aws

使用者指南

Amazon Managed Workflows for Apache Airflow

Amazon Managed Workflows for Apache Airflow: 使用者指南

Table of Contents

什麼是 Amazon 管理的 Apache 氣流工作流程？

適用於 Apache Airflow 的 Amazon 受管工作流程是 Apache Airflow 的受管協調服務，您可以使用它在雲端中大規模設定和操作資料管道。Apache Airflow 是一種開放原始碼工具，用於以程式設計方式撰寫、排程和監視稱為工作流程的程序和工作序列。使用 Amazon MWAA，您可以使用 Apache Airflow 和 Python 建立工作流程，而不必管理基礎設施以提高可擴展性、可用性和安全性。Amazon MWAA 可自動擴展其工作流程執行容量以滿足您的需求，Amazon MWAA 與 AWS 安全服務整合，協助您快速安全地存取資料。

內容

功能

- 自動氣流設定 — 當您建立 Amazon MWAA 環境時，選擇 Apache 氣流版本，即可快速設定 Apache 氣流。Amazon MWAA 使用可在網際網路上下載的相同 Apache 氣流使用者介面和開放原始碼，為您設定 Apache 氣流。
- 自動調整規模 — 設定環境中執行的工作者數目下限和上限，以自動調整 Apache Airflow Worker 的規模。Amazon MWAA 會監控您環境中的 Worker，並使用其自動調度資源元件新增 Worker 以滿足需求，直到達到達到您定義的最大 Worker 數為止。
- 內建驗證 — 透過在 AWS Identity and Access Management (IAM) 中定義存取控制政策，為您的 Apache Airflow Web 伺服器啟用角色型驗證和授權。Apache 氣流工作者會採用這些政策來安全存取 AWS 服務。
- 內建安全性 — Apache 氣流工作者和排程器在 Amazon MWAA 的 Amazon VPC 中執行。數據也會使用自動加密 AWS Key Management Service，因此默認情況下您的環境是安全的。
- 公開或私人存取模式 — 使用私人或公開存取模式存取您的 Apache Airflow 網頁伺服器。公用網路存取模式會針對可透過網際網路存取的 Apache Airflow 網頁伺服器使用 VPC 擬私人雲端端點。私人網路存取模式會將 VPC 端點用於您的虛擬私人雲端中存取的 Apache 氣流網頁伺服器。在這兩種情況下，Apache Airflow 使用者的存取權限都是由您在 AWS Identity and Access Management (IAM) 和 AWS SSO 中定義的存取控制政策所控制。

- 簡化升級和修補程式 — Amazon MWAA 會定期提供新版本的 Apache 氣流。Amazon MWAA 團隊將會更新和修補這些版本的映像檔。
- 工作流程監控 — 在 Amazon 中查看 Apache 氣流日誌和 Apache 氣流指標， CloudWatch 以識別 Apache 氣流任務延遲或工作流程錯誤，而無需使用其他第三方工具。Amazon MWAA 會自動將環境指標 (若已啟用) 傳送給 Apache 氣流記錄。 CloudWatch
- AWS 整合 — Amazon MVAA 支援與亞 Amazon Athena、Amazon、Amazon DynamoDB AWS Batch CloudWatch、Amazon EMR、Amazon EKS AWS DataSync、Amazon 資料 Firehose、亞馬遜 Redshift AWS Fargate、Amazon SQS、Amazon SNS、 AWS Glue亞馬遜和 Amazon S3 的開放原始碼整合，以及數百個內建和社群建立的操作員和感測器。 AWS Lambda SageMaker
- 工作者叢集 — Amazon MWAA 為使用容器提供支援，讓使用 Amazon ECS 隨需擴展工作者叢集，並減少排程器中斷情況。 AWS Fargate支援在 Amazon ECS 容器上叫用任務的操作員，以及在 Kubernetes 叢集上建立和執行網繭的 Kubernetes 操作員。

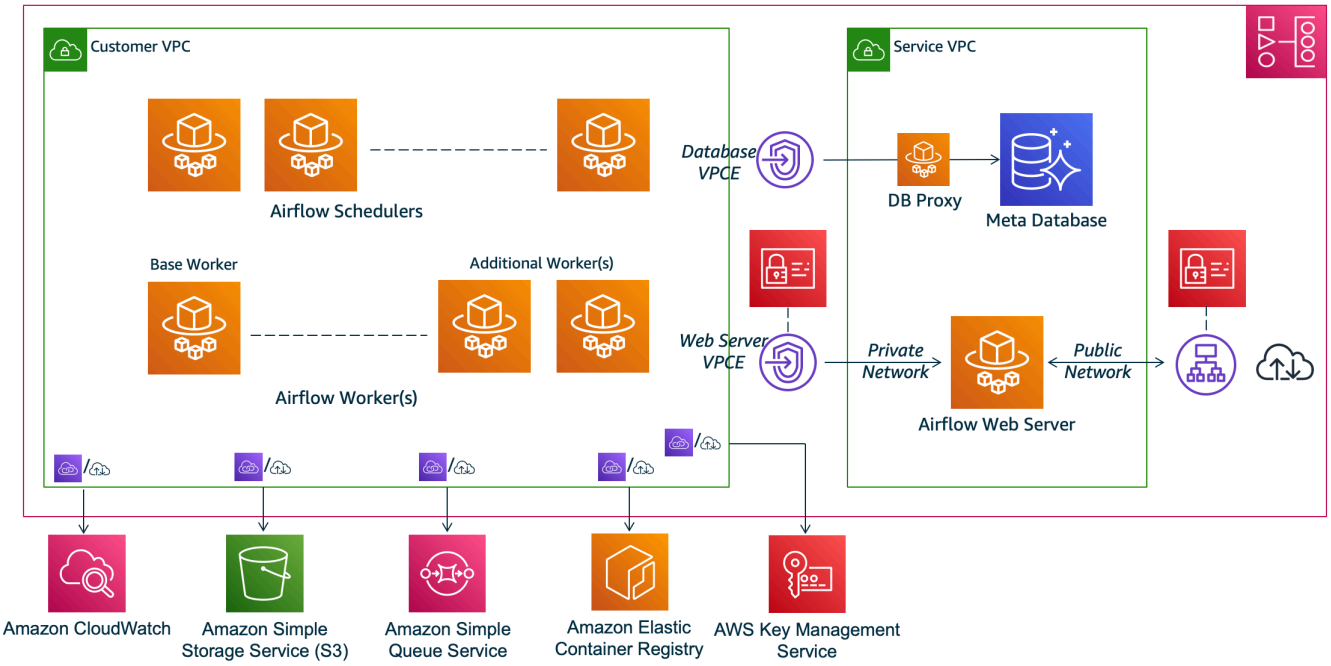
架構

外盒中包含的所有元件 (如下圖所示) 會在您的帳戶中顯示為單一 Amazon MWAA 環境。Apache 氣流排程 AWS Fargate (Fargate) 器和工作者是連線到 Amazon VPC 中針對您環境之私有子網路的容器。每個環境都有自己的 Apache Airflow 中繼資料庫，由 AWS 排程器和工作者 Fargate 容器透過私有安全的 VPC 端點存取。

Amazon CloudWatch、Amazon S3、Amazon SQS、Amazon ECR 和與 AWS KMS Amazon MWAA 分開存取，需要從 Apache 氣流排程器和 Fargate 容器中的工作者進行存取。

您可以選取公用網路 Apache 氣流存取模式，透過網際網路存取 Apache 氣流存取模式，或選取私人網路 Apache 氣流存取模 VPC 來存取 Apache 氣流網頁伺服器。在這兩種情況下，Apache Airflow 使用者的存取權限都是由您在 AWS Identity and Access Management (IAM) 中定義的存取控制政策所控制。

> **Note**
>
> 多個 Apache 氣流排程器僅適用於 Apache 氣流 v2 及以上版本。若要深入瞭解 Apache 氣流工作生命週期，請參閱 Apache 氣流參考指南中的概念。



整合

活躍且不斷增長的 Apache Airflow 開放原始碼社群為操作員 (簡化服務連線的外掛程式) 提供 Apache Airflow 與 AWS 服務整合。這包括 Amazon S3，亞馬 Amazon Redshift，Amazon EMR 和 Amazon 等服務 AWS Batch，以及其他雲平台上的服務。 SageMaker

搭配 Amazon MWAA 使用 Apache 氣流，可完全支援與 AWS 服務和熱門第三方工具 (例如 Apache Hadoop、普雷斯托、蜂巢和星火) 整合，以執行資料處理任務。Amazon MWAA 致力於維持與 Amazon MWAA API 的相容性，而 Amazon MWAA 打算提供可靠的 AWS 服務整合，並將其提供給社群，並參與社群功能開發。

如需程式碼範例，請參閱 Amazon Managed Workflows。

支援的版本

Amazon MWAA 支持阿帕奇氣流的多個版本。如需有關我們支援的 Apache 氣流版本以及每個版本隨附的 Apache 氣流元件的詳細資訊，請參閱適用於 Apache 氣流的 Amazon 氣流管理工作流程。

後續步驟？

- 開始使用為氣流 DAG 和支援檔案建立 Amazon S3 儲存貯體的單一 AWS CloudFormation 範本、具有公有路由的 Amazon VPC，以及中的 Amazon MWAA 環境。Amazon Apache 氣流管理工作流程的快速入門教學
- 透過為您的氣流 DAG 和支援檔案建立 Amazon S3 儲存貯體，從三個 Amazon VPC 聯網選項中選擇其中一個，然後在中建立 Amazon MWAA 環境，逐步開始使用。開始使用 Amazon Managed Workflows

Amazon Apache 氣流管理工作流程的快速入門教學

本快速入門教學使用可同時建立 Amazon VPC 基礎設施的 AWS CloudFormation 範本、一個含有dags資料夾的 Amazon S3 儲存貯體，以及適用於 Apache 氣流環境的 Amazon 受管工作流程。

主題

於本教學課程中

本教學會逐步引導您完成三個 AWS Command Line Interface (AWS CLI) 命令，以便將 DAG 上傳至 Amazon S3、在 Apache 氣流中執行 DAG，以及檢視日誌 CloudWatch。最後，我們會逐步引導您為 Apache 氣流開發團隊建立 IAM 政策的步驟。

> **Note**
>
> 此頁面上的 AWS CloudFormation 範本會針對中提供的最新版本的 Apache 氣流環境，建立適用於 Apache 氣流環境的 Amazon 受管工作流程 AWS CloudFormation。可用的最新版本是阿帕奇氣流 v2.8.1。

此頁面上的 AWS CloudFormation 範本會建立下列項目：

- VPC 基礎架構。範本使用網際網路上的公用路由。它使用中公共網路存取模式的 Apache 氣流網頁伺服器`WebserverAccessMode: PUBLIC_ONLY`。
- Amazon S3 桶。該模板創建一個帶有dags文件夾的 Amazon S3 存儲桶。它配置為阻止所有公共訪問，並啟用了存儲桶版本控制，如中的定義為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯。

- Amazon MWAA 環境。該範本會建立與 Amazon S3 儲存貯體上dags資料夾相關聯的 Amazon MWAA 環境、具有 Amazon MWAA 所使用 AWS 服務的許可執行角色，以及使用AWS 擁有金鑰加密的預設角色 (如中所定義)。建立一個 Amazon MWAA 環境
- CloudWatch 記錄檔。範本會針對氣流排程器記錄群組、Airflow Web 伺服器記錄群組、Airflow 工作者記錄群組、Airflow 工作者記錄群組、Airflow DAG 處理記錄群組，以及氣流工作記錄群組 (如中所定義) 啟用 Apache Airflow 登入在 Amazon 中查看氣流日誌 CloudWatch。 CloudWatch

在本教學課程中，您將完成下列工作：

- 上傳並執行 DAG。將 Apache 氣流的教程 DAG 上傳最新的 Amazon MWAA 支持 Apache 氣流版本到 Amazon S3，然後在 Apache 氣流用戶界面中運行，如中所定義。新增或更新 DAG
- 檢視記錄檔。在記錄檔中檢視 Airflow Web 伺服器 CloudWatch 記錄群組，如中所定義在 Amazon 中查看氣流日誌 CloudWatch。
- 建立存取控制原則。根據中的定義，在 IAM 中為您的 Apache 氣流開發團隊建立存取控制政策存取 Amazon MWAA 環境。

必要條件

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。若要完成此頁面上的步驟，您需要下列項目：

- AWS CLI — 安裝版本 2.
- AWS CLI — 快速配置 `aws configure`.

步驟 1：將 AWS CloudFormation 範本儲存在本機

- 複製以下範本的內容並在本機儲存為`mwaa_public_network.yml`。您也可以下載模板。

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

Parameters:

  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String
    Default: MWAAEnvironment
```

```
 VpcCIDR:
   Description: The IP range (CIDR notation) for this VPC
   Type: String
   Default: 10.192.0.0/16

 PublicSubnet1CIDR:
   Description: The IP range (CIDR notation) for the public subnet in the first
Availability Zone
   Type: String
   Default: 10.192.10.0/24

 PublicSubnet2CIDR:
   Description: The IP range (CIDR notation) for the public subnet in the second
Availability Zone
   Type: String
   Default: 10.192.11.0/24

 PrivateSubnet1CIDR:
   Description: The IP range (CIDR notation) for the private subnet in the first
Availability Zone
   Type: String
   Default: 10.192.20.0/24
 PrivateSubnet2CIDR:
   Description: The IP range (CIDR notation) for the private subnet in the second
Availability Zone
   Type: String
   Default: 10.192.21.0/24
 MaxWorkerNodes:
   Description: The maximum number of workers that can run in the environment
   Type: Number
   Default: 2
 DagProcessingLogs:
   Description: Log level for DagProcessing
   Type: String
   Default: INFO
 SchedulerLogsLevel:
   Description: Log level for SchedulerLogs
   Type: String
   Default: INFO
 TaskLogsLevel:
   Description: Log level for TaskLogs
   Type: String
   Default: INFO
 WorkerLogsLevel:
```

```
    Description: Log level for WorkerLogs
    Type: String
    Default: INFO
  WebserverLogsLevel:
    Description: Log level for WebserverLogs
    Type: String
    Default: INFO

Resources:

 ###############################################################################################
  # CREATE VPC

 ###############################################################################################

  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: MWAAEnvironment

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: MWAAEnvironment

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      CidrBlock: !Ref PublicSubnet1CIDR
```

```
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

  PublicSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

  PrivateSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet1CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

  PrivateSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet2CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ2)

  NatGateway1EIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc
```

```
  NatGateway2EIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc

  NatGateway1:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway1EIP.AllocationId
      SubnetId: !Ref PublicSubnet1

  NatGateway2:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway2EIP.AllocationId
      SubnetId: !Ref PublicSubnet2

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Routes

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet1

  PublicSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
```

```
      SubnetId: !Ref PublicSubnet2


  PrivateRouteTable1:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ1)

  DefaultPrivateRoute1:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway1

  PrivateSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      SubnetId: !Ref PrivateSubnet1

  PrivateRouteTable2:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ2)

  DefaultPrivateRoute2:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway2

  PrivateSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      SubnetId: !Ref PrivateSubnet2
```

```
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: "mwaa-security-group"
      GroupDescription: "Security group with a self-referencing inbound rule."
      VpcId: !Ref VPC

  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: "-1"
      SourceSecurityGroupId: !Ref SecurityGroup

  EnvironmentBucket:
    Type: AWS::S3::Bucket
    Properties:
      VersioningConfiguration:
        Status: Enabled
      PublicAccessBlockConfiguration:
        BlockPublicAcls: true
        BlockPublicPolicy: true
        IgnorePublicAcls: true
        RestrictPublicBuckets: true


##################################################################################################
  # CREATE MWAA

##################################################################################################

  MwaaEnvironment:
    Type: AWS::MWAA::Environment
    DependsOn: MwaaExecutionPolicy
    Properties:
      Name: !Sub "${AWS::StackName}-MwaaEnvironment"
      SourceBucketArn: !GetAtt EnvironmentBucket.Arn
      ExecutionRoleArn: !GetAtt MwaaExecutionRole.Arn
      DagS3Path: dags
      NetworkConfiguration:
        SecurityGroupIds:
          - !GetAtt SecurityGroup.GroupId
        SubnetIds:
```

```
          - !Ref PrivateSubnet1
          - !Ref PrivateSubnet2
      WebserverAccessMode: PUBLIC_ONLY
      MaxWorkers: !Ref MaxWorkerNodes
      LoggingConfiguration:
        DagProcessingLogs:
          LogLevel: !Ref DagProcessingLogs
          Enabled: true
        SchedulerLogs:
          LogLevel: !Ref SchedulerLogsLevel
          Enabled: true
        TaskLogs:
          LogLevel: !Ref TaskLogsLevel
          Enabled: true
        WorkerLogs:
          LogLevel: !Ref WorkerLogsLevel
          Enabled: true
        WebserverLogs:
          LogLevel: !Ref WebserverLogsLevel
          Enabled: true
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId: !Ref VPC
      GroupDescription: !Sub "Security Group for Amazon MWAA Environment
${AWS::StackName}-MwaaEnvironment"
      GroupName: !Sub "airflow-security-group-${AWS::StackName}-MwaaEnvironment"

  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: "-1"
      SourceSecurityGroupId: !Ref SecurityGroup

  SecurityGroupEgress:
    Type: AWS::EC2::SecurityGroupEgress
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: "-1"
      CidrIp: "0.0.0.0/0"

  MwaaExecutionRole:
    Type: AWS::IAM::Role
```

```
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - airflow-env.amazonaws.com
                - airflow.amazonaws.com
            Action:
             - "sts:AssumeRole"
      Path: "/service-role/"

  MwaaExecutionPolicy:
    DependsOn: EnvironmentBucket
    Type: AWS::IAM::ManagedPolicy
    Properties:
      Roles:
        - !Ref MwaaExecutionRole
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action: airflow:PublishMetrics
            Resource:
              - !Sub "arn:aws:airflow:${AWS::Region}:${AWS::AccountId}:environment/
${EnvironmentName}"
          - Effect: Deny
            Action: s3:ListAllMyBuckets
            Resource:
              - !Sub "${EnvironmentBucket.Arn}"
              - !Sub "${EnvironmentBucket.Arn}/*"

          - Effect: Allow
            Action:
              - "s3:GetObject*"
              - "s3:GetBucket*"
              - "s3:List*"
            Resource:
              - !Sub "${EnvironmentBucket.Arn}"
              - !Sub "${EnvironmentBucket.Arn}/*"
          - Effect: Allow
            Action:
              - logs:DescribeLogGroups
```

```
            Resource: "*"

          - Effect: Allow
            Action:
              - logs:CreateLogStream
              - logs:CreateLogGroup
              - logs:PutLogEvents
              - logs:GetLogEvents
              - logs:GetLogRecord
              - logs:GetLogGroupFields
              - logs:GetQueryResults
              - logs:DescribeLogGroups
            Resource:
              - !Sub "arn:aws:logs:${AWS::Region}:${AWS::AccountId}:log-
group:airflow-${AWS::StackName}*"
          - Effect: Allow
            Action: cloudwatch:PutMetricData
            Resource: "*"
          - Effect: Allow
            Action:
              - sqs:ChangeMessageVisibility
              - sqs:DeleteMessage
              - sqs:GetQueueAttributes
              - sqs:GetQueueUrl
              - sqs:ReceiveMessage
              - sqs:SendMessage
            Resource:
              - !Sub "arn:aws:sqs:${AWS::Region}:*:airflow-celery-*"
          - Effect: Allow
            Action:
              - kms:Decrypt
              - kms:DescribeKey
              - "kms:GenerateDataKey*"
              - kms:Encrypt
            NotResource: !Sub "arn:aws:kms:*:${AWS::AccountId}:key/*"
            Condition:
              StringLike:
                "kms:ViaService":
                  - !Sub "sqs.${AWS::Region}.amazonaws.com"
Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC
```

```
  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1

  PublicSubnet2:
    Description: A reference to the public subnet in the 2nd Availability Zone
    Value: !Ref PublicSubnet2

  PrivateSubnet1:
    Description: A reference to the private subnet in the 1st Availability Zone
    Value: !Ref PrivateSubnet1

  PrivateSubnet2:
    Description: A reference to the private subnet in the 2nd Availability Zone
    Value: !Ref PrivateSubnet2

  SecurityGroupIngress:
    Description: Security group with self-referencing inbound rule
    Value: !Ref SecurityGroupIngress

  MwaaApacheAirflowUI:
    Description: MWAA Environment
    Value: !Sub  "https://${MwaaEnvironment.WebserverUrl}"
```

第二步：使用 AWS CLI

1. 在命令提示符中，導航到存儲mwaa_public_network.yml的目錄。例如：

```
cd mwaaproject
```

2. 使用aws cloudformation create-stack指令建立使用的堆疊 AWS CLI。

```
aws cloudformation create-stack --stack-name mwaa-environment-public-network --
template-body file://mwaa_public_network.yml --capabilities CAPABILITY_IAM
```

> **Note**
>
> 建立 Amazon VPC 基礎設施、Amazon S3 儲存貯體和亞馬遜 MWAA 環境需要 30 分鐘以上的時間。

第三步：將 DAG 上傳到 Amazon S3 並在 Apache 氣流 UI 中運行

1. 複製最新支援 Apache 氣流版本的tutorial.py檔案內容，並在本機儲存為tutorial.py。
2. 在命令提示符中，導航到存儲tutorial.py的目錄。例如：

```
cd mwaaproject
```

3. 使用下列命令列出所有 Amazon S3 儲存貯體。

```
aws s3 ls
```

4. 使用下列命令列出您環境之 Amazon S3 儲存貯體中的檔案和資料夾。

```
aws s3 ls s3://YOUR_S3_BUCKET_NAME
```

5. 使用下面的腳本將tutorial.py文件上傳到您的文件dags夾。取代「*## _ S3_* 桶名稱」中的樣本值。

