiveEngagement</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	laC 產生器
<u>AWS::Shield::Protection</u>	 是	 是	 否
<u>AWS::Shield::ProtectionGroup</u>	 是	 是	 否
<u>AWS::Signer::ProfilePermission</u>	 是	 是	 否
<u>AWS::Signer::SigningProfile</u>	 是	 是	 是
<u>AWS::SimSpaceWeaver::Simulation</u>	 是	 是	 是
<u>AWS::StepFunctions::Activity</u>	 是	 是	 否
<u>AWS::StepFunctions::StateMachine</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::StepFunctions::StateMachineAlias	 是	 是	 是
AWS::StepFunctions::StateMachineVersion	 是	 是	 是
AWS::SupportApp::AccountAlias	 是	 是	 是
AWS::SupportApp::SlackChannelConfiguration	 是	 是	 是
AWS::SupportApp::SlackWorkspaceConfiguration	 是	 是	 是
AWS::Synthetics::Canary	 是	 是	 是
AWS::Synthetics::Group	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::SystemsManagerSAP::Application</u>	 是	 是	 否
<u>AWS::Timestream::Database</u>	 是	 是	 是
<u>AWS::Timestream::InfluxDBCluster</u>	 是	 是	 否
<u>AWS::Timestream::InfluxDBInstance</u>	 是	 是	 否
<u>AWS::Timestream::ScheduledQuery</u>	 是	 是	 否
<u>AWS::Timestream::Table</u>	 是	 是	 是
<u>AWS::Transfer::Agreement</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Transfer::Certificate</u>	 是	 是	 是
<u>AWS::Transfer::Connector</u>	 是	 是	 是
<u>AWS::Transfer::Profile</u>	 是	 是	 是
<u>AWS::Transfer::Server</u>	 是	 是	 否
<u>AWS::Transfer::User</u>	 是	 是	 否
<u>AWS::Transfer::WebApp</u>	 是	 是	 否
<u>AWS::Transfer::Workflow</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
AWS::VerifiedPermissions::IdentitySource	 是	 是	 是
AWS::VerifiedPermissions::Policy	 是	 是	 是
AWS::VerifiedPermissions::PolicyStore	 是	 是	 是
AWS::VerifiedPermissions::PolicyTemplate	 是	 是	 是
AWS::VoiceID::Domain	 是	 是	 否
AWS::VpcLattice::AccessLogSubscription	 是	 是	 是
AWS::VpcLattice::AuthPolicy	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::VpcLattice::DomainVerification</u>	 是	 是	 否
<u>AWS::VpcLattice::Listener</u>	 是	 是	 是
<u>AWS::VpcLattice::ResourceConfiguration</u>	 是	 是	 否
<u>AWS::VpcLattice::ResourceGateway</u>	 是	 是	 否
<u>AWS::VpcLattice::ResourcePolicy</u>	 是	 是	 否
<u>AWS::VpcLattice::Rule</u>	 是	 是	 是
<u>AWS::VpcLattice::Service</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::VpcLattice::ServiceNetwork</u>	 是	 是	 是
<u>AWS::VpcLattice::ServiceNetworkResourceAssociation</u>	 是	 是	 否
<u>AWS::VpcLattice::ServiceNetworkServiceAssociation</u>	 是	 是	 是
<u>AWS::VpcLattice::ServiceNetworkVpcAssociation</u>	 是	 是	 是
<u>AWS::VpcLattice::TargetGroup</u>	 是	 是	 是
<u>AWS::WAFv2::IPSet</u>	 是	 是	 是
<u>AWS::WAFv2::LoggingConfiguration</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::WAFv2::RegexPatternSet</u>	 是	 是	 是
<u>AWS::WAFv2::RuleGroup</u>	 是	 是	 是
<u>AWS::WAFv2::WebACL</u>	 是	 是	 是
<u>AWS::WAFv2::WebACLAssociation</u>	 是	 是	 否
<u>AWS::Wisdom::AIAgent</u>	 是	 是	 否
<u>AWS::Wisdom::AIAgentVersion</u>	 是	 是	 否
<u>AWS::Wisdom::AIGuardrail</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Wisdom::AIGuardrailVersion</u>	 是	 是	 否
<u>AWS::Wisdom::AIPrompt</u>	 是	 是	 否
<u>AWS::Wisdom::AIPromptVersion</u>	 是	 是	 否
<u>AWS::Wisdom::Assistant</u>	 是	 是	 是
<u>AWS::Wisdom::AssistantAssociation</u>	 是	 是	 是
<u>AWS::Wisdom::KnowledgeBase</u>	 是	 是	 是
<u>AWS::Wisdom::MessageTemplate</u>	 是	 是	 否

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::Wisdom::MessageTemplateVersion</u>	 是	 是	 否
<u>AWS::Wisdom::QuickResponse</u>	 是	 是	 否
<u>AWS::WorkSpaces::ConnectionAlias</u>	 是	 是	 否
<u>AWS::WorkSpaces::Workspace</u>	 是	 是	 否
<u>AWS::WorkSpaces::WorkspacesPool</u>	 是	 是	 否
<u>AWS::WorkSpacesThinClient::Environment</u>	 是	 是	 否
<u>AWS::WorkSpacesWeb::BrowserSettings</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::WorkSpacesWeb::DataProtectionSettings</u>	 是	 是	 否
<u>AWS::WorkSpacesWeb::IdentityProvider</u>	 是	 是	 是
<u>AWS::WorkSpacesWeb::IpAccessSettings</u>	 是	 是	 是
<u>AWS::WorkSpacesWeb::NetworkSettings</u>	 是	 是	 是
<u>AWS::WorkSpacesWeb::Portal</u>	 是	 是	 是
<u>AWS::WorkSpacesWeb::SessionLogger</u>	 是	 是	 否
<u>AWS::WorkSpacesWeb::TrustStore</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::WorkSpacesWeb::UserAccessLoggingSettings</u>	 是	 是	 是
<u>AWS::WorkSpacesWeb::UserSettings</u>	 是	 是	 是
<u>AWS::WorkspacesInstances::Volume</u>	 是	 是	 否
<u>AWS::WorkspacesInstances::VolumeAssociation</u>	 是	 是	 否
<u>AWS::WorkspacesInstances::WorkspaceInstance</u>	 是	 是	 否
<u>AWS::XRay::Group</u>	 是	 是	 是
<u>AWS::XRay::ResourcePolicy</u>	 是	 是	 是

資源	Import (匯入)	漂移偵測	IaC 產生器
<u>AWS::XRay::SamplingRule</u>	 是	 是	 否
<u>AWS::XRay::TransactionSearchConfig</u>	 是	 是	 否

使用快速建立連結來建立 CloudFormation 堆疊

快速建立連結提供簡化的方法，可直接從 CloudFormation 主控台的 URL 中啟動 CloudFormation 堆疊。透過將範本 URL、堆疊名稱和範本參數指定為 URL 查詢參數，可預先填入單個建立堆疊頁面並加快堆疊建立。可以透過減少精靈頁面數目和所需使用者輸入數量，來簡化建立堆疊的程序。它也會最佳化範本重複使用，因為您可以針對相同的範本指定不同的值來建立多個 URL。

URL 格式

快速建立連結遵循此 URL 格式：

```
https://region-code.console.aws.amazon.com/cloudformation/home?region=region-code#/stacks/create/review
?templateURL=TemplateURL
&stackName=StackName
&param_parameterName=parameterValue
```

CloudFormation 支援下列 URL 查詢參數：

範本 URL

必要. `templateURL` 參數可指定 Amazon S3 儲存貯體中堆疊範本的 URL。若要避免預先簽章的 S3 URL 的存取問題，請確保對 URL 進行 URL 編碼。

支援的 S3 URL 格式：

- `https://s3.region-code.amazonaws.com/bucket-name/template-name`
- `https://bucket-name.s3.region-code.amazonaws.com/template-name`
- `https://s3-region-code.amazonaws.com/bucket-name/template-name` (舊格式)

Stack name (堆疊名稱)

選用。使用 `stackName` 參數來指定要建立的 CloudFormation 堆疊的名稱。堆疊名稱只能包含英數字元 (區分大小寫) 和連字號。必須以字母字元開頭，且長度不可超過 128 個字元。

範本參數

選用。對於堆疊範本中非 `NoEcho` 參數類型的參數，請在 URL 查詢字串中使用 `param_parameterName` 格式。URL 參數必須包含 `param_` 字首，而且參數名稱區段必須與範本中的參數名稱完全相符。例如：`param_DBName`。

CloudFormation 會忽略不存在於範本中的參數，以及任何將 `NoEcho` 屬性設為 `true` 類型的已定義參數 (通常為使用者名稱和密碼)。URL 參數會覆寫範本中所指定的預設值。您可以包含所需數目的參數。

Important

我們建議您不要直接在 CloudFormation 範本中嵌入敏感資訊，而是在堆疊範本中使用動態參數來參考在 CloudFormation 外部存放和管理的敏感資訊，例如在 AWS Systems Manager 參數存放區或 AWS Secrets Manager 中。

如需詳細資訊，請參閱[請勿在範本中內嵌認證](#)的最佳實務。

所有查詢參數名稱都區分大小寫。使用者可以先在主控台中覆寫這些值，再建立堆疊。

範例

下列範例根據 [WordPress 基本單一執行個體](#) 範例範本。查詢字串包含所需的 `templateURL` 參數以及 `stackName`、`DBName`、`InstanceType` 和 `KeyName` 參數。

為了清楚起見，下列 URL 已新增分行符號。

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review
?templateURL=https://s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/
WordPress_Single_Instance.template
&stackName=MyWPBlog
```



```
&param_DBName=mywpblog  
&param_InstanceType=t2.medium
```

下列 URL 包含與前一個範例相同的參數，但會移除分行符號。這是實際 URL 格式。

```
https://us-east-2.console.aws.amazon.com/cloudformation/home?  
region=us-east-2#/stacks/create/review?templateURL=https://  
s3.us-east-2.amazonaws.com/cloudformation-templates-us-east-2/  
WordPress_Single_Instance.template&stackName=MyWPBlog&param_DBName=mywpblog&param_InstanceType=
```

使用快速建立連結來建立堆疊

當您開啟快速建立連結時，系統會將您導向至 CloudFormation 主控台。主控台會直接開啟快速建立堆疊頁面，其中提供的值會自動用於參數。

若要使用快速建立連結來建立堆疊 (主控台)

1. 在快速建立堆疊頁面中，針對範本、範本 URL，確認範本 URL 正確。
2. 展開檢視範本區段以驗證範本。
3. 針對堆疊名稱，驗證預先填入的堆疊名稱。
4. 檢閱參數區段。驗證預先填入的參數值是否正確。填寫 URL 中未指定的所有必要參數。視需要修改任何預先填入的值。
5. 接下來，可以進行下列設定：
 - 標籤 — 使用鍵值對組織資源。
 - 許可 — 選擇堆疊操作的 IAM 服務角色。
 - 堆疊失敗選項 — 選擇復原 (預設) 或保留資源。
 - 堆疊政策 — 控制資源更新許可。
 - 復原組態 — 設定 CloudWatch 警示型復原。
 - 通知選項 — 設定堆疊事件的 Amazon SNS 通知。
 - 堆疊建立選項 — 定義堆疊建立時間上限，並啟用終止保護以防止意外刪除。

如需詳細資訊，請參閱[設定堆疊選項](#)。

6. 對於功能，請完成任何必要的確認。例如，如果您的範本包含 IAM 資源，請選取我知道此範本可能會建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。

7. (選用) 您可以建立變更集，以在建立之前預覽堆疊的組態。選擇建立變更集並遵循指示。如需詳細資訊，請參閱[預覽堆疊的組態](#)。
8. 當您準備就緒時，請選擇建立堆疊以啟動堆疊，然後在事件索引標籤中監控堆疊建立進度。如需詳細資訊，請參閱[監控堆疊進度](#)。

AWS CLI 和 PowerShell 的 CloudFormation 堆疊操作命令範例

下列命令列範例示範如何使用 AWS CLI 和 PowerShell 執行個別 CloudFormation 動作。這些範例僅包含最常使用的動作。如需完整清單，請參閱 AWS CLI 命令參考中的 [cloudformation](#)。

本指南中的範例採用反斜線 (\) 的慣例，用於表示較長的命令列會延續至下一行。

主題

- [取消堆疊更新](#)
- [繼續復原更新](#)
- [建立堆疊](#)
- [建立包含轉換的堆疊](#)
- [刪除堆疊](#)
- [描述堆疊事件](#)
- [描述堆疊資源](#)
- [描述堆疊資源](#)
- [描述堆疊](#)
- [取得範本](#)
- [列出堆疊資源](#)
- [列出堆疊](#)
- [更新堆疊](#)
- [驗證您的範本](#)
- [使用 將本機成品上傳至 S3 儲存貯體 AWS CLI](#)

取消堆疊更新

使用 [cancel-update-stack](#) 命令以取消堆疊更新。如需詳細資訊，請參閱[取消堆疊更新](#)。

CLI

AWS CLI

取消進行中的堆疊更新

下列 `cancel-update-stack` 命令會取消 `myteststack` 堆疊上的堆疊更新：

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [CancelUpdateStack](#)。

PowerShell

Tools for PowerShell V4

範例 1：取消指定堆疊的更新。

```
Stop-CFNUpdateStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [CancelUpdateStack](#)。

Tools for PowerShell V5

範例 1：取消指定堆疊的更新。

```
Stop-CFNUpdateStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [CancelUpdateStack](#)。

繼續復原更新

使用 [continue-update-rollback](#) 命令繼續復原更新。如需詳細資訊，請參閱[繼續復原更新](#)。

CLI

AWS CLI

重試更新復原

下列 `continue-update-rollback` 範例從先前失敗的堆疊更新處繼續還原操作。

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ContinueUpdateRollback](#)。

PowerShell

Tools for PowerShell V4

範例 1：繼續還原具名堆疊，其應處於 'UPDATE_ROLLBACK_FAILED' 狀態。如果繼續還原成功，堆疊將進入狀態 'UPDATE_ROLLBACK_COMPLETE'。

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [ContinueUpdateRollback](#)。

Tools for PowerShell V5

範例 1：繼續還原具名堆疊，其應處於 'UPDATE_ROLLBACK_FAILED' 狀態。如果繼續還原成功，堆疊將進入狀態 'UPDATE_ROLLBACK_COMPLETE'。

```
Resume-CFNUpdateRollback -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [ContinueUpdateRollback](#)。

建立堆疊

使用 [create-stack](#) 命令來建立堆疊。您必須提供堆疊名稱、有效範本的位置和任何輸入參數。參數索引鍵名稱區分大小寫。若您輸入錯誤的參數金鑰名稱，CloudFormation 就不會建立堆疊，並會報告範本未包含該參數。

以下範例展示如何使用指定的名稱、範本和輸入參數建立新的堆疊。

CLI

AWS CLI

建立 an AWS CloudFormation 堆疊

以下 `create-stacks` 命令使用 `sampletemplate.json` 範本建立名為 `myteststack` 的堆疊：

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

輸出：

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的堆疊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [CreateStack](#)。

PowerShell

Tools for PowerShell V4

範例 1：使用指定的名稱建立新堆疊。範本是根據提供的內容和自訂參數 ('PK1' 和 'PK2' 代表範本內容中宣告的參數名稱，'PV1' 和 'PV2' 代表這些參數的值) 進行剖析。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果堆疊建立失敗，則不會還原。

```
New-CFNStack -StackName "myStack" `  
    -TemplateBody "{TEMPLATE CONTENT HERE}" `  
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },  
    @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `  
    -DisableRollback $true
```

範例 2：使用指定的名稱建立新堆疊。範本是根據提供的內容和自訂參數 ('PK1' 和 'PK2' 代表範本內容中宣告的參數名稱，'PV1' 和 'PV2' 代表這些參數的值) 進行剖析。自訂參數也可以使用

'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果堆疊建立失敗，則會還原。

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
    -TemplateBody "{TEMPLATE CONTENT HERE}" `
    -Parameter @( $p1, $p2 ) `
    -OnFailure "ROLLBACK"
```

範例 3：使用指定的名稱建立新堆疊。範本是從具有自訂參數 ('PK1' 代表範本內容中宣告的參數名稱，'PV1' 代表參數的值) 的 Amazon S3 URL 取得。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果建立堆疊失敗，則會還原 (與指定 -DisableRollback \$false 相同)。

```
New-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 4：使用指定的名稱建立新堆疊。範本是從具有自訂參數 ('PK1' 代表範本內容中宣告的參數名稱，'PV1' 代表參數的值) 的 Amazon S3 URL 取得。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果建立堆疊失敗，則會還原 (與指定 -DisableRollback \$false 相同)。指定的通知 AEN 會收到已發佈的堆疊相關事件。

```
New-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
    -NotificationARN @( "arn1", "arn2" )
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [CreateStack](#)。

Tools for PowerShell V5

範例 1：使用指定的名稱建立新堆疊。範本是根據提供的內容和自訂參數 ('PK1' 和 'PK2' 代表範本內容中宣告的參數名稱，'PV1' 和 'PV2' 代表這些參數的值) 進行剖析。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果堆疊建立失敗，則不會還原。

```
New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                             @{ ParameterKey="PK2"; ParameterValue="PV2" }) `
              -DisableRollback $true
```

範例 2：使用指定的名稱建立新堆疊。範本是根據提供的內容和自訂參數 ('PK1' 和 'PK2' 代表範本內容中宣告的參數名稱，'PV1' 和 'PV2' 代表這些參數的值) 進行剖析。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果堆疊建立失敗，則會還原。

```
$p1 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p1.ParameterKey = "PK1"
$p1.ParameterValue = "PV1"

$p2 = New-Object -Type Amazon.CloudFormation.Model.Parameter
$p2.ParameterKey = "PK2"
$p2.ParameterValue = "PV2"

New-CFNStack -StackName "myStack" `
              -TemplateBody "{TEMPLATE CONTENT HERE}" `
              -Parameter @( $p1, $p2 ) `
              -OnFailure "ROLLBACK"
```

範例 3：使用指定的名稱建立新堆疊。範本是從具有自訂參數 ('PK1' 代表範本內容中宣告的參數名稱，'PV1' 代表參數的值) 的 Amazon S3 URL 取得。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果建立堆疊失敗，則會還原 (與指定 -DisableRollback \$false 相同)。

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
              templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 4：使用指定的名稱建立新堆疊。範本是從具有自訂參數 ('PK1' 代表範本內容中宣告的參數名稱，'PV1' 代表參數的值) 的 Amazon S3 URL 取得。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。如果建立堆疊失敗，則會還原 (與指定 `-DisableRollback $false` 相同)。指定的通知 AEN 會收到已發佈的堆疊相關事件。

```
New-CFNStack -StackName "myStack" `
              -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
              -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" } `
              -NotificationARN @( "arn1", "arn2" )
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [CreateStack](#)。

Note

您可以使用 AWS CLI `--template-url` 選項，在 Amazon S3 或 中指定範本檔案位置 AWS Systems Manager。

對於 Amazon S3，URL 必須以 `https://` 開頭。不支援 S3 靜態網站 URL。

```
--template-url https://s3.region-code.amazonaws.com/bucket-name/template-name
```

對於 AWS Systems Manager，請使用下列格式：

```
--template-url "ssm-doc://arn:aws:ssm:region-code:account-id:document/document-name"
```

建立包含轉換的堆疊

使用 [deploy](#) 命令來建立包含轉換的堆疊。當您從包含轉換的範本建立堆疊時，必須使用變更集。deploy 命令會將兩個步驟 (建立變更集並執行) 合併為單一命令。

AWS CLI

下列 deploy 命令會建立具有指定名稱、範本和輸入參數的堆疊。

```
aws cloudformation deploy --stack-name myteststack \
```



```
--template /path_to_template/my-template.json \  
--parameter-overrides Key1=Value1 Key2=Value2
```

刪除堆疊

使用 [delete-stack](#) 命令來刪除堆疊。如需詳細資訊，請參閱[刪除堆疊](#)。

CLI

AWS CLI

刪除堆疊

以下 delete-stack 範例會刪除指定的堆疊。

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DeleteStack](#)。

PowerShell

Tools for PowerShell V4

範例 1：刪除指定的堆疊。

```
Remove-CFNStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [DeleteStack](#)。

Tools for PowerShell V5

範例 1：刪除指定的堆疊。

```
Remove-CFNStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [DeleteStack](#)。

若刪除失敗並傳回 `DELETE_FAILED` 狀態，您可選擇使用 `--deletion-mode` 選項強制刪除堆疊。下列是可與 `deletion-mode` 搭配使用的值：

- `STANDARD`：正常刪除堆疊。這是預設刪除模式。
- `FORCE_DELETE_STACK`：刪除堆疊並略過所有無法刪除的資源。

AWS CLI

下列 `delete-stack` 命令強制使用 `FORCE_DELETE_STACK` 值搭配 `deletion-mode` 參數刪除 *myteststack* 堆疊：

```
aws cloudformation delete-stack --stack-name myteststack \
  --deletion-mode FORCE_DELETE_STACK
```

此命令不會產生輸出。

使用 `FORCE_DELETE_STACK` 之後，您可使用 `list-stack-resources` 命令列出在堆疊刪除程序期間略過的資源。保留的資源會顯示 `DELETE_SKIPPED` 狀態。如需詳細資訊，請參閱[列出堆疊資源](#)。

描述堆疊事件

使用 [describe-stack-events](#) 命令來描述堆疊事件。如需詳細資訊，請參閱[監控堆疊進度](#)。

CLI

AWS CLI

描述堆疊事件

下列 `describe-stack-events` 範例顯示指定堆疊的 2 個最近事件。

```
aws cloudformation describe-stack-events \
  --stack-name my-stack \
  --max-items 2

{
  "StackEvents": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

        "EventId": "4e1516d0-e4d6-xmpl-b94f-0a51958a168c",
        "StackName": "my-stack",
        "LogicalResourceId": "my-stack",
        "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "ResourceType": "AWS::CloudFormation::Stack",
        "Timestamp": "2019-10-02T05:34:29.556Z",
        "ResourceStatus": "UPDATE_COMPLETE"
    },
    {
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "EventId": "4dd3c810-e4d6-xmpl-bade-0aaf8b31ab7a",
        "StackName": "my-stack",
        "LogicalResourceId": "my-stack",
        "PhysicalResourceId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "ResourceType": "AWS::CloudFormation::Stack",
        "Timestamp": "2019-10-02T05:34:29.127Z",
        "ResourceStatus": "UPDATE_COMPLETE_CLEANUP_IN_PROGRESS"
    }
],
"NextToken": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeStackEvents](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回特定堆疊的所有堆疊相關事件。

```
Get-CFNStackEvent -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [DescribeStackEvents](#)。

Tools for PowerShell V5

範例 1：傳回特定堆疊的所有堆疊相關事件。

```
Get-CFNStackEvent -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [DescribeStackEvents](#)。

描述堆疊資源

CLI

AWS CLI

取得堆疊資源的相關資訊

下列 `describe-stack-resource` 範例顯示指定的堆疊中名為 `MyFunction` 的資源詳細資訊。

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

輸出：

```
{  
  "StackResourceDetail": {  
    "StackName": "MyStack",  
    "StackId": "arn:aws:cloudformation:us-east-2:123456789012:stack/MyStack/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204",  
    "LogicalResourceId": "MyFunction",  
    "PhysicalResourceId": "my-function-SEZV4XMPL4S5",  
    "ResourceType": "AWS::Lambda::Function",  
    "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",  
    "ResourceStatus": "UPDATE_COMPLETE",  
    "Metadata": "{}",  
    "DriftInformation": {  
      "StackResourceDriftStatus": "IN_SYNC"  
    }  
  }  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeStackResource](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回透過邏輯 ID "MyDBInstance" 在與指定堆疊相關聯的範本中識別的資源描述。

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [DescribeStackResource](#)。

Tools for PowerShell V5

範例 1：傳回透過邏輯 ID "MyDBInstance" 在與指定堆疊相關聯的範本中識別的資源描述。

```
Get-CFNStackResource -StackName "myStack" -LogicalResourceId "MyDBInstance"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [DescribeStackResource](#)。

描述堆疊資源

CLI

AWS CLI

取得堆疊資源的相關資訊

下列 describe-stack-resources 範例顯示指定之堆疊中的資源詳細資訊。

```
aws cloudformation describe-stack-resources \  
  --stack-name my-stack
```

輸出：

```
{  
  "StackResources": [  
    {  
      "StackName": "my-stack",  
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-  
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

        "LogicalResourceId": "bucket",
        "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
        "ResourceType": "AWS::S3::Bucket",
        "Timestamp": "2019-10-02T04:34:11.345Z",
        "ResourceStatus": "CREATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    },
    {
        "StackName": "my-stack",
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "function",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "Timestamp": "2019-10-02T05:34:27.989Z",
        "ResourceStatus": "UPDATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    },
    {
        "StackName": "my-stack",
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
        "LogicalResourceId": "functionRole",
        "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
        "ResourceType": "AWS::IAM::Role",
        "Timestamp": "2019-10-02T04:34:06.350Z",
        "ResourceStatus": "CREATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
]
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeStackResources](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回最多 100 個與指定堆疊相關聯的資源 AWS 描述。若要取得與堆疊相關聯的所有資源詳細資訊，請使用 `Get-CFNStackResourceSummary`，這也支援手動分頁結果。

```
Get-CFNStackResourceList -StackName "myStack"
```

範例 2：傳回邏輯 ID "Ec2Instance" 在與指定堆疊相關聯的範本中識別的 Amazon EC2 執行個體描述。

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

範例 3：傳回最多 100 個與堆疊相關聯的資源描述，其中包含由執行個體 ID "i-123456" 識別的 Amazon EC2 執行個體。若要取得與堆疊相關聯的所有資源詳細資訊，請使用 `Get-CFNStackResourceSummary`，這也支援手動分頁結果。

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

範例 4：傳回堆疊範本中邏輯 ID "Ec2Instance" 所識別的 Amazon EC2 執行個體描述。堆疊是使用其包含的資源的實體資源 ID 來識別，在此情況下，也會使用執行個體 ID 為 "i-123456" 的 Amazon EC2 執行個體。根據範本內容，也可以使用不同的實體資源來識別堆疊，例如 Amazon S3 儲存貯體。

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [DescribeStackResources](#)。

Tools for PowerShell V5

範例 1：傳回最多 100 個與指定堆疊相關聯的資源 AWS 描述。若要取得與堆疊相關聯的所有資源詳細資訊，請使用 `Get-CFNStackResourceSummary`，這也支援手動分頁結果。

```
Get-CFNStackResourceList -StackName "myStack"
```

範例 2：傳回邏輯 ID "Ec2Instance" 在與指定堆疊相關聯的範本中識別的 Amazon EC2 執行個體描述。

```
Get-CFNStackResourceList -StackName "myStack" -LogicalResourceId "Ec2Instance"
```

範例 3：傳回最多 100 個與堆疊相關聯的資源描述，其中包含由執行個體 ID "i-123456" 識別的 Amazon EC2 執行個體。若要取得與堆疊相關聯的所有資源詳細資訊，請使用 `Get-CFNStackResourceSummary`，這也支援手動分頁結果。

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456"
```

範例 4：傳回堆疊範本中邏輯 ID "Ec2Instance" 所識別的 Amazon EC2 執行個體描述。堆疊是使用其包含的資源的實體資源 ID 來識別，在此情況下，也會使用執行個體 ID 為 "i-123456" 的 Amazon EC2 執行個體。根據範本內容，也可以使用不同的實體資源來識別堆疊，例如 Amazon S3 儲存貯體。

```
Get-CFNStackResourceList -PhysicalResourceId "i-123456" -LogicalResourceId  
"Ec2Instance"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [DescribeStackResources](#)。

描述堆疊

CLI

AWS CLI

描述 AWS CloudFormation 堆疊

以下 `describe-stacks` 範例顯示 `myteststack` 堆疊的摘要資訊：

```
aws cloudformation describe-stacks --stack-name myteststack
```

輸出：

```
{  
  "Stacks": [  
    {  
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
```



```
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if
you create a stack from this template.",
        "Tags": [],
        "Outputs": [
            {
                "Description": "Name of S3 bucket to hold website content",
                "OutputKey": "BucketName",
                "OutputValue": "myteststack-s3bucket-jssofilzie2w"
            }
        ],
        "StackStatusReason": null,
        "CreationTime": "2013-08-23T01:02:15.422Z",
        "Capabilities": [],
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE",
        "DisableRollback": false
    }
}
]
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的堆疊。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeStacks](#)。

Go

SDK for Go V2

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
import (
    "context"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
```

```
"github.com/aws/aws-sdk-go-v2/service/cloudformation"
)

// StackOutputs defines a map of outputs from a specific stack.
type StackOutputs map[string]string

type CloudFormationActions struct {
    CfnClient *cloudformation.Client
}

// GetOutputs gets the outputs from a CloudFormation stack and puts them into a
// structured format.
func (actor CloudFormationActions) GetOutputs(ctx context.Context, stackName
    string) StackOutputs {
    output, err := actor.CfnClient.DescribeStacks(ctx,
        &cloudformation.DescribeStacksInput{
            StackName: aws.String(stackName),
        })
    if err != nil || len(output.Stacks) == 0 {
        log.Panicf("Couldn't find a CloudFormation stack named %v. Here's why: %v\n",
            stackName, err)
    }
    stackOutputs := StackOutputs{}
    for _, out := range output.Stacks[0].Outputs {
        stackOutputs[*out.OutputKey] = *out.OutputValue
    }
    return stackOutputs
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 [DescribeStacks](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回描述所有使用者堆疊的堆疊執行個體集合。

```
Get-CFNStack
```

範例 2：傳回描述指定堆疊的堆疊執行個體

```
Get-CFNStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [DescribeStacks](#)。

Tools for PowerShell V5

範例 1：傳回描述所有使用者堆疊的堆疊執行個體集合。

```
Get-CFNStack
```

範例 2：傳回描述指定堆疊的堆疊執行個體

```
Get-CFNStack -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [DescribeStacks](#)。

根據預設，describe-stacks 命令會傳回參數值。若要避免傳回敏感參數值，如密碼，請在 CloudFormation 範本中包括 NoEcho 屬性，並將其設定為 TRUE。

Important

使用 NoEcho 屬性不會遮罩任何儲存在下列資訊中的資訊：

- Metadata 範本區段。CloudFormation 不會轉換、修改或標記您在 Metadata 區段中包含的任何資訊。若要取得更多資訊，請參閱 [中繼資料](#)。
- Outputs 範本區段。如需詳細資訊，請參閱 [輸出](#)。
- 資源定義的 Metadata 屬性。如需詳細資訊，請參閱 [Metadata 屬性](#)。

我們強烈建議您不要使用這些機制來包含敏感資訊，例如密碼或秘密。

Important

我們建議您不要直接在 CloudFormation 範本中嵌入敏感資訊，而是在堆疊範本中使用動態參數來參考在 CloudFormation 外部存放和管理的敏感資訊，例如在 AWS Systems Manager 參數存放區或 AWS Secrets Manager。

如需詳細資訊，請參閱[請勿在範本中內嵌認證](#)的最佳實務。

取得範本

CLI

AWS CLI

檢視 an AWS CloudFormation 堆疊的範本內文

下列 `get-template` 命令示範 `myteststack` 堆疊的範本：

```
aws cloudformation get-template --stack-name myteststack
```

輸出：

```
{
  "TemplateBody": {
    "AWSTemplateFormatVersion": "2010-09-09",
    "Outputs": {
      "BucketName": {
        "Description": "Name of S3 bucket to hold website content",
        "Value": {
          "Ref": "S3Bucket"
        }
      }
    },
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING** This
template creates an S3 bucket. You will be billed for the AWS resources used if
you create a stack from this template.",
    "Resources": {
      "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
          "AccessControl": "PublicRead"
        }
      }
    }
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [GetTemplate](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回與指定堆疊相關聯的範本。

```
Get-CFNTemplate -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [GetTemplate](#)。

Tools for PowerShell V5

範例 1：傳回與指定堆疊相關聯的範本。

```
Get-CFNTemplate -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [GetTemplate](#)。

列出堆疊資源

CLI

AWS CLI

列出堆疊中的資源

下列命令顯示指定堆疊中的資源清單。

```
aws cloudformation list-stack-resources \  
  --stack-name my-stack
```

輸出：

```
{  
  "StackResourceSummaries": [  

```

```

    {
      "LogicalResourceId": "bucket",
      "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
      "ResourceType": "AWS::S3::Bucket",
      "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "LogicalResourceId": "function",
      "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
      "ResourceType": "AWS::Lambda::Function",
      "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
      "ResourceStatus": "UPDATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    },
    {
      "LogicalResourceId": "functionRole",
      "PhysicalResourceId": "my-functionRole-HIZXMPLE0M9E",
      "ResourceType": "AWS::IAM::Role",
      "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
      "ResourceStatus": "CREATE_COMPLETE",
      "DriftInformation": {
        "StackResourceDriftStatus": "IN_SYNC"
      }
    }
  ]
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListStackResources](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回與指定堆疊相關聯的所有資源的描述。

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [ListStackResources](#)。

Tools for PowerShell V5

範例 1：傳回與指定堆疊相關聯的所有資源的描述。

```
Get-CFNStackResourceSummary -StackName "myStack"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [ListStackResources](#)。

列出堆疊

使用 [list-stacks](#) 命令列出堆疊。若要僅列出具有指定狀態碼的堆疊，請包含 `--stack-status-filter` 選項。您可以為 `--stack-status-filter` 選項指定一或多個堆疊狀態碼。如需詳細資訊，請參閱[堆疊狀態碼](#)。

CLI

AWS CLI

列出 AWS CloudFormation 堆疊

以下 `list-stacks` 命令示範狀態為 `CREATE_COMPLETE` 之所有堆疊的摘要：

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

輸出：

```
[
  {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
    "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket:
Sample template showing how to create a publicly accessible S3 bucket.
**WARNING** This template creates an S3 bucket. You will be billed for the AWS
resources used if you create a stack from this template.",
    "StackStatusReason": null,
    "CreationTime": "2013-08-26T03:27:10.190Z",
    "StackName": "myteststack",
```

```
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListStacks](#)。

PowerShell

Tools for PowerShell V4

範例 1：傳回所有堆疊的摘要資訊。

```
Get-CFNStackSummary
```

範例 2：傳回目前正在建立之所有堆疊的摘要資訊。

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

範例 3：傳回目前正在建立或更新之所有堆疊的摘要資訊。

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
    "UPDATE_IN_PROGRESS")
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [ListStacks](#)。

Tools for PowerShell V5

範例 1：傳回所有堆疊的摘要資訊。

```
Get-CFNStackSummary
```

範例 2：傳回目前正在建立之所有堆疊的摘要資訊。

```
Get-CFNStackSummary -StackStatusFilter "CREATE_IN_PROGRESS"
```

範例 3：傳回目前正在建立或更新之所有堆疊的摘要資訊。

```
Get-CFNStackSummary -StackStatusFilter @("CREATE_IN_PROGRESS",  
    "UPDATE_IN_PROGRESS")
```


- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [ListStacks](#)。

更新堆疊

使用 [update-stack](#) 命令直接更新堆疊。您可以指定要更新的堆疊、參數值和功能；如果您希望使用已更新的範本，也可以指定範本的名稱。如需詳細資訊，請參閱[直接更新堆疊](#)。

CLI

AWS CLI

更新 AWS CloudFormation 堆疊

以下 update-stack 命令更新 mystack 堆疊的範本和輸入參數：

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template --  
parameters ParameterKey=KeyPairName,ParameterValue=SampleKeyPair  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

以下 update-stack 命令僅更新 mystack 堆疊的 SubnetIDs 參數值。如果您未指定參數值，則會使用範本中所指定的預設值：

```
aws cloudformation update-stack --stack-name mystack --  
template-url https://s3.amazonaws.com/sample/updated.template  
--parameters ParameterKey=KeyPairName,UsePreviousValue=true  
ParameterKey=SubnetIDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

以下 update-stack 命令會將兩個堆疊通知主題新增至 mystack 堆疊：

```
aws cloudformation update-stack --stack-name mystack --use-previous-template --  
notification-arns "arn:aws:sns:us-east-1:123456789012:mytopic1" "arn:aws:sns:us-east-1:123456789012:mytopic2"
```

如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的 [AWS CloudFormation 堆疊更新](#)。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [UpdateStack](#)。

PowerShell

Tools for PowerShell V4

範例 1：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 代表範本中宣告的參數名稱，而 'PV1' 代表其值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 2：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
    -TemplateBody "{Template Content Here}" `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

範例 3：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 代表範本中宣告的參數名稱，而 'PV2' 代表其值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 4：使用從 Amazon S3 取得的指定範本和自訂參數來更新堆疊 'myStack'。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
    -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
    -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
    @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

範例 5：更新堆疊 'myStack'，在本範例中假設包含 IAM 資源，具有從 Amazon S3 取得的指定範本和自訂參數。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。

自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。包含 IAM 資源的堆疊需要指定 -Capabilities "CAPABILITY_IAM" 參數，否則更新會失敗，並顯示 'InsufficientCapabilities' 錯誤。

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
                -Capabilities "CAPABILITY_IAM"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [UpdateStack](#)。

Tools for PowerShell V5

範例 1：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 代表範本中宣告的參數名稱，而 'PV1' 代表其值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 2：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
                -TemplateBody "{Template Content Here}" `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
@{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

範例 3：使用指定的範本和自訂參數更新堆疊 'myStack'。'PK1' 代表範本中宣告的參數名稱，而 'PV2' 代表其值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" -TemplateBody "{Template Content Here}" -
Parameters @{ ParameterKey="PK1"; ParameterValue="PV1" }
```

範例 4：使用從 Amazon S3 取得的指定範本和自訂參數來更新堆疊 'myStack'。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                              @{ ParameterKey="PK2"; ParameterValue="PV2" } )
```

範例 5：更新堆疊 'myStack'，在本範例中假設包含 IAM 資源，具有從 Amazon S3 取得的指定範本和自訂參數。'PK1' 和 'PK2' 代表範本中宣告的參數名稱，'PV1' 和 'PV2' 代表其請求的值。自訂參數也可以使用 'Key' 和 'Value' (而不是 'ParameterKey' 和 'ParameterValue') 來指定。包含 IAM 資源的堆疊需要指定 -Capabilities "CAPABILITY_IAM" 參數，否則更新會失敗，並顯示 'InsufficientCapabilities' 錯誤。

```
Update-CFNStack -StackName "myStack" `
                -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template `
                -Parameter @( @{ ParameterKey="PK1"; ParameterValue="PV1" },
                              @{ ParameterKey="PK2"; ParameterValue="PV2" } ) `
                -Capabilities "CAPABILITY_IAM"
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [UpdateStack](#)。

Note

要移除所有通知，為 --notification-arns 選項指定 []。

驗證您的範本

使用 [validate-template](#) 命令檢查您的範本檔案中的語法錯誤。

在驗證期間，CloudFormation 會先檢查範本是否為有效的 JSON。如果不是，CloudFormation 會檢查範本是否為有效的 YAML。如果兩種檢查都失敗，則 CloudFormation 會傳回範本驗證錯誤。

CLI

AWS CLI

驗證 an AWS CloudFormation 範本

下列 `validate-template` 命令可驗證 `sampletemplate.json` 範本：

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

輸出：

```
{
  "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
showing how to create a publicly accessible S3 bucket. **WARNING** This template
creates an S3 bucket. You will be billed for the AWS resources used if you
create a stack from this template.",
  "Parameters": [],
  "Capabilities": []
}
```

如需詳細資訊，請參閱《AWS CloudFormation AWS 使用者指南》中的使用 CloudFormation 範本。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ValidateTemplate](#)。

PowerShell

Tools for PowerShell V4

範例 1：驗證指定的範本內容。輸出中詳細說明範本的功能、描述和參數。

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

範例 2：驗證透過 Amazon S3 URL 存取的指定範本。輸出中詳細說明範本的功能、描述和參數。

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/
templatefile.template
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V4)》中的 [ValidateTemplate](#)。

Tools for PowerShell V5

範例 1：驗證指定的範本內容。輸出中詳細說明範本的功能、描述和參數。

```
Test-CFNTemplate -TemplateBody "{TEMPLATE CONTENT HERE}"
```

範例 2：驗證透過 Amazon S3 URL 存取的指定範本。輸出中詳細說明範本的功能、描述和參數。

```
Test-CFNTemplate -TemplateURL https://s3.amazonaws.com/amzn-s3-demo-bucket/  
templatefile.template
```

- 如需 API 詳細資訊，請參閱《AWS Tools for PowerShell Cmdlet 參考 (V5)》中的 [ValidateTemplate](#)。

下列為產生驗證錯誤的回應範例。

```
{  
  "ResponseMetadata": {  
    "RequestId": "4ae33ec0-1988-11e3-818b-e15a6df955cd"  
  },  
  "Errors": [  
    {  
      "Message": "Template format error: JSON not well-formed. (line 11, column  
8)",  
      "Code": "ValidationError",  
      "Type": "Sender"  
    }  
  ],  
  "Capabilities": [],  
  "Parameters": []  
}  
A client error (ValidationError) occurred: Template format error: JSON not well-formed.  
(line 11, column 8)
```

Note

此 `validate-template` 命令旨在僅檢查範本的語法。它不能確保您為資源指定的屬性值對該資源有效。它也不能判斷建立堆疊時將存在的資源數量。

您需要嘗試建立堆疊才能檢查操作有效性。CloudFormation 堆疊沒有沙盒或測試區域，因此您需要為測試期間建立的資源付費。

使用 將本機成品上傳至 S3 儲存貯體 AWS CLI

您可以使用 AWS CLI 將 CloudFormation 範本參考的本機成品上傳至 Amazon S3 儲存貯體。本機成品指的是您在範本中參考的檔案。您不需要手動上傳檔案到 S3 儲存貯體，然後將其位置新增到您的範本，您可以在您的範本中指定本機成品，然後使用 [package](#) 命令來快速上傳。

本機成品是 `package` 命令上傳至 Amazon S3 的檔案或資料夾的路徑。例如，成品可以是 AWS Lambda 函數原始碼的本機路徑或 Amazon API Gateway REST API OpenAPI 檔案。

使用 `package` 命令時：

- 如果您指定檔案，命令會將它直接上傳至 S3 儲存貯體。
- 如果您指定資料夾，命令將建立該資料夾的 `.zip` 檔案，然後上傳 `.zip` 檔案。
- 如果您不指定路徑，命令將建立工作目錄的 `.zip` 檔案，然後上傳。

您可以指定絕對或相對路徑，相對路徑相對於範本的位置。

上傳成品之後，命令會傳回範本的副本，以命令上傳成品的 S3 位置取代本機成品的參考。然後，您可以使用傳回的範本來建立或更新堆疊。

Note

您只能在 `package` 命令支援的資源屬性中使用本機成品。如需有關此命令和受支援資源屬性清單的詳細資訊，請參閱 [AWS CLI 命令參考](#) 中的 [package](#) 文件。

先決條件

開始之前，您必須具備現有的 Amazon S3 儲存貯體。

封裝並部署包含本機成品的範本

以下範本指定用於 Lambda 函數原始程式碼的本機成品。原始程式碼存放在 `/home/user/code/lambdafunction` 資料夾中。

原始範本

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "/home/user/code/lambdafunction"
      }
    }
  }
}
```

下列 [package](#) 命令會建立函數原始碼資料夾的 .zip 檔案，並將其上傳至指定儲存貯體的根目錄。

```
aws cloudformation package \
  --s3-bucket amzn-s3-demo-bucket \
  --template /path_to_template/template.json \
  --output-template-file packaged-template.json \
  --output json
```

命令會在 --output-template-file 指定的路徑上產生新的範本。如下所示，它會將成品參考取代為 Amazon S3 位置。.zip 檔案會使用資料夾內容的 MD5 檢查總和來命名，而非資料夾本身的名稱。

產生範本

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Transform": "AWS::Serverless-2016-10-31",
  "Resources": {
    "MyFunction": {
      "Type": "AWS::Serverless::Function",
      "Properties": {
        "Handler": "index.handler",
        "Runtime": "nodejs18.x",
        "CodeUri": "s3://amzn-s3-demo-bucket/md5_checksum"
      }
    }
  }
}
```



```
}  
}  
}
```

封裝範本的成品後，請使用 [deploy](#) 命令部署已處理的範本。

```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name
```

部署超過 51,200 位元組的範本時，請搭配 `--s3-bucket` 選項使用 `deploy` 命令將範本上傳至 S3，如下列範例所示。

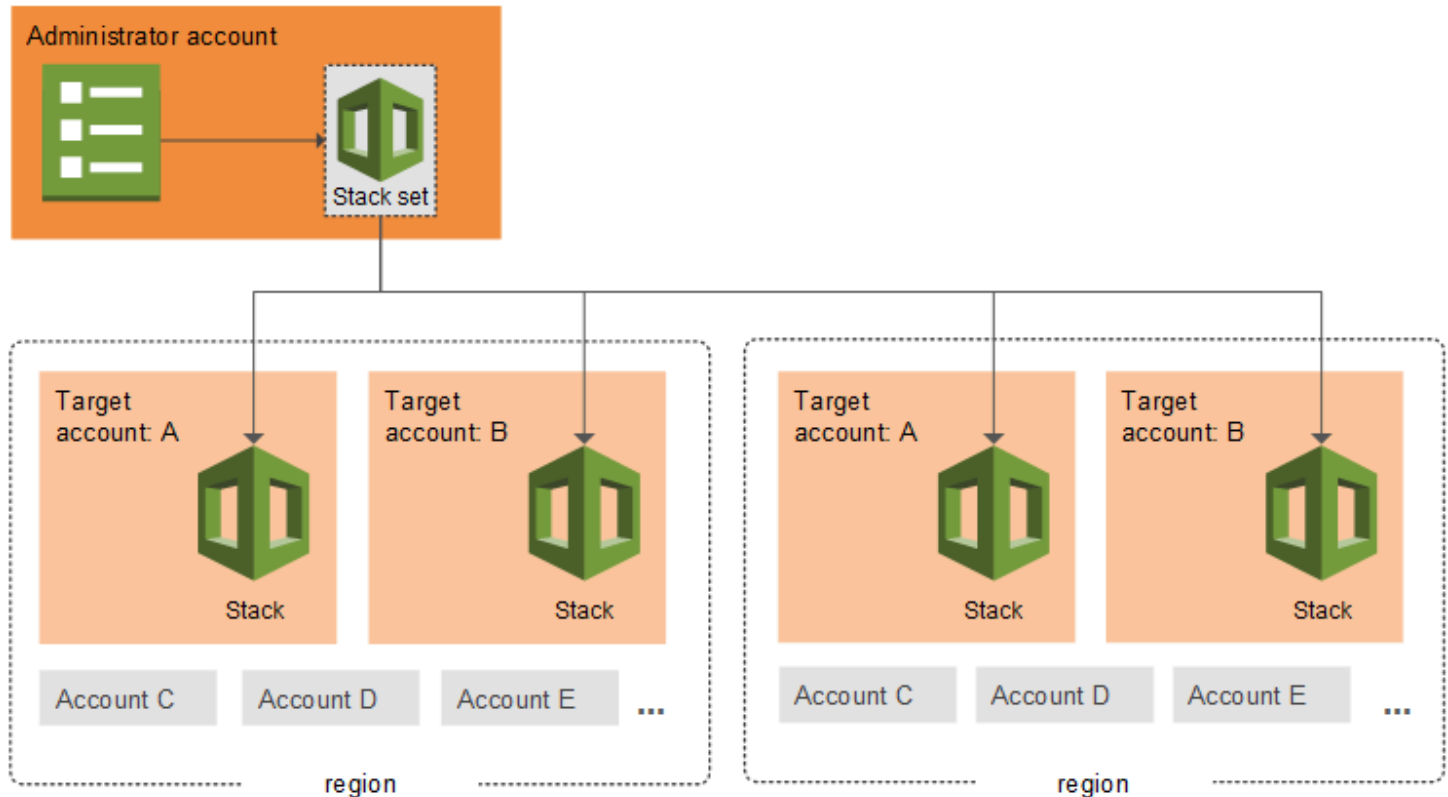
```
aws cloudformation deploy \  
  --template-file packaged-template.json \  
  --stack-name stack-name \  
  --s3-bucket amzn-s3-demo-bucket
```

Note

如需使用 `package` 命令上傳本機成品的另一個範例，請參閱 [運用巢狀堆疊，將範本分割成可重複使用的部分](#)。

透過 StackSets 跨帳戶和區域管理堆疊

CloudFormation StackSets 可讓您跨多個帳戶建立、更新或刪除堆疊，AWS 區域 並使用單一操作來擴展堆疊的功能。您使用管理員帳戶定義與管理 CloudFormation 範本，並使用該範本做為基礎，將堆疊佈建至指定 AWS 區域中選定的目標帳戶。



本節會協助您開始使用 StackSets，解答有關如何使用和疑難排解 StackSet 建立、更新和刪除等常見問題。

主題

- [StackSets 概念](#)
- [使用 CloudFormation StackSets 的先決條件](#)
- [使用範例範本開始使用 StackSets](#)
- [具有自我管理許可的 Create CloudFormation StackSets](#)
- [具有服務受管許可的 Create CloudFormation StackSets](#)
- [在 中啟用或停用 StackSets 的自動部署 AWS Organizations](#)
- [Update CloudFormation StackSets](#)
- [將堆疊新增至 CloudFormation StackSets](#)

- [覆寫 CloudFormation StackSet 中堆疊的參數值](#)
- [從 CloudFormation StackSets 刪除堆疊](#)
- [Delete CloudFormation StackSets](#)
- [使用目標帳戶閘道來防止失敗的 StackSets 部署](#)
- [選擇 CloudFormation StackSets 的並行模式](#)
- [在 CloudFormation StackSets 上執行漂移偵測](#)
- [將堆疊匯入 CloudFormation StackSets](#)
- [使用 CloudFormation StackSets 的最佳實務](#)
- [CloudFormation StackSets 範本範例](#)
- [疑難排解 CloudFormation StackSets](#)

StackSets 概念

下列術語和概念是您了解和使用 StackSets 的核心。

主題

- [管理員和目標帳戶](#)
- [CloudFormation StackSets](#)
- [StackSets 的許可模型](#)
- [堆疊執行個體](#)
- [StackSet 操作](#)
- [StackSet 操作選項](#)
- [Tags \(標籤\)](#)
- [StackSets 狀態碼](#)
- [堆疊執行個體狀態碼](#)

管理員和目標帳戶

管理員帳戶是您 AWS 帳戶 建立 StackSets 的。對於具有服務受管許可的 StackSets，管理員帳戶是組織的 管理帳戶 或委派管理員帳戶。您可以登入建立 StackSet 的 AWS 管理員帳戶來管理 StackSet。

「目標帳戶」是您建立、更新或刪除 StackSet 中一或多個堆疊的帳戶。先要設定管理員和目標帳戶之間的信任關係，才可以使用 StackSet 在目標帳戶中建立堆疊。

CloudFormation StackSets

StackSet 可做為跨指定 AWS 帳戶和區域部署之多個堆疊的容器。每個堆疊都基於相同的 CloudFormation 範本，但您可以使用參數自訂個別堆疊。

定義 StackSet 之後，您可以在目標帳戶中建立、更新或刪除堆疊 AWS 區域並指定堆疊。當您建立、更新或刪除堆疊時，您也可以指定操作偏好設定。例如，包括您要執行操作的區域順序、堆疊操作停止前的失敗容錯閾值，以及同時執行堆疊操作的帳戶數目。

StackSet 是區域性資源。如果您在其中建立 StackSet AWS 區域，您只能在檢視該區域時查看或變更它。

StackSets 的許可模型

您可以使用自我管理許可或服務受管許可來建立 StackSets。

透過自我管理許可，您可以建立跨帳戶和區域部署時 StackSets 所需的 IAM 角色。需要這些角色，才能在您用來管理 StackSet 的帳戶與堆疊執行個體部署目的地帳戶之間建立信任關係。使用此許可模型，StackSets 可以部署到 AWS 帳戶您擁有建立 IAM 角色許可的任何。

透過服務受管許可，您可以將堆疊執行個體部署到 AWS Organizations 所管理的帳戶。使用此許可模型時，您不必建立需要的 IAM 角色；StackSets 會代表您建立 IAM 角色。使用此模型時，您也可以自動部署到未來您新增至組織的帳戶。

AWS Organizations 與 CloudFormation 整合，可協助您在擴展和擴展 AWS 資源時集中管理和控管您的環境。

- 管理帳戶 – 您用於建立組織的帳戶。如需詳細資訊，請參閱 [AWS Organizations 的術語與概念](#)。
- 委派管理員 – 相容 AWS 服務可以將組織中的成員帳戶註冊 AWS 為該服務中組織帳戶的管理員。如需詳細資訊，請參閱 [AWS 可與搭配使用的服務 AWS Organizations](#)。

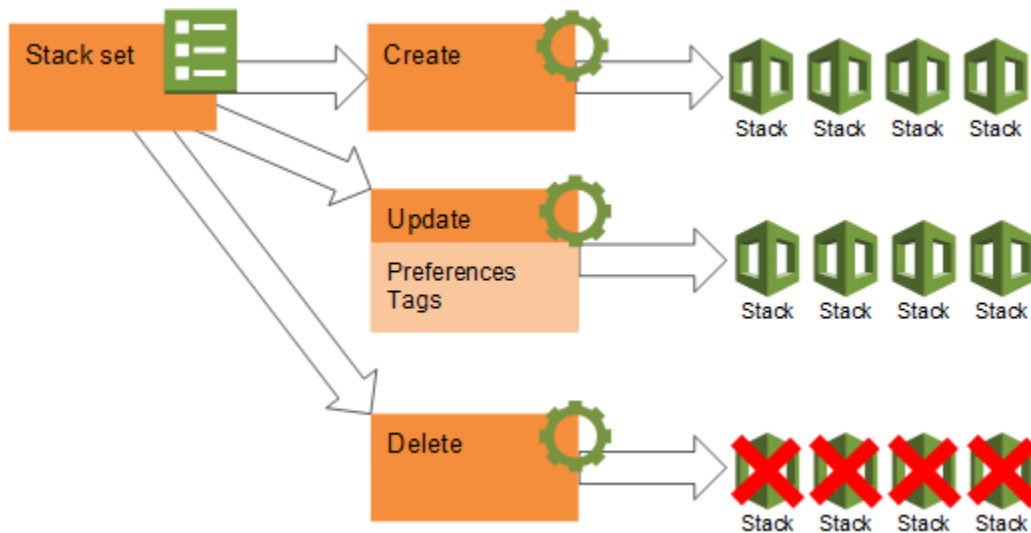
如需使用服務受管許可建立和管理 StackSets 的詳細資訊，請參閱下列主題：

- [使用 啟用 StackSets 的受信任存取 AWS Organizations](#)
- [註冊委派管理員會員帳戶](#)
- [具有服務受管許可的 Create CloudFormation StackSets](#)

堆疊執行個體

「堆疊執行個體」是針對區域內某個目標帳戶中的堆疊所做的參考。堆疊執行個體可以在沒有堆疊的情況下存在。例如，如果因為某種原因而無法建立堆疊，則堆疊執行個體會顯示堆疊建立失敗的原因。堆疊執行個體僅與一個 StackSet 相關聯。

下圖顯示 StackSets、堆疊操作和堆疊之間的邏輯關係。當您更新 StackSet 時，所有帳戶和區域中所有相關聯的堆疊執行個體都會更新。



StackSet 操作

您可以對 StackSets 執行下列操作。

建立 StackSet

建立新的 StackSet 包括指定您要用來建立堆疊的 CloudFormation 範本、指定您要在其中建立堆疊的目標帳戶，以及識別您要在目標帳戶中部署堆疊 AWS 區域的。StackSet 可確保在您選擇區域內的所有指定目標帳戶，相同的堆疊資源皆有一致的部署及相同的設定。

更新 StackSet

當您更新 StackSet 時，是將變更推送到 StackSet 中的堆疊。您可使用下列其中一種方法來更新 StackSet。您的範本更新一律影響所有堆疊；您無法選擇性更新 StackSet 中某些堆疊的範本，讓其他範本不受影響。

- 變更範本中的現有設定或新增新的資源，例如更新特定服務的參數設定，或者新增新的 Amazon EC2 執行個體。
- 使用不同的範本來取代範本。

- 跨現有或其他的區域，在現有或其他目標帳戶中新增堆疊。

刪除堆疊

當您刪除堆疊時，是從您指定區域內的指定目標帳戶，移除堆疊及其所有相關聯的資源。您可採用下列方式來刪除堆疊。

- 從某些目標帳戶刪除堆疊，同時讓其他目標帳戶中的其他堆疊繼續執行。
- 從某些區域刪除堆疊，同時讓其他區域中的堆疊繼續執行。
- 從您的 StackSet 刪除堆疊，但選擇保留堆疊選項，將其儲存，以繼續獨立執行您的 StackSet。隨後您可在 CloudFormation 中、StackSet 外，管理保留的堆疊。
- 刪除您 StackSet 中所有堆疊，準備刪除您的整個 StackSet。

刪除 StackSet

只有當 StackSet 中沒有任何堆疊執行個體時才可以刪除它。

StackSet 操作選項

本節所述的這些選項，有助於控制為成功執行 StackSet 操作所允許的故障時間和次數，並可防止您遺失堆疊資源。

同時帳戶上限

可用於建立、更新和刪除工作流程的此項設定，能讓您指定目標帳戶一次可執行之操作的數目或百分比上限。較低的數目或百分比，表示一次在較少的目標帳戶中執行某項操作。操作依 Deployment order (部署順序) 方塊中指定的順序，一次在一個區域中執行。例如，如果您要將堆疊部署到兩個區域內的 10 個目標帳戶，則 Maximum concurrent accounts (同時帳戶上限) 設定為 50 和 By percentage (依百分比) 就會將堆疊部署到第一個區域中的前五個帳戶，接著是第一個區域中的後五個帳戶，然後才會移至下一個區域並開始部署到前五個目標帳戶。

當您選擇 By percentage (依百分比) 時，如果指定的百分比不能代表您指定的帳戶整數，CloudFormation 會無條件捨去。例如，如果您要將堆疊部署到 10 個目標帳戶，而您將 Maximum concurrent accounts (同時帳戶上限) 設為 25 及 By percentage (依百分比)，CloudFormation 就會將同時部署 2.5 個堆疊 (這不可能) 無條件捨去為同時部署兩個堆疊。

請注意，此設定可讓您指定操作的「上限」。針對大型的部署，因為服務調節之故，在某些情況下實際同時執行的帳戶數目會略低。

根據並行模式，並行帳戶上限取決於容錯能力的值。若您的並行模式設定為嚴格容錯能力，則並行帳戶上限最多可比容錯能力設定多一個。

並行模式

此設定可用於建立、更新和刪除工作流程，讓您能夠選擇 StackSet 操作期間並行層級的運作方式。如需詳細資訊，請參閱[選擇 CloudFormation StackSets 的並行模式](#)。

容錯能力

可用於建立、更新和刪除工作流程的此項設定，能讓您指定每區域可發生之堆疊操作故障的數目或百分比上限，超過此值，CloudFormation 就會自動停止操作。較低的數目或百分比表示此操作在較少的堆疊上執行，但您能夠更快開始對失敗的操作進行疑難排解。例如，如果您要更新三個區域內 10 個目標帳戶中的 10 個堆疊，將容錯能力設定為 20 和依百分比，表示操作若要繼續，一個區域最多只能有兩個堆疊更新失敗。如果同一區域中發生第三個堆疊失敗，CloudFormation 就會停止操作。如果第一個區域中的堆疊無法更新，則該區域會繼續更新操作，然後移至下一個區域。如果第二個區域有兩個堆疊無法更新，即達到 20% 的容錯能力；如果該區域發生第三個堆疊更新失敗，CloudFormation 就會停止更新操作，不會推進到後續的區域。

當您選擇 By percentage (依百分比) 時，如果指定的百分比不能代表各個區域內您的堆疊整數，CloudFormation 會無條件捨去。例如，如果您要將堆疊部署到三個區域中的 10 個目標帳戶，而且您將容錯能力值設定為 25 和依百分比，CloudFormation 會將 2.5 個堆疊的容錯能力 (這不可能) 無條件捨去到每區域兩個堆疊的容錯能力。

保留堆疊

可用於刪除堆疊工作流程的此項設定，讓堆疊及其資源即使從 StackSet 移除後，仍能繼續執行。當您保留堆疊時，CloudFormation 會將堆疊留存在個別帳戶和區域中，保持不變。堆疊與 StackSet 會取消關聯，但堆疊及其資源仍儲存。完成刪除堆疊操作後，您可以在建立堆疊的目標帳戶 (非管理員帳戶) 中使用 CloudFormation 管理保留的堆疊。

區域並行

可用於建立、更新和刪除工作流程的此項設定，能讓您選擇如何將 StackSets 部署到區域內。

循序 – 只要一個區域的部署故障不超過指定的容錯能力，就可以依區域部署順序方塊中指定的，一次在一個區域中部署 StackSets 操作。順序部署是預設選項。

平行 – 只要一個區域的部署故障不超過指定的容錯能力，就可以平行在所有指定區域同步部署 StackSets 操作。

Tags (標籤)

您可以在 StackSet 的建立和更新操作期間，透過指定金鑰值對來新增標籤。標籤有助於排序和篩選 StackSet 資源，處理帳單和成本分配。如需如何在 中使用標籤的詳細資訊 AWS，請參閱AWS 帳單與

成本管理 《使用者指南》中的[使用成本分配標籤組織和追蹤 AWS 成本](#)。在您指定索引鍵/值組之後，請選擇 + 以儲存標籤。您可以選擇標籤右邊的紅色 X 來刪除您不再適用的標籤。

您套用至 StackSets 的標籤會套用至所有堆疊及您堆疊建立的資源。您還可以在 CloudFormation 中在僅限堆疊的層級新增標籤，但這些標籤可能不會出現在 StackSets 中。

雖然 StackSets 不會新增任何系統定義的標籤，但任何標籤的金鑰名稱都不應該以字串 `aws:` 開頭。

StackSets 狀態碼

CloudFormation StackSets 會產生 StackSet 操作的狀態碼。

下表說明 StackSet 操作狀態碼。

RUNNING

操作正在進行中。

SUCCEEDED

操作已完成，未超過操作的容錯能力。

FAILED

操作無法完成的堆疊數目已超出使用者定義的容錯能力。在堆疊建立和更新操作期間，您設定的操作容錯能力適用於每個區域。如果某個區域內失敗的堆疊數目超過容錯能力，則區域中的操作狀態會變更為 FAILED。操作的整體狀態也會設為 FAILED，CloudFormation 會取消任何剩餘區域的操作。

QUEUED

[Service-managed permissions] 對於需要一系列操作的自動部署，操作會進入佇列中等待執行。例如：

- 將帳戶從一個組織單位 (OU) OU1 移至另一個組織單位 OU2 會觸發自動部署。StackSets 會執行刪除操作，以從目標區域中的目標 OU1 帳戶刪除堆疊執行個體，並將建立操作排入佇列，以將堆疊執行個體新增至目標區域中的目標 OU2 帳戶。
- 將帳戶 AccountA 新增至 OU 會觸發自動部署。StackSets 會執行建立操作，以將堆疊執行個體新增至目標區域中的 AccountA。如果您在此建立操作執行時將另一個帳戶 AccountB 新增至 OU，StackSets 會將第二個建立操作排入佇列。當第一個建立操作完成時，StackSets 會執行第二個建立操作，以將堆疊執行個體新增至目標區域中的 AccountB。

STOPPING

應使用者請求，正在停止操作。

STOPPED

應使用者請求，已停止操作。

堆疊執行個體狀態碼

CloudFormation StackSets 會產生堆疊執行個體的状态碼。

下表說明 StackSets 內堆疊執行個體的状态碼。

CURRENT

StackSet 的堆疊現為最新。

OUTDATED

因為下列其中一個原因，StackSet 的堆疊不是最新。

- 相關聯堆疊上的 [CreateStackSet](#) 或 [UpdateStackSet](#) 操作失敗。
- 屬於 [CreateStackSet](#) 或 [UpdateStackSet](#) 操作的堆疊，在堆疊建立或更新前即已失敗或停止。

INOPERABLE

[DeleteStackInstances](#) 操作失敗，且堆疊的狀態不穩定。後續的 [UpdateStackSet](#) 操作會排除此狀態的堆疊。您可能需要將 `RetainStacks` 設定為 `true` 來執行 [DeleteStackInstances](#) 操作，以刪除堆疊執行個體，然後手動刪除堆疊。

CANCELLED

指定帳戶和區域中的操作已被取消。這可能是因為使用者已停止 StackSet 操作，或是因為 StackSet 操作超過了容錯能力。

FAILED

指定帳戶和區域中的操作失敗。如果區域內足夠多帳戶中的 StackSet 操作失敗，則可能會超過 StackSet 操作整體的容錯能力。

FAILED_IMPORT

在指定帳戶和區域匯入堆疊執行個體失敗，並使堆疊處於不穩定狀態。一旦修正造成失敗的問題，可重試匯入操作。如果區域內足夠多帳戶中的足夠 StackSet 操作失敗，則可能會超過 StackSet 操作整體的容錯能力。

PENDING

指定帳戶和區域中的操作尚未開始。

RUNNING

指定帳戶和區域中的操作目前正在進行中。

SKIPPED_SUSPENDED_ACCOUNT

已跳過指定帳戶和區域中的操作，因為帳戶在操作時被暫停。

SUCCEEDED

指定帳戶和區域中的操作已順利完成。

使用 CloudFormation StackSets 的先決條件

StackSets 擴充了堆疊功能，讓您可以跨多個帳戶和區域使用單次操作建立、更新或刪除堆疊。

由於 StackSets 跨多個帳戶執行堆疊操作，您需要有 AWS 帳戶中定義的必要許可，才能建立您的第一個 StackSet。

您可以使用自我管理或服務受管許可來管理 StackSets。

- 對於自我管理的 StackSets，必須在每個目標帳戶與 AWS 區域中建立並管理 IAM 角色。如需詳細資訊，請參閱[授與自我管理許可](#)。
- 對於服務管理的 StackSets，您不需要在每個帳戶中手動建立和管理 IAM 角色；為您 AWS 處理角色建立和許可。如需詳細資訊，請參閱[啟用受信任存取](#)。

主題

- [準備在預設停用 AWS 區域 的 中執行 StackSet 操作](#)
- [授與自我管理許可](#)
- [使用 啟用 StackSets 的受信任存取 AWS Organizations](#)

準備在預設停用 AWS 區域 的 中執行 StackSet 操作

AWS 區域 亞太區域（香港）等 2019 年 3 月 20 日之後推出的 預設為停用。您必須為您的帳戶啟用這些區域，才能加以使用。如需詳細資訊，請參閱《AWS 帳戶管理 參考指南》中的[在帳戶中啟用或停用 AWS 區域](#)。

若要從 StackSet 的管理員帳戶 (如果使用自我管理許可) 或組織的管理帳戶 (如果使用服務受管的許可) 建立 StackSet，則必須先針對管理員或管理帳戶啟用該區域。

若要 CloudFormation 讓成功建立或更新堆疊執行個體：

- 目標帳戶必須位於目前已針對該目標帳戶啟用的區域中。
- StackSet 的管理員帳戶或組織的管理帳戶必須啟用與目標帳戶相同的區域。

Important

請注意，在 StackSet 操作期間，管理員和目標帳戶會交換與帳戶本身相關的中繼資料，以及所涉及的 StackSet 和 StackSet 執行個體。

此外，如果您停用包含 StackSet 執行個體所在帳戶的區域，您必須視需要負責刪除任何此類執行個體或資源。此外，請注意，關於停用區域中目標帳戶的中繼資料將會保留在管理員帳戶中。

授與自我管理許可

本主題說明如何建立 StackSets 所需的 IAM 服務角色，以跨帳戶部署，並 AWS 區域具有自我管理許可。需要這些角色，才能在您用來管理 StackSet 的帳戶與堆疊執行個體部署目的地帳戶之間建立信任關係。使用此許可模型，StackSets 可以部署到 AWS 帳戶 您擁有建立 IAM 角色許可的任何。

若要使用服務受管許可，請改為參閱 [啟用受信任存取](#)。

主題

- [自我管理許可概觀](#)
- [為管理員帳戶的所有使用者提供可以在所有目標帳戶中管理堆疊的許可](#)
- [設定 StackSet 操作的進階許可](#)
- [設定全域索引鍵以減輕混淆代理人問題](#)

自我管理許可概觀

建立具有自我管理許可的 StackSet 之前，必須在每個帳戶中建立 IAM 服務角色。

基本步驟為：

1. 決定哪個 AWS 帳戶 是管理員帳戶。

StackSets 是在此管理員帳戶中建立。您在目標帳戶中建立屬於 StackSet 的個別堆疊。

2. 決定您要如何建構 StackSet 的許可。

最簡單 (也是最寬鬆) 的許可組態，是賦予管理員帳戶中所有使用者和群組建立和更新由該帳戶管理之所有 StackSets 的能力。如果您需要更精確的控制，可以設定許可來指定：

- 哪些使用者和群組可以在哪些目標帳戶中執行 StackSet 操作。
- 哪些資源使用者和群組可以包含在他們的 StackSets 中。
- 哪些特定 StackSet 操作的使用者和群組可以執行。

3. 在您的管理員和目標帳戶中建立必要的 IAM 服務角色以定義您要的許可。

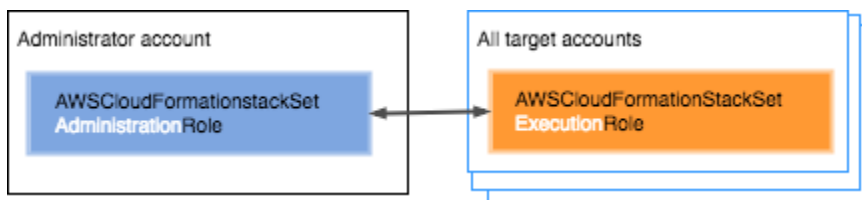
具體而言，這兩個必要角色為：

- `AWSCloudFormationStackSetAdministrationRole` – 此角色已部署至管理員帳戶。
- `AWSCloudFormationStackSetExecutionRole` – 此角色已部署至您建立堆疊執行個體的所有帳戶。

為管理員帳戶的所有使用者提供可以在所有目標帳戶中管理堆疊的許可

本節展示如何為管理員帳戶的所有使用者和群組設定可在所有目標帳戶中執行 StackSet 操作的許可。它會引導您在管理員和目標帳戶中建立所需 IAM 服務角色。管理員帳戶中的任何人則可以在任何目標帳戶中建立、更新或刪除任何堆疊。

若以此方式建構許可，使用者在建立或更新 StackSet 時，將不會傳遞管理員角色。



Administrator account

在管理員帳戶中，建立名為 `AWSCloudFormationStackSetAdministrationRole` 的 IAM 角色。

您可以從 CloudFormation 範本建立堆疊以執行此作業，此範本位於 <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetAdministrationRole.yml>。

Example 範例許可政策

上述範本建立的管理角色包含了下列許可政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Example 範例信任政策 1

上述範本也包含了下列信任政策，授予使用管理角色的服務許可，以及連接至角色的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Example 範例信任政策 2

若要將堆疊執行個體部署到位於預設為停用之區域的目標帳戶中，您也必須包含該區域的區域服務主體。每個預設停用的區域都會有自己的區域服務主體。

下列範例信任政策授予服務許可，以使用亞太區域 (香港) 區域 (ap-east-1) 中的管理角色，該區域預設為停用。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需詳細資訊，請參閱[準備在預設停用 AWS 區域的中執行 StackSet 操作](#)。如需區域代碼清單，請參閱《AWS 一般參考 指南》中的[區域端點](#)。

Target accounts

在每個目標帳戶中，建立信任管理員帳戶且名為 `AWSCloudFormationStackSetExecutionRole` 的服務角色。此角色必須具有此確切名稱。您可以從下列 CloudFormation 範本建立堆疊以執行此作業，此範本位於 <https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/AWSCloudFormationStackSetExecutionRole.yml>。當您使用此範本時，系統會提示您提供管理員帳戶的帳戶 ID，而且您的目標帳戶必須擁有與此管理員帳戶的信任關係。

Important

請注意，此範本會授予管理員存取。在您使用此範本建立目標帳戶執行角色之後，您必須在政策陳述式中，將許可的範圍限制在您使用 StackSets 所建立的資源類型。

目標帳戶服務角色需要許可，以執行您 CloudFormation 範本中指定的任何操作。例如，如果您的範本是建立 S3 儲存貯體，則您需要為 S3 建立新物件的許可。您的目標帳戶一律需要完整的 CloudFormation 許可，包括用於建立、更新、刪除及描述堆疊的許可。

Example範例許可政策 1

此範本建立的角色可在目標帳戶中啟用以下政策。

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

Example 範例許可政策 2

以下範例顯示的政策陳述式提供 StackSets 運作時所需的最低許可。若要在目標帳戶中建立使用 CloudFormation 以外服務資源的堆疊，您必須將這些服務動作和資源新增至每個目標帳戶的 `AWSCloudFormationStackSetExecutionRole` 政策陳述式。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 範例信任政策

以下信任關係是由範本所建立的。管理員帳戶的 ID 會顯示為 `admin_account_id`。

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