```
aws s3 cp tutorial.py s3://YOUR_S3_BUCKET_NAME/dags/
```

6. 在 Amazon MWAA 主控台上開啟「環境」頁面。
7. 選擇一個環境。
8. 選擇「開啟氣流 UI」。
9. 在 Apache 氣流使用者介面上，從可用的 DAG 清單中選擇教學課程 DAG。
10. 在 DAG 詳細資料頁面上，選擇 DAG 名稱旁的暫停/取消暫停 DAG 切換，以取消暫停 DAG。
11. 選擇 [觸發 DAG]。

步驟四：在記錄檔中檢視 CloudWatch 記錄

您可以在 CloudWatch 主控台中檢視 AWS CloudFormation 堆疊啟用的所有 Apache 氣流記錄檔的 Apache 氣流記錄檔。下節說明如何檢視 Airflow Web 伺服器記錄群組的記錄檔。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [監視] 窗格中選擇 Airflow Web 伺服器記錄群組。
4. 在「webserver_console_ip記錄資料流」中選擇記錄檔。

後續步驟？

- 進一步瞭解如何上傳 DAG、在中指定 Python 相依性，以requirements.txt及在中的自訂外掛程式中指定plugins.zip。在亞馬遜 MWAA 與 DAG 工作
- 進一步了解我們建議您調整環境效能的最佳實務Amazon MWAA 上阿帕奇氣流的性能調整。
- 在中為您的環境建立監視儀表板監控 Amazon MWAA 上的儀表板和警報。
- 在中執行一些 DAG 程式碼範例Amazon Managed Workflows。

開始使用 Amazon Managed Workflows

適用於 Apache 氣流的亞馬遜受管工作流程使用 Amazon VPC、DAG 程式碼和 Amazon S3 儲存貯體中的支援檔案來建立環境。本指南說明開始使用 Amazon MWAA 所需的先決條件和必要AWS資源。

主題

先決條件

若要建立 Amazon MWAA 環境，您可能需要採取其他步驟來確保擁有建立所需AWS資源的許可。

- AWS帳戶 — 有權使用 Amazon MWAA 以及您環境所使用之AWS服務和資源的AWS帳戶。

關於本指南

本節說明您將在本指南中建立的AWS基礎結構和資源。

- 亞馬遜 VPC — 亞馬遜 MWAA 環境所需的亞馬遜 VPC 聯網元件。您可以設定符合這些需求 (進階) 的現有 VPC (如中所示)關於 Amazon MWAA 上的聯網，或建立 VPC 和網路元件 (如中所定義)the section called “建立虛擬私人雲端網路”。
- Amazon S3 儲存貯體 — 用於存放 DAG 和相關檔案的 Amazon S3 儲存貯體，例如`plugins.zip`和`requirements.txt`。您的 Amazon S3 儲存貯體必須設定為封鎖所有公用存取，並啟用儲存貯體版本控制，如中所定義為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯。

- Amazon MWAA 環境 — 一個 Amazon MWAA 環境，設定了 Amazon S3 儲存貯體的位置、DAG 程式碼和任何自訂外掛程式或 Python 相依性的路徑，以及您的 Amazon VPC 及其安全群組 (如中所定義)建立一個 Amazon MWAA 環境。

開始之前

若要建立 Amazon MWAA 環境，您可能需要在建立環境之前採取其他步驟來建立和設定其他AWS資源。

若要建立環境，您必須準備好以下事項：

- AWS KMSkey — 在您的環境中進行資料加密的AWS KMS金鑰。您可以在 Amazon MWAA 主控台上選擇預設選項，以在建立環境時建立AWS擁有的金鑰，或指定現有的客戶受管金鑰，該金鑰具有環境所使用之其他AWS服務 (進階) 的許可。如需進一步了解，請參閱 使用客戶管理的金鑰進行加密。
- 執行角色 — 允許 Amazon MWAA 存取您環境中AWS資源的執行角色。您可以在 Amazon MWAA 主控台上選擇預設選項，以便在建立環境時建立執行角色。如需進一步了解，請參閱 Amazon MWAA 執行角色。
- VPC 安全群組 — VPC 安全群組，可讓 Amazon MWAA 存取 VPC 網路中的其他AWS資源。您可以在 Amazon MWAA 主控台上選擇預設選項，以在建立環境時建立安全群組，或為安全群組提供適當的輸入和輸出規則 (進階)。如需進一步了解，請參閱 Amazon MWAA 上 VPC 的安全政策。

可用地區

以下AWS區域提供 Amazon MWAA。

- 歐洲 (斯德哥爾摩) — eu-north-1
- 歐洲 (法蘭克福) — eu-central-1
- 歐洲 (愛爾蘭) — eu-west-1
- 歐洲 (倫敦) — eu-west-2
- 歐洲 (巴黎) — eu-west-3
- 亞太區域 (孟買) — ap-south-1
- 亞太區域 (新加坡) —ap-southeast-1
- 亞太區域 (雪梨) – ap-southeast-2
- 亞太區域 (東京) – ap-northeast-1

- 亞太區域 (首爾) — ap-northeast-2
- 美國東部 (維吉尼亞北部) – us-east-1
- 美國東部 (俄亥俄) - us-east-2
- 美國西部 (奧勒岡) - us-west-2
- 加拿大 (中部) — ca-central-1
- 南美洲 (聖保羅) — sa-east-1

為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯

本指南說明建立 Amazon S3 儲存貯體以將 Apache 氣流定向無環圖 (DAG)、檔案中的自訂外掛程式以及 Python 相依性存放在`plugins.zip`檔案中的步驟。`requirements.txt`

內容

開始之前

- Amazon S3 儲存貯貯貯存貯貯貯存貯貯貯存貯貯貯存貯貯貯存貯貯貯存貯貯貯 若要進一步了解，請參閱 Amazon 簡單儲存服務使用者指南中的儲存貯體命名規則。
- 用於 Amazon MWAA 環境的 Amazon S3 儲存貯體必須設定為封鎖所有公用存取，並啟用儲存貯體版本控制。
- 用於亞馬遜 MWAA 環境的 Amazon S3 儲存貯體必須位於與亞馬遜 MWAA 環境相同的AWS區域。若要檢視 Amazon MWAA 的AWS區域清單，請參閱中的亞馬遜 MWAA 端點和配額。AWS 一般參考

儲存貯存貯存貯

本節說明為您的環境建立 Amazon S3 儲存貯存貯存貯存貯存貯存貯存貯存貯存貯存貯

建立儲存貯體

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。

2. 選擇 Create bucket (建立儲存貯體)。
3. 在 Bucket name (儲存貯體名稱) 中，為儲存貯體輸入符合 DNS 規範的名稱。

   儲存貯體名稱必須；

   - 在所有 Amazon S3 中都為唯一。
   - 長度必須介於 3 與 63 個字元之間。
   - 不含大寫字元。
   - 以小寫字母或數字開頭。

   > **Important**
   >
   > 避免在儲存貯體名稱中包含敏感資訊，例如帳戶號碼。在指向儲存貯體中之物件的 URL 中，會顯示儲存貯體名稱。

4. 在「AWS區域」中選擇「地區」。這個區域必須與您的 Amazon MWAA 環境所在的AWS區域相同。

   - 建議您選擇接近您的區域以充分降低延遲及成本，並因應法規要求。

5. 選擇 Block all public access (封鎖所有公用存取)。
6. 選擇「在值區版本管理中啟用
7. 選用-標籤。新增鍵值標籤配對，以在標籤中識別您的 Amazon S3 儲存貯體。例如`Bucket：Staging`。
8. 選用-伺服器端加密。您可以選擇在 Amazon S3 儲存貯體上啟用下列其中一個加密選項。

   a. 在伺 Amazon S3 SSE-S3

   b. 選擇金AWS Key Management Service鑰 (SSE-KMS) 以在 Amazon S3 儲存貯體上使用金AWS KMS鑰進行加密：

      i. AWS受管金鑰 (aws/s3)-如果您選擇此選項，您可以使用 Amazon MWAA 管理的AWS擁有金鑰，或指定客戶受管金鑰來加密 Amazon MWAA 環境。

      ii. 從AWS KMS金鑰中選擇或輸入金AWS KMS鑰 ARN-如果您選擇在此步驟中指定客戶管理的金鑰，則必須指定金AWS KMS鑰 ID 或 ARN。 AWS KMSAmazon MWAA 不支援別名和多區域金鑰。您指定的AWS KMS金鑰也必須用於 Amazon MWAA 環境上的加密。

9. 選用-進階設定。如果您要啟用 Amazon S3 物件鎖定：

a. 選擇進階設定、啟用。

> **Important**
>
> 啟用物件鎖定將永久允許鎖定此值區中的物件。如需詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的使用 Amazon S3 的物件鎖定功能鎖定物件。

b. 選擇確認。

10. 選擇 建立儲存貯體 。

後續步驟？

- 了解如何在建立虛擬私人雲端網路中為環境建立所需的 Amazon 虛擬私人雲端網路。
- 在如何設定 ACL 值區權限？中了解如何管理存取權限？
- 如何刪除 S3 儲存貯貯存貯存貯存貯存貯存貯存貯存貯存貯存貯存貯存貯存貯存貯 。

建立虛擬私人雲端網路

適用於 Apache 氣流的 Amazon 受管工作流程需要 Amazon VPC 和特定聯網元件才能支援某個環境。本指南說明為 Apache 氣流環境的 Amazon 受管工作流程建立 Amazon 虛擬私人雲端網路的不同選項。

> **Note**
>
> Apache 氣流在低延遲的網路環境中發揮最佳效果。如果您使用的是將流量路由到其他區域或內部部署環境的現有 Amazon VPC，我們建議您新增AWS PrivateLink亞馬遜 SQS 的端點 CloudWatch, 亞馬遜 S3,AWS KMS和亞馬遜 ECR。如需有關設定的詳細資訊AWS PrivateLink 對於亞馬遜 MWAA，請參閱建立沒有網際網路存取權的 Amazon VPC 網路。

內容

先決條件

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面上的步驟，您需要下列項目：

- AWS CLI— 安裝版本 2。
- AWS CLI-快速配置`aws configure`。

開始之前

- 該虛私網在建立環境之後，您為環境指定的無法變更。
- 您可以為您的亞馬遜 VPC 和 Apache 氣流使用私有或公共路由網頁伺服器。若要檢視選項清單，請參閱the section called “Amazon VPC 和 Apache 氣流存取模式的範例使用案例”。

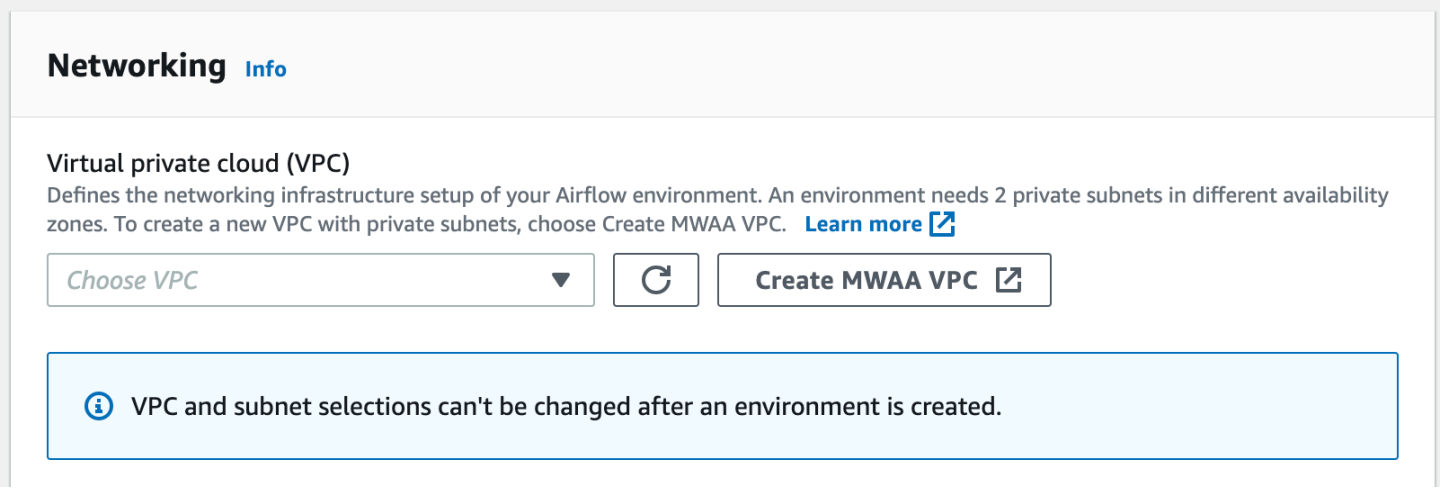
建立亞馬遜虛擬私人雲端網路的選項

下節說明為某個環境建立 Amazon 虛擬私人雲端網路的可用選項。

選項一：在亞馬遜 MWAA 主控台上建立虛擬私人雲端網路

以下部分說明如何在亞馬遜 MWAA 主控台上建立亞馬遜虛擬私人雲端網路。此選項使用網際網路上的公用路由。它可以用於一個阿帕奇氣流網頁伺服器與私人網路或者大眾網存取模式。

下圖顯示了您可以在哪裡找到建立虛擬私人雲端亞馬遜 MWAA 控制台上的按鈕。



選項二：建立亞馬遜虛擬私人雲端網路與互聯網接入

以下AWS CloudFormation範本建立一個亞馬遜 VPC 網路與互聯網接入在你的默認AWS區域。此選項使用網際網路上的公用路由。這個模板可用於一個 Apache 氣流網頁伺服器與私人網路或者大眾網存取模式。

1. 複製以下範本的內容並在本機儲存為`cfn-vpc-public-private.yaml`。你也可以下載範本。

```
Description:  This template deploys a VPC, with a pair of public and private
 subnets spread
  across two Availability Zones. It deploys an internet gateway, with a default
  route on the public subnets. It deploys a pair of NAT gateways (one in each AZ),
  and default routes for them in the private subnets.

Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String
    Default: mwaa-

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in
 the first Availability Zone
    Type: String
    Default: 10.192.10.0/24
```

```
  PublicSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet in
 the second Availability Zone
    Type: String
    Default: 10.192.11.0/24

  PrivateSubnet1CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet
 in the first Availability Zone
    Type: String
    Default: 10.192.20.0/24

  PrivateSubnet2CIDR:
    Description: Please enter the IP range (CIDR notation) for the private subnet
 in the second Availability Zone
    Type: String
    Default: 10.192.21.0/24

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnet1:
```

```
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs '' ]
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ1)

  PublicSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Subnet (AZ2)

  PrivateSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet1CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ1)

  PrivateSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet2CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Subnet (AZ2)

  NatGateway1EIP:
```

```
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc

  NatGateway2EIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc

  NatGateway1:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway1EIP.AllocationId
      SubnetId: !Ref PublicSubnet1

  NatGateway2:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGateway2EIP.AllocationId
      SubnetId: !Ref PublicSubnet2

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Public Routes

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet1
```

```
  PublicSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet2


  PrivateRouteTable1:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ1)

  DefaultPrivateRoute1:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway1

  PrivateSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      SubnetId: !Ref PrivateSubnet1

  PrivateRouteTable2:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub ${EnvironmentName} Private Routes (AZ2)

  DefaultPrivateRoute2:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway2
```

```
  PrivateSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      SubnetId: !Ref PrivateSubnet2

  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: "mwaa-security-group"
      GroupDescription: "Security group with a self-referencing inbound rule."
      VpcId: !Ref VPC

  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: "-1"
      SourceSecurityGroupId: !Ref SecurityGroup

Outputs:
  VPC:
    Description: A reference to the created VPC
    Value: !Ref VPC

  PublicSubnets:
    Description: A list of the public subnets
    Value: !Join [ ",", [ !Ref PublicSubnet1, !Ref PublicSubnet2 ]]

  PrivateSubnets:
    Description: A list of the private subnets
    Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

  PublicSubnet1:
    Description: A reference to the public subnet in the 1st Availability Zone
    Value: !Ref PublicSubnet1

  PublicSubnet2:
    Description: A reference to the public subnet in the 2nd Availability Zone
    Value: !Ref PublicSubnet2

  PrivateSubnet1:
    Description: A reference to the private subnet in the 1st Availability Zone
    Value: !Ref PrivateSubnet1
```

```
  PrivateSubnet2:
    Description: A reference to the private subnet in the 2nd Availability Zone
    Value: !Ref PrivateSubnet2

  SecurityGroupIngress:
    Description: Security group with self-referencing inbound rule
    Value: !Ref SecurityGroupIngress
```

2. 在命令提示符中，導航到目錄`cfn-vpc-public-private.yaml`被存儲。例如：

```
cd mwaaproject
```

3. 使用[aws cloudformation create-stack](#)指令，以建立使用AWS CLI。

```
aws cloudformation create-stack --stack-name mwaa-environment --template-body
 file://cfn-vpc-public-private.yaml
```

> **Note**
>
> 建立 Amazon VPC 基礎設施大約需要 30 分鐘。

選項三：建立亞馬遜虛擬私人雲端網路無互聯網接入

以下AWS CloudFormation範本建立一個亞馬遜 VPC 網路沒有互聯網接入在你的默認AWS區域。

> **Important**
>
> 在沒有網際網路存取權的情況下使用 Amazon VPC 時，您必須授與權限給 Amazon ECR，才能使用閘道端點存取 Amazon S3。您可以執行下列動作來建立閘道端點：
>
> 1. 複製以下內容JSONIAM 政策，並將其儲存為本機`s3-gw-endpoint-policy.json`。該政策授予 Amazon ECR 存取 Amazon S3 資源所需的最低權限。
>
> ```
> {
>   "Statement": [
>     {
>       "Sid": "Access-to-specific-bucket-only",
>       "Principal": "*",
>       "Action": [
> ```

```
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": ["arn:aws:s3:::prod-region-starport-layer-bucket/*"]
      }
    ]
}
```

2. 使用以下命令建立端點AWS CLI指令。取代下列項目的值`--vpc-id`和`--route-table-ids`與您的亞馬遜 VPC 的信息。取代`--service-name`根據您所在地區的名稱。

```
$ aws ec2 create-vpc-endpoint --vpc-id vpc-1a2b3c4d \
--service-name com.amazonaws.us-west-2.s3 \
--route-table-ids rtb-11aa22bb \
--vpc-endpoint-type Gateway \
--policy-document file://s3-gw-endpoint-policy.json
```

如需為 Amazon ECR 建立 Amazon S3 閘道端點的詳細資訊，請參閱建立亞馬遜 S3 閘道端點在亞馬遜彈性容器註冊用戶指南。

此選項使用沒有網際網路存取的私人。這個模板可用於一個 Apache 氣流網頁伺服器與私人網路僅限存取模式。它創建了所需的虛擬私人雲端端點AWS環境使用的服務。

1. 複製以下範本的內容並在本機儲存為`cfn-vpc-private.yaml`。你也可以下載範本。

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

Parameters:
   VpcCIDR:
     Description: The IP range (CIDR notation) for this VPC
     Type: String
     Default: 10.192.0.0/16

   PrivateSubnet1CIDR:
     Description: The IP range (CIDR notation) for the private subnet in the first
 Availability Zone
     Type: String
     Default: 10.192.10.0/24

   PrivateSubnet2CIDR:
```

```
    Description: The IP range (CIDR notation) for the private subnet in the second
 Availability Zone
    Type: String
    Default: 10.192.11.0/24

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
       - Key: Name
         Value: !Ref AWS::StackName

  RouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPC
      Tags:
       - Key: Name
         Value: !Sub "${AWS::StackName}-route-table"

  PrivateSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 0, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet1CIDR
      MapPublicIpOnLaunch: false
      Tags:
       - Key: Name
         Value: !Sub "${AWS::StackName} Private Subnet (AZ1)"

  PrivateSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select [ 1, !GetAZs  '' ]
      CidrBlock: !Ref PrivateSubnet2CIDR
      MapPublicIpOnLaunch: false
      Tags:
       - Key: Name
```

```
          Value: !Sub "${AWS::StackName} Private Subnet (AZ2)"

  PrivateSubnet1RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref PrivateSubnet1

  PrivateSubnet2RouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref RouteTable
      SubnetId: !Ref PrivateSubnet2

  S3VpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.s3"
      VpcEndpointType: Gateway
      VpcId: !Ref VPC
      RouteTableIds:
       - !Ref RouteTable

  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId: !Ref VPC
      GroupDescription: Security Group for Amazon MWAA Environments to access VPC
endpoints
      GroupName: !Sub "${AWS::StackName}-mwaa-vpc-endpoints"

  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: "-1"
      SourceSecurityGroupId: !Ref SecurityGroup

  SqsVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.sqs"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
```

```
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  CloudWatchLogsVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.logs"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  CloudWatchMonitoringVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.monitoring"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  KmsVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.kms"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
```

```
      - !Ref SecurityGroup

  EcrApiVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.ecr.api"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  EcrDkrVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.ecr.dkr"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  AirflowApiVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: !Sub "com.amazonaws.${AWS::Region}.airflow.api"
      VpcEndpointType: Interface
      VpcId: !Ref VPC
      PrivateDnsEnabled: true
      SubnetIds:
       - !Ref PrivateSubnet1
       - !Ref PrivateSubnet2
      SecurityGroupIds:
       - !Ref SecurityGroup

  AirflowEnvVpcEndoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
```

```
        ServiceName: !Sub "com.amazonaws.${AWS::Region}.airflow.env"
        VpcEndpointType: Interface
        VpcId: !Ref VPC
        PrivateDnsEnabled: true
        SubnetIds:
         - !Ref PrivateSubnet1
         - !Ref PrivateSubnet2
        SecurityGroupIds:
         - !Ref SecurityGroup

Outputs:
   VPC:
     Description: A reference to the created VPC
     Value: !Ref VPC

   MwaaSecurityGroupId:
     Description: Associates the Security Group to the environment to allow access
 to the VPC endpoints
     Value: !Ref SecurityGroup

   PrivateSubnets:
     Description: A list of the private subnets
     Value: !Join [ ",", [ !Ref PrivateSubnet1, !Ref PrivateSubnet2 ]]

   PrivateSubnet1:
     Description: A reference to the private subnet in the 1st Availability Zone
     Value: !Ref PrivateSubnet1

   PrivateSubnet2:
     Description: A reference to the private subnet in the 2nd Availability Zone
     Value: !Ref PrivateSubnet2
```

2. 在命令提示符中，導航到目錄cfn-vpc-private.yml被存儲。例如：

```
cd mwaaproject
```

3. 使用aws cloudformation create-stack指令，以建立使用AWS CLI。

```
aws cloudformation create-stack --stack-name mwaa-private-environment --template-
body file://cfn-vpc-private.yml
```

> **Note**
>
> 建立 Amazon VPC 基礎設施大約需要 30 分鐘。

4. 您需要建立一種機制，以便從電腦存取這些 VPC 端點。如需進一步了解，請參閱 在 Amazon MWAA 上管理服務特定 Amazon VPC 端點的存取。

> **Note**
>
> 您可以在 Amazon MVAA 安全群組的 CIDR 中進一步限制輸出存取。例如，您可以通過添加自我引用的出站規則來限制自己前綴列表適用於亞馬遜 S3 和您的亞馬遜 VPC 的 CIDR。

後續步驟？

- 了解如何在以下位置建立亞馬遜 MWAA 環境建立一個 Amazon MWAA 環境。
- 了解如何使用私有路由建立 VPN 通道，從您的電腦到 Amazon VPC教學課程：使用AWS Client VPN。

建立一個 Amazon MWAA 環境

適用於 Apache 氣流的 Amazon 受管工作流程，使用 Apache 提供的相同開放原始碼氣流和使用者介面，在您所選版本的環境中設定 Apache 氣流。本指南說明建立 Amazon MWAA 環境的步驟。

內容

開始之前

- 建立環境後，您為環境指定的虛擬私人 VPC 網路無法變更。
- 您需要設定為封鎖所有公用存取的 Amazon S3 儲存貯體，並啟用儲存貯體版本控制。
- 您需要具有許可的 AWS 帳戶才能使用 Amazon MWAA，以及在 AWS Identity and Access Management (IAM) 中建立 IAM 角色的許可。如果您選擇 Apache 氣流網頁伺服器的私人網路存取模式，該模式會限制 Amazon VPC 內的 Apache 氣流存取，則需要 IAM 中的許可，才能建立 Amazon VPC 端點。

阿帕奇氣流版本

Amazon 管理的 Apache 氣流程支援以下 Apache 氣流版本。

> **Note**
>
> - 從 Apache 氣流 v2.2.2 開始，Amazon MWAA 支援直接在 Apache 氣流網頁伺服器上安裝 Python 需求、供應商套件和自訂外掛程式。
> - 從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一`--constraint`份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。
>
>   如需在需求檔案中設定條件約束的詳細資訊，請參閱安裝 Python 相依性。

阿帕奇氣流版	阿帕奇氣流指南	阿帕奇氣流限制	Python 版本
v2.8.1	阿帕奇氣流 v2.8.1 參考指南	阿帕奇氣流 v2.8.1 約束文件	Python
v2.7.2	阿帕奇氣流 v2.7.2 參考指南	阿帕奇氣流 v2.7.2 約束文件	Python
v2.6.3	阿帕奇氣流 v2.6.3 參考指南	阿帕奇氣流 v2.6.3 約束文件	Python

阿帕奇氣流版	阿帕奇氣流指南	阿帕奇氣流限制	Python 版本
v2.5.1	阿帕奇氣流 v2.5.1 參考指南	阿帕奇氣流 v2.5.1 約束文件	Python
v2.4.3	阿帕奇氣流 v2.4.3 參考指南	阿帕奇氣流 v2.4.3 約束文件	Python
v2.2.2	阿帕奇氣流 v2.2.2 參考指南	阿帕奇氣流 v2.2.2 約束文件	Python 3.7
V2.0.2	阿帕奇氣流 v2.0.2 參考指南	阿帕奇氣流 v2.0.2 約束文件	Python 3.7

如需有關遷移自我管理的 Apache Airflow 部署或遷移現有 Amazon MWAA 環境的詳細資訊 (包括備份中繼資料資料庫的說明)，請參閱 Amazon M WAA 遷移指南。

建立環境

以下部分說明建立 Amazon MWAA 環境的步驟。

步驟一：指定詳細資料

若要指定環境的詳細資料

1. 開啟 Amazon MWAA 主控台。
2. 使用「 AWS 地區」選取器選取您的地區。
3. 選擇 Create environment (建立環境)。
4. 在 [指定詳細資料] 頁面的 [環境詳細資料] 下
   a. 在名稱中輸入環境的唯一名稱。
   b. 在氣流版本中選擇 Apache 氣流版本。

> **Note**
> 如果未指定任何值，則預設為最新的 Airflow 版本。可用的最新版本是阿帕奇氣流 v2.8.1。

5. 在 Amazon S3 的 DAG 代碼下，指定以下內容：

   a. S3 存儲桶。選擇瀏覽 S3 並選取您的 Amazon S3 儲存貯體，或輸入 Amazon S3 URI。

   b. DAG 文件夾。選擇瀏覽 S3，然後選取 Amazon S3 儲存貯體中的`dags`資料夾，或輸入 Amazon S3 URI。

   c. 插件文件-可選。選擇瀏覽 S3，然後選取 Amazon S3 儲存貯體上的`plugins.zip`檔案，或輸入 Amazon S3 URI。

   d. 需求文件-可選。選擇瀏覽 S3，然後選取 Amazon S3 儲存貯體上的`requirements.txt`檔案，或輸入 Amazon S3 URI。

   e. 啟動指令碼檔案-選用，選擇「瀏覽」S3 並選取 Amazon S3 儲存貯體上的指令碼檔案，或輸入 Amazon S3 URI。

6. 選擇下一步。

步驟二：設定進階設定

設定進階設定

1. 在「設定進階設定」 頁面的「網路」下：

   - 選擇您的 Amazon VPC。

     此步驟會填入 Amazon VPC 中的兩個私有子網路。

2. 在「網頁伺服器存取」下，選取您偏好的 Apache 氣流存取模式：

   a. 私人網路。這會將 Apache 氣流使用者介面的存取權限限制為您的 Amazon VPC 中已授予您環境的 IAM 政策存取權限的使用者。您需要獲得建立 Amazon VPC 端點的許可，才能執行此步驟。

   > **Note**
   >
   > 如果您的 Apache Airflow UI 只能在公司網路內存取，而您不需要存取公用儲存庫來安裝 Web 伺服器需求，請選擇私人網路選項。如果您選擇此存取模式選項，則需要建立一個機制來存取 Amazon VPC 中的 Apache 氣流網頁伺服器。如需詳細資訊，請參閱 存取 Apache 氣流網頁伺服器的 VPC 私人雲端端點 (私人網路存取)。

   b. 公共網絡。這允許被授權存取您環境的 IAM 政策的使用者透過網際網路存取 Apache 氣流使用者介面。

3. 在安全群組下，選擇用於保護 Amazon VPC 的安全群組：

   a. 根據預設，Amazon MWAA 會使用建立新安全群組中的特定輸入和輸出規則，在 Amazon VPC 中建立安全群組。

   b. 「選用」。取消選取 [建立新安全性群組] 中的核取方塊，最多可選取 5 個安全性群組。

   > **Note**
   >
   > 現有的 Amazon VPC 安全群組必須使用特定的輸入和輸出規則進行設定，以允許網路流量。如需進一步了解，請參閱 Amazon MWAA 上 VPC 的安全政策。

4. 在環境類別下，選擇環境類別。

   我們建議您選擇支援工作負載所需的最小尺寸。您可以隨時變更環境類別。

5. 針對最大工作者計數，請指定要在環境中執行的 Apache Airflow 工作程式數目上限。

   如需進一步了解，請參閱 範例高效能使用案例。

6. 在「加密」下，選擇資料加密選項：

   a. 根據預設，Amazon MWAA 會使用 AWS 擁有的金鑰來加密您的資料。

   b. 「選用」。選擇 [自訂加密設定 (進階)] 以選擇不同的 AWS KMS 金鑰。如果您選擇在此步驟中指定客戶管理的金鑰，則必須指定 AWS KMS 金鑰 ID 或 ARN。 AWS KMS Amazon MWAA 不支援別名和多區域金鑰。如果您在 Amazon S3 儲存貯體上為伺服器端加密指定了 Amazon S3 金鑰，則必須為 Amazon MWAA 環境指定相同的金鑰。

   > **Note**
   >
   > 您必須擁有金鑰的許可，才能在 Amazon MWAA 主控台上選取該金鑰。您還必須透過附加中所述的政策來授予 Amazon MWAA 使用金鑰的許可。附加金鑰原則

7. 建議使用。在監控下，為氣流記錄組態選擇一或多個記錄類別，以將 Apache Airflow 記錄傳送至記 CloudWatch 錄檔：

   a. 氣流工作記錄。選擇 Apache 氣流工作記錄檔的類型，以傳送至 CloudWatch 記錄層級的記錄檔。

   b. 氣流網頁伺服器記錄檔。選擇 Apache Airflow 網頁伺服器記錄檔的類型，以傳送至 CloudWatch 記錄層級的記錄檔。

c. 氣流排程器記錄。選擇 Apache Airflow 排程器記錄檔的類型，以傳送至 CloudWatch 記錄層級的記錄檔。

d. 氣流工作者日誌。選擇 Apache 氣流背景工作者記錄檔的類型，以傳送至 CloudWatch 記錄層級的記錄檔。

e. 氣流 DAG 處理記錄檔。選擇 Apache 氣流 DAG 處理記錄檔的類型，以傳送至 CloudWatch 記錄層級的記錄檔。

8. 「選用」。對於氣流組態選項，請選擇新增自訂組態選項。

   您可以從 Apache 氣流版本的 Apache 氣流組態選項的建議下拉式清單中選擇，或指定自訂組態選項。例如`core.default_task_retries：3`。

9. 「選用」。在「標籤」下，選擇「新增標籤」，將標籤與您的環境相關聯。例如`Environment：Staging`。

10. 在「權限」下，選擇一個執行角色：

    a. 根據預設，Amazon MWAA 會在建立新角色中建立執行角色。您必須擁有建立 IAM 角色的權限，才能使用此選項。

    b. 「選用」。選擇輸入角色 ARN 以輸入現有執行角色的 Amazon 資源名稱 (ARN)。

11. 選擇下一步。

步驟三：檢閱並建立

檢閱環境摘要的步驟

- 檢閱環境摘要，選擇建立環境。

> **Note**
>
> 創建環境大約需要二十到三十分鐘。

後續步驟？

- 了解如何授予使用者存取您的 Apache 氣流網路伺服器和 Amazon MWAA 環境的存取權。管理對 Amazon MWAA 環境的存取
- 了解如何在在 Amazon MWAA 上管理服務特定 Amazon VPC 端點的存取中存取 Apache 氣流網頁伺服器 (私人網路存取) 的 VPC 擬私人雲端端點。

- 探索用於在建立環境的 Amazon MWAA API 操作。CreateEnvironment

後續步驟？

- 了解如 Amazon S3 在為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯.

管理對 Amazon MWAA 環境的存取

Apache Airflow 的 Amazon 受管工作流程必須獲得允許，才能使用某個環境所使用的其他 AWS 服務和資源。您還需要獲得存取 Amazon MWAA 環境和 AWS Identity and Access Management (IAM) 中的 Apache 氣流使用者介面的權限。本節說明用來授予環境 AWS 資源存取權的執行角色，以及如何新增許可，以及存取 Amazon MWAA 環境和 Apache Airflow 使用者介面所需的 AWS 帳戶許可。

主題

存取 Amazon MWAA 環境

若要針對 Apache 氣流使用 Amazon 受管工作流程，您必須使用帳戶和具有必要許可的 IAM 實體。本頁說明您可以附加至 Apache 氣流開發團隊的存取政策，以及適用於 Apache 氣流環境的 Amazon 管理工作流程的 Apache 氣流使用者。

我們建議您使用臨時登入資料，並使用群組和角色設定聯合身分，以存取 Amazon MWAA 資源。最佳做法是避免將政策直接附加到 IAM 使用者，而是定義群組或角色以提供 AWS 資源的臨時存取權。

IAM 角色是您可以在帳戶中建立的另一種 IAM 身分，具有特定的許可。IAM 角色與 IAM 使用者類似，因為它是具有許可政策的 AWS 身分識別，可決定身分可以執行和不能在其中執行的操作 AWS。但是，角色的目的是讓需要它的任何人可代入，而不是單獨地與某個人員關聯。此外，角色沒有與之關聯的標準長期憑證，例如密碼或存取金鑰。反之，當您擔任角色時，其會為您的角色工作階段提供臨時安全性憑證。

若要將權限指派給同盟身分識別，您可以建立角色並定義角色的權限。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。

您可以在帳戶中使用 IAM 角色授予其他存取帳戶資源的 AWS 帳戶 許可。如需範例，請參閱 IAM 使用者指南中的教學課程：跨 AWS 帳戶 使用 IAM 角色委派存取權。

章節

運作方式

並非所有 AWS Identity and Access Management (IAM) 實體都能存取 Amazon MWAA 環境中使用的資源和服務。您必須建立政策，授與 Apache Airflow 使用者存取這些資源的權限。例如，您需要授與您的 Apache 氣流開發團隊的存取權限。

Amazon MWAA 使用這些政策來驗證使用者是否具有在 AWS 主控台上執行動作所需的許可，還是透過環境使用的 API 執行動作。

您可以使用本主題中的 JSON 政策為 IAM 中的 Apache Airflow 使用者建立政策，然後將該政策附加到 IAM 中的使用者、群組或角色。

- AmazonmWAA FullConsoleAccess — 使用此政策授與在 Amazon MWAA 主控台上設定環境的權限。
- AmazonmWAA FullApiAccess — 使用此政策授予對用於管理環境的所有 Amazon MWAA API 的存取權。
- AmazonmWAA ReadOnlyAccess — 使用此政策授予存取權，以檢視 Amazon MWAA 主控台上某個環境所使用的資源。
- 亞馬遜 WebServerAccess — 使用此原則可授予對 Apache 氣流網路伺服器的存取權。
- 亞馬遜 AirflowCliAccess — 使用此原則可授與執行 Apache 氣流 CLI 命令的存取權。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台)中的指示。

完整的控制台訪問策略:亞馬遜 FullConsoleAccess

如果使用者需要在 Amazon MWAA 主控台上設定環境，則可能需要存取`AmazonMWAAFullConsoleAccess`許可政策。

> **Note**
>
> 您的完整主控台存取原則必須包含執行權限`iam:PassRole`。這可讓使用者將服務連結的角色和執行角色傳遞給 Amazon MWAA。Amazon MWAA 擔任每個角色，以便代表您呼叫其他 AWS 服務。下列範例使用`iam:PassedToService`條件金鑰來指定 Amazon MWAA 服務主體 (`airflow.amazonaws.com`) 做為可傳遞角色的服務。
> 如需詳細資訊`iam:PassRole`，請參閱《IAM 使用者指南》中的授與使用者將角色傳遞給 AWS 服務的權限。

如果您想要使用靜態加密來建立和管理 Amazon MWAA 環境，請使AWS 擁有的金鑰用下列政策。

使用 AWS 擁有的金鑰

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"airflow:*",
         "Resource":"*"
      },
```

```
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "iam:PassedToService":"airflow.amazonaws.com"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreatePolicy"
         ],
         "Resource":"arn:aws:iam::YOUR_ACCOUNT_ID:policy/service-role/MWAA-Execution-
Policy*"
      },
            {
         "Effect":"Allow",
         "Action":[
            "iam:AttachRolePolicy",
            "iam:CreateRole"
         ],
         "Resource":"arn:aws:iam::YOUR_ACCOUNT_ID:role/service-role/AmazonMWAA*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreateServiceLinkedRole"
         ],
         "Resource":"arn:aws:iam::*:role/aws-service-role/airflow.amazonaws.com/
AWSServiceRoleForAmazonMWAA"
      },
      {
```

```
            "Effect":"Allow",
            "Action":[
               "s3:GetBucketLocation",
               "s3:ListAllMyBuckets",
               "s3:ListBucket",
               "s3:ListBucketVersions"
            ],
            "Resource":"*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "s3:CreateBucket",
               "s3:PutObject",
               "s3:GetEncryptionConfiguration"
            ],
            "Resource":"arn:aws:s3:::*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "ec2:DescribeSecurityGroups",
               "ec2:DescribeSubnets",
               "ec2:DescribeVpcs",
               "ec2:DescribeRouteTables"
            ],
            "Resource":"*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "ec2:AuthorizeSecurityGroupIngress",
               "ec2:CreateSecurityGroup"
            ],
            "Resource":"arn:aws:ec2:*:*:security-group/airflow-security-group-*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "kms:ListAliases"
            ],
            "Resource":"*"
         },
         {
```

```
         "Effect":"Allow",
         "Action":"ec2:CreateVpcEndpoint",
         "Resource":[
            "arn:aws:ec2:*:*:vpc-endpoint/*",
            "arn:aws:ec2:*:*:vpc/*",
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:security-group/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateNetworkInterface"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:network-interface/*"
         ]
      }
   ]
}
```

如果您想要使用客戶受管金鑰進行靜態加密，建立和管理 Amazon MWAA 環境，請使用下列政策。若要使用客戶受管金鑰，IAM 主體必須具有使用您帳戶中儲存的金鑰存取 AWS KMS 資源的權限。

使用客戶管理的金鑰

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"airflow:*",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
```

```
                "iam:PassedToService":"airflow.amazonaws.com"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreatePolicy"
         ],
         "Resource":"arn:aws:iam::YOUR_ACCOUNT_ID:policy/service-role/MWAA-Execution-
Policy*"
      },
            {
         "Effect":"Allow",
         "Action":[
            "iam:AttachRolePolicy",
            "iam:CreateRole"
         ],
         "Resource":"arn:aws:iam::YOUR_ACCOUNT_ID:role/service-role/AmazonMWAA*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreateServiceLinkedRole"
         ],
         "Resource":"arn:aws:iam::*:role/aws-service-role/airflow.amazonaws.com/
AWSServiceRoleForAmazonMWAA"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetBucketLocation",
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:ListBucketVersions"
         ],
         "Resource":"*"
```

```
        },
        {
            "Effect":"Allow",
            "Action":[
                "s3:CreateBucket",
                "s3:PutObject",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource":"arn:aws:s3:::*"
        },
        {
            "Effect":"Allow",
            "Action":[
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeRouteTables"
            ],
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":[
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup"
            ],
            "Resource":"arn:aws:ec2:*:*:security-group/airflow-security-group-*"
        },
        {
            "Effect":"Allow",
            "Action":[
                "kms:ListAliases"
            ],
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":[
                "kms:DescribeKey",
                "kms:ListGrants",
                "kms:CreateGrant",
                "kms:RevokeGrant",
                "kms:Decrypt",
                "kms:Encrypt",
```

```
            "kms:GenerateDataKey*",
            "kms:ReEncrypt*"
         ],
         "Resource":"arn:aws:kms:*:YOUR_ACCOUNT_ID:key/YOUR_KMS_ID"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateVpcEndpoint",
         "Resource":[
            "arn:aws:ec2:*:*:vpc-endpoint/*",
            "arn:aws:ec2:*:*:vpc/*",
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:security-group/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateNetworkInterface"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:network-interface/*"
         ]
      }
   ]
}
```

完整的 API 和控制台訪問策略：亞馬遜 FullApiAccess

如果使用者需要存取用於管理環境的所有 Amazon MWAA API，則可能需要存取`AmazonMWAAFullApiAccess`許可政策。它不會授與存取 Apache 氣流使用者介面的權限。

> **Note**
>
> 完整的 API 存取政策必須包含執行權限`iam:PassRole`。這可讓使用者將服務連結的角色和執行角色傳遞給 Amazon MWAA。Amazon MWAA 擔任每個角色，以便代表您呼叫其他 AWS 服務。下列範例使用`iam:PassedToService`條件金鑰來指定 Amazon MWAA 服務主體 (`airflow.amazonaws.com`) 做為可傳遞角色的服務。
> 如需詳細資訊`iam:PassRole`，請參閱《IAM 使用者指南》中的授與使用者將角色傳遞給 AWS 服務的權限。

如果您想要使用靜態加密來建立和管理 Amazon MWAA 環境，請使 AWS 擁有的金鑰 用下列政策。

使用 AWS 擁有的金鑰

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"airflow:*",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "iam:PassedToService":"airflow.amazonaws.com"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreateServiceLinkedRole"
         ],
         "Resource":"arn:aws:iam::*:role/aws-service-role/airflow.amazonaws.com/
AWSServiceRoleForAmazonMWAA"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DescribeRouteTables"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "s3:GetEncryptionConfiguration"
         ],
         "Resource":"arn:aws:s3:::*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateVpcEndpoint",
         "Resource":[
            "arn:aws:ec2:*:*:vpc-endpoint/*",
            "arn:aws:ec2:*:*:vpc/*",
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:security-group/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateNetworkInterface"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:network-interface/*"
         ]
      }
   ]
}
```

如果您想要使用客戶受管金鑰進行靜態加密，建立和管理 Amazon MWAA 環境，請使用下列政策。若要使用客戶受管金鑰，IAM 主體必須具有使用您帳戶中儲存的金鑰存取 AWS KMS 資源的權限。

使用客戶管理的金鑰

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"airflow:*",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "iam:PassRole"
         ],
         "Resource":"*",
         "Condition":{
            "StringLike":{
               "iam:PassedToService":"airflow.amazonaws.com"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:CreateServiceLinkedRole"
         ],
         "Resource":"arn:aws:iam::*:role/aws-service-role/airflow.amazonaws.com/
AWSServiceRoleForAmazonMWAA"
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DescribeRouteTables"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "kms:DescribeKey",
            "kms:ListGrants",
            "kms:CreateGrant",
            "kms:RevokeGrant",
            "kms:Decrypt",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt*"
         ],
         "Resource":"arn:aws:kms:*:YOUR_ACCOUNT_ID:key/YOUR_KMS_ID"
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "s3:GetEncryptionConfiguration"
         ],
         "Resource":"arn:aws:s3:::*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateVpcEndpoint",
         "Resource":[
            "arn:aws:ec2:*:*:vpc-endpoint/*",
            "arn:aws:ec2:*:*:vpc/*",
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:security-group/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateNetworkInterface"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:network-interface/*"
         ]
      }
   ]
}
```