```

```
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
        },
        "Action": "sts:AssumeRole"
    }
}
```

您可以設定現有目標帳戶執行角色的信任關係，以信任管理員帳戶中的特定角色。如果您刪除管理員帳戶中的該角色，並建立一個新角色來取代它，您必須設定您的目標帳戶與新管理員帳戶角色 (在前述範例中以 *admin_account_id* 表示) 的信任關係。

設定 StackSet 操作的進階許可

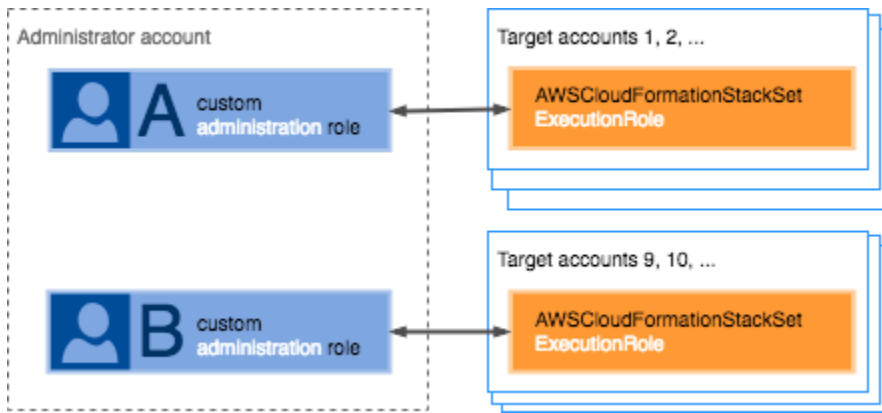
如果您需要對使用者和群組透過單一系統管理員帳戶建立的 StackSets 進行更精細的控制，您可以使用 IAM 角色來指定：

- 哪些使用者和群組可以在哪些目標帳戶中執行 StackSet 操作。
- 哪些資源使用者和群組可以包含在他們的 StackSets 中。
- 哪些特定 StackSet 操作的使用者和群組可以執行。

控制哪些使用者可在特定目標帳戶中執行 StackSet 操作

使用自訂的管理員角色，以控制哪些使用者和群組可執行哪些目標帳戶中的 StackSet 操作。您可能會想要控制管理員帳戶的哪些使用者可執行哪些目標帳戶中的 StackSet 操作。若要這樣做，您要在每個目標帳戶與特定自訂的管理角色之間建立信任關係，而非在管理員帳戶本身建立 `AWSCloudFormationStackSetAdministrationRole` 服務角色。然後，您讓特定使用者和群組在執行特定目標帳戶中的 StackSet 操作時，使用適當的自訂管理角色。

例如，您可以在管理員帳戶中建立角色 A 和角色 B。您可以提供角色 A 透過帳戶 8 以存取目標帳戶 1 的許可。您可以提供角色 B 透過帳戶 16 以存取目標帳戶 9 的許可。



設定與定義自訂管理員角色相關的必要許可，為目標帳戶建立服務角色，並授予使用者許可，以便在執行 StackSet 操作時傳遞自訂管理員的角色。

一般而言，一旦有了必要的許可，其運作方式為：在建立 StackSet 時，使用者必須指定自訂管理角色。使用者必須具有將該角色傳遞至 CloudFormation 的許可。此外，自訂管理員角色必須擁有與為 StackSet 指定之目標帳戶的信任關係。CloudFormation 建立 StackSet，並建立其與自訂管理員角色的關聯。更新 StackSet 時，使用者必須明確指定自訂管理員角色，即使先前對此 StackSet 使用相同的自訂管理角色也是一樣。CloudFormation 將依據上述要求，使用該角色來更新堆疊。

Administrator account

Example範例許可政策

對於每個 StackSet，建立自訂管理角色以及擔任目標帳戶中執行角色的許可。

每個目標帳戶中目標帳戶執行角色的名稱必須相同。若角色名稱為 `AWSCloudFormationStackSetExecutionRole`，則 StackSets 會在建立 StackSet 時自動使用該名稱。若您指定自訂角色名稱，使用者在建立 StackSet 時必須提供執行角色名稱。

使用下列許可政策，以自訂名稱建立 [IAM 服務角色](#)。下列範例中，`Custom_execution_role` 是指目標帳戶中的執行角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],

```

```
        "Resource": [
            "arn:aws:iam::111122223333:role/custom_execution_role"
        ],
        "Effect": "Allow"
    }
}
```

若要在單一陳述式中指定多個帳戶，請用逗號分隔。

```
"Resource": [
    "arn:aws:iam::target_account_id_1:role/custom_execution_role",
    "arn:aws:iam::target_account_id_2:role/custom_execution_role"
]
```

您可使用萬用字元 (*) 而不是帳戶 ID 來指定所有目標帳戶。

```
"Resource": [
    "arn:aws:iam::*:role/custom_execution_role"
]
```

Example 範例信任政策 1

您必須為服務角色提供信任政策，以定義哪些 IAM 主體可擔任該角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Example 範例信任政策 2

若要將堆疊執行個體部署到位於預設為停用之區域的目標帳戶中，您也必須包含該區域的區域服務主體。每個預設停用的區域都會有自己的區域服務主體。

下列範例信任政策授予服務許可，以使用亞太區域 (香港) 區域 (ap-east-1) 中的管理角色，該區域預設為停用。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "cloudformation.amazonaws.com",
          "cloudformation.ap-east-1.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

如需詳細資訊，請參閱[準備在預設停用 AWS 區域的中執行 StackSet 操作](#)。如需區域代碼清單，請參閱《AWS 一般參考指南》中的[區域端點](#)。

Example 傳遞角色政策範例

您也需要為您的 IAM 使用者制定 IAM 許可政策，以允許使用者在執行 StackSet 操作時傳遞自訂管理角色。如需詳細資訊，請參閱[授予使用者將角色傳遞至 AWS 服務](#)。

在以下範例中，*customized_admin_role* 是指使用者必須傳遞的管理員角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",

```

```
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::*:role/customized_admin_role"
    }
  ]
}
```

Target accounts

在每個目標帳戶中，建立信任您要在此帳戶使用之自訂管理角色的服務角色。

目標帳戶服務角色需要許可，以執行您 CloudFormation 範本中指定的任何操作。例如，如果您的範本是建立 S3 儲存貯體，則您需要在 S3 中建立新物件的許可。您的目標帳戶一律需要完整的 CloudFormation 許可，包括用於建立、更新、刪除及描述堆疊的許可。

每個目標帳戶中目標帳戶角色的名稱必須相同。若角色名稱為 `AWSCloudFormationStackSetExecutionRole`，則 `StackSets` 會在建立 `StackSet` 時自動使用該名稱。若您指定自訂角色名稱，使用者在建立 `StackSet` 時必須提供執行角色名稱。

Example範例許可政策

以下範例顯示的政策陳述式提供 `StackSets` 運作時所需的最低許可。若要在目標帳戶中建立使用 CloudFormation 以外服務資源的堆疊，您必須將這些服務動作和資源新增至許可政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example範例信任政策

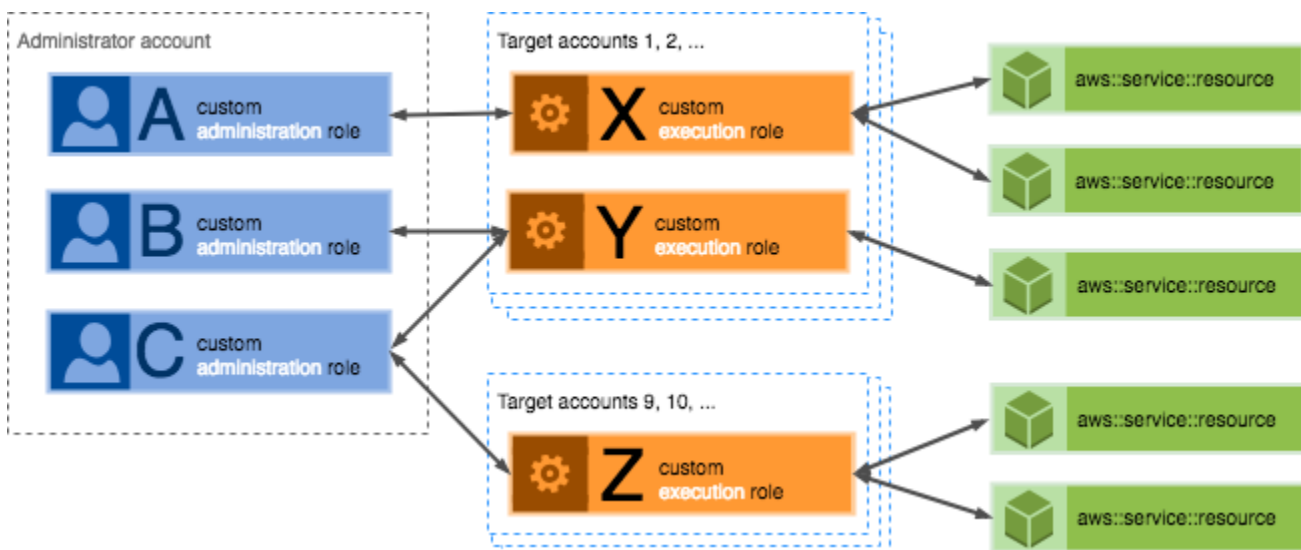
當您建立角色以定義信任關係時，必須提供以下信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

控制使用者可以在特定 StackSets 中包含的資源

使用自訂的執行角色來控制哪些資源堆疊使用者和群組可以包含在其 StackSets 中。例如，您可能想要設定一個群組，只包含他們建立的 StackSets 中與 Amazon S3 相關的資源，而另一個團隊只要包含 DynamoDB 資源。若要這樣做，您要在每個群組的自訂管理員角色與每個資源集的自訂執行角色之間建立信任關係。自訂的執行角色定義了 StackSets 中可以包含哪些堆疊資源。自訂的管理員的角色位於管理員帳戶中，而自訂的執行角色位於每個目標帳戶中 (您要在目標帳戶中使用定義的資源建立 StackSets)。然後，您啟用特定使用者和群組，以便在執行 StackSets 操作時使用適當的自訂管理角色。

例如，您可以在管理員帳戶中建立自訂的管理員角色 A、B、C。具有角色 A 使用許可的使用者和群組，可以建立包含自訂執行角色 X 中具體列出的堆疊資源的 StackSets，但不包含角色 Y 或 Z 的堆疊資源或任何執行角色不包含的資源。



更新 StackSet 時，使用者必須明確指定自訂管理員角色，即使先前對此 StackSet 使用相同的自訂管理角色也是一樣。CloudFormation 會使用指定的自訂管理員角色來執行更新，只要使用者具有在該 StackSet 上執行操作的許可。

同樣地，使用者也可以指定自訂的執行角色。如果指定自訂執行角色，CloudFormation 將依據上述要求，使用該角色更新堆疊。如果使用者未指定自訂執行角色，CloudFormation 將使用之前與 StackSet 關聯之自訂執行角色執行更新，只要該使用者擁有在該 StackSet 上執行操作的許可。

Administrator account

在您的管理員帳戶中建立自訂的管理角色，如 [控制哪些使用者可在特定目標帳戶中執行 StackSet 操作](#) 中所述。加入您要使用的自訂管理角色與自訂執行角色之間的信任關係。

Example 範例許可政策

以下範例會是許可政策，用於您為目標帳戶定義的 `AWSCloudFormationStackSetExecutionRole` 以及自訂的執行角色。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1487980684000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AWSCloudFormationStackSetExecutionRole",
        "arn:aws:iam::*:role/custom_execution_role"
      ]
    }
  ]
}
```

Target accounts

在您要建立 StackSets 的目標帳戶中，建立自訂執行角色，用來將許可授予您想讓使用者和群組可以包含在 StackSets 中的服務和資源。

Example範例許可政策

以下範例提供最低許可給 StackSets，以及建立 Amazon DynamoDB 資料表的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:createTable"
      ],
      "Resource": "*"
    }
  ]
}
```

Example範例信任政策

當您建立角色以定義信任關係時，必須提供以下信任政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/customized_admin_role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

設定特定 StackSet 操作的許可

此外，您可以設定許可來控制哪些使用者和群組可以執行特定 StackSets 操作，例如建立、更新或刪除 StackSets 或堆疊執行個體。如需詳細資訊，請參閱服務授權參考中的 [CloudFormation 的動作、資源和條件索引鍵](#)。

設定全域索引鍵以減輕混淆代理人問題

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在 AWS 中，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了防止這種情況，AWS 提供工具，協助您保護所有服務的資料，讓服務主體能夠存取您帳戶中的資源。

我們建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，以限制 CloudFormation StackSets 為資源提供其他服務的許可。如果同時使用全域條件內容金鑰，則在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的帳戶時，必須使用相同的帳戶 ID。

防範混淆代理人問題最有效的方法，是使用 `aws:SourceArn` 全域條件內容金鑰，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 `aws:SourceArn` 全域條件內容金鑰，同時使用萬用字元 (*) 表示 ARN 的未知部分。例如 `arn:aws:cloudformation::123456789012:*`。盡可能使用 `aws:SourceArn`，這更為具體。僅當您無法判斷正確的 ARN 或 ARN 模式時才使用 `aws:SourceAccount`。

當 StackSets 在您的管理員帳戶中擔任管理角色時，StackSets 會填入您的管理員帳戶 ID 和 StackSets Amazon Resource Name (ARN)。因此，您可以定義信任關係中的 [全域索引鍵](#) `aws:SourceAccount` 和 `aws:SourceArn`，以防止 [混淆代理人問題](#)。下列範例示範如何使用 StackSets 中的 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵，以預防混淆代理人問題。

Administrator account

Example `aws:SourceAccount` 和 `aws:SourceArn` 全域索引鍵

當使用 StackSets 時，在您的 `AWSCloudFormationStackSetAdministrationRole` 信任政策中定義 `aws:SourceAccount` 和 `aws:SourceArn` 全域索引鍵，以防止混淆代理人問題。

```
{  
  "Version": "2012-10-17",
```



```

"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "cloudformation.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cloudformation:*:111122223333:stackset/*"
      }
    }
  }
]
}

```

Example StackSets ARN

指定關聯的 StackSets ARN 以進行更精細的控制。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": [
            "arn:aws:cloudformation:STACKSETS-REGION:111122223333:stackset/STACK-SET-ID-1",
            "arn:aws:cloudformation:STACKSETS-REGION:111122223333:stackset/STACK-SET-ID-2"
          ]
        }
      }
    }
  ]
}

```

```
]
}
}
}
]
```

使用 啟用 StackSets 的受信任存取 AWS Organizations

本主題提供如何使用 啟用受信任存取的指示 AWS Organizations，這是 StackSets 跨帳戶部署及使用 AWS 區域 服務受管許可所必需。要使用自我管理許可，請改為參閱 [授與自我管理許可](#)。

建立具有服務受管許可的 StackSet 之前，您必須先完成下列任務：

- [啟用 中的所有功能](#) AWS Organizations。如果只啟用合併帳單功能，則無法建立具有服務受管許可的 StackSet。
- 使用 啟用受信任存取 AWS Organizations。透過此動作，可讓 CloudFormation 在管理帳戶中建立服務連結角色。啟用受信任存取後，當您建立具有服務受管許可的 StackSet 時，CloudFormation 會在目標 (會員) 帳戶中建立必要的服務連結角色和名為 `stacksets-exec-*` 的服務角色。

啟用受信任存取後，管理帳戶和委派管理員帳戶可以為其組織建立和管理服務受管的 StackSets。

要啟用受信任存取，您必須是管理帳戶中的管理員使用者。管理員使用者是對您的 AWS 帳戶具有完整許可的使用者。如需詳細資訊，請參閱《AWS 帳戶管理 參考指南》中的[建立管理員使用者](#)。如需保護管理帳戶安全的建議，請參閱《AWS Organizations 使用者指南》中的[管理帳戶的最佳實務](#)。

啟用受信任存取

1. 以管理帳戶的管理員 AWS 身分登入，並在 開啟 CloudFormation 主控台<https://console.aws.amazon.com/cloudformation>。
2. 從導覽窗格選擇 StackSets。如果停用受信任存取，則會顯示橫幅來提示您啟用受信任存取。

CloudFormation > StackSets

① Activate trusted access with AWS Organizations to use service-managed permissions. [Learn more](#)

Activate trusted access

3. 選擇啟用受信任存取。

出現下列橫幅時，表示已成功啟用受信任存取。

☑ Successfully activated trusted access with AWS Organizations. To change this setting, please work through the AWS Organizations service console.



Note

「啟動 Organizations 存取」與「啟用 Organizations 存取」相同，「停用 Organizations 存取」與「禁用 Organizations 存取」相同。這些條款已根據行銷指導方針更新。

停用受信任存取

請參閱《AWS Organizations 使用者指南》中的 [CloudFormation StackSets](#) 和 [AWS Organizations](#)。

您必須先取消註冊所有委派管理員 AWS Organizations，才能使用 停用受信任的存取。如需詳細資訊，請參閱[註冊委派管理員](#)。

Note

如需有關使用 API 操作而非主控台來啟用或停用受信任存取的資訊，請參閱：

- [ActivateOrganizationsAccess](#)
- [DeactivateOrganizationsAccess](#)
- [DescribeOrganizationsAccess](#)

服務連結角色

管理帳戶使用 `AWSServiceRoleForCloudFormationStackSetsOrgAdmin` 服務連結角色。只有在停用 AWS Organizations 的受信任存取時，才能修改或刪除此角色。

每個目標帳戶會使用 `AWSServiceRoleForCloudFormationStackSetsOrgMember` 服務連結角色。您只能在兩個條件下修改或刪除此角色：如果 AWS Organizations 已停用的受信任存取，或者如果帳戶已從目標組織或組織單位 (OU) 中移除。

如需詳細資訊，請參閱 AWS Organizations 使用者指南中的 [CloudFormation StackSets](#) 和 [AWS Organizations](#)。

註冊委派管理員會員帳戶

除了組織的管理帳戶外，具有委派管理員許可的會員帳戶還可以建立和管理具有組織服務受管許可的 StackSets。具有服務受管許可的 StackSets 會建立在管理帳戶中，包括由委派管理員建立的

StackSets。若要註冊為組織的委派管理員，您的成員帳戶必須在組織中。如需加入組織的詳細資訊，請參閱[邀請 AWS 帳戶 加入您的組織](#)。

您的組織一次最多可以有五個註冊的委派管理員。委派管理員可以選擇部署至組織或特定 OU 中的所有帳戶。AWS Organizations 必須先啟用的受信任存取，委派管理員才能部署到 Organizations 管理的帳戶。如需詳細資訊，請參閱[使用 啟用 StackSets 的受信任存取 AWS Organizations](#)。

Important

請注意以下事項：

- 委派管理員具有部署至組織中帳戶的完整許可。管理帳戶無法限制部署至特定 OU 或執行特定 StackSet 操作的委派管理員許可。
- 請確定您的委派管理員具有 `organizations:ListDelegatedAdministrators` 許可以避免任何潛在錯誤。

您可以在下列區域中為您的組織註冊委派管理員：美國東部（俄亥俄）、美國東部（維吉尼亞北部）、美國西部（加利佛尼亞北部）、美國西部（奧勒岡）、亞太區域（孟買）、亞太區域（首爾）、亞太區域（新加坡）、亞太區域（雪梨）、亞太區域（東京）、加拿大（中部）、歐洲（法蘭克福）、歐洲（愛爾蘭）、歐洲（倫敦）、歐洲（巴黎）、歐洲（斯德哥爾摩）、以色列（特拉維夫）、南美洲（聖保羅）、AWS GovCloud（美國東部）和 AWS GovCloud（美國西部）。

您可以使用 [CloudFormation 主控台](#)、[AWS CLI](#) 或 [AWS 開發套件](#) 註冊和取消註冊委派管理員。

註冊委派管理員 (主控台)

- 以管理帳戶的管理員 AWS 身分登入，並在開啟 CloudFormation 主控台<https://console.aws.amazon.com/cloudformation/>。
- 從導覽窗格選擇 StackSets。
- 在 Delegated administrators (委派管理員) 下，選擇 Register delegated administrator (註冊委派管理員)。
- 在 Register delegated administrator (註冊委派管理員) 對話方塊中，選擇 Register delegated administrator (註冊委派管理員)。

成功訊息表示成員帳戶已成功註冊為委派管理員。

取消註冊委派管理員 (主控台)

1. 以管理帳戶的管理員 AWS 身分登入，並在 開啟 CloudFormation 主控台 <https://console.aws.amazon.com/>。
2. 從導覽窗格選擇 StackSets。
3. 在 Delegated administrators (委派管理員) 下，選取您要取消註冊的帳戶，然後選擇 Deregister (取消註冊)。

成功訊息表示成員帳戶已成功取消註冊為委派系統管理員。

您可以隨時重新註冊此帳戶。

註冊委派管理員 (AWS CLI)

1. 開啟 AWS CLI。
2. 執行 register-delegated-administrator 命令。

```
$ aws organizations register-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. 執行 list-delegated-administrators 命令以確認指定的成員帳戶已成功註冊為委派管理員。

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

取消註冊委派管理員 (AWS CLI)

1. 開啟 AWS CLI。
2. 執行 deregister-delegated-administrator 命令。

```
$ aws organizations deregister-delegated-administrator \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com \
  --account-id="memberAccountId"
```

3. 執行 list-delegated-administrators 命令以確認指定的成員帳戶已成功取消註冊為委派管理員。

```
$ aws organizations list-delegated-administrators \
  --service-principal=member.org.stacksets.cloudformation.amazonaws.com
```

您可以隨時重新註冊此帳戶。

使用範例範本開始使用 StackSets

本教學課程將協助您開始使用 StackSets AWS 管理主控台。它會引導您使用範例範本來建立 StackSet。您將了解如何跨多個區域部署堆疊、監控 StackSet 操作以及檢視結果。

在本教學課程中，您將建立 StackSet，在美國西部（奧勒岡）區域 (us-west-2) 和美國東部（維吉尼亞北部）區域 () AWS 帳戶的 AWS Config 中啟用 us-east-1。使用 StackSets，您可透過單一操作跨多個帳戶和區域建立、更新或刪除堆疊，使其成為大規模管理基礎設施的理想解決方案。儘管本教學課程為了簡單起見而使用單一帳戶，但它有效地示範了 StackSets 的多區域功能。

範例範本位於以下 S3 儲存貯體：<https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnableAWSConfig.yml>。

Note

StackSets 是免費的，但您需要支付使用它建立 AWS 的資源的費用，例如在本教學 AWS Config 課程中。如需詳細資訊，請參閱 [AWS Config 定價](#)。

主題

- [先決條件](#)
- [從主控台使用範例範本建立 StackSet](#)
- [監控 StackSet 建立](#)
- [檢視 StackSet 結果](#)
- [更新您的 StackSet](#)
- [將堆疊新增至 StackSet](#)
- [清除](#)
- [後續步驟](#)

先決條件

在開始教學課程前，請確定您已完成以下先決條件：

- 您必須已設定自我管理許可所需的 IAM 角色。若要在單一帳戶中建立 StackSet 和部署堆疊，需要帳戶中的下列角色：
 - AWSCloudFormationStackSetAdministrationRole
 - AWSCloudFormationStackSetExecutionRole

如需設定這些角色的詳細說明，請參閱 [授與自我管理許可](#)。

從主控台使用範例範本建立 StackSet

建立可啟用的 StackSet AWS Config

1. 開啟 [CloudFormation 主控台](#)。
2. 在畫面頂端的導覽列上，選擇您要從 AWS 區域 中管理 StackSet 的。

您可以選擇支援 StackSets 的任何區域。您選取的區域不會影響您可使用 StackSet 部署到哪些區域。

3. 從導覽窗格選擇 StackSets。
4. 從 StackSets 頁面上方選擇建立 StackSet。
5. 在許可下，選擇自助服務許可，然後選擇您在先決條件中建立的 IAM 角色。
 - 對於 IAM 管理員角色，選擇 AWSCloudFormationStackSetAdministrationRole。
 - 對於 IAM 執行角色名稱，選擇 AWSCloudFormationStackSetExecutionRole。
6. 選擇 Prerequisite - Prepare template (事前準備 - 準備範本) 下的 Use a sample template (使用範例範本)。
7. 在選擇範例範本下，選擇啟用 AWS Config 範本。然後選擇下一步。

此範本會建立必要的資源，以在您的帳戶 AWS Config 中啟用，包括組態記錄器和交付管道。

8. 在指定堆疊詳細資訊頁面上，對於 StackSet 名稱，輸入 **my-awsconfig-stackset**。
9. 對於 StackSet 說明，輸入 **A StackSet that enables Config across multiple Regions**。
10. 在參數下，AWS Config 設定設定如下：

- a. 對於支援所有資源類型，請保留預設值 `true`，以記錄所有支援的資源類型。
- b. 對於包含全域資源類型，請保留預設值 `false`，以排除全域資源，如 IAM 角色。
- c. 將若未支援所有資源類型，請提供資源類型清單，保留設定為 `<All>`。
- d. 對於包含 Config 服務連結角色資源的區域，請以 `us-west-2` 取代 `<DeployToAnyRegion>`。

這表示只有在堆疊部署到美國西部 (奧勒岡) 區域時，才會建立名為 `AWSServiceRoleForConfig` 的服務連結角色。您將在此程序稍後選擇部署區域。

- e. 對於組態記錄器記錄頻率，選擇 `DAILY` 記錄。

11. 選擇 `Next` (下一步) 繼續。

12. 在設定 `StackSet` 選項頁面中，選擇新增標籤並透過指定鍵和值對來新增標籤。

- a. 在 `Key` (索引鍵) 欄位，輸入 `Stage`。
- b. 針對數值，輸入 `Test`。

您套用至 `StackSets` 的標籤將套用至您的堆疊建立的資源。

13. 對於執行組態，選擇作用中，以啟用 CloudFormation 的最佳化操作處理：

- 非衝突操作會同時執行，以便加快部署時間。
- 衝突操作會根據請求的順序，自動排入佇列和處理。

在有執行中或排入佇列的操作時，CloudFormation 會將所有傳入操作排入佇列，即使這些操作沒有衝突也一樣。在此期間，您無法變更執行設定。

14. 選擇下一步。

15. 在設定部署選項頁面上，對於將堆疊新增至 `StackSet`，選擇部署新堆疊。

16. 針對 `Accounts` (帳戶)，選擇 `Deploy stacks in accounts` (在帳戶中部署堆疊)。

17. 在文字方塊中，輸入您的 AWS 帳戶 ID。

18. 對於指定區域，依此順序選取下列區域：

1. 美國西部 (奧勒岡) 區域 (`us-west-2`)
2. 美國東部 (維吉尼亞北部) 區域 (`us-east-1`)

如有需要，使用美國西部 (奧勒岡) 區域旁邊的向上箭頭，將它上移成清單中的第一個項目。區域順序決定其部署順序。

19. 對於部署選項，設定下列設定：

- a. 對於 Maximum concurrent accounts (同時使用的帳戶上限)，保留 Number (數值) 和 1 的預設值。

對於多帳戶部署，該設定表示 CloudFormation 一次只能在一個帳戶中部署堆疊。

- b. 針對 Failure tolerance (容錯能力)，保留預設值 Number (數字) 和 0。

這表示在您指定的其中一個區域中，堆疊部署的最大失敗次數為零，否則 CloudFormation 會停止目前區域中的部署，並取消在剩餘區域中的部署。

- c. 針對區域並行，選擇循序 (預設)。

此設定可確保 CloudFormation 在移至下一個區域之前，先完成某個區域的部署。

- d. 對於並行模式，保留嚴格容錯能力的預設值。

對於多帳戶部署，這能減少發生故障時的帳戶並行層級，保持在容錯能力 +1 內。

20. 選擇下一步。

21. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。

22. 當您準備好建立 StackSet 時，請選擇提交。

監控 StackSet 建立

選擇提交後，CloudFormation 會開始建立 StackSet，並將堆疊部署到您帳戶中的指定區域。StackSet 詳細資訊頁面會自動開啟，您可在其中監控操作的進度。

監控 StackSet 建立

1. 在 StackSet 詳細資訊頁面上，預設顯示操作索引標籤，顯示目前進行中的操作。
2. 操作狀態一開始應是 RUNNING。CloudFormation 會根據您設定的部署選項，在指定的區域中建立堆疊。
3. 要查看操作的詳細資訊，請在清單中選取操作 ID。
4. 在操作詳細資訊頁面上，您可檢視在每個區域中建立的堆疊執行個體狀態。
5. 等待操作狀態變更為 SUCCEEDED，這表示已成功建立 StackSet 及其所有堆疊執行個體。

檢視 StackSet 結果

StackSet 建立完成後，您可以檢視已部署的堆疊執行個體，並確認 AWS Config 已在您的帳戶中跨指定區域啟用。

檢視 StackSet 結果

1. 在 StackSet 詳細資訊頁面上，選擇堆疊執行個體索引標籤。
2. 您應會看到在帳戶中跨指定區域建立的堆疊執行個體清單。每個堆疊執行個體的状态應為 SUCCEEDED，表示已成功部署。
3. 若要驗證您的帳戶是否已啟用 AWS Config，您可以檢查每個部署區域中的 AWS Config 主控台。

更新您的 StackSet

建立您的 StackSet 之後，您可能想要更新它以修改參數值或新增更多區域。本節說明如何更新 AWS Config 記錄頻率參數。

更新您的 StackSet

1. 在 StackSets 頁面上，選取 **my-awsconfig-stackset**。
2. 選擇 StackSet 後，從動作選單選擇編輯 StackSet 詳細資訊。
3. 在選擇範本頁面上，針對先決條件 - 準備範本，選擇使用目前範本。
4. 選擇下一步。
5. 在指定 StackSet 詳細資訊頁面的參數下，尋找組態記錄器記錄頻率，並將其從 DAILY 變更為 CONTINUOUS。
6. 選擇下一步。
7. 在設定 StackSet 選項頁面上，將設定保留原樣，再選擇下一步。
8. 在設定部署選項頁面上，指定帳戶 ID 和您在建立 StackSet 時所使用的相同區域。
9. 對於部署選項，保留與之前相同的設定。
10. 選擇下一步。
11. 在檢閱頁面上，檢閱您的選項，然後選擇提交。
12. CloudFormation 開始更新您的 StackSet。您可在 StackSet 詳細資訊頁面的操作索引標籤上監控進度。

將堆疊新增至 StackSet

您可部署到其他區域，將更多堆疊新增至 StackSet。本節顯示如何將堆疊新增至新區域。

將堆疊新增至 StackSet

1. 在 StackSets 頁面上，選取 **my-awsconfig-stackset**。
2. 選擇 StackSet 後，從動作選單選擇將堆疊新增至 StackSet。
3. 在設定部署選項頁面上，對於將堆疊新增至 StackSet，選擇部署新堆疊。
4. 針對帳戶，選擇在帳戶中部署堆疊並輸入帳戶 ID。
5. 針對指定區域，選取新區域，例如歐洲 (愛爾蘭) (eu-west-1)。
6. 對於部署選項，保留與之前相同的設定。
7. 選擇下一步。
8. 在指定覆寫頁面上，依指定保留屬性值並選擇下一步。
9. 在 Review (檢閱) 頁面上，檢閱您的選項，然後選擇 Submit (提交)。
10. CloudFormation 會在指定的區域開始建立新的堆疊。您可在 StackSet 詳細資訊頁面的操作索引標籤上監控進度。

清除

為了避免產生不必要的 AWS Config 資源費用，您應該從 StackSet 刪除堆疊、刪除 StackSet 本身，以及移除您在本教學課程中建立的 IAM 角色來清除。因所有資源都部署在您的帳戶中，因此清除非常簡單。

從您的 StackSet 刪除堆疊

1. 在 StackSets 頁面上，選取 **my-awsconfig-stackset**。
2. 選取 StackSet 後，從動作選單選擇從 StackSet 刪除堆疊。
3. 在設定部署選項頁面上，針對帳戶，選擇在帳戶中部署堆疊，然後輸入您的帳戶 ID。
4. 針對指定區域，選取您部署堆疊的所有區域。
5. 對於部署選項，保留預設的設定。
6. 確定未開啟保留堆疊，以便刪除堆疊及其資源。
7. 選擇下一步。

8. 在 Review (檢閱) 頁面上，檢閱您的選項，然後選擇 Submit (提交)。
9. CloudFormation 會開始從 StackSet 刪除堆疊。您可在 StackSet 詳細資訊頁面的操作索引標籤上監控進度。

刪除您的 StackSet

1. 刪除所有堆疊之後，在 StackSets 頁面上選取 **my-awsconfig-stackset**。
2. 選擇 StackSet 後，從動作選單選擇刪除 StackSet。
3. 出現確認提示時，請選擇刪除。

刪除 IAM 服務角色

由於您只部署到您的帳戶，因此只需要從此單一帳戶刪除 IAM 角色，使得清除比多帳戶部署更為簡單。

1. 開啟 [IAM 主控台](#)。
2. 從導覽窗格中，選擇 Roles (角色)。
3. 在搜尋方塊中，輸入 **AWSCloudFormationStackSet**，以尋找您為此教學課程建立的角色。
4. 選取 AWSCloudFormationStackSetAdministrationRole 旁邊的核取方塊。
5. 從頁面頂部，選擇刪除。
6. 在確認對話方塊中輸入 **delete**，然後選擇刪除。
7. 重複相同的程序，刪除 AWSCloudFormationStackSetExecutionRole。

刪除 StackSet 之後，AWS 區域 由於AWS::S3::Bucket資源上的 DeletionPolicy 屬性，Amazon S3 儲存貯體會保留在每個儲存貯體中。這會保留您的 AWS Config 歷史記錄資料。若您不再需要此資料，可以手動安全地刪除儲存貯體。您必須先清空儲存貯體，才能將其刪除。清空儲存貯體將會刪除其中的所有物件。

清空和刪除 Amazon S3 儲存貯體

1. 開啟 [Amazon S3 主控台](#)。
2. 在主控台左側的導覽窗格中，選擇 Buckets (儲存貯體)。
3. 在儲存貯體清單中，您會在部署的每個區域看到為此 StackSet 建立的儲存貯體。選取為此 StackSet 建立之儲存貯體名稱旁邊的選項，然後選擇空白。

4. 在清空儲存貯體頁面上的文字欄位中輸入 **permanently delete**，以確認您要清空儲存貯體，然後選擇清空。
5. 在 Empty bucket: Status (清空儲存貯體：狀態) 頁面上監控儲存貯體清空的進度。
6. 若要返回儲存貯體清單，請選擇 Exit (結束)。
7. 選取儲存貯體名稱旁邊的選項，然後選擇刪除。
8. 出現確認提示時，請輸入儲存貯體名稱，然後選擇刪除儲存貯體。
9. 從儲存貯體清單中，監控儲存貯體刪除程序的進度。當 Amazon S3 完成刪除儲存貯體時，它會從清單移除儲存貯體。
10. 針對 StackSet 在不同區域中建立的每個儲存貯體，重複此程序。

後續步驟

恭喜您！您已成功使用範例範本建立 StackSet，將堆疊部署到帳戶中的多個區域、更新 StackSet、新增更多堆疊，以及清理資源。透過專注於單一帳戶部署，您已簡化清除程序，同時仍學習 StackSets 的核心多區域功能。

若要進一步了解 StackSets，請探索下列主題：

- [覆寫 CloudFormation StackSet 中堆疊的參數值](#) – 了解如何覆寫特定帳戶及區域的參數值。
- [具有服務受管許可的 Create CloudFormation StackSets](#) – 探索使用 AWS Organizations 為多帳戶部署建立 StackSets。

具有自我管理許可的 Create CloudFormation StackSets

本主題說明如何建立具有自我管理許可的 StackSets，以跨 AWS 帳戶和區域部署堆疊。

Note

繼續之前，建立 StackSets 所需 IAM 服務角色，才能在您用來管理 StackSet 的帳戶與堆疊部署目的地帳戶之間建立信任關係。如需詳細資訊，請參閱[授與自我管理許可](#)。

主題

- [建立具有自我管理許可的 StackSet \(主控台\)](#)

- [建立具有自我管理許可的 StackSet \(AWS CLI\)](#)

建立具有自我管理許可的 StackSet (主控台)

建立 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇您要從中管理 StackSet 的 AWS 區域。
3. 從導覽窗格選擇 StackSets。
4. 從 StackSets 頁面上方選擇建立 StackSet。
5. 在許可下，選擇自助服務許可，然後選擇您建立的 IAM 角色。
6. 在準備範本下，請選擇範本已準備就緒。
7. 在 Specify template (指定範本) 下，選擇指定堆疊範本所在 S3 儲存貯體的 URL，或上傳堆疊範本檔案。然後選擇下一步。
8. 在指定 StackSet 詳細資訊頁面上，提供 StackSet 的名稱、指定任何參數，然後選擇下一步。
9. 選擇 Next (下一步) 繼續。
10. 在 Configure StackSet options (設定 StackSet 選項) 頁面的 Tags (標籤) 下，指定任何標籤以套用至堆疊中的資源。如需如何在 中使用標籤的詳細資訊 AWS，請參閱AWS 帳單與成本管理 《使用者指南》中的[使用成本分配標籤組織和追蹤 AWS 成本](#)。
11. 對於執行組態，選擇作用中，以啟用 CloudFormation 的最佳化操作處理：
 - 非衝突操作會同時執行，以便加快部署時間。
 - 衝突操作會根據請求的順序，自動排入佇列和處理。

在有執行中或排入佇列的操作時，CloudFormation 會將所有傳入操作排入佇列，即使這些操作沒有衝突也一樣。在此期間，您無法變更執行設定。
12. 如果範本包含 IAM 資源，在功能中選擇我知道此範本可建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。
13. 選擇下一步。
14. 在設定部署選項頁面上，對於將堆疊新增至 StackSet，選擇部署新堆疊。
15. 針對 Accounts (帳戶)，選擇 Deploy stacks in accounts (在帳戶中部署堆疊)。在文字方塊中貼上您的目標 AWS 帳戶 號碼，以逗號分隔多個號碼。

 Note

若想要在該帳戶中部署堆疊，也可包含您的管理員帳戶 ID。

16. 在指定區域下，選擇要部署堆疊的區域。
17. 對於部署選項，執行下列動作：
 - 針對並行帳戶數目上限，指定並行處理的帳戶數量。
 - 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
 - 對於區域並行，選擇如何處理區域：循序 (一次處理一個區域) 或平行 (同時處理多個區域)。
 - 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級 (最大並行帳戶的值)。
18. 選擇下一步。
19. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。
20. 當您準備好建立 StackSet 時，請選擇提交。

CloudFormation 開始建立 StackSet。在您選擇提交時開啟的“StackSet 詳細資訊”頁面中，查看 StackSet 中的堆疊建立進度和狀態。

建立具有自我管理許可的 StackSet (AWS CLI)

請依照本節中的步驟使用 AWS CLI 來：

- 建立 StackSet 容器。
- 部署堆疊執行個體。

建立 StackSet

1. 使用 [create-stack-set](#) 命令來建立新的 StackSet，並命名為 *my-stackset*。下列範例使用儲存於 S3 儲存貯體中的範本，並包含一個參數，該參數會將 *KeyPairName* 設定為數值 *TestKey*。

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url s3://aws-logs-us-east-1-123456789012-us-east-1/stack-sets/stack-set-template.yaml \  
  --parameters KeyPairName=TestKey
```



```
--template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
--parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. 在您的 `create-stack-set` 命令完成後，請執行 [list-stack-sets](#) 命令以查看已建立的 StackSet。您應該會在結果中看到新 StackSet。

```
aws cloudformation list-stack-sets
```

3. 使用 [create-stack-instances](#) 命令在 StackSet 中部署堆疊。下列範例在兩個 AWS 帳戶 (*account_ID_1* 和 *account_ID_2*) 中，跨兩個區域 (*us-west-2* 和 *us-east-1*) 部署堆疊。

使用 `--operation-preferences` 選項，設定並行帳戶處理和其他部署偏好設定。此範例使用基於數量的設定。請注意，`MaxConcurrentCount` 不得超過 `FailureToleranceCount + 1`。對於以百分比為基礎的設定，請使用 `FailureTolerancePercentage` 或 `MaxConcurrentPercentage`。

```
aws cloudformation create-stack-instances \  
  --stack-set-name my-stackset \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateStackInstances](#)。

4. 使用 [describe-stack-set-operation](#) 命令來確認您的堆疊已成功建立。對於 `--operation-id` 選項，指定作為 `create-stack-instances` 輸出一部分傳回的操作 ID。

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

具有服務受管許可的 Create CloudFormation StackSets

透過服務受管許可，您可以將堆疊部署到 AWS Organizations 特定區域中由管理的帳戶。使用此模型，您不需要在每個目標帳戶和 AWS 區域中建立 IAM 角色。CloudFormation 會代表您建立 IAM 角色。如需詳細資訊，請參閱[啟用受信任存取](#)。

主題

- [考量事項](#)
- [建立具有服務受管許可的 StackSet \(主控台\)](#)
- [建立具有服務受管許可的 StackSet \(AWS CLI\)](#)

考量事項

建立具有服務受管許可的 StackSet 之前，請考慮下列事項：

- 具有服務受管許可的 StackSets 可由組織的管理帳戶或委派的管理員帳戶啟動，但所有操作均由管理帳戶執行。
- CloudFormation 不會將堆疊部署到管理帳戶，即使該帳戶是您的組織的一部分或屬於組織單位。
- StackSet 可以將整個組織 (包括所有帳戶) 或指定的 OU 設為目標。當 StackSet 以父 OU 為目標時，會自動包含任何子 OU。根據預設，當 StackSet 以特定 OU 為目標時，它會包含這些 OU 中的所有帳戶。不過，您可使用帳戶篩選選項將特定帳戶設為目標。
- 多個 StackSets 可以將相同的組織或 OU 設為目標。
- 您無法以組織外部的帳戶為目標。
- 您部署 StackSet 的權限取決於您用來登入管理帳戶之 IAM 主體實體 (使用者、角色或群組) 的許可。關於授與許可部署至組織的 IAM 政策範例，請參閱[根據區域和資源類型限制堆疊集操作](#)。
- 委派管理員具有部署至組織中任何帳戶的完整許可。管理帳戶無法限制部署至特定 OU 或 StackSet 操作的委派管理員許可。
- 自動部署設定是在 StackSet 層級套用。您無法特別針對 OU、帳戶或區域來調整自動部署。
- 使用服務受管許可的 StackSets 不支援包含了巨集或轉換的巢狀堆疊或範本。

建立具有服務受管許可的 StackSet (主控台)

建立 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇您要從中管理 StackSet 的 AWS 區域。
3. 從導覽窗格選擇 StackSets。
4. 從 StackSets 頁面上方選擇建立 StackSet。
5. 在 Permissions (許可) 下，選擇 Service-managed permissions (服務管理許可)。

 Note

如果 AWS Organizations 停用的受信任存取，則會顯示橫幅。需要信任存取，才能建立或更新具有服務受管許可的 StackSet。只有組織管理帳戶中的管理員，才具有 [使用 啟用 StackSets 的受信任存取 AWS Organizations](#) 的許可。

6. 在準備範本下，請選擇範本已準備就緒。
 7. 在 Specify template (指定範本) 下，選擇指定堆疊範本所在 S3 儲存貯體的 URL，或上傳堆疊範本檔案。然後選擇下一步。
 8. 在指定 StackSet 詳細資訊頁面上，提供 StackSet 的名稱、指定任何參數，然後選擇下一步。
 9. 在 Configure StackSet options (設定 StackSet 選項) 頁面的 Tags (標籤) 下，指定任何標籤以套用至堆疊中的資源。如需如何在 中使用標籤的詳細資訊 AWS，請參閱AWS 帳單與成本管理 《使用者指南》中的[使用成本分配標籤組織和追蹤 AWS 成本](#)。
 10. 對於執行組態，選擇作用中，以啟用 CloudFormation 的最佳化操作處理：
 - 非衝突操作會同時執行，以便加快部署時間。
 - 衝突操作會根據請求的順序，自動排入佇列和處理。
- 在有執行中或排入佇列的操作時，CloudFormation 會將所有傳入操作排入佇列，即使這些操作沒有衝突也一樣。在此期間，您無法變更執行設定。
11. 如果範本包含 IAM 資源，在功能中選擇我知道此範本可建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。
 12. 選擇下一步以繼續，並啟用受信任存取 (如果尚未啟用)。
 13. 在設定部署選項頁面，部署目標下，執行下列其中一項操作：
 - 選擇部署至組織，以部署至組織中的所有帳戶。
 - 選擇部署至組織單位，以部署至特定 OU 中的所有帳戶。選擇 Add an OU (新增 OU)，然後將目標 OU ID 貼到文字方塊中。針對每個新的目標 OU 重複上述步驟。

若您選擇部署到組織單位，對於帳戶篩選類型，您可以選擇下列其中一個選項並提供帳戶號碼，將部署目標設定為特定個別帳戶。

- 無 (預設) – 將堆疊部署到指定 OU 中的所有帳戶。
- 交集 – 將堆疊部署到所選 OU 中的特定個別帳戶。

- 差集 – 將堆疊部署到所選 OU 中的所有帳戶，但特定帳戶除外。
 - 聯集 – 將堆疊部署到指定的 OU 和其他個別帳戶。
14. 在自動部署下，選擇是否自動部署到未來新增至目標組織或 OU 的帳戶。如需詳細資訊，請參閱 [在中啟用或停用 StackSets 的自動部署 AWS Organizations](#)。
 15. 如果您已啟用自動部署，請在 Account removal behavior (帳戶移除行為) 下，選擇在從目標組織或 OU 移除帳戶時，是否保留或刪除堆疊資源。

 Note

如果選取保留堆疊，則會從 StackSet 移除堆疊執行個體，但會保留堆疊及其相關資源。資源會保持在當前狀態，但不再是 StackSet 的一部分。

16. 在指定區域下，選擇要部署堆疊的區域。
17. 對於部署選項，執行下列動作：
 - 針對並行帳戶數目上限，指定並行處理的帳戶數量。
 - 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
 - 對於區域並行，選擇如何處理區域：循序 (一次處理一個區域) 或平行 (同時處理多個區域)。
 - 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級 (最大並行帳戶的值)。
 - 對於 StackSet 相依性，新增相依的 StackSet ARNs，最多保持在 10 個相依性內。如需詳細資訊，請參閱 [在中啟用或停用 StackSets 的自動部署 AWS Organizations](#)。
18. 選擇 Next (下一步) 繼續。
19. 在檢閱頁面上，確認 StackSet 將部署到正確區域中的正確帳戶，然後選擇建立 StackSet。

StackSet details (StackSet 詳細資訊) 頁面隨即開啟。您可以檢視 StackSet 的堆疊建立進度和狀態。

建立具有服務受管許可的 StackSet (AWS CLI)

請依照本節中的步驟使用 AWS CLI 來：

- 建立 StackSet 容器。

- 部署堆疊執行個體。

Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

Deploy to your organization

建立 StackSet

1. 使用 [create-stack-set](#) 命令來建立新的 StackSet，並命名為 *my-stackset*。下列範例使用儲存在 S3 儲存貯體中的範本，啟用自動部署，並在移除帳戶時保留堆疊。如需詳細資訊，請參閱 [在中啟用或停用 StackSets 的自動部署 AWS Organizations](#)。

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment  
Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

2. 使用 [list-stack-sets](#) 命令確認您的 StackSet 已建立。結果中會列出新的 StackSet。

```
aws cloudformation list-stack-sets
```

- 如果您在登入會員帳戶時，將 `--call-as` 選項設定為 `DELEGATED_ADMIN`，則 `list-stack-sets` 會傳回組織管理帳戶中具有服務管理許可的所有 StackSets。
 - 如果您在登入 AWS 帳戶時，將 `--call-as` 選項設定為 `SELF`，則 `list-stack-sets` 會傳回 AWS 帳戶中的所有自我管理 StackSets。
 - 如果您在登入組織的管理帳戶時，將 `--call-as` 選項設定為 `SELF`，則 `list-stack-sets` 會傳回組織管理帳戶中的所有 StackSets。
3. 使用 [create-stack-instances](#) 命令在 StackSet 中新增堆疊。針對 `--deployment-targets` 選項，指定要部署到組織中所有帳戶的組織根 ID。

使用 `--operation-preferences` 選項，設定並行帳戶處理和其他部署偏好設定。此範例使用基於數量的設定。請注意，`MaxConcurrentCount` 不得超過 `FailureToleranceCount`

+ 1。對於以百分比為基礎的設定，請使用 `FailureTolerancePercentage` 或 `MaxConcurrentPercentage`。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=r-a1b2c3d4e5 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateStackInstances](#)。

4. 使用 `create-stack-instances` 輸出中傳回的 `operation-id`，使用下列 [describe-stack-set-operation](#) 命令，以驗證是否已成功建立堆疊。

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to organizational units (OUs)

建立 StackSet

1. 使用 [create-stack-set](#) 命令來建立新的 StackSet，並命名為 *my-stackset*。下列範例使用儲存於 S3 儲存貯體中的範本，並包含一個參數，該參數會將 *KeyPairName* 設定為數值 *TestKey*

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stackset \  
  --template-url https://s3.region-code.amazonaws.com/amzn-s3-demo-bucket/MyApp.template \  
  --permission-model SERVICE_MANAGED \  
  --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
```

2. 使用 [list-stack-sets](#) 命令確認您的 StackSet 已建立。結果中會列出新的 StackSet。

```
aws cloudformation list-stack-sets
```

- 如果您在登入會員帳戶時，將 `--call-as` 選項設定為 `DELEGATED_ADMIN`，則 `list-stack-sets` 會傳回組織管理帳戶中具有服務管理許可的所有 StackSets。

- 如果您在登入 AWS 帳戶時，將 `--call-as` 選項設定為 `SELF`，則 `list-stack-sets` 會傳回 AWS 帳戶中的所有自我管理 StackSets。
 - 如果您在登入組織的管理帳戶時，將 `--call-as` 選項設定為 `SELF`，則 `list-stack-sets` 會傳回組織 管理帳戶 中的所有 StackSets。
3. 使用 [create-stack-instances](#) 命令在 StackSet 中新增堆疊。針對 `--deployment-targets` 選項，指定要部署的 OU ID。

使用 `--operation-preferences` 選項，設定並行帳戶處理和其他部署偏好設定。此範例使用基於數量的設定。請注意，`MaxConcurrentCount` 不得超過 `FailureToleranceCount` + 1。對於以百分比為基礎的設定，請使用 `FailureTolerancePercentage` 或 `MaxConcurrentPercentage`。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateStackInstances](#)。

4. 使用 `create-stack-instances` 輸出中傳回的 `operation-id`，使用下列 [describe-stack-set-operation](#) 命令，以驗證是否已成功建立堆疊。

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name my-stackset \  
  --operation-id operation_ID
```

Deploy to specific accounts in OUs

您可以指定特定組織單位並使用帳戶篩選，精確控制哪些帳戶會接收堆疊部署。根據預設，如果未指定帳戶篩選，堆疊會部署到指定 OU 中的所有帳戶。

在 AWS CLI 中，您可以使用 `--deployment-targets` 選項指定帳戶篩選。關於詳細資訊，請參閱 [DeploymentTargets](#)。

使用 `create-stack-set` 命令建立 StackSet 容器之後，請使用下列其中一個範例，將堆疊部署至特定帳戶。

以 OU 中的特定帳戶為目標

以下範例只會將堆疊部署到 OU1 中的帳戶 A1 和 A2。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets  
  OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=INTERSECTION \  
  --regions us-west-2 us-east-1
```

從 OU 排除帳戶

下列範例會將堆疊部署到 OU1 中的所有帳戶，帳戶 A1 和 A2 除外。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
  --deployment-targets  
  OrganizationalUnitIds=OU1,Accounts=A1,A2,AccountFilterType=DIFFERENCE \  
  --regions us-west-2 us-east-1
```

在中啟用或停用 StackSets 的自動部署 AWS Organizations

當新帳戶新增至您的目標組織或組織單位 (OUs) 時，CloudFormation 可以自動將其他堆疊部署到新 AWS Organizations 帳戶。您可以啟用自動部署，並選擇當帳戶從目標 OU 中移除時，是刪除還是保留堆疊及其相關聯的資源。對於使用服務受管許可的 StackSets，這些設定隨時可以修改。

自動部署的運作方式

如果您啟用自動部署，當帳戶新增至目標組織或 OU、從目標組織或 OU 移除，或在目標 OU 之間移動時，將會觸發自動部署。

例如，假設 StackSet1 以 us-east-1 區域的 OU1 為目標，StackSet2 以 us-east-1 區域的 OU2 為目標。OU1 包含 AccountA。

在已啟用自動部署的情況下，如果您將 AccountA 從 OU1 移至 OU2，CloudFormation 會自動執行刪除操作以從 AccountA 中移除 StackSet1 堆疊，並將新增 StackSet2 堆疊至 AccountA 的建立操作排入佇列。

考量事項

以下是使用自動部署時的考量事項：

- 自動部署功能在 StackSet 層級啟用。您無法特別針對 OU、帳戶或區域來調整自動部署。

- 覆寫的參數值僅套用至目標 OU 及其子 OU 中目前的帳戶。未來新增至目標 OU 及其子 OU 的帳戶將使用 StackSet 預設值，而非覆寫的值。
- 自動部署不會考慮帳戶層級鎖定篩選條件。如果您以特定帳戶為目標並啟用自動部署，您的 StackSet 將繼續部署到已部署組織內新增的任何帳戶。若要避免向新增帳戶部署，請停用自動部署。
- 相依性管理：為每個 StackSet 定義最多 10 個相依性，每個帳戶最多 100 個相依性。例如，如果您有五個 StackSets，每個都有五個相依性，則有 25 個相依性計入 100 個相依性限制。您可以透過[服務配額主控台](#)請求提高限制。刪除 StackSets 或停用 Organizations 時，會移除相依性。
- 建議在使用自動部署時啟用受管執行。受管執行將允許多個目標帳戶中的自動部署操作在 StackSet 內同時執行，從而提高處理速度，尤其是針對較大的 Organizations。

啟用或停用自動部署 (主控台)

啟用或停用自動部署

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 從導覽窗格選擇 StackSets。
4. 在 StackSets 頁面上，選取要更新的 StackSet 名稱旁的選項。
5. 在右上角的動作功能表中，選擇編輯自動部署。
6. 在開啟的對話方塊中，執行下列動作：
 - a. 針對自動部署，選擇啟用或停用。
 - b. 對於帳戶移除行為，選擇刪除堆疊或保留堆疊。保留的資源會保持在當前狀態，但不再是 StackSet 的一部分。
 - c. 對於 StackSet 相依性，新增相依的 StackSet ARNs，最多保持在 10 個相依性內。
7. 選擇儲存。

啟用或停用自動部署 (AWS CLI)

啟用或停用自動部署

1. 使用 [update-stack-set](#) 命令搭配 `--auto-deployment` 選項。

下列命令會啟用自動部署。

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment  
  Enabled=true,RetainStacksOnAccountRemoval=true,DependsOn=ARN1,ARN2
```

或者，要停用自動部署，請將 Enabled=false 指定為 --auto-deployment 選項的值，如下列範例所示。

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template --auto-deployment Enabled=false
```

2. 使用 update-stack-set 輸出中傳回的操作 ID，執行 [describe-stack-set-operation](#)，以驗證是否已成功更新 StackSet。

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

Update CloudFormation StackSets

您可以使用 CloudFormation 主控台或 AWS CLI 來更新 StackSet。

要從 StackSet 新增和移除帳戶和區域，請參閱 [將堆疊新增至 StackSets](#) 和 [從 StackSets 刪除堆疊](#)。若要覆寫堆疊的參數值，請參閱 [覆寫堆疊上的參數](#)。

主題

- [更新您的 StackSet \(主控台\)](#)
- [更新您的 StackSet \(AWS CLI\)](#)

更新您的 StackSet (主控台)

更新 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 從導覽窗格選擇 StackSets。

4. 在 StackSets 頁面上，選取要更新的 StackSet。
5. 選擇 StackSet 後，從動作選單選擇編輯 StackSet 詳細資訊。
6. 在選擇範本頁面上，視需要更新許可區段，或跳至下個步驟。
7. 針對先決條件 - 準備範本，選擇使用目前範本，以使用目前範本，或取代目前範本，以指定 S3 URL 至另一個範本或上傳新範本。
8. 選擇下一步。
9. 在指定 StackSet 詳細資訊頁面上，針對 StackSet 描述，根據需要更新 StackSet 的描述。
10. 對於參數，根據需要更新參數值。
11. 選擇下一步。
12. 在設定 StackSet 選項頁面上，針對標籤，根據需要修改標籤。您可以新增、更新或刪除標籤。如需如何在 中使用標籤的詳細資訊 AWS，請參閱AWS 帳單與成本管理 《使用者指南》中的[使用成本分配標籤組織和追蹤 AWS 成本](#)。
13. 對於執行組態，您可根據需要更新執行組態。

 Note

請注意，當操作正在執行或處於佇列中時，您無法變更執行設定。

14. 如果範本包含 IAM 資源，在功能中選擇我知道此範本可建立 IAM 資源，以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。
15. 選擇下一步。
16. 在設定部署選項頁面上，提供要更新的帳戶和區域。

CloudFormation 會在第一個區域內的指定帳戶中部署堆疊更新，然後移至下一個區域，以此類推。只要一個區域的部署失敗不超過指定的容錯能力。

- a. [自我管理許可] 對於帳戶、部署位置，選擇在帳戶中部署堆疊。在文字方塊中貼上您用來建立 StackSet 的目標帳戶 ID，並以逗號分隔多個 ID。

[服務受管許可] 執行下列其中一項操作：

- 選擇 Deploy to organizational units (OUs) (部署至組織單位 (OU))。輸入您用來建立 StackSet 的目標 OU。
- 選擇 Deploy to accounts (部署至帳戶)。貼上您用來建立 StackSet 的目標 OU ID 或帳戶 ID。

- b. 針對指定區域，指定希望 CloudFormation 部署更新的順序。

c. 對於部署選項，執行下列動作：

- 針對並行帳戶數目上限，指定並行處理的帳戶數量。
- 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
- 對於區域並行，選擇如何處理區域：循序（一次處理一個區域）或平行（同時處理多個區域）。
- 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級（最大並行帳戶的值）。
- 【服務受管許可】對於 StackSet 相依性，新增相依的 StackSet ARNs，最多保持在 10 個相依性內。如需詳細資訊，請參閱[在中啟用或停用 StackSets 的自動部署 AWS Organizations](#)。

d. 選擇 Next (下一步) 繼續。

17. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。

18. 準備好繼續時，請選擇提交。

CloudFormation 開始將更新套用到 StackSet，並顯示 StackSet 詳細資訊頁面的操作索引標籤。您可以在 Operations (操作) 標籤中檢視更新操作的進度和狀態。

更新您的 StackSet (AWS CLI)

Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

1. 更新 StackSet

使用 [update-stack-set](#) 命令以變更您的 StackSet。

在下列範例中，我們使用 `--parameters` 選項更新 StackSet。尤其是我們將交付管道組態的預設快照交付頻率，從 `TwentyFour_Hours` 變更為 `Twelve_Hours`。因為我們仍使用目前的範本，因此我們新增 `--use-previous-template` 選項。

使用 `--operation-preferences` 選項，設定並行帳戶處理和其他部署偏好設定。這些範例使用基於數量的設定。請注意，`MaxConcurrentCount` 不得超過 `FailureToleranceCount` + 1。對於以百分比為基礎的設定，請使用 `FailureTolerancePercentage` 或 `MaxConcurrentPercentage`。

[自我管理許可] 對於 `--accounts` 選項，提供更新的目標帳戶 ID。

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

[服務受管許可] 對於 `--deployment-targets` 選項，提供更新的目標組織根 ID 或組織單位 ID。

```
aws cloudformation update-stack-set --stack-set-name my-stackset \  
  --use-previous-template \  
  --parameters ParameterKey=MaximumExecutionFrequency,ParameterValue=Twelve_Hours \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo,ou-rcuk-slr5lh0a \  
  --regions us-west-2 us-east-1 \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

如需詳細資訊，請參閱《https://docs.aws.amazon.com/AWSCloudFormation/latest/APIReference/API_UpdateStackSet.html API 參考》中的 AWS CloudFormation `UpdateStackSet`。

- 藉由執行 `describe-stack-set-operation` 命令以顯示更新操作的狀態和結果，確認您的 `StackSet` 已成功更新。針對 `--operation-id`，使用您的 `update-stack-set` 命令所傳回的操作 ID。

```
aws cloudformation describe-stack-set-operation \  
  --operation-id operation_ID
```

將堆疊新增至 CloudFormation StackSets

當您建立 StackSet 時，可為該 StackSet 建立堆疊。CloudFormation 也可以讓您在 StackSet 建立之後的任何時間點，為其他帳戶和區域新增更多堆疊。您可以透過 CloudFormation 主控台或 AWS CLI 新增堆疊。

主題

- [將堆疊新增至 StackSet \(主控台\)](#)
- [將堆疊新增至 StackSet \(AWS CLI\)](#)

將堆疊新增至 StackSet (主控台)

將堆疊新增至 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 從導覽窗格選擇 StackSets。在 StackSets 頁面上，選取您建立的 StackSet。
4. 選擇 StackSet 後，從動作選單選擇將堆疊新增至 StackSet。
5. 在設定部署選項頁面上進行以下操作：
 - a. 在將堆疊新增至 StackSet，選擇部署新堆疊。
 - b. 接著，根據您 StackSet 的許可設定執行下列操作：
 - [自我管理許可] 對於帳戶、部署位置，選擇在帳戶中部署堆疊。在文字方塊中貼上您的目標帳戶號碼，並以逗號分隔多個號碼。
 - [服務受管許可] 針對部署目標，執行下列其中一項：
 - 選擇 Deploy to organization (部署至組織)，以部署至組織中的所有帳戶。
 - 選擇 Deploy to organizational units (OUs) (部署至組織單位 (OU))，以部署至特定 OU 中的所有帳戶。選擇 Add another OU (新增另一個 OU)，然後將目標 OU ID 貼到文字方塊中。針對每個新的目標 OU 重複上述步驟。CloudFormation 也會以所選目標的任何子 OU 為目標。

 Note

如果您新增的 OU 已是 StackSet 的目標，CloudFormation 會在 OU 中的任何帳戶中 (尚無 StackSet 的堆疊)，建立新的堆疊 (例如，在建立 StackSet 後新增至 OU 且已停用自動部署的帳戶)。

- c. 對於指定區域，指定要在先前步驟指定的目標帳戶中部署至哪些 AWS 區域。根據預設，CloudFormation 會在第一個區域內的指定帳戶中部署堆疊，然後移至下一個區域，以此類推。只要一個區域的部署失敗不超過指定的容錯能力。
 - d. 對於部署選項，執行下列動作：
 - 針對並行帳戶數目上限，指定並行處理的帳戶數量。
 - 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
 - 對於區域並行，選擇如何處理區域：循序 (一次處理一個區域) 或平行 (同時處理多個區域)。
 - 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級 (最大並行帳戶的值)。
 - e. 選擇下一步。
6. 在 Specify Overrides (指定覆寫) 頁面上，依指定保留屬性值。您不需要覆寫您要建立之堆疊的任何屬性值。選擇下一步。
 7. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。
 8. 準備好繼續時，請選擇提交。

CloudFormation 開始建立堆疊。在您選擇提交時開啟的“StackSet 詳細資訊”頁面中，查看 StackSet 中的堆疊建立進度和狀態。完成後，您的新堆疊應該會列在堆疊執行個體索引標籤中。

將堆疊新增至 StackSet (AWS CLI)

 Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

將堆疊新增至具有自我管理許可的 StackSet

使用 `create-stack-instances` CLI 命令。對於 `--accounts` 選項，提供您要建立堆疊的帳戶 ID。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--accounts account_id --regions eu-west-1 us-west-2
```

使用服務受管許可將堆疊新增至 StackSet

使用 `create-stack-instances` CLI 命令。針對 `--deployment-targets` 選項，提供您要為其建立堆疊的組織 (根) ID 或 OU ID。如需目標為特定帳戶的範例命令，請參閱 [建立具有服務受管許可的 StackSet \(AWS CLI\)](#)。

```
aws cloudformation create-stack-instances --stack-set-name my-stackset \  
--deployment-targets OrganizationalUnitIds=ou-rcuk-r1qi0wl7 --regions eu-west-1 us-west-2
```

Note

如果您新增的 OU 已是 StackSet 的目標，CloudFormation 會在 OU 中的任何帳戶中 (尚無 StackSet 的堆疊)，建立新的堆疊 (例如，在建立 StackSet 後新增至 OU 且已停用自動部署的帳戶)。

覆寫 CloudFormation StackSet 中堆疊的參數值

在某些情況下，您可能會希望特定區域或帳戶中的堆疊，擁有不同於 StackSet 本身指定的屬性值。例如，您可能想要根據帳戶用於開發或生產，為指定的參數指定不同的值。針對這些情況，CloudFormation 可讓您依據帳戶與區域覆寫堆疊中的參數值。您可以先在建立堆疊時覆寫範本參數值，也可以覆寫現有堆疊的參數值。您只能將之前在堆疊中覆寫的參數，設定回 StackSet 中指定的值。

參數值覆寫套用至您選取之帳戶和區域中的堆疊。在 StackSet 更新期間，針對堆疊覆寫的任何參數值皆不會更新，而是會保留其覆寫值。

您只能覆寫 StackSet 中指定的參數值，若要新增或刪除參數本身，您必須更新 StackSet 範本。如果您將參數新增至 StackSet 範本，那麼在您可以覆寫堆疊中的該參數值之前，您必須先以 StackSet 之中指定的新參數與值來更新所有堆疊。一旦所有堆疊以新參數完成更新，您就能視需要覆寫個別堆疊中的參數值。

若要了解如何在建立堆疊時覆寫 StackSet 參數值，請參閱 [將堆疊新增至 StackSets](#)。

主題

- [覆寫堆疊上的參數 \(主控台\)](#)
- [覆寫堆疊上的參數 \(AWS CLI\)](#)

覆寫堆疊上的參數 (主控台)

覆寫特定堆疊的參數

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 從導覽窗格選擇 StackSets。在 StackSets 頁面上，選取您的 StackSet。
4. 選擇 StackSet 後，從動作選單選擇覆寫 StackSet 參數。
5. 在設定部署選項頁面上，提供您要為其建立覆寫之堆疊的帳戶和區域。

根據預設，CloudFormation 會在第一個區域內的指定帳戶中部署堆疊，然後移至下一個區域，以此類推。前提是一個區域的部署失敗不超過指定的容錯能力。

- a. [自我管理許可] 對於部署目標，選擇在帳戶中部署堆疊。貼上用來建立 StackSet 的部分或全部目標帳戶 ID。

[服務受管許可] 執行下列其中一項操作：

- 選擇 Deploy to organizational units (OUs) (部署至組織單位 (OU))。輸入您在中用來建立 StackSet 的一或多個目標 OU。覆寫的參數值僅套用至目標 OU 及其子 OU 中目前的帳戶。未來新增至目標 OU 及其子 OU 的帳戶將使用 StackSet 預設值，而非覆寫的值。
 - 選擇 Deploy to accounts (部署至帳戶)。貼上您用來建立 StackSet 的部分或所有目標 OU ID 或帳戶 ID。
- b. 針對指定區域，新增一或多個已部署此 StackSet 之堆疊執行個體的區域。

如果您新增多個區域，則 Specify regions (指定區域) 下的區域順序決定其部署順序。

- c. 對於部署選項，執行下列動作：
 - 針對並行帳戶數目上限，指定並行處理的帳戶數量。

- 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
 - 對於區域並行，選擇如何處理區域：循序 (一次處理一個區域) 或平行 (同時處理多個區域)。
 - 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級 (最大並行帳戶的值)。
- d. 選擇下一步。
6. 在指定覆寫頁面上，勾選要覆寫的參數的核取方塊，然後從編輯覆寫值選單選擇覆寫 StackSet 值。
 7. 在覆寫 StackSet 參數值頁面上，進行變更，接著選擇儲存變更。

Note

若要將任何已覆寫的參數設定回使用 StackSet 中指定的值，請勾選所有參數並從編輯覆寫值選單選擇設定為 StackSet 值。這麼做會您在更新堆疊時，移除所有覆寫的值。

8. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。
9. 準備好繼續時，請選擇提交。

CloudFormation 開始更新您的堆疊。在選擇提交時開啟的 StackSet 詳細資訊頁面中，檢視堆疊的進度和狀態。

覆寫堆疊上的參數 (AWS CLI)

Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

覆寫特定堆疊的參數

1. 使用 [update-stack-instances](#) AWS CLI 命令並指定 `--parameter-overrides` 選項。

[自我管理許可] 對於 `--accounts` 選項，提供您要在堆疊上覆寫參數值的帳戶 ID。

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --accounts account_id --regions us-east-1
```

【服務受管許可】 針對 `--deployment-targets` 選項，提供您要覆寫堆疊參數的組織根 ID、OU IDs 或 AWS Organizations 帳戶 IDs。在此範例中，我們在 ID 為 *ou-rcuk-1x5j1lwo* ID 的 OU 中的所有帳戶中，覆寫堆疊的參數值。

覆寫的參數值僅套用至目標 OU 及其子 OU 中目前的帳戶。未來新增至目標 OU 及其子 OU 的帳戶將使用 StackSet 預設值，而非覆寫的值。

```
aws cloudformation update-stack-instances --stack-set-name my-stackset \  
  --parameter-overrides ParameterKey=Subnets,ParameterValue=subnet-1baa3351\  
  \,subnet-27b86940 \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1lwo \  
  --regions us-east-1
```

2. 執行 `describe-stack-set-operation` 命令來顯示更新操作的狀態和結果，以確認堆疊上已成功覆寫您的參數值。針對 `--operation-id`，使用您的 `update-stack-instances` 命令所傳回的操作 ID。

```
aws cloudformation describe-stack-set-operation --operation-id operation_ID
```

從 CloudFormation StackSets 刪除堆疊

您可以透過 CloudFormation 主控台或 AWS CLI，從 StackSets 刪除堆疊。

主題

- [從您的 StackSet 刪除堆疊 \(主控台\)](#)
- [從您的 StackSet 刪除堆疊 \(AWS CLI\)](#)

Note

從頂層組織單位刪除堆疊時，該 OU 將不再是 StackSet 的目標。

從您的 StackSet 刪除堆疊 (主控台)

刪除堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 從導覽窗格選擇 StackSets。在 StackSets 頁面上，選取 StackSet。
4. 選取 StackSet 後，從動作選單選擇從 StackSet 刪除堆疊。
5. 在設定部署選項頁面上，先選擇您要刪除堆疊的帳戶和區域。
 - a. [自我管理許可] 對於帳戶，選擇在帳戶中部署堆疊或在組織單位中部署堆疊。

若您選擇在帳戶中部署堆疊，將目標帳戶號碼貼到帳戶號碼文字方塊中，以逗號分隔多個號碼。

若選擇以組織單位部署堆疊，請在組織號碼文字方塊中貼上目標 OU ID，以鎖定屬於指定組織的所有帳戶。

- b. [服務受管許可] 對於組織單位，指定目標 OU ID。

Important

CloudFormation 將從指定的目標 OU 及其子 OU 中刪除堆疊。

對於帳戶篩選條件類型，您可選擇下列其中一個選項並提供帳戶號碼，進一步精選要刪除堆疊的帳戶。

- 無 (預設) – 從指定 OU 中的所有帳戶刪除堆疊。
- 交集 – 僅從所選 OU 中的特定個別帳戶刪除堆疊。
- 差集 – 從所選 OU 中的所有帳戶刪除堆疊，但特定帳戶除外。
- 聯集 – 從指定的 OU 以及其他個別帳戶刪除堆疊。

- c. 針對指定區域，選擇要刪除目標帳戶內堆疊的區域。
6. 對於部署選項，執行下列動作：
 - 針對並行帳戶數目上限，指定並行處理的帳戶數量。

- 針對容錯能力，指定每個區域允許的帳戶失敗數目上限。達到此限制後，操作將停止，且不會繼續前往其他區域。
 - 對於保留堆疊，啟用此選項可在從 StackSet 中移除堆疊及其相關資源時，保留堆疊及其相關資源。資源會保持在當前狀態，但不再是 StackSet 的一部分。
 - 對於區域並行，選擇如何處理區域：循序 (一次處理一個區域) 或平行 (同時處理多個區域)。
 - 針對並行模式，選擇並行在操作執行期間的行為方式。
 - 嚴格容錯能力 – 降低發生故障時的帳戶並行層級，保持在容錯能力 +1 的範圍內。
 - 軟性容錯能力 – 無論失敗為何，維持您指定的並行層級 (最大並行帳戶的值)。
7. 選擇下一步。
 8. 在 Review (檢閱) 頁面上，檢視您的選擇。選擇編輯以對相關區段進行變更。
 9. 準備好從 StackSet 移除堆疊時，請選擇提交。

刪除堆疊後，您可以在 StackSet 詳細資訊頁面的堆疊執行個體索引標籤中，驗證是否已刪除 StackSet 的堆疊執行個體。

從您的 StackSet 刪除堆疊 (AWS CLI)

Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

使用 `delete-stack-instances` 命令搭配您的 StackSet 名稱。

在這些範例中，我們使用 `--no-retain-stacks` 選項，因為我們未保留任何堆疊。如果您想要保留堆疊及其資源，請使用 `--retain-stacks` 而不是 `--no-retain-stacks`。

針對 `--regions`，指定 AWS 區域 您想要從中刪除堆疊的，例如 `us-west-2` 和 `us-east-1`。

使用 `--operation-preferences` 選項，設定並行帳戶處理和其他偏好設定。這些範例使用基於數量的設定。請注意，`MaxConcurrentCount` 不得超過 `FailureToleranceCount + 1`。對於以百分比為基礎的設定，請使用 `FailureTolerancePercentage` 或 `MaxConcurrentPercentage`。

刪除堆疊 (自我管理許可)

針對 `--accounts` 選項，指定要從中刪除堆疊的帳戶 ID。

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
  --accounts account_ID_1 account_ID_2 \  
  --regions us-west-2 us-east-1 \  
  --no-retain-stacks \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

刪除堆疊 (服務受管許可)

對於 `--deployment-targets`，請指定要從中刪除堆疊的組織根 ID 或組織單位 ID。

Important

CloudFormation 將從指定的目標 OU 及其子 OU 中刪除堆疊。

```
aws cloudformation delete-stack-instances --stack-set-name my-stackset \  
  --deployment-targets OrganizationalUnitIds=ou-rcuk-1x5j1wo,ou-rcuk-slr5lh0a \  
  --regions us-west-2 us-east-1 \  
  --no-retain-stacks \  
  --operation-preferences MaxConcurrentCount=1,FailureToleranceCount=0
```

如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [DeleteStackInstances](#)。

在堆疊刪除完成之後，選擇性地執行 `describe-stack-set-operation` 命令來顯示刪除堆疊操作的狀態和結果，以確認堆疊已從您的 StackSet 刪除。針對 `--operation-id`，使用您的 `delete-stack-instances` 命令所傳回的操作 ID。

```
aws cloudformation describe-stack-set-operation --stack-set-name my-stackset \  
  --operation-id ddf16f54-ad62-4d9b-b0ab-3ed8e9example
```

Delete CloudFormation StackSets

若要刪除 StackSet，您必須先刪除 StackSet 中的所有堆疊。有關如何刪除所有堆疊的詳細資訊，請參閱 [從 StackSets 刪除堆疊](#)。

主題

- [刪除 StackSet \(主控台\)](#)

- [刪除 StackSet \(AWS CLI\)](#)
- [刪除服務角色 \(選用\)](#)

刪除 StackSet (主控台)

刪除 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在螢幕上方的導覽列中，選擇您在其中建立 StackSet 所在的 AWS 區域。
3. 在 StackSets 頁面上，選取 StackSet。
4. 選擇 StackSet 後，從動作選單選擇刪除 StackSet。
5. 當系統提示您確認是否要刪除堆疊集時，請選擇刪除 StackSet。

刪除 StackSet (AWS CLI)

Note

做為委派管理員時，您必須在命令中包含 `--call-as DELEGATED_ADMIN`。

刪除 StackSet

1. 使用下列 `delete-stack-set` 命令。當系統提示您確認時，請鍵入 **y**，然後按下 Enter 鍵。

```
aws cloudformation delete-stack-set --stack-set-name my-stackset
```

2. 執行 `list-stack-sets` 命令以確認 StackSet 已刪除。`list-stack-sets` 命令的結果應該會顯示堆疊及其 DELETED 狀態。

```
aws cloudformation list-stack-sets
```

刪除服務角色 (選用)

若您不再需要 CloudFormation 執行 StackSet 操作所需的 IAM 服務角色，建議您刪除角色。

對於自我管理的 StackSets，是您建立的角色。如需使用者角色的詳細資訊，請參閱 [授與自我管理許可](#)。

對於服務受管的 StackSets，本指南中由 StackSets 自動建立的角色在組織管理帳戶中的字尾為 CloudFormationStackSetsOrgAdmin，在每個目標帳戶中為 CloudFormationStackSetsOrgMember。如需詳細資訊，請參閱 [服務連結角色](#)。

刪除服務角色 (主控台)

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色，然後勾選您所要刪除之角色旁的核取方塊。
3. 在頁面頂端的 Role actions (角色動作) 選單中，選擇 Delete role (刪除角色)。
4. 在確認對話方塊中，選擇 Yes, Delete (是的，刪除)。如果您確定，可以繼續刪除，即使服務上次存取的資料仍持續載入。

刪除服務角色 (AWS CLI)

- 使用下列 delete-role 命令。當系統提示您確認時，請鍵入 **y**，然後按下 Enter。

```
aws iam delete-role --role-name role name
```

如需有關刪除角色的詳細資訊，請參閱《IAM 使用者指南》中的 [刪除角色或執行個體設定檔](#)。

使用目標帳戶閘道來防止失敗的 StackSets 部署

帳戶閘道是一項選用功能，可讓協助您確認，在 CloudFormation 於目標帳戶中開始 StackSets 操作之前驗證此帳戶是否符合特定需求。此驗證是透過做為先決條件檢查的 AWS Lambda 函數執行。

帳戶閘道的常見範例，是驗證目標帳戶上不存在作用中或未解決的 CloudWatch 警示。CloudFormation 會在您每次在目標帳戶中開始堆疊操作前調用該 Lambda 函式，並且僅在此函數中傳回 SUCCEEDED 程式碼。如果 Lambda 函式傳回 FAILED 狀態，則 CloudFormation 不會繼續您請求的操作。如果您未設定帳戶閘道 Lambda 函式，則 CloudFormation 會略過檢查，並繼續您的操作。

如果您目標帳戶的帳戶閘道檢查失敗，則失敗操作將計入您指定的堆疊容錯能力數目或百分比。如需容錯能力的詳細資訊，請參閱 [StackSet 操作選項](#)。

帳戶閘道僅適用於 StackSets 操作。此功能不適用於 StackSets 之外的其他 CloudFormation 操作。

要求

帳戶閘道必須符合以下要求：

- 您的 Lambda 函式必須命名為 `AWSCloudFormationStackSetAccountGate`，才能使用此功能。
- `AWSCloudFormationStackSetExecutionRole` 需要許可，才能調用您的 Lambda 函式。如果沒有這些許可，CloudFormation 會略過帳戶閘道檢查，並繼續堆疊操作。
- Lambda `InvokeFunction` 許可必須新增至目標帳戶，帳戶閘道才能運作。目標帳戶信任政策必須具有與管理員帳戶的信任關係。下列是授予 Lambda `InvokeFunction` 許可的範例政策聲明。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "*"
    }
  ]
}
```

用於建立 Lambda 函式的 CloudFormation 範本

使用以下範例範本來建立 Lambda `AWSCloudFormationStackSetAccountGate` 函式。要使用任一範本建立新堆疊，請參閱 [從 CloudFormation 主控台建立堆疊](#)。

範本位置	Description
https://s3.amazonaws.com/cloudformation-sta-ckset-sample-templates-us-east-1/AccountGateSucceeded.yml	建立實作 Lambda 帳戶閘道函數 (將傳回 SUCCEEDED 狀態) 的堆疊。

範本位置	Description
https://s3.amazonaws.com/cloudformation-templates-us-east-1/AccountGateFailed.yml	建立實作 Lambda 帳戶閘道函數 (將傳回 FAILED 狀態) 的堆疊。

選擇 CloudFormation StackSets 的並行模式

並行模式是一個參數，讓 [StackSetOperationPreferences](#) 允許您選擇並行層級在堆疊集操作期間的行為方式。您可以選擇以下模式：

- **嚴格容錯能力：**此選項會動態降低並行層級，以確保故障帳戶數目絕不會超過容錯能力 +1 的值。初始實際並行值會設定為並行帳戶數目上限值，或容錯能力 +1 的值，兩者中的較低值。實際並行值隨後會依故障次數按比例減少。這是預設行為。
- **軟性容錯能力：**此選項會將容錯能力與實際並行值分離。這可讓堆疊集操作以並行帳戶數目上限值設定的並行層級執行，無論故障次數為何。

嚴格容錯能力會降低部署速度，由於每次故障會使並行值減少，因此堆疊集操作會發生故障。軟性容錯能力會優先考慮部署速度，同時仍然利用 CloudFormation 安全功能。這可讓您檢閱並處理堆疊集操作故障問題，以解決與現有資源、服務配額和許可相關的常見問題。

如需 StackSets 堆疊操作故障的詳細資訊，請參閱 [堆疊操作失敗的常見原因](#)。

如需有關並行帳戶數目上限與容錯能力的詳細資訊，請參閱 [StackSet 操作選項](#)。

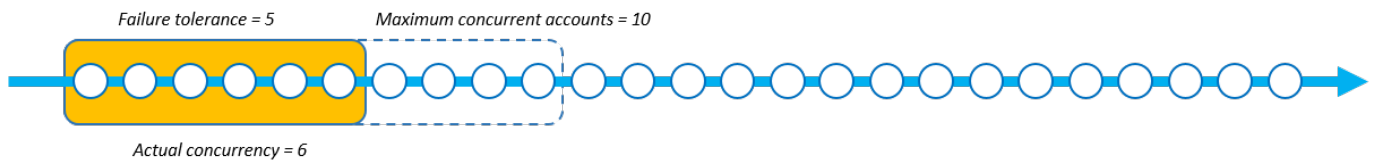
每個並行模式的運作方式

下圖提供了堆疊集操作期間，每個並行模式運作方式的視覺化呈現。節點字串代表單一 的部署 AWS 區域，而每個節點都是目標 AWS 帳戶。

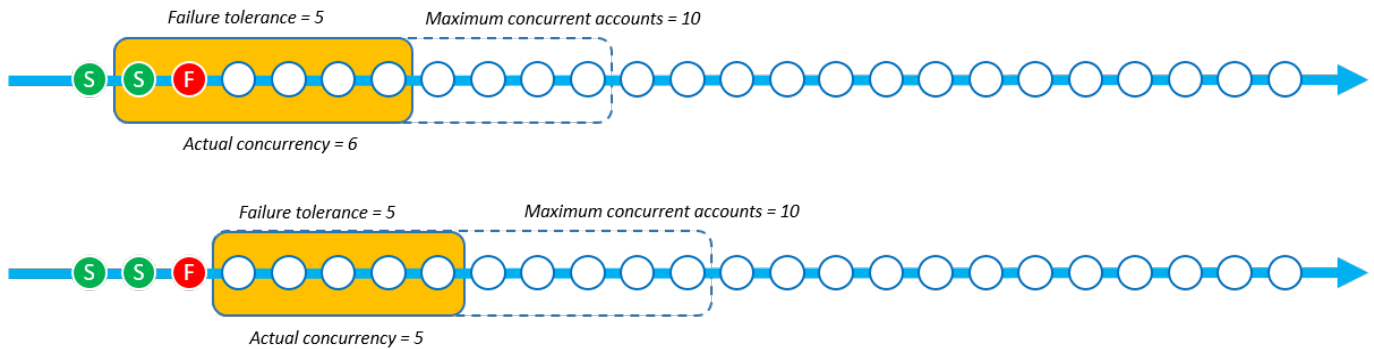
嚴格容錯能力

當使用嚴格容錯能力的堆疊集操作將容錯能力值設定為 5，並將並行帳戶數目上限值設定為 10 時，則實際並行值為 6。實際並行值為 6，因為此容錯能力 5 +1 低於並行帳戶數目上限的值。

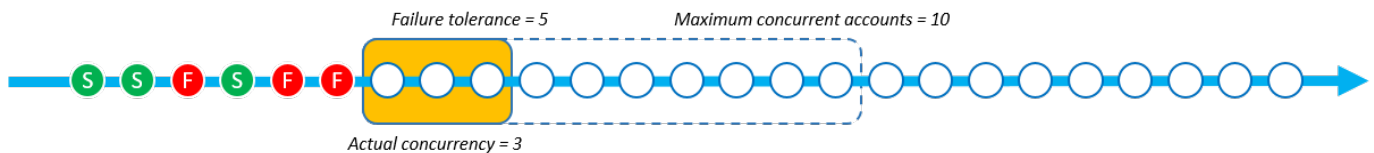
下圖顯示容錯能力對並行帳戶數目上限值的影響，以及對堆疊集操作實際並行值的影響：



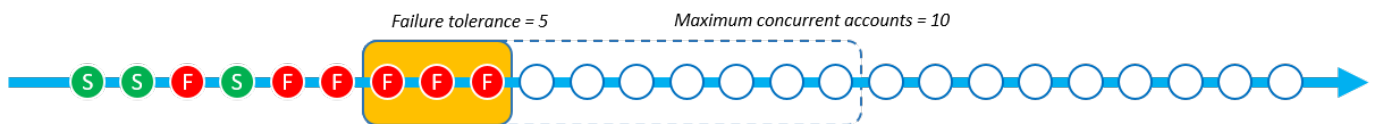
當部署開始且存在故障堆疊執行個體時，實際並行值會減少，以提供安全的部署體驗。當 StackSets 無法部署 1 個堆疊執行個體時，實際並行值會從 6 減少到 5。



嚴格容錯能力模式會按比例將實際並行值減少至故障堆疊執行個體數目。在下列範例中，當 StackSets 無法部署 2 個以上的堆疊執行個體時，實際並行值會從 5 減少到 3，讓故障堆疊執行個體總計達到 3。



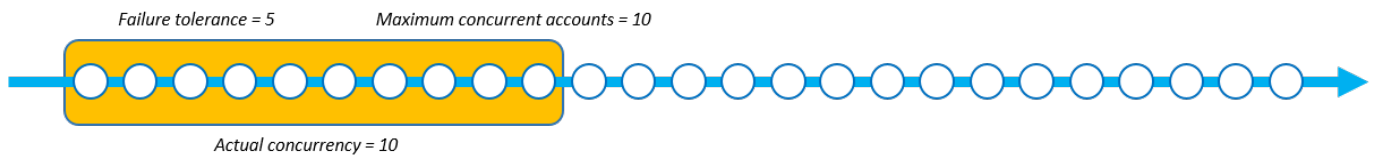
當故障堆疊執行個體數目等於容錯能力 +1 的定義值時，StackSets 無法執行堆疊集操作。在下列範例中，當有 6 個故障堆疊執行個體且容錯能力值為 5 時，StackSets 無法執行操作。



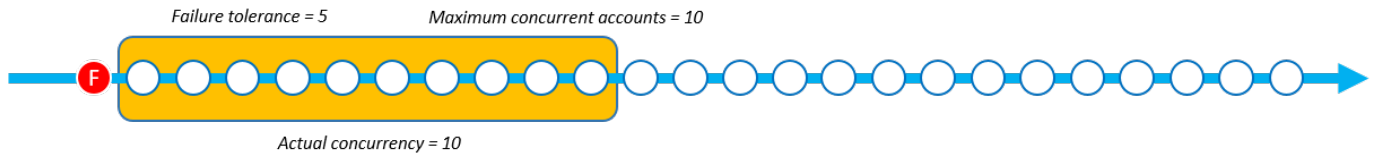
在此範例中，CloudFormation 在停止堆疊集操作之前部署了 9 個堆疊執行個體 (3 個成功和 6 個失敗)。

軟性容錯能力

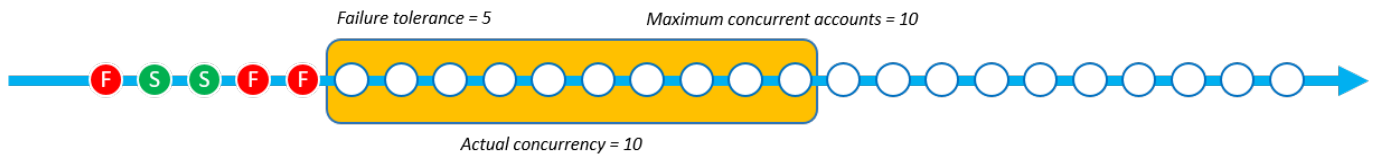
當使用軟性容錯能力的堆疊集操作將容錯能力設定為 5，並將並行帳戶數目上限值設定為 10 時，實際並行值為 10。



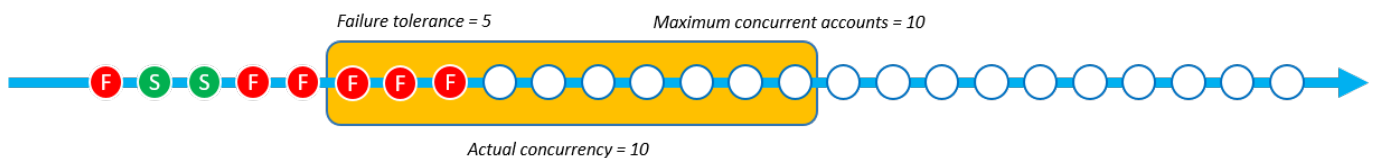
當部署開始且有故障堆疊執行個體時，不會變更實際並行值。在下列範例中，1 個堆疊操作失敗，但實際並行值保持在 10。



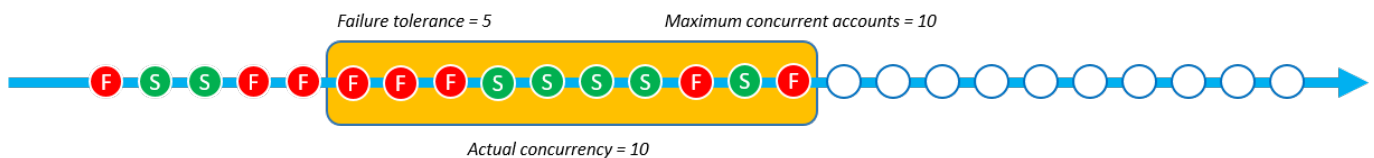
即使再出現 2 個堆疊執行個體故障，實際並行值仍會保持在 10。



當失敗的堆疊執行個體超過容錯能力時，StackSets 無法執行堆疊集操作。在下列範例中，當有 6 個故障堆疊執行個體且容錯能力計數為 5 時，StackSets 無法執行操作。但是，並行佇列中的其餘操作完成之前，操作不會結束。



StackSets 會繼續部署已位於並行佇列中的堆疊執行個體。這意味著故障堆疊執行個體數目可能會高於容錯能力。在下列範例中，有 8 個故障堆疊執行個體，因為即使堆疊集操作已達到容錯能力 5，並行佇列仍有 7 個要執行的操作。



在此範例中，StackSets 在停止堆疊操作之前部署了 15 個堆疊執行個體 (7 個成功和 8 個失敗)。

根據部署速度，選擇嚴格容錯能力和軟性容錯能力

依據您偏好的堆疊集部署速度和許可的部署故障次數，選擇嚴格容錯能力和軟性容錯能力模式。

下表顯示了每個並行模式如何在嘗試部署共計 1000 個堆疊執行個體時，處理失敗的堆疊集操作。在每個案例中，將容錯能力值設定為 100 個堆疊執行個體，並將並行帳戶數目上限值設定為 250 個堆疊執行個體。

StackSets 實際上會將帳戶做為滑動視窗排入佇列 (請參閱 [每個並行模式的運作方式](#))，但此範例會按批次形式顯示操作以示範每種模式的速度。

嚴格容錯能力

此範例使用嚴格容錯能力模式，降低相對於每個前述批次中發生之故障次數的實際並行值。每個批次有 20 個故障執行個體，隨後將下列批次的實際並行值降低 20，直至堆疊集操作達到容錯能力值為 100 為止。

在下表中，第一個批次的初始實際並行值為 101 個堆疊執行個體。實際並行值為 101，因為該值是並行帳戶上限值 (250) 和容錯能力 (100) +1 兩者中的較低值。每個批次包含 20 個故障堆疊執行個體部署，這隨後會將每個後續批次的實際並行值減少 20 個堆疊執行個體。

嚴格容錯能力	第 1 批次	第 2 批次	第 3 批次	第 4 批次	第 5 批次	第 6 批次
實際並行計數	101	81	61	4.1	21	-
故障執行個體計數	20	20	20	20	20	-
成功堆疊執行個體計數	81	61	4.1	21	1	-

使用嚴格容錯能力的操作會在堆疊集操作達到 100 個堆疊執行個體的容錯能力時，分 5 個批次完成 305 個堆疊執行個體部署。堆疊集操作會在失敗之前成功部署 205 個堆疊執行個體。

軟性容錯能力

此範例使用軟性容錯能力模式，無論故障執行個體數目為何，都會保持與並行帳戶數目上限值定義的實際並行計數相同 (250 個堆疊執行個體)。堆疊集操作會保持與實際並行值相同，直至達到 100 個執行個體的容錯能力為止。

在下表中，第一個批次的初始實際並行值為 250 個堆疊執行個體。實際並行值為 250，因為已將並行帳戶數目上限值設定為 250，並且軟性容錯能力模式允許 StackSets 使用此值做為實際並行值，而無論故障次數為何。即使此範例中的每個批次都有 50 次故障，實際並行值仍不會受到影響。

軟性容錯能力	第 1 批次	第 2 批次	第 3 批次	第 4 批次	第 5 批次	第 6 批次
實際並行計數	250	250	-	-	-	-
故障執行個體計數	50	50	-	-	-	-
成功堆疊執行個體計數	200	200	-	-	-	-

使用相同的並行帳戶數目上限值和容錯能力值時，使用軟性容錯能力模式的操作會以 2 個批次完成 500 個堆疊執行個體部署。堆疊集操作會在失敗之前成功部署 400 個堆疊執行個體。

選擇並行模式 (主控台)

建立或更新堆疊集時，請在設定部署選項頁面的並行模式中選擇嚴格容錯能力或軟性容錯能力。

選擇並行模式 (AWS CLI)

您可使用 `ConcurrencyMode` 參數搭配下列 StackSets 命令：

- [create-stack-instances](#)
- [delete-stack-instances](#)
- [detect-stack-set-drift](#)
- [import-stacks-to-stack-set](#)
- [update-stack-instances](#)

- [update-stack-set](#)

這些命令具有一個稱為 `--operation-preferences` 並且可使用 `ConcurrencyMode` 設定的現有參數。 `ConcurrencyMode` 可設定為下列其中一個值：

- `STRICT_FAILURE_TOLERANCE`
- `SOFT_FAILURE_TOLERANCE`

下列範例使用 `STRICT_FAILURE_TOLERANCE` `ConcurrencyMode` 建立堆疊執行個體，並將 `FailureToleranceCount` 設定為 10，將 `MaxConcurrentCount` 設定為 5。

```
aws cloudformation create-stack-instances \  
  --stack-set-name example-stackset \  
  --accounts 123456789012 \  
  --regions eu-west-1 \  
  --operation-preferences  
  ConcurrencyMode=STRICT_FAILURE_TOLERANCE,FailureToleranceCount=10,MaxConcurrentCount=5
```

 Note

如需建立和更新堆疊集的詳細程序，請參閱下列主題：

- [建立 StackSets \(自我管理許可\)](#)
- [建立 StackSets \(具有服務受管許可\)](#)
- [更新 StackSets](#)

在 CloudFormation StackSets 上執行漂移偵測

即使您透過 CloudFormation 管理您的堆疊，以及這些堆疊中包含的資源，使用者仍然可以在 CloudFormation 外部變更這些資源。使用者可以使用建立資源的基礎服務來直接編輯資源。藉由在 StackSet 上執行漂移偵測，您可以判斷是否有任何屬於該 StackSet 的堆疊執行個體與其預期組態出現差異 (或是產生了漂移)。

主題

- [CloudFormation 在 StackSet 上執行漂移偵測的方式](#)
- [偵測 StackSet 上的漂移 \(主控台\)](#)

- [偵測 StackSet 上的漂移 \(AWS CLI\)](#)
- [停止 StackSet 上的漂移偵測](#)

CloudFormation 在 StackSet 上執行漂移偵測的方式

CloudFormation 在 StackSet 上執行漂移偵測時，會在與 StackSet 中每個堆疊執行個體相關聯的堆疊上執行漂移偵測。為了執行此作業，CloudFormation 會將堆疊中每個資源目前的狀態與該資源的預期狀態進行比較，如堆疊範本及任何指定輸入參數所定義。如果資源目前的狀態與預期狀態不同，便會將該資源視為已偏離。如果堆疊中有一或多個資源產生偏離，便會將堆疊本身視為已偏離，且也會將與該堆疊建立關聯的堆疊執行個體視為已偏離。如果堆疊中有一或多個堆疊執行個體產生漂移，也會將 StackSet 本身視為已漂移。

漂移偵測可以識別非受管的變更，即在 CloudFormation 外部對堆疊進行的變更。直接透過 CloudFormation 對堆疊進行的變更，而非在 StackSet 層級進行的變更，則不會視為漂移。例如，假設您有一個與 StackSet 中堆疊執行個體建立關聯的堆疊。如果您使用 CloudFormation 來更新堆疊，以使用不同的範本，這便不會被視為漂移，即使該堆疊現在具備的範本與屬於該 StackSet 的任何其他堆疊不同也一樣。這是因為堆疊仍然與其預期範本和 CloudFormation 中的參數組態相符。

如需 CloudFormation 在堆疊上執行漂移偵測方式的詳細資訊，請參閱 [透過漂移偵測功能，偵測堆疊和資源的未受管組態變更](#)。

因為 CloudFormation 會在每個堆疊上個別執行漂移偵測，其會在判斷堆疊是否漂移時，將任何覆寫的參數值納入考量。如需覆寫堆疊執行個體中範本參數的詳細資訊，請參閱 [覆寫 CloudFormation StackSet 中堆疊的參數值](#)。

如果您[直接在與堆疊執行個體建立關聯的堆疊上](#)執行漂移偵測，這些漂移結果便無法從 StackSets (堆疊集合) 主控台頁面取得。

偵測 StackSet 上的漂移 (主控台)

偵測 StackSet 上的漂移

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在 StackSets 頁面上，選取您希望執行漂移偵測的 StackSets。
3. 在 Actions (動作) 選單中，選取 Detect drifts (偵測偏離)。

CloudFormation 會顯示資訊列，說明已啟動所選 StackSet 的漂移偵測。

4. 選用：監控漂移偵測操作的進度：

- a. 選取 StackSet 名稱，以顯示 StackSet 詳細資訊頁面。
- b. 選取 Operations (操作) 標籤、選取漂移偵測操作，然後選取 View drift details (檢視漂移詳細資訊)。

CloudFormation 會顯示 Operation details (操作詳細資訊) 對話方塊。

5. 等待 CloudFormation 完成漂移偵測操作。當漂移偵測操作完成後，CloudFormation 會更新堆疊的漂移狀態和上次檢查漂移的時間。這些欄位會列在所選取 StackSet 的 StackSet 詳細資訊頁面的概觀索引標籤上。

漂移偵測操作可能會耗費一些時間，這取決於 StackSet 中所包含的堆疊執行個體數，及包含在 StackSet 中的資源數。您在指定 StackSet 上一次只能執行單一漂移偵測操作。即使關閉資訊列，CloudFormation 仍會繼續執行漂移偵測操作。

6. 如要檢閱 StackSet 中堆疊執行個體的漂移偵測結果，請選取堆疊執行個體索引標籤。

Stack name (堆疊名稱) 行會列出與每個堆疊執行個體建立關聯的堆疊名稱，Drift status (偏離狀態) 行則會列出該堆疊的偏離狀態。如果堆疊的一個或多個資源已偏離，堆疊就視為已偏離。

7. 檢閱與特定堆疊執行個體建立關聯的堆疊漂移偵測結果：
 - a. 選擇操作子標籤。
 - b. 選擇您想要檢視漂移偵測結果的漂移操作。分割面板將顯示堆疊執行個體狀態和所選操作的原因。對於漂移操作，狀態原因欄顯示堆疊執行個體的漂移狀態。
 - c. 選擇您想檢視漂移詳細資訊的堆疊執行個體，然後選擇檢視資源漂移。在資源漂移頁面上，資源漂移狀態表單會列出每項堆疊資源及其漂移狀態，以及上次在資源上啟動漂移偵測的時間。每個資源會顯示邏輯 ID 和實體 ID，以協助您識別它們。
8. 您可以根據資源的偏移狀態，使用 Drift status (偏離狀態) 欄排序資源。

檢視已修改過的資源的詳細資訊：

- 在選取資源的情況下，選擇檢視漂移詳細資訊。

CloudFormation 會顯示該特定資源的漂移詳細資訊頁面。此頁面列出了資源的差異。其還會列出資源的預期屬性值和目前屬性值。

 Note

如果堆疊屬於與您目前登入的區域和帳戶不同的區域和帳戶，偵測漂移按鈕將會停用，且您將無法檢視詳細資訊。

偵測 StackSet 上的漂移 (AWS CLI)

若要使用 偵測整個堆疊上的偏離 AWS CLI，請使用下列程序：

偵測 StackSet 上的漂移

1. 使用 [detect-stack-set-drift](#) 命令以偵測整個 StackSet 以及其相關聯堆疊執行個體上的漂移。

以下範例會在 StackSet `stack-set-drift-example` 上啟動漂移偵測。

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

輸出：

```
{  
  "OperationId": "c36e44aa-3a83-411a-b503-cb611example"  
}
```

2. 因為 StackSet 漂移偵測操作可能會長時間執行，請使用 [describe-stack-set-operation](#) 命令來監控漂移操作的狀態。此命令接受由 `detect-stack-set-drift` 命令傳回的 StackSet 操作 ID。

下列範例會使用先前範例的操作 ID 來傳回 StackSet 漂移偵測操作的資訊。在此範例中，操作仍正在執行。在與 StackSet 建立關聯的七個堆疊執行個體中，已找到一個堆疊執行個體已產生漂移，兩個執行個體正在同步，而其餘四個堆疊執行個體的漂移偵測則仍在進行。由於一個執行個體已產生漂移，StackSet 本身的漂移狀態現在是 DRIFTED。

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name stack-set-drift-example \  
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

輸出：

```
{
  "StackSetOperation": {
    "Status": "RUNNING",
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "OperationPreferences": {
      "RegionOrder": []
    },
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "StackSetDriftDetectionDetails": {
      "DriftedStackInstancesCount": 1,
      "TotalStackInstancesCount": 7,
      "LastDriftCheckTimestamp": "2019-12-04T20:34:28.543Z",
      "InSyncStackInstancesCount": 2,
      "InProgressStackInstancesCount": 4,
      "DriftStatus": "DRIFTED",
      "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}
```

稍後執行相同的命令，此範例便會顯示漂移偵測操作完成後傳回的資訊。與此 StackSet 建立關聯的七個堆疊執行個體中，有兩個已產生漂移，使 StackSet 本身的漂移狀態成為 DRIFTED。

```
aws cloudformation describe-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id c36e44aa-3a83-411a-b503-cb611example
```

輸出：

```
{
  "StackSetOperation": {
    "Status": "SUCCEEDED",
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "OperationPreferences": {
```

```

        "RegionOrder": []
    }
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "EndTimestamp": "2019-12-04T20:37:32.829Z",
    "StackSetDriftDetectionDetails": {
        "DriftedStackInstancesCount": 2,
        "TotalStackInstancesCount": 7,
        "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
        "InSyncStackInstancesCount": 5,
        "InProgressStackInstancesCount": 0,
        "DriftStatus": "DRIFTED",
        "FailedStackInstancesCount": 0
    },
    "Action": "DETECT_DRIFT",
    "CreationTimestamp": "2019-12-04T20:33:13.673Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",
    "OperationId": "c36e44aa-3a83-411a-b503-cb611example"
}
}

```

3. 完成 StackSet 的漂移偵測操作時，請使用 `describe-stack-set`、`list-stack-instances`、`describe-stack-instance` 和 `list-stack-instance-resource-drifts` 命令來檢閱結果。

[describe-stack-set](#) 命令會包含 `describe-stack-set-operation` 命令傳回的相同詳細漂移資訊。

```

aws cloudformation describe-stack-set \
  --stack-set-name stack-set-drift-example

```

輸出：

```

{
  "StackSet": {
    "Status": "ACTIVE",
    "Description": "Demonstration of drift detection on stack sets.",
    "Parameters": [],
    "Tags": [
      {
        "Value": "Drift detection",
        "Key": "Feature"
      }
    ],
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",

```

```

    "Capabilities": [],
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "StackSetDriftDetectionDetails": {
        "DriftedStackInstancesCount": 2,
        "TotalStackInstancesCount": 7,
        "LastDriftCheckTimestamp": "2019-12-04T20:36:55.612Z",
        "InProgressStackInstancesCount": 0,
        "DriftStatus": "DRIFTED",
        "DriftDetectionStatus": "COMPLETED",
        "InSyncStackInstancesCount": 5,
        "FailedStackInstancesCount": 0
    },
    "StackSetARN": "arn:aws:cloudformation:us-east-1:123456789012:stackset/
stack-set-drift-example:bd1f4017-d4f9-432e-a73f-8c22example",
    "TemplateBody": [details omitted],
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22ebexample",
    "StackSetName": "stack-set-drift-example"
}
}

```

您可以使用 [list-stack-instances](#) 命令來傳回與 StackSet 建立關聯的堆疊執行個體摘要資訊，包括每個堆疊執行個體的漂移狀態。

在此範例中，在範例 StackSet 上執行 list-stack-instances 並將漂移狀態篩選設為，DRIFTED可讓您識別哪兩個堆疊執行個體的漂移狀態為 DRIFTED。

```

aws cloudformation list-stack-instances \
  --stack-set-name stack-set-drift-example \
  --filters Name=DRIFT_STATUS,Values=DRIFTED

```

輸出：

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
StackSet-stack-set-drift-example-b0fb6083-60c0-4e39-
af15-2f071e0db90c/0e4f0940-16d4-11ea-93d8-0641cexample",
      "Status": "CURRENT",
      "Account": "012345678910",

```

```

        "Region": "eu-west-1",
        "LastDriftCheckTimestamp": "2019-12-04T20:37:32.687Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    {
        "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
        "Status": "CURRENT",
        "Account": "123456789012",
        "Region": "us-east-1",
        "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
        "DriftStatus": "DRIFTED",
        "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
        "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
    },
    [additional stack instances omitted]
]
}

```

[describe-stack-instance](#) 命令也會傳回此資訊，但只會傳回單一堆疊執行個體的資訊，如以下範例所示。

```

aws cloudformation describe-stack-instance \
  --stack-set-name stack-set-drift-example \
  --stack-instance-account 012345678910 --stack-instance-region us-east-1

```

輸出：

```

{
  "StackInstance": {
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
StackSet-stack-set-drift-example-b7fde68e-e541-44c2-b33d-
ef2e2988071a/008e6030-16d4-11ea-8090-12f89example",
    "Status": "CURRENT",
    "Account": "123456789012",

```

```

    "Region": "us-east-1",
    "ParameterOverrides": [],
    "DriftStatus": "DRIFTED",
    "LastDriftCheckTimestamp": "2019-12-04T20:34:28.275Z",
    "StackSetId": "stack-set-drift-example:bd1f4017-d4f9-432e-
a73f-8c22eexample",
    "LastOperationId": "c36e44aa-3a83-411a-b503-cb611example"
  }
}

```

4. 在您識別漂移的堆疊執行個體後，您可以使用 `list-stack-instances` 或 `describe-stack-instance` 命令傳回的堆疊執行個體資訊，來執行 [list-stack-instance-resource-drifts](#) 命令。此命令會傳回堆疊中已為特定漂移操作產生漂移的資源的詳細資訊。

以下範例使用 `--stack-instance-resource-drift-statuses` 參數，以請求在上一個漂移操作範例中已修改或已刪除資源的堆疊漂移資訊。該請求傳回已修改的一項資源的相關資訊，包括它的兩項屬性及其更改值的詳細資訊。未刪除任何資源。

```

aws cloudformation list-stack-instance-resource-drifts \
  --stack-set-name my-stack-set-with-resource-drift \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-east-1 \
  --operation-id c36e44aa-3a83-411a-b503-cb611example \
  --stack-instance-resource-drift-statuses MODIFIED DELETED

```

輸出：

```

{
  "Summaries": [
    {
      "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-
stack-set-with-resource-drift/489e5570-df85-11e7-a7d9-50example",
      "ResourceType": "AWS::SQS::Queue",
      "Timestamp": "2018-03-26T17:23:34.489Z",
      "PhysicalResourceId": "https://sqs.us-
east-1.amazonaws.com/123456789012/my-stack-with-resource-drift-Queue-494PBHC076H4",
      "StackResourceDriftStatus": "MODIFIED",
      "PropertyDifferences": [
        {
          "PropertyPath": "/DelaySeconds",
          "ActualValue": "120",
          "ExpectedValue": "20",

```

```
        "DifferenceType": "NOT_EQUAL"
      },
      {
        "PropertyPath": "/RedrivePolicy/maxReceiveCount",
        "ActualValue": "12",
        "ExpectedValue": "10",
        "DifferenceType": "NOT_EQUAL"
      }
    ],
    "LogicalResourceId": "Queue"
  }
]
```

停止 StackSet 上的漂移偵測

因為 StackSet 上的漂移偵測操作可能會耗費相當長的時間，有時候您可能會希望停止目前正在 StackSet 上執行的漂移偵測操作。

在 StackSet 上停止漂移偵測 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在 StackSets 頁面上，選取 StackSet 的名稱。

CloudFormation 會顯示所選取 StackSets 的 StackSets 詳細資訊。

3. 在 StackSets details (堆疊集合詳細資訊) 頁面上，選取 Operations (操作) 標籤，然後選取漂移偵測操作。
4. 選取 Stop operation (停止操作)。

在 StackSet 上停止漂移偵測 (AWS CLI)

- 使用 [stop-stack-set-operation](#) 命令。您必須提供 StackSet 名稱和漂移偵測 StackSet 操作的操作 ID。

```
aws cloudformation stop-stack-set-operation \
  --stack-set-name stack-set-drift-example \
  --operation-id 624af370-311a-11e8-b6b7-500cexample
```

將堆疊匯入 CloudFormation StackSets

堆疊匯入操作可以將現有堆疊匯入新的或現有的 StackSets，以便您可以在一次操作中將現有 StackSets 移轉至堆疊集。

針對自我管理 StackSets，匯入操作可以在管理員帳戶或不同的目標帳戶和 AWS 區域中匯入堆疊。對於服務受管 StackSets，匯入操作可以在 AWS Organizations 與管理帳戶相同的中匯入任何堆疊。

將堆疊匯入 StackSets 時的考量和限制如下：

- 匯入操作最多可以使用內嵌堆疊 ID 匯入 10 個堆疊，或使用 Amazon S3 物件匯入最多 200 個堆疊。
- 不支援 NoEcho 屬性。包含 NoEcho 的堆疊將不會透過 StackSets 匯入匯入到新的堆疊集。
- StackSet 只能屬於一個 StackSet。
- 您可以透過在堆疊匯入操作中明確指定標籤作為參數來實作 StackSet 的堆疊標籤。
- 匯入操作期間不會影響堆疊的自訂參數覆寫。
- 無法透過編輯 StackSet 詳細資訊選項來更新已匯入堆疊執行個體的參數。您必須使用覆寫 StackSet 參數。如需覆寫參數的詳細資訊，請參閱 [覆寫堆疊上的參數](#)。
- 匯入堆疊時，StackSets 配額和堆疊執行個體會套用。如需配額的詳細資訊，請參閱 [了解 CloudFormation 配額](#)。

主題

- [for CloudFormation StackSets 的自我管理堆疊匯入](#)
- [for CloudFormation StackSets 的服務受管堆疊匯入](#)
- [還原堆疊匯入至 CloudFormation StackSets](#)

for CloudFormation StackSets 的自我管理堆疊匯入

CloudFormation 堆疊匯入操作可以將現有堆疊匯入新的或現有的 StackSets，以便您可以在一次操作中將現有堆疊移轉至 StackSets。透過使用堆疊匯入，您可以避免停機時間和中斷，而無需刪除和重新建立這些資源。一旦堆疊匯入到 StackSet 中，原始堆疊將成為指定 StackSet 的堆疊執行個體。

自我管理堆疊匯入的考量事項

- 堆疊匯入操作需要您建立 StackSet 所在的管理員帳戶，以及包含堆疊的目標帳戶。

- 目標帳戶必須具備許可來搭配使用 GetTemplate 操作與堆疊 ID 或 ARN 輸入。因此，您的管理員帳戶必須獲得 `AWSCloudFormationStackSetAdministrationRole` 或 `AWSCloudFormationStackSetsExecutionRole` 許可。

主題

- [將現有堆疊匯入新 StackSet \(主控台\)](#)
- [將現有堆疊匯入現有 StackSet \(主控台\)](#)
- [將堆疊匯入 StackSet \(AWS CLI\)](#)

將現有堆疊匯入新 StackSet (主控台)

開始之前，識別您要匯入的堆疊。

- 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
- 從導覽窗格選擇 StackSets。
- 選擇 StackSets 頁面上方的 Create StackSet (建立 StackSet)。
- 在選擇範本頁面上，使用下列其中一個選項指定範本，然後選擇下一步。
 - 選擇 Amazon S3 URL，然後在文字方塊中為您的範本指定 URL。
 - 選擇 Upload a template file (上傳範本檔案)，然後瀏覽您的範本。
 - 選擇 From stack ID (通過堆疊 ID)，然後輸入您的堆疊 ID。
- 在指定 StackSet 詳細資訊頁面上，輸入您要建立的堆疊集名稱並選擇下一步。

(選用) 輸入 StackSet 說明。

- 在設定 StackSet 選項頁面上檢閱所需選項，然後選擇下一步。
- 在設定部署選項頁面上，選擇將堆疊匯入堆疊集。
- 在 Stacks to import (要匯入的堆疊) 欄位中輸入您想要匯入的堆疊的堆疊 ID。例如 `arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786`。

(選用) 選擇新增另一個堆疊 ID，然後輸入您想要匯入的另一個堆疊的堆疊 ID。每個堆疊匯入操作最多可以新增 10 個堆疊。

- 檢閱您的部署選項，然後選擇下一步。

10. 在檢閱頁面上，檢閱您的選擇以及 StackSet 的屬性。當您準備好將堆疊匯入堆疊集時，請選擇提交。

結果：匯入的堆疊現在是指定 StackSet 的堆疊執行個體。要進一步了解堆疊匯入狀態，請參閱 [StackSets 狀態碼](#)。

將現有堆疊匯入現有 StackSet (主控台)

開始之前，識別您要匯入的堆疊。

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從導覽窗格選擇 StackSets。
3. 在 StackSets 頁面上，選擇您想要匯入堆疊的 StackSet。
4. 選擇 StackSet 後，從動作選單選擇將堆疊新增至 StackSet。
5. 在設定部署選項頁面上，選擇將堆疊匯入堆疊集，然後在要匯入的堆疊欄位輸入您想要匯入之堆疊的堆疊 ID。例如 *arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786*。

(選用) 選擇新增另一個堆疊 ID，然後輸入您想要匯入的另一個堆疊的堆疊 ID。每個堆疊匯入操作最多可以新增 10 個堆疊。

6. 選擇下一步。
7. 在指定覆寫頁面上，檢閱您的選擇，然後選擇下一步。
8. 在檢閱頁面上，檢閱您的選擇以及 StackSet 的屬性。當您準備好建立 StackSet 時，請選擇提交。

結果：匯入的堆疊現在是指定 StackSet 的堆疊執行個體。要進一步了解堆疊匯入狀態，請參閱 [StackSets 狀態碼](#)。

將堆疊匯入 StackSet (AWS CLI)

將現有堆疊匯入新 StackSet

下列 `create-stack-set` 命令會建立 StackSet，並匯入指定的堆疊。要匯入的堆疊由其 ARN 識別。以您自己的資訊取代預留位置文字。

```
aws cloudformation create-stack-set \  
  --stack-set-name MyStackSet \  
  --template-url ...
```

```
--stack-id arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/466df9e0-0dff-08e3-8e2f-5088487c4896 \  
--administration-role-arn arn:aws:iam::123456789012:role/AWSCloudFormationStackSetAdministrationRole \  
--execution-role-name AWSCloudFormationStackSetExecutionRole
```

將現有堆疊匯入現有 StackSet

下列 `import-stacks-to-stack-sets` 命令會將指定的堆疊匯入 *MyStackSet* StackSet。要匯入的堆疊由其 ARN 識別。以您自己的資訊取代預留位置文字。

```
aws cloudformation import-stacks-to-stack-set \  
--stack-set MyStackSet \  
--stack-ids arn:aws:cloudformation:us-east-1:123456789012:stack/StackToImport/f449b250-b969-11e0-a185-5081d0136786
```

要指定多個堆疊，請針對 `--stack-ids` 選項的值使用下列格式。

```
--stack-ids "arn_1" "arn_2"
```

將匯入的堆疊複製到其他區域和帳戶

執行 `create-stack-instances` 命令，將堆疊執行個體新增至 StackSet。以您自己的資訊取代預留位置文字。

```
aws cloudformation create-stack-instances \  
--stack-set-name MyStackSet \  
--accounts '["account_ID_1","account_ID_2"]' \  
--regions '["region_1","region_2"]'
```

for CloudFormation StackSets 的服務受管堆疊匯入

CloudFormation 堆疊匯入操作可以將現有堆疊匯入新的或現有的 StackSets，以便您可以在一次操作中將現有堆疊移轉至 StackSets。StackSets 擴充了堆疊功能，讓您可以跨多個帳戶和區域使用單次操作建立、更新或刪除堆疊。

服務受管堆疊匯入的考量事項

- 堆疊匯入操作需要管理帳戶或委派管理員帳戶，您可以在其中管理相關聯的，AWS Organizations 例如使用 StackSets 啟用信任存取。

- 目標帳戶必須是管理帳戶或委派管理員帳戶 AWS Organizations 所管理 的成員。
- 目標堆疊會在其中一個目標 OU 中存在。
- 目標帳戶應為 的成員 AWS Organizations。
- AWS Organizations 存取權應處於 Organizations ACTIVATED 的狀態。
- 匯入的堆疊應該出現在任何成員帳戶中，而不是管理帳戶中。

主題

- [將服務受管堆疊匯入新的 StackSet \(主控台\)](#)
- [建立服務受管堆疊並將其匯入現有 StackSet \(主控台\)](#)
- [將服務受管堆疊匯入現有 StackSet \(主控台\)](#)
- [將服務受管堆疊匯入 StackSet \(AWS CLI\)](#)

將服務受管堆疊匯入新的 StackSet (主控台)

使用 將堆疊匯入新的 StackSet AWS 管理主控台

若要將新堆疊匯入 StackSet，請識別包含要匯入之資源的堆疊。

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從導覽窗格選擇 StackSets。
3. 選擇 StackSets 頁面上方的 Create StackSet (建立 StackSet)。
4. 在選擇客群頁面上，執行下列動作：
 - a. 對於 StackSet 許可模型，選擇服務受管許可。
 - b. 針對先決條件 - 準備範本，選擇範本已就緒，然後使用下列選項之一選擇您的範本：
 - 對於 Amazon S3 URL，可在 Amazon S3 URL 欄位中輸入您的 Amazon S3 URL。
 - 對於上傳範本檔案，選擇本機電腦上的 CloudFormation 範本。

接受您的設定並選擇 Next (下一步)。

5. 在指定堆疊詳細資訊頁面上，執行下列操作：
 - a. 在 StackSet 名稱方塊中輸入 StackSet 名稱。

- b. (選用) 在 StackSet description (StackSet 描述) 區段中輸入描述。

在設定 StackSet 選項頁面上檢閱所需選項，然後選擇下一步。

6. 在設定部署選項頁面上進行以下操作：

- a. 對於將堆疊新增至堆疊集，選擇將堆疊匯入堆疊集。
- b. 對於 Stacks to import (要匯入的堆疊)，選擇您的堆疊匯入方法。
 - i. 對於 Stack ID (堆疊 ID)，輸入您的堆疊 ID。
 - ii. 對於堆疊 URL，輸入 Amazon S3 URL。

7. 在相關單位單位下，執行以下操作：

- a. 選擇與組織關聯，以使用根 OU。
- b. 選擇與組織單位關聯，以輸入要匯入之堆疊的父 OU ID。例如，如果 Stack 1 和 Stack 2 在 OU1 下方，並且 Stack 3 在 OU2 下方，則輸入 OU1 和 OU2。

接受您的設定並選擇 Next (下一步)。

8. 在 Review (檢閱) 頁面上，檢閱您的設定，然後選擇 Submit (提交)。

建立服務受管堆疊並將其匯入現有 StackSet (主控台)

若要將現有堆疊匯入新的 StackSet，請識別包含您要匯入之資源的堆疊。

建立 StackSet 和匯入堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從導覽窗格選擇 StackSets。
3. 選擇 StackSets 頁面上方的 Create StackSet (建立 StackSet)。
4. 在選擇客群頁面上，執行下列動作：
 - a. 對於 StackSet 許可模型，選擇服務受管許可。
 - b. 針對先決條件 - 準備範本，選擇範本已就緒，然後使用下列選項之一選擇您的範本：
 - 對於 Amazon S3 URL，可在 Amazon S3 URL 欄位中輸入您的 Amazon S3 URL。
 - 對於上傳範本檔案，選擇本機電腦上的 CloudFormation 範本。

接受您的設定並選擇 Next (下一步)。

5. 在指定堆疊詳細資訊頁面上，執行下列操作：

- a. 在 StackSet 名稱方塊中輸入 StackSet 名稱。
- b. (選用) 在 StackSet description (StackSet 描述) 區段中輸入描述。

在設定 StackSet 選項頁面上檢閱所需選項，然後選擇下一步。

6. 在設定部署選項頁面上進行以下操作：

- 在 Add stacks to stack set (將堆疊新增至堆疊集) 中，選擇 Deploy new stacks (部署新堆疊)。

7. 對於關聯組織單位區段，執行以下操作：

- a. 選擇與組織關聯，以使用根 OU。
- b. 選擇與組織單位關聯，以輸入要匯入之堆疊的父 OU ID。例如，如果 Stack 1 和 Stack 2 在 OU1 下方，並且 Stack 3 在 OU2 下方，則輸入 OU1 和 OU2。

8. 對於 Specify regions (指定區域) 和 Deployment options (部署選項)，檢視您的選擇。

接受您的設定並選擇 Next (下一步)。

9. 在 Review (檢閱) 頁面上，檢閱您的設定，然後選擇 Submit (提交)。

將服務受管堆疊匯入現有 StackSet (主控台)

選擇 StackSet 並識別您要匯入的堆疊。

將堆疊匯入現有 StackSet

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 從導覽窗格選擇 StackSets。
3. 選擇要匯入堆疊的 StackSet，然後從動作下拉式清單中選擇將堆疊新增至 StackSet。
4. 在設定部署選項頁面上進行以下操作：
 - a. 對於將堆疊新增至堆疊集，選擇將堆疊匯入堆疊集。
 - b. 在要匯入的 Stacks 下方，執行下列動作

- i. 對於 Stack ID (堆疊 ID)，輸入您的堆疊 ID。
- ii. 對於堆疊 URL，輸入 Amazon S3 URL。
- c. 在相關單位單位下，執行以下操作：
 - i. 選擇與組織關聯，以使用根 OU。
 - ii. 選擇與組織單位關聯，以輸入要匯入之堆疊的父 OU ID。例如，如果 Stack 1 和 Stack 2 在 OU1 下方，並且 Stack 3 在 OU2 下方，則輸入 OU1 和 OU2。

接受您的設定並選擇 Next (下一步)。

5. 檢閱 Specify overrides (指定覆寫) 頁面，然後選擇 Next (下一步)。
6. 確認並檢閱 Review (檢閱) 頁面並選擇 Submit (提交)。

將服務受管堆疊匯入 StackSet (AWS CLI)

建立 StackSet 後，您可以透過傳遞所匯入堆疊的堆疊 ID 來匯入堆疊。您也可以將 OU ID 清單傳遞至您要映射的 OU ID 清單。

CloudFormation 會在這些 OU 中匯入使用者提供的堆疊，並使用這些 OU 作為 StackSet 的部署目標。輸入中顯示的堆疊 ID 會對應至內部 OU ID 清單輸入中最接近的 OU。如果堆疊不屬於輸入清單中現有的 OU ID，則 AWS CLI 會傳回 `StackNotFoundException` 錯誤。

`import-stacks-to-stack-set` 操作可為 OU ID 輸入中的堆疊建立堆疊執行個體。下列 AWS CLI 範例使用 `import-stacks-to-stack-set` 操作將堆疊匯入 StackSet。

- 若要使用 `import-stacks-to-stack-sets` 操作，請指定您想匯入到堆疊集的 `stack-ids` 或 `stack-ids-url`。

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceMangedStackSet \  
  --stack-ids "arn:123456789012:us-east-1:Stack1" \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

```
aws cloudformation import-stacks-to-stack-set \  
  --stack-set-name ServiceMangedStackSet \  
  --stack-ids-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-name.json \  
  --organizational-unit-ids ou-examplerootid111-exampleouid111
```

 Note

`import-stacks-to-stack-sets` 操作會要求您至少指定一個組織單位 ID (OU ID)，以便將匯入的堆疊與該特定 OU 建立關聯。此操作不會為關聯 OU 中的其他成員帳戶建立堆疊執行個體。若要更新關聯 OU 的成員帳戶，請使用 `create-stack-instances` 或 `update-stack-instances`。

`create-stack-set` 利用使用者提供的範本 (無論是直接上傳或來自 Amazon S3)，為 OU 下的所有帳戶建立堆疊執行個體。下列 AWS CLI 範例使用 `create-stack-set` 操作將堆疊匯入新的 StackSet。

- 若要使用 `create-stack-set` 操作，請指定您的 StackSet 名稱，並將堆疊匯入新建立的 StackSet。

```
aws cloudformation create-stack-set \  
  --template-url https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/file-name.json \  
  --permission-model SERVICE_MANAGED \  
  --auto-deployment Enabled=true
```

還原堆疊匯入至 CloudFormation StackSets

若堆疊執行個體中出現非預期的變更，您可以還原堆疊執行個體匯入。

還原堆疊執行個體匯入時，CloudFormation 會從 StackSet 中刪除該堆疊執行個體，但會保留堆疊的資源。

若要還原堆疊匯入操作，請完成以下程序：

- 為想要在刪除堆疊執行個體後要保留的每個資源指定 Retain 的 DeletionPolicy 屬性。如需詳細資訊，請參閱[還原匯入操作](#)。
- 從 StackSet 刪除堆疊執行個體。如需詳細資訊，請參閱[從 CloudFormation StackSets 刪除堆疊](#)。
- 刪除您的 StackSet。如需詳細資訊，請參閱[Delete CloudFormation StackSets](#)。

使用 CloudFormation StackSets 的最佳實務

本節描述定義 StackSet 範本、建立或新增堆疊至 StackSet，或更新 StackSet 的最佳實務。

若您是 CloudFormation 新手，請檢閱 [CloudFormation 最佳實務](#) 主題，以取得更多建議，協助您更有效且安全地使用 CloudFormation。

主題

- [定義範本](#)
- [建立或將堆疊新增至 StackSet](#)
- [更新 StackSet 中的堆疊](#)

定義範本

- 定義您希望在多個區域內多個帳戶中標準化的範本。
- 當您建立範本時，請確認在相同帳戶內超過一個以上的區域中建立全域資源 (例如 IAM 角色和 Amazon S3 儲存貯體) 時沒有發生命名衝突。
- StackSet 具有一個單一範本及參數組。與 StackSet 關聯的所有帳戶內都會建立堆疊。當您編寫範本時，請讓其維持適當的精細程度，以在控制和標準化間取得良好的平衡。
- 我們建議您將您的範本存放在 Amazon S3 儲存貯體中。

建立或將堆疊新增至 StackSet

- 請在將大量的堆疊執行個體新增至您的 StackSet 前，先確認新增堆疊執行個體至您的初始 StackSet 可正常運作。
- 選擇適合您使用案例的部署 (推出) 選項。
 - 如需進行較保守的部署，請將 Maximum Concurrent Accounts (最大同時帳戶數) 設為 1，並將 Failure Tolerance (容錯能力) 設為 0。將您的最低影響區域設為 Region Order (區域順序) 清單中的第一位。從使用一個區域開始。
 - 如需進行較快速的部署，請視需要增加 Maximum Concurrent Accounts (最大同時帳戶數) 及 Failure Tolerance (容錯能力) 的值。
- StackSets 上的操作取決於涉及的堆疊執行個體數，並且可能會耗費大量的時間。

更新 StackSet 中的堆疊

- 在預設情況下，更新 StackSet 會更新所有堆疊執行個體。若您在兩個區域中分別具有 20 個帳戶，您便擁有 40 個堆疊執行個體，並且所有的堆疊執行個體都會在您更新 StackSet 時更新。

對於含有大量堆疊執行個體的 StackSets，若要測試更新版本的範本，我們建議先選擇性地在幾個測試帳戶中更新堆疊執行個體，再更新所有堆疊執行個體。

- 若要在更新 StackSet 內的個別堆疊時進行更細微的控制，請規劃建立多個 StackSets。
- 更新包含大量堆疊的 StackSet 可能需要耗費大量的時間。在此版本中，一個 StackSet 上一次只能進行一項操作。計劃您的更新，以免您無法在 StackSet 上執行其他操作。

CloudFormation StackSets 範本範例

本節所涵蓋的部分範例 CloudFormation 範本連結，有助於您在企業中善用 CloudFormation StackSets。本節列出的範本可支援 [AWS CloudTrail](#) 或 [AWS Config](#) 及其所包含的規則。

Important

作為允許 AWS Config 存取 Amazon S3 儲存貯體的安全最佳實務，強烈建議您使用 `AWS:SourceAccount` 條件限制儲存貯體政策中的存取。新範本會更新為具有 `AWS:SourceAccount`。如果您現有的儲存貯體政策未遵循此安全最佳作法，強烈建議您編輯該儲存貯體政策以納入此保護。這可確保 AWS Config 僅代表預期的使用者授予存取權。

Description	S3 連結
啟用 AWS CloudTrail	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnabledAWSCloudtrail.yml
啟用 AWS Config	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnabledAWSConfig.yml
AWS Config 使用中央記錄啟用	https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnabledAWSConfigForOrganizations.yml
跨 AWS 組織或特定 AWS 帳戶啟用 Amazon Data Lifecycle Manager 預設政策	• EBS 快照的預設政策 — https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/EnabledEBSSnapshotPolicy.yml

Description	S3 連結
	templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml 基於 EBS 的 AMI 預設原則 — https://s3.amazonaws.com/cloudformation-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml https://s3.amazonaws.com/cloudformation-templates-us-east-1/ConfigRuleEncryptedVolumes.yml
設定 AWS Config 規則以判斷 CloudTrail 是否已啟用	https://s3.amazonaws.com/cloudformation-templates-us-east-1/ConfigRuleCloudtrailEnabled.yml
設定 AWS Config 規則以判斷是否已啟用根 MFA	https://s3.amazonaws.com/cloudformation-templates-us-east-1/ConfigRuleRootAccountMFAEnabled.yml
設定 AWS Config 規則以判斷是否已連接 EIPs	https://s3.amazonaws.com/cloudformation-templates-us-east-1/ConfigRuleEipAttached.yml
設定 AWS Config 規則以判斷 EBS 磁碟區是否已加密	https://s3.amazonaws.com/cloudformation-templates-us-east-1/ConfigRuleEncryptedVolumes.yml

疑難排解 CloudFormation StackSets

本主題涵蓋部分常見的 StackSets 問題，以及這些問題的建議解決方案。

主題

- [堆疊操作失敗的常見原因](#)
- [重試失敗的堆疊建立或更新操作](#)

- [堆疊執行個體刪除失敗](#)
- [堆疊匯入操作失敗](#)
- [StackSets 操作的堆疊執行個體失敗計數](#)

堆疊操作失敗的常見原因

問題：堆疊操作失敗，且堆疊執行個體的状态為 OUTDATED。

原因：堆疊操作失敗可能有幾種常見原因。

- 目標帳戶沒有足夠的許可，無法建立範本中指定的資源。
- CloudFormation 範本可能有錯誤。在建立 StackSet 之前，請驗證 CloudFormation 中的範本並修正錯誤。
- 範本正在嘗試建立的全域資源 (如 S3 儲存貯體) 必須獨一無二，但該資源已存在。
- 指定的目標帳戶號碼並不存在；請檢查您在精靈的 Set deployment options (設定部署選項) 頁面中所指定之目標帳戶號碼。
- 管理員帳戶與目標帳戶沒有信任關係。
- 目標帳戶中的資源已達範本所指定的最大數量。舉例來說，目標帳戶可能已達到 IAM 角色數量的上限，但範本仍持續建立更多 IAM 角色。
- StackSet 中允許的堆疊數已達到上限，如需每個 StackSet 的堆疊數目上限，請參閱 [了解 CloudFormation 配額](#)。

解決方案：如需詳細資訊，以了解建立 StackSets 前目標與管理員帳戶要具備的許可，請參閱 [為管理員帳戶的所有使用者提供可以在所有目標帳戶中管理堆疊的許可](#) 小節。

重試失敗的堆疊建立或更新操作

問題：堆疊建立或更新失敗，且堆疊執行個體的状态為 OUTDATED。若要對堆疊建立或更新失敗的原因進行故障診斷，請開啟 CloudFormation 主控台以檢視堆疊事件；在建立操作失敗的情況下，這些事件的状态將為 DELETED，而在更新操作失敗的情況下，則會顯示 FAILED 狀態。請瀏覽堆疊事件，並找到 Status reason (狀態原因) 欄。透過 Status reason (狀態原因) 的數值，即可掌握堆疊操作失敗的原因。

若您將堆疊建立失敗的潛在原因修正完成，且準備好重試堆疊建立操作，則請執行下列步驟。

解決方案：執行下列步驟，進而重試堆疊操作。

1. 在主控台中，選取包含操作失敗堆疊的 StackSet。
2. 在 Actions (動作) 選單中，選擇 Edit StackSet details (編輯 StackSet 詳細資訊) 重試建立或更新堆疊。
3. 若要在指定範本頁面上使用相同的 CloudFormation 範本，請保留預設選項使用目前的範本。如果是因範本需要變更而導致堆疊操作失敗，且您想上傳修訂的範本，則請改為選擇 Upload a template to Amazon S3 (將範本上傳到 Amazon S3)，接著選擇 Browse (瀏覽) 並選取更新後的範本。完成修訂範本上傳作業時，選擇 Next (下一步)。
4. 若您沒有要在指定堆疊詳細資訊 頁面上變更任何範本特定的參數，請選擇下一步。
5. 在設定部署選項頁面上，您可以視需要變更並行帳戶數目上限和容錯能力的預設值。如需這些設定的詳細資訊，請參閱 [StackSet 操作選項](#)。
6. 在 Review (檢閱) 頁面上，檢視您的選項並勾選核取方塊，藉此確認所需的 IAM 功能。選擇提交。
7. 如果無法成功更新堆疊，請先解決任何阻礙堆疊建立的潛在原因，然後重複此程序。

堆疊執行個體刪除失敗

問題：無法刪除堆疊。

原因：如果任何堆疊啟用終止保護，將無法刪除該堆疊。

解決方案：請判斷堆疊是否啟用終止保護。如果堆疊已啟用終止保護，請先停用該功能，然後再次刪除堆疊執行個體。

堆疊匯入操作失敗

問題：堆疊匯入操作無法將現有堆疊匯入新的或現有的 StackSets。堆疊執行個體處於 INOPERABLE 狀態。

解決方案：完成下列任務，還原堆疊匯入操作。

1. 在設定期間使用從 StackSets 刪除堆疊選項並啟用 RetainStacks，然後繼續刪除 StackSet 中的堆疊執行個體。如需詳細資訊，請參閱 [從 CloudFormation StackSets 刪除堆疊](#)。
2. 您將看到 Stackset 的堆疊執行個體已更新，以便刪除 INOPERABLE 堆疊執行個體。
3. 根據匯入失敗錯誤修正堆疊執行個體，然後重試堆疊匯入操作。

StackSets 操作的堆疊執行個體失敗計數

堆疊執行個體失敗計數會在堆疊執行個體無法佈建或更新時提醒您。由於以下一個或多個原因，導致這些堆疊執行個體並未部署：

- 現有資源具備類似組態
- 缺少相依性，例如 AWS Identity and Access Management (IAM) 角色
- 其他衝突因素

如果您要使用並行數目上限進行部署，並行計數上限最多會比故障容錯計數多一。例如，如果容錯計數為 9，則最大並行計數不能超過 10。這樣一來，即使某些堆疊執行個體無法更新，操作仍會傳回 SUCCEEDED。新的堆疊執行個體失敗計數可讓您判斷操作是否僅因容錯計數設定為允許所有失敗而有條件地成功。

您可以使用 AWS 管理主控台、AWS SDK 或 AWS CLI 取得故障計數並篩選堆疊執行個體，以判斷需要重新部署哪些執行個體。

使用主控台

檢視失敗的堆疊執行個體數：

1. 開啟 [CloudFormation 主控台](#)，然後選擇 StackSet。
2. 選擇您的 StackSet，然後選擇 Operations (操作) 索引標籤。
3. 選擇 Status (狀態) 資料欄中的狀態，以檢視狀態的詳細資訊。您可以在狀態詳細資訊中找到特定操作的失敗堆疊執行個體數目。

檢視操作的堆疊執行個體的帳戶、區域和狀態：

1. 在狀態詳細資訊中，選擇失敗的堆疊執行個體計數。範例：Stack instances (堆疊執行個體)：<number of failed stack instances>。
2. 選擇面板標頭，展開側邊面板。側邊面板中的結果是選取的操作完成後堆疊執行個體的狀態。

檢視操作的目前堆疊執行個體詳細資訊：

1. 選擇 Stack Instances (堆疊執行個體) 索引標籤。

2. 依 Last operation ID (上次操作 ID) 篩選。結果是上次修改執行個體的操作的目前狀態和狀態原因。您可以將此篩選條件與 AWS 帳戶、AWS 區域、詳細狀態和漂移狀態合併使用，以取得更精確的搜尋結果。

使用 AWS CLI

若要取得失敗的堆疊執行個體數量，請呼叫 `describe-stack-set-operation` 或 `list-stack-set-operations` 並查看 `StatusDetails`。

```
aws cloudformation describe-stack-set-operation --stack-set-name ss1 \
  --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60
```

```
{
  "StackSetOperation": {
    "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
    "Action": "CREATE",
    "Status": "SUCCEEDED",
    "OperationPreferences": {
      "RegionOrder": [],
      "FailureToleranceCount": 10,
      "MaxConcurrentCount": 10
    },
    "AdministrationRoleARN": "arn:aws:iam::123456789012:role/
AWSCloudFormationStackSetAdministrationRole",
    "ExecutionRoleName": "AWSCloudFormationStackSetExecutionRole",
    "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
    "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
    "StatusDetails": {
      "FailedStackInstancesCount": 3
    }
  }
}
```

```
aws cloudformation list-stack-set-operations --stack-set-name ss1
```

```
{
  "Summaries": [
    {
      "OperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60",
```

```

    "Action": "CREATE",
    "Status": "SUCCEEDED",
    "CreationTimestamp": "2022-10-26T17:18:53.947000+00:00",
    "EndTimestamp": "2022-10-26T17:19:35.304000+00:00",
    "StatusDetails": {
      "FailedStackInstancesCount": 3
    },
    "OperationPreferences": {
      "RegionOrder": [],
      "FailureToleranceCount": 10,
      "MaxConcurrentCount": 10
    }
  }
]
}

```

若要取得特定操作的歷史概觀，請使用 `list-stack-set-operation-results` 檢視操作完成後每個堆疊執行個體的狀態和狀態原因。請見以下範例，找出 `Status` 和 `StatusReason`：

```

aws cloudformation list-stack-set-operation-results --stack-set-name ss1 \
  --operation-id 5550e62f-c822-4331-88fa-21c1d7bafc60 --
  filters Name=OPERATION_RESULT_STATUS,Values=FAILED

```

```

{
  "Summaries": [
    {
      "Account": "123456789012",
      "Region": "us-west-2",
      "Status": "FAILED",
      "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
      "AccountGateResult": {
        "Status": "SKIPPED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
      },
      "OrganizationalUnitId": ""
    },
    {
      "Account": "123456789012",
      "Region": "us-west-1",

```



```

        "Status": "FAILED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "AccountGateResult": {
            "Status": "SKIPPED",
            "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
        },
        "OrganizationalUnitId": ""
    },
    {
        "Account": "123456789012",
        "Region": "us-east-1",
        "Status": "FAILED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "AccountGateResult": {
            "Status": "SKIPPED",
            "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'."
        },
        "OrganizationalUnitId": ""
    }
]
}

```

使用 `list-stack-instances` 搭配 `DETAILED_STATUS` 和 `LAST_OPERATION_ID` 篩選條件，取得上次嘗試部署堆疊執行個體的操作中失敗的堆疊執行個體清單。請見範例中的 `--filters` 標記具有 `DETAILED_STATUS` 和 `LAST_OPERATION_ID`：

```

aws cloudformation list-stack-instances --stack-set-name ss1 \
  --filters Name=DETAILED_STATUS,Values=FAILED Name=LAST_OPERATION_ID,Values=5550e62f-
c822-4331-88fa-21c1d7bafc60

```

```

{
  "Summaries": [
    {
      "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",

```

```

        "Region": "us-east-1",
        "Account": "123456789012",
        "Status": "OUTDATED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    },
    {
        "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
        "Region": "us-west-1",
        "Account": "123456789012",
        "Status": "OUTDATED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    },
    {
        "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",
        "Region": "us-west-2",
        "Account": "123456789012",
        "Status": "OUTDATED",
        "StatusReason": "Account 123456789012 should have
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role
'AWSCloudFormationStackSetAdministrationRole'.",
        "StackInstanceStatus": {
            "DetailedStatus": "FAILED"
        },
        "OrganizationalUnitId": "",
        "DriftStatus": "NOT_CHECKED",
        "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"
    }
}
]

```

```
}
```

若要尋找上次修改堆疊執行個體的操作 ID，請使用 `list-stack-instances` 或 `describe-stack-instance` 來取得 `LastOperationId`：

```
aws cloudformation describe-stack-instance --stack-set-name ss1 \  
--stack-instance-account 123456789012 --stack-instance-region us-east-2
```

```
{  
  "StackInstance": {  
    "StackSetId": "ss1:9101ca57-49fc-4a61-a5a6-4c97b8adb08f",  
    "Region": "us-west-2",  
    "Account": "123456789012",  
    "ParameterOverrides": [],  
    "Status": "OUTDATED",  
    "StackInstanceStatus": {  
      "DetailedStatus": "FAILED"  
    },  
    "StatusReason": "Account 123456789012 should have  
'AWSCloudFormationStackSetExecutionRole' role with trust relationship to Role  
'AWSCloudFormationStackSetAdministrationRole'.",  
    "OrganizationalUnitId": "",  
    "DriftStatus": "NOT_CHECKED",  
    "LastOperationId": "5550e62f-c822-4331-88fa-21c1d7bafc60"  
  }  
}
```

使用 Git 同步功能，將堆疊與儲存在 Git 儲存庫中的來源程式碼同步

使用 Git 同步，您可以使用原始碼控制來管理 CloudFormation 堆疊。可以設定 CloudFormation 以監視 Git 儲存庫來執行此操作。系統會監視儲存庫是否有對兩個檔案的變更：

- 定義堆疊的 CloudFormation 範本檔案
- 包含設定堆疊之參數的堆疊部署檔案

使用 Git 同步功能，就可以使用提取請求和版本追蹤，從集中位置中設定、部署和更新 CloudFormation 堆疊。當您對範本或部署檔案遞交變更時，CloudFormation 會自動更新堆疊。如果使用提取請求，CloudFormation 可以在提取請求上留下註解，說明在實際更新堆疊之前要對堆疊進行哪些變更。不過，需要先啟用此功能。

Git 同步提供主控台介面，您可以使用它來連結至儲存庫、產生堆疊部署檔案、更新 CloudFormation 範本，以及將提取請求提交至儲存庫。Git 同步也提供了一個狀態儀表板，您可以用來監視、編輯和疑難排解作用中的 Git 同步堆疊部署。當您[建立堆疊](#)時，會透過 [CloudFormation 主控台](#)存取 Git 同步。也可以使用 CodeConnections 來存取 Git 同步。如需詳細資訊，請參閱《開發人員工具主控台使用者指南》中的[使用連結儲存庫的同步組態](#)。

Git 同步支援 [GitHub](#)、[GitHub Enterprise](#)、[GitLab](#)、[Bitbucket](#) 以及 [GitLab 自我管理的](#)儲存庫。

Note

Git 同步可在下列區域使用：美國東部 (維吉尼亞北部)、美國東部 (俄亥俄)、美國西部 (加利佛尼亞北部)、美國西部 (奧勒岡)、加拿大 (中部)、亞太區域 (孟買)、亞太區域 (東京)、亞太區域 (首爾)、亞太區域 (新加坡)、亞太區域 (雪梨)、歐洲 (愛爾蘭)、歐洲 (倫敦)、歐洲 (巴黎)、歐洲 (斯德哥爾摩)、歐洲 (法蘭克福)、歐洲 (米蘭) 和南美洲 (聖保羅)。