只讀控制台訪問策略:亞馬遜 ReadOnlyAccess

如果使用者需要在 Amazon MWAA 主控台環境詳細資料頁面上檢視某個環境使用的資源，則可能需要存取`AmazonMWAAReadOnlyAccess`許可政策。它不允許使用者建立新的環境、編輯現有環境，或允許使用者檢視 Apache Airflow 使用者介面。

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "airflow:ListEnvironments",
                "airflow:GetEnvironment",
                "airflow:ListTagsForResource"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess

如果使用者需要存取 Apache 氣流使用者介面，可能需要存取AmazonMWAAWebServerAccess權限原則。它不允許使用者在 Amazon MWAA 主控台上檢視環境，也不允許使用 Amazon MWAA API 執行任何動作。在中指定Admin0p、User、Viewer或Public角色，{airflow-role}以自訂 Web Token 使用者的存取層級。如需詳細資訊，請參閱 Apache 氣流參考指南中的預設角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "airflow:CreateWebLoginToken",
            "Resource": [
                "arn:aws:airflow:{your-region}:YOUR_ACCOUNT_ID:role/{your-environment-name}/{airflow-role}"
            ]
        }
    ]
}
```

> **Note**
>
> Amazon MWAA 提供與五個預設 Apache 氣流角色型存取控制 (RBAC) 角色的 IAM 整合。如需使用自訂 Apache 氣流角色的詳細資訊，請參閱the section called “教學課程：將使用者限制為 DAG 的子集”。

阿帕奇氣流 CLI 政策:亞馬遜 AirflowCliAccess

如果使用者需要執行 Apache 氣流 CLI 命令 (例如trigger_dag)，則可能需要存取AmazonMWAAAirflowCliAccess權限原則。它不允許使用者在 Amazon MWAA 主控台上檢視環境，也不允許使用 Amazon MWAA API 執行任何動作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "airflow:CreateCliToken"
            ],
            "Resource": "*"
        }
    ]
}
```

建立 JSON 政策

您可以建立 JSON 政策，並將政策附加到 IAM 主控台上的使用者、角色或群組。下列步驟說明如何在 IAM 中建立 JSON 政策。

若要建立 JSON 政策

1. 在 IAM 主控台上開啟「政策」頁面。
2. 選擇 Create policy (建立政策)。
3. 請選擇 JSON 標籤。
4. 新增您的 JSON 政策。
5. 選擇檢閱政策。
6. 在「名稱」和「說明」的文字欄位中輸入值 (選擇性)。

   例如，您可以命名策略`AmazonMWAAReadOnlyAccess`。
7. 選擇建立政策。

將原則附加至開發人員群組的範例使用案例

假設您正在使用 IAM 中命名的群組，`AirflowDevelopmentGroup`將許可套用至 Apache Airflow 開發團隊的所有開發人員。這些使用者需要存取`AmazonMWAAFullConsoleAccess`、`AmazonMWAAAirflowCliAccess`和`AmazonMWAAWebServerAccess`權限原則。本節說明如何在 IAM 中建立群組、建立和附加這些政策，以及如何將該群組與 IAM 使用者建立關聯。這些步驟假設您使用的是AWS 擁有的金鑰。

若要建立亞馬遜 WAA 政策 FullConsoleAccess

1. 下載亞馬遜訪問政策. FullConsoleAccess
2. 在 IAM 主控台上開啟「政策」頁面。
3. 選擇 Create policy (建立政策)。
4. 請選擇 JSON 標籤。
5. 貼上的 JSON 政策AmazonMWAAFullConsoleAccess。
6. 取代下列值：
   a. *{your-account-id}* — 您的 AWS 帳戶 ID（例如0123456789）
   b. *{your-kms-id}* — 客戶管理金鑰的唯一識別碼，僅在您使用客戶管理的金鑰進行靜態加密時適用。
7. 選擇 [檢閱] 原則。
8. 輸AmazonMWAAFullConsoleAccess入名稱。
9. 選擇建立政策。

若要建立亞馬遜 WAA 政策 WebServerAccess

1. 下載亞馬遜訪問政策. WebServerAccess
2. 在 IAM 主控台上開啟「政策」頁面。
3. 選擇 Create policy (建立政策)。
4. 請選擇 JSON 標籤。
5. 貼上的 JSON 政策AmazonMWAAWebServerAccess。
6. 取代下列值：
   a. *{#####} – ## Amazon MWAA ##### (*例如) us-east-1
   b. *{your-account-id}*-您的 AWS 帳戶 ID（例如0123456789）
   c. *{your-environment-name}* — 您的 Amazon MWAA 環境名稱 (例如) MyAirflowEnvironment
   d. *{####}-Admin* 阿帕奇氣流默認角色
7. 選擇檢閱政策。
8. 輸AmazonMWAAWebServerAccess入名稱。
9. 選擇建立政策。

若要建立亞馬遜 WAA 政策 AirflowCliAccess

1. 下載亞馬遜訪問政策. AirflowCliAccess
2. 在 IAM 主控台上開啟「政策」頁面。
3. 選擇 Create policy (建立政策)。
4. 請選擇 JSON 標籤。
5. 貼上的 JSON 政策`AmazonMWAAAirflowCliAccess`。
6. 選擇 [檢閱] 原則。
7. 輸`AmazonMWAAAirflowCliAccess`入名稱。
8. 選擇建立政策。

若要建立群組

1. 在 IAM 主控台上開啟「群組」頁面。
2. 輸入的名稱`AirflowDevelopmentGroup`。
3. 選擇 Next Step (後續步驟)。
4. 輸入`AmazonMWAA`以在篩選中篩選結果。
5. 選取您建立的三個策略。
6. 選擇 Next Step (後續步驟)。
7. 選擇 Create Group (建立群組)。

關聯至使用者

1. 在 IAM 主控台上開啟「使用者」頁面。
2. 選擇使用者。
3. 選擇 Groups (群組)。
4. 選擇 [新增使用者至群組]。
5. 選取AirflowDevelopmentGroup。
6. 選擇 Add to Groups (新增至群組)。

後續步驟？

- 瞭解如何在中產生權杖以存取 Apache 氣流使用者介面訪問阿帕奇氣流用戶界面。

- 請參閱建立 IAM 政策，進一步了解如何建立 IAM 政策。

Amazon MWAA 的服務連結角色

適用於 Apache 氣流的 Amazon 受管工作流程使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是一種獨特的 IAM 角色類型，可直接連結至 Amazon MWAA。Amazon MWAA 預先定義服務連結角色，包含服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定 Amazon MWAA，因為您不必手動新增必要的許可。Amazon MWAA 會定義其服務連結角色的許可，除非另有定義，否則只有 Amazon MWAA 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。這樣可以保護您的 Amazon MWAA 資源，因為您無法意外移除存取資源的權限。

如需支援服務連結角色之其他服務的相關資訊，請參閱可搭配 IAM 運作的AWS 服務，並尋找 Service-linked roles (服務連結角色) 資料行中顯示為 Yes (是) 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

Amazon MWAA 的服務連結角色許可

Amazon MWAA 使用名為的服務連結角色 `AWSServiceRoleForAmazonMWAA` — 在您帳戶中建立的服務連結角色會授予 Amazon MWAA 存取下列服務的權限： AWS

- Amazon CloudWatch 日誌 (CloudWatch 日誌) — 建立 Apache 氣流日誌的日誌群組。
- Amazon CloudWatch (CloudWatch) — 將與您的環境及其基礎元件相關的指標發佈到您的帳戶。
- Amazon Elastic Compute Cloud (Amazon EC2) — 創建以下資源：
  - 虛擬私人雲端中的一個 Amazon VPC 端點，用於由 Apache AWS氣流排程器和工作者使用的受管 Amazon Aurora PostgreSQL 資料庫叢集。
  - 如果您選擇 Apache 氣流網頁伺服器的私人網路選項，則可使用另一個 Amazon VPC 端點來啟用網頁伺服器的網路存取。
  - Amazon VPC 中的彈性網路界面 (ENI) 可讓您對 Amazon VPC 中託管的 AWS 資源進行網路存取。

下列信任原則可讓服務主體擔任服務連結角色。Amazon MWAA 的服務主體`airflow.amazonaws.com`如本政策所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "airflow.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

名為的角色許可政策AmazonMWAAServiceRolePolicy允許 Amazon MWAA 對指定的資源完成下列動作：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:airflow-*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:DetachNetworkInterface"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "AmazonMWAAManaged"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyVpcEndpoint",
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonMWAAManaged": false
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:ModifyVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:subnet/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
```

```
                    "ec2:CreateAction": "CreateVpcEndpoint"
                },
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "AmazonMWAAManaged"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/MWAA"
                    ]
                }
            }
        }
    ]
}
```

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

為 Amazon MWAA 建立服務連結角色

您不需要手動建立一個服務連結角色。當您使用 AWS Management Console、或 AWS API 建立新的 Amazon MWAA 環境時 AWS CLI，Amazon MWAA 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立其他環境時，Amazon MWAA 會再次為您建立服務連結角色。

編輯 Amazon MWAA 的服務連結角色

Amazon MWAA 不允許您編輯 AWSServiceRoleForAmazonMWAA 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 IAM 使用者指南中的編輯服務連結角色。

刪除 Amazon MWAA 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。

當您刪除 Amazon MWAA 環境時，Amazon MWAA 會刪除其作為服務一部分使用的所有關聯資源。不過，您必須等待 Amazon MWAA 完成刪除環境，才能嘗試刪除服務連結角色。如果您在 Amazon MWAA 刪除環境之前刪除服務連結角色，Amazon MWAA 可能無法刪除環境的所有相關資源。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台或 AWS API 刪除 AWSServiceRoleForAmazonMWAA服務連結角色。 AWS CLI如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

支援 Amazon MWAA 服務連結角色的區域

Amazon MWAA 支援在提供服務的所有區域使用服務連結角色。如需詳細資訊，請參閱適用於 Apache 氣流端點和配額的 Amazon 受管工作流程。

政策更新

變更	描述	日期
Amazon MWAA 更新其服務連結角色許可政策	AmazonMWAAServiceRolePolicy — Amazon MWAA 更新其服務連結角色的許可政策，以授與 Amazon MWAA 將與服務基礎資源相關的其他指標發佈到客戶帳戶的權限。這些新量度會在 AWS/MWAA	2022 年 11 月 18 日
Amazon MWAA 開始追蹤變更	Amazon MWAA 開始追蹤其 AWS 受管服務連結角色權限政策的變更。	2022 年 11 月 18 日

Amazon MWAA 執行角色

執行角色是具有許可政策的 AWS Identity and Access Management (IAM) 角色，可授予適用於 Apache Airflow 的 Amazon 受管工作流程權限，以代表您叫用其他 AWS 服務的資源。這可能包括 Amazon S3 儲存貯體、AWS 擁有的金鑰和 CloudWatch 日誌等資源。Amazon MWAA 環境在每個環境中需要一個執行角色。本頁說明如何使用和設定環境的執行角色，以允許 Amazon MWAA 存取環境使用的其他 AWS 資源。

內容

執行角色概觀

Amazon MWAA 使用您的環境使用其他 AWS 服務的許可是從執行角色取得。Amazon MWAA 執行角色需要對環境使用的下列 AWS 服務的許可：

- Amazon CloudWatch （CloudWatch）-發送阿帕奇氣流指標和日誌。
- 亞馬遜簡單儲存服務 (Amazon S3) — 剖析您環境的 DAG 程式碼和支援檔案 (例如`requirements.txt`)。
- Amazon Simple Queue Service (Amazon SQS) — 將您環境的 Apache 氣流任務排入 Amazon MWAA 擁有的 Amazon SQS 隊列中。

- AWS Key Management Service (AWS KMS) — 用於環境的資料加密 (使用AWS 擁有的金鑰或客戶管理的金鑰)。

> **Note**
>
> 如果您已選擇讓 Amazon MWAA 使用 AWS 受管 KMS 金鑰來加密資料，則必須在附加至 Amazon MWAA 執行角色的政策中定義許可，以授與透過 Amazon SQS 存放在帳戶外部的任意 KMS 金鑰的存取權。為了讓您的環境的執行角色存取任意 KMS 金鑰，需要下列兩個條件：
>
> - 第三方帳戶中的 KMS 金鑰必須透過其資源原則允許此跨帳戶存取。
> - 您的 DAG 程式碼需要存取以第三方帳戶開頭的 `airflow-celery-` Amazon SQS 佇列，並使用相同的 KMS 金鑰進行加密。
>
> 為了減輕跨帳戶存取資源的相關風險，建議您檢閱放置在 DAG 中的程式碼，以確保您的工作流程不會存取帳戶外的任意 Amazon SQS 佇列。此外，您可以使用存放在自己帳戶中的客戶受管 KMS 金鑰來管理 Amazon MWAA 上的加密。這會限制環境的執行角色，只能存取帳戶中的 KMS 金鑰。
>
> 請記住，在您選擇加密選項之後，就無法變更現有環境的選擇。

執行角色也需要下列 IAM 動作的權限：

- `airflow:PublishMetrics`— 允許 Amazon MWAA 監控環境的健康狀態。

依預設附加權限

您可以使用 Amazon MWAA 主控台上的預設選項來建立執行角色和AWS 擁有的金鑰，然後使用此頁面上的步驟將許可政策新增至執行角色。

- 當您在主控台上選擇「建立新角色」選項時，Amazon MWAA 會將環境所需的最低許可附加到您的執行角色。
- 在某些情況下，Amazon MWAA 會附加最大許可。例如，我們建議您在 Amazon MWAA 主控台上選擇該選項，以便在建立環境時建立執行角色。Amazon MWAA 會使用執行角色中的正則表達式模式，自動為所有 CloudWatch 日誌群組新增許可政策。`"arn:aws:logs:your-region:your-account-id:log-group:airflow-your-environment-name-*"`

如何新增使用其他 AWS 服務的權限

建立環境後，Amazon MWAA 無法新增或編輯現有執行角色的許可政策。您必須使用環境所需的其他權限原則來更新您的執行角色。例如，如果您的 DAG 需要存取權 AWS Glue，Amazon MWAA 無法自動偵測您的環境需要這些許可，或將許可新增至您的執行角色。

您可以透過兩種方式將權限新增至執行角色：

- 透過內嵌修改執行角色的 JSON 政策。您可以使用此頁面上的範例 JSON 政策文件，在 IAM 主控台上新增或取代執行角色的 JSON 政策。
- 透過為 AWS 服務建立 JSON 政策，並將其附加至您的執行角色。您可以使用此頁面上的步驟，將 AWS 服務的新 JSON 政策文件與 IAM 主控台上的執行角色建立關聯。

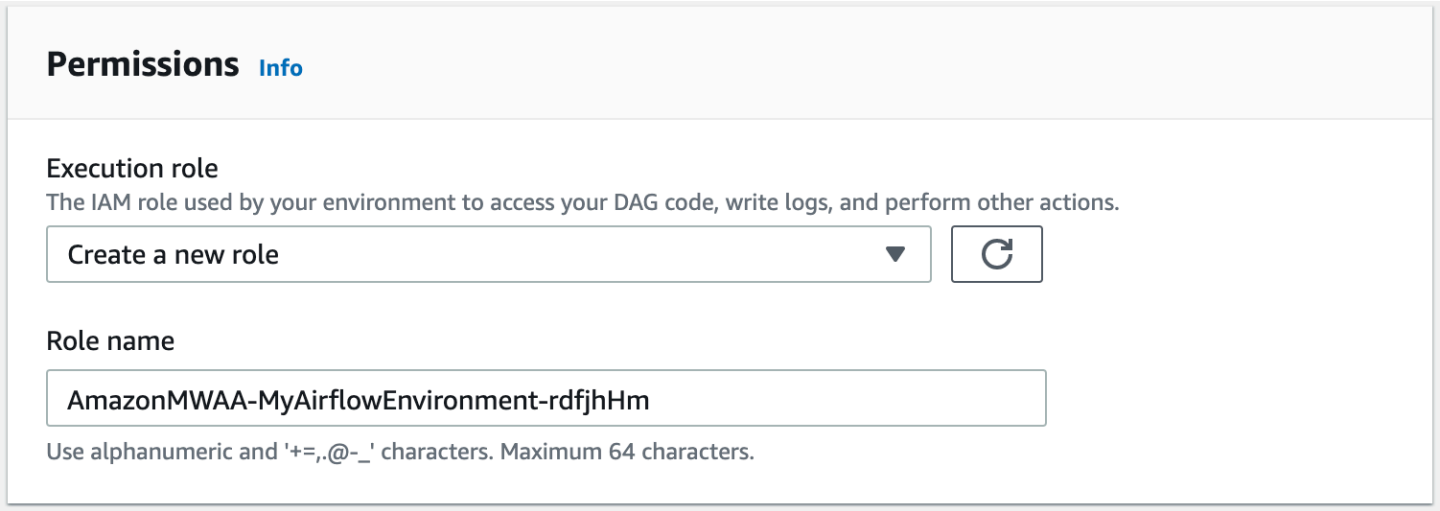
假設執行角色已與您的環境相關聯，Amazon MWAA 可以立即開始使用新增的許可政策。這也表示如果您從執行角色移除任何必要的權限，DAG 可能會失敗。

如何關聯新的執行角色

您可以隨時變更環境的執行角色。如果新的執行角色尚未與您的環境相關聯，請使用此頁面上的步驟建立新的執行角色原則，並將角色與您的環境建立關聯。

Create a new role (建立新角色)

根據預設，Amazon MWAA 會為資料加密建立AWS 擁有的金鑰，並代表您建立執行角色。建立環境時，您可以在 Amazon MWAA 主控台上選擇預設選項。下圖顯示了為環境建立執行角色的預設選項。



檢視和更新執行角色原則

您可以在 Amazon MWAA 主控台上檢視環境的執行角色，並在 IAM 主控台上更新該角色的 JSON 政策。

更新執行角色原則

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在「權限」窗格上選擇執行角色，以在 IAM 中開啟許可頁面。
4. 選擇執行角色名稱以開啟權限原則。
5. 選擇 Edit Policy (編輯政策)。
6. 選擇 JSON 標籤。
7. 更新您的 JSON 政策。
8. 選擇檢閱政策。
9. 選擇儲存變更。

附加 JSON 政策以使用其他 AWS 服務

您可以為 AWS 服務建立 JSON 政策，並將其附加至您的執行角色。例如，您可以附加下列 JSON 政策，以授與中所有資源的唯讀存取權 AWS Secrets Manager。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "secretsmanager:GetResourcePolicy",
            "secretsmanager:GetSecretValue",
            "secretsmanager:DescribeSecret",
            "secretsmanager:ListSecretVersionIds"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
```

```
}
```

若要將原則附加至您的執行角色

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在「權限」窗格中選擇您的執行角色。
4. 選擇連接政策。
5. 選擇 Create policy (建立政策)。
6. 選擇 JSON。
7. 貼上 JSON 政策。
8. 選擇「下一步」：「標籤」，「下一步」：
9. 輸入原則的描述性名稱 (例如`SecretsManagerReadPolicy`) 和說明。
10. 選擇建立政策。

透過帳戶層級公有存取區塊授予對 Amazon S3 儲存貯體的存取權

您可能想要使用 `PutPublicAccessBlock`Amazon S3 操作封鎖對帳戶中所有儲存貯體的存取。當您封鎖對帳戶中所有值區的存取時，您的環境執行角色必須在權限原則中包含該`s3:GetAccountPublicAccessBlock`動作。

以下範例示範在封鎖對帳戶中所有 Amazon S3 儲存貯體的存取時，必須附加至執行角色的政策。

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Effect": "Allow",
       "Action": "s3:GetAccountPublicAccessBlock",
       "Resource": "*"
     }
  ]
}
```

如需有關限制 Amazon S3 儲存貯體存取的詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的封鎖對 Amazon S3 儲存的公開存取。

使用阿帕奇氣流連接

您也可以建立 Apache 氣流連線，並在 Apache 氣流連線物件中指定您的執行角色及其 ARN。如需進一步了解，請參閱管理與 Apache 氣流的連線。

執行角色的 JSON 原則範例

本節中的範例權限原則顯示兩個原則，您可以用來取代現有執行角色所使用的權限原則，或建立新的執行角色並用於您的環境。這些政策包含 Apache 氣流日誌群組、Amazon S3 儲存貯體和 Amazon MWAA 環境的資源 ARN 預留位置。

建議您複製範例原則，取代範例 ARN 或預留位置，然後使用 JSON 原則建立或更新執行角色。例如，取代`{your-region}`為`us-east-1`。

客戶管理金鑰的範例政策

下列範例顯示可用於客戶受管金鑰的執行角色原則。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:GetBucket*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::{your-s3-bucket-name}",
                "arn:aws:s3:::{your-s3-bucket-name}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
```

```
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "logs:GetLogRecord",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:{your-region}:{your-account-id}:log-group:airflow-{your-environment-name}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage"
            ],
```

```
            "Resource": "arn:aws:sqs:{your-region}:*:airflow-celery-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey*",
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:{your-region}:{your-account-id}:key/{your-kms-cmk-
id}",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "sqs.{your-region}.amazonaws.com",
                        "s3.{your-region}.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

接下來，您需要允許 Amazon MWAA 擔任此角色，才能代表您執行動作。這可以透過使用 IAM 主控台將這些"airflow-env.amazonaws.com"服務主體新增"airflow.amazonaws.com"至此執行角色的受信任實體清單，或透過使用的 IAM create- role 命令，將這些服務主體放置在此執行角色的假設角色政策文件中。 AWS CLI假設角色政策文件範例可在下方找到：

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
            "Service": ["airflow.amazonaws.com","airflow-env.amazonaws.com"]
        },
        "Action": "sts:AssumeRole"
      }
   ]
}
```

然後將下列 JSON 政策附加至您的客戶管理金鑰。此原則使用kms:EncryptionContext條件 key prefix 來允許存取記錄檔中的 Apache Airflow 記 CloudWatch 錄群組。

```
{
    "Sid": "Allow logs access",
    "Effect": "Allow",
    "Principal": {
        "Service": "logs.{your-region}.amazonaws.com"
    },
    "Action": [
        "kms:Encrypt*",
        "kms:Decrypt*",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:Describe*"
    ],
    "Resource": "*",
    "Condition": {
        "ArnLike": {
            "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:{your-region}:{your-
account-id}:*"
        }
    }
}
```

AWS 擁有金鑰的範例政策

下列範例顯示可用於AWS 擁有金鑰的執行角色原則。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "airflow:PublishMetrics",
            "Resource": "arn:aws:airflow:{your-region}:{your-account-id}:environment/
{your-environment-name}"
        },
        {
            "Effect": "Deny",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:GetBucket*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::{your-s3-bucket-name}",
                "arn:aws:s3:::{your-s3-bucket-name}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "logs:GetLogRecord",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:{your-region}:{your-account-id}:log-group:airflow-{your-
environment-name}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock"
            ],
            "Resource": [
                "*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:{your-region}:*:airflow-celery-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey*",
                "kms:Encrypt"
            ],
            "NotResource": "arn:aws:kms:*:{your-account-id}:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "sqs.{your-region}.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

後續步驟？

- 瞭解您和 Apache Airflow 使用者在中存取環境所需的必要權限存取 Amazon MWAA 環境。
- 了解 使用客戶管理的金鑰進行加密。

- 探索更多客戶管理政策範例。

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，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。在中 AWS，跨服務模擬可能會導致混淆的副問題。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。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

我們建議您在環境的執行角色中使用aws:SourceArn和aws:SourceAccount全域條件上下文金鑰，以限制 Amazon MWAA 提供其他服務存取資源的許可。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。

防範混淆代理人問題的最有效方法是使用 aws:SourceArn 全域條件內容索引鍵，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 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 arn:aws:airflow:*:*123456789012*:environment/*。

的值aws:SourceArn必須是您要為其建立執行角色的 Amazon MWAA 環境 ARN。

下列範例顯示如何在環境的執行角色信任原則中使用aws:SourceArn和aws:SourceAccount全域條件內容索引鍵，以防止混淆的副問題。當您建立新的執行角色時，您可以使用下列信任原則。

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
            "Service": ["airflow.amazonaws.com","airflow-env.amazonaws.com"]
        },
        "Action": "sts:AssumeRole",
        "Condition":{
            "ArnLike":{
               "aws:SourceArn":"arn:aws:airflow:your-
region:123456789012:environment/your-environment-name"
            },
            "StringEquals":{
```

```
                "aws:SourceAccount":"123456789012"
            }
         }
      }
   ]
}
```

阿帕奇氣流存取模式

適用於 Apache 氣流主控台的 Amazon 受管工作流程包含內建選項，可設定連接至環境中 Apache 氣流網路伺服器的私有或公有路由。本指南說明適用於 Apache 氣流環境的 Amazon 受管工作流程上 Apache Airflow 網路伺服器可用的存取模式，以及如果您選擇私有網路選項，則需要在 Amazon VPC 中設定的其他資源。

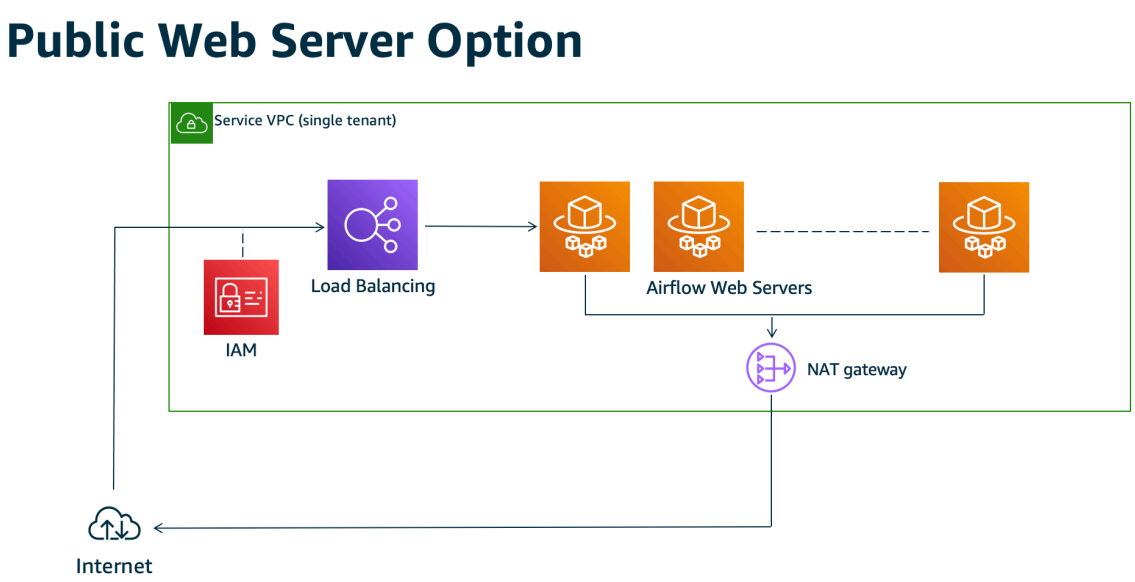
內容

阿帕奇氣流存取模式

您可以為 Apache 氣流網頁伺服器選擇私人或公用路由。若要啟用私人路由，請選擇「私人網路」。這會限制使用者只能在 Amazon VPC 內存取 Apache 氣流網頁伺服器。若要啟用公用路由，請選擇 [公用網路]。這可讓使用者透過網際網路存取 Apache 氣流網頁伺服器。

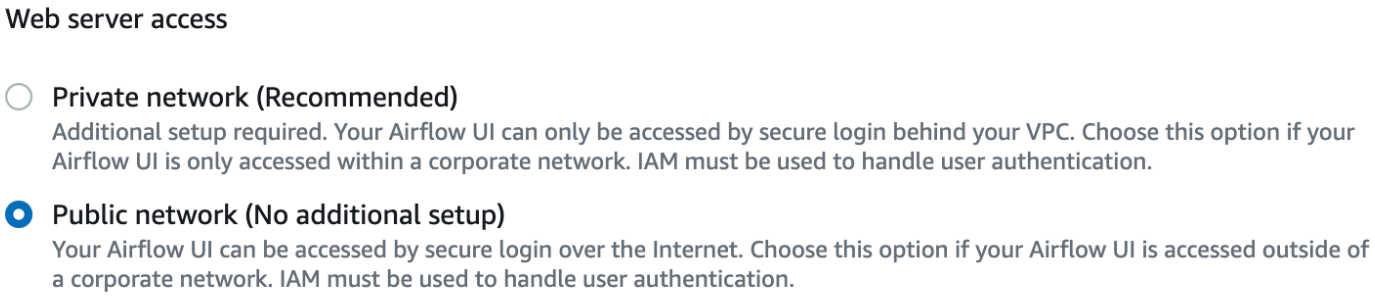
大眾網

下列架構圖顯示具有公用 Web 伺服器的 Amazon MWAA 環境。



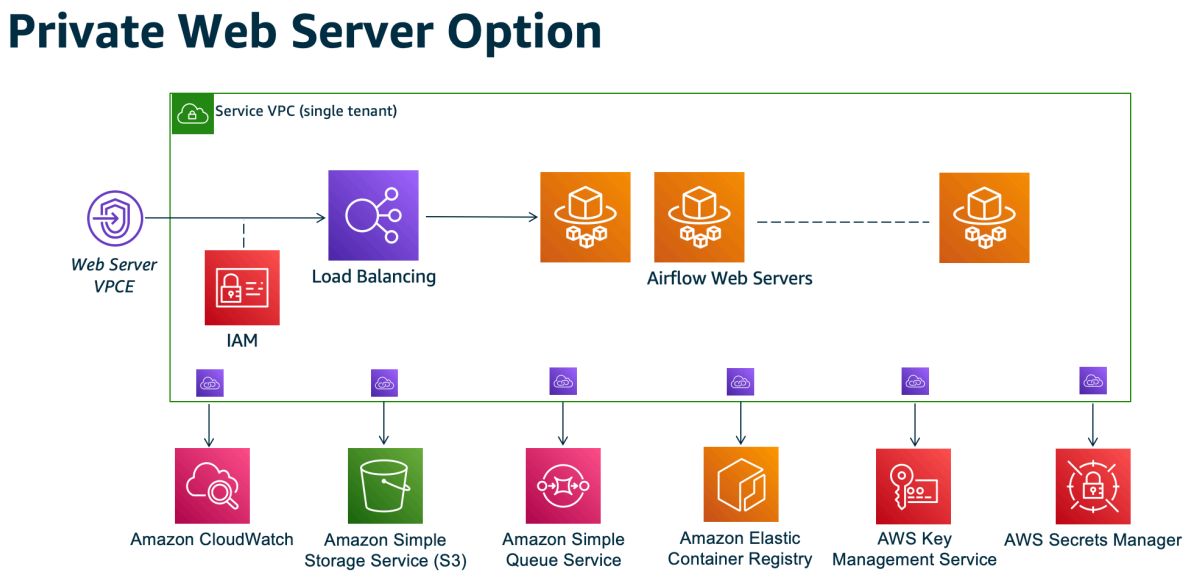
公用網路存取模式允許被授予您環境 IAM 政策存取權的使用者透過網際網路存取 Apache Airflow UI。

下圖顯示 Amazon MWAA 主控台上哪裡可以找到公用網路選項。



私人網路

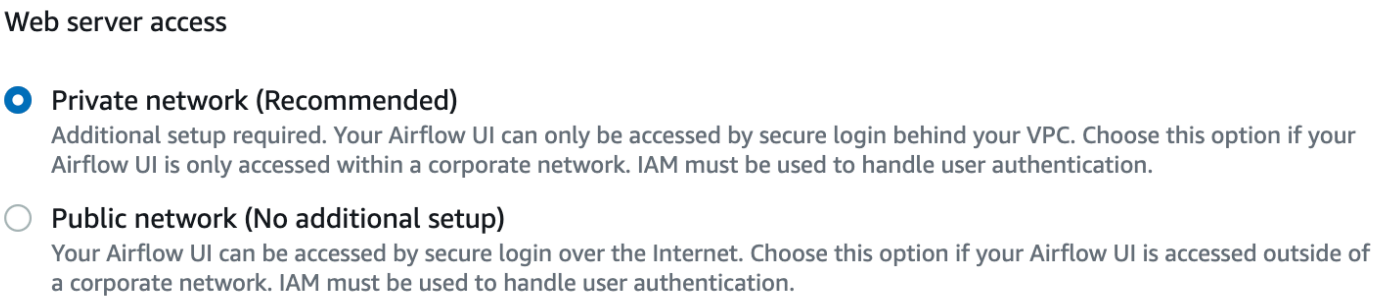
下列架構圖顯示具有私有網頁伺服器的 Amazon MWAA 環境。



私有網路存取模式可將 Apache Airflow UI 的存取權限限制為 Amazon VPC 中已授予您環境 IAM 政策存取權的使用者。

當您創建具有私有 Web 服務器訪問權限的環境時，必須將所有依賴項打包到 Python wheel 存檔（.whl）中，然後.whl在requirements.txt. 有關使用 wheel 打包和安裝依賴項的說明，請參閱使用 Python wheel 管理依賴關係。

下圖顯示 Amazon MWAA 主控台上哪裡可以找到私人網路選項。



存取模式概觀

本節說明當您選擇公用網路或私人網路存取模式時，在 Amazon VPC 中建立的 VPC 端點 (AWS PrivateLink)。

公共網路存取模式

如果您為 Apache Airflow 網頁伺服器選擇公用網路存取模式，網路流量會透過網際網路公開路由。

- Amazon MWAA 為您的 Amazon Aurora PostgreSQL 中繼資料資料庫建立 VPC 界面端點。端點是在對應至私人子網路的可用區域中建立，且獨立於其他 AWS 帳戶。
- 然後，Amazon MWAA 會將 IP 位址從您的私有子網路繫結到介面端點。這是為了支援從 Amazon VPC 的每個可用區域繫結單一 IP 的最佳實務而設計。

私人網路存取模式

如果您為 Apache Airflow 網頁伺服器選擇私人網路存取模式，網路流量會在您的 Amazon VPC 中以私密方式路由。

- Amazon MWAA 為您的 Apache 氣流網頁伺服器建立 VPC 界面端點，並為您的 Amazon Aurora PostgreSQL 中繼資料資料庫建立一個介面端點。端點是在對應至私人子網路的可用區域中建立，且獨立於其他 AWS 帳戶。
- 然後，Amazon MWAA 會將 IP 位址從您的私有子網路繫結到介面端點。這是為了支援從 Amazon VPC 的每個可用區域繫結單一 IP 的最佳實務而設計。

如需進一步了解，請參閱the section called “Amazon VPC 和 Apache 氣流存取模式的範例使用案例”。

設置私人和公共訪問模式

下節根據您為環境選擇的 Apache Airflow 存取模式，說明您需要的其他設定和組態。

設定公用網路

如果您選擇 Apache 氣流網頁伺服器的公用網路選項，您可以在建立環境之後開始使用 Apache 氣流使用者介面。

您需要採取下列步驟來設定使用者的存取權限，以及環境使用其他 AWS 服務的權限。

1. 新增權限。Amazon MWAA 需要許可才能使用其他 AWS 服務。當您建立環境時，Amazon MWAA 會建立一個服務連結角色，允許該角色對亞馬遜彈性容器登錄 (Amazon ECR)、CloudWatch 日誌和 Amazon EC2 使用特定 IAM 動作。

   您可以新增對這些服務使用其他動作或使用其他 AWS 服務的權限，方法是將權限新增至您的執行角色。如需進一步了解，請參閱Amazon MWAA 執行角色。
2. 建立使用者策略。您可能需要為使用者建立多個 IAM 政策，以設定對環境和 Apache Airflow 使用者介面的存取權限。如需進一步了解，請參閱存取 Amazon MWAA 環境。

設定私人網路

如果您為 Apache Airflow Web 伺服器選擇私人網路選項，則需要為使用者設定存取權限、允許您的環境使用其他 AWS 服務，以及建立機制以從電腦存取 Amazon VPC 中的資源。

1. 新增權限。Amazon MWAA 需要許可才能使用其他 AWS 服務。當您建立環境時，Amazon MWAA 會建立一個服務連結角色，允許該角色對亞馬遜彈性容器登錄 (Amazon ECR)、CloudWatch 日誌和 Amazon EC2 使用特定 IAM 動作。

   您可以新增對這些服務使用其他動作或使用其他 AWS 服務的權限，方法是將權限新增至您的執行角色。如需進一步了解，請參閱Amazon MWAA 執行角色。
2. 建立使用者策略。您可能需要為使用者建立多個 IAM 政策，以設定對環境和 Apache Airflow 使用者介面的存取權限。如需進一步了解，請參閱存取 Amazon MWAA 環境。
3. 啟用網路存取。您需要在 Amazon VPC 中建立一個機制，以連接到 Apache 氣流網頁伺服器的 VPC 擬私人雲端端點 (AWS PrivateLink)。例如，透過使用 AWS Client VPN.

存取 Apache 氣流網頁伺服器的 VPC 私人雲端端點 (私人網路存取)

如果您選擇了私人網路選項，則需要在 Amazon VPC 中建立一個機制，以存取 Apache 氣流網頁伺服器的 VPC 擬私人雲端端點 (AWS PrivateLink)。對於這些資源，我們建議您使用與 Amazon MWAA 環境相同的 Amazon VPC、VPC 安全群組和私有子網路。

若要深入了解，請參閱管理 VPC 端點的存取權。

訪問阿帕奇氣流用戶界面

建立環境之後，Apache 氣流主控台上的亞馬遜管理工作流程即可使用 Apache 氣流使用者介面連結。您可以使用亞馬遜 MWAA 主控台在 Apache 氣流使用者介面中檢視和叫用 DAG，或使用亞馬遜 MWAA API 取得權杖並呼叫 DAG。本節說明存取 Apache 氣流使用者介面所需的權限、如何產生權杖以直接在命令殼層中進行 Amazon MWAA API 呼叫，以及 Apache 氣流 CLI 中支援的命令。

主題

先決條件

下節說明使用本節中指令和指令碼所需的初步步驟。

Access (存取)

- AWSAWS Identity and Access Management(IAM) 中的帳戶存取權限，可存取中的亞馬遜 MWAA 許可政策阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess。
- AWS在AWS Identity and Access Management (IAM) 中存取亞馬遜 MWAA 許可政策的帳戶完整的 API 和控制台訪問策略：亞馬遜 FullApiAccess。

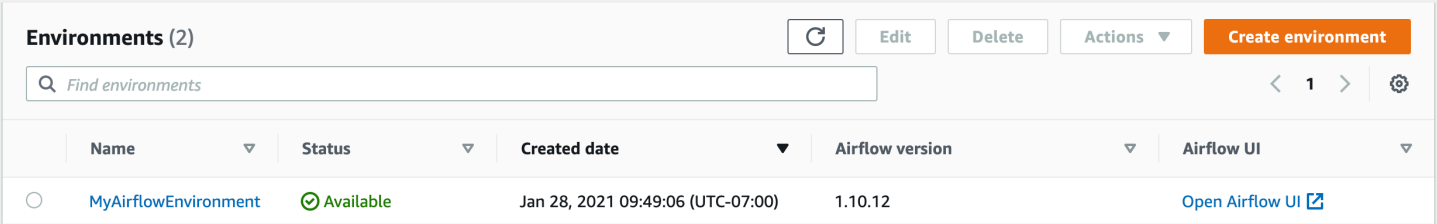
AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成本頁面的步驟，您需要執行下列操作：

- AWS CLI— 安裝第 2 版
- AWS CLI— 快速配置`aws configure`.

開啟 Airflow

下圖顯示了亞馬遜 MWAA 主控台上的 Apache 氣流使用者介面的連結。



登錄到阿帕奇氣流

您需要AWS Identity and Access Management (IAM) 中的AWS帳戶阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess許可，才能檢視您的 Apache 氣流使用者介面。

若要存取您的 Apache

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇「開啟氣流 UI」。

創建一個 Apache 氣流網絡登錄令牌

您可以使用此頁面上的命令產生網頁登入權杖，然後直接在命令殼層中呼叫 Apache Airflow API 的 Amazon 受管工作流程。例如，您可以取得權杖，然後使用 Amazon MWAA API 以程式設計方式部署 DAG。下一節包含使用 bash 指令碼AWS CLI、POST API 要求或 Python 指令碼建立 Apache 氣流網頁登入權杖的步驟。響應中返回的令牌有效期為 60 秒。

內容

- 使用 Python 指令碼
- 後續步驟？

先決條件

以下段落說明使用此頁面上的命令和命令檔所需的初步步驟。

Access (存取)

- AWSAWS Identity and Access Management(IAM) 中的帳戶存取權限，可存取中的亞馬遜 MWAA 許可政策阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess。
- AWS在AWS Identity and Access Management (IAM) 中存取亞馬遜 MWAA 許可政策的帳戶完整的 API 和控制台訪問策略：亞馬遜 FullApiAccess。

AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面上的步驟，您需要執行下列操作：

- AWS CLI— 安裝第 2 版
- AWS CLI— 快速配置`aws configure`.

使用 AWS CLI

下列範例會使用中的create-web-login-token命令AWS CLI來建立 Apache 氣流網頁登入權杖。

```
aws mwaa create-web-login-token --name YOUR_ENVIRONMENT_NAME
```

使用 bash 腳本

下列範例使用 bash 指令碼呼叫中的create-web-login-token命令，AWS CLI以建立 Apache 氣流網頁登入權杖。

1. 複製下列程式碼範例的內容，並在本機儲存為`get-web-token.sh`。

```
#!/bin/bash
```

```
HOST=YOUR_HOST_NAME
YOUR_URL=https://$HOST/aws_mwaa/aws-console-sso?login=true#
WEB_TOKEN=$(aws mwaa create-web-login-token --name YOUR_ENVIRONMENT_NAME --query
 WebToken --output text)
echo $YOUR_URL$WEB_TOKEN
```

2. 將##的預留位置替換為YOUR_HOST_NAME和YOUR_ENVIRONMENT_NAME。例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. (可選) macOS 和 Linux 使用者可能需要執行下列命令，以確保指令碼可執行。

```
chmod +x get-web-token.sh
```

4. 執行下列指令碼可以取得 Web 登錄權杖。

```
./get-web-token.sh
```

5. 您應在命令提示下看到以下內容：

```
https://123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com/
aws_mwaa/aws-console-sso?login=true#{your-web-login-token}
```

使用發布 API 請求

下列範例會使用 POST API 要求來建立 Apache 氣流網頁登入權杖。

1. 複製以下 URL 並將您的 REST API 用戶端的 URL 欄位中貼到您的 URL 欄位中。

```
https://YOUR_HOST_NAME/aws_mwaa/aws-console-sso?login=true#WebToken
```

2. 以##取代預留位置YOUR_HOST_NAME。例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. 複製下列 JSON 並貼到您的 REST API 用戶端的內文欄位中。

```
{
  "name": "YOUR_ENVIRONMENT_NAME"
```

```
}
```

4. 以##取代預留位置YOUR_ENVIRONMENT_NAME。
5. 在授權字段中添加鍵值對。例如，如果您使用郵遞員，請選擇「AWS簽名」，然後輸入您的：
   - AWS_ACCESS_KEY_ID 中的 AccessKey
   - AWS_SECRET_ACCESS_KEY 中的 SecretKey
6. 您應該會看到下列回應：

```
{
  "webToken": "<Short-lived token generated for enabling access to the Apache
 Airflow Webserver UI>",
  "webServerHostname": "<Hostname for the WebServer of the environment>"
}
```

使用 Python 指令碼

下列範例會在 Python 指令碼中使用 boto3 Create_web_login_token 方法來建立 Apache 氣流網路登入權杖。您可以在亞馬遜 MWAA 以外執行此指令碼。您只需安裝第 2 版 您可能想要建立虛擬環境來安裝程式庫。它假設您已經為您的帳戶配置了AWS身份驗證憑據。

1. 複製下列程式碼範例的內容，並在本機儲存為create-web-login-token.py。