如需有關搭配使用 Git 同步與多帳戶策略的詳細資訊，請參閱下列部落格文章[使用 CloudFormation Git 同步來設定客戶帳戶中的資源](#)。

主題

- [Git 同步如何與 CloudFormation 搭配運作](#)
- [使用 Git 同步功能將堆疊同步至 Git 儲存庫的先決條件](#)

- [使用 Git 同步從儲存庫原始程式碼建立堆疊](#)
- [讓 CloudFormation 在提取請求中發佈堆疊變更的摘要](#)
- [Git 同步狀態儀表板](#)

Git 同步如何與 CloudFormation 搭配運作

本主題說明 Git 同步的運作方式，並介紹使用它所需的重要概念。

主題

- [Git 同步的運作方式](#)
- [提取請求的註解](#)
- [堆疊部署檔案](#)
- [CloudFormation 範本檔案](#)
- [範本定義儲存庫](#)

Git 同步的運作方式

若要使用 Git 同步，必須先使用 [CodeConnections](#) 服務將 Git 提供者連線至 CloudFormation。在本指南的程序中，透過 CodeConnections 主控台建立連線。或者，也可以使用 AWS CLI來建立連線。可以使用下列任何 Git 提供者：

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab 自我管理](#)

接下來，建立一個可定義堆疊的 CloudFormation 範本，並將其新增至您的儲存庫。此範本檔案會受到監控。CloudFormation 會在堆疊遞交變更時自動更新堆疊。

在 CloudFormation 主控台中，您會建立新堆疊，然後選擇從 Git 同步，以指示 CloudFormation 使用 Git 同步。您將指定要 CloudFormation 監控的儲存庫和分支，並在定義堆疊的儲存庫中指定 CloudFormation 範本。

在設定過程中，您可以從儲存庫提供自己的堆疊部署檔案，也可以讓 Git 同步為您生成一個檔案。堆疊部署檔案包含設定堆疊中資源的參數和值。此堆疊部署檔案會受到監控。CloudFormation 會在堆疊遞交變更時自動更新堆疊。

Git 同步會在您的儲存庫中建立一個提取請求，以便將您的堆疊與 CloudFormation 範本檔案和堆疊部署檔案同步。如果 Git 同步為您生成堆疊部署檔案，則檔案會透過 Git 同步提交到您的儲存庫。

然後，將提取請求合併至儲存庫，讓 CloudFormation 佈建堆疊、使用您的部署參數設定堆疊，並開始監控儲存庫以瞭解是否有任何變更。

從那時起，每當變更範本檔案或堆疊部署檔案並將其遞交至儲存庫時，CloudFormation 都會自動偵測變更。如果您的團隊使用提取請求，則團隊成員就可以在部署變更之前檢閱和核准變更。接受提取請求後，CloudFormation 會部署變更。

在 CloudFormation 主控台中，您可監視堆疊的 Git 同步組態的狀態，並查看套用至堆疊的遞交歷史記錄。主控台也提供重新設定 Git 同步和疑難排解問題的工具。

提取請求的註解

您可以選擇讓 CloudFormation 透過 CodeConnections 服務在提取請求中建立程式碼變更的摘要，方法是在主控台中開啟對提取請求啟用註解選項。提供提取請求中的變更摘要意味著團隊成員可以在合併提取請求之前輕鬆檢閱和了解提議修改的影響。如需詳細資訊，請參閱[讓 CloudFormation 在提取請求中發佈堆疊變更的摘要](#)。

堆疊部署檔案

堆疊部署檔案是 JavaScript 物件標記法 (JSON) 或 YAML 標準格式檔案，其中包含管理 CloudFormation 堆疊的參數和值。它受到是否有任何變更的監控。當檔案的變更遞交至儲存庫時，關聯的堆疊會自動更新。

堆疊部署檔案包含鍵值組及兩部字典：

- `template-file-path`

這是 CloudFormation 範本檔案的完整儲存庫路徑。範本檔案會針對與此部署檔案關聯的 CloudFormation 堆疊宣告資源。

- `parameters`

參數字典包含堆疊中設定資源的鍵值組。堆疊部署檔案最多可以有 50 個參數。

- tags

標籤字典包含選用鍵值組，您可以用來識別和分類堆疊中的資源。堆疊部署檔案最多可以有 50 個標籤。

您可以提供自己的堆疊部署檔案，或讓 Git 同步為您建立一個檔案，然後自動將提取要求提交至您的儲存庫。您可以透過編輯堆疊部署檔案並將變更遞交至儲存庫的方式來管理參數和標籤。

以下是 Git 同步堆疊部署檔案的範例：

```
template-file-path: fargate-srvs/my-stack-template.yaml

parameters:
  image: public.ecr.aws/lts/nginx:latest
  task_size: x-small
  max_capacity: 5
  port: 8080
  env: production
tags:
  cost-center: '123456'
  org: 'AWS'
```

CloudFormation 範本檔案

範本檔案包含組成 CloudFormation 堆疊 AWS 的資源宣告。運用 Git 同步，範本檔案會儲存在 Git 儲存庫中，並由堆疊部署檔案參照。您可以透過編輯範本檔案並將變更遞交至儲存庫的方式來管理堆疊。

如需詳細資訊，請參閱[使用 CloudFormation 範本](#)。

範本定義儲存庫

範本定義儲存庫是透過 Git 同步連結到 CloudFormation 的 Git 儲存庫。儲存庫會受到是否有對 CloudFormation 範本和堆疊部署檔案進行任何變更的監控。當您遞交對檔案的變更時，關聯的堆疊會自動更新。

Important

當您在 Git 同步主控台中設定範本定義儲存庫時，請從 Git 連線選取正確的儲存庫和分支。Git 同步只會監控已設定的儲存庫和分支是否有對 CloudFormation 範本和堆疊部署檔案的變更。

使用 Git 同步功能將堆疊同步至 Git 儲存庫的先決條件

將 CloudFormation 堆疊同步至 Git 儲存庫之前，請確認滿足下列要求。

主題

- [Git 儲存庫](#)
- [CloudFormation 範本](#)
- [Git 同步服務角色](#)
- [主控台使用者的 IAM 許可](#)

Git 儲存庫

必須在下列其中一個平台上託管 Git 儲存庫。

- [GitHub](#)
- [GitHub Enterprise](#)
- [GitLab](#)
- [Bitbucket](#)
- [GitLab 自我管理](#)

儲存庫可以是公有或私有。可以透過[連線主控台](#)將 Git 儲存庫連線至 CloudFormation。

CloudFormation 範本

Git 儲存庫必須擁有一個 [CloudFormation 範本檔案](#)，該範本檔案已簽入到您想要與 Git 同步連線的分支中。[堆疊部署檔案](#)將參考此範本。

Git 同步服務角色

Git 同步需要 IAM 角色。可以選擇在設定 Git 同步時為堆疊建立 IAM 角色，也可以使用現有角色。

Note

自動產生的 IAM 角色僅會將許可套用至為其產生角色的堆疊。若要重複使用自動產生的 IAM 角色，您必須編輯新堆疊的角色。

Git 同步服務角色所需的許可

您為 Git 同步提供的 IAM 角色需要下列許可。

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:ListChangeSets`
- `cloudformation:ValidateTemplate`
- `events:PutRule`
- `events:PutTargets`

Note

上述必要許可會自動新增至 Git 同步產生的 IAM 角色。

下列 IAM 角色範例包含 Git 同步的先決條件許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SyncToCloudFormation",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation>DeleteChangeSet",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStacks",
```

```

        "cloudformation:ExecuteChangeSet",
        "cloudformation:GetTemplate",
        "cloudformation:ListChangeSets",
        "cloudformation:ListStacks",
        "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
},
{
    "Sid": "PolicyForManagedRules",
    "Effect": "Allow",
    "Action": [
        "events:PutRule",
        "events:PutTargets"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "events:ManagedBy":
["cloudformation.sync.codeconnections.amazonaws.com"]
        }
    }
},
{
    "Sid": "PolicyForDescribingRule",
    "Effect": "Allow",
    "Action": "events:DescribeRule",
    "Resource": "*"
}
]
}

```

信任政策

當您建立角色以定義信任關係時，必須提供以下信任政策。

JSON

```

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

```

```
"Sid": "CfnGitSyncTrustPolicy",
"Effect": "Allow",
"Principal": {
  "Service": "cloudformation.sync.codeconnections.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
```

建議您使用 `aws:SourceArn` 和 `aws:SourceAccount` 條件金鑰，保護自己免受混淆代理人問題的困擾。來源帳戶是您的帳戶 ID，而來源 ARN 是允許 CloudFormation 連線至 Git 儲存庫的 [CodeConnections](#) 服務中的連線 ARN。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CfnGitSyncTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.sync.codeconnections.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codeconnections:us-east-1:123456789012:connection/EXAMPLE64-8aad-4d5d-8878-dfcab0bc441f"
        }
      }
    }
  ]
}
```

如需有關混淆代理人問題的詳細資訊，請參閱 [預防跨服務混淆代理人](#)。

主控台使用者的 IAM 許可

若要透過 CloudFormation 主控台成功設定 Git 同步，還必須透過 IAM 授予最終使用者許可。

需要下列 `codeconnections` 許可才能建立和管理 Git 儲存庫的連線。

- `codeconnections:CreateRepositoryLink`
- `codeconnections:CreateSyncConfiguration`
- `codeconnections>DeleteRepositoryLink`
- `codeconnections>DeleteSyncConfiguration`
- `codeconnections:GetRepositoryLink`
- `codeconnections:GetSyncConfiguration`
- `codeconnections:ListRepositoryLinks`
- `codeconnections:ListSyncConfigurations`
- `codeconnections:ListTagsForResource`
- `codeconnections:TagResource`
- `codeconnections:UntagResource`
- `codeconnections:UpdateRepositoryLink`
- `codeconnections:UpdateSyncBlocker`
- `codeconnections:UpdateSyncConfiguration`
- `codeconnections:UseConnection`

主控台使用者也必須具備下列 `cloudformation` 許可，才能在 Git 同步設定程序期間檢視和管理堆疊。

- `cloudformation:CreateChangeSet`
- `cloudformation>DeleteChangeSet`
- `cloudformation:DescribeChangeSet`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStacks`
- `cloudformation:ExecuteChangeSet`
- `cloudformation:GetTemplate`

- `cloudformation:ListChangeSets`
- `cloudformation:ListStacks`
- `cloudformation:ValidateTemplate`

Note

雖然僅使用主控台可能並不嚴格要求變更集許可

(`cloudformation:CreateChangeSet`、`cloudformation>DeleteChangeSet`、`cloudformation:UpdateChangeSet`)

但建議您啟用全堆疊檢查和管理功能。

下列範例 IAM 政策包含透過主控台設定 Git 同步所需的使用者許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeConnectionsPermissions",
      "Effect": "Allow",
      "Action": [
        "codeconnections:CreateRepositoryLink",
        "codeconnections:CreateSyncConfiguration",
        "codeconnections>DeleteRepositoryLink",
        "codeconnections>DeleteSyncConfiguration",
        "codeconnections:GetRepositoryLink",
        "codeconnections:GetSyncConfiguration",
        "codeconnections:ListRepositoryLinks",
        "codeconnections:ListSyncConfigurations",
        "codeconnections:ListTagsForResource",
        "codeconnections:TagResource",
        "codeconnections:UntagResource",
        "codeconnections:UpdateRepositoryLink",
        "codeconnections:UpdateSyncBlocker",
        "codeconnections:UpdateSyncConfiguration",
        "codeconnections:UseConnection",
        "codeconnections:CreateForcedTargetSync",
        "codeconnections:CreatePullRequestForResource"
      ]
    }
  ],
}
```

```
        "Resource": "*"
      },
      {
        "Sid": "CloudFormationConsolePermissions",
        "Effect": "Allow",
        "Action": [
          "cloudformation:CreateChangeSet",
          "cloudformation:DeleteChangeSet",
          "cloudformation:DescribeChangeSet",
          "cloudformation:DescribeStackEvents",
          "cloudformation:DescribeStacks",
          "cloudformation:ExecuteChangeSet",
          "cloudformation:GetTemplate",
          "cloudformation:ListChangeSets",
          "cloudformation:ListStacks",
          "cloudformation:ValidateTemplate"
        ],
        "Resource": "*"
      }
    ]
  }
}
```

Note

建立包含 `codeconnections:CreateForcedTargetSync` 和 `codeconnections:CreatePullRequestForResource` 許可的 IAM 政策時，可能會在 IAM 主控台中看到警告，指出這些動作不存在。可以忽略此警告，且仍會成功建立政策。即使 IAM 主控台無法辨識，某些 Git 同步操作仍需要這些許可。

使用 Git 同步從儲存庫原始程式碼建立堆疊

本主題說明如何建立 CloudFormation 堆疊，它透過 Git 同步功能同步至 Git 儲存庫。

Important

繼續操作之前，請先完成上一節中的所有[先決條件](#)。

從儲存庫原始程式碼建立堆疊

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇要在其中 AWS 區域 建立堆疊的。
3. 在堆疊頁面中，選擇建立堆疊，並選擇使用新資源 (標準)。
4. 在 Create stack (建立堆疊) 頁面上，執行下列動作：
 - a. 對於先決條件 - 準備範本，保持選取選擇現有範本。
 - b. 針對指定範本，選擇從 Git 同步，然後選擇下一步。
5. 在指定堆疊詳細資訊頁面的堆疊名稱中，輸入堆疊名稱。堆疊名稱可包含字母 (A-Z 和 a-z)、數字 (0-9) 和破折號 (-)。
6. 對於堆疊部署檔案，部署檔案建立：
 - 如果尚未建立堆疊部署檔案並將其新增至儲存庫，請選擇使用下列參數建立檔案，並將其放在我的儲存庫中。
 - 如果儲存庫中有堆疊部署檔案，請選擇我在儲存庫中提供自己的檔案。
7. 針對範本定義儲存庫，選取選擇連結的 Git 儲存庫以選擇已連結至 CloudFormation 的 Git 儲存庫，或選擇連結 Git 儲存庫以連結新的儲存庫。如果選擇連結 Git 儲存庫，請執行下列動作：
 - a. 在選取儲存庫提供者中，選擇下列其中一項：
 - GitHub
 - GitHub Enterprise Server
 - GitLab
 - Bitbucket
 - GitLab 自我管理
 - b. 在連線中，請從清單中選擇一個連線。如果連線清單中沒有出現任何選項，請選擇新增一個新連線，以轉至[連線主控台](#)並建立與儲存庫的連線。
8. 在儲存庫清單中，選取包含堆疊範本檔案的 Git 儲存庫。
9. 在分支清單中，選取您希望 Git 同步監視的分支。

 Note

Git 同步只會監控所選分支是否有對 CloudFormation 範本和堆疊部署檔案的變更。您想要套用至堆疊的任何變更，均必須遞交到此分支。

10. 若為部署檔案路徑，請指定完整路徑，包含儲存庫分支根目錄中的堆疊部署檔案名稱。

如果 CloudFormation 正在為您產生檔案，這就是檔案將在您的儲存庫中遞交的位置。如果您要提供檔案，這是檔案在儲存庫中的位置。

11. 新增 IAM 角色。IAM 角色包含 CloudFormation 從 Git 儲存庫同步堆疊所需的許可。選擇新的 IAM 角色來產生新角色，或選擇現有 IAM 角色，從 AWS 帳戶中選取現有角色。如果您選擇產生新角色，所需的許可會包含在角色中。

12. 啟用或關閉提取請求的註解：

- 若要讓 CloudFormation 在提取請求中發佈變更集資訊以進行堆疊更新，請保持啟用提取請求的註解開關開啟。
- 如果關閉此開關，當 repo 檔案更新時，CloudFormation 不會說明目前堆疊組態與提取請求中所提議變更之間的差異。

13. 若為範本檔案路徑，請為堆疊範本檔案指定儲存庫根目錄的完整路徑。

14. (選用) 若要指定堆疊參數，請選擇新增參數，為每個參數提供索引鍵和值，然後選擇下一步。如需詳細資訊，請參閱[堆疊部署檔案](#)。

例如，若要在堆疊部署檔案中指定 **port=8080** 參數，請執行下列動作：

- a. 選擇 Add parameter (新增參數)。
- b. 在 Key (索引鍵) 欄位，輸入 **port**。
- c. 針對數值，輸入 **8080**。

15. (選用) 若要指定堆疊標籤，選取新增新標籤，提供每個標籤的標籤鍵和值，然後選擇下一步。如需詳細資訊，請參閱[堆疊部署檔案](#)。

16. 選擇下一步繼續設定堆疊選項。如需組態堆疊選項的相關資訊，請參閱[設定堆疊選項](#)。

完成堆疊組態後，請選擇下一步以繼續。

17. 檢閱堆疊設定並確認下列事項：

- 堆疊範本已正確設定，並設為從 Git 同步。
- 部署檔案設定正確。

- 範本定義儲存庫已正確設定，尤其是選取了正確的儲存庫和分支名稱。
- 部署檔案的預覽是正確的，且包含預期的參數與值。

18. 選擇提交以建立堆疊。

選擇提交後，會在 Git 儲存庫中自動建立一個提取請求。必須將此提取請求合併到 Git 儲存庫中，才能建立堆疊。建立堆疊之後，CloudFormation 會監控 Git 儲存庫是否有變更。

從 Git 儲存庫中更新堆疊

若要更新堆疊，請直接變更範本檔案或 Git 儲存庫中的堆疊部署檔案。將變更遞交至受監控分支後，CloudFormation 會自動更新堆疊。如果使用提取請求，會在更新堆疊之前，在您的 Git 儲存庫中自動建立提取請求。必須將此提取請求合併到 Git 儲存庫中，才能更新堆疊。

在 CloudFormation 控制台中，可以選取堆疊並選擇 Git 同步索引標籤，以檢視堆疊狀態和同步事件的相關資訊。如需詳細資訊，請參閱[Git 同步狀態儀表板](#)。

讓 CloudFormation 在提取請求中發佈堆疊變更的摘要

本主題說明如何讓 CloudFormation 在 Git 儲存庫的提取請求中發佈堆疊變更的摘要。

透過啟用提取請求功能的註解，可以允許 CloudFormation 發佈註解，以便在儲存庫檔案更新時，描述目前堆疊組態與所提議變更之間的差異。此註解提供將新增、修改或刪除的資源摘要，可讓您在合併提取請求之前執行徹底的程式碼檢閱。

若要對新堆疊的提取請求啟用註解 (主控台)

建立堆疊時，請在指定堆疊詳細資訊頁面的範本定義儲存庫下，確定開啟啟用提取請求的註解開關。這是新堆疊的預設設定。

若要對現有堆疊的提取請求啟用註解 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇您在其中建立堆疊 AWS 區域 的。
3. 在堆疊頁面中，選擇要更新的執行中堆疊。
4. 選擇 Git 同步索引標籤，然後選擇編輯。
5. 在編輯 Git 同步設定頁面的範本定義儲存庫下，開啟啟用提取請求的註解開關。
6. 選擇 Update configuration (更新組態)。

Git 同步狀態儀表板

若要檢視 AWS CloudFormation Git 同步部署的狀態，請在 CloudFormation 主控台中選取堆疊，然後選擇 Git 同步索引標籤。

Git 同步索引標籤分為兩個面板：Git 同步狀態和最新同步事件。

Git 同步狀態

頂端面板提供下列有關堆疊的 Git 同步組態之資訊。

儲存庫

連接到 Git 同步的儲存庫連結

儲存庫提供者

儲存庫提供者的名稱

分支

Git 同步正在監視的分支名稱

部署檔案路徑

此堆疊之堆疊部署檔案的完整路徑

儲存庫同步狀態

最近一次同步操作的狀態

儲存庫同步狀態訊息

最近一次同步操作的訊息

Git 同步狀態

此堆疊的 Git 同步狀態

佈建狀態

佈建操作的狀態

在面板的右上角，使用下列按鈕來修改或更新 Git 同步：

- 編輯 - 編輯 Git 同步組態。
- 重試最新遞交 - 根據最新遞交到儲存庫來更新堆疊。

- 中斷連線 - 中斷 Git 同步與堆疊的連線。
- 重新整理 - 重新整理 Git 同步狀態面板。

最新同步事件

最新同步事件面板會顯示已套用至堆疊的遞交之資料表。

您可以使用每個直欄標題中的方向鍵來排序資料表。資料表可以根據下列內容以遞增或遞減順序來排序：

- 日期
- 遞交 ID
- 事件
- 日期
- 事件類型

支援的堆疊狀態

只有當堆疊處於下列其中一種支援狀態時，才能為堆疊設定 Git 同步：

- CREATE_COMPLETE
- UPDATE_COMPLETE
- UPDATE_ROLLBACK_COMPLETE
- IMPORT_COMPLETE
- IMPORT_ROLLBACK_COMPLETE

下列資料表包含具有描述的堆疊狀態代碼的完整清單：

堆疊狀態和選用的詳細狀態	Description
CREATE_COMPLETE	一或多個堆疊建立成功。
CREATE_IN_PROGRESS	一或多個堆疊建立中。
CREATE_FAILED	一或多個堆疊建立失敗。檢視堆疊事件，查看任何相關聯的錯誤訊息。建立失敗的可能原因包括權限不足，無法使用堆疊中

堆疊狀態和選用的詳細狀態	Description
	的所有資源、AWS 服務拒絕的參數值，或在資源建立期間逾時。
DELETE_COMPLETE	一或多個堆疊成功刪除。刪除的堆疊仍保留，90 天內仍可檢視。
DELETE_FAILED	一或多個堆疊刪除失敗。因為刪除失敗，所以您可能有一些資源仍在執行，但是您無法使用或更新堆疊。再次刪除堆疊，或檢視堆疊事件，查看任何相關聯的錯誤訊息。
DELETE_IN_PROGRESS	一或多個堆疊移除中。
REVIEW_IN_PROGRESS	一或多個堆疊建立中，這些堆疊具有預期的 StackId，但沒有任何範本或資源。  Important 針對最大可能堆疊數計數有此狀態碼的堆疊。
ROLLBACK_COMPLETE	在堆疊建立失敗或明確取消建立堆疊之後，成功移除一或多個堆疊。堆疊返回到之前的工作狀態。在建立堆疊操作期間建立的所有資源都予以刪除。 此狀態只會在堆疊建立失敗後出現。此狀態表示來自所建立之部分堆疊的所有操作均已適當清除。在此狀態時，只能執行刪除操作。
ROLLBACK_FAILED	在堆疊建立失敗或明確取消建立堆疊之後，移除一或多個堆疊失敗。刪除堆疊，或檢視堆疊事件，查看任何相關聯的錯誤訊息。
ROLLBACK_IN_PROGRESS	在堆疊建立失敗或明確取消建立堆疊之後，持續移除一或多個堆疊。
UPDATE_COMPLETE	一或多個堆疊更新成功。

堆疊狀態和選用的詳細狀態	Description
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	成功更新堆疊之後，持續移除一或多個堆疊的舊資源。針對需要替換資源的堆疊更新，CloudFormation 會先建立新資源，然後刪除舊資源，藉以減少任何堆疊中斷。在此狀態下，堆疊已更新且可用，但 CloudFormation 仍會刪除舊資源。
UPDATE_FAILED	一或多個堆疊更新不成功。檢視堆疊事件，查看任何相關聯的錯誤訊息。
UPDATE_IN_PROGRESS	持續更新一或多個堆疊。
UPDATE_ROLLBACK_COMPLETE	堆疊更新失敗之後，成功將一或多個堆疊返回到之前的運作狀態。
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	更新堆疊失敗之後，持續移除一或多個堆疊的新資源。在此狀態下，堆疊已復原到之前的運作狀態且可使用，但 CloudFormation 仍將刪除在堆疊更新期間建立的任何新資源。
UPDATE_ROLLBACK_FAILED	堆疊更新失敗之後，將一或多個堆疊返回到之前的運作狀態失敗。在此狀態時，您可以刪除堆疊或 繼續復原 。您可能需要先修正錯誤，您的堆疊才能回到運作狀態。或者，您可以聯絡 將堆疊 支援 還原為可用狀態。
UPDATE_ROLLBACK_IN_PROGRESS	堆疊更新失敗之後，持續將一或多個堆疊返回到之前的運作狀態。
IMPORT_IN_PROGRESS	匯入操作正在進行。
IMPORT_COMPLETE	已針對支援 resource import 堆疊中的所有資源順利完成匯入操作。
IMPORT_ROLLBACK_IN_PROGRESS	匯入將復原回先前的範本組態。
IMPORT_ROLLBACK_FAILED	堆疊中至少一個資源的匯入復原操作失敗。成功匯入資源 CloudFormation 的結果將可提供使用。

堆疊狀態和選用的詳細狀態	Description
IMPORT_ROLLBACK_COMPLETE	匯入已成功復原回先前的範本組態。

使用 CloudFormation 登錄檔管理擴充功能

CloudFormation 登錄檔做為集中式中樞，用於管理可整合到中 CloudFormation 範本的延伸項目 AWS 帳戶。延伸模組包括來自和第三方發佈者的資源類型、模組 AWS 和勾點，以及您自己的自訂延伸模組。登錄檔可讓您更輕鬆地探索和佈建 CloudFormation 範本中的擴充功能，方式與您使用 AWS 提供的資源相同。

本節說明如何在您的帳戶中使用 CloudFormation 登錄檔啟用第三方擴充功能，其中包括：

- 啟用公有擴充功能
- 註冊並啟用私有擴充功能

主題

- [相關文件](#)
- [CloudFormation 登錄檔概念](#)
- [在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)
- [從 CloudFormation 登錄檔使用第三方公有擴充功能](#)
- [使用已與您共用的第三方私有擴充功能](#)
- [編輯您帳戶中擴充功能的組態資料](#)
- [在中記錄資源類型 AWS Config](#)

相關文件

若您是有興趣建立自己的擴充功能的開發人員，請參閱下列文件：

- 《CloudFormation 命令列介面使用者指南》中的[使用 CloudFormation CLI 開發模組](#)
- CloudFormation 命令列介面使用者指南中的[使用 CloudFormation CLI 立資源類型](#)
- 《CloudFormation Hooks 使用者指南》中的[使用 CloudFormation CLI 開發自訂勾點](#)

此外，CloudFormation 登錄檔中所有可用的可佈建 AWS 資源類型都可以與搭配使用 AWS 雲端控制 API，其屬性和在標準 JSON 結構描述中定義的屬性。如需詳細資訊，請參閱《[Cloud Control API 使用者指南](#)》。使用 Cloud Control API 對 AWS 資源執行 CRUDL（建立、讀取、更新、刪除、列出）操作時，您只能對自己的 AWS 資源執行此操作 AWS 帳戶。

CloudFormation 登錄檔概念

本主題說明核心概念，協助您了解並開始使用 CloudFormation 登錄檔。

擴充功能類型

CloudFormation 登錄檔提供下列擴充功能類型：

勾點

勾點是驗證檢查，會在 CloudFormation 建立、更新或刪除您的堆疊或特定資源前進行檢查。勾點也可在建立變更集操作期間調用。其提供一種機制，透過依據特定需求驗證資源組態，強制執行組織標準和最佳實務。若勾點偵測到任何不符合其邏輯的組態，可發出警告或使佈建程序失敗，以防止部署不符合規範的資源。如需詳細資訊，請參閱《[CloudFormation Hooks 使用者指南](#)》。

如需有關如何透過 CloudFormation 主控台設定勾點的文件，請參閱《CloudFormation Hooks 使用者指南》的下列章節：

- [AWS Control Tower 做為勾點的主動控制](#)
- [Guard 勾點](#)
- [Lambda 勾點](#)

模組

模組是可重複使用的資源組態，可跨多個 CloudFormation 堆疊範本納入使用。其將複雜或常用的資源組態封裝為可重複使用的元件，簡化 CloudFormation 範本的建立和維護作業。這有助於在您組織的基礎設施部署中實現一致性和標準化。

資源類型

資源類型可讓您建模和自動化 CloudFormation 原生不支援的第三方資源或自訂資源。透過開發資源類型，您可以擴充 CloudFormation 的功能，以佈建和管理來自各種第三方服務的資源。

公有擴充功能

公有擴充功能是在登錄檔中公開發佈供所有人使用的 CloudFormation 擴充功能 AWS 帳戶。其中包含：

- AWS 延伸模組 – 由發佈的延伸模組一律 AWS 公開，並依預設啟用，因此您不需要在帳戶中使用這些延伸模組之前採取任何動作。會 AWS 控制這些延伸模組的版本控制，因此您一律使用最新的可用版本。

- 第三方公有擴充功能 – 這些是可供 AWS 以外的發佈者普遍使用的擴充功能。若要使用第三方公有擴充功能，您必須先在帳戶和區域中啟用它。

已啟用的擴充功能

您特定 中的 CloudFormation 登錄檔 AWS 帳戶 包含三種類型的已啟動擴充功能：

- AWS 擴充功能 – 所有 AWS 公有擴充功能都會自動啟用。
- 已啟用的第三方擴充功能 – 這些是您為帳戶和區域啟用的第三方擴充功能之本機複本。您啟用第三方公有擴充功能時，CloudFormation 會在您的帳戶登錄檔中建立本機複本。如需詳細資訊，請參閱 [從 CloudFormation 登錄檔使用第三方公有擴充功能](#)。
- 已註冊的私有擴充功能 – 這些是未在公有 CloudFormation 登錄檔中列出的私有擴充功能。這些可以是您自行建立的擴充功能，也可以是組織或其他第三方與您共用的擴充功能。若要在帳戶中使用這種私有擴充功能，您必須先註冊該擴充功能。註冊擴充功能會將其複本上傳到帳戶中的 CloudFormation 登錄檔並啟用它。如需詳細資訊，請參閱 [使用已與您共用的第三方私有擴充功能](#)。

在您的帳戶中使用私人註冊的擴充功能和第三方發行者的已啟用公有擴充功能，就如同在沙盒環境中進行使用。擴充功能透過版本控制進行管理，因此其佈建行為與特定版本綁定。因此，這些擴充功能的運作方式與公有擴充功能相同，遵循相同的版本特定規則。

Note

私有註冊擴充功能和已啟用第三方公有擴充功能可實作在建立、讀取、更新、列出和刪除操作期間執行的事件處理常式。在您的 CloudFormation 堆疊中使用這些擴充功能，除了建立資源的相關費用外，可能還會向您的帳戶收取費用。如需詳細資訊，請參閱 [CloudFormation 定價](#)。

在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能

若要在 CloudFormation 登錄檔中檢視可用和已啟用的擴充功能，您可以使用 AWS 管理主控台 或 AWS CLI。

檢視可用和已啟用擴充功能 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。

2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 從瀏覽窗格登錄檔下，選擇您要檢視的擴充功能類別：
 - 公有擴充功能顯示帳戶中可用的公有擴充功能。
 1. 針對篩選條件、擴充功能類型，選擇您的擴充功能類型：資源類型、模組或勾點。
 2. 針對篩選條件、發布者，選擇發布者：AWS 或第三方。
 - 已啟用的擴充功能顯示帳戶中已啟用的公有和私有擴充功能。
 1. 選擇擴充功能類型：Resource types (資源類型)、Modules (模組) 或 Hooks (掛鉤)。
 2. 使用篩選條件下拉式選單進一步選擇要檢視的擴充功能：
 - AWS – 列出 發佈的擴充功能 AWS。預設 AWS 會啟用 發佈的延伸模組。
 - 第三方 – 列出 AWS 您在此帳戶中啟用之發佈者以外的任何公開延伸。
 - Registered (已註冊) – 列出此帳戶中已啟用的任何私有擴充功能。
 - Publisher (發佈者) 顯示您使用此帳戶發佈的任何公有擴充功能。
4. 搜尋或選擇擴充功能名稱以檢視擴充功能詳細資訊。

檢視可用和已啟用擴充功能 (AWS CLI)

使用 [list-types](#) 命令。

從 CloudFormation 登錄檔使用第三方公有擴充功能

若要在範本中使用第三方公有擴充功能，您必須先為要在其中使用它的帳戶和區域啟用該擴充功能。啟用擴充功能以便用於啟用該擴充功能之帳戶和區域中的堆疊操作。

啟用第三方公有擴充功能時，CloudFormation 會在帳戶的擴充功能登錄檔中為已啟用之擴充功能建立項目，作為私有擴充功能。這使您可以設定該擴充功能包含的所有組態屬性。組態屬性會定義如何為指定 AWS 帳戶 和 區域設定延伸模組。

除了設定組態屬性外，您還可以透過下列方式自訂擴充功能：

- 指定 CloudFormation 用來啟動擴充功能的執行角色，以及設定擴充功能的記錄。
- 指定新的次要或修補程式版本可用時是否自動更新擴充功能。
- 指定要使用的別名，而不是第三方公有擴充功能名稱。這有助於避免第三方擴充功能之間的命名衝突。

主題

- [為公有擴充功能存取設定具有 IAM 許可和信任政策的執行角色](#)
- [自動使用擴充功能的新版本](#)
- [使用別名來稱呼擴充功能](#)
- [使用公有擴充功能的常用 AWS CLI 命令](#)
- [在您的帳戶中啟用第三方公有擴充功能](#)
- [在您的帳戶中更新公有第三方擴充功能](#)
- [停用帳戶中的第三方公有擴充功能](#)

為公有擴充功能存取設定具有 IAM 許可和信任政策的執行角色

當您從 CloudFormation 登錄檔啟用公有擴充功能時，您可以提供執行角色，讓 CloudFormation 擁有在 AWS 帳戶和區域中叫用該擴充功能的必要許可。

在擴充功能結構描述的處理常式區段中定義執行角色的必要許可。您必須建立一個 IAM 政策，授予該擴充功能所需的特定許可，並將其附加至執行角色。

除了許可政策外，執行角色還必須具備信任政策，允許 CloudFormation 擔任該角色。遵循《IAM 使用者指南》中的[使用自訂信任政策建立角色](#)的指引，建立具有自訂信任政策的角色。

信任關係

以下顯示您可使用的範例信任政策。

您可以選擇性地限制跨服務混淆代理人預防的許可範圍，方法是使用一或多個全域條件內容索引鍵搭配 Condition 欄位。如需詳細資訊，請參閱[預防跨服務混淆代理人](#)。

- 將 `aws:SourceAccount` 值設定為您的帳戶 ID。
- 將 `aws:SourceArn` 值設定為擴充功能的 ARN。

範例信任政策 1

以下為資源類型擴充功能的範例 IAM 角色信任政策。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "resources.cloudformation.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:cloudformation:us-
west-2:123456789012:type/resource/Organization-Service-Resource"
      }
    }
  }
]
```

範例信任政策 2

以下為勾點擴充功能的範例 IAM 角色信任政策。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com",
          "hooks.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```
    },  
    "ArnLike": {  
        "aws:SourceArn": "arn:aws:cloudformation:us-  
west-2:123456789012:type/hook/Organization-Service-Hook"  
    }  
}  
}  
]  
}
```

自動使用擴充功能的新版本

啟用擴充功能時，您也可以指定要使用最新次要版本的擴充功能類型。發佈者在已啟用的擴充功能上發佈新版本時，擴充功能類型就會更新次要版本。

例如，下次您使用包含該擴充功能的範本執行堆疊操作 (例如建立或更新堆疊) 時，CloudFormation 會使用新的次要版本。

自動或手動更新為新的擴充功能版本，不會影響堆疊中已佈建的任何擴充功能執行個體。

CloudFormation 會將擴充功能的主要版本更新視為可能包含重大變更，因此需要您手動更新為擴充功能的新主要版本。

依預設，發佈的延伸 AWS 項目會針對可用的所有帳戶和區域啟用，並一律使用每個 中可用的最新版本 AWS 區域。

Important

由於您可以控制擴充功能是否以及何時更新為帳戶中的最新版本，最終可能會在不同帳戶和區域部署相同擴充功能的不同版本。

在這些帳戶和區域中使用包含該擴充功能的相同範本時，這可能會產生未預期的結果。

使用別名來稱呼擴充功能

您無法在指定 AWS 帳戶 和區域中使用指定名稱啟用一個以上的擴充功能。由於不同的發佈者可能會提供具有相同擴充功能名稱的公有擴充功能，CloudFormation 讓您可以為啟用的任何第三方公有擴充功能指定別名。

如果您為擴充功能指定別名，CloudFormation 會將別名視為此帳戶和區域內的擴充功能類型名稱。您必須使用別名來參考範本、API 呼叫和 CloudFormation 主控台中的擴充功能。

在指定帳戶和區域中的擴充功能別名必須是唯一的。您可以使用不同類型名稱別名，在同一帳戶和區域中多次啟用相同的公有資源。

Important

雖然擴充功能別名只需要在指定帳戶和區域中為唯一，但我們強烈建議使用者不要為跨帳戶和區域的不同第三方公有擴充功能指派相同的別名。使用包含跨多個帳戶或區域之擴充功能別名的範本時，這樣做可能會產生未預期的結果。

使用公有擴充功能的常用 AWS CLI 命令

使用公有擴充功能的常用命令包括：

- [activate-type](#)：在您的帳戶中啟用公有第三方模組或資源類型。
- [set-type-configuration](#)：用於指定您帳戶中某個擴充功能的組態資料，以及停用和啟用勾點。
- [list-types](#)：列出您帳戶中的擴充功能。
- [describe-type](#)：傳回特定擴充功能或特定擴充功能版本的詳細資訊，包括目前的組態資料。
- [set-type-default-version](#)：指定哪個版本的擴充功能是預設版本。
- [deactivate-type](#)：在您帳戶中停用先前啟用的公有第三方模組或資源類型。

在您的帳戶中啟用第三方公有擴充功能

下列主題說明如何在您的帳戶中啟用第三方公有擴充功能，啟用後即可在該帳戶及啟用所在的區域中使用。

Note

繼續之前，請確認您已建立將用於此擴充功能的 [IAM 角色](#)。

主題

- [啟用公有擴充功能 \(主控台\)](#)

- [啟用公有擴充功能 \(AWS CLI\)](#)

啟用公有擴充功能 (主控台)

請依照本節中的步驟，使用主控台執行下列操作：

- 啟用第三方公有擴充功能
- 為您的帳戶指定額外的擴充功能組態資料

啟用公有擴充功能，以便在帳戶中使用

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在導覽窗格的登錄檔下方，選擇公有擴充功能。
4. 使用篩選條件選擇擴充功能類型，然後選擇第三方。(AWS 發佈的擴充功能預設為已啟用。)
5. 選擇擴充功能，然後選擇啟用。

如果擴充功能有多個版本，您可以使用版本選單選擇要啟用的擴充功能版本。預設為最新版本。

6. 對於擴充功能名稱，您可保持選取使用預設值，或選擇覆寫預設值，然後輸入要與此擴充功能搭配使用的擴充功能類型別名。別名必須遵循擴充功能類型的推薦格式。如需詳細資訊，請參閱[使用別名來稱呼擴充功能](#)。
7. 若您要啟用的擴充功能是勾點或資源類型，請針對執行角色 ARN，指定 CloudFormation 在調用擴充功能時要扮演的 IAM 角色。如需詳細資訊，請參閱[為公有擴充功能存取設定具有 IAM 許可和信任政策的執行角色](#)。
8. 對於記錄設定，指定擴充功能的記錄組態資訊 (若需要)。例如：

```
{
  "logRoleArn": "arn:aws:iam::account:role/rolename",
  "logGroupName": "log-group-name"
}
```

記錄組態資訊不是必需的，但建議作為除錯用途使用。若要使用帶勾點的記錄組態，請新增與指定執行角色相同的信任政策，以便日誌角色可以將日誌寫入您的日誌群組。

logRoleArn 和 logGroupName 金鑰名稱不區分大小寫。

9. 針對版本控制、自動更新，選擇接收更新的方式。

- On (打開) – 自動更新為最新次要版本。主要版本需手動更新。
- Off (關閉) – 永不自動更新為最新版本。所有版本需手動更新。

如需詳細資訊，請參閱[自動使用擴充功能的新版本](#)。

如果擴充功能需要其他組態，您可以選擇現在指定組態資料，或在啟用擴充功能之後指定。

Important

如果您要啟用的擴充功能是勾點，則需要此步驟。您必須為 `HookInvocationStatus` 屬性指定 `ENABLED`。此操作將啟用勾點結構描述 `properties` 區段中定義的掛鉤屬性。有關詳細資訊，請參閱《CloudFormation 勾點使用者指南》中的[勾點組態結構描述語法參考](#)。

指定組態資料

1. 針對組態，選擇立即設定，然後選擇啟用擴充功能。

CloudFormation 會顯示 `Configure extension` (設定擴充功能) 頁面。若要檢視擴充功能的目前組態結構描述，請確保已啟用 `View configuration schema` (檢視組態結構描述)。

2. 在組態 JSON 文字方塊中，輸入 JSON 字串，代表您要為此擴充功能指定的組態資料。您指定的 JSON 必須根據擴充功能的組態結構描述進行驗證。
3. 選擇設定擴充功能。

若您偏好在啟用後再設定擴充功能，可跳過此步驟，稍後再提供組態資料。

1. 針對組態，選擇稍後設定，然後選擇啟用擴充功能。
2. 擴充功能啟用後，您可從已啟用擴充功能頁面進入該擴充功能，並提供組態資料進行設定。

啟用公有擴充功能 (AWS CLI)

請依照本節中的步驟使用 AWS CLI 來：

- 啟用第三方公有擴充功能
- 為您的帳戶指定額外的擴充功能組態資料

啟用公有勾點

在您的帳戶中啟用勾點，即表示授權該勾點使用您 AWS 帳戶中定義的許可。CloudFormation 會在將您的許可傳遞給勾點之前，移除非必要的許可。CloudFormation 建議客戶或勾點使用者在帳戶中啟用勾點之前，先檢閱勾點許可，並了解勾點被允許使用的許可範圍。

啟用公有勾點，以便在帳戶中使用 (AWS CLI)

1. 取得勾點的 ARN 並儲存。您可以使用 AWS 管理主控台 或 取得勾點的 ARN AWS CLI。如需更多資訊，請參閱[在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)。

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. 若要指定組態資料，請使用 [set-type-configuration](#) 命令。您為 `--configuration` 傳遞的 JSON 必須根據勾點的組態結構描述進行驗證。若要為所有堆疊操作啟用勾點，您必須在 `HookConfiguration` 區段將 `HookInvocationStatus` 屬性設定為 `ENABLED`。

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

如需 `HookConfiguration` 組態選項的詳細資訊，請參閱《CloudFormation 勾點使用者指南》中的[勾點組態結構描述語法參考](#)。

啟用公有模組和資源類型

啟用公有擴充功能，以便在帳戶中使用 (AWS CLI)

- 使用 [activate-type](#) 命令啟用擴充功能，並指定是否在發佈擴充功能的新次要版本時自動更新擴充功能。

下列範例指定要為此帳戶啟用之公有擴充功能的公有 Amazon Resource Name (ARN)。此外，其還指定發佈新次要版本時，CloudFormation 將更新擴充功能。

```
aws cloudformation activate-type \  
  --public-type-arn public_extension_ARN \  
  --execution-role-arn arn:aws:iam::123456789012:role/my-execution-role \  
  --auto-update true --region us-west-2
```

此命令會傳回已啟用擴充功能的 ARN。

```
{
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example"
}
```

更新公有擴充功能的版本 (AWS CLI)

使用 [activate-type](#) 再次啟用擴充功能。

使用 `--version-bump` 選項指定是否將擴充功能更新為最新 MAJOR 版本、最新 MINOR 版本。

```
aws cloudformation activate-type --type RESOURCE \
  --type-name Example::Test::1234567890abcdef0 \
  --type-name-alias Example::Test::Alias \
  --version-bump MAJOR --region us-west-2
```

在您的帳戶中更新公有第三方擴充功能

啟用第三方公有擴充功能後，您可從自己的帳戶更新大部分擴充功能詳細資訊。

在您的帳戶中更新公有擴充功能 (主控台)

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在導覽窗格的登錄檔下方，選擇已啟用的擴充功能。
4. 找到要更新的擴充功能並選取。如需詳細資訊，請參閱[在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)。
5. 在動作選單中，選擇編輯，然後選擇適當的編輯選項：
 - 若要更新組態結構描述，請參閱 [編輯您帳戶中擴充功能的組態資料](#)。
 - 若要啟用或停用自動更新：
 1. 選擇編輯自動更新。
 2. 選擇開啟或關閉，然後選擇儲存。如需詳細資訊，請參閱[自動使用擴充功能的新版本](#)。
 - 若要更新執行角色：

1. 選擇編輯執行角色。
 2. 指定您想要 CloudFormation 調用此擴充功能時使用的 IAM 角色 ARN，然後選擇儲存。如需詳細資訊，請參閱[為公有擴充功能存取設定具有 IAM 許可和信任政策的執行角色](#)。
- 若要更新記錄組態：
 1. 選擇編輯記錄組態。
 2. 編輯記錄組態 JSON，然後選擇儲存。

停用帳戶中的第三方公有擴充功能

當您不再需要某個已啟用的第三方公有擴充功能時，可透過下列程序在帳戶中停用。

主題

- [在您的帳戶中停用公有擴充功能 \(主控台\)](#)
- [停用帳戶中的公有擴充功能 \(AWS CLI\)](#)
- [停用帳戶中的勾點 \(AWS CLI\)](#)

在您的帳戶中停用公有擴充功能 (主控台)

停用帳戶中的公有擴充功能

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 在導覽窗格的登錄檔下方，選擇已啟用的擴充功能。
4. 找到要停用更新的擴充功能並選取。如需詳細資訊，請參閱[在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)。
5. 在動作選單中，選擇停用。
6. 選擇停用。

停用帳戶中的公有擴充功能 (AWS CLI)

使用下列 [deactivate-type](#) 命令。

```
aws cloudformation deactivate-type --type MODULE \
```

```
--type-name Example::Test::Type::MODULE \  
--region us-west-2
```

停用帳戶中的勾點 (AWS CLI)

停用勾點可防止勾點在 中執行，AWS 帳戶 而不會將其移除。

使用 [set-type-configuration](#) 命令，並將 HookInvocationStatus 指定為 DISABLED 以停用勾點。

下列範例指定要停用之勾點的 AWS 區域 和 Amazon Resource Name (ARN)。

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
{"HookInvocationStatus": "DISABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn "arn:aws:cloudformation:us-west-2:123456789012:type/hook/MyTestHook" --  
region us-west-2
```

如需詳細資訊，請參閱[CloudFormation 《勾點使用者指南》中的停用和啟用 CloudFormation 勾點](#)。

使用已與您共用的第三方私有擴充功能

若要使用已與您共用的第三方私有擴充功能，您必須先在希望使用這些擴充功能的帳戶及區域中，向 CloudFormation 註冊這些資源提供者。註冊擴充功能會將其複本上傳到帳戶中的 CloudFormation 登錄檔並啟用它。註冊私有擴充功能後，該擴充功能會出現在該 AWS 帳戶 和 區域的 CloudFormation 登錄檔中，您可以在堆疊範本中使用它。

主題

- [註冊私有擴充功能的第三方 IAM 許可](#)
- [使用私有擴充功能的常用 AWS CLI 命令](#)
- [在您的帳戶中註冊第三方私有擴充功能](#)
- [從帳戶移除第三方私有擴充功能](#)

註冊私有擴充功能的第三方 IAM 許可

作為註冊私有擴充功能的一部分，您可能需要指定一個包含擴充功能專案套件的 Amazon S3 儲存貯體。此套件包含要註冊的擴充功能所需之任何來源檔案。註冊擴充功能的使用者必須能夠存取 Amazon S3 儲存貯體中的專案套件。為此，使用者必須具備該擴充功能套件的 [GetObject](#) 許可。

無論您是使用的 [register-type](#) 命令 AWS CLI，還是 CloudFormation CLI 的 [submit](#) 命令，都是如此。

如需詳細資訊，請參閱服務授權參考中的 [Amazon S3 的動作、資源和條件索引鍵](#)。

使用私有擴充功能的常用 AWS CLI 命令

處理私有擴充功能的常用命令包括：

- [register-type](#)：在您的帳戶中註冊私有擴充功能。
- [describe-type-registration](#)：傳回註冊請求的目前狀態。
- [list-types](#)：列出您帳戶中的擴充功能。
- [describe-type](#)：傳回特定擴充功能或特定擴充功能版本的詳細資訊，包括目前的組態資料。
- [set-type-configuration](#)：用於指定您帳戶中某個擴充功能的組態資料，以及停用和啟用勾點。
- [set-type-default-version](#)：指定哪個版本的擴充功能是預設版本。
- [deregister-type](#)：從您的帳戶中移除私有擴充功能或擴充功能版本。

在您的帳戶中註冊第三方私有擴充功能

本主題涵蓋註冊與您共用的第三方私有擴充功能的步驟，使其可在您的帳戶中使用。

Note

在繼續之前，請確認您擁有註冊私有擴充功能所需的 [IAM 許可](#)。

註冊與您共用的私有擴充功能 (AWS CLI)

1. 找出包含您想在帳戶中註冊之私有擴充功能專案套件的 Amazon S3 儲存貯體。
2. 使用 [register-type](#) 命令在帳戶中註冊擴充功能。

例如，以下命令會在指定的 AWS 帳戶中註冊 `My::Resource::Example` 資源類型：

```
aws cloudformation register-type --type RESOURCE \  
  --type-name My::Resource::Example \  
  --schema-handler-package [s3 object path] --region us-west-2
```

`RegisterType` 是一種非同步操作，會傳回您可以用來追蹤註冊請求進度的註冊權杖。

```
{
```

```
"RegistrationToken": "f5525280-104e-4d35-bef5-8f1fexample"
}
```

如果您的延伸模組呼叫 AWS APIs 做為其功能的一部分，您必須建立 IAM 執行角色，其中包含呼叫這些 AWS APIs 的必要許可，並在您的帳戶中佈建該執行角色。然後您可以使用 `--execution-role-arn` 選項來指定此執行角色。CloudFormation 接著會取得該執行角色，為您的資源類型提供適當的憑證。

```
--execution-role-arn arn:aws:iam::123456789012:role/MyIAMRole
```

3. (選用) 搭配 [describe-type-registration](#) 命令使用註冊字符，追蹤您註冊請求的進度。

當 CloudFormation 完成註冊請求時，會將請求的進度狀態設為 COMPLETE。

以下範例會使用以上 `describe-type-registration` 命令傳回的註冊權杖來傳回註冊狀態資訊。

```
aws cloudformation describe-type-registration \
  --registration-token f5525280-104e-4d35-bef5-8f1fexample \
  --region us-west-2
```

命令會傳回下列輸出：

```
{
  "ProgressStatus": "COMPLETE",
  "TypeArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example",
  "Description": "Deployment is currently in DEPLOY_STAGE of status COMPLETED; ",
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/My-Resource-Example/00000001"
}
```

Important

若您要註冊的擴充功能是勾點，則必須執行下一個步驟。您必須為 `HookInvocationStatus` 屬性指定 `ENABLED`。此操作將啟用勾點結構描述 `properties` 區段中定義的掛鉤屬性。有關詳細資訊，請參閱《CloudFormation Hooks 使用者指南》中的 [勾點組態結構描述語法參考](#)。

若要指定勾點 (AWS CLI) 的組態資料

1. 取得勾點的 ARN 並儲存。您可以使用 AWS 管理主控台 或 取得勾點的 ARN AWS CLI。如需更多資訊，請參閱[在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)。

```
export HOOK_TYPE_ARN="arn:aws:cloudformation:us-west-2:123456789012:type/hook/  
Organization-Service-Hook/"
```

2. 若要指定組態資料，請使用 [set-type-configuration](#) 命令。您為 `--configuration` 傳遞的 JSON 必須根據勾點的組態結構描述進行驗證。要啟用勾點，您必須在 `HookConfiguration` 區段將 `HookInvocationStatus` 屬性設定為 `ENABLED`。

```
aws cloudformation set-type-configuration \  
  --configuration '{"CloudFormationConfiguration":{"HookConfiguration":  
  {"HookInvocationStatus": "ENABLED", "FailureMode": "FAIL", "Properties":{}}}}' \  
  --type-arn $HOOK_TYPE_ARN --region us-west-2
```

有關詳細資訊，請參閱《CloudFormation Hooks 使用者指南》中的[勾點組態結構描述語法參考](#)。

從帳戶移除第三方私有擴充功能

要移除第三方私有擴充功能或擴充功能版本，請使用 [deregister-type](#) 命令。

您可取消註冊特定的擴充功能版本，或整個擴充功能。要取消註冊擴充功能，必須分別取消註冊該擴充功能的全部已註冊版本。若某個擴充功能僅有一個已註冊版本，取消註冊該版本會導致取消註冊擴充功能本身。您無法取消註冊擴充功能的預設版本，除非是該擴充功能的唯一已註冊版本，這種情況下擴充功能本身也會被取消註冊。

Warning

私人擴充功能的取消註冊操作無法撤銷。此動作將會：

- 使該擴充功能在所有 CloudFormation 操作中無法使用。
- 導致未來使用此擴充功能的堆疊更新失敗 (適用於模組和資源類型)。儘管您稍後可以重新私下註冊該擴充功能，但如果 CloudFormation 依賴舊版本，可能會導致失敗。

繼續之前，使用 [list-stacks](#) 與 [get-template](#) 命令，驗證是否沒有作用中的堆疊使用此擴充功能。

取消註冊擴充功能命令的範例

本節提供範例，說明取消註冊私有擴充功能的不同方式。

依據類型名稱取消註冊

搭配 `--type` 和 `--type-name` 選項使用 [deregister-type](#) 命令，即可取消註冊您的擴充功能。

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE
```

依據類型名稱和版本取消註冊

要取消註冊特定版本的擴充功能，請在命令中指定 `--version-id` 選項。

```
aws cloudformation deregister-type \  
  --type MODULE \  
  --type-name My::S3::SampleBucket::MODULE \  
  --version-id 00000001
```

Tip

要先將不同版本的擴充功能設定為預設版本，使用 [set-type-default-version](#) 命令。

依據 ARN 取消註冊

搭配 `--arn` 選項並指定您擴充功能的 ARN，即可取消註冊該擴充功能。

```
aws cloudformation deregister-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Organization-Service-Resource
```

編輯您帳戶中擴充功能的組態資料

本主題提供在特定區域中編輯您帳戶內擴充功能組態資料的指引。擴充功能可以包含套用至指定帳戶和區域擴充功能之所有執行個體的組態屬性。擴充功能作者會在擴充功能的組態定義中定義這些內容。如果擴充功能組態定義中有任何所需屬性，您必須指定這些屬性，才能在帳戶和區域中使用擴充功能。

如需有關開發擴充功能時如何定義組態定義的更多資訊，請參閱下列文件。

- [勾點組態結構描述語法參考](#)
- [定義擴充功能的帳戶層級組態](#)

主題

- [使用動態參考所需的許可](#)
- [編輯擴充功能的組態資料 \(主控台\)](#)
- [編輯擴充功能的組態資料 \(AWS CLI\)](#)

使用動態參考所需的許可

如果您的組態資料包含儲存在 AWS Systems Manager 或 中的值的動態參考 AWS Secrets Manager，則用於佈建 類型的任何角色（例如，建立或更新堆疊時）必須具有適當的許可才能擷取該值。具體而言：

- 如果組態資料包含存儲在 AWS Systems Manager Parameter Store 中的參數，則用於佈建類型的使用者或角色必須具備呼叫 [GetParameter](#) 的許可。
- 如果組態資料包含存放在 中的秘密 AWS Secrets Manager，則用來佈建 類型的使用者或角色必須具有呼叫 的許可 [GetSecretValue](#)。

如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

編輯擴充功能的組態資料 (主控台)

請依照本節中的步驟，使用主控台執行下列操作：

- 檢視擴充功能的目前組態資料
- 更新您帳戶的擴充功能組態資料

檢視擴充功能的目前組態資料

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。

3. 在導覽窗格的登錄檔下方，選擇已啟用的擴充功能。
4. 找到您要檢視的擴充功能。如需詳細資訊，請參閱[在 CloudFormation 登錄檔中檢視可用的與已啟用擴充功能](#)。
5. 選擇擴充功能以檢視擴充功能詳細資訊。
6. 在擴充功能詳細資訊頁面上，選擇 Configuration (組態) 索引標籤。
7. 展開 Configuration schema (組態結構描述) 索引標籤，以查看為擴充功能定義的組態結構描述。
8. 展開 Configuration (組態) 索引標籤，以查看您為此擴充功能設定的目前組態。

更新擴充功能的組態資料

1. 在擴充功能詳細資訊頁面上，從組態索引標籤中，選擇編輯組態。

或者，從 Actions (動作) 中，選擇編輯，然後選擇編輯組態。

CloudFormation 會顯示 Configure extension (設定擴充功能) 頁面。確保已開啟 View configuration schema (檢視組態結構描述) 以查看擴充功能目前的組態定義結構描述。

2. 在 Configuration JSON (組態 JSON) 文字方塊中，輸入 JSON 字串，代表您要為此擴充功能設定的組態結構描述。此字串必須根據 Configuration schema (組態結構描述) 中定義的結構描述進行驗證。
3. 選擇設定擴充功能。

編輯擴充功能的組態資料 (AWS CLI)

請依照本節中的步驟使用 AWS CLI 來：

- 檢視擴充功能的目前組態資料
- 更新您帳戶的擴充功能組態資料

檢視擴充功能的目前組態資料

- 使用 [describe-type](#) 命令傳回有關擴充功能的詳細資訊。該輸出的 ConfigurationSchema 元素包含指定區域中擴充功能的目前組態定義。

或者，使用 [batch-describe-type-configurations](#) 命令傳回有關多個擴充功能的組態資料。

更新擴充功能的組態資料

- 若要指定組態資料，請使用 [set-type-configuration](#) 命令。您為 `--configuration` 傳遞的 JSON 必須根據擴充功能的組態結構描述進行驗證。

在下列範例中，`set-type-configuration` 命令會指定 `--configuration` 選項組態資料 `"{"CredentialKey": "testUserCredential"}"`。

```
aws cloudformation set-type-configuration --type RESOURCE \
  --type-name My::Resource::Example \
  --configuration-alias default \
  --configuration '{"CredentialKey": "testUserCredential"}' \
  --region us-west-2
```

在中記錄資源類型 AWS Config

您可以指定 AWS Config 自動追蹤私有資源類型，並將這些資源的變更記錄為組態項目。這可讓您檢視這些私有資源類型的組態歷史記錄，以及撰寫 AWS Config 規則 規則來驗證組態最佳實務。Hook 延伸 AWS Config 模組需要。

若要讓 AWS Config 自動追蹤您的私有資源類型：

- 通過 CloudFormation 管理資源。這包括透過 CloudFormation 執行所有建立，更新和刪除資源的操作。

Note

如果您使用 IAM 角色來執行堆疊操作，則該 IAM 角色必須具有呼叫下列 AWS Config 動作的許可：

- [PutResourceConfig](#)
- [DeleteResourceConfig](#)

- 設定 AWS Config 以記錄所有資源類型。如需詳細資訊，請參閱《AWS Config 開發人員指南》中的 [使用 AWS CLI 記錄第三方資源的組態](#)。

Note

AWS Config 不支援錄製包含定義為必要和唯讀屬性的私有資源。

根據設計，用於建立 AWS Config 組態項目的結構描述中不會傳回定義為唯寫的資源屬性。因此，如果包含定義為唯寫和必要的屬性，則會導致建立組態項目失敗，因為將不存在必要屬性。若要檢視用來建立組態項目的結構描述，您可以檢閱 [DescribeType](#) 動作的 schema 屬性。

如需有關組態項目的詳細資訊，請參閱《AWS Config 開發人員指南》中的[組態項目](#)。

防止在組態項目中記錄感應性屬性

您的資源類型可能包含您認為機密資訊 (例如密碼、秘密或其他機密資料) 的內容，意即您不想記錄為組態項目的一部分。若要防止將內容記錄在組態項目中，您可以在資源類型結構描述的 `writeOnlyproperties` 清單中包含該內容。使用者可指定列為 `writeOnlyproperties` 的資源屬性，但不會由 `read` 或 `list` 請求傳回。

如需詳細資訊，請參閱《CloudFormation 使用者指南》中的 [writeOnlyProperties](#)。

使用 CodePipeline 進行持續交付

持續交付屬於一種發佈實務，系統可透過該實務自動建置、測試及準備程式碼變更，以便將變更發佈至生產程序。您能夠善用 CloudFormation 與 CodePipeline 進行持續交付，藉此自動建置並測試 CloudFormation 範本的變更，隨後將這些範本推動至生產堆疊。此發行程序可讓您快速且可靠地變更 AWS 基礎設施。

舉例來說，您能夠建立一個工作流程，進而在將更新後範本提交至程式碼儲存庫時，透過該工作流程自動建置測試堆疊。待 CloudFormation 建置測試堆疊完成，您即可進行測試，接著決定是否要將變更推送至生產堆疊。如需持續交付優點的詳細資訊，請參閱[什麼是持續交付？](#)說明。

使用 CodePipeline 透過為 CloudFormation 堆疊建置管道，建置持續交付工作流程。由於 CodePipeline 已與 CloudFormation 內建整合，您可以在管道中指定 CloudFormation 特定動作，例如建立、更新或刪除堆疊。如需有關 CodePipeline 的詳細資訊，請見《AWS CodePipeline 使用者指南》<https://docs.aws.amazon.com/codepipeline/latest/userguide/>。

主題

- [逐步解說：建置管線以用於測試和生產堆疊](#)
- [CloudFormation 組態屬性參考](#)
- [CloudFormation 成品](#)
- [搭配 CodePipeline 管道使用參數覆寫函數](#)

逐步解說：建置管線以用於測試和生產堆疊

試想您提交 CloudFormation 範本的發行程序，然後 CloudFormation 會使用它來自動建置測試堆疊。檢閱測試堆疊後，您可以預覽您的變更將如何修改您的生產堆疊，然後選擇是否要實作。為達成此工作流程，您可以使用 CloudFormation 建置您的測試堆疊、刪除測試堆疊、建立變更集，然後執行變更集。不過，每個動作您都必須手動與 CloudFormation 互動。在此逐步教學中，我們將建置 CodePipeline 管道，它可自動化許多這些動作，協助您以 CloudFormation 堆疊達成持續交付工作流程。

先決條件

本演練假設您已使用 CodePipeline 和 CloudFormation，並了解管道、AWS CloudFormation 範本和堆疊的運作方式。如需有關 CodePipeline 的詳細資訊，請見《AWS CodePipeline 使用者指南》<https://docs.aws.amazon.com/codepipeline/latest/userguide/>。您也需要在建立管道的相同 中擁有 Amazon S3 AWS 區域 儲存貯體。

⚠ Important

範例 WordPress 範本建立 EC2 執行個體，它需要網際網路連線。確認您有允許將流量傳送至網際網路的預設 VPC 和子網路。

逐步解說概觀

此逐步解說為堆疊中的範例 WordPress 網站建置一個管道。此管道分為三個階段。每個階段皆必須至少包含一個動作，它是管道在您的成品 (您的輸入) 上執行的任務。階段會組織管道中的動作。在階段處理新的成品之前，CodePipeline 必須完成階段中的所有動作，例如，如果您提交新的輸入以重新執行管道。

在此逐步教學結束時，您將擁有可執行以下工作流程的管道：

1. 管道的第一個階段會從儲存庫擷取來源成品 (CloudFormation 範本及其組態檔案)。

您將準備成品，其中包含範例 WordPress 範本，並將它上傳至 S3 儲存貯體。

2. 在第二個階段，管道會建立測試堆疊，然後等待您的核准。

在您檢閱測試堆疊之後，可選擇繼續使用原始管道或建立並提交另一個成品來進行變更。如果您核准，此階段將會刪除測試堆疊，然後管道將持續至下一個階段。

3. 在第三個階段，管道會針對生產堆疊建立變更集，然後等待您的核准。

在您初次執行時，不會有生產堆疊。變更集向您顯示 CloudFormation 將建立的所有資源。如果您核准，此階段將執行變更集並建置您的生產堆疊。

i Note

CloudFormation 是免費服務。不過，您需要按每個的目前費率支付堆疊中包含 AWS 的資源費用，例如 EC2 執行個體。如需 AWS 定價的詳細資訊，請參閱 <https://aws.amazon.com> 中每個產品的詳細資訊頁面。

步驟 1：編輯成品並將它上傳到 S3 儲存貯體

在您建置管道之前，必須設定來源儲存庫與檔案。CodePipeline 會將這些來源檔案複製到管道的 [成品存放區](#)，然後使用這些檔案在管道中執行動作，例如建立 CloudFormation 堆疊。

當您使用 Amazon Simple Storage Service (Amazon S3) 作為來源儲存庫時，CodePipeline 要求您將來源檔案進行壓縮，然後再上傳到 S3 儲存貯體。壓縮的檔案為 CodePipeline 成品，它可包含 CloudFormation 範本、範本組態檔案，或兩者兼具。我們提供的成品包含範例 WordPress 範本及兩個範本組態檔案。這兩個組態檔案指定 WordPress 範本的參數值。CodePipeline 在建立 WordPress 堆疊時，將使用這些參數值。其中一個檔案包含測試堆疊的參數值，另一個則包含生產堆疊的參數值。您將必須編輯組態檔案，例如，指定您擁有的現有 EC2 金鑰對名稱。如需成品的詳細資訊，請參閱 [CloudFormation 成品](#)。

在您建置成品之後，會將它上傳到 S3 儲存貯體。

編輯和上傳成品

1. 下載並開啟範例成品：<https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/wordpress-single-instance.zip>。

此成品包含三個檔案：

- 範例 WordPress 範本：`wordpress-single-instance.yaml`
- 測試堆疊的範本組態檔案：`test-stack-configuration.json`
- 生產堆疊的範本組態檔案：`prod-stack-configuration.json`

2. 擷取所有檔案，然後使用任何文字編輯器來修改範本組態檔案。

開啟組態檔案，查看它們包含對應至 WordPress 範本參數的索引鍵/值組。組態檔案指定參數值，您的管道在建立測試和生產堆疊時，將會使用這些參數值。

編輯 `test-stack-configuration.json` 檔案以指定測試堆疊的參數值，並指定 `prod-stack-configuration.json` 檔案以指定生產堆疊的參數值。

- 將 `DBPassword` 值與 `DBRootPassword` 金鑰變更為密碼，您可用它來登入 WordPress 資料庫。如同 WordPress 範本中的定義，參數值必須僅包含英數字元。
- 將 `KeyName` 金鑰值變更為您將建立管道之區域中現有的 EC2 金鑰對名稱。

3. 將修改過的組態檔新增至原始成品 (`.zip`) 檔案，以取代重複的檔案。

您現在可以自訂成品，並可將它上傳到 S3 儲存貯體。

4. [將成品上傳到您擁有的 S3 儲存貯體。](#)

請注意檔案的位置。當您建置管道時，將指定此檔案的位置。

有關成品和 S3 儲存貯體的說明：

- 使用位於您要建立管道之相同 AWS 區域中的儲存貯體。
- CodePipeline 要求儲存貯體 [已啟用版本控制](#)。
- 您的來源儲存庫也可以使用不要求在上傳前壓縮檔案的服務，例如 GitHub 或 CodeCommit。
- 成品可以包含敏感的資訊，例如密碼。限制存取，只有獲得允許使用者可以檢視檔案。當您這麼做時，請確保 CodePipeline 仍然可以存取檔案。例如，如果您將成品上傳至 S3 儲存貯體，則請使用 [S3 儲存貯體政策或使用者政策](#) 來限制存取權。

您現在有一個可讓 CodePipeline 放入您管道的成品。在接下來的步驟中，您將指定成品的位置並建置 WordPress 管道。

步驟 2：建立管道堆疊

若要建立 WordPress 管道，您將使用範例 CloudFormation 範本。除了建置管道之外，範本還會為 CodePipeline 和 CloudFormation 設定 AWS Identity and Access Management (IAM) 服務角色、CodePipeline 成品存放區的 S3 儲存貯體，以及管道傳送通知的 Amazon Simple Notification Service (Amazon SNS) 主題，例如有關檢閱的通知。範例範本可讓您輕鬆佈建和設定單一 CloudFormation 堆疊中的這些資源。

如需管道設定的詳細資訊，請參閱 [管道有何功能](#)。

Important

範例 WordPress 範本建立 EC2 執行個體，它需要網際網路連線。確認您的預設 VPC 和子網路允許將流量傳送至網際網路。

建立管道堆疊

1. 請至 <https://s3.amazonaws.com/cloudformation-examples/user-guide/continuous-deployment/basic-pipeline.yml> 下載範例範本。並將它儲存在您的電腦。
2. 開啟位在 <https://console.aws.amazon.com/cloudformation/> 的 CloudFormation 主控台。
3. 選擇支援 CodePipeline 和 CloudFormation AWS 的區域。

如需詳細資訊，請參閱 AWS 一般參考 中的 [CodePipeline 端點和配額](#) 以及 [CloudFormation 端點和配額](#)。

4. 選擇建立堆疊。

5. 在 Specify template (指定範本) 下方，選擇 Upload a template file (上傳範本檔案)，然後選擇您剛剛下載的範本 `basic-pipeline.yml`。
6. 選擇下一步。
7. 在 Stack Name (堆疊名稱) 中輸入 `sample-WordPress-pipeline`。
8. 在 Parameters (參數) 區段中指定以下參數值，然後選擇 Next (下一步)。設定堆疊參數時，如果您維持與 WordPress 範本及其組態檔案相同的名稱，就可以使用預設值。否則，請指定您使用的檔案名稱。

PipelineName

您的管道名稱，例如 `WordPress-test-pipeline`。

S3 儲存貯體

您用於儲存成品 (`.zip` 檔案) 的 S3 儲存貯體名稱。

SourceS3Key

成品的檔案名稱。如果您將成品儲存於資料夾，請將其作為檔案名稱的一部分包含在內，例如 `folder/subfolder/wordpress-single-instance.zip`。

電子郵件

CodePipeline 傳送管道通知的電子郵件地址，例如 `myemail@example.com`。

9. 在此逐步解說中，您不需要新增標籤或指定進階設定，因此請選擇 Next (下一步)。
10. 確認堆疊名稱和範本 URL 正確，然後選擇 Create stack (建立堆疊)。
11. 為了確認您知道 CloudFormation 可能會建立 IAM 資源，請勾選該核取方塊。

建立堆疊可能需要幾分鐘 AWS CloudFormation 的時間。若要監控進度，請檢視堆疊事件。如需詳細資訊，請參閱[監控堆疊進度](#)。

在建立堆疊之後，CodePipeline 將開始您的新管道。若要查看其狀態，請參閱[CodePipeline 主控台](#)。從管道清單中選擇 `WordPress-test-pipeline`。

管道有何功能

本節使用範例 WordPress 管道範本的程式碼片段，說明管道的三個階段。

階段 1：來源

管道的第一階段是來源階段，您在其中指定原始程式碼的位置。每次您將修訂推送至此位置時，CodePipeline 就會重新執行您的管道。

原始程式碼位於 S3 儲存貯體，並以其檔案名稱識別。您指定這些值作為您建立管道堆疊時的輸入參數值。為了允許在後續階段使用來源成品，程式碼片段指定 `OutputArtifacts` 屬性，其名為 `TemplateSource`。為了在之後的階段使用此成品，您指定 `TemplateSource` 作為輸入成品。

```
- Name: S3Source
  Actions:
    - Name: TemplateSource
      ActionTypeId:
        Category: Source
        Owner: AWS
        Provider: S3
        Version: '1'
      Configuration:
        S3Bucket: !Ref 'S3Bucket'
        S3ObjectKey: !Ref 'SourceS3Key'
      OutputArtifacts:
        - Name: TemplateSource
```

階段 2：TestStage

在 `TestStage` 階段，管道會建立測試堆疊、等待您的核准，然後刪除測試堆疊。

對於 `CreateStack` 動作，管道會使用測試組態檔案和 WordPress 範本來建立測試堆疊。這兩個檔案包含在從來源階段引進的 `TemplateSource` 輸入成品中。程式碼片段使用 `REPLACE_ON_FAILURE` 動作模式。如果堆疊建立失敗，管道將會取代它，因此在重新執行管道之前，您不需要清除或排解堆疊的問題。動作模式對於測試堆疊的快速更新很有用。對於 `RoleArn` 屬性，值是範本中其他位置宣告 AWS CloudFormation 的服務角色。

`ApproveTestStack` 動作會暫停管道，並傳送通知給您在建立管道堆疊時指定的電子郵件地址。雖然管道已暫停，您還是可以檢查 WordPress 測試堆疊及其資源。使用 CodePipeline [核准或拒絕](#) 此動作。`CustomData` 屬性包含您要核准之動作的描述，核准之後管道會將通知新增至電子郵件。

在您核准此動作之後，CodePipeline 會移至 `DeleteTestStack` 動作，然後刪除測試 WordPress 堆疊及其資源。