```
import boto3
mwaa = boto3.client('mwaa')
response = mwaa.create_web_login_token(
    Name="YOUR_ENVIRONMENT_NAME"
)
webServerHostName = response["WebServerHostname"]
webToken = response["WebToken"]
airflowUIUrl = 'https://{0}/aws_mwaa/aws-console-sso?
login=true#{1}'.format(webServerHostName, webToken)
print("Here is your Airflow UI URL: ")
print(airflowUIUrl)
```

2. 以##取代預留位置YOUR_ENVIRONMENT_NAME。
3. 執行下列指令碼可以取得 Web 登錄權杖。

```
python3 create-web-login-token.py
```

後續步驟？

- 探索用於在建立網頁登入權杖的 Amazon MWAA API 操作CreateWebLoginToken。

創建一個阿帕奇氣流 CLI 令牌

您可以使用此頁面上的命令產生 CLI 權杖，然後直接在命令殼層中呼叫 Apache Airflow API 的 Amazon 受管工作流程。例如，您可以取得權杖，然後使用 Amazon MWAA API 以程式設計方式部署 DAG。下一節包含使用、捲曲指令碼AWS CLI、Python 指令碼或 bash 指令碼建立 Apache 氣流 CLI 權杖的步驟。響應中返回的令牌有效期為 60 秒。

> **Note**
>
> 該令AWS CLI牌旨在用於替代同步 shell 操作，而不是異步 API 命令。因此，可用的並發是有限的。為了確保 Web 服務器對用戶保持響應，建議在上一個AWS CLI請求成功完成之前不要打開新請求。

內容

先決條件

以下段落說明使用此頁面上的命令和命令檔所需的初步步驟。

Access (存取)

- AWS中的帳戶存取AWS Identity and Access Management (IAM)，可存取中的亞馬遜 MWAA 許可政策阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess。

- AWS在AWS Identity and Access Management (IAM) 中存取亞馬遜 MWAA 許可政策的帳戶完整的 API 和控制台訪問策略：亞馬遜 FullApiAccess。

AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面的步驟，您需要執行下列操作：

- AWS CLI— 安裝第 2 版
- AWS CLI— 快速配置`aws configure`.

使用 AWS CLI

下列範例會使用中的create-cli-token命令AWS CLI來建立 Apache 氣流 CLI 權杖。

```
aws mwaa create-cli-token --name YOUR_ENVIRONMENT_NAME
```

使用捲曲腳本

下列範例使用 curl 指令碼呼叫中的create-web-login-token命令，透過 Apache 氣流網頁伺服器上的端點叫用 Apache 氣流 CLI。AWS CLI

Apache Airflow v2

1. 從文本文件複製 curl 語句並將其粘貼到命令 shell 中。

> **Note**
>
> 將其複製到剪貼板後，您可能需要從 shell 菜單中使用 「編輯」>「粘貼」。

```
CLI_JSON=$(aws mwaa --region YOUR_REGION create-cli-token --
name YOUR_ENVIRONMENT_NAME) \
  && CLI_TOKEN=$(echo $CLI_JSON | jq -r '.CliToken') \
  && WEB_SERVER_HOSTNAME=$(echo $CLI_JSON | jq -r '.WebServerHostname') \
  && CLI_RESULTS=$(curl --request POST "https://$WEB_SERVER_HOSTNAME/aws_mwaa/
cli" \
  --header "Authorization: Bearer $CLI_TOKEN" \
```

```
  --header "Content-Type: text/plain" \
  --data-raw "dags trigger YOUR_DAG_NAME") \
  && echo "Output:" \
  && echo $CLI_RESULTS | jq -r '.stdout' | base64 --decode \
  && echo "Errors:" \
  && echo $CLI_RESULTS | jq -r '.stderr' | base64 --decode
```

2. 將的預留位置替換為環境YOUR_DAG_NAME、和的AWS區域YOUR_ENVIRONMENT_NAME。YOUR_REGION例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. 您應在命令提示中看到以下內容：

```
{
  "stderr":"<STDERR of the CLI execution (if any), base64 encoded>",
  "stdout":"<STDOUT of the CLI execution, base64 encoded>"
}
```

Apache Airflow v1

1. 從文本文件複製 cURL 語句並將其粘貼到命令 shell 中。

> **Note**
>
> 將其複製到剪貼板後，您可能需要從 shell 菜單中使用 「編輯」>「粘貼」。

```
CLI_JSON=$(aws mwaa --region YOUR_REGION create-cli-token --
name YOUR_ENVIRONMENT_NAME) \
  && CLI_TOKEN=$(echo $CLI_JSON | jq -r '.CliToken') \
  && WEB_SERVER_HOSTNAME=$(echo $CLI_JSON | jq -r '.WebServerHostname') \
  && CLI_RESULTS=$(curl --request POST "https://$WEB_SERVER_HOSTNAME/aws_mwaa/
cli" \
  --header "Authorization: Bearer $CLI_TOKEN" \
  --header "Content-Type: text/plain" \
  --data-raw "trigger_dag YOUR_DAG_NAME") \
  && echo "Output:" \
  && echo $CLI_RESULTS | jq -r '.stdout' | base64 --decode \
```

```
  && echo "Errors:" \
  && echo $CLI_RESULTS | jq -r '.stderr' | base64 --decode
```

2. 將的預留位置替換為環境YOUR_DAG_NAME、和的AWS區域YOUR_HOST_NAME。YOUR_REGION例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. 您應在命令提示中看到以下內容：

```
{
  "stderr":"<STDERR of the CLI execution (if any), base64 encoded>",
  "stdout":"<STDOUT of the CLI execution, base64 encoded>"
}
```

4. 取代YOUR_ENVIRONMENT_NAME和的預留位置YOUR_DAG_NAME。

使用 bash 腳本

下列範例使用 bash 指令碼呼叫中的create-cli-token命令，AWS CLI以建立 Apache 氣流 CLI 權杖。

Apache Airflow v2

1. 複製下列程式碼範例的內容，並在本機儲存為get-cli-token.sh。

```
# brew install jq
  aws mwaa create-cli-token --name YOUR_ENVIRONMENT_NAME | export CLI_TOKEN=$(jq
 -r .CliToken) && curl --request POST "https://YOUR_HOST_NAME/aws_mwaa/cli" \
      --header "Authorization: Bearer $CLI_TOKEN" \
      --header "Content-Type: text/plain" \
      --data-raw "dags trigger YOUR_DAG_NAME"
```

2. 將##的預留位置替換為YOUR_ENVIRONMENT_NAMEYOUR_HOST_NAME、和YOUR_DAG_NAME。例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. (選用) macOS 和 Linux 使用者可能需要執行下列命令，以確保指令碼可執行。

```
chmod +x get-cli-token.sh
```

4. 執行下列指令碼來建立 Apache 氣流 CLI 權杖。

```
./get-cli-token.sh
```

Apache Airflow v1

1. 複製下列程式碼範例的內容，並在本機儲存為get-cli-token.sh。

```
# brew install jq
  aws mwaa create-cli-token --name YOUR_ENVIRONMENT_NAME | export CLI_TOKEN=$(jq
 -r .CliToken) && curl --request POST "https://YOUR_HOST_NAME/aws_mwaa/cli" \
      --header "Authorization: Bearer $CLI_TOKEN" \
      --header "Content-Type: text/plain" \
      --data-raw "trigger_dag YOUR_DAG_NAME"
```

2. 將##的預留位置替換為YOUR_ENVIRONMENT_NAMEYOUR_HOST_NAME、和YOUR_DAG_NAME。例如，公用網路的主機名稱可能如下所示 (沒有 https://)：

```
123456a0-0101-2020-9e11-1b159eec9000.c2.us-east-1.airflow.amazonaws.com
```

3. (選用) macOS 和 Linux 使用者可能需要執行下列命令，以確保指令碼可執行。

```
chmod +x get-cli-token.sh
```

4. 執行下列指令碼來建立 Apache 氣流 CLI 權杖。

```
./get-cli-token.sh
```

使用 Python 指令碼

下列範例會在 Python 指令碼中使用 boto3 Create_cli_token 方法來建立 Apache 氣流 CLI 權杖並觸發 DAG。您可以在亞馬遜 MWAA 以外執行此指令碼。您只需安裝第 3 版 您可能想要建立虛擬環境來安裝程式庫。它假設您已經為您的帳戶配置了AWS身份驗證憑據。

Apache Airflow v2

1. 複製下列程式碼範例的內容，並在本機儲存為create-cli-token.py。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
import boto3
import json
import requests
import base64

mwaa_env_name = 'YOUR_ENVIRONMENT_NAME'
dag_name = 'YOUR_DAG_NAME'
mwaa_cli_command = 'dags trigger'

client = boto3.client('mwaa')

mwaa_cli_token = client.create_cli_token(
    Name=mwaa_env_name
)

mwaa_auth_token = 'Bearer ' + mwaa_cli_token['CliToken']
mwaa_webserver_hostname = 'https://{0}/aws_mwaa/
cli'.format(mwaa_cli_token['WebServerHostname'])
raw_data = '{0} {1}'.format(mwaa_cli_command, dag_name)

mwaa_response = requests.post(
        mwaa_webserver_hostname,
        headers={
```

```
            'Authorization': mwaa_auth_token,
            'Content-Type': 'text/plain'
            },
        data=raw_data
        )

mwaa_std_err_message = base64.b64decode(mwaa_response.json()
['stderr']).decode('utf8')
mwaa_std_out_message = base64.b64decode(mwaa_response.json()
['stdout']).decode('utf8')

print(mwaa_response.status_code)
print(mwaa_std_err_message)
print(mwaa_std_out_message)
```

2. 取代YOUR_ENVIRONMENT_NAME和的預留位置YOUR_DAG_NAME。
3. 執行下列指令碼來建立 Apache 氣流 CLI 權杖。

```
python3 create-cli-token.py
```

Apache Airflow v1

1. 複製下列程式碼範例的內容，並在本機儲存為create-cli-token.py。

```
import boto3
import json
import requests
import base64

mwaa_env_name = 'YOUR_ENVIRONMENT_NAME'
dag_name = 'YOUR_DAG_NAME'
mwaa_cli_command = 'trigger_dag'

client = boto3.client('mwaa')

mwaa_cli_token = client.create_cli_token(
    Name=mwaa_env_name
)

mwaa_auth_token = 'Bearer ' + mwaa_cli_token['CliToken']
mwaa_webserver_hostname = 'https://{0}/aws_mwaa/
cli'.format(mwaa_cli_token['WebServerHostname'])
```

```
raw_data = '{0} {1}'.format(mwaa_cli_command, dag_name)

mwaa_response = requests.post(
        mwaa_webserver_hostname,
        headers={
            'Authorization': mwaa_auth_token,
            'Content-Type': 'text/plain'
            },
        data=raw_data
        )

mwaa_std_err_message = base64.b64decode(mwaa_response.json()
['stderr']).decode('utf8')
mwaa_std_out_message = base64.b64decode(mwaa_response.json()
['stdout']).decode('utf8')

print(mwaa_response.status_code)
print(mwaa_std_err_message)
print(mwaa_std_out_message)
```

2. 取代YOUR_ENVIRONMENT_NAME和的預留位置YOUR_DAG_NAME。
3. 執行下列指令碼來建立 Apache 氣流 CLI 權杖。

```
python3 create-cli-token.py
```

後續步驟？

- 探索用於在建立 CLI 權杖的 Amazon MWAA API 操作CreateCliToken。

阿帕奇氣流 CLI 命令參考

本頁說明在 Amazon 管理的 Apache 氣流程中支援和不支援的 Apache 氣流 CLI 命令。

內容

必要條件

以下段落說明使用此頁面上的命令和命令檔所需的初步步驟。

存取

- AWS 中的帳戶存取 AWS Identity and Access Management (IAM)，可存取中的 Amazon MWAA 許可政策。阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess
- AWS 在 AWS Identity and Access Management (IAM) 中存取 Amazon MWAA 許可政策的帳戶。完整的 API 和控制台訪問策略：亞馬遜 FullApiAccess

AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您使用命令列殼層中的命令與 AWS 服務互動。若要完成此頁面上的步驟，您需要下列項目：

- AWS CLI — 安裝版本 2.
- AWS CLI — 快速配置 `aws configure`.

v2 中有什麼變化

- 新增：氣流 CLI 命令結構。Apache 氣流 v2 CLI 的組織，以便相關的命令被組合在一起作為子命令，這意味著你需要更新 Apache 氣流 v1 腳本，如果你想升級到 Apache 氣流 V2。例如，`unpause`在阿帕奇氣流 V1 現在是`dags unpause`在阿帕奇氣流 V2。若要深入了解，請參閱 Apache 氣流參考指南中的 2 中的氣流 CLI 變更。

支援的 CLI 指令

下一節列出 Amazon MWAA 上可用的 Apache 氣流 CLI 命令。

支援的命令

Apache Airflow v2

次要版本	Command	
V2.0+	作弊表	
V2.0+	連接:加入	
V2.0+	連線:刪除	
版本 2.2 以上（註釋）	DAG 回填	
V2.0+	刪除 DAG	
版本 2.2 以上（註釋）	爸爸列表	
V2.0+	DAGS 列表工作	
版本	糞便 list-import-errors	
版本 2.2 以上（註釋）	DAGS 列表運行	
版本 2.2 以上（註釋）	DAG 下一次執行	
V2.0+	爸爸暫停	
V2.0+	爸爸報告	
2.4 版以上	DAG 重序列化	
V2.0+	爸爸展示	
V2.0+	丹格斯州	
V2.0+	堵嘴測試	

次要版本	Command
V2.0+	DAG 觸發器
V2.0+	爸爸取消暫停
2.4 版以上	數據庫清潔
V2.0+	供應商行為
V2.0+	供應商取得
V2.0+	供應商掛鉤
V2.0+	提供商鏈接
V2.0+	提供商列表
v2.8+	提供者通知
版本	供應商秘密
v2.7+	提供商更加耐用
V2.0+	供應商部件
版本	角色增益集
版本	角色德爾燙髮
版本	角色增益集
V2.0+	角色清單
V2.0+	任務清除
V2.0+	任務失敗-DEP
V2.0+	工作清單
V2.0+	工作彩現

次要版本	Command
V2.0+	工作狀態
V2.0+	任務 states-for-dag-run
V2.0+	任務測試
V2.0+	變數:刪除
V2.0+	變量獲取
V2.0+	變數設定
V2.0+	變數清單
V2.0+	version

使用剖析 DAG 的命令

如果您的環境正在執行 Apache 氣流 v1.10.12 或 v2.0.2，如果 DAG 使用的外掛程式依賴透過下列方式安裝的套件，剖析 DAG 的 CLI 命令將會失敗：`requirements.txt`

阿帕奇氣流 v2.0.2

- `dags backfill`
- `dags list`
- `dags list-runs`
- `dags next-execution`

如果您的 DAG 不使用相依於透過`requirements.txt`.

範本程式碼

下一節包含使用 Apache 氣流 CLI 的不同方式的範例。

設置，獲取或刪除阿帕奇氣流 v2 變量

您可以使用下列範例程式碼來設定、取得或刪除格式的變數<script> <mwaa env name> get | set | delete <variable> <variable value> </variable> </variable>。

```
[ $# -eq 0 ] && echo "Usage: $0 MWAA environment name " && exit

if [[ $2 == "" ]]; then
    dag="variables list"

elif  [ $2 == "get" ] ||  [ $2 == "delete" ] ||  [ $2 == "set" ]; then
    dag="variables $2 $3 $4 $5"

else
    echo "Not a valid command"
    exit 1
fi

CLI_JSON=$(aws mwaa --region $AWS_REGION create-cli-token --name $1) \
    && CLI_TOKEN=$(echo $CLI_JSON | jq -r '.CliToken') \
    && WEB_SERVER_HOSTNAME=$(echo $CLI_JSON | jq -r '.WebServerHostname') \
    && CLI_RESULTS=$(curl --request POST "https://$WEB_SERVER_HOSTNAME/aws_mwaa/cli" \
    --header "Authorization: Bearer $CLI_TOKEN" \
    --header "Content-Type: text/plain" \
    --data-raw "$dag" ) \
    && echo "Output:" \
    && echo $CLI_RESULTS | jq -r '.stdout' | base64 --decode \
    && echo "Errors:" \
    && echo $CLI_RESULTS | jq -r '.stderr' | base64 --decode
```

觸發 DAG 時新增組態

您可以使用下列範例程式碼搭配 Apache 氣流 v1 和 Apache 氣流 v2，在觸發 DAG 時新增組態，例如airflow trigger_dag 'dag_name' —conf '{"key":"value"}'.

```
import boto3
import json
import requests
import base64

mwaa_env_name = 'YOUR_ENVIRONMENT_NAME'
dag_name = 'YOUR_DAG_NAME'
```

```
key = "YOUR_KEY"
value = "YOUR_VALUE"
conf = "{\"" + key + "\":\"" + value + "\"}"

client = boto3.client('mwaa')

mwaa_cli_token = client.create_cli_token(
  Name=mwaa_env_name
)

mwaa_auth_token = 'Bearer ' + mwaa_cli_token['CliToken']
mwaa_webserver_hostname = 'https://{0}/aws_mwaa/
cli'.format(mwaa_cli_token['WebServerHostname'])
raw_data = "trigger_dag {0} -c '{1}'".format(dag_name, conf)

mwaa_response = requests.post(
      mwaa_webserver_hostname,
      headers={
          'Authorization': mwaa_auth_token,
          'Content-Type': 'text/plain'
          },
      data=raw_data
      )

mwaa_std_err_message = base64.b64decode(mwaa_response.json()['stderr']).decode('utf8')
mwaa_std_out_message = base64.b64decode(mwaa_response.json()['stdout']).decode('utf8')

print(mwaa_response.status_code)
print(mwaa_std_err_message)
print(mwaa_std_out_message)
```

在 SSH 通道上執行 CLI 命令到防禦主機

下列範例顯示如何使用安全殼層通道代理伺服器對 Linux 防禦主機執行氣流 CLI 命令。

使用捲曲

1. ```
   ssh -D 8080 -f -C -q -N YOUR_USER@YOUR_BASTION_HOST
   ```
2. ```
   curl -x socks5h://0:8080 --request POST https://YOUR_HOST_NAME/aws_mwaa/cli --
   header YOUR_HEADERS --data-raw YOUR_CLI_COMMAND
   ```

範例 GitHub 和教 AWS 學課程

- 在 Amazon 管理的工作流程中使用 Apache 氣流 v2.0.2 參數和變量
- 通過命令行與 Amazon MWAA 上的阿帕奇氣流 v1.10.12 進行交互
- 交互式命令與阿帕奇氣流 v1.10.12 在 Amazon MWAA 和 Bash 運算符上 GitHub

管理與 Apache 氣流的連線

本節說明為 Apache Airflow 環境設定 Apache Airflow。

主題

Apache 氣流變數和連線概觀

在某些情況下，您可能想要為環境 (例如設定AWS檔) 指定其他連線或變數，或在 Apache Airflow 中繼存放區中的連線物件中新增執行角色，然後參考來自 DAG 內的連線。

- 自我管理的 Apache Airflow。在自我管理的 Apache 氣流安裝上，您可以在中設定 Apache 氣流組態選項`airflow.cfg`。

```
[secrets]
backend = airflow.providers.amazon.aws.secrets.secrets_manager.SecretsManagerBackend
backend_kwargs = {"connections_prefix" : "airflow/connections", "variables_prefix" :
 "airflow/variables"}
```

- 阿帕奇氣流在亞馬遜 MWAA。在亞馬遜 MWAA 上，您需要將這些組態設定新增為 Amazon MWAA 主控台上的 Apache 氣流組態選項。Apache Airflow 組態選項會以環境變數的形式寫入您的環境，並覆寫相同設定的所有其他現有組態。

Amazon MWAA 環境上安裝的 Apache 氣流供應商套件

當您建立新環境時，Amazon MWAA 會為 Apache 氣流 v2 及以上連線類型安裝提供者額外服務。安裝提供者套件可讓您在 Apache 氣流使用者介面中檢視連線類型。這也意味著您不需要將這些軟件包指定為`requirements.txt`文件中的 Python 依賴項。本頁列出由 Amazon MWAA 針對所有 Apache 氣流 v2 環境安裝的 Apache 氣流提供者套件。

> **Note**
>
> 對於 Apache 氣流 v2 及更高版本，Amazon MWAA 在執行後安裝守望台版本 2.0.1`pip3 install -r requirements.txt`，以確保與 CloudWatch 日誌記錄的兼容性不會被其他 Python 庫安裝覆蓋。

內容

阿帕奇氣流 v2.8.1 連接的提供程序包

當您在 Apache 氣流 v2.8.1 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

> **Note**
>
> 您可以指定最新的受支援版本`apache-airflow-providers-amazon`來升級此提供者。如需指定較新版本的詳細資訊，請參閱the section called “指定新的提供者套件”。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon[輔助核心]==8.16.0
郵政連接	apache-airflow-providers-postgres==5.10.0

連線類型	套件
FTP 連接	apache-airflow-providers-ftp==3.7.0
芹菜連接	apache-airflow-providers-celery==3.5.1
HTTP 連接	apache-airflow-providers-http==4.8.0
連線	apache-airflow-providers-imap==3.5.0
常見的 SQL	apache-airflow-providers-common-平方米
SQLite 連接	apache-airflow-providers-sqlite==3.7.0

阿帕奇氣流 v2.7.2 連線的提供者套件

當您在 Apache 氣流 v2.7.2 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

> **Note**
>
> 您可以指定最新的受支援版本`apache-airflow-providers-amazon`來升級此提供者。如需指定較新版本的詳細資訊，請參閱the section called “指定新的提供者套件”。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon[因果核心]==8.7.1
郵政連接	apache-airflow-providers-postgres==5.6.1
FTP 連接	apache-airflow-providers-ftp==3.5.2
芹菜連接	apache-airflow-providers-celery==3.3.4
HTTP 連接	apache-airflow-providers-http==4.5.2
連線	apache-airflow-providers-imap==3.3.2