```
- Name: TestStage
  Actions:
```

```

- Name: CreateStack
  ActionTypeId:
    Category: Deploy
    Owner: AWS
    Provider: CloudFormation
    Version: '1'
  InputArtifacts:
    - Name: TemplateSource
  Configuration:
    ActionMode: REPLACE_ON_FAILURE
    RoleArn: !GetAtt [CFNRole, Arn]
    StackName: !Ref TestStackName
    TemplateConfiguration: !Sub "TemplateSource::${TestStackConfig}"
    TemplatePath: !Sub "TemplateSource::${TemplateFileName}"
  RunOrder: '1'
- Name: ApproveTestStack
  ActionTypeId:
    Category: Approval
    Owner: AWS
    Provider: Manual
    Version: '1'
  Configuration:
    NotificationArn: !Ref CodePipelineSNSTopic
    CustomData: !Sub 'Do you want to create a change set against the production
stack and delete the ${TestStackName} stack?'
  RunOrder: '2'
- Name: DeleteTestStack
  ActionTypeId:
    Category: Deploy
    Owner: AWS
    Provider: CloudFormation
    Version: '1'
  Configuration:
    ActionMode: DELETE_ONLY
    RoleArn: !GetAtt [CFNRole, Arn]
    StackName: !Ref TestStackName
  RunOrder: '3'

```

階段 3 : ProdStage

管道的 ProdStage 階段會針對現有的生產堆疊建立變更集、等待核准，然後執行變更集。

變更集提供在實作之前對生產堆疊 AWS CloudFormation 進行的所有修改的預覽。在您第一個管道執行時，您不會有執行中的生產堆疊。變更集顯示建立測試堆疊時 AWS CloudFormation 執行的動作。

若要建立變更集，CreateChangeSet 動作將使用 WordPress 範例範本以及來自 TemplateSource 輸入成品的生產範本組態。

類似上個階段，ApproveChangeSet 動作將暫停管道，並傳送電子郵件通知。雖然管道已暫停，您還是可以檢視變更集以查看生產 WordPress 堆疊的所有建議修改。使用 CodePipeline [核准或拒絕](#)此動作以繼續或停止管道。

在您核准此動作後，ExecuteChangeSet 動作會執行變更集，讓 AWS CloudFormation 執行變更集中所述的所有動作。對於初始執行，會 AWS CloudFormation 建立 WordPress 生產堆疊。在後續執行時，AWS CloudFormation 將更新堆疊。

```
- Name: ProdStage
  Actions:
    - Name: CreateChangeSet
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      InputArtifacts:
        - Name: TemplateSource
      Configuration:
        ActionMode: CHANGE_SET_REPLACE
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: !Ref ProdStackName
        ChangeSetName: !Ref ChangeSetName
        TemplateConfiguration: !Sub "TemplateSource:${ProdStackConfig}"
        TemplatePath: !Sub "TemplateSource:${TemplateFileName}"
      RunOrder: '1'
    - Name: ApproveChangeSet
      ActionTypeId:
        Category: Approval
        Owner: AWS
        Provider: Manual
        Version: '1'
      Configuration:
        NotificationArn: !Ref CodePipelineSNSTopic
        CustomData: !Sub 'A new change set was created for the ${ProdStackName} stack.
Do you want to implement the changes?'
      RunOrder: '2'
    - Name: ExecuteChangeSet
      ActionTypeId:
        Category: Deploy
```

```
Owner: AWS
Provider: CloudFormation
Version: '1'
Configuration:
  ActionMode: CHANGE_SET_EXECUTE
  ChangeSetName: !Ref ChangeSetName
  RoleArn: !GetAtt [CFNRole, Arn]
  StackName: !Ref ProdStackName
RunOrder: '3'
```

步驟 3：檢視 WordPress 堆疊

CodePipeline 通過管道時會使用 CloudFormation 建立測試和生產堆疊。若要查看這些堆疊及其輸出的狀態，請使用 CloudFormation 主控台。

檢視堆疊

1. 開啟位在 <https://console.aws.amazon.com/cloudformation/> 的 CloudFormation 主控台。
2. 根據您的管道處於測試或生產階段，選擇 Test-MyWordPressSite 或 Prod-MyWordPressSite 堆疊。
3. 若要查看您堆疊的狀態，請檢視堆疊[事件](#)。

如果堆疊處於故障狀態，請檢視狀態原因以找出堆疊錯誤。修正錯誤，然後重新執行管道。如果堆疊處於 CREATE_COMPLETE 狀態，請檢視其輸出以取得您 WordPress 網站的 URL。

您已成功使用 CodePipeline 為範例 WordPress 網站建置持續交付工作流程。如果您提交變更到 S3 儲存貯體，CodePipeline 將自動偵測新版本，然後重新執行您的管道。此工作流程可讓您更容易提交和測試變更，然後再變更您的生產網站。

步驟 4：清除資源

為避免您為不需要的服務支付費用，請刪除您的資源。

Important

請先刪除測試和生產 WordPress 堆疊，然後才能刪除管道堆疊。管道堆疊包含刪除 WordPress 堆疊時所需要的服務角色。如果您已先刪除管道堆疊，可以將其他服務角色 Amazon Resource Name (ARN) 關聯至 WordPress 堆疊，然後將它們刪除。

刪除成品存放區中的物件

1. 開啟位於 <https://console.aws.amazon.com/s3/> 的 Amazon S3 主控台。
2. 選擇 S3 儲存貯體，讓 CodePipeline 使用作為您管道的成品存放區。

儲存貯體的名稱需使用此：*stackname-artifactstorebucket-id*。如果您依照此逐步解說，儲存貯體名稱可能類似於以下範例：sample-WordPress-pipeline-artifactstorebucket-12345abcd12345。

3. 刪除成品存放區 S3 儲存貯體中的所有物件。

當您在下個步驟中刪除管道堆疊時，此儲存貯體必須是空的。否則，CloudFormation 將無法刪除該儲存貯體。

刪除堆疊

1. 從 CloudFormation 主控台選擇您要刪除的堆疊。

如果管道建立的 WordPress 堆疊仍在執行中，請先選擇它們。在預設情況下，堆疊名稱為 Test-MyWordPressSite 和 Prod-MyWordPressSite。

如果您已刪除 WordPress 堆疊，請選擇 sample-WordPress-pipeline 堆疊。

2. 選擇 Actions (動作)，然後選擇 Delete Stack (刪除堆疊)。
3. 在確認訊息中，選擇 Yes, Delete (是，刪除)。

CloudFormation 刪除堆疊所有的堆疊資源，例如 EC2 執行個體、通知主題、服務角色和管道。

現在您了解如何以 CodePipeline 建置基本 CloudFormation 工作流程，您可以使用範例範本和成品做為起點來建置自己的工作流程。

另請參閱

以下相關資源可協助您使用這些參數。

- 如需有關 CodePipeline 中 CloudFormation 動作參數的詳細資訊，請參閱《AWS CodePipeline 使用者指南》中的 [CloudFormation 部署動作組態參考](#)。
- 如需動作提供者的範例範本值 (例如 Owner 欄位或 configuration 欄位)，請參閱 AWS CodePipeline User Guide (《使用者指南》) 中的 [Action structure reference](#) (動作結構參考)。

- 若要下載 YAML 或 JSON 格式的管道堆疊範本範例，請參閱《AWS CodePipeline 使用者指南》中的[教學課程：使用 CloudFormation 建立管道](#)。

CloudFormation 組態屬性參考

當您建置 CodePipeline 管道時，您可以將 Deploy 動作以 CloudFormation 做為提供者新增至管道。然後，您必須指定管道叫用 CloudFormation 的動作和動作的設定。本主題說明 CloudFormation 組態屬性。若要指定屬性，您可以使用 CodePipeline 主控台，也可以建立 JSON 物件以用於 AWS CLI、CodePipeline API 或 CloudFormation 範本。

主題

- [組態屬性 \(主控台\)](#)
- [組態屬性 \(JSON 物件\)](#)
- [另請參閱](#)

組態屬性 (主控台)

CodePipeline [主控台](#) 會顯示組態屬性，根據您選擇的動作模式指示哪些屬性為必要項目。

Note

當您建立管道時，您只能指定 Create or update a stack (建立或更新堆疊) 或 Create or replace a change set (建立或取代變更組) 動作模式。Advanced (進階) 區段中的屬性只有在您編輯管道時才可使用。

Action mode (動作模式)

CodePipeline 在處理關聯階段時呼叫 CloudFormation 的動作。選擇下列其中一個動作模式：

- Create or replace a change set (建立或取代變更組) 會根據您提交的堆疊名稱和範本建立變更組 (若不存在的話)。如果變更集存在，會將其 CloudFormation 刪除，然後建立新的變更集。
- Create or update a stack (建立或更新堆疊) 會在指定堆疊不存在時建立堆疊。如果堆疊存在，會 CloudFormation 更新堆疊。使用此動作來更新現有堆疊。CodePipeline 不會取代堆疊。
- Delete a stack (刪除堆疊) 刪除堆疊。若您指定不存在的堆疊，動作會成功完成，而不會刪除任何堆疊。
- Execute a change set (執行變更組) 執行變更組。

- Replace a failed stack (取代故障堆疊) 會在指定堆疊不存在時建立堆疊。如果堆疊存在且處於失敗狀態 (報告為 ROLLBACK_COMPLETE、ROLLBACK_FAILED、DELETE_FAILED、或 UPDATE_ROLLBACK_FAILED)CREATE_FAILED，會 CloudFormation 刪除堆疊，然後建立新的堆疊。如果堆疊未處於失敗狀態，會 CloudFormation 更新它。使用此動作來取代故障的堆疊，無須復原或故障診斷。您通常會選擇此模式以進行測試。

Stack name (堆疊名稱)

與現有堆疊或您希望建立的堆疊建立關聯的名稱。在您 AWS 建立堆疊的區域中，名稱必須是唯一的。

Note

堆疊名稱僅能使用英數字元 (區分大小寫) 和連字號。必須以字母字元開頭，且長度不可超過 128 個字元。

Change set name (變更組合名稱)

現有變更組或您希望為指定堆疊建立之新變更組的名稱。

Template (範本)

CloudFormation 範本檔案的位置，其格式為 *ArtifactName::TemplateFileName*。

Template configuration (範本組態)

範本組態檔案的位置，其格式為 *ArtifactName::TemplateConfigurationFileName*。範本組態檔案可包含範本參數值、堆疊政策和標籤。若您在其中包含敏感資訊 (例如密碼)，請限制此檔案的存取。如需詳細資訊，請參閱 [CloudFormation 成品](#)。

功能

對於包含特定資源的堆疊，明確認可 CloudFormation 可能會建立或更新這些資源。例如，若您的堆疊範本包含 AWS Identity and Access Management (IAM) 資源，您必須指定 CAPABILITY_IAM。如需詳細資訊，請參閱 [CreateStack](#) API 操作請求參數。

若您在您的堆疊範本中具有 IAM 資源，您必須指定此屬性。

您可以指定多個功能。

Role name (角色名稱)

在指定堆疊中的資源上操作時，CloudFormation 擔任的 IAM 服務角色名稱。

Output file name (輸出檔案名稱)

在 Advanced (進階) 區段中，您可以指定輸出檔案名稱 (例如 `CreateStackOutput.json`)，CodePipeline 會在執行指定動作之後將其新增至[輸出成品](#)。輸出成品包含 JSON 檔案，其中包含 CloudFormation 範本 Outputs 區段的內容。

若您沒有指定名稱，CodePipeline 便不會產生輸出成品。

Parameter overrides (參數覆寫)

參數是在您的範本中定義，而且可讓您在建立或更新堆疊時輸入自訂值。您可以指定覆寫範本組態檔案中範本參數值的 JSON 物件。所有參數名稱都必須在堆疊範本中存在。如需詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

Note

能存放在 `ParameterOverrides` 屬性的 JSON 物件具有 1 KB 的最大大小限制。

我們建議您使用範本組態檔案來指定您大多數的參數值。僅使用參數覆寫來指定動態參數值。除非您執行管道，否則動態參數是未知的。

下列範例會使用參數覆寫函數定義 `ParameterName` 參數的值。函數會從 CodePipeline 輸入成品擷取值。如需參數覆寫函數的詳細資訊，請參閱[搭配 CodePipeline 管道使用參數覆寫函數](#)。

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

組態屬性 (JSON 物件)

當您指定 CloudFormation 作為階段動作的提供者時，請在 `Configuration` 屬性中定義以下屬性。將 JSON 物件用於 AWS CLI、CodePipeline API 或 CloudFormation 範本。如需範例，請參閱[逐步解說：建置管線以用於測試和生產堆疊](#)和[CloudFormation 組態屬性參考](#)。

ActionMode

CodePipeline 在處理關聯階段時呼叫 CloudFormation 的動作。僅指定下列其中一個動作模式：

- `CHANGE_SET_EXECUTE` 執行變更集。

- `CHANGE_SET_REPLACE` 根據您提交的堆疊名稱和範本建立變更組 (若不存在的話)。如果變更集存在，會將其 CloudFormation 刪除，然後建立新的變更集。
- `CREATE_UPDATE` 在指定堆疊不存在時建立堆疊。若堆疊存在，則 CloudFormation 會更新堆疊。使用此動作來更新現有堆疊。CodePipeline 不會取代堆疊。
- `DELETE_ONLY` 刪除堆疊。若您指定不存在的堆疊，動作會成功完成，而不會刪除任何堆疊。
- `REPLACE_ON_FAILURE` 在指定堆疊不存在時建立堆疊。如果堆疊存在且處於失敗狀態 `ROLLBACK_FAILED` (報告為 `ROLLBACK_COMPLETE`、`DELETE_FAILED`、或 `UPDATE_ROLLBACK_FAILED`) `CREATE_FAILED`，會 CloudFormation 刪除堆疊，然後建立新的堆疊。如果堆疊未處於失敗狀態，會 CloudFormation 更新它。使用此動作來自動取代故障的堆疊，無須復原或故障診斷。您通常會選擇此模式以進行測試。

此屬性為必要。

Capabilities

對於包含特定資源的堆疊，明確認可 CloudFormation 可能會建立或更新這些資源。例如，若您的堆疊範本包含 AWS Identity and Access Management (IAM) 資源，您必須指定 `CAPABILITY_IAM`。如需詳細資訊，請參閱 [CreateStack](#) API 操作請求參數。

此屬性為條件式。若您在您的堆疊範本中具有 IAM 資源，您必須指定此屬性。

您可以指定多個功能。下列範例會將 `CAPABILITY_IAM` 和 `CAPABILITY_AUTO_EXPAND` 屬性新增至範本：

YAML

```
configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
  ChangeSetName: pipeline-changeset
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'
```

JSON

```
"configuration": {
  "ActionMode": "CHANGE_SET_REPLACE",
  "Capabilities": "CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND",
  "ChangeSetName": "pipeline-changeset",
  "RoleArn": "CloudFormation_Role_ARN",
```

```

    "StackName": "my-pipeline-stack",
    "TemplateConfiguration": "my-pipeline-stack::template-configuration.json",
    "TemplatePath": "my-pipeline-stack::template-export.yml"
  }

```

ChangeSetName

現有變更組或您希望為指定堆疊建立之新變更組的名稱。

針對下列動作模式，此屬性為必要：CHANGE_SET_REPLACE 和 CHANGE_SET_EXECUTE。針對其他所有動作模式，可忽略此屬性。

OutputFileName

輸出檔案的名稱，例如 CreateStackOutput.json。CodePipeline 會在執行指定動作之後將檔案新增至[輸出成品](#)。輸出成品包含 JSON 檔案，其中包含 CloudFormation 範本 Outputs 區段的內容。

此屬性為選擇性。若您沒有指定名稱，CodePipeline 便不會產生輸出成品。

ParameterOverrides

參數是在您的範本中定義，而且可讓您在建立或更新堆疊時輸入自訂值。您可以指定覆寫範本組態檔案中範本參數值的 JSON 物件。所有參數名稱都必須在堆疊範本中存在。如需詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

下列範例會將 InstanceType 和 KeyName 參數覆寫新增至範本：

YAML

```

configuration:
  ActionMode: CHANGE_SET_REPLACE
  Capabilities: CAPABILITY_NAMED_IAM
  ChangeSetName: pipeline-changeset
  ParameterOverrides: '{"InstanceType": "t2.small", "KeyName": "my-keypair"}'
  RoleArn: CloudFormation_Role_ARN
  StackName: my-pipeline-stack
  TemplateConfiguration: 'my-pipeline-stack::template-configuration.json'
  TemplatePath: 'my-pipeline-stack::template-export.yml'

```

JSON

```

"configuration": {

```

```
"ActionMode": "CHANGE_SET_REPLACE",
"Capabilities": "CAPABILITY_NAMED_IAM",
"ChangeSetName": "pipeline-changeset",
"ParameterOverrides": "{\"InstanceType\": \"t2.small\", \"KeyName\": \"my-keypair\"}",
"RoleArn": "CloudFormation_Role_Arn",
"StackName": "my-pipeline-stack",
"TemplateConfiguration": "my-pipeline-stack::template-configuration.json",
"TemplatePath": "my-pipeline-stack::template-export.yml"
}
```

Note

可存放至 `ParameterOverrides` 屬性的 JSON 物件大小上限為 1 KB。

我們建議您使用範本組態檔案來指定您大多數的參數值。僅使用參數覆寫來指定動態參數值。除非您執行管道，否則動態參數值都是未知的。

下列範例會使用參數覆寫函數定義 `ParameterName` 參數的值。函數會從 CodePipeline 輸入成品擷取值。如需參數覆寫函數的詳細資訊，請參閱[搭配 CodePipeline 管道使用參數覆寫函數](#)。

```
{
  "ParameterName" : { "Fn::GetParam" : ["ArtifactName", "config-file-name.json",
    "ParamName"]}
}
```

此屬性為選擇性。

RoleArn

IAM 服務角色的 Amazon Resource Name (ARN)，其會在堆疊中的資源上操作時 CloudFormation 擔任。

針對下列動作模式，此屬性為必要：CREATE_UPDATE、REPLACE_ON_FAILURE、DELETE_ONLY 及 CHANGE_SET_REPLACE。執行變更集時，不會套用 RoleArn。若您不使用 CodePipeline 來建立變更集，請確定變更集或堆疊具有關聯角色。

StackName

現有堆疊或您希望建立之堆疊的名稱。

針對所有動作模式，此屬性為必要。

TemplateConfiguration

TemplateConfiguration 是範本組態檔案。您將檔案包含在此動作的輸入成品中。範本組態檔案名稱遵循此格式：

Artifactname::TemplateConfigurationFileName

Artifactname 是出現在 CodePipeline 中的輸入成品名稱。例如，來源階段的成品名為 SourceArtifact 且 test-configuration.json 檔案名稱建立 TemplateConfiguration 名稱，如此範例所顯示：

```
"TemplateConfiguration": "SourceArtifact::test-configuration.json"
```

範本組態檔案可包含範本參數值和堆疊政策。若您在其中包含敏感資訊 (例如密碼)，請限制此檔案的存取。如需範例範本組態檔，請參閱 [CloudFormation 成品](#)。

此屬性為選擇性。

TemplatePath

TemplatePath 代表 CloudFormation 範本檔案。您將檔案包含在此動作的輸入成品中。該檔案名稱遵循此格式：

Artifactname::TemplateFileName

Artifactname 是出現在 CodePipeline 中的輸入成品名稱。例如，來源階段的成品名為 SourceArtifact 且 template.yaml 檔案名稱建立 TemplatePath 名稱，如此範例所顯示：

```
"TemplatePath": "SourceArtifact::template.yaml"
```

針對下列動作模式，此屬性為必要：CREATE_UPDATE、REPLACE_ON_FAILURE 和 CHANGE_SET_REPLACE。針對其他所有動作模式，可忽略此屬性。

另請參閱

以下相關資源可協助您使用這些參數。

- 如需有關 CodePipeline 中 CloudFormation 動作參數的詳細資訊，請參閱《AWS CodePipeline 使用者指南》中的 [CloudFormation 部署動作組態參考](#)。

- 如需動作提供者的範例範本值 (例如 Owner 欄位或 configuration 欄位)，請參閱 AWS CodePipeline User Guide (《使用者指南》) 中的 [Action structure reference](#) (動作結構參考)。
- 若要下載 YAML 或 JSON 格式的管道堆疊範本範例，請參閱《AWS CodePipeline 使用者指南》中的[教學課程：使用 CloudFormation 建立管道](#)。

CloudFormation 成品

CodePipeline 會對成品執行任務，因為 CodePipeline 執行管道。對於 CloudFormation，成品可以包含堆疊範本檔案、範本組態檔案或兩者。CodePipeline 使用這些成品來使用 CloudFormation 堆疊和變更集。

如果您使用 Amazon Simple Storage Service (Amazon S3) 作為來源儲存庫，則必須先將範本和範本組態檔案壓縮為單一檔案，再將檔案上傳至 S3 儲存貯體。對於其他儲存庫，例如 GitHub 和 AWS CodeCommit，上傳成品而不壓縮。如需詳細資訊，請參閱《AWS CodePipeline 使用者指南》中的[建立管道、階段和動作](#)。

您可以將所需數目的檔案新增至儲存庫。例如，建議您包括相同範本的兩個不同組態：一個用於測試組態，另一個用於生產組態。

本主題說明每種成品類型。

主題

- [堆疊範本檔案](#)
- [範本組態檔案](#)
- [另請參閱](#)

堆疊範本檔案

堆疊範本檔案會定義 CloudFormation 佈建和設定的資源。這些檔案與您使用 建立或更新堆疊時所使用的範本檔案相同 CloudFormation。您可以使用 YAML 或 JSON 格式的範本。如需範本的詳細資訊，請參閱[CloudFormation 範本區段](#)。

範本組態檔案

範本組態檔案是一種 JSON 格式的文字檔案，可指定範本參數值、[堆疊政策](#)和標籤。使用這些組態檔案，以指定參數值或堆疊的堆疊政策。您指定的所有參數值都必須宣告於相關聯的範本中。

若您在檔案包含敏感資訊 — 例如密碼 — 請限制此檔案的存取。例如，如果您將成品上傳至 S3 儲存貯體，則請使用 [S3 儲存貯體政策或使用者政策](#) 來限制存取權。

若要建立組態檔案，請使用下列格式：

```
{
  "Parameters" : {
    "NameOfTemplateParameter" : "ValueOfParameter",
    ...
  },
  "Tags" : {
    "TagKey" : "TagValue",
    ...
  },
  "StackPolicy" : {
    "Statement" : [
      StackPolicyStatement
    ]
  }
}
```

下列範例指定 TestEC2Key 參數的 KeyName，並新增值為 Department 的 Marketing 標籤，然後新增允許所有更新動作的堆疊政策，但刪除資源的更新除外。

```
{
  "Parameters" : {
    "KeyName" : "TestEC2Key"
  },
  "Tags" : {
    "Department" : "Marketing"
  },
  "StackPolicy" : {
    "Statement" : [
      {
        "Effect" : "Allow",
        "NotAction" : "Update:Delete",
        "Principal": "*",
        "Resource" : "*"
      }
    ]
  }
}
```

另請參閱

以下相關資源可協助您使用這些參數。

- 如需有關 CodePipeline 中 CloudFormation 動作參數的詳細資訊，請參閱《AWS CodePipeline 使用者指南》中的 [CloudFormation 部署動作組態參考](#)。
- 如需動作提供者的範例範本值 (例如 Owner 欄位或 configuration 欄位)，請參閱 AWS CodePipeline User Guide (《使用者指南》) 中的 [Action structure reference](#) (動作結構參考)。
- 若要下載 YAML 或 JSON 格式的管道堆疊範本範例，請參閱《AWS CodePipeline 使用者指南》中的 [教學課程：使用 CloudFormation 建立管道](#)。

搭配 CodePipeline 管道使用參數覆寫函數

在 CodePipeline 階段中，您可以指定 CloudFormation 動作的 [參數覆寫](#)。參數覆寫可讓您指定範本參數值，以覆寫範本組態檔案中的值。CloudFormation 提供 函數，協助您指定動態值 (管道執行之前未知的值)。

主題

- [Fn::GetArtifactAtt](#)
- [Fn::GetParam](#)
- [另請參閱](#)

Fn::GetArtifactAtt

Fn::GetArtifactAtt 函數會從輸入成品擷取屬性值，例如存放成品的 S3 儲存貯體名稱。使用此函數來指定成品的屬性，例如其檔案名稱或儲存 Amazon S3 貯體名稱。

執行管道時，CodePipeline 會複製並將檔案寫入管道的成品存放區 (S3 儲存貯體)。CodePipeline 會在成品存放區中產生檔案名稱。在您執行管道前，這些檔案名稱為未知。

例如，在您的管道中，您可能會有來源階段，在此階段中 CodePipeline 會將您的 AWS Lambda 函數原始程式碼複製到成品存放區。在下一個階段，您有一個 CloudFormation 範本可建立 Lambda 函數，但 CloudFormation 需要檔案名稱才能建立函數。您必須使用 Fn::GetArtifactAtt 函數傳遞正確的 S3 儲存貯體和檔案名稱。

語法

使用下列語法來擷取成品的屬性值。

```
{ "Fn::GetArtifactAtt" : [ "artifactName", "attributeName" ] }
```

artifactName

輸入成品的名稱。您必須為關聯動作將此成品宣告為輸入。

attributeName

您希望擷取其值的成品屬性名稱。如需每一個成品屬性的詳細資訊，請參閱以下的「屬性」一節。

範例

下列參數覆寫透過擷取 S3 儲存貯體名稱和 BucketName 成品的檔案名稱，指定 ObjectKey 和 LambdaFunctionSource 參數。此範例假設 CodePipeline 已複製 Lambda 函數原始程式碼，並已將其儲存為成品 (例如，作為來源階段的一部分)。

```
{  
  "BucketName" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "BucketName"] },  
  "ObjectKey" : { "Fn::GetArtifactAtt" : ["LambdaFunctionSource", "ObjectKey"] }  
}
```

屬性

您可以擷取成品的下列屬性。

BucketName

存放成品的 S3 儲存貯體名稱。

ObjectKey

包含由 CodePipeline 產生之成品的 .zip 檔案名稱，例如 1ABCyZZ.zip。

URL

成品的 Amazon Simple Storage Service (Amazon S3) URL，例如 `https://s3.us-west-2.amazonaws.com/artifactstorebucket-yivczw8jma0c/test/TemplateSo/1ABCyZZ.zip`。

`Fn::GetParam`

`Fn::GetParam` 函數會以 JSON 格式檔案傳回鍵/值對的值。JSON 檔案必須包含在成品中。

使用此函數從 CloudFormation 堆疊擷取輸出值，並將其用作另一個動作的輸入。例如，如果您指定 CloudFormation 動作的輸出檔案名稱，CodePipeline 會將輸出儲存在 JSON 檔案中，然後將其新增至輸出成品的 .zip 檔案。使用 `Fn::GetParam` 函數擷取輸出值，並用它作為另一的動作的輸入。

語法

使用以下語法來擷取鍵/值對的值。

```
{ "Fn::GetParam" : [ "artifactName", "JSONFileName", "keyName" ] }
```

`artifactName`

成品的名稱，必須包含在關聯動作的輸入成品中。

`JSONFileName`

包含在成品中的 JSON 檔案名稱。

`keyName`

您希望擷取其值的鍵名稱。

範例

以下範例示範如何在參數覆寫中使用 `Fn::GetParam` 函數。

語法

以下參數覆寫透過從位於 `WebSiteURL` 成品內的 URL 檔案擷取 `stack-output.json` 鍵的值，指定 `WebStackOutput` 參數。

```
{  
  "WebSiteURL" : { "Fn::GetParam" : ["WebStackOutput", "stack-output.json", "URL"] }  
}
```

CloudFormation 範本程式碼片段

下列 CloudFormation 範本程式碼片段來自 CodePipeline 管道，示範如何傳遞堆疊輸出。這些程式碼片段顯示管道定義的兩個階段。第一個階段會建立堆疊並將其輸出儲存在 TestOutput.json 成品內的 StackAOutput 檔案中。這些值會以 OutputFileName 和 OutputArtifacts 屬性指定。

該階段來源輸出品品的名稱為 TemplateSource。堆疊範本的檔案名稱為 teststackA.yaml，而組態檔案的檔案名稱為 test-configuration.json。在這兩個階段中，這些值都會指定給 TemplateConfiguration 和 TemplatePath 屬性，如下所示：

```
TemplateConfiguration: TemplateSource::test-configuration.json
TemplatePath: TemplateSource::teststackA.yaml
```

Example 建立堆疊 A 階段

```
- Name: CreateTestStackA
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        OutputFileName: TestOutput.json
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackA
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackA.yaml
      InputArtifacts:
        - Name: TemplateSource
      OutputArtifacts:
        - Name: StackAOutput
      RunOrder: '1'
```

在後續階段中，堆疊 B 會使用堆疊 A 的輸出。在 ParameterOverrides 屬性中，範例使用 Fn::GetParam 函數指定 StackBInputParam 參數。產生的值是和 StackAOutputName 鍵關聯的值。

Example 建立堆疊 B 階段

```
- Name: CreateTestStackB
  Actions:
    - Name: CloudFormationCreate
      ActionTypeId:
        Category: Deploy
        Owner: AWS
        Provider: CloudFormation
        Version: '1'
      Configuration:
        ActionMode: CREATE_UPDATE
        Capabilities: CAPABILITY_IAM
        RoleArn: !GetAtt [CFNRole, Arn]
        StackName: StackB
        TemplateConfiguration: TemplateSource::test-configuration.json
        TemplatePath: TemplateSource::teststackB.yaml
        ParameterOverrides: |
          {
            "StackBInputParam" : { "Fn::GetParam" : ["StackAOutput", "TestOutput.json",
"StackAOutputName"]}
          }
      InputArtifacts:
        - Name: TemplateSource
        - Name: StackAOutput
      RunOrder: '1'
```

另請參閱

以下相關資源可協助您使用這些參數。

- 如需有關 CodePipeline 中 CloudFormation 動作參數的詳細資訊，請參閱《AWS CodePipeline 使用者指南》中的 [CloudFormation 部署動作組態參考](#)。
- 如需動作提供者的範例範本值 (例如 Owner 欄位或 configuration 欄位)，請參閱 AWS CodePipeline User Guide (《使用者指南》) 中的 [Action structure reference](#) (動作結構參考)。
- 若要下載 YAML 或 JSON 格式的管道堆疊範本範例，請參閱《AWS CodePipeline 使用者指南》中的 [教學課程：使用 CloudFormation 建立管道](#)。

CloudFormation 範本參考指南

建立 CloudFormation 範本所需的資源與元件詳細資訊，現可在 [CloudFormation 範本參考指南](#) 中查閱。本指南可做為範本的技術參考，並補充《CloudFormation 使用者指南》中的入門內容。

CloudFormation 範本參考指南提供關於以下項目的詳細資訊：

- 資源類型及其屬性
- DeletionPolicy 及其他資源層級屬性
- 內建函數，如 Ref 和 Sub
- 轉換 (CloudFormation 巨集)
- 用於 EC2 執行個體組態的協助程式指令碼
- 機器可讀的資源類型定義

本指南會定期更新，以反映下列內容：

- 新資源類型
- 屬性變更
- 新的傳回值
- 說明文件更新

中的安全性 CloudFormation

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間共同責任。[共同責任模式](#)將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。在[AWS 合規計畫](#)中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用的合規計劃 CloudFormation，請參閱[AWS 合規計劃的服務範圍](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規。

本文件可協助您了解如何在使用時套用共同責任模型 CloudFormation。下列主題說明如何設定 CloudFormation 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 CloudFormation 資源。

主題

- [防止 CloudFormation 堆疊遭到刪除](#)
- [避免更新堆疊資源](#)
- [中的資料保護 CloudFormation](#)
- [使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)
- [使用 記錄 CloudFormation API 呼叫 AWS CloudTrail](#)
- [中的基礎設施安全 CloudFormation](#)
- [中的彈性 CloudFormation](#)
- [的合規驗證 CloudFormation](#)
- [中的組態和漏洞分析 CloudFormation](#)
- [CloudFormation 的安全最佳實務](#)
- [CloudFormation 使用界面端點存取 \(AWS PrivateLink\)](#)

防止 CloudFormation 堆疊遭到刪除

您可以在堆疊上啟用終止保護來防止堆疊遭意外刪除。如果使用者嘗試刪除啟用終止保護的堆疊，則刪除會失敗，且堆疊及其狀態保持不變。您可以在建立新的堆疊時啟用終止保護。堆疊上的終止保護預設

為停用。除了 DELETE_IN_PROGRESS 或 DELETE_COMPLETE 以外，您可以為處於任何狀態的堆疊設定終止保護。

在堆疊上啟用或停用終止保護的同時，也將對屬於該堆疊的任何巢狀堆疊套用相同選擇。您無法直接在巢狀堆疊上啟用或停用終止保護。如果使用者嘗試直接刪除屬於已啟用終止保護之堆疊的巢狀堆疊，則操作會失敗，且巢狀堆疊會保持不變。

不過，如果使用者執行會刪除巢狀堆疊的堆疊更新，則 CloudFormation 會相應刪除巢狀堆疊。

終止保護與停用復原不同。終止保護僅適用於嘗試刪除堆疊，而停用復原適用於堆疊建立失敗時的自動復原。

在建立堆疊時啟用終止保護

在 Create stack (建立堆疊) 精靈的 Specify stack options (指定堆疊選項) 頁面上，展開 Advanced options (進階選項) 下的 Termination Protection (終止保護) 區段，然後選取 Enable (啟用)。如需詳細資訊，請參閱[設定堆疊選項](#)。

在現有堆疊上啟用或停用終止保護

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 選取您想要設定的堆疊。

Note

如果堆疊名稱旁顯示 NESTED (巢狀)，則該堆疊為巢狀堆疊。您只能在該巢狀堆疊所屬的根堆疊上啟用終止保護。

4. 在 stack details (堆疊詳細資訊) 窗格中，選取 Stack actions (堆疊動作)，然後選取 Edit termination protection (編輯終止保護)。

CloudFormation 會顯示 Edit termination protection (編輯終止保護) 對話方塊。

5. 選擇 Enable (啟用) 或 Disable (停用)，然後選取 Save (儲存)。

在巢狀堆疊上啟用或停用終止保護

如果堆疊名稱旁顯示 NESTED (巢狀)，則該堆疊為巢狀堆疊。您只能在該巢狀堆疊所屬的根堆疊上啟用終止保護。在根堆疊上變更終止保護：

1. 登入 AWS 管理主控台 並在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面上方的導覽列上，選擇 AWS 區域。
3. 選取您想要設定的巢狀堆疊。
4. 在 Stack info (堆疊資訊) 窗格的 Overview (概觀) 區段中，選取列為 Root stack (根堆疊) 的堆疊名稱。

CloudFormation 會顯示根堆疊的堆疊詳細資訊。

5. 選擇 Stack actions (堆疊動作)，然後選擇 Edit Termination Protection (編輯終止保護)。

CloudFormation 會顯示 Edit termination protection (編輯終止保護) 對話方塊。

6. 選擇 Enable (啟用) 或 Disable (停用)，然後選取 Save (儲存)。

欲使用命令列啟用或停用終止保護

使用 [update-termination-protection](#) 命令。

控制誰可以在堆疊上變更終止保護

若要在堆疊上啟用或停用終止保護，使用者需要具有進行 `cloudformation:UpdateTerminationProtection` 動作的許可。例如，下列政策可讓使用者在堆疊上啟用或停用終止保護。

如需在 中指定許可的詳細資訊 CloudFormation，請參閱 [使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

Example 授予變更堆疊終止保護許可的範例政策

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:UpdateTerminationProtection"
    ],
    "Resource": "*"
  }]
}
```



```
}
```

避免更新堆疊資源

當您建立堆疊時，可以對所有的資源執行所有更新動作。根據預設，有堆疊更新許可的任何人都可以更新堆疊中的所有資源。在更新期間，有些資源可能需要中斷或是完全替換，以致產生新的實體 ID 或全新的儲存體。您可避免在使用堆疊政策更新堆疊期間，無意中更新或刪除堆疊資源。堆疊政策是一種 JSON 文件，定義可在指定資源上執行的更新動作。

在您設定堆疊政策之後，堆疊中的所有資源預設都會受到保護。若要允許特定資源的更新，您可以為您堆疊政策中的這些資源指定明確的 Allow 陳述式。您每個堆疊只能定義一個堆疊政策，但可以保護單一政策內的多個資源。堆疊政策適用於所有嘗試更新堆疊的 CloudFormation 使用者。您無法建立不同堆疊政策與不同使用者的關聯。

堆疊政策僅適用於堆疊更新期間。此政策不提供類似 AWS Identity and Access Management (IAM) 政策的存取控制。堆疊政策僅作為容錯移轉的安全機制，以避免意外更新特定的堆疊資源。若要控制對 AWS 資源或動作的存取，請使用 IAM。

主題

- [範例堆疊政策](#)
- [定義堆疊政策](#)
- [設定堆疊政策](#)
- [更新受保護的資源](#)
- [修改堆疊政策](#)
- [更多範例堆疊政策](#)

範例堆疊政策

下列範例堆疊政策會避免更新 ProductionDatabase 資源：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

```
    },  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/ProductionDatabase"  
    }  
  ]  
}
```

在您設定堆疊政策時，所有資源預設都會受到保護。為允許更新所有資源，我們新增在所有資源上允許所有動作的 Allow 陳述式。雖然 Allow 陳述式指定所有的資源，但明確的 Deny 陳述式會針對邏輯 ID 為 ProductionDatabase 的資源覆寫它。此 Deny 陳述式會避免 ProductionDatabase 資源上所有的更新動作，例如替換或刪除。

Principal 元素是必要的，但只支援萬用字元 (*)，這表示此陳述式適用於所有[主體](#)。

Note

在堆疊更新期間，CloudFormation 會自動更新依賴其他已更新資源的資源。例如，CloudFormation 會更新參考已更新資源的資源。CloudFormation 不對自動更新的資源 (例如資源 ID) 執行任何實體變更，但如果堆疊政策與這些資源相關聯，您必須有進行更新的許可。

定義堆疊政策

當您建立堆疊時，未設定任何堆疊政策，所以可對所有的資源執行所有更新動作。為了保護堆疊資源不受更新動作影響，請定義堆疊政策，然後在您的堆疊上設定它。堆疊政策是一種 JSON 文件，可定義 CloudFormation 使用者可以執行的 CloudFormation 堆疊更新動作以及可以套用這些動作的資源。您在建立堆疊時，透過指定包含您堆疊政策的文字檔案或將它輸出的方式，設定堆疊政策。當您在堆疊上設定堆疊政策時，預設拒絕任何未明確允許的更新。

您可以使用五種元素定義堆疊政策：Effect、Action、Principal、Resource 和 Condition。下列虛擬程式碼顯示堆疊政策的語法。

```
{  
  "Statement" : [  
    {  
      "Effect" : "Deny_or-Allow",  
      "Action" : "update_actions",
```

```
"Principal" : "*",
"Resource" : "LogicalResourceId/resource_logical_ID",
"Condition" : {
  "StringEquals_or_StringLike" : {
    "ResourceType" : [resource_type, ...]
  }
}
}
```

Effect

決定在您指定的資源上拒絕或允許您指定的動作。您只能指定 Deny 或 Allow，例如：

```
"Effect" : "Deny"
```

Important

如果堆疊政策包含重疊的陳述式 (允許和拒絕對資源的更新)，則 Deny 陳述式一律會覆寫 Allow 陳述式。為了確保資源受到保護，請為該資源使用 Deny 陳述式。

Action

指定拒絕或允許的更新動作：

Update:Modify

指定在其執行期間，資源可能不會中斷，或套用變更時有些中斷的更新動作。所有資源都保持它們的實體 ID。

Update:Replace

指定在其執行期間，會重新建立資源的更新動作。CloudFormation 使用指定的更新建立新資源，然後刪除舊資源。由於資源是重新建立的，所以新資源的實體 ID 可能不同。

Update>Delete

指定在其執行期間，會移除資源的更新動作。從堆疊範本完全移除資源的更新，需要此動作。

更新:*

指定所有更新動作。星號是萬用字元，代表所有更新動作。

下列範例說明如何只指定取代和刪除動作：

```
"Action" : ["Update:Replace", "Update:Delete"]
```

若要允許某個除外的所有更新動作，請使用 `NotAction`。例如，若要允許 `Update:Delete` 除外的所有更新動作，請使用 `NotAction`，如這個範例所示：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

Principal

`Principal` 元素指定套用政策的實體。此元素是必要的，但只支援萬用字元 (*)，這表示政策適用於所有[主體](#)。

資源

指定套用政策的資源邏輯 ID。若要指定資源類型，請使用 `Condition` 元素。

若要指定單一資源，請使用它的邏輯 ID。例如：

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

邏輯 ID 可以使用萬用字元。例如，如果所有相關資源都使用常見的邏輯 ID 字首，您可以用萬用字元指定它們全部：

```
"Resource" : ["LogicalResourceId/CriticalResource*"]
```

資源也可以使用 `Not` 元素。例如，若要允許某個以外的所有資源更新，請使用 `NotResource` 元素來保護該資源：

```
{
```

```
"Statement" : [
  {
    "Effect" : "Allow",
    "Action" : "Update:*",
    "Principal": "*",
    "NotResource" : "LogicalResourceId/ProductionDatabase"
  }
]
```

當您設定堆疊政策時，預設拒絕任何未明確允許的更新。透過允許 ProductionDatabase 資源以外的所有資源更新，您拒絕更新 ProductionDatabase 資源。

條件

指定套用政策的資源類型。若要指定特定資源的邏輯 ID，請使用 Resource 元素。

您可以指定資源類型，例如所有 EC2 和 RDS 資料庫執行個體，如下列範例所示：

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Principal" : "*",
      "Action" : "Update:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Principal" : "*",
      "Action" : "Update:*",
      "Resource" : "*"
    }
  ]
}
```

Allow 陳述式將更新許可授予所有資源，而 Deny 陳述式拒絕更新 EC2 和 RDS 資料庫執行個體。Deny 陳述式一律會覆寫允許動作。

資源類型可以使用萬用字元。例如，您可以使用萬用字元拒絕所有 Amazon EC2 資源的更新許可——例如執行個體、安全群組和子網路——如下列範例所示：

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

使用萬用字元時，您必須使用 StringLike 條件。

設定堆疊政策

您可以使用主控台 AWS CLI 或在建立堆疊時套用堆疊政策。您也可以使用 AWS CLI 將堆疊政策套用到現有堆疊。套用堆疊政策後，您無法將其從堆疊中移除，但您可以使用 AWS CLI 進行修改。

堆疊政策適用於所有嘗試更新堆疊的 CloudFormation 使用者。您無法建立不同堆疊政策與不同使用者的關聯。

如需撰寫堆疊政策的資訊，請參閱[定義堆疊政策](#)。

在建立堆疊時設定堆疊政策 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 在畫面頂端的導覽列上，選擇要在其中 AWS 區域 建立堆疊的。
3. 在 CloudFormation Stacks (CloudFormation 堆疊) 頁面上，選擇 Create stack (建立堆疊)。
4. 在 Create Stack (建立堆疊) 精靈的 Configure stack options (設定堆疊選項) 頁面上，展開 Advanced (進階) 區段，然後選擇 Stack policy (堆疊政策)。
5. 指定堆疊政策：
 - 若要直接在主控台中編寫政策，請選擇 Enter stack policy (輸入堆疊政策)，然後直接在文字欄位中輸入堆疊政策。
 - 若要使用其他檔案定義的政策，請選擇 Upload a file (上傳檔案)，然後選擇 Choose file (選擇檔案)，選取包含堆疊政策的檔案。

在建立堆疊時設定堆疊政策 (AWS CLI)

- 使用 [create-stack](#) 命令搭配 `--stack-policy-body` 選項輸入修改的政策，或搭配 `--stack-policy-url` 選項指定包含政策的檔案。

在現有堆疊上設定堆疊政策 (AWS CLI 僅限)

- 使用 [set-stack-policy](#) 命令搭配 `--stack-policy-body` 選項輸入修改的政策，或搭配 `--stack-policy-url` 選項指定包含政策的檔案。

Note

若要將政策新增到現有的堆疊，您必須擁有 CloudFormation [SetStackPolicy](#) 動作的許可。

更新受保護的資源

若要更新受保護的資源，請建立覆寫堆疊政策的暫時政策，並允許更新這些資源。在更新堆疊時您指定覆寫政策。覆寫的政策不會永久變更堆疊政策。

若要更新受保護的資源，您必須有使用 CloudFormation [SetStackPolicy](#) 動作的許可。如需設定 CloudFormation 許可的資訊，請參閱[使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

Note

在堆疊更新期間，CloudFormation 會自動更新依賴其他已更新資源的資源。例如，CloudFormation 會更新參考已更新資源的資源。CloudFormation 不對自動更新的資源 (例如資源 ID) 執行任何實體變更，但如果堆疊政策與這些資源相關聯，您必須有進行更新的許可。

更新受保護的資源 (主控台)

1. 在 <https://console.aws.amazon.com/cloudformation> 開啟 CloudFormation 主控台。
2. 選取您希望更新的堆疊，選擇 Stack actions (堆疊動作)，然後選擇 Update stack (更新堆疊)。

3. 如「尚未」修改堆疊範本，請選取 Use current template (使用目前的範本)，然後按一下 Next (下一步)。如已修改範本，請選取 Replace current template (取代目前的範本)，然後在 Specify template (指定範本) 區段中指定更新的範本位置：
 - 如果是存放在本機電腦上的範本，請選取 Upload a template file (上傳範本檔案)。選擇 Choose File (選擇檔案) 以瀏覽至該檔案，然後按一下 Next (下一步)。
 - 如需存放在 Amazon S3 儲存貯體的範本，請選取 Amazon S3 URL。輸入或貼上範本的 URL，然後按一下 Next (下一步)。

如果您擁有已啟用版本控制的儲存貯體中的範本，則可以指定範本的特定版本，方法是將 `versionId=version-id` 附加到 URL。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[使用已啟用版本控制之儲存貯體中的物件](#)。

4. 如果您的範本包含參數，請在 Specify stack details (指定堆疊詳細資訊) 頁面上輸入或修改參數值，然後選擇 Next (下一步)。

CloudFormation 以堆疊中目前設定的值填入每個參數，但以 NoEcho 屬性宣告的參數除外。您可以選擇 Use existing value (使用現有的值)，為這些參數使用目前的值。

如需使用 NoEcho 遮罩敏感資訊，以及使用動態參數管理密碼的詳細資訊，請參閱 [請勿在您的範本中內嵌憑證](#) 最佳實務。

5. 指定覆寫堆疊政策。
 - a. 在 Configure stack options (設定堆疊選項) 頁面的 Advanced options (進階選項) 區段中，選取 Stack policy (堆疊政策)。
 - b. 選取 Upload a file (上傳檔案)。
 - c. 按一下 Choose file (選擇檔案)，導覽至包含覆寫堆疊政策的檔案，或輸入政策。
 - d. 選擇下一步。

覆寫政策必須為您要更新的受保護資源指定 Allow 陳述式。例如，若要更新所有受保護的資源，請指定允許所有更新的暫時覆寫政策：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```



```
}  
]  
}
```

Note

CloudFormation 只會在此更新期間套用覆寫政策。覆寫的政策不會永久變更堆疊政策。若要修改堆疊政策，請參閱[修改堆疊政策](#)。

6. 檢閱堆疊資訊和您提交的變更。

檢查是否提交了正確的資訊，例如正確的參數值或範本 URL。如果您的範本包含 IAM 資源，請選擇 I acknowledge that this template may create IAM resources (我知道此範本可能會建立 IAM 資源) 以指定您要使用此範本中的 IAM 資源。如需詳細資訊，請參閱[認可 CloudFormation 範本中的 IAM 資源](#)。

在預覽您的變更區段中，檢查 CloudFormation 是否將會進行您預期的所有變更。例如，檢查 CloudFormation 是否會新增、移除及修改您要新增、移除或修改的資源。CloudFormation 透過建立針對堆疊設定的變更，產生此預覽。如需詳細資訊，請參閱[透過變更集更新 CloudFormation 堆疊](#)。

7. 當您完成變更後，請按一下 Update (更新)。

Note

此時，您也可以選擇檢視變更集，更仔細地檢閱您提出的更新。若要執行此動作，請按一下 View change set (檢視變更集) 而不是 Update (更新)。CloudFormation 會顯示根據更新所產生的變更集。當您準備好執行堆疊更新時，請按一下 Execute (執行)。

CloudFormation 會顯示您堆疊的 Stack details (堆疊詳細資訊) 頁面。您的堆疊目前的狀態為 UPDATE_IN_PROGRESS。CloudFormation 成功更新堆疊之後，就會將堆疊狀態設為 UPDATE_COMPLETE。

如果堆疊更新失敗，CloudFormation 會自動復原變更，並將堆疊狀態設為 UPDATE_ROLLBACK_COMPLETE。

更新受保護的資源 (AWS CLI)

- 使用 [update-stack](#) 命令搭配 `--stack-policy-during-update-body` 選項輸入修改的政策，或搭配 `--stack-policy-during-update-url` 選項指定包含政策的檔案。

Note

CloudFormation 只會在此更新期間套用覆寫政策。覆寫的政策不會永久變更堆疊政策。若要修改堆疊政策，請參閱[修改堆疊政策](#)。

修改堆疊政策

若要保護其他資源或移除對資源的保護，請修改堆疊政策。例如，當您將想要保護的資料庫新增到您的堆疊時，請為該資料庫將 Deny 陳述式新增到堆疊政策。若要修改政策，您必須有使用 [SetStackPolicy](#) 動作的許可。

使用 AWS CLI 修改堆疊政策。

修改堆疊政策 (AWS CLI)

- 使用 [set-stack-policy](#) 命令搭配 `--stack-policy-body` 選項輸入修改的政策，或搭配 `--stack-policy-url` 選項指定包含政策的檔案。

您無法刪除堆疊政策。若要移除所有資源的所有保護，您要修改政策，明確允許對所有資源的所有動作。下列政策允許所有資源的所有更新：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

更多範例堆疊政策

下列範例政策示範如何防止更新所有堆疊資源和特定資源，並防止特定類型的更新。

避免更新所有堆疊資源

為了防止更新所有堆疊資源，下列政策為所有資源的所有更新動作指定 Deny 陳述式。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

避免更新單一資源

下列政策拒絕邏輯 ID 為 MyDatabase 之資料庫的所有更新動作。Allow 陳述式允許所有其他堆疊資源的所有更新動作。Allow 陳述式不適用於 MyDatabase 資源，因為 Deny 陳述式一律會覆寫允許動作。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

使用預設拒絕可以達到和前例相同的結果。當您設定堆疊政策時，CloudFormation 拒絕任何未明確允許的更新。下列政策允許更新所有資源，預設拒絕的 `ProductionDatabase` 資源除外。

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

Important

使用預設拒絕有風險。如果政策的其他位置有一個 `Allow` 陳述式 (例如使用萬用字元的 `Allow` 陳述式)，您可能會在不知情的狀況下將更新許可授予您不打算更新的資源。由於明確拒絕會覆寫任何允許動作，所以您可以使用 `Deny` 陳述式確保資源受到保護。

避免更新某種資源類型的所有執行個體

下列政策拒絕 RDS 資料庫執行個體資源類型的所有更新動作。`Allow` 陳述式允許所有其他堆疊資源的所有更新動作。`Allow` 陳述式不適用於 RDS 資料庫執行個體資源，因為 `Deny` 陳述式一律會覆寫允許動作。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
```

```
    "Effect" : "Allow",
    "Action" : "Update:*",
    "Principal": "*",
    "Resource" : "*"
  }
]
```

避免更換執行個體的更新

下列政策拒絕會造成更換邏輯 ID 為 MyInstance 之執行個體的更新。Allow 陳述式允許所有其他堆疊資源的所有更新動作。Allow 陳述式不適用於 MyInstance 資源，因為 Deny 陳述式一律會覆寫允許動作。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

避免巢狀堆疊的更新

下列政策拒絕 CloudFormation 堆疊資源類型 (巢狀堆疊) 的所有更新動作。Allow 陳述式允許所有其他堆疊資源的所有更新動作。Allow 陳述式不適用於 CloudFormation 堆疊資源，因為 Deny 陳述式一律會覆寫允許動作。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
```

```
    "Action" : "Update:*",
    "Principal": "*",
    "Resource" : "*",
    "Condition" : {
        "StringEquals" : {
            "ResourceType" : ["AWS::CloudFormation::Stack"]
        }
    }
},
{
    "Effect" : "Allow",
    "Action" : "Update:*",
    "Principal": "*",
    "Resource" : "*"
}
]
```

中的資料保護 CloudFormation

AWS [共同責任模型](#)適用於 中的資料保護 AWS CloudFormation。如此模型所述，AWS 負責保護執行所有 的全域基礎設施 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 共同責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 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 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《使用者指南》中的[使用 CloudTrail 追蹤](#)。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 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 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，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 \(FIPS\) 140-3](#)。

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 CloudFormation 或使用主控台、API AWS CLI 或其他 AWS 服務 AWS SDKs 時。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。

靜態加密

遵循 AWS 共同責任模型，會 CloudFormation 儲存靜態加密的資料。客戶必須負責為儲存在其帳戶中的資料設定加密和儲存政策。例如，建議您針對儲存在 S3 儲存貯體或 SNS 主題中的範本和其他資料啟用靜態加密。客戶也可以為 CloudFormation 佈建的任何資料儲存系統定義加密設定。

傳輸中加密

CloudFormation 在共同責任模型下，使用加密通道進行服務通訊。

網際網路流量隱私權

CloudFormation 服務通訊預設會在區域或可用區域之間安全加密。

使用 控制 CloudFormation 存取 AWS Identity and Access Management

透過 AWS Identity and Access Management (IAM)，您可以建立 IAM 使用者，並控制他們對中特定資源的存取 AWS 帳戶。使用 IAM 時，可控制使用者可以使用 CloudFormation 做什麼，例如他們是否可以檢視堆疊範本、建立堆疊或刪除堆疊。

除了 CloudFormation 特定的動作之外，您還可以管理每個使用者可用的 AWS 服務和資源。如此一來，您就可以控制使用者在使用 CloudFormation 時可以存取哪些資源。例如，您可以指定哪些使用者可以建立 Amazon EC2 執行個體、終止資料庫執行個體，或更新 VPC。只要他們使用 CloudFormation 執行這些動作，就會套用這些相同許可。

使用下列各節中的資訊來控制誰可以存取 CloudFormation。我們也將探索如何在範本中授權建立 IAM 資源、給予在 EC2 執行個體上執行的應用程式所需的許可，以及在您的 AWS 環境中使用臨時安全登入資料來增強安全性。

定義 IAM 基於身分識別的政策

若要授予對 CloudFormation 的存取權，您需要建立和指派 IAM 政策，以授予您的 IAM 身分 (例如使用者或角色) 呼叫其所需 API 動作的許可。

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。CloudFormation 支援特定動作、資源和條件金鑰。

如果您是初次使用 IAM，請先自行熟悉 IAM JSON 政策的元素。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素參考](#)。若要了解如何建立 IAM 政策，請完成 IAM 文件中的教學課程[建立並連接您的第一個客戶管理政策](#)。

主題

- [CloudFormation 的政策動作](#)
- [CloudFormation 的主控制台特定動作](#)
- [CloudFormation 的政策資源](#)
- [CloudFormation 的政策條件索引鍵](#)

CloudFormation 的政策動作

在 IAM 政策陳述式的 Action 元素中，您可以指定 CloudFormation 提供的任何 API 動作。必須在動作名稱前面加上小寫字串 `cloudformation:`。例如：`cloudformation:CreateStack`、`cloudformation:CreateChangeSet` 和 `cloudformation:UpdateStack`。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

```
"Action": [ "cloudformation:action1", "cloudformation:action2" ]
```

您也可以使用萬用字元指定多個動作。例如，您可以指定名稱開頭為 `Get` 文字的所有動作，如下所示：

```
"Action": "cloudformation:Get*"
```

若要查看與 `cloudformation` 服務字首相關聯的動作完整清單，請參閱服務授權參考中的[適用於 CloudFormation 的動作、資源和條件索引鍵](#)以及[適用於 AWS 雲端控制 API 的動作、資源和條件索引鍵](#)。

範例

以下顯示授予檢視 CloudFormation 堆疊許可的許可政策範例。

Example 1: 授予檢視堆疊許可的範例政策

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:DescribeStackResources"
    ],
    "Resource": "*"
  }]
}
```

建立或刪除堆疊的使用者需要基於其堆疊範本的其他許可。例如，如果您的範本描述 Amazon SQS 佇列，則使用者必須同時擁有 CloudFormation 和 Amazon SQS 動作的許可，如下列範例政策所示。

Example 2: 授予建立和檢視堆疊動作及所有 Amazon SQS 動作的範例政策

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "sqs:*",
      "cloudformation:CreateStack",
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResources",
      "cloudformation:GetTemplate",
      "cloudformation:ValidateTemplate"
    ],
    "Resource": "*"
  }]
}
```

```
}
```

CloudFormation 的主控台特定動作

CloudFormation 主控台的使用者需要 AWS Command Line Interface 或 CloudFormation APIs 所需的額外許可。這些額外許可支援主控台特定的功能，例如範本上傳至 Amazon S3 儲存貯體，以及 AWS 特定參數類型的下拉式清單。

針對以下列出的所有動作，向所有資源授予許可；不要將其限制在特定堆疊或儲存貯體。

以下動作僅供 CloudFormation 主控台使用，不會收錄在 API 參考中。此動作允許使用者將範本上傳至 Amazon S3 儲存貯體。

- `cloudformation:CreateUploadBucket`

當使用者上傳範本時，他們也需要下列 Amazon S3 許可：

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetObject`
- `s3:CreateBucket`

若要查看具有 AWS 特定參數類型的範本參數下拉式清單中的值，使用者需要許可才能進行對應的描述 API 呼叫。例如，當範本中使用這些參數類型時，需要下列許可：

- `ec2:DescribeKeyPairs` – `AWS::EC2::KeyPair::KeyName` 參數類型的必要項目。
- `ec2:DescribeSecurityGroups` – `AWS::EC2::SecurityGroup::Id` 參數類型的必要項目。
- `ec2:DescribeSubnets` – `AWS::EC2::Subnet::Id` 參數類型的必要項目。
- `ec2:DescribeVpcs` – `AWS::EC2::VPC::Id` 參數類型的必要項目。

如需 AWS 特定參數類型的詳細資訊，請參閱 [使用 CloudFormation 提供的參數類型，在執行時期指定現有資源](#)。

CloudFormation 的政策資源

在 IAM 政策陳述式中，Resource 元素指定陳述式所涵蓋的一個或多個物件。對於 CloudFormation，每個 IAM 政策陳述式都會套用到您使用其 Amazon Resource Name (ARN) 指定的資源。特定 ARN 格式視資源而定。

如需 CloudFormation 資源類型及其 ARN 的完整清單，請參閱《服務授權參考》中的 [CloudFormation 定義的資源類型](#)。若要了解您可以使用每個資源的 ARN 指定哪些動作，請參閱 [CloudFormation 定義的動作](#)。

可以指定特定堆疊的動作，如下列政策範例所示。當您提供 ARN 時，請將 *placeholder text* 取代為您的資源特定資訊。

Example 1: 拒絕刪除和更新指定堆疊動作的範例政策

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloudformation:DeleteStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:111122223333:stack/MyProductionStack/*"
    }
  ]
}
```

上例中的政策在堆疊名稱結尾處使用萬用字元，所以全堆疊 ID (例如 `arn:aws:cloudformation:region:account-id:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c`) 和堆疊名稱 (例如 `MyProductionStack`) 會拒絕刪除堆疊和更新堆疊。

若要讓 AWS::Serverless 轉換來建立變更集，應包含 `arn:aws:cloudformation:region:aws:transform/Serverless-2016-10-31` 資源層級許可，如下列政策所示。

Example 2: 允許針對指定轉換建立變更集動作的範例政策

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "arn:aws:cloudformation:us-east-1:aws:transform/Serverless-2016-10-31"
    }
  ]
}
```

CloudFormation 的政策條件索引鍵

在 IAM 政策陳述式中，您可以選擇性地指定控制政策生效時機的條件。例如，您可以定義一項政策，讓使用者只能在他們指定特定範本 URL 的時候建立堆疊。您可以定義 CloudFormation 特定條件和 AWS 整體條件，例如 `DateLessThan`，指定政策何時停止生效。如需詳細資訊和 AWS 整體條件清單，請參閱 [《IAM 使用者指南》中的 IAM 政策元素參考](#) 中的條件。

Note

請勿使用 `aws:SourceIp` AWS 通用條件。CloudFormation 會使用自己的 IP 位址而非產生請求的 IP 位址來佈建資源。例如，當您建立堆疊時，CloudFormation 會從自己的 IP 位址提出請求，以啟動 Amazon EC2 執行個體或建立 Amazon S3 儲存貯體，而不是從 `CreateStack` 呼叫或 `create-stack` 命令中的 IP 位址。

下列清單說明 CloudFormation 專屬條件。這些條件只適用於使用者建立或更新堆疊時：

`cloudformation:ChangeSetName`

您想要與政策建立關聯的 CloudFormation 變更集名稱。使用此條件控制使用者可以執行或刪除的變更集。

cloudformation:ImportResourceTypes

您希望與政策建立關聯的範本資源類型，例如 `AWS::EC2::Instance`。使用此條件來控制 使用者在將資源匯入堆疊時，能夠使用的資源類型。此條件會針對使用者在 `ResourcesToImport` 參數中宣告的資源類型進行檢查，目前僅支援 AWS CLI 和 API 請求。使用此參數時，您必須指定您希望使用者在匯入操作期間控制的所有資源類型。如需 `ResourcesToImport` 參數的詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateChangeSet](#) 動作。

如需可能的 `ResourcesToImport` 清單，請參閱 [資源類型支援](#)。

使用三段式資源命名慣例，以指定使用者可使用的資源類型，範圍大至整個組織的所有資源，小至個別的資源類型。

*organization::**

指定給定組織的所有資源類型。

*organization::service_name::**

指定給定組織內特定服務的所有資源類型。

organization::service_name::resource_type

指定特定的資源類型。

例如：

*AWS::**

指定所有支援 AWS 的資源類型。

*AWS::service_name::**

指定特定 AWS 服務的所有支援資源。

AWS::service_name::resource_type

指定特定 AWS 資源類型，例如 `AWS::EC2::Instance` (所有 EC2 執行個體) 。

cloudformation:ResourceTypes

您希望與政策建立關聯的範本資源類型，例如 `AWS::EC2::Instance`。使用此條件控制 使用者在建立或更新堆疊時可以使用的資源類型。此條件的檢查目標是使用者在 `ResourceTypes` 參數中宣告的資源類型，目前僅支援 AWS CLI 和 API 請求。使用此參數時，使用者必須指定其範本中的所

有資源類型。如需 ResourceTypes 參數的詳細資訊，請參閱《AWS CloudFormation API 參考》中的 [CreateStack](#) 動作。

如需資源類型的清單，請參閱 [CloudFormation 範本參考指南](#)。

使用三段式資源命名慣例，以指定使用者可使用的資源類型，範圍大至整個組織的所有資源，小至個別的資源類型。

*organization::**

指定給定組織的所有資源類型。

*organization::service_name::**

指定給定組織內特定服務的所有資源類型。

organization::service_name::resource_type

指定特定的資源類型。

例如：

*AWS::**

指定所有支援 AWS 的資源類型。

*AWS::service_name::**

指定特定 AWS 服務的所有支援資源。

AWS::service_name::resource_type

指定特定 AWS 資源類型，例如 *AWS::EC2::Instance*（所有 EC2 執行個體）。

*Alexa::ASK::**

指定 Alexa Skill Kit 中的所有資源類型。

Alexa::ASK::Skill

指定個別 [Alexa::ASK::Skill](#) 資源類型。

*Custom::**

指定所有自訂資源。

如需詳細資訊，請參閱[使用自訂資源建立自訂佈建邏輯](#)。

Custom::*resource_type*

指定特定自訂資源類型。

如需詳細資訊，請參閱[使用自訂資源建立自訂佈建邏輯](#)。

cloudformation:RoleARN

您想要與政策建立關聯之 IAM 服務角色的 Amazon Resource Name (ARN)。使用此條件控制 使用者在使用堆疊或變更集時可以使用的服務角色。

cloudformation:StackPolicyUrl

您想要與政策建立關聯的 Amazon S3 堆疊政策 URL。使用此條件控制 使用者可在建立或更新堆疊動作期間，與堆疊建立關聯的堆疊政策。如需堆疊政策的詳細資訊，請參閱[避免更新堆疊資源](#)。

Note

為確保使用者只能使用您上傳的堆疊政策建立或更新堆疊，請針對這些使用者將 S3 儲存貯體設定為唯讀。

cloudformation:TemplateUrl

您想要與政策建立關聯的 Amazon S3 範本 URL。使用此條件控制 使用者在建立或更新堆疊時可以使用的範本。

Note

為確保使用者只能使用您上傳的範本建立或更新堆疊，請針對這些使用者將 S3 儲存貯體設定為唯讀。

Note

下列 CloudFormation 特定條件適用於相同名稱的 API 參數：

- `cloudformation:ChangeSetName`
- `cloudformation:RoleARN`
- `cloudformation:StackPolicyUrl`

- `cloudformation:TemplateUrl`

例如，`cloudformation:TemplateUrl` 僅適用於 `CreateStack`、`UpdateStack`、和 `CreateChangeSet` API 的 `TemplateUrl` 參數。

如需使用條件索引鍵來控制存取的 IAM 政策範例，請參閱 [CloudFormation 的 IAM 基於身分識別的 policy 範例](#)。

認可 CloudFormation 範本中的 IAM 資源

CloudFormation 會先驗證您的範本，您才可以建立堆疊。在驗證期間，CloudFormation 會檢查您的範本是否有它可能建立的 IAM 資源。IAM 資源 (例如具有完全存取權的使用者) 可以存取和修改您的 AWS 帳戶中的任何資源。因此，我們建議您先檢閱與每個 IAM 資源相關聯的許可再繼續操作，以免無意中建立提升許可的資源。為確保您已如此做，您必須認可範本包含這些資源，在 CloudFormation 建立堆疊前先授予其指定的功能。

您可以使用 CloudFormation 主控台、AWS Command Line Interface (AWS CLI) 或 API 來確認 CloudFormation 範本的功能：

- 在 CloudFormation 主控台，在「建立堆疊」或「更新堆疊」精靈的設定堆疊選項頁面中，選擇我知道此範本可能會建立 IAM 資源。
- 在 AWS CLI，當您使用 [create-stack](#) 和 [update-stack](#) 命令時，請指定 `--capabilities` 選項的 `CAPABILITY_IAM` 或 `CAPABILITY_NAMED_IAM` 值。如果您的範本包含 IAM 資源，您可以指定任一功能。如果您的範本包含 IAM 資源的自訂名稱，您必須指定 `CAPABILITY_NAMED_IAM`。
- 在 API 中，當您使用 [CreateStack](#) 和 [UpdateStack](#) 動作時，請指定 `Capabilities.member.1=CAPABILITY_IAM` 或 `Capabilities.member.1=CAPABILITY_NAMED_IAM`。如果您的範本包含 IAM 資源，您可以指定任一功能。如果您的範本包含 IAM 資源的自訂名稱，您必須指定 `CAPABILITY_NAMED_IAM`。

Important

如果您的範本包含自訂的具名 IAM 資源，請不要重複使用相同的範本建立多個堆疊。IAM 資源在您的帳戶中必須是全域唯一的。如果您在不同的區域使用相同的範本建立多個堆疊，您的堆疊可能會共用相同的 IAM 資源，而不是一個堆疊獨有一個資源。在堆疊中共享資源可能會產生各種無法恢復的意外結果。例如，如果您刪除或更新一個堆疊中的共享 IAM 資源，您會在無意中修改其他堆疊的資源。

管理在 Amazon EC2 執行個體上執行之應用程式的憑證

如果您有應用程式在 Amazon EC2 執行個體上執行，且需要向 AWS 資源 (例如 Amazon S3 儲存貯體或 DynamoDB 資料表) 提出請求，則此應用程式需要 AWS 安全憑證。不過，在每個您啟動的執行個體中分發和內嵌長期的安全憑證，是一項挑戰，同時可能有安全風險。與其使用像 IAM 使用者憑證的長期憑證，我們建議您在執行個體啟動時，建立與 Amazon EC2 執行個體相關聯的 IAM 角色。然後，應用程式就可以從 Amazon EC2 執行個體取得暫時的安全憑證。您不必在執行個體上內嵌長期的憑證。此外，為使管理憑證變得更輕鬆，您可以為多個 Amazon EC2 執行個體僅指定單一角色，不必為每個執行個體建立唯一的憑證。

如需示範如何使用角色啟動執行個體的範本程式碼片段，請參閱 [IAM 角色範本範例](#)。

Note

使用暫時安全登入資料之執行個體上的應用程式，可以呼叫任意 CloudFormation 動作。不過，由於 CloudFormation 與許多其他 AWS 服務互動，所以您必須確認您想要使用的所有服務，都支援暫時安全憑證。如需可接受臨時安全憑證的服務清單，請參閱《IAM 使用者指南》中的 [可搭配 IAM 運作的AWS 服務](#)。

授予暫時存取 (聯合身分存取)

在某些情況下，您可能想要授予沒有 AWS 登入資料的使用者暫時存取您的 AWS 帳戶。當您想要授予暫時存取權時，不要建立和刪除長期登入資料，而是使用 AWS Security Token Service (AWS STS)。例如，您可以使用 IAM 角色。從一個 IAM 角色，您可以透過程式設計方式建立，然後分發許多暫時安全憑證 (包括存取金鑰、私密存取金鑰和安全權杖)。這些登入資料生命週期有限，因此在過期 AWS 帳戶後就無法用來存取您的。您也可以建立多個 IAM 角色，以授與個別使用者不同層級的許可。IAM 角色對聯合身分和單一登入等案例很有用。

聯合身分是可以跨多個系統使用的獨特身分。對於已建立內部部署身分系統 (如 LDAP 或 Active Directory) 的企業使用者，您可用您的現場部署身分系統處理所有身分驗證。驗證完使用者身分後，您會從適當的 IAM 使用者或角色提供暫時安全憑證。例如，您可以建立管理員角色和開發人員角色，其中管理員可以完整存取 AWS 帳戶，而且開發人員只能使用 CloudFormation 堆疊。驗證管理員身分後，就會授權管理員從管理員角色中取得暫時安全憑證。不過，開發人員只能從開發人員角色中取得暫時安全憑證。

您也可以授予聯合身分使用者對的存取權 AWS 管理主控台。在使用者向您的內部部署身分系統驗證後，您可以透過程式設計方式建構暫時的 URL，藉以直接存取 AWS 管理主控台。當使用者使用暫時

URL 時，他們不需要登入，AWS 因為他們已經過身分驗證（單一登入）。此外，因為 URL 是從使用者的暫時安全憑證所建構，所以這些憑證可用的許可會決定使用者在 AWS 管理主控台許可。

您可以使用數個不同的 AWS STS APIs 來產生臨時安全登入資料。如需有關使用何種 API 的詳細資訊，請參閱《IAM 使用者指南》中的[比較 AWS STS 憑證](#)。

Important

當您使用 `GetFederationToken` API 產生的暫時安全憑證時，您不能使用 IAM。如果您需要使用 IAM，請改用來自角色的暫時安全憑證。

CloudFormation 會與許多其他 AWS 服務互動。當您搭配使用暫時安全憑證與 CloudFormation 時，請確認您想要使用的所有服務都支援暫時安全憑證。如需可接受臨時安全憑證的服務清單，請參閱《IAM 使用者指南》中的[可搭配 IAM 運作的 AWS 服務](#)。

如需詳細資訊，請參閱《IAM 使用者指南》中的下列相關資源：

- [臨時登入資料的常見案例](#)
- [啟用 AWS 主控台的自訂身分代理程式存取](#)

CloudFormation 的 IAM 基於身分識別的政策範例

依預設，使用者和角色不具備建立或修改 CloudFormation 資源的許可。他們也無法使用 AWS 管理主控台、AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。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。如需詳細資訊，請參閱[定義 IAM 基於身分識別的政策](#)。

下列範例顯示政策陳述式，可用於允許或拒絕使用一項或多項 CloudFormation 動作的許可。

主題

- [需要特定的範本 URL](#)
- [拒絕所有 CloudFormation 匯入操作](#)
- [允許特定資源類型的匯入操作](#)
- [拒絕堆疊範本中的 IAM 資源](#)
- [允許使用特定資源類型建立堆疊](#)
- [根據資源變動 API 動作來控制存取權](#)

- [根據區域和資源類型限制堆疊集操作](#)
- [允許所有 IaC 產生器操作](#)

需要特定的範本 URL

以下政策授予許可只使用 *https://s3.amazonaws.com/amzn-s3-demo-bucket/test.template* 範本 URL 建立或更新堆疊。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudformation:TemplateUrl": [
            "https://s3.amazonaws.com/amzn-s3-demo-bucket/
test.template"
          ]
        }
      }
    }
  ]
}
```

拒絕所有 CloudFormation 匯入操作

以下政策會授予使用者許可完成所有 CloudFormation 操作，但匯入操作「除外」。

JSON

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowAllStackOperations",
    "Effect": "Allow",
    "Action": "cloudformation:*",
    "Resource": "*"
  },
  {
    "Sid": "DenyImport",
    "Effect": "Deny",
    "Action": "cloudformation:*",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringLike": {
        "cloudformation:ImportResourceTypes": [
          "*"
        ]
      }
    }
  }
]
```

允許特定資源類型的匯入操作

以下政策授予所有堆疊操作的許可，以及僅限指定資源上的匯入操作 (在此範例中為 `AWS::S3::Bucket`)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImport",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
```

```

        "cloudformation:ImportResourceTypes": [
            "AWS::S3::Bucket"
        ]
    }
}
]
}