連線類型	套件
常見的 SQL	apache-airflow-providers-common-平方米
SQLite 連接	apache-airflow-providers-sqlite==3.4.3

阿帕奇氣流 v2.6.3 連接的提供程序包

當您在 Apache 氣流 v2.6.3 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

> **Note**
>
> 您可以指定最新的受支援版本`apache-airflow-providers-amazon`來升級此提供者。如需指定較新版本的詳細資訊，請參閱the section called “指定新的提供者套件”。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon[因果核心]==8.2.0
郵政連接	apache-airflow-providers-postgres==5.5.1
FTP 連接	apache-airflow-providers-ftp==3.4.2
芹菜連接	apache-airflow-providers-celery==3.2.1
HTTP 連接	apache-airflow-providers-http==4.4.2
連線	apache-airflow-providers-imap==3.2.2
常見的 SQL	apache-airflow-providers-common-平方米
SQLite 連接	apache-airflow-providers-sqlite==3.4.2

阿帕奇氣流 v2.5.1 連接的提供程序包

當您在 Apache 氣流 v2.5.1 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

> **Note**
>
> 您可以指定最新的受支援版本`apache-airflow-providers-amazon`來升級此提供者。如需指定較新版本的詳細資訊，請參閱the section called “指定新的提供者套件”。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon==7.1.0
郵政連接	apache-airflow-providers-postgres==5.4.0
FTP 連接	apache-airflow-providers-ftp==3.3.0
芹菜連接	apache-airflow-providers-celery==3.1.0
HTTP 連接	apache-airflow-providers-http==4.1.1
連線	apache-airflow-providers-imap==3.1.1
常見的 SQL	apache-airflow-providers-common-平方米
SQLite 連接	apache-airflow-providers-sqlite==3.3.1

阿帕奇氣流 v2.4.3 連接的提供程序包

當您在 Apache 氣流 v2.4.3 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon==6.0.0
郵政連接	apache-airflow-providers-postgres==5.2.2

連線類型	套件
FTP 連接	apache-airflow-providers-ftp==3.1.0
芹菜連接	apache-airflow-providers-celery==3.0.0
HTTP 連接	apache-airflow-providers-http==4.0.0
連線	apache-airflow-providers-imap==3.0.0
常見的 SQL	apache-airflow-providers-common-平方米
SQLite 連接	apache-airflow-providers-sqlite==3.2.1

阿帕奇氣流 v2.2.2 連接的提供程序包

當您在 Apache 氣流 v2.2.2 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

連線類型	套件
AWS 連接	apache-airflow-providers-amazon==2.4.0
郵政連接	apache-airflow-providers-postgres==2.3.0
FTP 連接	apache-airflow-providers-ftp==2.0.1
芹菜連接	apache-airflow-providers-celery==2.1.0
HTTP 連接	apache-airflow-providers-http==2.0.1
連線	apache-airflow-providers-imap==2.0.1
SQLite 連接	apache-airflow-providers-sqlite==2.0.1

阿帕奇氣流 v2.0.2 連接的提供程序包

當您在 Apache 氣流 v2.0.2 中建立 Amazon MWAA 環境時，Amazon MWAA 會安裝用於 Apache 氣流連線的下列供應商套件。

連線類型	套件
畫面連接	apache-airflow-providers-tableau==1.0.0
數據庫連接	apache-airflow-providers-databricks==1.0.1
SSH 連線	apache-airflow-providers-ssh==1.3.0
郵政連接	apache-airflow-providers-postgres==1.0.2
碼頭連接	apache-airflow-providers-docker==1.2.0
甲骨文連接	apache-airflow-providers-oracle==1.1.0
普雷斯托連接	apache-airflow-providers-presto==1.0.2
SFTP 連線	apache-airflow-providers-sftp==1.2.0

指定新的提供者套件

從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一`--constraint`份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。

Apache 氣流限制檔案會指定 Apache 氣流發行時可用的提供者版本。然而，在許多情況下，較新的提供者與該版本的 Apache Airflow 相容。因為您必須使用條件約束來指定較新版本的提供者套件，因此您可以修改特定提供者版本的條件約束檔案：

1. 請從 https://raw.githubusercontent.com/apache/airflow/constraints-2.7.2/constraints-3.11.txt "下載版本特定條件約束檔案
2. 將約束檔案中的`apache-airflow-providers-amazon`版本修改為您要使用的版本。
3. 將修改後的限制檔案儲存到 Amazon MWAA 環境的 Amazon S3 Dags 資料夾中，例如，`constraints-3.11-updated.txt`
4. 指定您的需求，如下所示。

```
--constraint "/usr/local/airflow/dags/constraints-3.11-updated.txt"

apache-airflow-providers-amazon==version-number
```

> **Note**
>
> 如果您使用的是私有 Web 伺服器，建議您使用 Amazon MWAA 本機執行器將所需的程式庫封裝為 WHL 檔案。

連線類型概觀

阿帕奇氣流存儲連接作為連接 URI 字符串。它會在 Apache 氣流使用者介面中提供連線範本，以產生連線 URI 字串，而不論連線類型為何。如果 Apache Airflow UI 中沒有連線範本，則可以使用替代連線範本來產生此連線 URI 字串，例如使用 HTTP 連線範本。主要的差異在於 URI 前綴，例如`my-conn-type://` Apache 氣流提供者通常會忽略連接的 URI 前綴。本頁說明如何針對不同的連線類型互換使用 Apache Airflow UI 中的連線範本。

> **Warning**
>
> 請勿覆寫亞馬遜 MWAA 中的`aws_default`連線。Amazon MWAA 使用此連線來執行各種關鍵任務，例如收集任務日誌。覆寫此連線可能會導致資料遺失和中斷您的環境可用性。

主題

- 連接 URI 字串範例
- 範例連線範例
- 使用 HTTP 連接模板進行 Jdbc 連接的示例

連接 URI 字串範例

下面的例子顯示了 MySQL 連接類型的連接 URI 字符串。

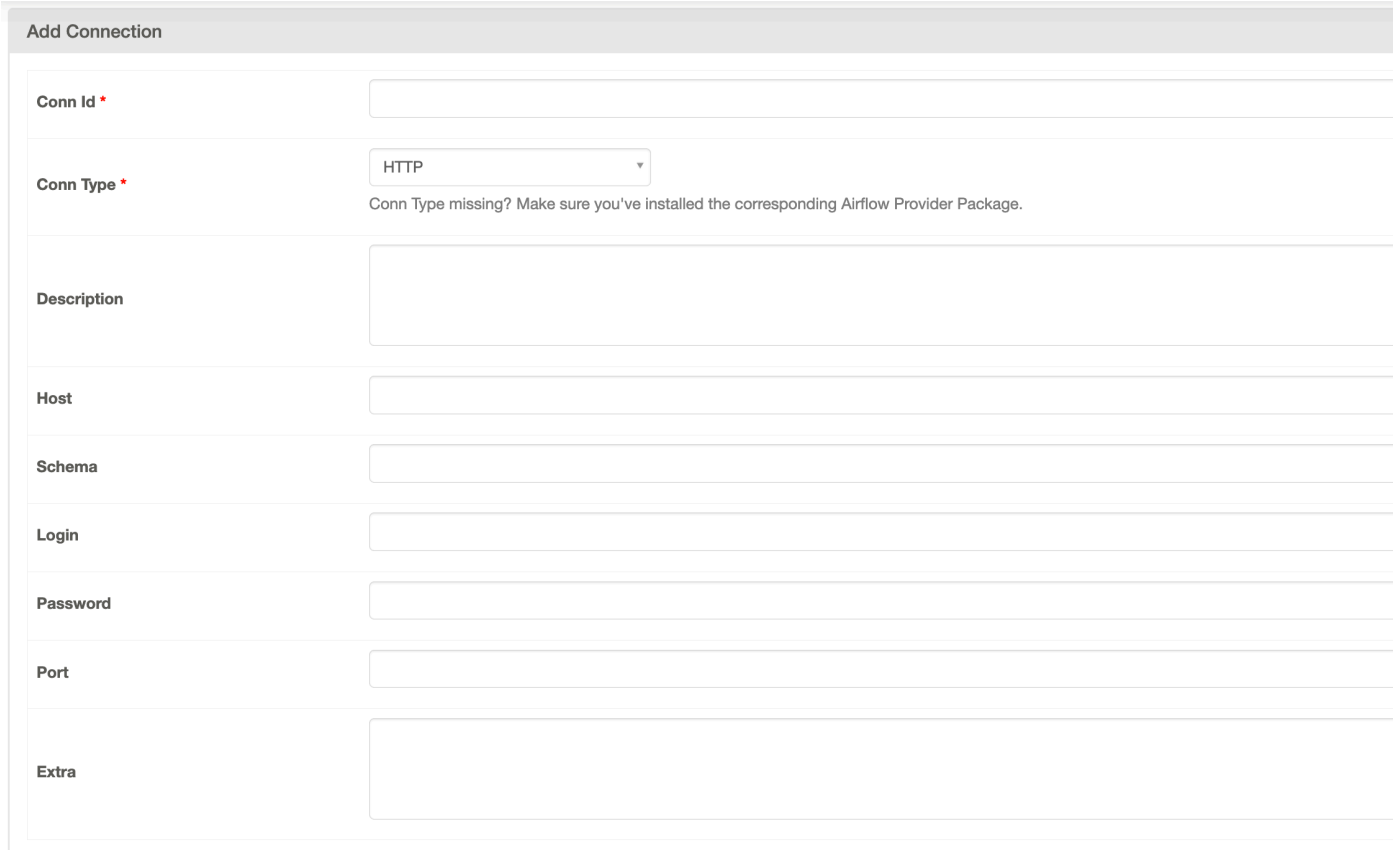
```
'mysql://288888a0-50a0-888-9a88-1a111aaa0000.a1.us-east-1.airflow.amazonaws.com
%2Fhome?role_arn=arn%3Aaws%3Aiam%3A%3A001122332255%3Arole%2Fservice-role%2FAmazonMWAA-
MyAirflowEnvironment-iAaaaA&region_name=us-east-1'
```

範例連線範例

下列範例示範 Apache 氣流 UI 中的 HTTP 連線範本。

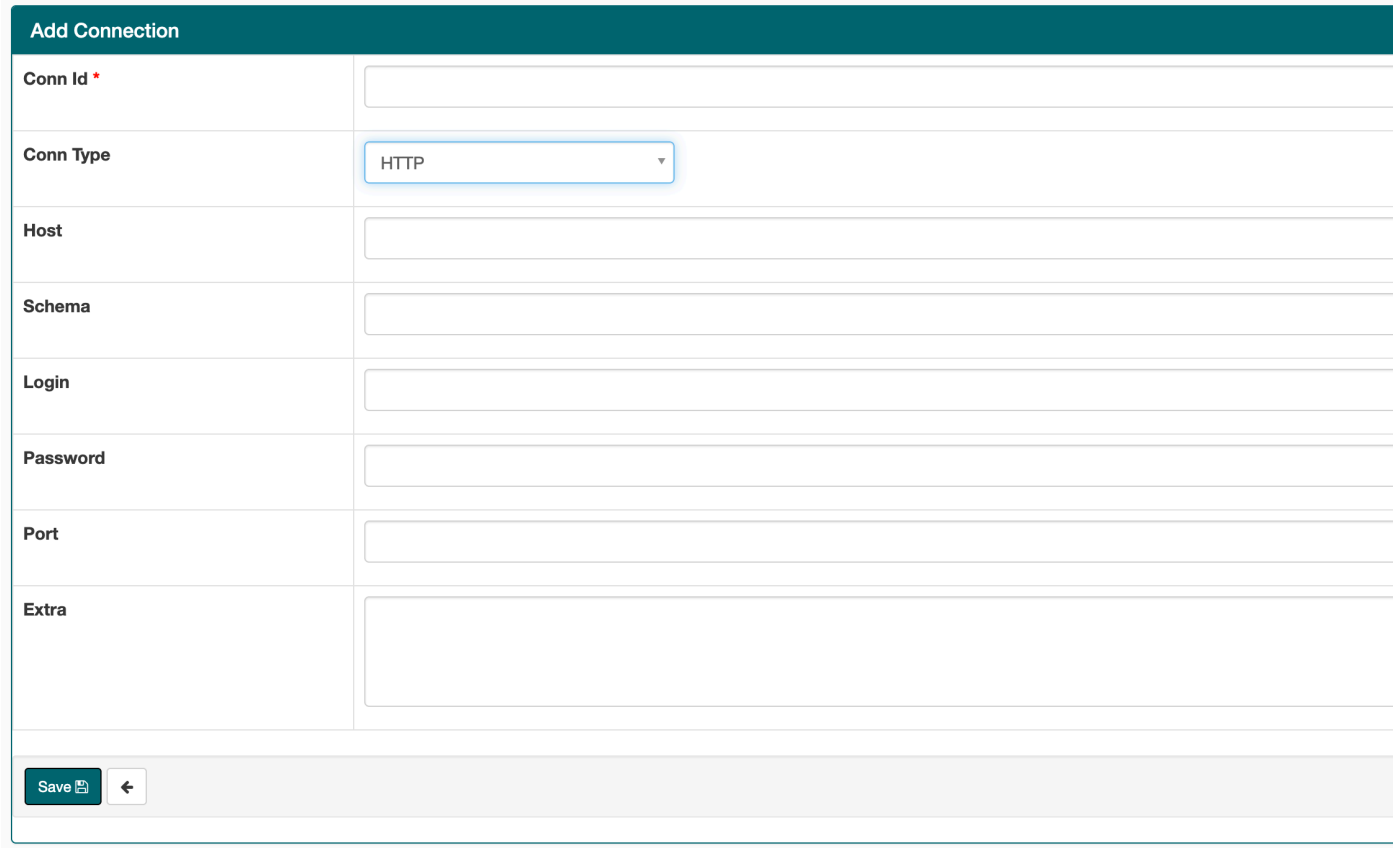
Apache Airflow v2

下面的例子顯示了在阿帕奇氣流用戶界面阿帕奇氣流 V2 的 HTTP 連接模板。



Apache Airflow v1

下面的例子顯示了在阿帕奇氣流用戶界面阿帕奇氣流 V1 的 HTTP 連接模板。



使用 HTTP 連接模板進行 Jdbc 連接的示例

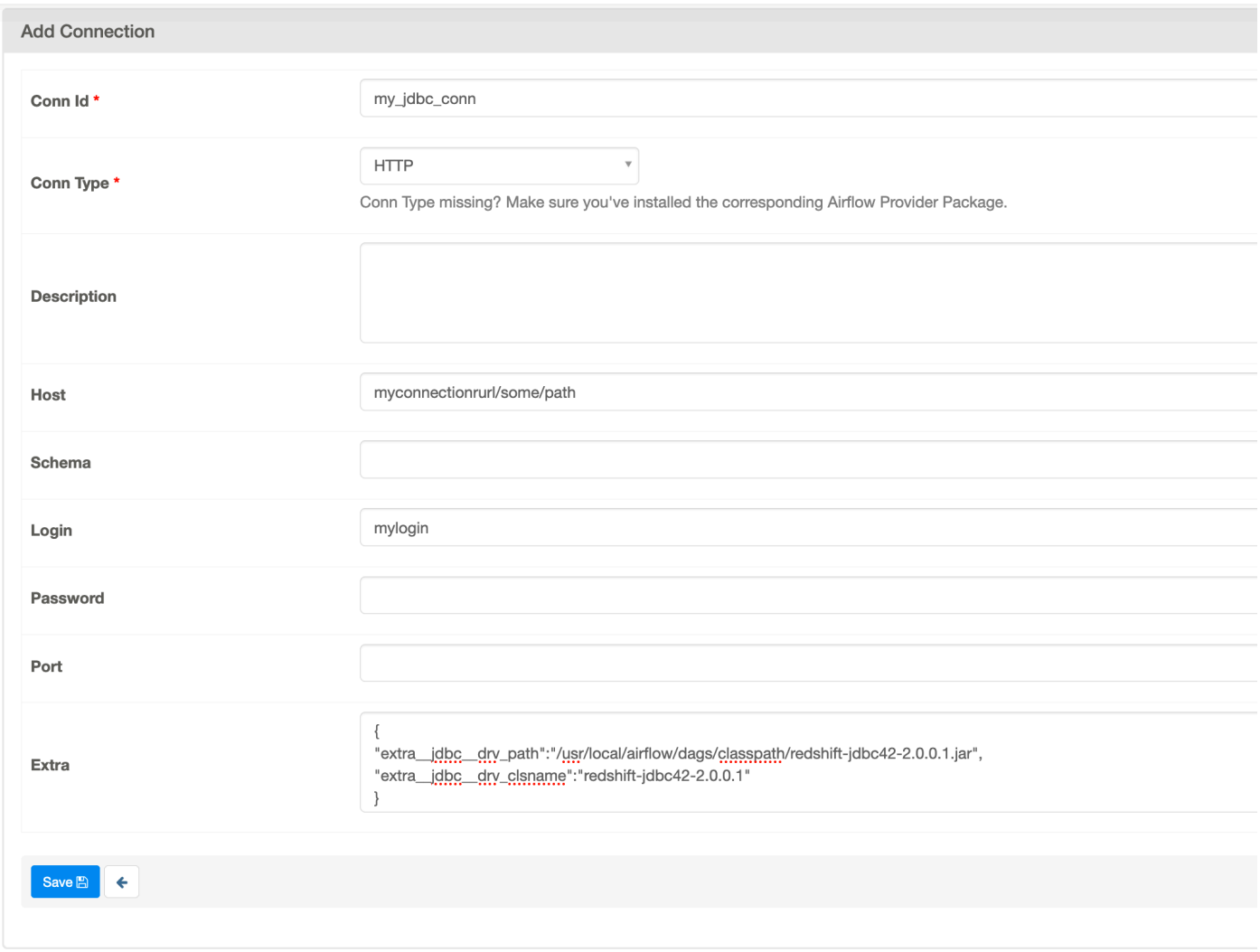
下列範例會示範如何在 Apache 氣流 v2.0.2 中使用 Jd bc 連線類型的 HTTP 連線範本，以及在 Apache 氣流 UI 中的阿帕奇氣流 v1.10.12 的 Jd bc 連線範本中使用相同的值。

Apache Airflow v2

下列範例顯示由 Apache 氣流產生的連線 URI 字串，作為本節中的範例。

```
http://myconnectionurl/some/path&login=mylogin&extra__jdbc__dry__path=usr/local/
airflow/dags/classpath/redshif-
jdbc42-2.0.0.1.jar&extra__jdbc__dry__clsname=redshift-jdbc42-2.0.0.1
```

下面的例子演示了如何使用 HTTP 連接模板的 Jdbc 連接阿帕奇氣流 V2 在阿帕奇氣流 UI。

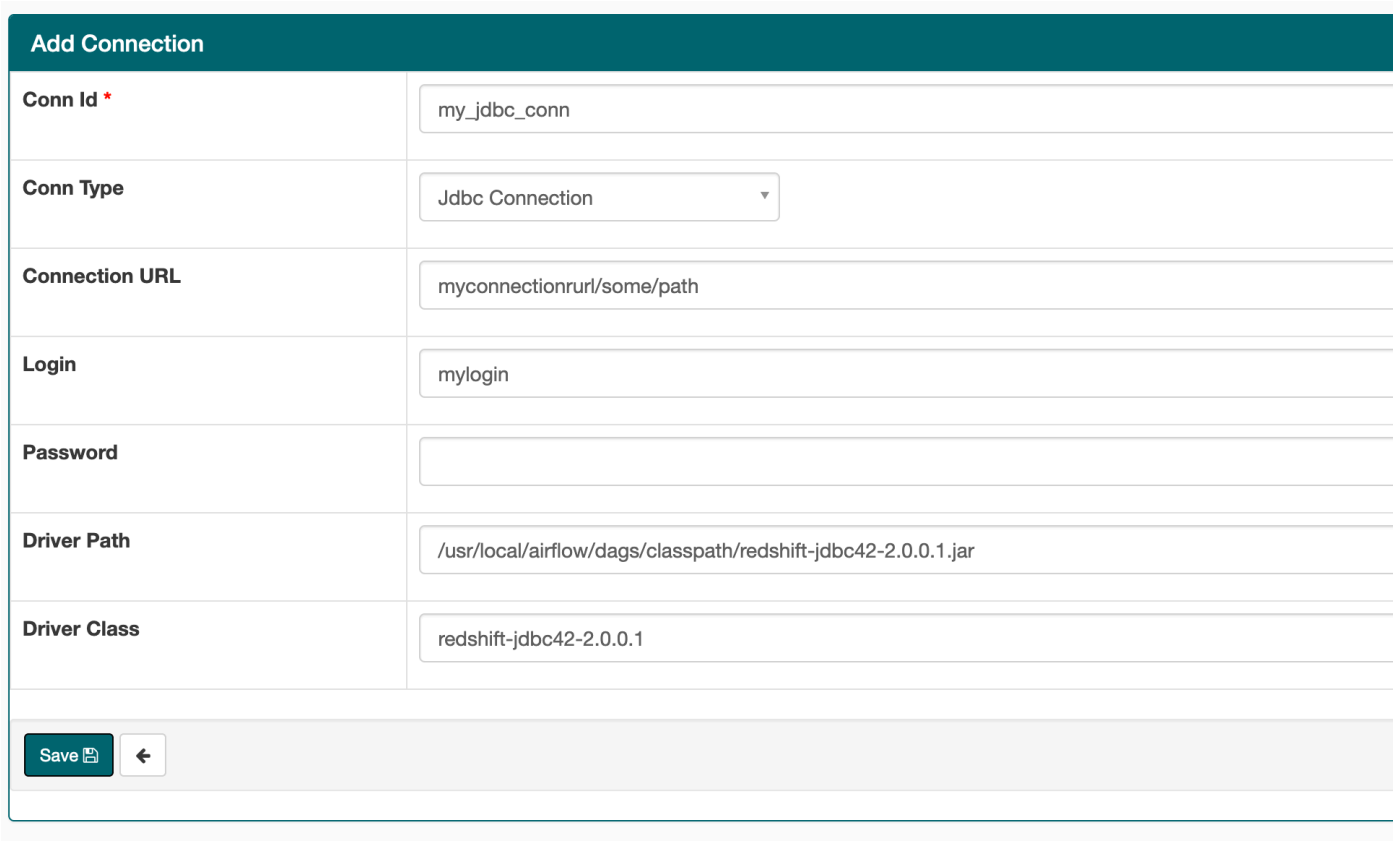


Apache Airflow v1

下列範例顯示由 Apache 氣流產生的連線 URI 字串，作為本節中的範例。

```
jdbc://myconnectionurl/some/path&login=mylogin&extra__jdbc__dry__path=usr/local/
airflow/dags/classpath/redshif-
jdbc42-2.0.0.1.jar&extra__jdbc__dry__clsname=redshift-jdbc42-2.0.0.1
```