```

拒絕堆疊範本中的 IAM 資源

以下政策授予許可以建立堆疊，但如果堆疊範本包含任何來自 IAM 服務的資源，則拒絕請求。該政策也需要使用者指定 `ResourceTypes` 參數，僅供 AWS CLI 和 API 請求使用。此政策使用明確拒絕陳述式，因此，如有其他政策授予額外許可，此政策一律有效 (明確拒絕陳述式一律覆寫明確允許陳述式)。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringLikeIfExists": {
          "cloudformation:ResourceTypes": [ "AWS::IAM::*" ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {

```

```

        "Null": {
            "cloudformation:ResourceTypes": "true"
        }
    }
}
]
}

```

允許使用特定資源類型建立堆疊

以下政策類似前述範例。此政策授予許可可以建立堆疊，除非堆疊範本包含任何來自 IAM 服務的資源。它還要求使用者指定 參數，該ResourceTypes參數僅適用於 AWS CLI 和 API 請求。此政策更為簡單，但不使用明確拒絕陳述式。授予額外許可的其他政策，可以覆寫此政策。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "cloudformation:CreateStack" ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringNotLikeIfExists" : {
          "cloudformation:ResourceTypes" : [ "AWS::IAM::*" ]
        },
        "Null": {
          "cloudformation:ResourceTypes": "false"
        }
      }
    }
  ]
}

```

根據資源變動 API 動作來控制存取權

下列政策授予依據資源變更 API 動作名稱篩選存取權的許可。這用於控制 IAM 使用者可使用哪些 API 來在堆疊或堆疊集上新增或移除標籤。用於新增或移除標籤的操作應做為條件索引鍵的值新增。下列政策會授予 TagResource 和 UntagResource 變更操作 CreateStack 的許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CreateActionConditionPolicyForTagUntagResources",
    "Effect": "Allow",
    "Action": [
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudformation:CreateAction": [
          "CreateStack"
        ]
      }
    }
  ]
}
```

根據區域和資源類型限制堆疊集操作

下列政策授與服務受管堆疊集許可。只有當堆疊集的範本包含 Amazon S3 資源類型 (AWS::S3::*) 或 AWS::SES::ConfigurationSet 資源類型時，具有此政策的使用者才能在堆疊集上執行操作。如果使用 ID 登入組織管理帳戶 123456789012，使用者也只能對以 ID 為的 OU 為目標的堆疊集執行操作 *ou-1fsfsrsdsfrewr*，而且只能對以 ID AWS 帳戶 為目標 stack-set-id 的堆疊集執行操作 *987654321012*。

如果堆疊集範本包含的資源類型不是政策中指定的資源類型，或部署目標不是政策中為對應的管理帳戶和堆疊集所指定的 OU 或帳戶 ID，則堆疊集操作會失敗。

只有在堆疊集操作以 us-east-1、us-west-2 或 eu-west-2 AWS 區域為目標時，才會套用這些政策限制。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stackset/*",
        "arn:aws:cloudformation:*:*:type/resource/AWS-S3-*",
        "arn:aws:cloudformation:us-west-2:111122223333:type/resource/AWS-SES-ConfigurationSet",
        "arn:aws:cloudformation:*:111122223333:stackset-target/*/ou-1fsfsrdsdfrewr",
        "arn:aws:cloudformation:*:111122223333:stackset-target/stack-set-id/444455556666"
      ],
      "Condition": {
        "ForAllValues:StringEqualsIgnoreCase": {
          "cloudformation:TargetRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    }
  ]
}
```

允許所有 IaC 產生器操作

下列政策允許存取與 IaC 產生器資源掃描和範本管理相關的 CloudFormation 動作。第一個陳述式會授予描述、列出和啟動資源掃描的許可。它也允許存取其他必要的許可 (cloudformation:GetResource、cloudformation:ListResources 和

cloudformation:ListTypes), 這些許可可讓 IaC 產生器擷取有關資源和可用資源類型的資訊。第二個陳述式會授予建立、刪除、描述、列出和更新產生範本的完整許可。

您也必須將目標 AWS 服務的讀取許可授予使用 IaC 產生器掃描資源的任何人。如需詳細資訊, 請參閱[掃描資源所需的 IAM 許可](#)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ResourceScanningOperations",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeResourceScan",
        "cloudformation:GetResource",
        "cloudformation:ListResources",
        "cloudformation:ListResourceScanRelatedResources",
        "cloudformation:ListResourceScanResources",
        "cloudformation:ListResourceScans",
        "cloudformation:ListTypes",
        "cloudformation:StartResourceScan"
      ],
      "Resource": "*"
    },
    {
      "Sid": "TemplateGeneration",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateGeneratedTemplate",
        "cloudformation>DeleteGeneratedTemplate",
        "cloudformation:DescribeGeneratedTemplate",
        "cloudformation:GetResource",
        "cloudformation:GetGeneratedTemplate",
        "cloudformation:ListGeneratedTemplates",
        "cloudformation:UpdateGeneratedTemplate"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS 的 受管政策 AWS CloudFormation

AWS 受管政策是由 建立和管理的獨立政策 AWS。AWS 受管政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住，AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義特定於使用案例的[客戶管理政策](#)，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新 AWS 受管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。當新 AWS 服務啟動或新 API 操作可用於現有服務時，AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#)。

AWS 受管政策：AWSCloudFormationFullAccess

您可以將 AWSCloudFormationFullAccess 連接至使用者、群組與角色。

此政策會授予許可，以允許完整存取 CloudFormation 動作和資源。

許可詳細資訊

此政策包含以下許可。

- `cloudformation` – 允許主體對所有資源執行所有 CloudFormation 動作。

若要檢視此政策的許可，請參閱《AWS 受管政策參考指南》中的 [AWSCloudFormationFullAccess](#)。

AWS 受管政策：AWSCloudFormationReadOnlyAccess

您可以將 AWSCloudFormationReadOnlyAccess 連接至使用者、群組與角色。

此政策授予許可，允許唯讀存取 CloudFormation 資源和動作。

許可詳細資訊

此政策包含以下許可。

- `cloudformation` – 允許主體執行唯讀 CloudFormation 動作，例如描述堆疊、列出資源和檢視範本，但不允許建立、更新或刪除堆疊。

若要檢視此政策的許可，請參閱《AWS 受管政策參考指南》中的 [AWSCloudFormationReadOnlyAccess](#)。

AWS 受管政策的 CloudFormation 更新

檢視自此服務開始追蹤這些變更以來CloudFormation AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 CloudFormation 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	Date
AWSCloudFormationReadOnlyAccess – 更新現有政策	CloudFormation 新增了新的許可，以允許批次描述操作cloudformation:BatchDescribe* 的動作。	2026 年 1 月 30 日
AWSCloudFormationReadOnlyAccess – 更新現有政策	CloudFormation 新增了新的許可，以允許堆疊偏離偵測功能cloudformation:Detect* 的動作。	2019 年 11 月 13 日
AWSCloudFormationReadOnlyAccess – 更新現有政策	CloudFormation 新增了允許 cloudformation:EstimateTemplateCost 、 cloudformation:Get* 和 cloudformation:ValidateTemplate 動作的許可。	2017 年 11 月 2 日
AWSCloudFormationFullAccess – 新政策	CloudFormation 新增了新的 AWS 受管政策，提供 CloudFormation 動作和資源的完整存取權。	2019 年 7 月 26 日
AWSCloudFormationReadOnlyAccess – 更新現有政策	CloudFormation 新增了允許 cloudformation:DetectStackDrift 和 cloudformation:DetectStackResourceDr	2019 年 2 月 6 日

變更	描述	Date
	ift 動作進行堆疊偏離偵測的許可。	
AWSCloudFormationReadOnlyAccess – 新政策	CloudFormation 新增了新的 AWS 受管政策，提供 CloudFormation 動作和資源的唯讀存取權。	2015 年 2 月 6 日
CloudFormation 開始追蹤變更	CloudFormation 開始追蹤其 AWS 受管政策的變更。	2015 年 2 月 6 日

CloudFormation 服務角色

服務角色是 AWS Identity and Access Management (IAM) 角色，可讓 CloudFormation 代表您呼叫堆疊中的資源。您可以指定 IAM 角色，讓 CloudFormation 建立、更新或刪除您的堆疊資源。根據預設，CloudFormation 會使用從您的堆疊操作使用者登入資料所產生的暫時工作階段。如果您指定服務角色，CloudFormation 就會使用角色的登入資料。

使用服務角色明確指定 CloudFormation 可以執行的動作，但不一定是您或其他使用者可以執行的相同動作。例如，您可能有管理權限，但您可以限制 CloudFormation 只能存取 Amazon EC2 動作。

您使用 IAM 服務建立服務角色及其許可政策。如需建立服務角色的詳細資訊，請參閱《IAM 使用者指南》中的[建立角色以委派許可給 AWS 服務](#)。指定 CloudFormation (cloudformation.amazonaws.com) 做為可擔任該角色的服務。

若要建立服務角色與堆疊的關聯，請在您建立堆疊時指定角色。如需詳細資訊，請參閱[設定堆疊選項](#)。您也可以更新主控台堆疊時變更服務角色，或透過 API [DeleteStack](#) 堆疊。在您指定服務角色之前，請確保您有傳遞許可 (iam:PassRole)。iam:PassRole 許可會指定您可使用哪些角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[授予使用者將角色傳遞至 AWS 服務的許可](#)。

Important

當您指定某個服務角色時，CloudFormation 一律使用該角色處理在該堆疊上執行的所有操作。建立堆疊後，就無法移除連接至堆疊的服務角色。其他具有在此堆疊上執行這些操作之許可的使用者，都能夠使用此角色，不論這些使用者是否有 iam:PassRole 許可。如果該角色包

含使用者不該有的許可，您可能在無意中提升使用者的許可。確定該角色授予最低權限。如需詳細資訊，請參閱《IAM 使用者指南》中的[套用最低權限許可](#)。

預防跨服務混淆代理人

混淆代理人問題屬於安全性議題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在 AWS 中，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了防止這種情況，AWS 提供工具，協助您保護所有服務的資料，讓服務主體能夠存取您帳戶中的資源。

我們建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，以限制將另一個服務 CloudFormation 提供給特定資源的許可，例如 CloudFormation 延伸。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

確定的值 `aws:SourceArn` 是 CloudFormation 存放之資源的 ARN。

防範混淆代理人問題最有效的方法，是使用 `aws:SourceArn` 全域條件內容金鑰，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用 `aws:SourceArn` 全域條件內容金鑰，同時使用萬用字元 (*) 表示 ARN 的未知部分。例如 `arn:aws:cloudformation:*:123456789012:*`。

如果 `aws:SourceArn` 值不包含帳戶 ID，則必須使用這兩個全域條件內容索引鍵來限制許可。

下列範例示範如何在 AWS 中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵 CloudFormation，以防止混淆代理人問題。

使用 `aws:SourceArn` 和 `aws:SourceAccount` 條件索引鍵的信任政策範例

對於登錄服務，CloudFormation 會呼叫 AWS Security Token Service (AWS STS) 以在您的帳戶中擔任服務角色。在 [RegisterType](#) 操作中為 `ExecutionRoleArn` 設定此角色，在 [LoggingConfig](#) 操作中為 `LogRoleArn` 設定。如需詳細資訊，請參閱[為公有擴充功能存取設定具有 IAM 許可和信任政策的執行角色](#)。

此範例角色信任政策使用條件陳述式，將服務角色上的 `AssumeRole` 功能限制為僅對指定帳戶中指定的 CloudFormation 延伸執行動作。獨立評估 `aws:SourceArn` 和 `aws:SourceAccount` 條件。使用服務角色的任何請求都必須滿足這兩個條件。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "resources.cloudformation.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/Organization-Service-Resource"
        }
      }
    }
  ]
}
```

其他資訊

如需使用 StackSets 所用服務角色的 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵的政策範例，請參閱 [設定全域索引鍵以減輕混淆代理人問題](#)。

如需詳細資訊，請參閱《IAM 使用者指南》中的 [更新角色信任政策](#)。

轉送存取工作階段 (FAS) 請求和許可評估

建立、更新和刪除 CloudFormation 堆疊時，使用者可以選擇性地指定 IAM 角色 ARN。如果未提供任何角色，CloudFormation 會使用其預設服務機制與其他 AWS 服務互動。在此案例中，呼叫者必須擁有所管理資源的必要許可。或者，當使用者提供自己的 IAM 角色時，CloudFormation 將擔任該角色以代表他們執行服務互動。

無論使用者是否提供 IAM 角色，CloudFormation 都會為每個資源操作產生新的範圍縮小的 FAS 權杖。因此，在這兩種情況下都會填入 [FAS 相關條件金鑰](#) `aws:ViaAWSService`，包括。

使用 FAS 會影響 CloudFormation 操作期間的 IAM 政策評估方式。使用範本 (該範本包含受 FAS 相關條件金鑰影響的資源) 建立堆疊時，可能會發生許可拒絕。

IAM 政策範例

請考慮下列 IAM 政策。Statement2 會持續阻止在 CloudFormation 中建立 `AWS::KMS::Key` 資源。無論堆疊操作期間是否提供 IAM 角色，都會一致地強制執行該限制。這是因為由於使用 FAS，`aws:ViaAWSService` 條件金鑰始終設為 `true`。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Statement2",
      "Effect": "Deny",
      "Action": [
        "kms:CreateKey"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Bool": {
          "aws:ViaAWSService": "true"
        }
      }
    }
  ]
}
```

```
]
}
```

堆疊範本範例

例如，當使用者使用下列範例範本建立堆疊時，`aws:ViaAWSService` 會設定為 `true`，而 FAS 政策會覆寫角色許可。堆疊建立會受到拒絕 `CreateKey` 動作的 IAM 政策的 `Statement2` 的影響。這會導致許可遭拒錯誤。

```
Resources:
  myPrimaryKey:
    Type: AWS::KMS::Key
    Properties:
      Description: An example multi-Region primary key
      KeyPolicy:
        Version: '2012-10-17'
        Id: key-default-1
        Statement:
          - Sid: Enable IAM User Permissions
            Effect: Allow
            Principal:
              AWS: !Join
                - ''
                - - 'arn:aws:iam::'
                  - !Ref AWS::AccountId
                  - ':root'
            Action: kms:*
            Resource: '*'
```

如需 FAS 的詳細資訊，請參閱《IAM 使用者指南》中的[轉送存取工作階段](#)。

Note

大多數資源都遵循此行為。不過，如果在建立、更新或刪除資源時遇到意外的成功或失敗，且 IAM 政策包含 FAS 相關條件金鑰，則相關資源可能屬於不遵循此標準模式的小部分資源。

使用 記錄 CloudFormation API 呼叫 AWS CloudTrail

CloudFormation 已與 服務整合 AWS CloudTrail，此服務提供由使用者、角色或 CloudFormation 中的 AWS 服務所採取之動作的記錄。CloudTrail 將 CloudFormation 的所有 API 呼叫擷取為事件，包括來自 CloudFormation 主控台的呼叫，以及來自對 CloudFormation API 發出的程式碼呼叫。如果您建立追蹤記錄，就可以持續傳送 CloudTrail 事件至 Amazon S3 儲存貯體，包括 CloudFormation 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台的事件歷史記錄檢視最新事件。您可以使用由 CloudTrail 收集的資訊來判斷對 CloudFormation 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [「AWS CloudTrail 使用者指南」](#)。

主題

- [CloudTrail 中的 CloudFormation 資訊](#)
- [了解 CloudFormation 日誌檔案項目](#)

CloudTrail 中的 CloudFormation 資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 CloudFormation 中發生時，該活動會與事件歷史記錄中的其他服務 AWS 事件一起記錄在 CloudTrail 事件中。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](#)。

若要持續記錄您 AWS 帳戶中的事件，包括 CloudFormation 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立線索時，線索會套用到所有 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon 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：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案](#)

所有 CloudFormation 動作皆由 CloudTrail 記錄，並記錄在 [AWS CloudFormation API 參考](#) 中。例如，對 CreateStack、DeleteStack 及 ListStacks 區段的呼叫，都會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案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 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。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

了解 CloudFormation 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件代表任何來源提出的單一請求，並包含所請求之操作的相關資訊、操作的日期和時間、請求參數等等。CloudTrail 日誌檔案並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範 CreateStack 操作：該操作由名為 Alice 的 IAM 使用者發出。

Note

僅記錄輸入參數索引鍵名稱。不記錄參數值。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:02:43Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "CreateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
```

```
"templateURL": "templateURL",
"tags": [
  {
    "key": "test",
    "value": "tag"
  }
],
"stackName": "my-test-stack",
"disableRollback": true,
"parameters": [
  {
    "parameterKey": "password"
  },
  {
    "parameterKey": "securitygroup"
  }
],
},
"responseElements": {
  "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/
a38e6a60-b397-11e3-b0fc-08002755629e"
},
"requestID": "9f960720-b397-11e3-bb75-a5b75389b02d",
"eventID": "9bf6cfb8-83e1-4589-9a70-b971e727099b"
}
```

以下範例顯示 Alice 在 my-test-stack 堆疊上呼叫了 UpdateStack 操作：

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:04:29Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "UpdateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
```

```

"userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
"requestParameters": {
  "templateURL": "templateURL",
  "parameters": [
    {
      "parameterKey": "password"
    },
    {
      "parameterKey": "securitygroup"
    }
  ],
  "stackName": "my-test-stack"
},
"responseElements": {
  "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-stack/
a38e6a60-b397-11e3-b0fc-08002755629e"
},
"requestID": "def0bf5a-b397-11e3-bb75-a5b75389b02d",
"eventID": "637707ce-e4a3-4af1-8edc-16e37e851b17"
}

```

以下範例顯示 Alice 呼叫了 ListStacks 操作。

```

{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHJKLMNOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "ListStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
  "eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}

```

```
}
```

以下範例顯示 Alice 在 my-test-stack 堆疊上呼叫了 DescribeStacks 操作。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmnopq",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
  "eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

以下範例顯示 Alice 在 my-test-stack 堆疊上呼叫了 DeleteStack 操作。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHijklmnopq",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DeleteStack",

```

```
"awsRegion": "us-east-1",
"sourceIPAddress": "127.0.0.1",
"userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
"requestParameters": {
  "stackName": "my-test-stack"
},
"responseElements": null,
"requestID": "42dae739-b398-11e3-bb75-a5b75389b02d",
"eventID": "4965eb38-5705-4942-bb7f-20ebe79aa9aa"
}
```

中的基礎設施安全 CloudFormation

作為受管服務，AWS CloudFormation 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱[AWS 雲端安全](#)。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的[基礎設施保護](#)。

您可以使用 AWS 發佈的 API 呼叫，CloudFormation 透過網路存取。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。

您可以從任何網路位置呼叫這些 API 操作，但 CloudFormation 支援以資源為基礎的存取政策，其中可能包括根據來源 IP 地址的限制。您也可以使用 CloudFormation 政策來控制來自特定 Amazon Virtual Private Cloud (Amazon VPC) 端點或特定 VPCs存取。實際上，這只會隔離網路內特定 VPC 對指定 CloudFormation 資源 AWS 的網路存取。

中的彈性 CloudFormation

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置的。AWS 區域提供多個實體隔離和隔離的可用區域，這些區域以低延遲、高輸送量和高度備援的網路連接。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 AWS 區域和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

的合規驗證 CloudFormation

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱[AWS 服務 合規計劃範圍內](#)然後選擇您感興趣的合規計劃。如需一般資訊，請參閱[AWS 合規計劃](#)。

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[下載報告 in AWS Artifact](#)

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 [AWS 安全文件](#)。

中的組態和漏洞分析 CloudFormation

組態和 IT 控制是 AWS 與身為我們客戶的您共同的責任。如需詳細資訊，請參閱 AWS [共同的責任模型](#)。

CloudFormation 的安全最佳實務

CloudFormation 提供許多安全功能，供您在開發和實作自己的安全政策時考慮。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

主題

- [使用 IAM 控制存取](#)
- [請勿在您的範本中內嵌憑證](#)
- [使用 AWS CloudTrail 記錄 CloudFormation 呼叫](#)

使用 IAM 控制存取

IAM 是一項 AWS 服務，可用來在 中管理使用者及其許可 AWS。您可以使用 IAM 搭配 CloudFormation 來指定使用者可執行的 CloudFormation 動作，例如檢視堆疊範本、建立堆疊或刪除堆疊。此外，管理 CloudFormation 堆疊的任何人都需要這些堆疊中的資源許可。例如，如果使用者想要使用 CloudFormation 來啟動、更新或終止 Amazon EC2 執行個體，則必須具備呼叫相關 Amazon EC2 動作的許可。

在大多數情況下，使用者需要完整存取權，才能管理範本中的所有資源。CloudFormation 會呼叫來代替使用者建立、修改及刪除這些資源。若要區隔使用者與 CloudFormation 服務的許可，請使用服務角色。CloudFormation 使用服務角色的政策來呼叫，而不是使用者的政策。如需詳細資訊，請參閱[CloudFormation 服務角色](#)。

請勿在您的範本中內嵌憑證

建議您在堆疊範本中使用動態參考，而不要在 CloudFormation 範本中嵌入敏感資訊。

動態參考提供精簡且強大的方式，讓您參考儲存在其他服務中的外部值，例如 AWS Systems Manager 參數存放區或 AWS Secrets Manager。當您使用動態參考時，在堆疊和變更集操作期間，CloudFormation 會在必要時擷取指定參考的值，並將值傳遞至適當的資源。不過，CloudFormation 絕不會存放實際參考值。如需詳細資訊，請參閱[使用動態參考取得存放在其他服務中的值](#)。

[AWS Secrets Manager](#) 協助您安全地加密、儲存和擷取資料庫和其他服務的憑證。[AWS Systems Manager 參數存放區](#) 提供安全的階層式儲存，以管理組態資料。

如需定義範本參數的詳細資訊，請參閱[CloudFormation 範本 Parameters 語法](#)。

使用 AWS CloudTrail 記錄 CloudFormation 呼叫

AWS CloudTrail 會追蹤在您的 中進行 CloudFormation API 呼叫的任何人 AWS 帳戶。只要有人使用 CloudFormation API、CloudFormation 主控台、後端主控台或 CloudFormation AWS CLI 命令，就會記錄 API 呼叫。啟用記錄，然後指定 Amazon S3 儲存貯體來存放日誌。如此一來，如果您需要，即可稽核誰在您的帳戶中發出哪些 CloudFormation 呼叫。如需詳細資訊，請參閱[使用 記錄 CloudFormation API 呼叫 AWS CloudTrail](#)。

CloudFormation 使用界面端點存取 (AWS PrivateLink)

您可以使用在 VPC 與 之間 AWS PrivateLink 建立私有連線 CloudFormation。您可以 CloudFormation 像在 VPC 中一樣存取，無需使用網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址即可存取 CloudFormation。

您可以建立由 AWS PrivateLink 提供支援的介面端點來建立此私有連線。我們會在您為介面端點啟用的每個子網中建立端點網路介面。這些是請求者管理的網路介面，可作為目的地為 CloudFormation 之流量的進入點。

CloudFormation 支援透過介面端點呼叫其所有 API 動作。

CloudFormation VPC 端點的考量事項

在您設定介面端點之前，請先確定您已符合《AWS PrivateLink 指南》中的[使用介面 VPC 端點主題存取 AWS 服務](#)中的先決條件。

設定介面端點時，適用下列額外的先決條件和考量事項 CloudFormation：

- 若您的 VPC 中有資源必須回應自訂資源請求或等待條件，請確保這些資源可存取所需的 CloudFormation 專屬 Amazon S3 儲存貯體。CloudFormation 在每個區域中擁有 S3 儲存貯體，以監控對於[自訂資源](#)請求或[等待條件](#)的回應。如果範本包含 VPC 中的自訂資源或等待條件，VPC 端點政策必須允許使用者將回應傳送至以下儲存貯體：
 - 對於自訂資源，允許流至 `cloudformation-custom-resource-response-region` 儲存貯體的流量。使用自訂資源時，AWS 區域 名稱不包含破折號。例如 `uswest2`。
 - 對於等待條件，允許流至 `cloudformation-waitcondition-region` 儲存貯體的流量。使用等待條件時，AWS 區域 名稱會包含破折號。例如 `us-west-2`。

如果端點政策封鎖流至這些儲存貯體的流量，CloudFormation 將不會收到回應，堆疊操作將會失敗。例如，如果您在必須回應等待條件之 `us-west-2` 區域中的 VPC 有一個資源，此資源必須能夠傳送回應至 `cloudformation-waitcondition-us-west-2` 儲存貯體。

如需 CloudFormation 目前可用 AWS 區域 位置的清單，請參閱 中的[CloudFormation 端點和配額](#)頁面 Amazon Web Services 一般參考。

- VPC 端點目前不支援跨區域請求 – 請確定在您打算對 CloudFormation 發出 API 呼叫的同一區域中建立端點。
- 透過 Route 53，VPC 端點僅支援 Amazon 提供的 DNS。如果您想要使用自己的 DNS，您可以使用條件式 DNS 轉送。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[Amazon VPC 中的 DHCP 選項集](#)。
- 連接到 VPC 端點的安全群組，必須允許從 VPC 的私有子網路，透過 443 埠傳入的連線。

建立的介面 VPC 端點 CloudFormation

您可以使用 Amazon VPC CloudFormation 主控台或 AWS Command Line Interface (AWS CLI) 來建立 VPC 端點。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的[建立 VPC 端點](#)。

CloudFormation 使用下列服務名稱建立 的介面端點：

- `com.amazonaws.region.cloudformation` – 建立用於 CloudFormation API 操作的端點。

如果您為介面端點啟用私有 DNS，您可以使用 CloudFormation 其預設的區域 DNS 名稱向 提出 API 請求。例如 `cloudformation.us-east-1.amazonaws.com`。

在支援 FIPS 特定端點 AWS 區域 的情況下，您也可以 CloudFormation 使用下列服務名稱建立 的介面端點：

- `com.amazonaws.region.cloudformation-fips` – 建立符合[聯邦資訊處理標準 \(FIPS\) 140-2](#) 的 CloudFormation API 端點。

如需 CloudFormation 端點的完整清單，請參閱 Amazon Web Services 一般參考 中的 [CloudFormation 端點與配額](#)。

為 建立 VPC 端點政策 CloudFormation

端點政策為 IAM 資源，您可將其連接至介面端點。預設端點政策允許 CloudFormation 透過介面端點完整存取。若要控制允許 CloudFormation 從您的 VPC 存取，請將自訂端點政策連接至介面端點。

端點政策會指定以下資訊：

- 可執行動作 (AWS 帳戶、IAM 使用者和 IAM 角色) 的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的[使用端點政策控制 VPC 端點的存取](#)。

範例：CloudFormation 動作的 VPC 端點政策

以下是 端點政策的範例 CloudFormation。連接至 端點時，此政策會授予所有資源上所有委託人所列出 CloudFormation 動作的存取權。下列範例拒絕所有使用者透過 VPC 端點建立堆疊的許可，並允許完整存取 CloudFormation 服務上的所有其他動作。

```
{
  "Statement": [
    {
      "Action": "cloudformation:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "cloudformation:CreateStack",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

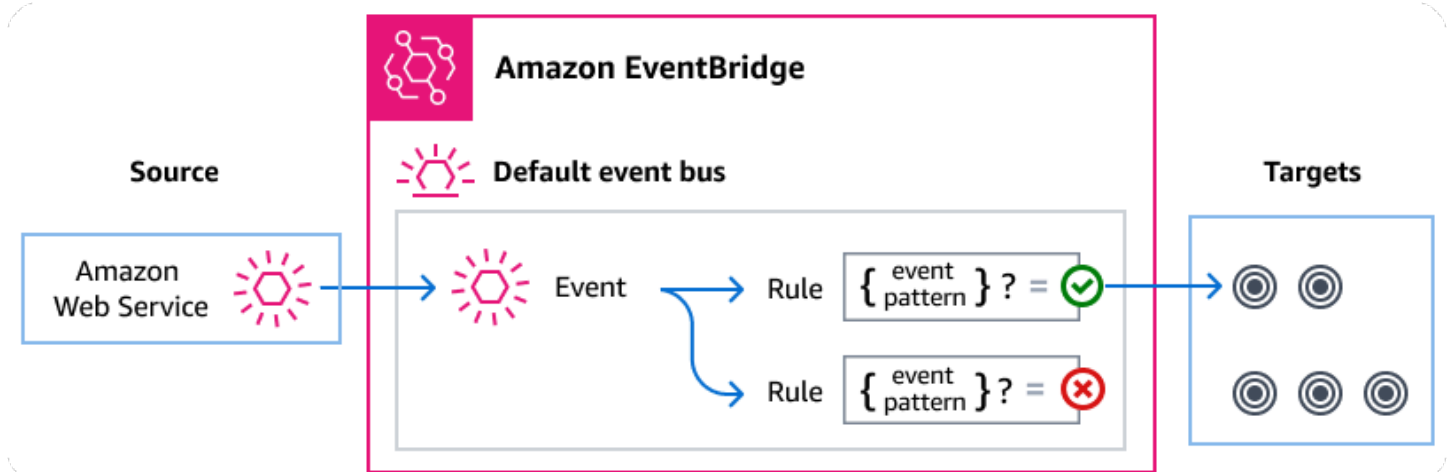
```
}
```

使用 EventBridge 來監控 CloudFormation 和 Git 同步事件

Amazon EventBridge 是一種無伺服器服務，該服務使用事件將應用程式元件連接在一起，讓您更輕鬆地建置可擴展的事件驅動型應用程式。事件驅動型架構是一種建置鬆耦合軟體系統的方式，透過發出和回應事件來協作。事件代表資源或環境中的變更。

如同許多 AWS 服務，CloudFormation 會產生事件並將其傳送至 EventBridge 預設事件匯流排，而 EventBridge 預設事件匯流排會在每個匯流排中自動佈建 AWS 帳戶。事件匯流排是接收事件，並將事件傳遞至零個或多個目的地或目標的路由器。為事件匯流排指定的規則會在事件到達時評估事件。每項規則都會檢查事件是否與規則的事件模式相符。如果事件不相符，事件匯流排會將事件傳送至指定的目標。

如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 [Amazon EventBridge 入門](#)。



主題

- [CloudFormation 和 Git 同步事件概觀](#)
- [Amazon EventBridge 許可](#)
- [針對 EventBridge 規則建立自訂事件模式](#)
- [CloudFormation 活動詳細資訊參考](#)

CloudFormation 和 Git 同步事件概觀

每當對堆疊執行建立、更新、刪除或漂移偵測作業時，CloudFormation 都會將事件傳送到 EventBridge。CloudFormation 也會針對堆疊集和堆疊集執行個體的狀態變更，將事件傳送至

EventBridge。您可以使用 EventBridge 規則將事件路由至您定義的目標。保證會交付這些事件，但是可能無法按順序交付。

由於 CloudFormation 事件代表堆疊或堆疊集及其資源的變更，因此，您可以使用它們來啟動與個別事件關聯的工作流程。例如：

- 在透過 CloudFormation 佈建的所有資源上建立堆疊或堆疊集特定標籤。
- 在 CloudFormation 堆疊或堆疊集和 Amazon WorkSpaces Application Manager (Amazon WAM) 之間建立關聯。
- 為建立的堆疊或堆疊集指定與 AppRegistry 的關聯。

下列事件由 CloudFormation 產生，並會將事件傳送至 EventBridge 中的預設事件匯流排。如需詳細資訊，請參閱[???](#)。

事件類型	Description	事件來源
資源狀態變更	在會變更基礎資源屬性之堆疊上執行的任何更新。 如需支援 AWS 的資源類型完整清單，請參閱AWS 資源和屬性類型參考。	AWS CloudFormation
堆疊狀態變更	代表指定堆疊的狀態變更。 如需程式碼詳細資訊，請參閱 堆疊狀態碼。	AWS CloudFormation
偏離偵測狀態變更	代表使用者在指定堆疊上啟動的漂移偵測更新。 如需支援偏離偵測的完全可變和不可變類型的完整清單，請參閱資源類型支援	AWS CloudFormation
StackSet 狀態變更	代表指定堆疊集的狀態變更。	AWS CloudFormation
StackSet 堆疊執行個體狀態變更	代表特定 StackSet 堆疊執行個體的狀態變更。 如需程式碼詳細資訊，請參閱 堆疊執行個體狀態碼。	AWS CloudFormation
StackSet 操作狀態	代表指定 StackSet 操作的狀態變更。	AWS CloudFormation

事件類型	Description	事件來源
	如需程式碼詳細資訊，請參閱 StackSets 狀態碼 。	

此外，AWS CloudFormation Git 同步會將儲存庫同步和資源同步的狀態變更事件傳送至 EventBridge。

下列 Git 同步事件由 CodeConnections 產生，並會將事件傳送至 EventBridge 中的預設事件匯流排。如需詳細資訊，請參閱[???](#)。

事件類型	Description	事件來源
儲存庫同步狀態變更	表示 Git 儲存庫同步的狀態變更。	AWS CodeConnections
資源同步狀態變更	表示 Git 資源同步的狀態變更。	AWS CodeConnections

Amazon EventBridge 許可

CloudFormation 無需任何額外許可，即可將事件傳遞給 EventBridge。這些事件包含已透過 CloudFormation 的 API 操作提供的資訊。

您指定的目標可能需要特定的許可或組態。如需針對目標使用特定服務的更多詳細資訊，請參閱《Amazon EventBridge 使用者指南中》的 [Amazon EventBridge 目標](#)。

針對 EventBridge 規則建立自訂事件模式

可以在 EventBridge for CloudFormation 和 Git 同步事件中找到數個預先定義的模式。這簡化了事件模式的建立方式。可以選取表單上的欄位值，而無需撰寫自己的事件模式，EventBridge 便會為您產生模式。可以使用其中一個預先定義的事件模式來建立新規則，或建立自己的自訂事件模式。

當 CloudFormation 等服務將事件交付至預設事件匯流排時，EventBridge 會使用規則中定義的事件模式來判斷事件是否應交付至規則的目標。事件模式符合所需 CloudFormation 事件中的資料。

每個事件模式都是 JSON 物件，它包含：

- 識別事件來源之服務的 `source` 屬性。例如 `aws.cloudformation` 或 `aws.codeconnections`。
- (選擇性)：包含要比對之事件類型陣列的 `detail-type` 屬性。
- (選擇性)：包含要比對的任何其他事件資料的 `detail` 屬性。

例如，堆疊 ID、涉及的資源、各種資源的狀態，以及與特定類型的事件相關的其他資料。

例如，下列事件模式會與所有資源狀態變更事件進行比對：

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"]
}
```

下列事件模式會使用事件詳細資料，僅比對 CloudFormation 建立新 `AWS::S3::Bucket` 或 `AWS::SNS::Topic` 資源的資源狀態變更事件：

```
{
  "source": ["aws.cloudformation"],
  "detail-type": ["CloudFormation Resource Status Change"],
  "detail": {
    "status-details": {
      "status": ["CREATE_COMPLETE"]
    },
    "resource-type": ["AWS::S3::Bucket", "AWS::SNS::Topic"]
  }
}
```

如需有關寫入事件模式的詳細資訊，請參閱 Amazon EventBridge 使用者指南中的[事件模式](#)。

CloudFormation 活動詳細資訊參考

所有來自 AWS 服務的事件都有一組通用的欄位，其中包含事件的中繼資料，例如事件來源 AWS 的服務、事件產生的時間、事件發生的帳戶和區域，以及其他。如需這些一般欄位的定義，請參閱 AWS 事件參考中的[AWS 服務事件中繼資料](#)。

此外，每個事件都有一個 `detail` 欄位，其中包含該特定事件的特定資料。以下參考定義了各種 CloudFormation 事件的詳細資訊欄位。

使用 EventBridge 選取並管理 CloudFormation 事件時，請務必記住下列事項：

- `source` 欄位可指定事件來源。

例如 `aws.cloudformation` 或 `aws.codeconnections`。

- `detail-type` 欄位指定事件類型。

例如 `CloudFormation Resource Status Change` 或 `CloudFormation Drift Detection Status Change`。

- `detail` 欄位包含該特定事件的特定資料。

例如，堆疊 ID、涉及的資源、各種資源的狀態，以及與特定類型的事件相關的其他資料。

如需建構可讓規則與 CloudFormation 事件相符的事件模式相關資訊，請參閱《Amazon EventBridge 使用者指南》中的[事件模式](#)。

如需有關事件以及 EventBridge 如何處理事件的詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的[EventBridge 事件](#)。

主題

- [「資源狀態變更」事件詳細資訊](#)
- [「堆疊狀態變更」事件詳細資訊](#)
- [「偏離偵測狀態變更」事件詳細資訊](#)
- [「StackSet 狀態變更」事件詳細資訊](#)
- [「StackSet 堆疊執行個體狀態變更」事件詳細資訊](#)
- [「StackSet 操作狀態變更」事件詳細資訊](#)
- [「儲存庫同步狀態變更」事件詳細資訊](#)
- [「資源同步狀態變更」事件詳細資訊](#)

「資源狀態變更」事件詳細資訊

以下是「資源狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "CloudFormation Resource Status Change",
  "source": "aws.cloudformation",
```



```
. . . ,
"detail": {
  "stack-id" : "string",
  "logical-resource-id" : "string",
  "physical-resource-id": "string",
  "status-details": {
    "status": "string",
    "status-reason": "string"
  },
  "resource-type": "string",
  "client-request-token": "string"
}
```

detail-type

識別事件的類型。

若是資源狀態事件，此值為 CloudFormation Resource Status Change。

source

識別產生事件的服務。若是 CloudFormation 事件，此值為 `aws.cloudformation`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是資源狀態事件，此資料包括：

stack-id

與堆疊關聯的唯一的堆疊 ID。

logical-resource-id

範本中指定的資源邏輯名稱。

physical-resource-id

與 CloudFormation 支援之資源的實體執行個體 ID 對應的名稱或唯一識別碼。

status-details

status

資源狀態。

status-reason

資源的狀態原因。

resource-type

資源類型。例如 AWS::S3::Bucket。

client-request-token

用於呼叫 API 的存取權杖。針對特定堆疊操作啟動的所有事件，系統會指派相同的用戶端請求權杖，您即可透過該權杖追蹤操作。為了讓您能輕鬆地辨識堆疊操作，從主控台啟動的堆疊操作皆會採用 Console-StackOperation-ID 字符格式。舉例來說，若您使用主控台建立堆疊，則系統將指派相同字符給每個產生的堆疊事件，該字符格式如下所示：Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002。

Example範例：「資源狀態變更」事件

下列為範例資源狀態事件。此事件詳細說明了 CloudFormation 已在指定堆疊中成功建立請求的資源 (Amazon S3 儲存貯體)。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Resource Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
  ],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "logical-resource-id": "my-s3-bucket",
    "physical-resource-id": "arn:aws:s3::my-s3-bucket-us-east-1",
    "status-details": {
      "status": "CREATE_COMPLETE",
      "status-reason": ""
    },
    "resource-type": "AWS::S3::Bucket",
    "client-request-token": ""
  }
}
```

```
}
```

「堆疊狀態變更」事件詳細資訊

以下為「堆疊狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "CloudFormation Stack Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-id": "string",
    "status-details": {
      "status": "string",
      "status-reason": "string"
    },
    "client-request-token": "string"
  }
}
```

detail-type

識別事件的類型。

若是堆疊狀態事件，此值為 `CloudFormation Stack Status Change`。

source

識別產生事件的服務。若是 CloudFormation 事件，此值為 `aws.cloudformation`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是堆疊狀態事件，此資料包括：

stack-id

與堆疊關聯的不重複堆疊 ID。

status-details

status

堆疊的狀態。

如需堆疊狀態碼的完整清單，請參閱 [堆疊狀態碼](#)。

status-reason

資源的狀態原因。

client-request-token

用於呼叫 API 的存取權杖。針對特定堆疊操作啟動的所有事件，系統會指派相同的用戶端請求權杖，您即可透過該權杖追蹤操作。為了讓您能輕鬆地辨識堆疊操作，從主控台啟動的堆疊操作皆會採用 Console-StackOperation-ID 字符格式。舉例來說，若您使用主控台建立堆疊，則系統將指派相同字符給每個產生的堆疊事件，該字符格式如下所示：Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002。

Example範例：堆疊狀態事件

以下是範例堆疊狀態事件，其中 CloudFormation 已成功建立請求的堆疊 teststack。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Stack Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack"
  ],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "status-details": {
      "status": "CREATE_COMPLETE",
      "status-reason": ""
    },
    "client-request-token": ""
  }
}
```

「偏離偵測狀態變更」事件詳細資訊

以下是堆疊漂移偵測事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "CloudFormation Drift Detection Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-id": "string",
    "stack-drift-detection-id": "string",
    "status-details": {
      "stack-drift-status": "string",
      "detection-status": "string"
    },
    "drift-detection-details": {
      "drifted-stack-resource-count": integer
    },
    "client-request-token": "string"
  }
}
```

detail-type

識別事件的類型。

若是堆疊漂移偵測事件，此值為 `CloudFormation Drift Detection Status Change`。

source

識別產生事件的服務。若是 `CloudFormation` 事件，此值為 `aws.cloudformation`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是堆疊漂移偵測事件，此資料包括：

stack-id

與堆疊關聯的唯一的堆疊 ID。

`stack-drift-detection-id`

堆疊漂移偵測 ID。

`status-details`

`stack-drift-status`

堆疊的漂移狀態。

`detection-status`

漂移偵測操作狀態。

`drift-detection-details`

`drifted-stack-resource-count`

漂移資源的數量。當值為 -1 時，代表漂移偵測正在進行。所有其他非負整數代表漂移資源的實際數量。

`client-request-token`

用於呼叫 API 的存取權杖。針對特定堆疊操作啟動的所有事件，系統會指派相同的用戶端請求權杖，您即可透過該權杖追蹤操作。為了讓您能輕鬆地辨識堆疊操作，從主控台啟動的堆疊操作皆會採用 Console-StackOperation-ID 字符格式。舉例來說，若您使用主控台建立堆疊，則系統將指派相同字符給每個產生的堆疊事件，該字符格式如下所示：Console-CreateStack-7f59c3cf-00d2-40c7-b2ff-e75db0987002。

Example 範例：「堆疊偏離偵測」事件

以下為「堆疊偏離偵測」事件的範例。此事件詳細說明了 CloudFormation 已在指定堆疊完成漂移偵測，以及堆疊目前由於一個資源漂移而具有 DRIFTED 漂移狀態。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "CloudFormation Drift Detection Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": ["string"],
  "detail": {
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
```

```

    "stack-drift-detection-id": "624af370-311a-11e8-b6b7-500cexample",
    "status-details": {
      "stack-drift-status": "DRIFTED",
      "detection-status": "DETECTION_COMPLETE"
    },
    "drift-detection-details": {
      "drifted-stack-resource-count": 1
    },
    "client-request-token": ""
  }
}

```

「StackSet 狀態變更」事件詳細資訊

以下為「StackSet 狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```

{
  . . . ,
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "status-details": {
      "status": "string"
    }
  }
}

```

detail-type

識別事件的類型。

若是 StackSet 狀態事件，此值為 `CloudFormation StackSet Status Change`。

source

識別產生事件的服務。若是 CloudFormation 事件，此值為 `aws.cloudformation`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是 StackSet 狀態事件，此資料包括：

stack-set-arn

與堆疊集關聯的 Amazon Resource Name (ARN)。

status-details

status

StackSet 狀態。

有效值：ACTIVE | DELETED

Example範例：「StackSet 狀態變更」事件

以下為「StackSet 狀態變更」事件範例。此事件詳細說明了 CloudFormation 已刪除指定的堆疊集。

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xhnh0934z09",
  "detail-type": "CloudFormation StackSet Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-23T17:06:18Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663"
  ],
  "detail": {
    "stack-set-arn" : "arn:aws:cloudformation:us-east-1:111122223333:stackset/test12345:3f3a3fbe-c937-4eb3-a87d-e36a0af3f663",
    "status-details": {
      "status":"DELETED"
    }
  }
}
```

「StackSet 堆疊執行個體狀態變更」事件詳細資訊

以下是 StackSet 堆疊執行個體狀態事件的詳細欄位。

包含 source 和 detail-type 欄位，因為其中包含事件的特定值。


```
{
  . . . ,
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-id" : "string",
    "action" : "string",
    "status-details": {
      "status": "string",
      "status-reason": "string",
      "detailed-status": "string"
    }
  }
}
```

detail-type

識別事件的類型。

若是 StackSet 堆疊執行個體狀態事件，此值為 CloudFormation StackSet StackInstance Status Change。

source

識別產生事件的服務。若是 CloudFormation 事件，此值為 aws.cloudformation。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是 StackSet 堆疊執行個體狀態事件，此資料包括：

stack-set-arn

與 StackSet 關聯的 Amazon Resource Name (ARN)。

stack-id

與堆疊執行個體關聯的唯一的堆疊 ID。

action

堆疊集操作的類型。

有效值：CREATE | UPDATE | DELETE | DETECT_DRIFT

status-details

status

StackSet 執行個體狀態。

如需詳細資訊，請參閱[堆疊執行個體狀態碼](#)。

有效值：CURRENT | OUTDATED | INOPERABLE

status-reason

StackSet 執行個體的狀態原因。

detailed-status

詳細的 StackSet 執行個體狀態。

有效值：CANCELLED | FAILED | FAILED_IMPORT | INOPERABLE | PENDING | RUNNING | SKIPPED_SUSPENDED_ACCOUNT | SUCCEEDED

Example範例：「StackSet 堆疊執行個體狀態變更」事件

以下為「StackSet 堆疊執行個體狀態變更」事件範例。

```
{
  "version": "0",
  "id": "42h6hb90-hg0w-11op-b01v-0xnhn0934z09",
  "detail-type": "CloudFormation StackSet StackInstance Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T19:19:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59"
  ],
  "detail": {
    "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
    "stack-id": "arn:aws:cloudformation:us-west-1:111122223333:stack/teststack",
    "status-details": {
      "status": "OUTDATED",
```

```
        "status-reason": "User Initiated",
        "detailed-status": "PENDING"
    }
}
```

「StackSet 操作狀態變更」事件詳細資訊

以下是「StackSet 操作狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  . . . ,
  "detail": {
    "stack-set-arn" : "string",
    "stack-set-operation-id" : "string",
    "status-details": {
      "status": "string"
    }
  }
}
```

detail-type

識別事件的類型。

若是 StackSet 操作狀態事件，此值為 CloudFormation StackSet Operation Status Change。

source

識別產生事件的服務。若是 CloudFormation 事件，此值為 `aws.cloudformation`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若是 StackSet 操作狀態事件，此資料包括：

`stack-set-arn`

與 StackSet 關聯的 Amazon Resource Name (ARN)。

`stack-set-operation-id`

與 StackSet 操作關聯的唯一 ID。

`status-details`

`status`

StackSet 操作狀態。

如需詳細資訊，請參閱[StackSets 狀態碼](#)。

有效值：RUNNING | SUCCEEDED | FAILED | STOPPING | STOPPED | QUEUED

Example範例：「StackSet 操作狀態變更」事件

以下為「StackSet 操作狀態變更」事件範例。CloudFormation 已在指定堆疊集上成功完成請求操作的事件詳細資訊。

```
{
  "version": "0",
  "id": "4de89905-fd92-6a6b-9509-23c04bcb6a21",
  "detail-type": "CloudFormation StackSet Operation Status Change",
  "source": "aws.cloudformation",
  "account": "111122223333",
  "time": "2021-09-22T05:46:24Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59"
  ],
  "detail": {
    "stack-set-arn": "arn:aws:cloudformation:us-east-1:111122223333:stackset/test1234:e5f54eea-d041-44ad-94f8-b8268aca1e59",
    "stack-set-operation-id": "ce69adce-2221-4483-8c4b-c51f284f25e8",
    "status-details": {
      "status": "SUCCEEDED"
    }
  }
}
```

「儲存庫同步狀態變更」事件詳細資訊

以下是「儲存庫同步狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "Git Sync Repository Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
    "connectionArn" : "string",
    "providerType" : "string",
    "repositoryName": "string",
    "providerType": "string",
    "repositoryName": "string",
    "repositoryArn": "string",
    "repositoryLinkId": "string",
    "ownerId": "string",
    "commit": "string",
    "branch": "string",
    "syncType": "string",
    "status": "string",
    "previousSync": "string"
  }
}
```

detail-type

識別事件的類型。

對於儲存庫同步狀態事件，此值為 `Git Sync Repository Sync Status Change`。

source

識別產生事件的服務。對於 Git 同步事件，此值為 `aws.codeconnections`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

對於儲存庫同步狀態事件，此資料包括：

`connectionArn`

與 CodeConnections 相關聯的 Amazon Resource Name (ARN)。

`providerType`

連線至 CloudFormation 的 Git 提供者。

有效值：GitHub | GitHub Enterprise | GitLab | BitBucket

`repositoryName`

Git 儲存庫名稱。

`repositoryArn`

與 Git 儲存庫相關聯的 ARN。

`repositoryLinkId`

與儲存庫連結相關聯的唯一 ID。

`ownerId`

與儲存庫擁有者相關聯的唯一 ID。

`commit`

與儲存庫遞交相關聯的唯一 ID。

`branch`

與儲存庫分支相關聯的唯一 ID。

`syncType`

正在執行的同步類型。

`status`

目前的儲存庫同步狀態。

有效值：FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

`previousSync`

目前狀態之前的同步狀態。

有效值：FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example範例：「儲存庫同步狀態變更」事件

以下是「儲存庫同步狀態變更」事件的範例。CodeConnections 已成功同步儲存庫的事件詳細資訊。

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "GitSync Repository Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000"],
  "detail": {
    "connectionArn": "arn:aws:codestar-connections:us-east-1:111122223333:connection/sample-connection-id",
    "providerType": "GitHub",
    "repositoryName": "sample-repository-name",
    "repositoryArn": "arn:aws:aws:codestar-connections:us-east-1:111122223333:repository-link/550e8400-e29b-41d4-a716-446655440000",
    "repositoryLinkId": "550e8400-e29b-41d4-a716-446655440000",
    "ownerId": "sample-owner-id",
    "commit": "sample-commit-id",
    "branch": "main",
    "syncType": "CFN_STACK_SYNC",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS",
  }
}
```

「資源同步狀態變更」事件詳細資訊

以下是「資源同步狀態變更」事件的詳細資訊欄位。

包含 `source` 和 `detail-type` 欄位，因為其中包含事件的特定值。

```
{
  . . . ,
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  . . . ,
  "detail": {
```

```
"providerType" : "string",
"commit" : "string",
"repositoryName": "string",
"branch": "string",
"syncType": "string",
"syncTarget": "string",
"status": "string",
"previousSync": "string"
}
}
```

detail-type

識別事件的類型。

對於儲存庫同步狀態事件，此值為 Git Sync Repository Sync Status Change。

source

識別產生事件的服務。對於 Git 同步事件，此值為 `aws.codeconnections`。

detail

包含事件相關資訊的 JSON 物件。產生事件的服務會決定此欄位的內容。

若為資源同步狀態事件，此資料包括：

providerType

連線至 CloudFormation 的 Git 提供者。

有效值：GitHub | GitHub Enterprise | GitLab | BitBucket

commit

與儲存庫遞交相關聯的唯一 ID。

repositoryName

Git 儲存庫名稱。

branch

與儲存庫分支相關聯的唯一 ID。

syncType

正在執行的同步類型。

syncTarget

資源同步的目標堆疊。

status

目前的儲存庫同步狀態。

有效值：FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

previousSync

目前狀態之前的同步狀態。

有效值：FAILED | INITIATED | IN_PROGRESS | SUCCEEDED

Example範例：「資源同步狀態變更」事件

下列為資源同步狀態變更事件範例。CodeConnections 已成功同步資源的事件詳細資訊。

```
{
  "version": "0",
  "id": "1b5d8feb-agbv-4cf7-a9f1-bf3703467718",
  "detail-type": "Git Sync Resource Sync Status Change",
  "source": "aws.codeconnections",
  "account": "111122223333",
  "time": "2023-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": ["arn:aws:aws:cloudformation:us-east-1:111122223333:stack/targetStack1"],
  "detail": {
    "providerType": "GitHub",
    "commit": "sample-commit-id",
    "repositoryName": "sample-repository-name",
    "branch": "main",
    "syncType": "CFN_STACK_SYNC",
    "syncTarget": "arn:aws:aws:cloudformation:us-east-1:111122223333:stack/targetStack1",
    "status": "SUCCEEDED",
    "previousStatus": "IN_PROGRESS"
  }
}
```

了解 CloudFormation 配額

您的 AWS 帳戶 具有 CloudFormation 配額，您在撰寫範本和建立堆疊時可能需要知道這些配額。透過了解這些配額，您可以避免要求您重新設計範本或堆疊的限制錯誤。

下表顯示 CloudFormation 配額。

配額	Description	Value	調校策略
cfn-signal 等待條件資料	cfn-signal 可以傳遞的最大資料量。	4,096 位元組	若要傳遞較大的數量，請將資料傳送到 Amazon S3 儲存貯體，然後使用 cfn-signal 將 Amazon S3 URL 傳遞到該儲存貯體。
自訂資源回應	自訂資源提供者可以傳遞的最大資料量。	4,096 位元組	
每個範本的動態參考	單一 CloudFormation 堆疊範本中允許的動態參考數目上限。	一個堆疊範本中包含 60 個動態參考	
每個帳戶的勾點	每個區域每個帳戶允許的勾點數量上限。	100 個勾點	
每個資源的勾點	每個資源可設定的勾點數量上限。	100 個勾點	
勾點組態大小	勾點可以存放的組態資料量上限。	204.8 KB	
映射項目	可以在 CloudFormation 範本中宣告的映射數目上限。	200 個映射	若要指定更多映射，請使用例如 巢狀堆疊 ，將您的範本分成多個範本。

配額	Description	Value	調校策略
映射屬性	可以在 CloudFormation 範本中宣告之每個映射的映射屬性數目上限。	200 個屬性	若要指定更多的映射屬性，請將屬性分成多個映射。
映射名稱和映射屬性名稱	每個映射名稱的大小上限。	255 個字元	
模組	每個帳戶和區域可以在 CloudFormation 登錄中註冊的模組數量上限。	100 個模組	
模組版本	您可以在指定模組的 CloudFormation 登錄中註冊的版本數上限。	100 個版本	若要註冊新版本，請先使用 DeregisterType ，取消註冊您不再使用的版本。
巢狀堆疊	巢狀堆疊每次操作可以建立、更新或刪除的 CloudFormation 資源的數量上限。 例如，您可以擁有總計超過 2500 個資源的巢狀堆疊階層，但無法在單一部署中建立、更新或刪除超過 2500 個資源。	2500 項資源	將堆疊階層分割為不同的堆疊。
輸出	您可以在 CloudFormation 範本中宣告的輸出數目上限。	200 個輸出	
輸出名稱	輸出名稱的大小上限。	255 個字元	

配額	Description	Value	調校策略
參數	您可以在 CloudFormation 範本中宣告的參數數目上限。	200 個參數	若要指定更多參數，您可以使用映射或清單，將多個值依序指派給單一參數。
參數名稱	參數名稱的大小上限。	255 個字元	
參數值	參數值的大小上限。	4,096 位元組	若要使用較大的參數化值，請建立多個參數，然後使用 Fn::Join 函數將多個值附加到單一值的後面。
私有資源	每個區域每個帳戶可以在 CloudFormation 登錄中註冊的私有資源數量上限。	50 個私有資源	
私有資源版本	您可以在 CloudFormation 登錄中為指定的私有資源註冊的版本數目上限。	50 個私有資源	若要註冊新版本，請先使用 DeregisterType ，取消註冊您不再使用的版本。
資源	您可以在 CloudFormation 範本中宣告的資源數目上限。	500 項資源	若要指定更多資源，請使用例如 巢狀堆疊 ，將您的範本分成多個範本。
並行堆疊操作中的資源限制	在指定時間內，您可以在您區域中參與堆疊操作 (建立、更新或刪除操作) 的最大資源數。	使用 DescribeAccountLimits API 來判斷特定區域中帳戶的目前限制。	

配額	Description	Value	調校策略
資源名稱	資源名稱的大小上限。	255 個字元	
堆疊	您可以建立的 CloudFormation 堆疊數目上限。	2000 個堆疊	若要建立更多堆疊，請刪除您不需要的堆疊，或請求提高 AWS 帳戶的堆疊數目上限。如需詳細資訊，請參閱《AWS》中的 AWS 一般參考服務配額 。
Stack name (堆疊名稱)	堆疊名稱的大小上限。	128 個字元	
StackSets	您在管理員帳戶中可以建立的 CloudFormation 堆疊集數目上限。	1000 個堆疊集	若要建立更多堆疊集，請刪除您不需要的堆疊集，或請求提高 AWS 帳戶的堆疊集數目上限。如需詳細資訊，請參閱《AWS》中的 AWS 一般參考服務配額 。
堆疊執行個體	每個堆疊集可以建立的堆疊執行個體數目上限。	每個堆疊集 100,000 個堆疊執行個體	若要建立更多堆疊執行個體，請刪除您不需要的堆疊執行個體，或請求提高 AWS 帳戶的堆疊執行個體數目上限。如需詳細資訊，請參閱《AWS》中的 AWS 一般參考服務配額 。

配額	Description	Value	調校策略
StackSets 執行個體操作	每個管理員帳戶可同時在每個區域中 (跨所有堆疊集) 執行作業的堆疊執行個體數目上限。	10,000 次操作	此限制適用於區域中涉及的所有堆疊集。包括受堆疊集建立和更新操作所影響的堆疊執行個體，以及直接建立、更新或刪除堆疊執行個體。
StackSets 佇列操作	指定時間內堆疊集的佇列操作數目上限。	10,000 次操作	
每個堆疊集操作使用 S3 物件匯入的堆疊	每個堆疊集操作可以使用 S3 物件匯入的堆疊數量上限。	200 個堆疊	
每個堆疊集操作使用內嵌堆疊 ID 匯入的堆疊	每個堆疊集操作可以使用內嵌堆疊 ID 匯入的堆疊數量上限。	10 個堆疊	
請求的範本內文大小	您可以在 CreateStack、UpdateStack 或 ValidateTemplate 請求中傳遞的範本內文大小上限。	51,200 位元組	若要使用較大的範本內文，請使用例如 巢狀堆疊 ，將您的範本分成多個範本。或將範本上傳至 Amazon S3 儲存貯體。
Amazon S3 物件中的範本內文大小	您可以在 CreateStack、UpdateStack、ValidateTemplate 請求的 Amazon S3 物件中以 Amazon S3 範本 URL 傳遞的範本內文大小上限。	1 MB	若要使用較大的範本內文，請使用例如 巢狀堆疊 ，將您的範本分成多個範本。或者使用縮小來減少 CloudFormation 範本的大小。

配額	Description	Value	調校策略
範本描述	範本描述的大小上限。 。	1,024 位元組	
每個勾點的版本數	可以為每個勾點建立的版本數量上限。	100 個版本	

功能可用性

並非每個區域都提供 CloudFormation 的所有功能。如需有關 AWS 區域的詳細資訊，請參閱[全球基礎設施區域表](#)。

- 目前無法在下列區域中使用[巨集](#)：
 - 亞太地區 (雅加達)
- 目前在下列區域無法使用[透過 CodeDeploy 使用 CloudFormation 執行 ECS 藍色/綠色部署](#)：
 - 非洲 (開普敦)
 - 亞太地區 (大阪)
 - 歐洲 (米蘭)

StackSets 和巨集

StackSets 目前不支援從包含巨集的範本建立或更新具有服務管理許可之堆疊集。其中包括轉換，這是由 CloudFormation 託管的巨集。如需巨集的詳細資訊，請參閱[範本巨集](#)。

疑難排解 CloudFormation 的問題

使用 CloudFormation 時，在建立、更新或刪除 CloudFormation 堆疊時您可能會遇到問題。下列各節可協助您故障診斷一些您可能會遇到的常見問題。

如需有關 CloudFormation 的一般問題資訊，請參閱 [CloudFormation 常見問答集](#)。您也可以在此 AWS re:Post 上的 [CloudFormation 社群論壇](#) 中搜尋解答和張貼問題。

主題

- [故障診斷指南](#)
- [故障診斷錯誤](#)
- [聯絡支援](#)

故障診斷指南

如果 CloudFormation 無法建立、更新或刪除您的堆疊，您可以檢視錯誤訊息或日誌，協助您進一步了解該問題。下列任務說明用於 CloudFormation 問題疑難排解的一般方法。如需特定錯誤和解決方案的資訊，請參閱 [故障診斷錯誤](#) 一節。

- 使用操作 IDs 和故障篩選進行目標故障診斷。CloudFormation 會將唯一的操作 IDs 指派給部署事件，讓您更輕鬆地追蹤和關聯相關事件。您可以依操作 ID 篩選堆疊事件，以專注於特定部署階段，並使用失敗篩選條件快速識別部署失敗的根本原因。請參閱 [依操作檢視堆疊事件](#)
- 使用 CloudFormation 主控台來檢視堆疊的狀態。在主控台中，您可以檢視堆疊事件的清單，同時建立、更新或刪除您的堆疊。堆疊事件包含操作 IDs，可協助您關聯相關事件並追蹤個別操作。您可以按一下操作 ID 進行重點疑難排解。從操作事件清單中，尋找失敗事件，然後檢視該事件的狀態原因。狀態原因可能包含來自 CloudFormation 或來自可協助您故障診斷之特定服務的錯誤訊息。如需使用操作 IDs 檢視堆疊事件的詳細資訊，請參閱 [依操作檢視堆疊事件](#)。
- 針對 Amazon EC2 問題，請檢視 cloud-init 和 cfn 日誌。這些日誌發佈於 Amazon EC2 執行個體的 `/var/log/` 目錄中。這些日誌會在 CloudFormation 設定您的執行個體時擷取程序和命令輸出。針對 Windows，請檢視位於 `%ProgramFiles%\Amazon\EC2ConfigService` 的 EC2Configure 服務、位於 `%ProgramData%\Amazon\EC2-Windows\Launch\Logs` 的 EC2 Launch、位於 `%ProgramData%\Amazon\EC2Launch\log` 的 EC2 Launch v2 和位於 `C:\cfn\log` 的 cfn 日誌。

您也可以設定 CloudFormation 範本，以便將日誌發佈至 Amazon CloudWatch，這會在 [中顯示日誌](#) AWS 管理主控台，因此您不需要連線到 Amazon EC2 執行個體。如需詳細資訊，請參閱應用程式管理部落格中的[在主控台中檢視 CloudFormation 日誌](#)。

故障診斷錯誤

當您遇到下列 CloudFormation 堆疊的錯誤，您可以使用下列解決方案來協助您尋找問題來源並將其修復。

主題

- [刪除堆疊失敗](#)
- [相依性錯誤](#)
- [AWS Config 和 AWS Systems Manager 衝突](#)
- [傳遞清單時剖析參數出現錯誤](#)
- [IAM 許可不足](#)
- [無效的值或不支援的資源屬性](#)
- [超過配額](#)
- [巢狀堆疊停滯於 UPDATE_COMPLETE_CLEANUP_IN_PROGRESS、UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS 或 UPDATE_ROLLBACK_IN_PROGRESS](#)
- [無需執行更新](#)
- [在建立、更新或刪除堆疊操作期間，資源無法穩定](#)
- [安全群組不存在於 VPC](#)
- [更新復原失敗](#)
- [等待條件未收到來自 Amazon EC2 執行個體所需數目的訊號](#)
- [從堆疊中移除但未刪除的資源](#)

刪除堆疊失敗

為解決此狀況，請嘗試下列作業：

- 某些資源在刪除之前必須先清空。例如，您必須刪除 Amazon S3 儲存貯體中的所有物件或移除 Amazon EC2 安全群組中的所有執行個體，才能刪除該儲存貯體或安全群組。

- 請確保您擁有必要 IAM 許可來刪除堆疊中的資源。除了 CloudFormation 許可外，您也必須有權使用基礎服務 (例如 Amazon S3 或 Amazon EC2)。
- 當堆疊因為 CloudFormation 無法刪除資源而處於 DELETE_FAILED 狀態時，請使用 RetainResources 參數並指定 CloudFormation 無法刪除的資源，來重新執行[刪除](#)。CloudFormation 刪除堆疊時不會刪除保留資源。如果無法刪除某個資源 (如包含您希望保留之物件的 S3 儲存貯體)，但仍需要刪除堆疊，則保留資源很有用。

刪除堆疊之後，您可以使用其相關聯的 AWS 服務手動刪除保留的資源。

或者，您可考慮將 FORCE_DELETE_STACK 選項與 DeletionMode 參數搭配使用。如需強制刪除閘道的詳細資訊，請參閱 [從 CloudFormation 主控台中刪除堆疊](#)。

- 您無法刪除啟用終止保護的堆疊。如果您嘗試刪除啟用終止保護的堆疊，則刪除會失敗，且堆疊及其狀態保持不變。請在堆疊上停用終止保護，然後再次執行刪除操作。

這包括根堆疊已啟用終止保護的[巢狀堆疊](#)。請在根堆疊上停用終止保護，然後再次執行刪除操作。強烈建議您不要直接刪除巢狀堆疊，而只在刪除根堆疊及其所有資源時作為其一部分來刪除。

如需詳細資訊，請參閱[防止 CloudFormation 堆疊遭到刪除](#)。

- 對於所有其他問題，如果您有 AWS 支援，您可以建立 支援 案例。請參閱 [聯絡支援](#)。

相依性錯誤

若要解決相依性錯誤，請將 DependsOn 屬性新增至依賴於您範本中其他資源的資源。在某些情況下，您必須明確宣告相依性，以便 CloudFormation 可以按正確的順序建立或刪除資源。例如，如果您使用網際網路閘道在相同的堆疊中建立彈性 IP 和 VPC，則該彈性 IP 必須依賴於網際網路閘道附件。如需詳細資訊，請參閱 [DependsOn 屬性](#)。

AWS Config 和 AWS Systems Manager 衝突

AWS Config 和 AWS Systems Manager 可以自動化基礎設施管理任務，這可能會導致與 CloudFormation 堆疊的部署發生衝突。執行下列動作，以避免任何潛在的衝突：

- 檢閱相關聯 AWS 帳戶 AWS Config 和 中 Systems Manager 的組態 AWS 區域。
- 檢查在 CloudFormation 部署期間，可能觸發的作用中規則或自動化文件。這些有可能會導致衝突或資源相依性與您的部署衝突。
- 檢查您的 CloudFormation 範本是否有任何由 AWS Config 和 Systems Manager 管理的資源。檢查是否有潛在的重疊或相互依存性，並考慮調整範本或自動化組態，以避免衝突。

- 在 CloudFormation 部署期間暫時停用或暫停任何相關 AWS Config 規則或 Systems Manager 自動化。請記得在成功部署後，還原原始組態，以維持所需的自動化和合規層級。
- 檢閱 CloudFormation 日誌和錯誤訊息，以取得與 AWS Config 和 Systems Manager 相關問題的任何參考，以協助找出衝突的來源。

如需 AWS Config 規則的詳細資訊，請參閱[使用 評估資源 AWS Config 規則](#)。

如需 Systems Manager 自動化的詳細資訊，請參閱[AWS Systems Manager 自動化](#)。

傳遞清單時剖析參數出現錯誤

當您使用 AWS CLI 或 CloudFormation 傳入清單時，請在每個逗號之前新增逸出字元 (\)。下列範例說明如何在使用 AWS CLI 時指定輸入參數。

```
ParameterKey=CIDR,ParameterValue='10.10.0.0/16\,10.10.0.0/24\,10.10.1.0/24'
```

IAM 許可不足

當您使用 CloudFormation 堆疊時，您不僅需要有使用 CloudFormation 的許可，您還必須有使用範本中所述之基礎服務的許可。例如，如果您建立 Amazon S3 儲存貯體或啟動 Amazon EC2 執行個體，您需要 Amazon S3 或 Amazon EC2 的許可。先檢閱您的 IAM 政策並驗證您擁有必要的許可，再使用 CloudFormation 堆疊。如需詳細資訊，請參閱[使用 控制 CloudFormation 存取 AWS Identity and Access Management](#)。

無效的值或不支援的資源屬性

當您建立或更新 CloudFormation 堆疊時，堆疊會因為無效的輸入參數、不支援的資源屬性名稱，或不支援的資源屬性值而失敗。針對輸入參數，請驗證資源是否存在。例如，當您指定 Amazon EC2 金鑰對或 VPC ID 時，資源必須存在於您建立或更新堆疊的帳戶和區域中。您可以使用[AWS 特定的參數類型](#)，以確保您使用有效的值。

針對資源屬性名稱和值，請更新您的範本，以使用有效的名稱和值。如需詳細資訊，請參閱《[AWS 資源和屬性類型參考](#)》。

超過配額

確認您未達到資源配額。例如，您可以啟動的 Amazon EC2 隨需執行個體預設數目上限為 5。如果嘗試建立的 Amazon EC2 隨需執行個體數量超出帳戶配額，則執行個體建立會失敗，且您會收到錯誤

Status=start_failed。若要依服務檢視預設的 AWS 配額，請參閱 AWS 一般參考 中的 [AWS 服務配額](#)。

針對 CloudFormation 配額和調整策略，請參閱 [了解 CloudFormation 配額](#)。

此外，如果在更新期間替換資源，則 CloudFormation 會先建立新的資源，然後刪除舊資源。此替換可能會讓您的帳戶超過資源配額，而導致您的更新失敗。您可以刪除多餘的資源，或請求[提高配額](#)。

巢狀堆疊停滯於

UPDATE_COMPLETE_CLEANUP_IN_PROGRESS、UPDATE_ROLLBACK_COMPLETE或 UPDATE_ROLLBACK_IN_PROGRESS

巢狀堆疊無法復原。由於各巢狀堆疊之間存在潛在的資源相依性，因此在更新或復原所有巢狀堆疊之前，CloudFormation 不會開始清理巢狀資源。當巢狀堆疊無法復原時，無論其他巢狀堆疊處於何種狀態，CloudFormation 都會取消所有操作。巢狀堆疊已完成更新或復原，但由於另一個巢狀堆疊無法復原而導致該堆疊未收到 CloudFormation 發出開始清除的訊號，則該堆疊會處於 UPDATE_COMPLETE_CLEANUP_IN_PROGRESS 或 UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS 狀態。更新失敗的巢狀堆疊未收到開始復原訊號，會處於 UPDATE_ROLLBACK_IN_PROGRESS 狀態。

由於巢狀堆疊可能曾經在 CloudFormation 進行變更而無法復原，此時堆疊範本無法準確反映堆疊的狀態。如果巢狀堆疊中的 Auto Scaling 群組在建立或更新時沒有足夠資源訊號逾時期間，則該巢狀堆疊也可能無法復原。

若要修正堆疊，請聯絡 [AWS 支援](#)。

無需執行更新

若要更新 CloudFormation 堆疊，您必須向 CloudFormation 提交範本或參數值的變更。不過，CloudFormation 不會將某些範本變更視為更新，例如刪除政策、更新政策、條件宣告或輸出宣告等變更。如果您需要做此類變更但無需其他變更，您可以新增或修改任何資源的中繼資料屬性。中繼資料屬性可是任何任意值，因為 CloudFormation 不會解譯其內容。如需詳細資訊，請參閱 [Metadata 屬性](#)。

在建立、更新或刪除堆疊操作期間，資源無法穩定

資源未回應，因為操作超過 CloudFormation 逾時期間或 AWS 服務中斷。對於服務中斷，[請檢查](#)相關 AWS 服務是否正在執行，然後重試堆疊操作。

如果 AWS 服務已成功執行，請檢查您的堆疊是否包含下列其中一個資源：

- `AWS::AutoScaling::AutoScalingGroup` 用於建立、更新和刪除操作
- `AWS::CertificateManager::Certificate` 用於建立操作
- `AWS::CloudFormation::Stack` 用於建立、更新和刪除操作
- `AWS::ElasticSearch::Domain` 用於更新操作
- `AWS::RDS::DBCluster` 用於建立和更新操作
- `AWS::RDS::DBInstance` 用於建立、更新和刪除操作
- `AWS::Redshift::Cluster` 用於更新操作

這些資源的操作時間可能會超過預設逾時期間。逾時期間取決於您使用的資源和憑證。若要延長逾時期間，請在執行堆疊操作時指定[服務角色](#)。如果您已使用服務角色，或您的堆疊包含未列出的資源，請聯絡 [AWS 支援](#)。

如果您的堆疊處於 `UPDATE_ROLLBACK_FAILED` 狀態，請參閱[更新復原失敗](#)。

安全群組不存在於 VPC

請確認您指定的 VPC 中存在安全群組。如果安全群組存在，請確認您指定安全群組 ID，而不是安全群組名稱。例如，`AWS::EC2::SecurityGroupIngress` 資源具有 `SourceSecurityGroupName` 和 `SourceSecurityGroupId` 屬性。針對 VPC 安全群組，您必須使用 `SourceSecurityGroupId` 屬性，並指定安全群組 ID。

更新復原失敗

依存資源無法返回其原始狀態，導致復原失敗 (`UPDATE_ROLLBACK_FAILED` 狀態)。例如，您可能有一個正在復原到已在 CloudFormation 之外刪除之舊資料庫執行個體的堆疊。因為 CloudFormation 並不知道資料庫已刪除，因此會假設此資料庫執行個體仍然存在，並嘗試復原回該資料庫執行個體，造成更新復原失敗。

根據失敗的原因，您可以手動修正錯誤並繼續進行復原。透過繼續執行復原，您便可以讓您的堆疊回到運作狀態 (`UPDATE_ROLLBACK_COMPLETE` 狀態)，然後再次嘗試更新堆疊。針對造成更新復原失敗常見的錯誤，下列清單說明解決方案：

- 無法接收所需數目的訊號

使用 [signal-resource](#) 命令對等待訊號的資源手動傳送所需數目的成功訊號，然後繼續復原更新。例如，在更新復原期間，Auto Scaling 群組中的執行個體可能無法在指定逾時期間內成功發出訊號。請手動傳送成功訊號到 Auto Scaling 群組。當您繼續執行更新復原時，CloudFormation 會收到您的訊號並繼續復原。

- 在 CloudFormation 外變更資源

請手動同步資源，使其符合原始堆疊的範本，然後繼續復原更新。例如，如果您手動刪除 CloudFormation 嘗試復原到的資源，您必須在原始堆疊中手動建立具有相同名稱和屬性的資源。

- 許可不足

確認您有足夠的 IAM 許可來修改資源，然後繼續更新復原。例如，您的 IAM 政策可能可讓您建立 S3 儲存貯體，但不能修改該儲存貯體。將修改動作新增至您的政策。

- 無效的安全權杖

CloudFormation 需要一組新的登入資料。無需做出任何變更。繼續復原更新，這會重新整理憑證。

- 限制錯誤

刪除您不需要的資源，或請求[提高配額](#)，然後繼續復原更新。例如，如果您帳戶的 EC2 隨需執行個體數目配額為 5，而更新復原超過該配額，則復原將會失敗。

- 資源不穩定

資源未回應，因為操作可能已超過 CloudFormation 逾時期間，或是 AWS 服務可能已中斷。無需做出任何變更。資源操作完成或 AWS 服務恢復操作後，請繼續復原更新。

若要繼續復原更新，您可以使用 CloudFormation 主控台或 AWS 命令列介面 (AWS CLI)。如需詳細資訊，請參閱[繼續復原更新](#)。

如果這些解決方案都無效，您可以略過 CloudFormation 無法成功復原的資源。如需詳細資訊，請參閱《AWS CloudFormation API 參考》中 [ContinueUpdateRollback](#) API 操作的 `ResourcesToSkip` 參數。CloudFormation 會將指定資源的狀態設為 `UPDATE_COMPLETE` 並繼續復原堆疊。在復原完成後，跳過資源的狀態會和堆疊範本中資源的狀態不一致。在您執行下一次堆疊更新前，您必須修改資源或更新堆疊，使其和彼此保持一致。若您未這麼做，後續的堆疊更新可能會失敗，使堆疊無法復原。

等待條件未收到來自 Amazon EC2 執行個體所需數目的訊號

為解決此狀況，請嘗試下列作業：

- 請確認您使用的 AMI 已安裝 CloudFormation 協助程式指令碼。如果 AMI 不包含協助程式指令碼，您也可以將指令碼下載到您的執行個體。如需詳細資訊，請參閱《CloudFormation 範本參考指南》中的 [CloudFormation 協助程式指令碼參考](#)。

- 請確認已成功在執行個體上執行 `cfn-signal` 命令。您可以檢視日誌，例如 `/var/log/cloud-init.log` 或 `/var/log/cfn-init.log`，以協助您除錯執行個體啟動。您可以登入您的執行個體來擷取日誌，但您必須[停用失敗時的復原](#)，否則 CloudFormation 會在堆疊建立失敗後刪除執行個體。您也可以[發佈日誌](#)到 Amazon CloudWatch。針對 Windows，您可以檢視 `C:\cfn\log` 中的 `cfn` 日誌和 `%ProgramFiles%\Amazon\EC2ConfigService` 中的 `EC2Config` 服務日誌。
- 請確認該執行個體已連線到網際網路。如果執行個體位於 VPC，則執行個體可以透過 NAT 裝置 (在私有子網路中時) 或透過網際網路閘道 (在公有子網路中時) 連線到網際網路。若要測試執行個體的網際網路連線，請嘗試存取公有網頁，例如 `http://aws.amazon.com`。例如，您可以在執行個體上執行下列命令。它應會傳回 HTTP 200 狀態碼。

```
curl -I https://aws.amazon.com
```

如需有關設定 NAT 裝置的資訊，請參閱《Amazon VPC 使用者指南》中的 [NAT](#)。

從堆疊中移除但未刪除的資源

在堆疊更新期間，CloudFormation 已從堆疊中移除資源，但未刪除資源。資源仍然存在，但無法再透過 CloudFormation 存取。這可能會發生在堆疊更新期間，其中：

- CloudFormation 需要取代現有資源，因此它會先建立新的資源，然後嘗試刪除舊的資源。
- 您已經從堆疊範本中移除資源，因此 CloudFormation 會嘗試將資源從堆疊中刪除。

不過，在某些情況下，CloudFormation 可能無法刪除資源。例如，如果使用者沒有刪除指定類型資源的許可。

CloudFormation 會嘗試三次刪除舊資源。如果 CloudFormation 無法刪除舊資源，即會從堆疊中移除舊資源並繼續更新堆疊。堆疊更新完成時，CloudFormation 會發出 `UPDATE_COMPLETE` 堆疊事件，但內含 `StatusReason`，其會指出無法刪除一或多個資源。CloudFormation 也會針對特定資源發出 `DELETE_FAILED` 事件，內含相應的 `StatusReason`，其會說明 CloudFormation 為何無法刪除此資源的更多詳細資訊。

若要解決此情況，請直接使用主控台或基礎服務的 API 來刪除資源。

聯絡支援

如果您有 AWS 支援，您可以在 [https://console.aws.amazon.com/support/home/#/](https://console.aws.amazon.com/support/home#/) 建立技術支援案例。在您聯絡支援人員前，請收集下列資訊：

- 堆疊 ID。您可以在 [CloudFormation 主控台](#) 的概觀索引標籤上找到堆疊 ID。如需詳細資訊，請參閱 [在 CloudFormation 主控台中檢視堆疊資訊](#)。

 Important

請勿在 CloudFormation 外變更堆疊中的資源。在 CloudFormation 外變更您的堆疊中的資源可能會使其處於不可復原的狀態。

- 任何堆疊錯誤訊息。如需檢視堆疊錯誤訊息的資訊，請參閱 [故障診斷指南](#) 一節。
- 針對 Amazon EC2 問題，請檢視 cloud-init 和 cfn 日誌。這些日誌發佈於 Amazon EC2 執行個體的 /var/log/ 目錄中。這些日誌會在設定您的執行個體時擷取程序和命令輸出。針對 Windows，請在 %ProgramFiles%\Amazon\EC2ConfigService 和 C:\cfn\log 中收集 EC2Configure 服務和 cfn 日誌。

您也可以在此 AWS re:Post 上的 [CloudFormation 社群論壇](#) 中搜尋解答和張貼問題。

使用 Amazon Q Developer，對失敗的 CloudFormation 堆疊部署進行疑難排解

Amazon Q Developer 是生成式人工智慧 (AI) 支援的對話式助理，可協助您了解、建置、擴展和操作 AWS 應用程式。如需有關 Amazon Q Developer 的詳細資訊，請參閱 [Amazon Q Developer 使用者指南](#)。

本文件說明如何使用 Amazon Q Developer 來協助您疑難排解失敗的 CloudFormation 部署。此功能旨在協助您快速發現和解決堆疊部署期間遇到的問題。

由 Amazon Bedrock 提供支援：Amazon Q 以 Amazon Bedrock 為基礎建置，包含在 Amazon Bedrock 中實作的 [自動濫用偵測](#) 功能，來加強安全性、資安和負責任的人工智慧 (AI) 運用。

主題

- [功能](#)
- [運作方式](#)
- [定價](#)
- [區域可用性](#)
- [限制](#)

功能

使用 Amazon Q Developer 解決方案對堆疊部署進行疑難排解可提供下列功能：

- 錯誤分析 – 提供易於理解的 CloudFormation 堆疊部署錯誤分析。
- 解決建議 – 提供逐步指示來解決發現的問題。

運作方式

當 CloudFormation 堆疊建立或更新失敗時，錯誤訊息會顯示在 AWS 管理主控台中。

可以選擇錯誤訊息旁的使用 Q 進行診斷按鈕。

Amazon Q Developer 會分析錯誤並提供問題說明。

如需其他協助，請選擇協助我解決問題，以接收詳細的解決步驟。

定價

如需有關定價的資訊，請參閱 [Amazon Q Developer 定價](#)。

區域可用性

Amazon Q Developer 適用於大多數 AWS 區域。如需目前可使用 Amazon Q Developer 的所有區域清單，請參閱《Amazon Q Developer 使用者指南》中的 [Amazon Q Developer 的支援區域](#)。

限制

- 只能透過 AWS 管理主控台來存取此功能。
- 此功能涵蓋前 20 個最常見錯誤。它未涵蓋所有可能的錯誤。
- 複雜的故障診斷案例仍可能需要支援介入。

AWS CloudFormation 使用者指南的文件歷史記錄

下表說明 2018 年 5 月之後 AWS CloudFormation 使用者指南內容的重要變更。若要接收有關此文件更新的通知，可以訂閱 RSS 摘要。

Important

從 2018 年 5 月起，說明範本參考內容更新的資料表列已移至新的 AWS CloudFormation 範本參考。如需這些更新和任何未來的變更，請參閱[AWS CloudFormation 範本參考中的範本參考的文件歷史記錄](#)。AWS CloudFormation

變更	描述	日期
提供 PDF 指南	現在可以下載 PDF 格式的《AWS CloudFormation 使用者指南》。	2025 年 5 月 30 日
已將範本參考內容移至新指南	CloudFormation 已發佈 CloudFormation 範本參考指南。如需詳細資訊，請參閱 CloudFormation 範本參考指南 。	2025 年 5 月 30 日
IaC 產生器支援部分掃描	現在可以選擇要掃描的特定資源類型，讓您更輕鬆地從現有資源產生基礎結構即程式碼 (IaC) 範本。如需詳細資訊，請參閱 使用 CloudFormation IaC 產生器啟動資源掃描 。	2025 年 3 月 27 日
堆疊重構	堆疊重構可簡化 CloudFormation 堆疊中的資源重組，同時仍保留現有的資源屬性和資料。如需詳細資訊，請參閱 堆疊重構 。	2025 年 2 月 6 日

[使用 Amazon Q Developer 對堆疊部署進行疑難排解](#)

現在可以使用 Amazon Q Developer 在部署 CloudFormation 堆疊時疑難排解常見錯誤。如需詳細資訊，請參閱[使用 Amazon Q Developer 對失敗的 CloudFormation 堆疊部署進行疑難排解](#)。

2024 年 11 月 22 日

[堆疊部署時間軸圖表](#)

您現在可以看到堆疊部署的視覺化呈現。堆疊部署時間軸圖表顯示堆疊部署狀態、個別資源部署狀態以及部署狀態變更的時間。如需詳細資訊，請參閱[檢視 CloudFormation 堆疊部署的時間軸](#)。

2024 年 11 月 11 日

[視覺化掃描的資源和產生的範本](#)

透過視覺化掃描摘要詳細資訊並在部署基礎設施堆疊之前預覽產生的範本，現在可以簡化基礎結構即程式碼 (IaC) 產生器工作流程。如需詳細資訊，請參閱[在 CloudFormation 主控台中檢視掃描摘要](#)和[從掃描的資源中建立 CloudFormation 堆疊](#)。

2024 年 8 月 22 日

[Amazon EventBridge 與 AWS CloudFormation Git 同步整合](#)

AWS CloudFormation Git 同步現在會將同步狀態變更作為事件發佈至 Amazon EventBridge。如需詳細資訊，請參閱[儲存庫同步狀態變更事件詳細資訊](#)和[資源同步狀態變更事件詳細資訊](#)。

2024 年 7 月 29 日

[強制刪除停滯的堆疊](#)

強制刪除堆疊的兩個新選項可用於處於停滯狀態的堆疊刪除操作。現在可以選擇強制刪除堆疊但保留資源，或強制刪除整個堆疊。如需詳細資訊，請參閱[從 CloudFormation 主控台中刪除堆疊](#)。

2024 年 5 月 22 日

[AWS CloudTrail 事件堆疊操作根本原因](#)

CloudFormation 透過新的 AWS CloudTrail 深層連結整合來改善堆疊操作的疑難排解體驗。此功能會將 CloudFormation 主控台內的堆疊操作事件直接連結至相關的 CloudTrail 事件。如需詳細資訊，請參閱[確定堆疊失敗的原因](#)。

2024 年 5 月 15 日

[屬性層級變更集](#)

屬性層級變更集可讓您預覽 CloudFormation 部署對資源屬性值所做的變更。如需詳細資訊，請參閱[檢視 CloudFormation 堆疊的變更集](#)。

2024 年 4 月 12 日

[CloudFormation 引入了 CONFIGURATION_COMPLETE 事件](#)

現在可以使用 CONFIGURATION_COMPLETE 事件來啟用更快速的工作流程，包括建立資源。如需詳細資訊，請參閱[了解 CloudFormation 堆疊建立事件](#)。

2024 年 3 月 11 日

[從現有 AWS 資源產生 AWS CloudFormation 範本和 AWS CDK 應用程式](#)

現在可以使用您帳戶中已佈建且尚未由 CloudFormation 管理的資源來產生範本。如需詳細資訊，請參閱[使用 IaC 產生器從現有資源中產生範本](#)。

2024 年 2 月 2 日

[StackSets 並行模式](#)

並行模式是 `StackSetOperationPreference` 的參數 `StackSetOperationPreference`，可讓您選擇並行層級在堆疊集操作期間的行為。如需詳細資訊，請參閱[選擇 CloudFormation StackSets 的並行模式](#)。

2023 年 11 月 9 日

[詳細的 StackSet 漂移資訊](#)

下列 API 可讓您查看哪些堆疊執行個體已從 StackSet 範本漂移，以及哪些資源已經漂移。

2023 年 7 月 24 日

[ListStackInstanceResourceDrifts](#)

為堆疊執行個體中的資源傳回漂移資訊。

[StackInstanceResourceDrifts Summary](#)

結構包含與某個堆疊執行個體的資源漂移有關的摘要資訊。

[CloudFormation StackSets APIs控制 AWS Organizations 信任存取](#)

CloudFormation StackSets 為客戶提供以下 API，用於管理 AWS Organizations 信任存取：

2023 年 6 月 5 日

[ActivateOrganizationsAccess](#)

使用 啟用受信任存取 AWS Organizations。啟用 StackSets 和 Organizations 之間的受信任存取後，管理帳戶會擁有為您的組織建立和管理 StackSets 的許可。

[DeactivateOrganizationsAccess](#)

使用 停用受信任存取 AWS Organizations。如果停用受信任存取，管理帳戶就會失去許可，無法為您的組織建立及管理服務代管的 StackSets。

[DescribeOrganizationsAccess](#)

擷取有關帳戶 OrganizationAccess 狀態的資訊。管理帳戶或委派的管理員可透過使用 CallAs 參數呼叫此 API。管理帳戶也可以在沒有 CallAs 參數的情況下呼叫此 API。

[DescribeStackSet API](#)

DescribeStackSet API 有一項「區域」清單的新參數，在該清單中顯示指定的堆疊集。如需詳細資訊，請參閱[DescribeStackSet](#)。

2023 年 2 月 1 日

[使用 CloudFormation 和 Amazon EventBridge 來管理 StackSets 事件](#)

CloudFormation StackSets 會透過 Amazon EventBridge 啟動事件通知。在建立、更新或刪除 CloudFormation 堆疊集之後，您可以觸發事件驅動的動作。如需詳細資訊，請參閱[使用 EventBridge 來監控 CloudFormation 和 Git 同步事件](#)。

2022 年 11 月 16 日

[為堆疊集操作改善有關堆疊執行個體的洞見](#)

CloudFormation StackSets 針對堆疊集操作提供有關堆疊執行個體的更詳細資訊：

2022 年 11 月 4 日

[DescribeStackSetOperation](#)

您現在可以使用 `DescribeStackSetOperation` 在部署期間為堆疊集操作提供失敗的堆疊執行個體計數。

[ListStackInstances](#)

您現在可以使用篩選選項 `LastOperationID` 為堆疊集操作列出堆疊執行個體。

[使用 CloudFormation 和 Amazon EventBridge 來管理事件](#)

在特定 CloudFormation 事件發生時接收通知。如需詳細資訊，請參閱[使用 EventBridge 來監控 CloudFormation 和 Git 同步事件](#)。

2022 年 7 月 20 日

[帳戶層級](#)

CloudFormation 宣布正式推出帳戶篩選條件類型，此功能可讓客戶將部署目標限制為個別帳戶，或包含具有提供的 OU 的其他帳戶。如需詳細資訊，請參閱[服務受管 StackSets 的帳戶層級目標](#)。

2022 年 7 月 7 日

[CloudFormation 登錄檔](#)

CloudFormation 宣布正式推出 Hooks，此功能可讓客戶在建立、更新或刪除堆疊操作之前叫用自訂邏輯以自動化動作或檢查資源組態。如需詳細資訊，請參閱《[CloudFormation Hooks 使用者指南](#)》。

2022 年 2 月 10 日

[堆疊失敗選項](#)

在發生佈建失敗時，您可以從失敗點開始，而不復原成功佈建的資源，以反覆開發應用程式。透過指定堆疊失敗選項，您可以疑難排解處於 CREATE_FAILED 或 UPDATE_FAILED 狀態的資源。您可以為所有堆疊部署和變更集操作佈建失敗選項。如需詳細資訊，請參閱[選擇在佈建資源時如何處理失敗](#)。

2021 年 8 月 30 日

[將堆疊匯入堆疊集](#)

現在可以將現有堆疊匯入新的或現有的堆疊集。如需詳細資訊，請參閱[將堆疊匯入 CloudFormation StackSets](#)。

2021 年 7 月 28 日

[增加的配額](#)

現在可以在 AWS 帳戶帳戶中宣告堆疊的預設數目上限 2000。如需詳細資訊，請參閱[了解 CloudFormation 配額](#)。

2021 年 7 月 15 日

[發佈公有第三方擴充功能](#)

您現在可以使用第三方發佈者提供的公有擴充功能，就像從中擴充功能一樣 AWS。如需詳細資訊，請參閱[使用 CloudFormation 登錄檔中的第三方公有擴充功能](#)。

2021 年 6 月 21 日

[參考堆疊集範本中的巨集](#)

StackSets 現在支援從參考巨集的範本建立或更新具有自我管理許可的堆疊集。如需有關巨集的詳細資訊，請參閱[使用範本巨集在 CloudFormation 範本上執行自訂處理](#)。

2021 年 4 月 14 日

[使用動態參考中 Systems Manager 參數的最新值](#)

現在可以讓 CloudFormation 在建立或更新堆疊時使用最新版本的 Systems Manager 參數。您不再需要指定特定版本。如需詳細資訊，請參閱[從 Systems Manager 參數存放區中取得純文字值](#)。

2021 年 4 月 13 日

[模組支援在資源名稱中使用句號分隔符號](#)

您現在可以在指定模組中包含之資源的完整邏輯名稱時，使用句號作為分隔符號。如需詳細資訊，請參閱[CloudFormation 範本中的參考模組資源](#)。

2021 年 4 月 8 日

[CloudFormation StackSets 現在支援平行區域部署](#)

您現在可以選擇按順序或以平行方式將 StackSets 部署至區域。如需詳細資訊，請參閱[堆疊集操作選項](#)。

2021 年 4 月 6 日

[CloudFormation StackSets](#) [現在支援使用 的委派管理員](#) [AWS Organizations](#)

除了組織的 管理帳戶 外，委派管理員帳戶還可以為其組織建立和管理具有服務管理許可的堆疊集。如需詳細資訊，請參閱[註冊委派管理員會員帳戶](#)和[使用服務管理許可來建立 CloudFormation StackSets](#)。

2021 年 2 月 18 日

[CloudFormation StackSets 區域可用性](#)

亞太區域 (大阪) 區域現在已提供 CloudFormation StackSets。如需詳細資訊，請參閱[使用 StackSets 管理跨帳戶和跨區域的堆疊](#)。

2021 年 2 月 10 日

[模組](#)

模組可讓您以透明、易於管理且可重複的方式，來封裝可包含在堆疊範本之間的資源組態。模組可透過可自訂的模組化建置區塊形式，來封裝通用服務組態和最佳實務，以便您將其納入堆疊範本中。如需詳細資訊，請參閱[使用 CloudFormation 模組建立可跨範本包含的可重複使用資源組態](#)。

2020 年 11 月 24 日

[巢狀堆疊變更集](#)

透過巢狀堆疊變更集，您可以預覽整個巢狀堆疊階層中對應應用程式和基礎設施資源所做的變更，並在確認所有變更都如預期時繼續進行更新。若要取得更多資訊，請參閱[巢狀堆疊變更集](#)。

2020 年 11 月 18 日

[增加的配額](#)

已更新下列 AWS CloudFormation 配額。

2020 年 10 月 22 日

- 您現在可以在 AWS CloudFormation 範本中宣告映射的上限為 200。
- 您現在可以為 AWS CloudFormation 範本中的每個 200 映射宣告最多 100 個映射屬性。
- 您現在可以在 AWS CloudFormation 範本中宣告輸出上限為 200。
- 您現在可以在 AWS CloudFormation 範本中宣告參數上限為 200。
- 您現在可以在 AWS CloudFormation 範本中宣告資源上限為 500。
- 您現在可以在 Amazon S3 物件中傳遞範本主體，大小上限為 1 MB。

[私有資源的漂移偵測](#)

CloudFormation 現在支援對 AWS 資源的展開清單以及在 CloudFormation 儲存庫中定義為可佈建的私有資源進行漂移偵測操作。如需詳細資訊，請參閱[支援的資源類型](#)。

2020 年 10 月 1 日

[註冊資源提供者所需的更新許可](#)

在您的帳戶中註冊資源的提供者，現在需要您具有存取上傳至該資源提供者 S3 儲存貯體的架構處理常式套件權限。如需詳細資訊，請參閱[用於註冊第三方私有延伸模組的 IAM 許可](#)。

2020 年 8 月 7 日

[資源匯入支援可佈建的私有資源類型](#)

匯入作業現在支援 provisionable (可佈建) 的私有資源類型；意即為 FULLY_MUTABLE 或 IMMUTABLE 的佈建類型。如需詳細資訊，請參閱[支援匯入操作的資源](#)。

2020 年 6 月 3 日

[透過 CodeDeploy 進行 ECS 藍/綠部署](#)

您現在可以使用 CloudFormation 透過 CodeDeploy 進行 ECS 藍/綠部署。藍/綠部署是 AWS CodeDeploy 提供的安全部署策略，可將變更應用程式版本所造成的中斷降至最低。如需詳細資訊，請參閱[使用 CloudFormation 透過 CodeDeploy 執行 ECS 藍/綠部署](#)。

2020 年 5 月 19 日

[CloudFormation StackSets 區域可用性](#)

CloudFormation StackSets 現已在 AWS GovCloud (美國西部) 區域提供。

2020 年 5 月 18 日

[AWS CloudFormation StackSets 與整合 AWS Organizations](#)

現在可使用 StackSets 集中管理對您組織中所有帳戶或 AWS Organizations 中的特定組織單位 (OU) 的部署。您可以自動部署到新增至組織或 OU 的任何新帳戶。跨帳戶部署所需的許可將由 StackSets 自動處理。如需詳細資訊，請參閱[使用 StackSets 管理跨帳戶和跨區域的堆疊](#)。

2020 年 2 月 11 日

[StackSets 的漂移偵測](#)

您現在可以在堆疊集合以及其包含的所有堆疊執行個體上執行漂移偵測。如需詳細資訊，請參閱[在 CloudFormation StackSets 上執行漂移偵測](#)。

2019 年 11 月 19 日

[CloudFormation 登錄檔現已開放使用](#)

現在可使用 CloudFormation 主控台來檢帳戶中可供使用的私有和公有資源。如需詳細資訊，請參閱[檢視 CloudFormation 登錄檔中的可用和已啟用的擴充功能](#)。

2019 年 11 月 18 日

[CloudFormation 登錄檔 API 動作](#)

在 CloudFormation 登錄中用於管理類型的下列 API 動作已開放使用。

2019 年 11 月 18 日

[DeregisterType](#)

從 CloudFormation 登錄內正在作用的項目中移除類型或類型版本。

[DescribeType](#)

傳回註冊類型的詳細資訊。

[DescribeTypeRegistration](#)

傳回類型註冊的資訊，包括其目前狀態、類型和版本識別碼。

[ListTypeRegistrations](#)

傳回指定類型的註冊請求識別碼清單。

[ListTypes](#)

傳回已向 CloudFormation 註冊類型的摘要資訊。

[ListTypeVersions](#)

傳回類型版本的摘要資訊。

[RegisterType](#)

向 CloudFormation 登錄檔註冊類型。註冊一個類型，以便在 AWS 帳戶中的 CloudFormation 範本中使用。

[SetTypeDefaultVersion](#)

指定類型的預設版本。
類型的預設版本將會在
CloudFormation 操作中使用。

如需 CloudFormation 登錄
檔的詳細資訊，請參閱[使用
CloudFormation 登錄檔管理擴
充功能](#)

[已新增資源匯入](#)

如果您在 CloudFormation
管理之外建立 AWS 資源，
您可以使用 將此現有資源
帶入 CloudFormation 管
理resource import。如
需詳細資訊，請參閱[使用
AWS 資源匯入將資源匯入
CloudFormation 堆疊](#)。

2019 年 11 月 11 日

[提升堆疊集限額](#)

您現在可以在管理員帳戶中最
多建立 100 個堆疊集、針對
每個堆疊集最多建立 2000 個
堆疊執行個體，以及針對每個
管理員帳戶，同時在每個區域
執行最多 3500 個堆疊執行個
體操作。如需詳細資訊，請參
閱[了解 CloudFormation 配額](#)。

2019 年 8 月 2 日

[並行堆疊操作中資源的限制](#)

CloudFormation 現在對並行
堆疊操作中的資源數量強制執
行帳戶限制。此限制取決於區
域。如需詳細資訊，請參閱[了
解 CloudFormation 配額](#)。

2019 年 4 月 30 日

[堆疊執行個體操作限制](#)

對於 StackSets，現在每個管理員帳戶在任一區域中最多可同時執行 1500 個堆疊執行個體操作。如需詳細資訊，請參閱[了解 CloudFormation 配額](#)。

2018 年 12 月 13 日

[現在可使用 CAPABILITY_AUTO_EXPAND 功能](#)

現在可使用 CAPABILITY_AUTO_EXPAND 功能直接從包含巨集的堆疊範本建立或更新堆疊，而不需要先在變更集檢閱產生的變更。如需詳細資訊，請參閱《AWS CloudFormation API 參考》中的[CreateStack](#) 或 [UpdateStack](#)。

2018 年 12 月 7 日

[已新增堆疊漂移偵測](#)

現在可偵測堆疊的實際組態是否偏離 CloudFormation 內定義的預期範本組態。可以偵測整個堆疊或個別堆疊資源上的偏離。如需詳細資訊，請參閱[使用偏離偵測功能來偵測堆疊和資源的未受管組態變更](#)。

2018 年 11 月 13 日

[現在可使用 secretsmanager 動態參考](#)

現在可使用 secretsmanager 動態參考來擷取存放於 AWS Secrets Manager 中的全部秘密或秘密值。秘密可能是資料庫憑證、密碼、第三方 API 金鑰，甚至是任意文字。使用 secretsmanager 動態參考可保證 Secrets Manager 和 CloudFormation 都不記錄或存留任何已解析的秘密值。如需詳細資訊，請參閱[從 Secrets Manager 中取得秘密或秘密值](#)。

2018 年 11 月 9 日

[現在可使用巨集](#)

現在可使用巨集在範本上執行自訂處理，例如尋找和取代操作這種簡單動作，乃至於全部範本的大規模轉換。如需詳細資訊，請參閱[使用範本巨集在 CloudFormation 範本上執行自訂處理](#)。

2018 年 9 月 6 日

[CloudFormation 現在支援採用 AWS PrivateLink 技術的 VPC 端點](#)

您可以使用 VPC 端點在 VPC 與 CloudFormation 之間建立私有連線，而不需要透過網際網路、NAT 執行個體、VPN 連線或 Direct Connect 進行存取。如需詳細資訊，請參閱[使用介面端點 \(AWS PrivateLink\) 存取 CloudFormation](#)。

2018 年 8 月 22 日

[動態參考支援安全字串](#)

在您的堆疊範本中，現在可使用新的動態參考來指定在其他服務中存放和管理的值，包括 Systems Manager Parameter Store SecureString 類型參數。如需詳細資訊，請參閱[從 Systems Manager Parameter Store 取得安全字串值](#)。

2018 年 8 月 16 日

[堆疊集現在支援自訂的執行角色](#)

在目標帳戶中現在可使用自訂的執行角色來控制使用者或群組可在其堆疊集包含的堆疊資源。如需詳細資訊，請參閱[設定堆疊集操作的進階許可](#)。

2018 年 5 月 30 日

[堆疊執行個體的選擇性更新](#)

現在可使用選用的 Accounts 和 Regions 參數，來指定要在堆疊集更新操作期間更新其中堆疊執行個體的帳戶和區域。如需詳細資訊，請參閱 [CloudFormation API 參考中的 UpdateStackSet](#)。

2018 年 5 月 30 日

[CloudFormation 現在可以建立啟用加密的 S3 儲存貯體](#)

對於 CloudFormation 建立以存放上傳之堆疊範本的 Amazon S3 儲存貯體，現在預設會啟用伺服器端加密，藉此加密那些儲存貯體中存放的所有物件。如需詳細資訊，請參閱 [從 CloudFormation 主控台中建立堆疊](#)。

2018 年 5 月 24 日

[已新增 FIPS 端點](#)

CloudFormation 現在提供新的端點，其會在下列公有 US 區域中使用 FIPS 140-2 驗證的加密模組：US-East-1、US-East-2、US-West-1 和 US-West-2。請參閱 Amazon Web Services 一般參考 中的 [CloudFormation 端點和配額](#)，以取得新的 FIPS 相容端點 URL。

2018 年 5 月 17 日

如需 CloudFormation Hooks 使用者指南的更新，請參閱 [CloudFormation 《Hooks 使用者指南》中的 Hooks 使用者指南的文件歷史記錄](#)。CloudFormation

封存的更新

下表說明 AWS CloudFormation 使用者指南 2018 年 5 月之前每個版本的重要變更。

變更	版本日期	Description	API 版本
更新的資源	2019 年 7 月 22 日	使用 <code>encryptionOptions</code> 屬性指定 AWS 擁有的金鑰或適用於 Amazon MQ 代理程式的客戶受管 CMK。	2010-05-15
堆疊集命名慣例	2018 年 4 月 10 日	使用堆疊集建立的 CloudFormation 堆疊現在遵循新的命名慣例，其堆疊名稱包含堆疊集名稱。	2010-05-15
新資源	2018 年 4 月 10 日	AWS::AppSync::ApiKey 使用 <code>AWS::AppSync::ApiKey</code> 資源建立唯一金鑰，您可以將其分發給執行 GraphQL 操作的用戶端 AWS AppSync。 AWS::AppSync::DataSource 使用 <code>AWS::AppSync::DataSource</code> 資源為中的解析程式建立資料來源 AWS AppSync。 AWS::AppSync::GraphQLApi 使用 <code>AWS::AppSync::GraphQLApi</code> 資源建立新的 AWS AppSync GraphQL API。 AWS::AppSync::GraphQLSchema 使用 <code>AWS::AppSync::GraphQLSchema</code> 資源為您的 AWS AppSync GraphQL API 建立資料模型。 AWS::AppSync::Resolver 使用 <code>AWS::AppSync::Resolver</code> 資源來定義您將連接至結構描述之欄位的邏輯 GraphQL 解析程式。	2010-05-15
更新的資源	2018 年 4 月 10 日	AWS::Config::ConfigurationAggregator 使用 <code>OrganizationAggregationSource</code> 屬性類型來指定要彙總至 AWS Config 組態彙總工具 AWS Config 的資料區域，以及用於擷取 AWS Organizations 詳細資訊的 IAM 角色。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2018 年 4 月 4 日	AWS::Config::AggregationAuthorization 使用 <code>AWS::Config::AggregationAuthorization</code> 資源授予 彙總工具帳戶收集 AWS Config 資料的許可。 AWS::Config::ConfigurationAggregator 使用 <code>AWS::Config::ConfigurationAggregator</code> 資源為 AWS Config 建立組態彙總工具。	2010-05-15
堆疊集現在支援自訂的管理員角色	2018 年 3 月 29 日	使用自訂的管理員角色可控制哪些使用者或群組可管理相同管理員帳戶中的特定堆疊集。如需詳細資訊，請參閱設定堆疊集操作的進階許可。	2010-05-15
新資源	2018 年 3 月 29 日	AWS::EC2::LaunchTemplate 使用 <code>AWS::EC2::LaunchTemplate</code> 資源來建立 Amazon EC2 執行個體的啟動範本。	2010-05-15
更新的資源	2018 年 3 月 29 日	AWS::AutoScaling::AutoScalingGroup 使用 <code>LaunchTemplate</code> 屬性來指定用於啟動執行個體的啟動範本。 AWS::EC2::SpotFleet 在 SpotFleetRequestConfigData 屬性類型中，請使用 <code>LaunchTemplateConfigs</code> 屬性來說明啟動範本和覆寫。	2010-05-15
新的 <code>Fn::Cidr</code> 內部函數	2018 年 3 月 6 日	傳回指定的 Cidr 地址區塊。如需詳細資訊，請參閱Fn::Cidr。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2018 年 3 月 6 日	AWS::ApiGateway::VpcLink 使用 <code>AWS::ApiGateway::VpcLink</code> 資源來指定 <code>AWS::ApiGateway::RestApi</code> 的 API Gateway VPC 連結，以存取 Amazon Virtual Private Cloud (VPC) 中的資源。 AWS::GuardDuty::Master 使用 <code>AWS::GuardDuty::Master</code> 資源建立 GuardDuty 主要帳戶。 AWS::GuardDuty::Member 使用 <code>AWS::GuardDuty::Member</code> 資源建立 GuardDuty 成員帳戶。 AWS::SES::ConfigurationSet 使用 <code>AWS::SES::ConfigurationSet</code> 資源建立可套用至您傳送之電子郵件的規則群組。 AWS::SES::ConfigurationSetEventDestination 使用 <code>AWS::SES::ConfigurationSetEventDestination</code> 資源指定組態集事件目的地。 AWS::SES::ReceiptFilter 使用 <code>AWS::SES::ReceiptFilter</code> 資源指定是否接受或拒絕來自一組 IP 地址或一個 IP 地址範圍的郵件。 AWS::SES::ReceiptRule 使用 <code>AWS::SES::ReceiptRule</code> 資源來指定當 Amazon SES 代表您擁有的一或多個電子郵件地址或網域接收郵件時，應採取的動作。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::SES::ReceiptRuleSet 使用 <code>AWS::SES::ReceiptRuleSet</code> 資源為 Amazon SES 指定空的規則集。 AWS::SES::Template 使用 <code>AWS::SES::Template</code> 資源指定電子郵件的內容、主旨行的組成、HTML 部分以及純文字部分。	

變更	版本日期	Description	API 版本
更新的資源	2018 年 3 月 6 日	AWS::AutoScaling::AutoScalingGroup 使用 <code>AutoScalingGroupName</code> 屬性指定 Auto Scaling 群組的名稱。 AWS::ApiGateway::RestApi 使用 <code>ApiKeySourceType</code> 屬性指定根據用量計畫進行量測請求的 API 金鑰來源， 使用 <code>MinimumCompressionSize</code> 屬性指定一個可為 Null 的整數，它可用來啟用或停用 API 的壓縮。 AWS::ApplicationAutoScaling::ScalingPolicy 在 TargetTrackingScalingPolicyConfiguration 屬性類型中，請使用 <code>DisableScaleIn</code> 屬性指定是否停用依照目標追蹤政策來縮減規模。 AWS::EC2::SpotFleet 在 LaunchSpecifications 屬性類型中，請使用 <code>TagSpecifications</code> 屬性指定 SpotFleet 建立期間要套用的標籤。 AWS::Elasticsearch::Domain 使用 <code>Arn</code> 屬性使 <code>Fn::GetAtt</code> 傳回網域的 Amazon Resource Name (ARN)。 <code>DomainArn</code> 的 <code>Fn::GetAtt</code> 屬性已廢除。 AWS::RDS::DBCluster 使用 <code>DBClusterIdentifier</code> 屬性指定資料庫叢集識別碼。 AWS::RDS::DBCluster 使用 <code>DBClusterIdentifier</code> 屬性指定資料庫叢集識別碼。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::Redshift::Cluster 使用 <code>ClusterIdentifier</code> 屬性指定叢集的唯一識別碼。 AWS::Route53::HealthCheck 在 HealthCheckConfig 屬性類型中，使用 <code>Regions</code> 屬性來指定您希望 Route 53 運作狀態檢查工具檢查指定端點的區域。 AWS::SSM::Document 使用 <code>Tags</code> 屬性來指定要套用至文件的 CloudFormation 資源標籤。	
更新的資源	2018 年 2 月 19 日	AWS::CodeBuild::Project 使用 <code>Triggers</code> 屬性來設定專案的網路鉤手，以開始在每次程式碼變更被推送至儲存庫時，自動重建原始程式碼。這僅適用於 CloudFormation 中的 GitHub 專案。它不適用於 GitHub Enterprise 專案。	2010-05-15
更新的資源	2018 年 2 月 8 日	AWS::DynamoDB::Table 使用 <code>SSESpecification</code> 屬性來指定設定以啟用伺服器端加密。	2010-05-15
更新的資源	2018 年 2 月 5 日	AWS::CodeBuild::Project 在 Source CodeBuild Project Source 屬性類型中： - 使用 <code>GitCloneDepth</code> 屬性指定要下載的歷史記錄深度。 - 使用 <code>InsecureSsl</code> 屬性指定是否要忽略在連接至 GitHub Enterprise 專案儲存庫時出現的 SSL 警告。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2018 年 1 月 23 日	AWS::AutoScaling::LifecycleHook 使用 LifecycleHookName 屬性指定 lifecycle hook 的名稱。 AWS::DynamoDB::Table 現在 AttributeDefinitions 屬性在更新時將需要替換。 AWS::EC2::Instance 使用 CreditSpecification 屬性指定 T2 執行個體的 CPU 使用量積分選項。 使用 ElasticGpuSpecifications 屬性指定 Elastic GPU，它是可連接至執行個體以加速應用程式圖形效能的 GPU 資源。 AWS::EC2::VPC 現在 InstanceTenancy 屬性從 "dedicated" 更新為 "default" 時，不需要中斷。 AWS::ECS::Service 使用 HealthCheckGracePeriodSeconds 屬性來指定 Amazon ECS 服務排程器在任務第一次啟動之後，應忽略狀態不良之 Elastic Load Balancing 目標運作狀態檢查的時間長度 (秒數)。 AWS::IoT::TopicRule 在 DynamoDBAction 屬性類型中，已不再需要 RangeKeyField 與 RangeKeyValue 屬性。 AWS::KinesisAnalytics::ApplicationOutput 在 ApplicationOutput 屬性類型中，使用 LambdaOutput 屬性來識別 Lambda 函數，作為在設定應用程式輸出時的目的地。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::Kinesis::Stream 使用 <code>StreamEncryption</code> 屬性，AWS KMS key 針對指定的串流使用 來啟用或停用伺服器端加密。 AWS::Lambda::Function 使用 <code>ReservedConcurrentExecutions</code> 屬性指定您希望為此函數保留的最大並行執行數。 AWS::RDS::DBSubnetGroup 使用 <code>DBSubnetGroupName</code> 屬性指定資料庫子網路群組的名稱。 AWS::S3::Bucket 使用 <code>BucketEncryption</code> 屬性，使用伺服器端加密搭配 Amazon S3-managed 金鑰 SSE-S3 或 AWS KMS keys (SSE-KMS) 儲存貯體來指定儲存貯體的預設加密。 在 ReplicationRule 屬性類型中，請使用 <code>SourceSelectionCriteria</code> 屬性指定在識別您要複寫的來源物件時的額外篩選條件。 在 ReplicationDestination 屬性類型中： • 使用 <code>AccessControlTranslation</code> 屬性指定擁有目的地儲存貯體之 AWS 帳戶的複本擁有權。 • 使用 <code>Account</code> 屬性指定目的地儲存貯體擁有者帳戶 ID。 • 使用 <code>EncryptionConfiguration</code> 屬性指定複寫物件目標儲存貯體的加密相關資訊。	

變更	版本日期	Description	API 版本
		AWS::SSM::Association 使用 AssociationName 屬性指定 SSM 文件與 EC2 執行個體 (內含處理文件的組態代理程式) 之間關聯的名稱。	
復原觸發條件新增至 CloudFormation 主控台。	2018 年 1 月 15 日	復原觸發條件可讓 CloudFormation 監控應用程式在堆疊建立和更新期間的狀態；如果應用程式超過您指定之任何警示的閾值，也可轉返該操作。如需詳細資訊，請參閱 監控和復原堆疊操作 。	2010-05-15
更新的資源	2018 年 1 月 12 日	AWS::SSM::Parameter 使用 AllowedPattern 屬性指定用於驗證參數值的規則表達式。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2017 年 12 月 5 日	AWS::Inspector::AssessmentTarget 使用 <code>AWS::Inspector::AssessmentTarget</code> 資源建立 Amazon Inspector 評估目標。 AWS::Inspector::AssessmentTemplate 使用 <code>AWS::Inspector::AssessmentTemplate</code> 資源建立 Amazon Inspector 評估範本。 AWS::Inspector::ResourceGroup 使用 <code>AWS::Inspector::ResourceGroup</code> 資源建立 Amazon Inspector 資源群組，以定義用於識別構成 Amazon Inspector 評估目標之 AWS 資源的標籤。 AWS::ServiceDiscovery::Instance 使用 <code>AWS::ServiceDiscovery::Instance</code> 資源來指定 Amazon Route 53 建立的執行個體的相關資訊。 AWS::ServiceDiscovery::PrivateDnsNamespace 使用 <code>AWS::ServiceDiscovery::PrivateDnsNamespace</code> 資源來指定 Amazon Route 53 的私有命名空間的相關資訊。 AWS::ServiceDiscovery::PublicDnsNamespace 使用 <code>AWS::ServiceDiscovery::PublicDnsNamespace</code> 資源來指定 Amazon Route 53 的公有命名空間的相關資訊。 AWS::ServiceDiscovery::Service 使用 <code>AWS::ServiceDiscovery::Service</code> 資源來定義您希望 Amazon Route 53 在您註冊執行個體時建立的範本 (最多五筆記錄和一個選用的運作狀態檢查)。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 12 月 5 日	AWS::KinesisAnalytics::Application 在 Input 屬性類型中，使用 <code>InputProcessingConfiguration</code> 屬性在從串流接收到記錄時轉換記錄。	2010-05-15
更新的資源	2017 年 12 月 1 日	AWS::CodeBuild::Project 使用 <code>BadgeEnabled</code> 屬性為專案的建置徽章產生可公開存取的 URL。 使用 <code>Cache</code> 屬性為建置相依性設定快取設定。 使用 <code>VpcConfig</code> 屬性以允許 CodeBuild 存取 Amazon VPC 中的資源。 在 EnvironmentVariable 屬性類型中，使用 <code>Type</code> 屬性指定環境變數類型。	2010-05-15
新資源	2017 年 11 月 30 日	AWS::Cloud9::EnvironmentEC2 使用 <code>AWS::Cloud9::EnvironmentEC2</code> 資源在 AWS Cloud9 中建立 Amazon EC2 開發環境。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 11 月 29 日	AWS::ECS::TaskDefinition 使用 Cpu 屬性指定任務所需的 cpu 單元數量。 使用 ExecutionRoleArn 屬性指定執行角色的 ARN。 使用 Memory 屬性指定任務所需的記憶體數量 (以 MiB 為單位)。 使用 RequiresCompatibilities 屬性指定任務所需要的啟動類型。 AWS::ECS::Service 使用 LaunchType 屬性指定用於執行您服務的啟動類型。 使用 NetworkConfiguration 屬性指定服務的網路組態。 使用 PlatformVersion 屬性指定用於執行您服務的平台版本。	2010-05-15
新資源	2017 年 11 月 28 日	AWS::GuardDuty::Detector 使用 AWS::GuardDuty::Detector 資源建立單一 Amazon GuardDuty 偵測器。 AWS::GuardDuty::IPSet 使用 AWS::GuardDuty::IPSet 資源建立 Amazon GuardDuty IP 組。 AWS::GuardDuty::ThreatIntelSet 使用 AWS::GuardDuty::ThreatIntelSet 資源建立 ThreatIntelSet。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 11 月 28 日	AWS::CodeDeploy::Application 使用 ComputePlatform 屬性來指定供 CodeDeploy 部署應用程式的 AWS Lambda 運算平台。 AWS::CodeDeploy::DeploymentGroup 在 DeploymentStyle 屬性類型中，使用 DeploymentType 屬性來指定 Lambda 運算平台上的藍/綠部署。 AWS::EC2::SpotFleet 在 SpotFleetRequestConfigData 屬性類型中，SpotPrice 屬性現在為選用。 AWS::Lambda::Alias 使用 RoutingConfig 屬性指定兩個不同的 AWS Lambda 函數版本，可讓您指定每個版本將叫用多少百分比的流量。	2010-05-15
新的 CodeDeployLambdaAliasUpdate 更新政策	2017 年 11 月 28 日	當 <code>AWS::Lambda::Alias</code> 資源上的版本變更時，使用 CodeDeployLambdaAliasUpdate 更新政策來執行 CodeDeploy 部署。如需詳細資訊，請參閱 UpdatePolicy 屬性。	2010-05-15
新的 SSM 參數類型	2017 年 11 月 21 日	使用 SSM 參數類型以使用 Systems Manager 參數存放區中現有的參數。注意：CloudFormation 目前不支援 SecureString 類型。如需詳細資訊，請參閱 SSM 參數類型。	2010-05-15

變更	版本日期	Description	API 版本
用於 ResolvedValue 資料類型的新 Parameter 欄位	2017 年 11 月 21 日	ResolvedValue 欄位傳回用於堆疊定義中 SSM 參數的值。如需詳細資訊，請參閱《CloudFormation API 參考》中的 Parameter 資料類型。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 11 月 20 日	AWS::ApiGateway::ApiKey 使用 CustomerId 屬性指定 AWS Marketplace 客戶識別碼。 使用 GenerateDistinctId 屬性指定金鑰識別碼是否要與所建立的 API 金鑰值不同。 AWS::ApiGateway::Authorizer 使用 AuthType 屬性指定客戶定義的欄位，此欄位用於 Swagger 匯入和匯出，而不影響功能。 AWS::ApiGateway::DomainName 使用 EndpointConfiguration 屬性來指定 API Gateway 網域名稱的端點類型。 使用 RegionalCertificateArn 屬性來參考區域端點用於網域名稱的憑證。 AWS::ApiGateway::Method 在 Integration 和 IntegrationResponse 屬性類型中，使用 ContentHandling 屬性指定如何處理請求承載內容類型轉換。 AWS::ApiGateway::RestApi 使用 EndpointConfiguration 屬性來指定 API Gateway REST API 的端點類型。 AWS::ApplicationAutoScaling::ScalableTarget 使用 ScheduledActions 屬性來指定 Application Auto Scaling 可擴展目標的排程動作。 AWS::ECR::Repository 使用 LifecyclePolicy 屬性來指定 Amazon ECR 儲存庫的生命週期政策。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::ECS::TaskDefinition 在 ContainerDefinition 屬性類型中，使用 <code>LinuxParameters</code> 屬性來指定 Amazon ECS 容器的 Linux 特定選項。 AWS::ElastiCache::ReplicationGroup 使用 <code>AtRestEncryptionEnabled</code> 屬性啟用靜態加密。 使用 <code>AuthToken</code> 屬性指定用於存取密碼保護伺服器的密碼。 使用 <code>TransitEncryptionEnabled</code> 屬性啟用傳輸中加密。 AWS::ElasticLoadBalancingV2::TargetGroup 使用 <code>TargetGroupName</code> 屬性搭配 <code>Fn::GetAtt</code> 函數以取得 Elastic Load Balancing 目標群組的名稱。 AWS::Elasticsearch::Domain 使用 <code>VPCOptions</code> 屬性指定 OpenSearch Service 網域的 VPC 組態。 AWS::EMR::Cluster 使用 <code>EbsRootVolumeSize</code> 屬性來指定 Amazon EMR 叢集的 EBS 根磁碟區大小。 AWS::RDS::DBInstance 使用 <code>SourceRegion</code> 和 <code>KmsKeyId</code> 屬性從跨區域來源資料庫執行個體建立加密的僅供讀取複本。 AWS::Route53::HostedZone 使用 <code>QueryLoggingConfig</code> 屬性指定 DNS 查詢記錄的組態。	

變更	版本日期	Description	API 版本
自訂資源 NoEcho 物件的新 Response 欄位	2017 年 11 月 20 日	您現在可以使用可選的 NoEcho 欄位，遮蔽自訂資源的輸出。如需詳細資訊，請參閱 自訂資源回應物件 。 noEcho 方法支援的對應 send 參數。如需詳細資訊，請參閱 cfn-response 模組 。	2010-05-15
堆疊集新增堆疊執行個體覆寫。	2017 年 11 月 17 日	CloudFormation StackSets 現在可讓您在堆疊執行個體中，依據帳戶與區域覆寫參數值。當您建立堆疊執行個體或更新現有堆疊執行個體時，可以覆寫參數值。如需詳細資訊，請參閱 覆寫堆疊執行個體上的參數 。	2010-05-15
更新的資源	2017 年 11 月 15 日	AWS::StepFunctions::StateMachine 您可以在建立狀態機器時使用 AWS::Step Functions::StateMachine 指定 StateMachineName，並可更新 DefinitionString 和 RoleArn，無需取代狀態機器。	2010-05-15
StackSets 現在支援每個堆疊集最多 500 個堆疊執行個體。	2017 年 11 月 6 日	現在您可以在每個堆疊集建立最多 500 個堆疊執行個體。如需 CloudFormation 限制的詳細資訊，請參閱 了解 CloudFormation 配額 。	2010-05-15
新資源	2017 年 11 月 2 日	AWS::CloudFront::CloudFrontOriginAccessIdentity 使用 AWS::CloudFront::CloudFrontOriginAccessIdentity 資源來指定 Amazon CloudFront 原始存取身分，以便與 CloudFront 分佈的來源建立關聯。 AWS::CloudFront::StreamingDistribution 使用 AWS::CloudFront::StreamingDistribution 資源來指定 CloudFront 的 Adobe 即時簡訊通訊協定 (RTMP) 串流分發。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 11 月 2 日	AWS::ApiGateway::Deployment StageName 屬性已在 StageDescription 屬性類型中棄用。 AWS::ApiGateway::Method 使用 OperationName 屬性來指派 API Gateway 方法的易記名稱。 使用 RequestValidatorId 屬性以關聯請求驗證程式與方法。 AWS::AutoScaling::AutoScalingGroup 使用 LifecycleHookSpecificationList 屬性來指定當 Auto Scaling 啟動或終止執行個體時要執行的動作。 AWS::CloudFront::Distribution 使用 Tags 屬性來指定要與 CloudFront 分佈建立關聯的一組任意標籤 (鍵值組)。 在 CacheBehavior 和 DefaultCacheBehavior 屬性類型中，使用 LambdaFunctionAssociations 屬性來指定 CloudFront 分佈的 Lambda 函式關聯。 在 CustomOriginConfig 屬性類型中，使用 OriginKeepaliveTimeout 屬性指定自訂持續作用逾時，然後使用 OriginReadTimeout 屬性指定自訂原始伺服器讀取逾時。 在 DistributionConfig 屬性類型中，使用 IPV6Enabled 屬性指定 CloudFront 是否以分佈的 IPv6 地址來回應 IPv6 DNS 請求。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::CodeDeploy::DeploymentGroup 在 LoadBalancerInfo 屬性類型中，使用 <code>TargetGroupInfoList</code> 屬性來指定 Elastic Load Balancing 中用於部署的目標群組的相關資訊。 AWS::EC2::SecurityGroup、AWS::EC2::SecurityGroupEgress 與 AWS::EC2::SecurityGroupIngress 使用 <code>Description</code> 屬性指定安全群組規則的描述。 AWS::EC2::Subnet <code>Ipv6CidrBlock</code> 屬性現在支援 <code>No interruption</code> 更新。 AWS::EC2::VPNGateway 使用 <code>AmazonSideAsn</code> 屬性指定 BGP 工作階段 Amazon 端的私有自發系統編號 (ASN)。 AWS::EC2::VPNConnection 使用 <code>VpnTunnelOptionsSpecifications</code> 屬性來設定 VPN 連線的通道選項。 AWS::ElasticBeanstalk::ConfigurationTemplate 和 AWS::ElasticBeanstalk::Environment 在 ConfigurationOptionSetting 和 OptionSetting 屬性類型中，使用 <code>ResourceName</code> 屬性指定以時間為基礎之擴展組態選項的資源名稱。 AWS::EMR::Cluster 使用 <code>CustomAmiId</code> 屬性指定叢集的自訂 Amazon Linux AMI。 AWS::KinesisFirehose::DeliveryStream 使用 <code>Arn</code> 屬性與 <code>Fn::GetAtt</code> 函數以取得交付串流的 Amazon Resource Name (ARN)。	

變更	版本日期	Description	API 版本
		AWS::KMS::Key 使用 Tags 屬性來指定要與客戶受管金鑰建立關聯的一組任意標籤 (鍵/值對)。 AWS::OpsWorks::Layer 和 AWS::OpsWorks::Stack 使用 Tags 屬性指定要與 OpsWorks layer 或堆疊建立關聯的任意標籤集 (索引鍵/值對)。 AWS::RDS::OptionGroup 在 OptionConfiguration 屬性類型中，使用 OptionVersion 屬性指定此選項的版本。 AWS::S3::Bucket 使用 AnalyticsConfigurations 屬性來設定 Amazon S3 儲存貯體的分析篩選條件。	

變更	版本日期	Description	API 版本
新資源	2017 年 10 月 24 日	AWS::Glue::Classifier 使用 AWS::Glue::Classifier 資源來建立 AWS Glue 分類器。 AWS::Glue::Connection 使用 AWS::Glue::Connection 資源來指定與資料來源的 AWS Glue 連線。 AWS::Glue::Crawler 使用 AWS::Glue::Crawler 資源來指定 & AWS Glue 爬蟲程式。 AWS::Glue::Database 使用 AWS::Glue::Database 資源來建立 AWS Glue 資料庫。 AWS::Glue::DevEndpoint 使用 AWS::Glue::DevEndpoint 資源指定開發端點以遠端偵錯 ETL 指令碼。 AWS::Glue::Job 使用 AWS::Glue::Job 資源來指定資料目錄中的 AWS Glue 任務。 AWS::Glue::Partition 使用 AWS::Glue::Partition 資源來建立 AWS Glue 分割區，以代表資料表的資料配量。 AWS::Glue::Table 使用 AWS::Glue::Table 資源來建立 AWS Glue 資料表。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::Glue::Trigger 使用 AWS::Glue::Trigger 資源指定執行 AWS Glue 任務的觸發條件。	
新資源	2017 年 10 月 11 日	AWS::SSM::MaintenanceWindow 使用 AWS::SSM::MaintenanceWindow 資源建立 AWS Systems Manager 維護時段。 AWS::SSM::MaintenanceWindowTarget 使用 AWS::SSM::MaintenanceWindowTarget 資源註冊目標的維護時段。 AWS::SSM::MaintenanceWindowTask 使用 AWS::SSM::MaintenanceWindowTask 資源定義維護時段任務。 AWS::SSM::PatchBaseline 使用 AWS::SSM::PatchBaseline 資源來定義 Systems Manager 修補程式基準。	2010-05-15
新資源	2017 年 10 月 10 日	AWS::ElasticLoadBalancingV2::ListenerCertificate 使用 AWS::ElasticLoadBalancingV2::ListenerCertificate 資源來指定 Elastic Load Balancing 接聽程式的憑證。	2010-05-15
新資源	2017 年 9 月 27 日	AWS::Athena::NamedQuery 使用 AWS::Athena::NamedQuery 資源來建立 Amazon Athena 查詢。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 9 月 27 日	AWS::EC2::NatGateway 使用 Tags 屬性指定 NAT 閘道的資源標籤。 AWS::ElasticBeanstalk::Application 使用 ResourceLifecycleConfig 屬性為屬於應用程式的資源，定義生命週期設定，以及由 Elastic Beanstalk 擔任以套用生命週期設定的服務角色。 AWS::ElasticBeanstalk::ConfigurationTemplate 和 AWS::ElasticBeanstalk::Environment 使用 PlatformArn 屬性指定 Elastic Beanstalk 的自訂平台。 AWS::ElasticLoadBalancingV2::TargetGroup 在 TargetDescription 屬性類型中，使用 AvailabilityZone 屬性指定要註冊 IP 地址的可用區域。 AWS::Events::Rule 在 Target 屬性類型中，使用下列屬性以執行事件的輸入轉換，以及設定 Amazon ECS 任務和 Kinesis 串流目標。 • EcsParameters • InputTransformer • KinesisParameters • RunCommandParameters AWS::KinesisFirehose::DeliveryStream 使用 DeliveryStreamType 屬性指定串流類型，使用 KinesisStreamSourceConfiguration 屬性指定串流以及 Kinesis 串流用作交付串流來源的角色 ARN。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::RDS::DBInstance 對於 Engine 屬性，如果您指定 <code>oracle-se</code> 或 <code>oracle-se1</code>，您可以更新 <code>oracle-se2</code> 而無需替換資料庫執行個體。 AWS::S3::Bucket 使用 <code>AccelerateConfiguration</code> 屬性來設定 Amazon S3 儲存貯體的傳輸加速狀態。	
新增至堆疊的終止保護。	2017 年 9 月 26 日	在堆疊上啟用終止保護可防止意外刪除。使用者無法刪除已啟用終止保護的堆疊。如需詳細資訊，請參閱 防止堆疊遭刪除 。	2010-05-15
從版本 1.4-22 開始變更預設 <code>umask</code> 值	2017 年 9 月 14 日	<code>cfn-hup.conf</code> 組態檔案的預設 <code>umask</code> 參數值現在為 <code>022</code> 。如需詳細資訊，請參閱 cfn-hup 。	
更新的資源	2017 年 9 月 7 日	AWS::ElasticLoadBalancingV2::LoadBalancer 使用 <code>SubnetMappings</code> 屬性指定要連接至負載平衡器之子網路的 ID。 使用 <code>Type</code> 屬性指定要建立的負載平衡器類型。 AWS::ElasticLoadBalancingV2::TargetGroup 使用 <code>TargetType</code> 屬性指定此目標群組中的目標的註冊類型。	2010-05-15
新增至 CloudFormation API 的復原觸發條件	2017 年 8 月 31 日	復原觸發條件可讓 CloudFormation 監控應用程式在堆疊建立和更新期間的狀態；如果應用程式超過您指定之任何警示的閾值，也可轉返該操作。如需詳細資訊，請參閱 CloudFormation API 參考中的 RollbackConfiguration 。	2010-05-15

變更	版本日期	Description	API 版本
cfn-hup.conf 檔案的新 umask 參數	2017 年 8 月 31 日	使用 cfn-hup.conf 組態檔案中的 umask 參數，以控制 cfn-hup 協助程式 (版本 1.4-21) 使用的檔案許可。如需詳細資訊，請參閱 cfn-hup 。	
適用於 VPC Sizing 支援的更新資源	2017 年 8 月 29 日	AWS::EC2::VPCCidrBlock 使用 CidrBlock 屬性建立 IPv4 CIDR 區塊與 VPC 的關聯。 AWS::EC2::VPC 使用 CidrBlockAssociations 屬性與 Fn::GetAtt 函數，以取得與 VPC 關聯之 IPv4 CIDR 區塊關聯 ID 的清單。	2010-05-15
更新的資源	2017 年 8 月 23 日	AWS::S3::Bucket 在 Rule 屬性類型中，使用 TagFilters 屬性來指定標籤，以用於識別 Amazon S3 儲存貯體的物件子集。 使用 MetricsConfiguration 屬性為 Amazon S3 儲存貯體中的 CloudWatch 請求指標，指定其指標組態。 AWS::IoT::TopicRule 在 Action 屬性類型中，使用 DynamoDBv2Action 屬性來描述將資料寫入 DynamoDB 資料表的 AWS IoT 動作。 在 Action 屬性類型中，DynamoDBAction 屬性現在支援 HashKeyType 和 RangeKeyType 屬性。 AWS::Lambda::Permission 使用 EventSourceToken 屬性指定唯一的符記，此符記必須由主體叫用函數來提供。	2010-05-15

變更	版本日期	Description	API 版本
新的虛擬參數	2017 年 8 月 23 日	使用 <code>AWS::Partition</code> 虛擬參數傳回資源所在的分割區。 使用 <code>AWS::URLSuffix</code> 虛擬參數傳回網域的字尾。 如需詳細資訊，請參閱虛擬參數參考。	2010-05-15
適用於 DAX 支援的新資源	2017 年 8 月 22 日	AWS::DAX::Cluster 使用 <code>AWS::DAX::Cluster</code> 資源來建立 DAX 叢集以搭配 Amazon DynamoDB 使用。 AWS::DAX::ParameterGroup 使用 <code>AWS::DAX::ParameterGroup</code> 資源來建立參數群組以搭配 Amazon DynamoDB 使用。 AWS::DAX::SubnetGroup 使用 <code>AWS::DAX::SubnetGroup</code> 資源建立子網路群組以搭配 DAX (DynamoDB Accelerator) 使用。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2017 年 8 月 18 日	AWS::ApiGateway::DocumentationPart 和 AWS::ApiGateway::DocumentationPart 使用 <code>AWS::ApiGateway::DocumentationPart</code> 和 <code>AWS::ApiGateway::DocumentationVersion</code> 資源來建立 API Gateway API 的文件。 AWS::ApiGateway::GatewayResponse 使用 <code>AWS::ApiGateway::GatewayResponse</code> 資源來建立 API Gateway API 的自訂回應。 AWS::ApiGateway::RequestValidator 使用 <code>AWS::ApiGateway::RequestValidator</code> 資源為傳入 API Gateway API 的請求設定驗證規則。 AWS::EC2::NetworkInterfacePermission 使用 <code>AWS::EC2::NetworkInterfacePermission</code> 資源將 AWS 帳戶許可授予網路界面。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 8 月 18 日	AWS::ApiGateway::Stage 使用 DocumentationVersion 屬性指定 API 文件的版本控制快照。 AWS::AutoScaling::ScalingPolicy 使用 TargetTrackingConfiguration 屬性來指定 Auto Scaling 目標追蹤擴展政策組態。 AWS::CloudTrail::Trail 使用 EventSelectors 屬性以提供 Amazon S3 資料事件支援。 AWS::CodeDeploy::DeploymentGroup 使用 LoadBalancerInfo 和 DeploymentStyle 屬性來指定適用於就地部署的 Elastic Load Balancing 負載平衡器。 使用 AutoRollbackConfiguration 屬性設定部署的自動復原。 AWS::EC2::SpotFleet 在 SpotFleetRequestConfigData 屬性類型中，使用 ReplaceUnhealthyInstances 屬性以指示 Spot 機群是否應取代運作狀態不佳的執行個體，以及使用 Type 屬性以指定請求的類型。 AWS::EC2::Subnet 使用 AssignIpv6AddressOnCreation 和 Ipv6CidrBlock 屬性以 IPv6 CIDR 區塊建立子網路。 AWS::KinesisFirehose::DeliveryStream 使用 ExtendedS3DestinationConfiguration 屬性在 Amazon S3 中設定目的地。	2010-05-15

變更	版本日期	Description	API 版本
		使用 <code>ProcessingConfiguration</code> 各目的地組態中的子屬性，以叫用 Lambda 函數，轉換傳入來源資料並將轉換後的資料傳遞至目的地。 AWS::RDS::DBCluster 和 AWS::RDS::DBInstance 對於不指定 <code>DBClusterIdentifier</code> 屬性的 <code>AWS::RDS::DBCluster</code> 資源和 <code>AWS::RDS::DBInstance</code> 資源而言，預設的 <code>DeletionPolicy</code> 現在為 <code>Snapshot</code>。如需詳細資訊，請參閱 DeletionPolicy 屬性。 AWS::S3::Bucket 在 Rule 屬性類型中，使用 <code>AbortIncompleteMultipartUpload</code> 屬性來指定生命週期規則，以便在分段上傳至 Amazon S3 儲存貯體不完整時中止上傳。 AWS::SQS::Queue 使用 <code>KmsMasterKeyId</code> 和 <code>KmsDataKeyReusePeriodSeconds</code> 屬性來設定 Amazon SQS 的伺服器端加密。 新增 <code>Arn</code> 屬性至 <code>Fn::GetAtt</code> 內部函數以用於下列資源： • AWS::CloudTrail::Trail。另外新增 <code>SnsTopicArn</code>。 • AWS::CloudWatch::Alarm • AWS::DynamoDB::Table • AWS::ECS::Cluster • AWS::IoT::Policy • AWS::IoT::TopicRule • AWS::Logs::Destination	

變更	版本日期	Description	API 版本
支援 CodePipeline 成品中的堆疊標籤	2017 年 8 月 18 日	您現在可以在範本組態檔案中指定堆疊標籤，用作 CodePipeline 管道的成品。指定的標籤會套用至使用範本組態檔案建立的堆疊。如需詳細資訊，請參閱 CloudFormation 成品 。	2010-05-15
建立加密的檔案系統	2017 年 8 月 14 日	AWS::EFS::FileSystem 使用 Encrypted 屬性在建立 Amazon EFS 檔案系統的過程中加密。 使用 KmsKeyId 屬性以選擇性指定自訂客戶受管金鑰，以用於保護加密的檔案系統。	2010-05-15
AWS Batch 支援的新資源	2017 年 8 月 8 日	AWS::Batch::ComputeEnvironment 使用 AWS::Batch::ComputeEnvironment 資源來定義您的 AWS Batch 運算環境。 AWS::Batch::JobDefinition 使用 AWS::Batch::JobDefinition 資源來指定 AWS Batch 任務定義的參數。 AWS::Batch::JobQueue 使用 AWS::Batch::JobQueue 資源來定義您的 AWS Batch 任務佇列。	2010-05-15

變更	版本日期	Description	API 版本
適用於 Amazon Managed Service for Apache Flink 支援的新資源	2017 年 7 月 28 日	AWS::KinesisAnalytics::Application 使用 <code>AWS::KinesisAnalytics::Application</code> 資源來建立 Amazon Managed Service for Apache Flink 應用程式。 AWS::KinesisAnalytics::ApplicationOutput 使用 <code>AWS::KinesisAnalytics::ApplicationOutput</code> 資源來新增外部目的地到您的 Amazon Managed Service for Apache Flink 應用程式。 AWS::KinesisAnalytics::ApplicationReferenceDataSource 使用 <code>AWS::KinesisAnalytics::ApplicationReferenceDataSource</code> 資源來新增參考資料來源到現有的 Amazon Managed Service for Apache Flink 應用程式。	2010-05-15
使用 StackSets 集中管理跨帳戶與區域的堆疊	2017 年 7 月 25 日	StackSets 可讓您在單一操作中建立、更新或刪除跨多個帳戶與區域的堆疊。您使用管理員帳戶定義與管理 CloudFormation 範本，並使用該範本做為基礎，將堆疊佈建至指定區域中選定的目標帳戶。如需詳細資訊，請參閱 使用 StackSets 管理跨帳戶和跨區域的堆疊 。	2010-05-15
透過用戶端請求字符檢視堆疊事件	2017 年 7 月 14 日	在主控台中，堆疊操作會在 Events (事件) 索引標籤中顯示用戶端請求字符。特定堆疊操作觸發的所有事件，都會被指派相同的用戶端請求字符，您可以用它追蹤操作。如需詳細資訊，請參閱 AWS CloudFormation API 參考中的 檢視 AWS 管理主控台上的 CloudFormation 堆疊資料與資源 和 StackEvent 。	2010-05-15
使用堆疊快速建立連結	2017 年 7 月 14 日	使用快速建立連結以快速啟動並執行堆疊。您可以指定範本 URL、堆疊名稱和範本參數，以填入單一 Create Stack Wizard (建立堆疊精靈) 頁面。如需詳細資訊，請參閱 建立堆疊的快速建立連結 。	2010-05-15

變更	版本日期	Description	API 版本
AWS Database Migration Service 支援的新資源	2017 年 7 月 12 日	AWS::DMS::Certificate 使用 <code>AWS::DMS::Certificate</code> 資源建立 SSL 憑證，以加密 AWS DMS 端點與複寫執行個體之間的連線。 AWS::DMS::Endpoint 使用 <code>AWS::DMS::Endpoint</code> 資源來建立 AWS DMS 端點。 AWS::DMS::EventSubscription 使用 <code>AWS::DMS::EventSubscription</code> 資源透過 Amazon Simple Notification Service 取得 AWS DMS 事件的通知。 AWS::DMS::ReplicationInstance 使用 <code>AWS::DMS::ReplicationInstance</code> 資源來建立 AWS DMS 複寫執行個體。 AWS::DMS::ReplicationSubnetGroup 使用 <code>AWS::DMS::ReplicationSubnetGroup</code> 資源建立 AWS DMS 複寫子網路群組。 AWS::DMS::ReplicationTask 使用 <code>AWS::DMS::ReplicationTask</code> 資源建立 AWS DMS 複寫任務。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2017 年 7 月 5 日	AWS::CloudWatch::Dashboard 使用 <code>AWS::CloudWatch::Dashboard</code> 資源來指定 CloudWatch 主控台的自訂 CloudWatch 儀表板。 AWS::ApiGateway::DomainName 使用 <code>AWS::ApiGateway::DomainName</code> 資源為部署至 Amazon API Gateway 的 API 指定自訂、易記的 URL。 AWS::EC2::EgressOnlyInternetGateway 使用 <code>AWS::EC2::EgressOnlyInternetGateway</code> 資源，為您的 VPC 建立僅限輸出的網際網路閘道。 InstanceFleetConfig 使用 <code>InstanceFleetConfig</code> 資源來設定 Amazon EMR 叢集的 Spot 執行個體機群。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 7 月 5 日	AWS::ApiGateway::RestApi 使用 BinaryMediaTypes 屬性指定支援的二進位媒體類型。 AWS::ApplicationAutoScaling::ScalingPolicy 使用 TargetTrackingScalingPolicy Configuration 屬性指定目標追蹤擴展政策組態。 AWS::CloudTrail::Trail 使用 TrailName 屬性指定 AWS CloudTrail 資源的自訂名稱。 使用 Tags 屬性指定資源標籤。 AWS::CodeDeploy::DeploymentGroup 使用 AlarmConfiguration 屬性設定部署群組的警示。 使用 TriggerConfigurations 屬性設定部署群組的通知觸發。 AWS::EMR::Cluster 使用 JobFlowInstancesConfig 屬性類型中的 CoreInstanceFleet 屬性和 MasterInstanceFleet 屬性，來設定 Amazon EMR 叢集的 Spot 執行個體機群。 AWS::DynamoDB::Table 使用 TimeToLiveSpecification 屬性來指定 Amazon DynamoDB 資料表的存留時間 (TTL) 設定。 使用 Tags 屬性來指定 DynamoDB 資料表的資源標籤。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::EC2::Instance IamInstanceProfile 屬性現在支援 No interruption 更新。 AWS::EC2::Route 使用 EgressOnlyInternetGatewayId 屬性，為 EC2 路由指定僅限輸出的網際網路閘道。 AWS::Kinesis::Stream 使用 RetentionPeriodHours 屬性指定存放在碎片中的資料記錄維持可存取狀態的小時數。 AWS::RDS::DBCluster 使用 ReplicationSourceIdentifier 屬性建立資料庫叢集，作為其他資料庫叢集或 Amazon RDS MySQL 資料庫執行個體的僅供讀取複本。 AWS::Redshift::Cluster 使用 LoggingProperties 屬性來建立稽核日誌檔，並存放在 Amazon S3。	
新資源	2017 年 6 月 6 日	AWS::EMR::SecurityConfiguration 使用 AWS::EMR::SecurityConfiguration 資源建立安全組態，它存放在服務中，並可在建立叢集時指定。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 6 月 6 日	AWS::AutoScaling::LifecycleHook NotificationTargetARN 和 RoleARN 屬性現在是選用的。 AWS::CloudWatch::Alarm 您現在可以在建立 EvaluateLowSampleCountPercentile 資源時，使用 ExtendedStatistic、TreatMissingData 和 AWS::CloudWatch::Alarm 屬性。 AWS::EC2::SpotFleet CloudFormation 支援對 Spot 機群屬性進行可變的變更。 SpotFleetRequestConfigData 屬性的以下屬性支援 Replacement 更新： • AllocationStrategy • IamFleetRole • LaunchSpecifications • SpotPrice • TerminateInstancesWithExpiration • ValidFrom • ValidUntil SpotFleetRequestConfigData 屬性的以下屬性支援 No interruption 更新： • ExcessCapacityTerminationPolicy • TargetCapacity AWS::EMR::InstanceGroupConfig 針對 Amazon EMR 任務執行個體群組，CloudFormation 現在支援 Auto Scaling。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::Events::Rule RoleArn 屬性在 Rule 資源上已經棄用。 使用 Target 屬性類型上的 RoleArn 屬性，以指定要用於目標的 IAM 角色。 AWS::Kinesis::Stream ShardCount 屬性現在支援 No interruption 更新。 AWS::Lambda::Function 使用 TracingConfig 屬性來設定 Lambda 函數的追蹤設定。 AWS::Redshift::Cluster, AWS::Redshift::ClusterParameterGroup, AWS::Redshift::ClusterSecurityGroup, 和 AWS::Redshift::ClusterSubnetGroup 使用 Tags 屬性指定資源標籤。 AWS::RDS::DBCluster 新增 ReadEndpoint.Address 屬性至 Fn::GetAtt 內部函數。 AWS::S3::Bucket 新增 Arn 屬性至 Fn::GetAtt 內部函數。	

變更	版本日期	Description	API 版本
新資源	2017 年 5 月 11 日	下列新資源支援 AWS WAF 搭配 Elastic Load Balancing (ELB) Application Load Balancer 使用。 AWS::WAFRegional::ByteMatchSet 使用 <code>AWS::WAFRegional::ByteMatchSet</code> 資源識別您要檢查的 Web 請求的一部分。 AWS::WAFRegional::IPSet 使用 <code>AWS::WAFRegional::IPSet</code> 資源，依據發出請求的 IP 地址，指定要允許或封鎖的 Web 請求。 AWS::WAFRegional::Rule 使用 <code>AWS::WAFRegional::Rule</code> 資源指定 <code>IPSet</code>、<code>ByteMatchSet</code> 及 <code>SqlInjectionMatchSet</code> 物件的組合，以識別要允許、封鎖或計數的 Web 請求。 AWS::WAFRegional::SizeConstraintSet 使用 <code>AWS::WAFRegional::SizeConstraintSet</code> 資源指定用於檢查 Web 請求大小的大小限制，以及要檢查請求的哪些部分。 AWS::WAFRegional::SqlInjectionMatchSet 使用 <code>AWS::WAFRegional::SqlInjectionMatchSet</code> 資源以允許、封鎖或計數在 Web 請求的特定部分內含惡意 SQL 程式碼的請求。 AWS::WAFRegional::WebACL 使用 <code>AWS::WAFRegional::WebACL</code> 資源識別您要允許、封鎖或計數的 Web 請求。	2010-05-15

變更	版本日期	Description	API 版本
		<u>AWS::WAFRegional::WebACLAssociation</u> 使用 <code>AWS::WAFRegional::WebACLAssociation</code> 資源建立 Web 存取控制群組 (ACL) 與資源的關聯。 <u>AWS::WAFRegional::XssMatchSet</u> 使用 <code>AWS::WAFRegional::XssMatchSet</code> 資源指定您希望 AWS WAF 檢查跨網站指令碼攻擊的 Web 請求的部分，以及要檢查的標頭名稱。	

變更	版本日期	Description	API 版本
新資源	2017 年 4 月 28 日	AWS::Cognito::IdentityPool 使用 AWS::Cognito::IdentityPool 資源來建立 Amazon Cognito 身分集區。 AWS::Cognito::IdentityPoolRoleAttachment 使用 AWS::Cognito::IdentityPoolRoleAttachment 資源來管理 Amazon Cognito 身分集區的角色組態。 AWS::Cognito::UserPool 使用 AWS::Cognito::UserPool 資源來建立 Amazon Cognito 使用者集區。 AWS::Cognito::UserPoolClient 使用 AWS::Cognito::UserPoolClient 資源建立使用者集區用戶端。 AWS::Cognito::UserPoolGroup 使用 AWS::Cognito::UserPoolGroup 資源在 Amazon Cognito 使用者集區中建立使用者群組。 AWS::Cognito::UserPoolUser 使用 AWS::Cognito::UserPoolUser 資源來建立 Amazon Cognito 使用者集區使用者。 AWS::Cognito::UserPoolUserToGroupAttachment 使用 AWS::Cognito::UserPoolUserToGroupAttachment 資源將使用者連接至 Amazon Cognito 使用者集區群組。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 4 月 28 日	SourceDetails 使用 <code>MaximumExecutionFrequency</code> 資源的 <code>AWS::Config::ConfigRule</code> 子資源，以定期觸發執行自訂規則的評估。 AWS::EC2::Volume 我們現在支援 CloudFormation 中的 Elastic Volumes for Amazon Elastic Block Store (Amazon EBS)。我們現在支援三個屬性的 <code>No interruption</code> 更新：<code>VolumeType</code>、<code>Size</code> 和 <code>Iops</code>。 AWS::EC2::SecurityGroup 使用 <code>GroupName</code> 屬性來指定 Amazon EC2 安全群組的名稱。 AWS::ECS::Service <code>AWS::ECS::Service</code> 有三種新的屬性：<code>PlacementConstraints</code>、<code>PlacementStrategies</code> 和 <code>ServiceName</code>。 AWS::ECS::TaskDefinition 使用 <code>PlacementConstraints</code> 屬性定義服務中的任務置放限制。 AWS::ElastiCache::ReplicationGroup 已將 <code>ConfigurationEndPoint.Address</code> 屬性與 <code>ConfigurationEndPoint.Port</code> 屬性新增至 <code>Fn::GetAtt</code> 內部函數。 AWS::ElasticLoadBalancingV2::LoadBalancer 使用 <code>IpAddressType</code> 屬性指定負載平衡器之子網路使用的 IP 地址類型。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::EMR::Cluster 針對 Amazon EMR 叢集，CloudFormation 現在支援 Auto Scaling。 AWS::IAM::ManagedPolicy 使用 ManagedPolicyName 屬性來指定 IAM 管理政策的自訂名稱。 AWS::Lambda::Function 使用 Tags 屬性將標籤新增至 Lambda 函數。 AWS::OpsWorks::Instance 將以下屬性新增至 Fn::GetAtt 內部函數：AvailabilityZone、PrivateDnsName、PrivateIp 及 PublicDnsName。 AWS::OpsWorks::UserProfile 使用 SshUsername 屬性指定使用者的 SSH 名稱。 新增 SshUsername 屬性至 Fn::GetAtt 內部函數。 AWS::Redshift::Cluster 使用 IamRoles 屬性提供 Amazon Redshift 叢集可用來存取其他服務的一或多個 AWS Identity and Access Management 角色清單 AWS。	
使用 CloudFormation 設計工具編輯 YAML 和 JSON 中的範本	2017 年 4 月 6 日	當您使用設計工具建立 CloudFormation 範本時，現在可以在整合式編輯器中以 YAML 和 JSON 來編輯範本。您也可以將 JSON 範本轉換為 YAML，反之亦然，這取決於您慣用的範本編寫語言。如需詳細資訊，請參閱 CloudFormation Designer 是什麼？。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2017 年 4 月 6 日	AWS::SSM::Parameter 使用 <code>AWS::SSM::Parameter</code> 資源在參數存放區中建立 SSM 參數。	2010-05-15
AWS::Include 轉換程序	2017 年 3 月 28 日	使用 <code>AWS::Include</code> 轉換來參考存放於 Amazon S3 儲存貯體中可重複使用的程式碼片段。如需詳細資訊，請參閱 AWS::Include 轉換 。	2010-05-15
對等您的 Amazon VPC 與另一個帳戶	2017 年 3 月 28 日	您現在可以使用 CloudFormation 將您的 Amazon VPC 與另一個 AWS 帳戶中的 VPC 對等。如需詳細資訊，請參閱 其他 AWS 帳戶中具有 Amazon VPC 的對等 。	2010-05-15
新資源	2017 年 3 月 28 日	AWS::ApiGateway::UsagePlanKey 使用 <code>AWS::ApiGateway::UsagePlanKey</code> 資源關聯用量計劃金鑰，並判斷用量計劃套用到哪些使用者。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 3 月 28 日	AWS::EC2::VPCPeeringConnection 使用 PeerOwnerId 屬性和 PeerRoleArn 屬性，以便與另一個 AWS 帳戶中的 VPC 對等。 如需詳細資訊，請參閱其他 AWS 帳戶中具有 Amazon VPC 的對等。 AWS::IAM::InstanceProfile 使用 InstanceProfileName 屬性設定執行個體設定檔。 AWS::Lambda::Function 使用 DeadLetterConfig 屬性來設定 AWS Lambda 如何處置無法處理的事件。 Node.js v0.10 不再支援 Runtime 屬性。 AWS::Route53::HealthCheck HealthCheckConfig HealthCheckConfig 屬性現在有七種新的資源子屬性類型：AlarmIdentifier、ChildHealthChecks、EnableSNITimeout、HealthThreshold、InsufficientDataHealthStatus、Inverted 和 MeasureLatency。 AWS::SQS::Queue 使用 ContentBasedDeduplication 和 FifoQueue 屬性來建立先進先出 (FIFO) Amazon Simple Queue Service 佇列。 AWS::S3::Bucket 您現在可以為 Amazon S3 儲存貯體指定 IPv6 網域名稱。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2017 年 2 月 10 日	AWS::StepFunctions::Activity 使用 <code>AWS::StepFunctions::Activity</code> 資源來建立 AWS Step Functions 活動。 AWS::StepFunctions::StateMachine 使用 <code>AWS::StepFunctions::StateMachine</code> 資源來建立 Step Functions 狀態機器。	2010-05-15
新的內部函數	2017 年 1 月 17 日	使用 <code>Fn::Split</code> 函數將字串分割為字串值的清單。如需詳細資訊，請參閱 Fn::Split 。	2010-05-15
主控台支援匯入清單	2017 年 1 月 17 日	使用 CloudFormation 主控台查看所有匯入已匯出之輸出值的堆疊。如需詳細資訊，請參閱 列出匯入匯出的輸出值的堆疊 。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2017 年 1 月 17 日	AWS::AutoScaling::AutoScalingGroup 無需替換 Auto Scaling 群組即可更新 LoadBalancerNames 屬性。 AWS::ECS::TaskDefinition 新增 NetworkMode 和 MemoryReservation 屬性。 AWS::RDS::DBCluster CloudFormation 支援更新 Tags 屬性。 AWS::RDS::DBInstance 新增 Timezone 屬性。 FirehoseAction 新增 Separator 屬性。 AWS::OpsWorks::Instance 新增 PublicIp 內部函數的 Fn::GetAtt 屬性。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2016 年 01 月 12 日	AWS::CodeBuild::Project 使用 <code>AWS::CodeBuild::Project</code> 資源建立 AWS CodeBuild 專案，定義 CodeBuild 如何建置您的原始程式碼。 AWS::SSM::Association 使用 <code>AWS::SSM::Association</code> 資源將 Amazon EC2 Systems Manager 文件與 EC2 執行個體建立關聯。 AWS::EC2::SubnetCidrBlock 使用 <code>AWS::EC2::SubnetCidrBlock</code> 資源建立單一 IPv6 CIDR 區塊與 Amazon VPC 子網路的關聯。 AWS::EC2::VPCCidrBlock 使用 <code>AWS::EC2::VPCCidrBlock</code> 資源建立單一 Amazon 提供的 IPv6 CIDR 區塊與 Amazon VPC 的關聯。	2010-05-15

變更	版本日期	Description	API 版本
適用於 IPv6 支援的更新資源	2016 年 01 月 12 日	AWS::EC2::Instance 新增 Ipv6AddressCount 和 Ipv6Addresses 屬性。 AWS::EC2::NetworkAclEntry 新增 Ipv6CidrBlock 屬性。 AWS::EC2::NetworkInterface 新增 Ipv6AddressCount 和 Ipv6Addresses 屬性。 AWS::EC2::Route 新增 DestinationIpv6CidrBlock 屬性。 AWS::EC2::SecurityGroupEgress 新增 CidrIpv6 屬性。 AWS::EC2::SecurityGroupIngress 新增 CidrIpv6 屬性。 AWS::EC2::SpotFleet 新增網路介面啟動規格的 Ipv6AddressCount 和 Ipv6Addresses 屬性。 AWS::EC2::Subnet 新增 Ipv6CidrBlocks 函數的 Fn::GetAtt 屬性。 AWS::EC2::VPC 新增 Ipv6CidrBlocks 函數的 Fn::GetAtt 屬性。 AWS::SSM::Document 新增 DocumentType 屬性。	2010-05-15

變更	版本日期	Description	API 版本
資源規格	2016 年 11 月 22 日	使用 CloudFormation 資源規格來建置可協助您建立 CloudFormation 範本的工具。規格是一種機器可讀取、JSON 格式的文字檔案。如需詳細資訊，請參閱 CloudFormation 資源規格 。	2010-05-15
新資源	2016 年 11 月 22 日	AWS::OpsWorks::UserProfile 使用 AWS::OpsWorks::UserProfile 資源為需要存取 OpsWorks 堆疊中執行個體的使用者設定 SSH 存取。 AWS::OpsWorks::Volume 使用 AWS::OpsWorks::Volume 資源向 OpsWorks 堆疊註冊 Amazon Elastic Block Store 磁碟區。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 11 月 22 日	AWS::OpsWorks::App 新增 DataSources 屬性。 AWS::OpsWorks::Instance 新增 BlockDeviceMappings 、 AgentVersion 、 ElasticIps 、 Hostname、 Tenancy 和 Volumes 屬性。 AWS::OpsWorks::Layer 新增 CustomJson 和 VolumeConfigurations 屬性。 AWS::OpsWorks::Stack 新增 ElasticIps 、 EcsClusterArn 、 RdsDbInstances 、 CloneAppIds 、 ClonePermissions 和 SourceStackId 屬性。 AWS::RDS::DBInstance 新增 CopyTagsToSnapshot 屬性。	2010-05-15
匯入清單	2016 年 11 月 22 日	匯入已匯出之輸出值的清單，追蹤哪些 CloudFormation 堆疊已匯入值。如需詳細資訊，請參閱 列出匯入匯出的輸出值的堆疊 。	2010-05-15
轉換	2016 年 11 月 17 日	指定 CloudFormation 用來處理無 AWS 伺服器應用程式的 AWS SAM 語法的無伺服器應用程式模型 (AWS SAM)。如需詳細資訊，請參閱 轉換 。	2010-05-15
新資源	2016 年 11 月 17 日	AWS::SNS::Subscription 使用 AWS::SNS::Subscription 資源讓端點訂閱 Amazon Simple Notification Service 主題。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 11 月 17 日	AWS::Lambda::Function 使用 Environment 屬性來指定 AWS Lambda 函數可存取的鍵/值對（環境變數）。 使用 KmsKeyArn 屬性指定 AWS Lambda 用來加密和解密環境變數的 KMS 金鑰。	2010-05-15
新 CLI 命令	2016 年 11 月 17 日	上傳本機成品到 S3 儲存貯體 使用 package 命令將 CloudFormation 範本中參考的本機成品上傳至 S3 儲存貯體。 使用轉換快速部署範本 使用 deploy 命令，將建立和執行變更集的動作結合為單一命令。此命令對於快速建立或更新內含轉換之堆疊很有用。	2010-05-15
更新的資源	2016 年 11 月 03 日	AWS::CloudFront::Distribution 對於 DistributionConfig 屬性，使用 HttpVersion 屬性來指定可供檢視器用於與 Amazon CloudFront 通訊的最新 HTTP 版本。 對於 ForwardedValues 屬性，使用 QueryStringCacheKeys 屬性來指定查詢字串參數，供 CloudFront 用來決定要快取的內容。	2010-05-15
堆疊匯出清單	2016 年 11 月 03 日	使用 CloudFormation 主控台、API 或 AWS CLI 來查看區域所有匯出輸出值的清單。如需詳細資訊，請參閱 匯出堆疊輸出值 。	2010-05-15
使用堆疊進行持續交付	2016 年 11 月 03 日	使用 AWS CodePipeline 以使用 CloudFormation 堆疊建置持續交付工作流程。如需詳細資訊，請參閱 使用 CodePipeline 執行持續交付 。	2010-05-15

變更	版本日期	Description	API 版本
在復原過程中略過資源	2016 年 11 月 03 日	如果您有堆疊處於 UPDATE_ROLLBACK_FAILED 狀態，請使用 ContinueUpdateRollback 動作的 ResourcesToSkip 參數來略過 CloudFormation 無法轉返的資源。如需詳細資訊，請參閱 更新復原失敗 中的「故障診斷」一節。	2010-05-15
變更集增強功能	2016 年 11 月 03 日	您可以 使用變更集建立新的堆疊 。	2010-05-15
更新的資源	2016 年 10 月 12 日	AWS::ElastiCache::CacheCluster 更新 CacheNodeType 屬性，無需替換叢集。 AWS::ElastiCache::ReplicationGroup 您可以建立能包含多個節點群組 (碎片) 的 Redis (已啟用叢集模式) 複寫群組，每個群組皆有主要叢集和僅供讀取複本。 AWS::ElastiCache::SubnetGroup 使用 CacheSubnetGroupName 屬性來指定 Amazon ElastiCache 子網路群組的名稱。	2010-05-15
新資源	2016 年 10 月 06 日	AWS::ApiGateway::UsagePlan 使用 AWS::ApiGateway::UsagePlan 資源為已部署的 Amazon API Gateway API 指定用量計劃。 AWS::CodeCommit::Repository 使用 AWS::CodeCommit::Repository 資源來建立由 Amazon Web Services 託管的 CodeCommit 儲存庫。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 10 月 06 日	AWS::ApiGateway::Authorizer 使用 ProviderARNs 屬性將 Amazon Cognito 使用者集區當作 Amazon API Gateway API 授權方。 AWS::ApiGateway::Deployment StageName 屬性已不再需要。 AWS::ElasticLoadBalancingV2::TargetGroup 對於 GetAtt 函數，請使用 LoadBalancerArns 屬性以擷取將流量路由至目標群組之負載平衡器的 Amazon Resource Name (ARN)。 AWS::RDS::DBInstance 使用 Domain 和 DomainIAMRoleName 屬性，在使用者連接至 RDS 資料庫執行個體時使用 Windows 身分驗證。 AWS::EC2::SecurityGroupEgress 使用 DestinationPrefixListId 屬性指定 Amazon VPC 端點 AWS 的服務字首。	2010-05-15
跨堆疊參考增強功能	2016 年 10 月 06 日	使用內部函數自訂 匯出 的 Name 值，或參考 ImportValue 函數中的值。	2010-05-15
CloudFormation 服務角色	2016 年 9 月 26 日	使用 CloudFormation 堆疊操作的 AWS Identity and Access Management (IAM) 服務角色。CloudFormation 使用該角色的憑證以代表您呼叫堆疊資源。如需詳細資訊，請參閱 CloudFormation 服務角色 。	2010-05-15
新功能	2016 年 9 月 19 日	您可以使用 Export 輸出欄位和 Fn::ImportValue 內部函數，讓一個堆疊參考至另一個堆疊的資源輸出。如需詳細資訊，請參閱 輸出 、 Fn::ImportValue 和 逐步解說：參考其他 CloudFormation 堆疊中的資源輸出 。	2010-05-15

變更	版本日期	Description	API 版本
YAML 支援	2016 年 9 月 19 日	您可以使用 YAML 格式來編寫 CloudFormation 範本。例如，YAML 也可讓您將評論新增至您的範本，或使用短格式的內部函數。如需詳細資訊，請參閱 CloudFormation 範本格式 。	2010-05-15
新的內部函數	2016 年 9 月 19 日	使用 <code>Fn::Sub</code> 函數以您指定的值替代輸入字串中的變數。如需詳細資訊，請參閱 Fn::Sub 。	2010-05-15
新資源	2016 年 9 月 19 日	AWS::KMS::Alias 使用 <code>AWS::KMS::Alias</code> 資源建立 AWS KMS key 的別名。	
更新的資源	2016 年 9 月 19 日	AWS::EC2::SpotFleet 對於 <code>LaunchSpecifications</code> 屬性，請使用 <code>SpotPrice</code> 屬性指定特定執行個體類型的競價。 AWS::ECS::Cluster 使用 <code>ClusterName</code> 屬性來指定 Amazon Elastic Container Service 叢集的名稱。 AWS::ECS::TaskDefinition 使用 <code>TaskRoleArn</code> 屬性指定 Amazon Elastic Container Service 容器用來代表您進行 AWS 呼叫 AWS Identity and Access Management 的角色。 使用 <code>Family</code> 屬性將任務定義註冊至特定系列。 AWS::Elasticsearch::Domain 使用 <code>ElasticsearchVersion</code> 屬性來指定要使用哪個版本的 Elasticsearch。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2016 年 8 月 11 日	使用以下 Elastic Load Balancing Application Load Balancer 資源，將傳入的應用程式資料分發至多個可用區域中的多個目標，例如 EC2 執行個體： • AWS::ElasticLoadBalancingV2::Listener • AWS::ElasticLoadBalancingV2::ListenerRule • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup	2010-05-15
更新的資源	2016 年 8 月 11 日	AWS::AutoScaling::AutoScalingGroup 使用 TargetGroupARNs 屬性將 Auto Scaling 群組與一或多個 Application Load Balancer 目標群組建立關聯。 AWS::ECS::Service 對於負載 LoadBalancers 屬性，使用 TargetGroupArn 屬性將 Amazon Elastic Container Service 服務與 Application Load Balancer 目標群組建立關聯。	2010-05-15
新資源	2016 年 8 月 09 日	CloudFormation 已新增下列資源： AWS::ApplicationAutoScaling::ScalableTarget 和 AWS::ApplicationAutoScaling::ScalingPolicy 使用 Application Auto Scaling 擴展政策來定義目標資源何時及如何擴展。 AWS::CertificateManager::Certificate 佈建可與其他 AWS 服務搭配使用的 AWS Certificate Manager 憑證，以啟用安全連線。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 8 月 09 日	CloudFormation 已更新下列資源： AWS::CloudFront::Distribution 對於分佈組態 ViewerCertificate 屬性，您可以指定 AWS Certificate Manager 憑證。對於分發組態 Origin 屬性，您可以為自訂原始伺服器指定自訂標頭和 SSL 通訊協定。 AWS::EFS::FileSystem 您可以為 Amazon Elastic File System 檔案系統指定效能模式。	2010-05-15
新資源	2016 年 7 月 20 日	AWS IoT 使用 AWS IoT 宣告 AWS IoT 政策、X.509 憑證、政策與委託人之間的關聯 (X.509 憑證或其他憑證)、AWS IoT 物件、委託人與物件之間的關聯，或 AWS IoT 規則。 • AWS::IoT::Certificate • AWS::IoT::Policy • AWS::IoT::PolicyPrincipalAttachment • AWS::IoT::Thing • AWS::IoT::ThingPrincipalAttachment • AWS::IoT::TopicRule	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 7 月 20 日	CloudFormation 已更新下列資源： AWS::IAM::Group, AWS::IAM::Role, AWS::IAM::User 使用名稱屬性來指定 AWS Identity and Access Management (IAM) 資源的自訂名稱。 AWS::ApiGateway::Method 對於 Integration 屬性，您可以使用 PassthroughBehavior 屬性來指定 Amazon API Gateway 何時將請求傳遞至目標後端。 AWS::ApiGateway::Model 和 AWS::ApiGateway::RestApi 您可以指定 Schema 的 JSON 物件和 Body 屬性。	2010-05-15
Auto Scaling 群組 UpdatePolicy	2016 年 6 月 9 日	對於 UpdatePolicy 屬性，使用 AutoScalingReplacingUpdate 屬性指定當您更新 Auto Scaling 群組時，是否替換 Auto Scaling 群組及其包含的執行個體。在替換過程中，CloudFormation 將保留舊的 Auto Scaling 群組直到它成功建立新的群組，因此，如果更新失敗，CloudFormation 可以轉返至舊的 Auto Scaling 群組。如需詳細資訊，請參閱 UpdatePolicy 屬性。	2010-05-15
新資源	2016 年 6 月 9 日	CloudFormation 已新增下列資源： AWS::EC2::FlowLog 建立 Amazon Elastic Compute Cloud 流程日誌，它可擷取指定的網路介面、子網路或 VPC 的 IP 流量。 AWS::KinesisFirehose::DeliveryStream 建立交付串流，將即時串流資料傳遞至目的地，例如 Amazon Simple Storage Service、Amazon Redshift 或 Amazon OpenSearch Service。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 6 月 9 日	CloudFormation 已更新下列資源： AWS::Kinesis::Stream 使用 Name 屬性來指定 Amazon Kinesis 串流的名稱。 AWS::Lambda::Function 對於 Code 屬性，您可以將 ZipFile 屬性和 cfn 回應模組用於 nodejs4.3 執行時間環境。 AWS::SNS::Topic CloudFormation 已啟用 Amazon Simple Notification Service 主題資源的更新。	2010-05-15
新資源	2016 年 4 月 25 日	使用 AWS::EC2::Host 資源可配置完全專屬的實體伺服器以啟動 EC2 執行個體。	2010-05-15

變更	版本日期	Description	API 版本
更新的資源	2016 年 4 月 25 日	AWS::EC2::Instance 使用 Affinity 和 HostId 屬性在 Amazon Elastic Compute Cloud 專用主機上啟動執行個體。 AWS::ECS::Service 使用 DeploymentConfiguration 屬性設定部署期間可執行多少個任務。 AWS::ECS::TaskDefinition CloudFormation 已新增支援其他 Amazon Elastic Container Service 容器定義屬性。 AWS::GameLift::Fleet 使用 MaxSize 和 MinSize 屬性來指定 Amazon GameLift Servers 機群中允許的 EC2 執行個體數目上限和下限。 AWS::Lambda::Function 使用 FunctionName 屬性來指定 AWS Lambda 函數的名稱。您也可以使用 Python 2.7 來指定內嵌函數。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2016 年 4 月 18 日	Amazon API Gateway 使用 Amazon API Gateway 資源來發佈、維護和監控任何規模的 API。您可以建立 API，用戶端可呼叫它來存取您的後端服務，例如執行 EC2 執行個體的應用程式或在 AWS Lambda 上執行的程式碼。 • AWS::ApiGateway::Account • AWS::ApiGateway::ApiKey • AWS::ApiGateway::Authorizer • AWS::ApiGateway::BasePathMapping • AWS::ApiGateway::ClientCertificate • AWS::ApiGateway::Deployment • AWS::ApiGateway::Method • AWS::ApiGateway::Model • AWS::ApiGateway::Resource • AWS::ApiGateway::RestApi • AWS::ApiGateway::Stage AWS::Events::Rule 建立 Amazon CloudWatch Events 規則，它可監控您帳戶 (事件) 中對 AWS 資源所做的變更。如果傳入事件符合您在規則中描述的條件，Amazon CloudWatch Events 會將訊息傳送至您指定的目標並啟動目標，例如 AWS Lambda 函數或 Amazon Simple Notification Service 主題。 AWS::WAF::SizeConstraintSet 和 AWS::WAF::XssMatchSet 使用兩個 AWS WAF 規則來檢查 Web 請求的大小或防止跨網站指令碼攻擊。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2016 年 3 月 31 日	使用 AWS::Lambda::Alias 資源為您的 AWS Lambda 函數建立別名，並使用 AWS::Lambda::Version 資源來建立函數的版本。	2010-05-15
更新的資源	2016 年 3 月 31 日	CloudFormation 已更新下列資源： AWS::EMR::Cluster 和 AWS::EMR::InstanceGroupConfig 使用 EbsConfiguration 屬性為 Amazon EMR 叢集或執行個體群組設定 Amazon Elastic Block Store 儲存磁碟區。 AWS::Lambda::Function 使用 VpcConfig 屬性來允許 AWS Lambda 函數存取 VPC 中的資源。 AWS::S3::Bucket 對於 Amazon Simple Storage Service 生命週期規則，您可以指定多個轉換規則，指定物件何時轉換到指定的儲存類別。	2010-05-15
變更集	2016 年 3 月 29 日	更新堆疊之前，請使用變更集，了解您的變更會如何影響執行中的資源。如需詳細資訊，請參閱 透過變更集更新堆疊 。	2010-05-15
新資源	2016 年 3 月 15 日	使用 AWS::GameLift::Alias 、 AWS::GameLift::Build 和資源 AWS::GameLift::Fleet 來部署多玩家遊戲伺服器 AWS。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2016 年 2 月 26 日	CloudFormation 已新增下列資源： AWS::ECR::Repository 建立 Amazon Elastic Container Registry 儲存庫，使用者可用於推送和提取 Docker 影像。 AWS::EC2::NatGateway 使用網路位址轉譯器 (NAT) 閘道，使私有子網路中的 EC2 執行個體可連接至網際網路。 AWS::Elasticsearch::Domain 建立執行舊版 Elasticsearch OSS 叢集的 Amazon OpenSearch Service 網域。 AWS::EMR::Cluster, AWS::EMR::InstanceGroupConfig, AWS::EMR::Step 使用 Amazon EMR 資源可協助您分析和處理大量資料。您可以建立叢集，然後在叢集上執行任務。	2010-05-15
更新的資源	2016 年 2 月 26 日	CloudFormation 已更新下列資源： AWS::CloudTrail::Trail 使用 <code>IsMultiRegionTrail</code> 屬性指定是否要在您建立堆疊的區域中或所有區域中建立 AWS CloudTrail 追蹤。 AWS::Config::ConfigurationRecorder 對於記錄群組，請使用 <code>IncludeGlobalResourceTypes</code> 屬性來記錄所有全域資源類型。 AWS::RDS::DBCluster 使用 <code>KmsKeyId</code> 和 <code>StorageEncrypted</code> 屬性來加密叢集中的資料庫執行個體。	2010-05-15

變更	版本日期	Description	API 版本
保留資源	2016 年 2 月 26 日	對處於 DELETE_FAILED 狀態中的堆疊，使用 RetainResources 參數來保留 CloudFormation 無法刪除的資源。如需詳細資訊，請參閱 刪除堆疊失敗 。	2010-05-15
更新堆疊標籤	2016 年 2 月 26 日	當您更新堆疊時，可以新增、修改或移除堆疊標籤。如需詳細資訊，請參閱 CloudFormation 堆疊更新 。	2010-05-15
繼續復原失敗的更新復原	2016 年 1 月 25 日	對處於 UPDATE_ROLLBACK_FAILED 狀態中的堆疊，您可以繼續復原更新，使您的堆疊處於運作狀態。透過此方式，您可以使堆疊回到其原始設定，然後嘗試再次更新。如需詳細資訊，請參閱 繼續復原更新 。	2010-05-15
新的範例範本可用於亞太區域 (首爾) 區域。	2016 年 1 月 7 日	以下 CloudFormation 範例範本集合適用於 ap-northeast-2 區域： • 範例解決方案 • 應用程式架構 • 服務 如需詳細資訊，請參閱使用 CloudFormation 範本。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2015 年 12 月 28 日	CloudFormation 已新增下列資源： AWS::DirectoryService::MicrosoftAD 使用 Microsoft Active Directory 資源在 中建立 Microsoft Active Directory 目錄 AWS。 AWS::Logs::Destination 和 AWS::Logs::LogStream 使用 Amazon CloudWatch Logs 資源來建立目的地，以分別即時處理日誌資料或建立日誌串流。 AWS::WAF::ByteMatchSet, AWS::WAF::IPSet, AWS::WAF::Rule, AWS::WAF::SqlInjectionMatchSet, 和 AWS::WAF::WebACL 使用 AWS WAF 資源來控制和監控對內容的 Web 請求。	2010-05-15

變更	版本日期	Description	API 版本
資源更新	2015 年 12 月 28 日	CloudFormation 已更新下列資源： AWS::CloudFront::Distribution 對於分佈組態，請使用 WebACLIId 屬性將 AWS WAF Web 存取控制清單 (ACL) 與 Amazon CloudFront 分佈建立關聯。對於快取行為和預設快取行為，您可以指定預設和最大的存留時間 (TTL) 值。 AWS::DynamoDB::Table 您可以建立、更新或刪除全域次要索引，無需替換您的 Amazon DynamoDB 資料表。 AWS::S3::Bucket 使用 ReplicationConfiguration 屬性指定要複寫的物件及其存放位置。 使用 NotificationConfiguration 屬性中的屬性來指定篩選條件，讓 Amazon Simple Storage Service 針對您指定的物件傳送通知。	2010-05-15
參數群組和排序	2015 年 12 月 3 日	當使用者以您的範本建立或更新堆疊時，使用 AWS::CloudFormation::Interface 中繼資料金鑰來分組和排序 CloudFormation 主控台參數。	2010-05-15
更新政策屬性	2015 年 12 月 3 日	對於 Auto Scaling 更新政策屬性，使用 MinSuccessfulInstancesPercent 屬性來指定必須為成功更新發出成功訊號的執行個體百分比。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2015 年 12 月 3 日	CloudFormation 已新增下列資源： AWS::CodePipeline::Pipeline 和 AWS::CodePipeline::CustomActionType 使用 CodePipeline 資源來建立管道，它可描述軟體變更如何進行發程序。 AWS::Config::ConfigurationRecorder、AWS::Config::DeliveryChannel 與 AWS::Config::ConfigRule 使用 AWS Config 資源來監控特定 AWS 資源的組態變更。 AWS::KMS::Key 使用 AWS Key Management Service (AWS KMS) 資源在 中建立客戶受管金鑰 AWS KMS，使用者可以使用該金鑰來加密少量資料。 AWS::SSM::Document 使用 Amazon EC2 Systems Manager 建立文件以指定執行個體上的組態。	2010-05-15

變更	版本日期	Description	API 版本
資源更新	2015 年 12 月 3 日	CloudFormation 已更新下列資源： AWS::AutoScaling::LaunchConfiguration 指定 EBS 磁碟區是否已加密。 AWS::AutoScaling::ScalingPolicy 您可以使用兩個不同的政策類型 (簡單和步進擴展)，以指定當突破 Amazon CloudWatch (CloudWatch) 警示時，Auto Scaling 群組如何擴展。 AWS::CloudTrail::Trail 使用 CloudWatch 屬性將日誌傳送至 CloudWatch 日誌群組。您可以將標籤新增至線索，並指定要用來加密日誌 AWS KMS key 的。 AWS::CodeDeploy::Application、AWS::CodeDeploy::DeploymentConfig 與 AWS::CodeDeploy::DeploymentGroup 使用 ApplicationName、DeploymentConfigName 及 DeploymentGroupName 屬性來指定 CodeDeploy 資源的自訂名稱。 AWS::DynamoDB::Table 使用 StreamSpecification 屬性來指定設定，以擷取對存放在 Amazon DynamoDB (DynamoDB) 資料表中的項目所做的變更。 AWS::EC2::Instance 使用 SsmAssociations 屬性將 Amazon EC2 Systems Manager 文件與執行個體建立關聯。 AWS::EC2::SpotFleet 使用 AllocationStrategy 屬性指定如何在所有 Spot 集區之間配置目標容量。使用 ExcessCap	2010-05-15

變更	版本日期	Description	API 版本
		acityTerminationPolicy 屬性指定如果目標容量低於 Spot 機群的大小，要如何終止執行個體。 AWS::Redshift::Cluster 使用 KmsKeyId 屬性指定 AWS KMS key，以加密 Amazon Redshift 叢集中的資料。 AWS::WorkSpaces::Workspace 使用加密屬性來加密存放在磁碟區的資料。	
資源更新	2015 年 11 月 4 日	對於 AWS::EC2::Volume 資源，請使用 AutoEnableIO 屬性，在磁碟區的資料變成不一致時自動恢復 I/O 操作。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2015 年 10 月 1 日	CloudFormation 已新增下列資源： AWS::CodeDeploy::Application、AWS::CodeDeploy::DeploymentGroup 與 AWS::CodeDeploy::DeploymentConfig 使用 CodeDeploy 資源，以建立和套用部署至 EC2 或內部部署執行個體。 AWS::DirectoryService::SimpleAD 使用 Simple Active Directory 資源來建立 Directory Service Simple AD，它是相容於 Microsoft Active Directory 的目錄。 AWS::EC2::PlacementGroup 使用置放群組在低延遲網路中建立執行個體叢集。 AWS::EC2::SpotFleet 使用 Spot 機群啟動 Spot 執行個體集合，它將執行可中斷的任務。 AWS::Lambda::EventSourceMapping 使用事件來源映射資源，將串流指定為 AWS Lambda (Lambda) 函數的事件來源。 AWS::Lambda::Permission 使用 Lambda 許可將陳述式新增至 Lambda 函數的政策。 AWS::Logs::SubscriptionFilter 使用訂閱篩選條件來定義哪些日誌事件會傳送到您的 Kinesis 串流。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::RDS::DBCluster 和 AWS::RDS::DBClusterParameterGroup 使用叢集和叢集參數群組資源，以建立 Amazon Aurora 資料庫叢集。 AWS::WorkSpaces::Workspace 使用 WorkSpaces 建立以雲端為基礎的桌面體驗。	
資源更新	2015 年 10 月 1 日	CloudFormation 已更新下列資源： AWS::ElastiCache::ReplicationGroup 使用 <code>Fn::GetAtt</code> 內部函數來取得唯讀複本地址和連接埠的清單。 AWS::OpsWorks::Stack 使用 <code>AgentVersion</code> 屬性指定特定 OpsWorks 代理程式。 AWS::OpsWorks::App 使用 <code>Environment</code> 屬性來指定 OpsWorks 應用程式的環境變數。 AWS::S3::Bucket 對於 NotificationConfiguration 屬性，您可以設定 Lambda 函數和 Amazon Simple Queue Service (Amazon SQS) 佇列的通知設定。	2010-05-15
IAM 條件金鑰	2015 年 10 月 1 日	對於 AWS Identity and Access Management (IAM) 政策，請使用 CloudFormation 特定條件金鑰來指定 IAM 政策何時生效。如需詳細資訊，請參閱使用 AWS Identity and Access Management 控制存取。	2010-05-15

變更	版本日期	Description	API 版本
CloudFormation 設計工具	2015 年 10 月 1 日	使用 CloudFormation 設計工具 以利用拖放式介面建立和修改範本。	2010-05-15
新資源	2015 年 8 月 24 日	使用 AWS::EC2::VPCEndpoint 資源在您的 VPC 與其他 AWS 服務之間建立私有連線。	2010-05-15
資源更新	2015 年 8 月 24 日	CloudFormation 已更新下列資源： AWS::ElasticBeanstalk::Environment 使用 Tags 屬性指定 (Elastic Beanstalk) 環境的標籤 AWS Elastic Beanstalk (鍵/值對)。 AWS::Lambda::Function 對於 Code 屬性，使用 ZipFile 屬性直接在範本中撰寫 Lambda 函數的原始程式碼。目前，您只能在 ZipFile 執行時間環境中使用 nodejs 屬性。您仍可在所有執行時間環境中指向 S3 儲存貯體中的檔案，例如 java8 和 nodejs。 AWS::OpsWorks::Instance 使用 EbsOptimized 屬性來指示執行個體是否已針對 Amazon Elastic Block Store (Amazon EBS) I/O 進行最佳化。 AWS::RDS::DBInstance 對於 SourceDBInstanceIdentifier 屬性，您可以指定另一個區域中的資料庫執行個體，以建立跨區域僅供讀取複本。	2010-05-15

變更	版本日期	Description	API 版本
Amazon S3 範本 URL	2015 年 8 月 24 日	對於已啟用版本控制的儲存貯體，在您建立或更新堆疊時，可在 Amazon S3 範本 URL 中指定版本 ID，例如 https://s3.amazonaws.com/templates/myTemplate.template?versionId=123ab1cdeKd0W5IH4GAcYbEngcpTJTdW 。	2010-05-15
新資源	2015 年 8 月 3 日	使用 AWS::EFS::FileSystem 資源來建立 Amazon Elastic File System (Amazon EFS) 檔案系統，使用 AWS::EFS::MountTarget 資源來建立檔案系統的掛載點。	2010-05-15
許可要求變更	2015 年 6 月 11 日	現在當您建立或更新 AWS::RDS::DBInstance 資源時，也必須具備呼叫 <code>ec2:DescribeAccountAttributes</code> 動作的許可。	2010-05-15

變更	版本日期	Description	API 版本
新資源	2015 年 6 月 11 日	CloudFormation 已新增下列資源： AWS::DataPipeline::Pipeline 使用資料管道，將資料移動與轉換作業自動化。 Amazon Elastic Container Service 資源 使用 AWS::ECS::Service、AWS::ECS::Cluster 和 AWS::ECS::TaskDefinition 資源，在 EC2 執行個體叢集上建立 Docker 容器。 AWS::ElastiCache::ReplicationGroup 使用複寫群組建立一個節點的集合，其中包含一個主要讀寫叢集和最多五個次要唯讀叢集。 AWS::IAM::ManagedPolicy 使用 受管政策在您的 AWS 帳戶中建立政策，可用來將許可套用至 IAM 使用者、群組和角色。 AWS::Lambda::Function 使用 Lambda 函數來執行程式碼以回應事件。 AWS::RDS::OptionGroup 使用選項群組以協助您建立和管理 Amazon Relational Database Service (Amazon RDS) 資料庫。	2010-05-15

變更	版本日期	Description	API 版本
資源更新	2015 年 6 月 11 日	CloudFormation 已更新下列資源： AWS::EC2::Subnet 使用 MapPublicIpOnLaunch 屬性，自動將公有 IP 地址指派給子網路中的執行個體。 AWS::ElastiCache::CacheCluster 使用 SnapshotName 屬性，將快照資料還原至新的 Redis 快取叢集。 AWS::IAM::User 對於 LoginProfile 屬性，使用 PasswordResetRequired 屬性以要求使用者在登入 AWS 管理主控台時必須設定新的密碼。 AWS::OpsWorks::Layer 使用 LifecycleEventConfiguration 屬性來設定 OpsWorks layer 的生命週期事件。 AWS::S3::Bucket 對於 LifecycleConfiguration 屬性，請使用 NoncurrentVersionExpirationInDays 和 NoncurrentVersionTransition 屬性，為不是目前的物件版本指定生命週期規則。	2010-05-15

變更	版本日期	Description	API 版本
新的參數類型	2015 年 5 月 19 日	每當您使用 CloudFormation 主控台建立或更新堆疊時，都可以依 ID、名稱或名稱標籤值來搜尋 AWS 特定的參數類型值。 CloudFormation 也新增了對下列 AWS 特定參數類型的支援。如需詳細資訊，請參閱參數。 • AWS::EC2::AvailabilityZone::Name • List<AWS::EC2::AvailabilityZone ::Name> • AWS::EC2::Instance::Id • List<AWS::EC2::Instance ::Id> • AWS::EC2::Image::Id • List<AWS::EC2::Image::Id > • AWS::EC2::SecurityGroup::GroupName • List<AWS::EC2::SecurityGroup::Gr oupName > • AWS::EC2::Volume::Id • List<AWS::EC2::Volume::Id > • AWS::Route53::HostedZone::Id • List<AWS::Route53::HostedZone ::Id>	2010-05-15
新資源	2015 年 4 月 16 日	CloudFormation 已新增下列資源： AWS::AutoScaling::LifecycleHook 使用 Auto Scaling lifecycle hook，在執行個體啟動或終止之後控制其狀態。 AWS::RDS::EventSubscription 使用事件訂閱以取得有關 Amazon RDS 事件的通知。	2010-05-15

變更	版本日期	Description	API 版本
資源更新	2015 年 4 月 16 日	CloudFormation 已更新下列資源： AWS::AutoScaling::AutoScalingGroup 使用 NotificationConfigurations 屬性指定多個通知。 AWS::AutoScaling::LaunchConfiguration 使用 PlacementTenancy 屬性指定執行個體的租用。 使用 ClassicLinkVPCId 和 ClassicLinkVPCSecurityGroups 屬性，將 EC2-Classic 執行個體連結至已啟用 ClassicLink 的 VPC。 AWS::AutoScaling::ScalingPolicy 使用 MinAdjustmentStep 屬性指定在擴展事件中新增或移除執行個體的最低數量。 AWS::CloudFront::Distribution 如需檢視器憑證，請使用 MinimumProtocolVersion 屬性指定最低通訊協定版本。對於快取行為，使用 CachedMethods 屬性來指定 Amazon CloudFront (CloudFront) 要快取哪些方法的回應。對於原始伺服器，使用 OriginPath 來指定供 CloudFront 用於請求內容的路徑。 AWS::ElastiCache::CacheCluster 對於 Memcached 快取叢集，請使用 AZMode 和 PreferredAvailabilityZones 屬性指定多個可用區域 (AZ) 中的節點。 AWS::EC2::Volume 使用 KmsKeyId 屬性為已加密磁碟區指定客戶受管金鑰。	2010-05-15

變更	版本日期	Description	API 版本
		AWS::OpsWorks::Instance 使用 TimeBasedAutoScaling 屬性以根據您指定的排程，自動擴展執行個體。 AWS::OpsWorks::Layer 使用 LoadBasedAutoScaling 屬性指定以負載為基礎的擴展政策。對於磁碟區組態、使用 VolumeType 和 Iops 屬性分別指定磁碟區類型和每秒 I/O 操作數量。 AWS::RDS::DBInstance 使用 CharacterSetName 屬性為支援的資料庫引擎指定字元集。 使用 StorageEncrypted 屬性指示資料庫執行個體是否要加密，並使用 KmsKeyId 為已加密的資料庫執行個體指定客戶受管金鑰。 AWS::Route53::HealthCheck 使用 HealthCheckTags 屬性建立標籤與運作狀態檢查的關聯。 AWS::Route53::HostedZone 使用 VPCs 屬性建立私有託管區域。 使用 HostedZoneTags 屬性建立標籤與託管區域的關聯。	
新的範本部分	2015 年 4 月 16 日	將 中繼資料 部分新增至您的範本，以包含描述範本的任意 JSON 物件，例如設計或實作詳細資訊。	2010-05-15
資源更新	2015 年 4 月 8 日	對於 AWS::CloudFormation::CustomResource 資源，您可以在 ServiceToken 屬性中指定 Lambda 函數 Amazon Resource Name (ARN)。	2010-05-15

變更	版本日期	Description	API 版本
Amazon RDS update	2014 年 12 月 24 日	CloudFormation 增加兩個適用於 RDS 資料庫執行個體的新屬性。您可以建立選項群組與資料庫執行個體的關聯，然後指定資料庫執行個體儲存體類型。如需詳細資訊，請參閱 AWS::RDS::DBInstance 。	2010-05-15
Elastic Load Balancing update	2014 年 12 月 24 日	您可以使用 <code>ConnectionSettings</code> 屬性指定連線可以保持閒置的時間長度。如需詳細資訊，請參閱 AWS::ElasticLoadBalancing::LoadBalancer 。	2010-05-15
Route 53 更新	2014 年 11 月 6 日	您現在可以佈建和管理 Route 53 託管區域 、 運作狀態檢查 、 容錯移轉記錄集 和 地理位置記錄集 。	2010-05-15
Auto Scaling 滾動更新增強功能	2014 年 11 月 6 日	在更新過程中，您可以使用 <code>WaitOnResourceSignals</code> 旗標以指示 CloudFormation 等待執行個體發出成功訊號。使用這種方式，在目前批次備妥之前，CloudFormation 不會更新下一個批次的執行個體。如需詳細資訊，請參閱 UpdatePolicy 屬性。	2010-05-15
新的 VPC Fn::GetAtt 屬性	2014 年 11 月 6 日	透過 VPC ID，您可以擷取該 VPC 的預設安全群組和網路 ACL。如需詳細資訊，請參閱 Fn::GetAtt 。	2010-05-15
新的 AWS 特定參數類型	2014 年 11 月 6 日	您可以在 CloudFormation 範本中指定 AWS 特定的參數類型。在 CloudFormation 主控台中，這些參數類型提供有效值的下拉式清單。使用 API 或 時 AWS CLI，CloudFormation 可以在建立或更新堆疊之前，快速驗證這些參數類型的值。如需詳細資訊，請參閱 參數 。	2010-05-15
CreationPolicy 屬性	2014 年 11 月 6 日	使用 <code>CreationPolicy</code> 屬性，您可以指示 CloudFormation 等待 EC2 執行個體上的應用程式備妥，然後再建立堆疊。您可以使用建立政策，而非等待條件和等待條件處理。如需詳細資訊，請參閱 CreationPolicy 屬性。	2010-05-15

變更	版本日期	Description	API 版本
Amazon CloudFront 轉發值	2014 年 9 月 29 日	對於快取行為，您可以轉發標頭到原始伺服器。請參閱 ForwardedValues 。	2010-05-15
AWS OpsWorks update	2014 年 9 月 29 日	對於 Chef 11.10，您可以使用 ChefConfiguration 屬性來啟用 Berkshelf。您也可以將 AWS OpsWorks 內建安全群組與 AWS OpsWorks 堆疊搭配使用。如需詳細資訊，請參閱 AWS::OpsWorks::Stack 。	2010-05-15
Elastic Load Balancing 標記支援	2014 年 9 月 29 日	AWS CloudFormation 使用堆疊層級標籤來標記 Elastic Load Balancing 負載平衡器。您也可以將自己的標籤新增至負載平衡器。請參閱 AWS::ElasticLoadBalancing::LoadBalancer 。	2010-05-15
Amazon Simple Notification Service 主題政策更新	2014 年 9 月 29 日	您現在可以更新 Amazon SNS 主題政策。如需詳細資訊，請參閱 AWS::SNS::TopicPolicy 。	2010-05-15
RDS 資料庫執行個體更新	2014 年 9 月 5 日	可以使用 AWS::RDS::DBInstance 資源中的 PubliclyAccessible 屬性，指定資料庫執行個體是否要面向國際網路。	2010-05-15
UpdatePolicy 屬性更新	2014 年 9 月 05 日	您可以為擁有關聯排程動作的 Auto Scaling 群組指定更新政策。如需詳細資訊，請參閱 UpdatePolicy 屬性 。	2010-05-15
Amazon CloudWatch 支援	2014 年 7 月 10 日	可以使用 CloudFormation 來佈建和管理 Amazon CloudWatch Logs (CloudWatch Logs) 日誌群組和指標篩選條件。如需詳細資訊，請參閱 AWS::Logs::LogGroup 或 AWS::Logs::MetricFilter 。	2010-05-15

變更	版本日期	Description	API 版本
Amazon CloudFront 分佈組態更新	2014 年 6 月 17 日	您可以指定其他 CloudFront 分佈組態屬性： • 自訂的錯誤回應可定義 4xx 和 5xx HTTP 狀態碼的自訂錯誤訊息。 • 價格分級可定義您要為 CloudFront 服務支付的最高價。 • 限制可定義何人可查看您的內容。 • 檢視器憑證可指定當檢視器使用 HTTPS 時，要使用的憑證。 • 對於快取行為，您可以指定允許 HTTP 方法並指示是否要轉發 Cookie。 如需詳細資訊，請參閱AWS::CloudFront::Distribution。	2010-05-15
EC2 執行個體更新	2014 年 6 月 17 日	您可以指定當您呼叫執行個體的作業系統關閉命令時，執行個體是否要停止或終止。如需詳細資訊，請參閱AWS::EC2::Instance。	2010-05-15
EBS 磁碟區更新	2014 年 6 月 17 日	您可以使用已加密的 EBS 磁碟區與支援的執行個體類型。如需詳細資訊，請參閱AWS::EC2::Volume。	2010-05-15
新的 Amazon VPC 對等互連	2014 年 6 月 17 日	您可以使用 CloudFormation 來建立 Amazon Virtual Private Cloud (Amazon VPC) 對等連線，它可在兩個 VPC 之間建立網路連線。如需詳細資訊，請參閱AWS::EC2::VPCPeeringConnection。	2010-05-15
Amazon EC2 Auto Scaling 群組更新	2014 年 6 月 17 日	您可以指定現有的叢集置放群組，在其中啟動 Amazon EC2 Auto Scaling 群組的執行個體。如需詳細資訊，請參閱AWS::AutoScaling::AutoScalingGroup。	2010-05-15
AWS CloudTrail 支援	2014 年 6 月 17 日	CloudFormation 支援 AWS CloudTrail，可擷取從 AWS 您的帳戶發出的 API 呼叫，並在您指定的位置發佈日誌。如需詳細資訊，請參閱AWS::CloudTrail::Trail。	2010-05-15

變更	版本日期	Description	API 版本
更新堆疊增強功能	2014 年 5 月 12 日	CloudFormation 支援更新堆疊的其他功能： - 可以更新 CloudFormation 堆疊參數，無需重新提交堆疊的範本。 - 可以新增或移除 CloudFormation 堆疊的 Amazon SNS 通知主題。 如需詳細資訊，請參閱 CloudFormation 堆疊更新。	2010-05-15
Amazon Kinesis 支援	2014 年 5 月 6 日	可以使用 CloudFormation 來建立 Amazon Kinesis 串流，以從資料來源擷取和傳輸資料記錄。如需詳細資訊，請參閱AWS::Kinesis::Stream。	2010-05-15
新的 S3 儲存貯體屬性	2014 年 5 月 5 日	CloudFormation 支援額外的 S3 儲存貯體屬性： - 跨來源資源共享 (CORS) 可定義儲存貯體中物件的跨來源資源共享。 - 生命週期可定義 Amazon S3 在物件存留期中管理物件的方式。 - 存取記錄政策可擷取有關對儲存貯體發出之請求的資訊。 - 通知可定義要報告哪些事件以及要傳送哪個 Amazon SNS 主題的訊息。 - 版本控制可啟用儲存貯體中所有物件的多個版本。 - 重新導向和路由規則可管理對儲存貯體之網站端點發出之請求的重新導向行為。 如需詳細資訊，請參閱AWS::S3::Bucket。	2010-05-15
Amazon EC2 Auto Scaling 支援	2014 年 5 月 5 日	CloudFormation 支援 Auto Scaling 群組的指標集合。如需詳細資訊，請參閱AWS::AutoScaling::AutoScalingGroup。	2010-05-15

變更	版本日期	Description	API 版本
Fn::If update	2014 年 5 月 5 日	您可以在範本的輸出部分中使用 Fn::If 內部函數。如需詳細資訊，請參閱 條件函數 。	2010-05-15
使用 記錄 API AWS CloudTrail	2014 年 4 月 2 日	您可以使用 AWS CloudTrail (CloudTrail) 記錄 CloudFormation 請求。可以使用 CloudTrail 來取得您帳戶的 CloudFormation API 呼叫歷史記錄。如需詳細資訊，請參閱 使用 CloudTrail 記錄 CloudFormation API 呼叫 AWS CloudTrail 。	2010-05-15
Elastic Load Balancing update	2014 年 3 月 20 日	您可以指定存取記錄政策來擷取有關對負載平衡器發出之請求的資訊。您也可以指定連接耗盡政策，說明當執行個體撤銷註冊或運作狀態不佳時，如何處理傳送中的請求。如需詳細資訊，請參閱 AWS::ElasticLoadBalancing::LoadBalancer 。	2010-05-15
OpsWorks 支援	2014 年 3 月 3 日	您可以使用 CloudFormation 來佈建和管理 OpsWorks 堆疊。如需詳細資訊，請參閱 AWS::OpsWorks::Stack 或 AWS OpsWorks 範本程式碼片段 。	2010-05-15
Amazon S3 範本大小限制提高	2014 年 2 月 18 日	您最多可以在 Amazon S3 中指定 460,800 位元組的範本大小。	2010-05-15
Amazon Redshift 支援	2014 年 2 月 10 日	可以使用 CloudFormation 來佈建並管理 Amazon Redshift 叢集。如需詳細資訊，請參閱 Amazon Redshift 範本程式碼片段 或 AWS::Redshift::Cluster 。	2010-05-15
S3 儲存貯體和儲存貯體政策更新	2014 年 2 月 10 日	您可以更新 S3 儲存貯體和儲存貯體政策資源的一些屬性。如需詳細資訊，請參閱 AWS::S3::Bucket 或 AWS::S3::BucketPolicy 。	2010-05-15

變更	版本日期	Description	API 版本
Elastic Beanstalk 環境和應用程式版本更新	2014 年 2 月 10 日	您可以更新 Elastic Beanstalk 環境資訊和應用程式版本。如需詳細資訊，請參閱 AWS::ElasticBeanstalk::Environment 、 AWS::ElasticBeanstalk::ConfigurationTemplate 或 AWS::ElasticBeanstalk::ApplicationVersion 。	2010-05-15
Amazon SQS update	2014 年 1 月 29 日	您可以為 Amazon SQS 佇列指定無效字母佇列。如需詳細資訊，請參閱 AWS::SQS::Queue 。	2010-05-15
Auto Scaling 排程動作	2014 年 1 月 27 日	您可以根據排程擴展 Auto Scaling 群組中的 EC2 執行個體數目。透過使用排程，您可以擴展應用程式以回應可預期負載變更。如需詳細資訊，請參閱 AWS::AutoScaling::ScheduledAction 。	2010-05-15
DynamoDB 次要索引	2014 年 1 月 27 日	您可以為 DynamoDB 資料庫建立本機和全域次要索引。透過使用次要索引，您可以使用主金鑰以外的屬性有效率地存取資料。如需詳細資訊，請參閱 AWS::DynamoDB::Table 。	2010-05-15
Auto Scaling update	2014 年 1 月 2 日	您可以指定 Auto Scaling 群組或啟動組態的執行個體 ID。您也可以指定其他 Auto Scaling 區塊型儲存設備屬性。如需詳細資訊，請參閱 AWS::AutoScaling::AutoScalingGroup 或 AWS::AutoScaling::LaunchConfiguration 。	2010-05-15
Amazon SQS update	2014 年 1 月 2 日	您可以更新 SQS 佇列，並指定額外的屬性。如需詳細資訊，請參閱 AWS::SQS::Queue 。	2010-05-15
限額提升	2014 年 1 月 2 日	在 CloudFormation 範本中最多可指定 60 個參數和 60 個輸出。	2010-05-15
新主控台	2013 年 12 月 19 日	新的 CloudFormation 主控台新增功能，例如自動重新整理堆疊事件和字母排序的堆疊參數。	2010-05-15
跨區域負載平衡	2013 年 12 月 19 日	您可以使用跨區域負載平衡功能，將流量路由至所有可用區域 (AZ) 的後端執行個體。如需詳細資訊，請參閱 AWS::ElasticLoadBalancing::LoadBalancer 。	2010-05-15

變更	版本日期	Description	API 版本
AWS Elastic Beanstalk 環境層	2013 年 12 月 19 日	您可以指定 AWS Elastic Beanstalk 佈建資源以支援 Web 伺服器或處理背景處理任務。如需詳細資訊，請參閱 AWS::ElasticBeanstalk::Environment 。	2010-05-15
資源名稱	2013 年 12 月 19 日	您可以將名稱 (實體 ID) 指派至下列資源： - ElastiCache 叢集 - Elastic Load Balancing 負載平衡器 - RDS 資料庫執行個體 如需詳細資訊，請參閱名稱類型。	2010-05-15
VPN 支援	2013 年 11 月 22 日	您可以啟用虛擬私有閘道 (VGW) 將路由傳播到 VPC 的路由表。如需詳細資訊，請參閱 AWS::EC2::VPNGatewayRoutePropagation 。	2010-05-15
有條件地建立資源並指派屬性	2013 年 11 月 8 日	使用輸入參數，您可以在 CloudFormation 範本中定義條件，以控制指定堆疊資源的建立和設定。例如，您可以使用條件在生產環境中建立堆疊資源。使用相同的範本，您可以為測試環境建立類似但容量較低的堆疊資源。如需詳細資訊，請參閱 條件函數 。	2010-05-15
避免意外更新堆疊資源	2013 年 11 月 8 日	您可以避免可能導致堆疊資源意外變更的堆疊更新。例如，如果您有一個堆疊，其資料庫層很少進行更新，您可以設定堆疊政策以避免大多數使用者更新該資料庫層。如需詳細資訊，請參閱 避免更新堆疊資源 。	2010-05-15

變更	版本日期	Description	API 版本
名稱資源	2013 年 11 月 8 日	您可以為特定資源指派名稱，而非使用 CloudFormation 產生的實體 ID。以下 CloudFormation 資源支援命名 • Amazon CloudWatch 警示 • DynamoDB 資料表 • AWS Elastic Beanstalk 應用程式和環境 • Amazon S3 儲存貯體 • Amazon SNS 主題 • Amazon SQS 佇列 如需詳細資訊，請參閱名稱類型。	2010-05-15
指派自訂資源類型	2013 年 11 月 8 日	在您的範本中，您可以為 CloudFormation 自訂資源指定您自己的資源類型 (AWS::CloudFormation::CustomResource)。透過使用自己的自訂資源類型名稱，可以快速識別您堆疊中的自訂資源類型。例如，您可以指定 "Type": "Custom:: <i>MyCustomResource</i>"。如需詳細資訊，請參閱AWS::CloudFormation::CustomResource。	2010-05-15
新增虛擬參數	2013 年 11 月 8 日	您現在可以參考AWS::AccountID 虛擬參數，在 CloudFormation 範本內參考 AWS AccountID。如需詳細資訊，請參閱虛擬參數參考。	2010-05-15
在 IAM 政策中指定堆疊	2013 年 11 月 8 日	您可以允許或拒絕 IAM 使用者、群組或角色在特定 CloudFormation 堆疊上進行操作。例如，您可以拒絕特定堆疊 ID 上的刪除堆疊動作。如需詳細資訊，請參閱使用 AWS Identity and Access Management 控制存取。	2010-05-15

變更	版本日期	Description	API 版本
聯合支援	2013 年 10 月 14 日	CloudFormation 支援來自 IAM 角色的臨時安全憑證，而能夠實現聯合身分和單一登入 AWS 管理主控台之類的案例。您也可以從 EC2 執行個體呼叫 CloudFormation，無需使用 IAM 角色嵌入長期安全登入資料。如需 CloudFormation 和 IAM 的詳細資訊，請參閱 使用 AWS Identity and Access Management 來控制存取 。	2010-05-15
Amazon RDS 僅供讀取複本支援	2013 年 9 月 24 日	您現在可以從來源資料庫執行個體建立 Amazon RDS 僅供讀取複本。如需詳細資訊，請參閱 AWS::RDS::DBInstance 資源中的 SourceDBInstanceId 屬性。	2010-05-15
建立公有 IP 地址與 Auto Scaling 群組內執行個體的關聯	2013 年 9 月 19 日	您現在可以建立公有 IP 地址與 Auto Scaling 群組內執行個體的關聯。如需詳細資訊，請參閱 AWS::AutoScaling::LaunchConfiguration 。	2010-05-15
其他 VPC 支援	2013 年 9 月 17 日	CloudFormation 新增數項增強功能以支援 VPC 和 VPN 功能 - 您可以建立公有 IP 地址和多個私有 IP 地址與 Amazon EC2 網路介面的關聯。如需詳細資訊，請參閱AWS::EC2::NetworkInterface。您也可以建立主要私有 IP 地址與彈性 IP 地址 (EIP) 的關聯。 - 您可以啟用 DNS 支援，並指定 DNS 主機名稱。如需詳細資訊，請參閱AWS::EC2::VPC。 - 您可以在虛擬私有閘道與您的 VPN 閘道之間指定靜態路由。如需詳細資訊，請參閱AWS::EC2::VPNConnectionRoute。	2010-05-15

變更	版本日期	Description	API 版本
Redis 和 VPC 安全群組支援 Amazon ElastiCache	2013 年 9 月 3 日	您現在可以指定 Redis 作為 Amazon ElastiCache (ElastiCache) 叢集的快取引擎。您現在也可以將 VPC 安全群組指派至 ElastiCache 叢集。如需詳細資訊，請參閱 AWS::ElastiCache::CacheCluster 。	2010-05-15
平行堆疊建立、更新和刪除及巢狀堆疊更新	2013 年 8 月 12 日	AWS CloudFormation 現在會平行建立、更新和刪除資源，以改善操作的效能。如果您更新最上層範本，AWS CloudFormation 會自動更新有變動的巢狀堆疊。如需詳細資訊，請參閱 CloudFormation 堆疊更新 。	2010-05-15
VPC 安全群組現在可以設定在 RDS 資料庫執行個體中	2013 年 2 月 28 日	您現在可以使用 CloudFormation 將 VPC 安全群組指派至 RDS 資料庫執行個體。如需詳細資訊，請參閱 AWS::RDS::DBInstance 中的 VPCSecurityGroups 屬性。	2010-05-15
滾動部署 Amazon EC2 Auto Scaling 群組	2013 年 2 月 20 日	CloudFormation 現在支援 Amazon EC2 Auto Scaling 群組上的更新政策，以描述當 Amazon EC2 Auto Scaling 群組新增或移除執行個體時，如何取代或修改 Amazon EC2 Auto Scaling 群組中的執行個體。您可以在建立堆疊時或堆疊更新期間，修改這些設定。 如需詳細資訊和範例，請參閱 UpdatePolicy 屬性 。	2010-05-15
堆疊更新的取消和復原動作	2013 年 2 月 20 日	CloudFormation 支援取消堆疊更新。提出更新請求時，堆疊必須處於 UPDATE_IN_PROGRESS 狀態。以下主題提供詳細資訊： • 取消堆疊更新 • cancel-update-stack • CloudFormation API 參考中的 CancelUpdateStack	2010-05-15

變更	版本日期	Description	API 版本
適用於 Amazon EC2 Auto Scaling 群組的 EBS 最佳化執行個體	2013 年 2 月 20 日	您現在可以在 Amazon EC2 Auto Scaling 群組中佈建 EBS 最佳化執行個體，為自動調整規模執行個體中的 Amazon Elastic Block Store (Amazon EBS) 提供專用傳輸量。此實作類似先前發佈的支援最佳化 Amazon EBS EC2 執行個體。 如需詳細資訊，請參閱 AWS::AutoScaling::LaunchConfiguration 中的新 EbsOptimized 屬性。	2010-05-15
新增文件	2012 年 12 月 21 日	AWS::EC2::Instance 現在提供 BlockDeviceMappings 屬性，讓您為 EC2 執行個體設定區塊型設備映射。 透過這項改變，新增了兩個新的類型： • BlockDeviceMapping • 區塊型儲存設備	2010-05-15
新增文件	2012 年 12 月 21 日	增加新的章節，說明使用最近重新設計的 AWS 管理主控台建立和檢視堆疊的程序。您可以在這裡找到它們： • 在 CloudFormation 主控台中建立堆疊 • 在 CloudFormation 主控台中檢視堆疊資訊	2010-05-15
新增文件	2012 年 11 月 15 日	下列主題提供有關自訂資源的資訊： • 自訂資源 • AWS::CloudFormation::CustomResource • 自訂資源參考	2010-05-15

變更	版本日期	Description	API 版本
已更新的文件	2012 年 11 月 15 日	CloudFormation 現在支援指定 RDS 資料庫執行個體的每秒佈建 I/O 操作數 (IOPS)。您可以使用 AWS::RDS::DBInstance 中的 Iops 屬性，將此值設定在 1000-10,000 (以 1000 IOPS 增量)。 如需有關為 RDS 資料庫執行個體指定 IOPS 的詳細資訊，請參閱 Amazon Relational Database Service 使用者指南 中的佈建 IOPS。	2010-05-15
新的與更新的文件	2012 年 8 月 27 日	主題已重新組織，以更清楚地提供有關使用 AWS 管理主控台 和使用 CloudFormation 命令列界面 (CLI) 的特定資訊。 已新增有關標記 CloudFormation 堆疊的資訊，包括新的指南和更新的參考主題： - 新主題：設定堆疊選項。 - CloudFormation API 參考中關於標籤的新資訊：CreateStack、Stack 和 Tag。 有關使用 Windows 堆疊的新資訊： Microsoft Windows Amazon Machine Image AMIs) 和 CloudFormation 範本 引導 CloudFormation Windows 堆疊 新主題：使用 CloudFormation 範本中的規則運算式。	2010-05-15

變更	版本日期	Description	API 版本
新功能	2012 年 4 月 25 日	CloudFormation 現在透過 Amazon EC2 為虛擬私有雲端 (VPC) 安全提供全面支援。現在可以使用單一 CloudFormation 範本建立 VPC，並以所有類型的 VPC 資源 (子網路、閘道、網路 ACL 及路由表等) 填入整個 VPC。 以下資源類型的文件已更新： • AWS::EC2::SecurityGroup • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::Instance • AWS::AutoScaling::AutoScalingGroup • AWS::EC2::EIP • AWS::EC2::EIPAssociation • AWS::ElasticLoadBalancing::LoadBalancer 新的資源類型已新增到文件： • AWS::EC2::VPC • AWS::EC2::InternetGateway • AWS::EC2::DHCPOptions • AWS::EC2::DHCPOptions • AWS::EC2::RouteTable • AWS::EC2::NetworkAcl • AWS::EC2::NetworkAclEntry • AWS::EC2::Subnet • AWS::EC2::VPNGateway • AWS::EC2::CustomerGateway	2010-05-15

變更	版本日期	Description	API 版本
新功能	2012 年 4 月 13 日	CloudFormation 現在可讓您在更新堆疊時，在堆疊中新增或移除元素。已更新 CloudFormation 堆疊更新 ，並新增新的章節至逐步解說： 變更堆疊的資源 ，說明如何在更新堆疊時新增和移除資源。	2010-05-15
新功能	2012 年 2 月 2 日	CloudFormation 現在支援 Amazon Virtual Private Cloud (Amazon VPC) 中現有的資源。發佈此版本後，您可以： • 將 EC2 專用執行個體啟動至現有的 Amazon VPC。如需詳細資訊，請參閱AWS::EC2::Instance。 • 針對位於現有 Amazon VPC 中的 EC2 執行個體設定 SourceDestCheck 屬性。如需詳細資訊，請參閱AWS::EC2::Instance。 • 在現有 Amazon VPC 中建立彈性 IP 地址。如需詳細資訊，請參閱AWS::EC2::EIP。 • 使用 CloudFormation 在現有 VPC 中建立 Amazon VPC 安全群組與流入/流出規則。如需詳細資訊，請參閱AWS::EC2::SecurityGroup。 • 設定 <code>AWS::AutoScaling::AutoScalingGroup</code> 資源的 <code>VPCZoneIdentifier</code> 屬性，以將 Auto Scaling 群組與現有的 Amazon VPC 建立關聯。如需詳細資訊，請參閱AWS::AutoScaling::AutoScalingGroup。 • 將 Elastic Load Balancing 負載平衡器連接到 Amazon VPC 子網路，並建立負載平衡器的安全群組。如需詳細資訊，請參閱AWS::ElasticLoadBalancing::LoadBalancer。 • 在現有 Amazon VPC 中建立 RDS 資料庫執行個體。如需詳細資訊，請參閱AWS::RDS::DBInstance。	2010-05-15

變更	版本日期	Description	API 版本
新功能	2012 年 2 月 2 日	您想在可更新現有堆疊中以下資源的屬性： • AWS::EC2::SecurityGroupIngress • AWS::EC2::SecurityGroupEgress • AWS::EC2::EIPAssociation • AWS::RDS::DBSubnetGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSecurityGroupIngress • AWS::Route53::RecordSetGroup 如需可更新資源的完整清單以及有關更新堆疊時應考量事項的詳細資訊，請參閱 CloudFormation 堆疊更新。	2010-05-15
重組指南	2012 年 2 月 2 日	將現有章節重新整理到新的章節： 使用 CloudFormation 範本 和管理堆疊。將 範本參考 移動至目錄的頂層。將 估計 CloudFormation 堆疊的成本 移至「入門」章節。	2010-05-15
新增內容	2012 年 2 月 2 日	已新增新的章節： • 演練：更新堆疊教學，會引領您演練 LAMP 堆疊的更新程序。 • 使用 AWS CloudFormation 在 Amazon EC2 上部署應用程式描述了如何使用 CloudFormation 協助程式指令碼，使用範本中存放的中繼資料來部署應用程式。 • CloudFormation 協助程式指令碼參考提供 CloudFormation 協助程式指令碼 (cfn-init、cfn-get-metadata、cfn-signal 和 cfn-hup) 的參考資料。	2010-05-15
新功能	2011 年 5 月 26 日	CloudFormation 現在提供 <code>list-stacks</code> 命令，讓您能夠列出依堆疊狀態篩選的堆疊。已刪除的堆疊可在刪除後最多 90 天內列出。如需詳細資訊，請參閱 描述並列出您的堆疊 。	2010-05-15

變更	版本日期	Description	API 版本
新功能	2011 年 5 月 26 日	<code>describe-stack-resources</code> 和 <code>get-template</code> 命令現在可讓您從已刪除 90 天的堆疊中取得資訊。如需詳細資訊，請參閱 列出資源 和 擷取範本 。	2010-05-15
新連結	2011 年 3 月 1 日	CloudFormation 端點資訊現在位於 中 AWS 一般參考。如需詳細資訊，請前往 AWS 一般參考 中的「區域與端點」。	2010-05-15
初始版本	2011 年 2 月 25 日	CloudFormation 的初始公有版本。	2010-05-15

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。