下面的例子顯示了阿帕奇氣流 V1.10.12 在阿帕奇氣流用戶界面的 Jd bc 連接模板。



使用AWS Secrets Manager機密來設定 Apache Airflow 連線

AWS Secrets ManagerAmazon Managed Workflows (Apache Managed Workflows) 上 Managed Workflows 的 Managed Workflows 支援的 Apache Managed Workflows。本指南說明如何在適用AWS Secrets Manager於 Apache 氣流的亞馬遜管理工作流程上安全地儲存 Apache 氣流變數和 Apache 氣流連線的密碼。

> **Note**
>
> - 您需為建立的機密支付費用。如需 Secrets Manager 定價的詳細資訊，請參閱AWS定價。

內容

第一步：向亞馬遜 MWAA 提供訪問 Secrets Manager 密鑰的權限

Amazon MWAA 環境的執行角色需要對中的密鑰進行讀取存取AWS Secrets Manager。下列 IAM 政策允許使用AWS受管政SecretsManagerReadWrite策進行讀寫存取。

若要將政策附加至您的執行角色

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在「權限」窗格中選擇您的執行角色。
4. 選擇 Attach policies (連接政策)。
5. `SecretsManagerReadWrite`在「過濾器策略」文字欄位中輸入。
6. 選擇 Attach policy (連接政策)。

如果您不想使用AWS受管理的權限原則，您可以直接更新環境的執行角色，以允許任何層級的 Secrets Manager 資源存取權。例如，下列政策聲明會授予讀取存取權限給您在 Secrets Manager 中特定AWS區域中建立的所有密碼。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:012345678910:secret:*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
            "Resource": "*"
        }
    ]
}
```

步驟二：建立 Secrets Manager 後端做為 Apache 氣流組態選項

下節說明如何在 Amazon MWAA 主控台上為AWS Secrets Manager後端建立 Apache 氣流組態選項。如果您在中使用相同名稱的組態設定`airflow.cfg`，您在以下步驟中建立的組態將優先並覆寫組態設定。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇 **編輯** 。
4. 選擇 **下一步**。
5. 在氣流組態選項窗格中選擇新增自訂組態。新增下列索引鍵/值組：

    a. **`secrets.backend`**: **`airflow.providers.amazon.aws.secrets.secrets_manager.SecretsManagerBacken`**

    b. **`secrets.backend_kwargs`**：**`{"connections_prefix" : "airflow/connections", "variables_prefix" : "airflow/variables"}`**這將配置 Apache 氣流以查找連接字符串和變量位於`airflow/connections/*`和`airflow/variables/*`路徑。

    您可以使用查閱模式來減少 Amazon MWAA 代表您對機 Secrets Manager 進行的 API 呼叫數量。如果您未指定查詢模式，Apache Airflow 會在設定的後端搜尋所有連線和變數。透過指定模式，您可以縮小 Apache 氣流所看起來的可能路徑。這樣可以降低使用 Secrets Manager 與亞馬遜 MWAA 時的成本。

    若要指定參數表樣式，請指定`connections_lookup_pattern`和`variables_lookup_pattern`參數。這些參數接受一個RegEx字符串作為輸入。例如，若要尋找開頭為的密碼`test`，請輸入以下內容`secrets.backend_kwargs`：

    ```
    {
    ```

```
  "connections_prefix": "airflow/connections",
  "connections_lookup_pattern": "^test",
  "variables_prefix" : "airflow/variables",
  "variables_lookup_pattern": "^test"
}
```

> **Note**
>
> 若要使用connections_lookup_pattern和variables_lookup_pattern，您必須安裝 7.3.0 或更高apache-airflow-providers-amazon版本。如需將 Provder pacakges 更新為較新版本的詳細資訊，請參閱the section called “指定新的提供者套件”。

6. 選擇 儲存 。

第三步：生成一個 Apache 氣流AWS連接 URI 字符串

TTO 創建一個連接字符串，使用鍵盤上的「tab」鍵縮進連接對象中的鍵-值對。我們也建議您在 shell 會話中為extra對象創建一個變量。下一節將逐步引導您完成使用 Apache 氣流或 Python 指令碼為亞馬遜 MWAA 環境產生 Apache 氣流連線 URI 字串的步驟。

Apache Airflow CLI

下列殼層工作階段會使用您的本機 Airflow CLI 來產生連接字串。如果您沒有安裝 CLI，我們建議您使用 Python 指令碼。

1. 打開一個 Python 外殼會話：

```
python3
```

2. 輸入以下命令：

```
>>> import json
```

3. 輸入以下命令：

```
>>> from airflow.models.connection import Connection
```

4. 在 shell 會話中為extra對象創建一個變量。將*#### ARN ####*值取代為執行角色 ARN，以及*#### (##us-east-1) ####*。

```
>>> extra=json.dumps({'role_arn': 'YOUR_EXECUTION_ROLE_ARN', 'region_name':
 'YOUR_REGION'})
```

5. 創建連接對象。以 Apache 氣流連線myconn的名稱取代中的範例值。

```
>>> myconn = Connection(
```

6. 使用鍵盤上的「tab」鍵縮排連接對像中的以下每個鍵值對。以##取代樣本值。

   a. 指定AWS連線類型：

   ```
   ... conn_id='aws',
   ```

   b. 指定 Apache 氣流資料庫選項：

   ```
   ... conn_type='mysql',
   ```

   c. 指定阿帕奇氣流用戶界面網址亞馬遜 MWAA:

   ```
   ... host='288888a0-50a0-888-9a88-1a111aaa0000.a1.us-
   east-1.airflow.amazonaws.com/home',
   ```

   d. 指定要登入 Amazon MWAA 的AWS存取金鑰識別碼 (使用者名稱)：

   ```
   ... login='YOUR_AWS_ACCESS_KEY_ID',
   ```

   e. 指定要登入 Amazon MWAA 的AWS秘密存取金鑰 (密碼)：

   ```
   ... password='YOUR_AWS_SECRET_ACCESS_KEY',
   ```

   f. 指定extra殼層階段作業變數：

   ```
   ... extra=extra
   ```

   g. 關閉連接物件。

   ```
   ... )
   ```

7. 打印連接 URI 字符串：

```
>>> myconn.get_uri()
```

您應該在響應中看到連接 URI 字符串：

```
'mysql://288888a0-50a0-888-9a88-1a111aaa0000.a1.us-east-1.airflow.amazonaws.com
%2Fhome?role_arn=arn%3Aaws%3Aiam%3A%3A001122332255%3Arole%2Fservice-role
%2FAmazonMWAA-MyAirflowEnvironment-iAaaaA&region_name=us-east-1'
```

Python script

下面的 Python 腳本不需要阿帕奇氣流 CLI。

1. 複製下列程式碼範例的內容，並在本機儲存為mwaa_connection.py。

```
import urllib.parse

conn_type = 'YOUR_DB_OPTION'
host = 'YOUR_MWAA_AIRFLOW_UI_URL'
port = 'YOUR_PORT'
login = 'YOUR_AWS_ACCESS_KEY_ID'
password = 'YOUR_AWS_SECRET_ACCESS_KEY'
role_arn = urllib.parse.quote_plus('YOUR_EXECUTION_ROLE_ARN')
region_name = 'YOUR_REGION'

conn_string = '{0}://{1}:{2}@{3}:{4}?
role_arn={5}&region_name={6}'.format(conn_type, login, password, host, port,
 role_arn, region_name)
print(conn_string)
```

2. 用##替換佔位符。
3. 執行下列指令碼來產生連接字串。

```
python3 mwaa_connection.py
```

第四步：在 Secrets Manager 中添加變量

以下部分說明如何在秘密管理員中為變數建立密碼。

若要建立機密

1. 開啟 AWS Secrets Manager 主控台。
2. 選擇 Store a new secret (存放新機密)。
3. 選擇其他類型的密碼。
4. 在 [指定要儲存在此密碼] 窗格中的金鑰/值組上，選擇 [純文字]。
5. 以下列格式將變數值新增為純文字。

   ```
   "YOUR_VARIABLE_VALUE"
   ```

   例如，要指定一個整數：

   ```
   14
   ```

   例如，要指定一個字符串：

   ```
   "mystring"
   ```
6. 對於加密金鑰，請從下拉式清單中選擇AWS KMS金鑰選項。
7. 在密碼名稱的文字欄位中輸入下列格式的名稱。

   ```
   airflow/variables/YOUR_VARIABLE_NAME
   ```

   例如：

   ```
   airflow/variables/test-variable
   ```
8. 選擇 下一步。
9. 在 [設定密碼] 頁面上的 [密碼名稱和說明] 窗格上，執行下列動作。

   a. 對於秘密名稱，請提供密碼的名稱。

   b. (選用) 在 Description (說明) 中，提供機密的說明。

   選擇 下一步。
10. 在配置旋轉-可選保留默認選項，然後選擇下一步。
11. 針對您要新增的任何其他變數，重複 Secrets Manager 中的這些步驟。

12. 在「檢閱」頁面上，檢閱您的密碼，然後選擇「商店」。

第五步：在 Secrets Manager 中添加連接

下一節說明如何在秘密管理員中建立連接字串 URI 的密碼。

若要建立機密

1. 開啟 AWS Secrets Manager 主控台。
2. 選擇 Store a new secret (存放新機密)。
3. 選擇其他類型的密碼。
4. 在 [指定要儲存在此密碼] 窗格中的金鑰/值組上，選擇 [純文字]。
5. 以下列格式將連接 URI 字串新增為純文字。

```
YOUR_CONNECTION_URI_STRING
```

例如：

```
mysql://288888a0-50a0-888-9a88-1a111aaa0000.a1.us-east-1.airflow.amazonaws.com
%2Fhome?role_arn=arn%3Aaws%3Aiam%3A%3A001122332255%3Arole%2Fservice-role
%2FAmazonMWAA-MyAirflowEnvironment-iAaaaA&region_name=us-east-1
```

> **Warning**
>
> Apache 氣流解析連接字符串中的每個值。您不能使用單引號或雙引號，否則它會將連接解析為單個字符串。

6. 對於加密金鑰，請從下拉式清單中選擇AWS KMS金鑰選項。
7. 在密碼名稱的文字欄位中輸入下列格式的名稱。

```
airflow/connections/YOUR_CONNECTION_NAME
```

例如：

```
airflow/connections/myconn
```

8. 選擇 下一步。

9. 在 [設定密碼] 頁面上的 [密碼名稱和說明] 窗格上，執行下列動作。

   a. 對於秘密名稱，請提供密碼的名稱。

   b. (選用) 在 Description (說明) 中，提供機密的說明。

   選擇 下一步。

10. 在配置旋轉-可選保留默認選項，然後選擇下一步。
11. 針對您要新增的任何其他變數，重複 Secrets Manager 中的這些步驟。
12. 在「檢閱」頁面上，檢閱您的密碼，然後選擇「商店」。

範本程式碼

- 使用範例程式碼，了解如何使用此頁面上 Apache 氣流連線的密鑰 (myconn)使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接。
- 使用範例程式碼，瞭解如何使用此頁面上 Apache 氣流變數 (test-variable) 的密碼金鑰使用密鑰 AWS Secrets Manager對於一個阿帕奇氣流變量。

資源

- 如需有關使用主控台設定 Secrets Manager 密碼的詳細資訊AWS CLI，請參閱使AWS Secrets Manager用指南中的建立密碼。
- 在將 Apache 氣流連線和變數移至中，使用 Python 指令碼將大量 Apache 氣流變數和連線移轉至機 Secrets ManagerAWS Secrets Manager。

後續步驟？

- 瞭解如何在中產生 Apache Airflow 的權杖訪問阿帕奇氣流用戶界面。

管理亞馬遜 MWAA 環境

適用於 Apache 氣流主控台的 Amazon 受管工作流程包含內建選項，可設定 Apache 氣流使用者介面的私人或公開存取權。它也包含內建選項，可用來設定環境大小、擴展工作者的時機，以及 Apache Airflow 組態選項，可讓您覆寫一般只能在中存取的 Apache Airflow 組態`airflow.cfg`。本指南說明如何在 Amazon MWAA 主控台上使用這些組態。

主題

設定 Amazon MWAA 環境類別

您為 Amazon MWAA 環境選擇的環境類別會決定 C elle 執行程式執行程式執行所在的 AWS受管 AWS Fargate 容器的大小，以及由 Apache AWS氣流排程器建立任務執行個體的受管 Amazon Aurora PostgreSQL 中繼資料資料庫。本頁說明每個 Amazon MWAA 環境類別，以及在 Amazon MWAA 主控台上更新環境類別的步驟。

章節

環境能力

下節包含每個環境類別的預設並行 Apache Airflow 工作、隨機存取記憶體 (RAM) 以及虛擬集中處理單元 (vCPUs)。列出的並行工作假設工作並行性未超過環境中的 Apache Airflow Worker 容量。

在下表中，DAG 容量是指 DAG 定義，而不是執行項目，並假設您的 DAG 在單一 Python 檔案中是動態的，並以 Apache 氣流最佳作法撰寫。

工作執行取決於同時排定的數目，並假設設定為同時啟動的 DAG 執行數目不超過預設值 `max_dagruns_per_loop_to_schedule`，以及本主題所詳述的 Worker 大小和數目。

mw1.small

- 最多 50 個天的容量
- 5 個並發任務（默認情況下）
- 1 個 vCPUs
- 2 GB 內存

mw1.medium

- 最多 20 個 DAG 容量
- 10 個並發任務（默認情況下）
- 2 個 vCPUs
- 4 GB 公羊

mw1.large

- 最高可達 1000 天儲存容量
- 20 個並發任務（默認情況下）
- 4 個 vCPUs
- 8 GB RAM

mw1.xlarge

- 高達 2000 個每日儲存容量
- 40 個並發任務（默認情況下）
- 8 個 vCPUs
- 24 GB 記憶體

mw1.2xlarge

- 最高可達 4000 天的儲存容量
- 80 個並發任務（默認情況下）
- 16 個 vCPU
- 48 GB 公斤記憶體

您可以使用`celery.worker_autoscale`來增加每個工作者的工作。如需更多資訊，請參閱the section called “範例高效能使用案例”。

阿帕奇氣流調度器

下節包含 Amazon MWAA 上可用的 Apache 氣流排程器選項，以及排程器數量如何影響觸發器的數量。

在 Apache Airflow 中，觸發器會管理工作，直到符合使用觸發器指定的特定條件為止。在 Amazon MWAA 中，觸發器會在同一個 Fargate 任務上與排程器一起執行。相應地增加排程器計數會增加可用的觸發器數目，進而最佳化環境管理延遲工作的方式。這樣可以確保任務的有效處理，並立即安排它們在滿足條件時運行。

Apache Airflow v2

- v2-接受2到之間5。預設為 2。

設定亞馬遜 MWAA 自動擴展

自動調度資源機制會自動增加 Apache Airflow 工作者的數量，以回應適用於 Apache Airflow 環境的 Amazon 受管工作流程中和排入佇列的任務，並在沒有其他任務排入佇列或執行中時處理額外的工作者。本頁說明如何透過使用 Amazon MWAA 主控台指定在環境中執行的 Apache Airflow 工作者數目上限，以設定自動調度資源。

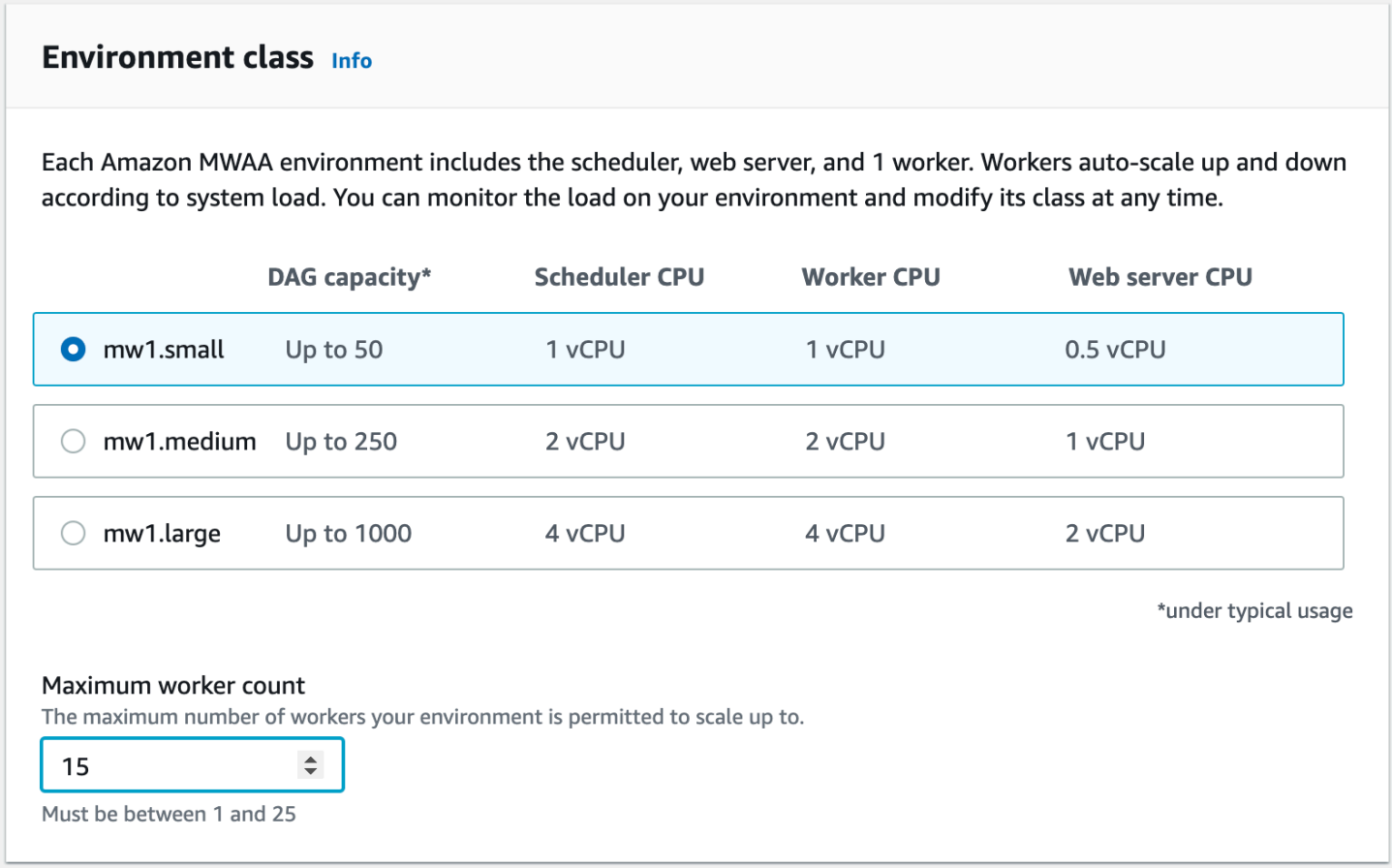
> **Note**
>
> Amazon MWAA 使用 Apache 氣流指標來判斷何時需要其他 C elery 執行程式工作者，並視需要將 Fargate 工作者的數目增加到指定的值。`max-workers`當該數字為零時，Amazon MWAA 會移除其他工作者，並將其縮減回值。`min-workers`如需詳細資訊，請參閱下一the section called “運作方式”節。
>
> 發生縮小規模時，可以排程新工作。此外，設定要刪除的 Worker 也可以在移除 Worker 容器之前挑選這些工作。由於多種因素的組合，此期間可能會持續 2 到 5 分鐘：傳送 Apache Airflow 指標所需的時間、偵測零任務穩定狀態的時間，以及移除 Fargate 工作者所需的時間。如果您在持續工作負載的情況下使用 Amazon MWAA，後面接著沒有工作負載的期間，則不會受到此限制的影響。不過，如果您有重複高使用率的間歇性工作負載，接著大約五分鐘沒有工作，則當在縮減規模 Worker 上執行的工作被刪除並標示為失敗時，您可能會受到此問題的影響。如果您受到此限制的影響，建議您執行下列其中一項操作：

- 設定為`min-workers`等於足夠容量以滿足您的平均工作負載，如果此模式在大部分 24 小時期間都持續存在，則最好是因為在這種情況下，自動調度資源的價值會有限。`max-workers`
- 請確定在一個 DAG 中至少有一項工作，例如，在這段間歇性活動期間執行 DateTimeSensor，以防止不必要的縮減規模。

章節

最大工作人數

下圖顯示您可以在哪裡自訂最大工作者計數，以便在 Amazon MWAA 主控台上設定自動調度資源。



運作方式

Amazon MWAA 使用RunningTasks和QueuedTasks指標，其中 (執行中的任務 + 排入佇列的任務)/(每個工作者的任務) = (必要的工作者)。如果所需的工作程式數目大於目前的工作者數目，Amazon MWAA 會將 Fargate 工作者容器新增至該值，直到指定的最大值為止。max-workers

當RunningTasks和QueuedTasks指標總和為零時間為兩分鐘時，Amazon MWAA 會要求 Fargate 將工作者數量設定為環境的值。min-workersAmazon MWAA 為 Fargate 提供 120 秒 (目前最長可用時間) 的stopTimeout值，以允許工作人員完成任何工作，之後移除容器並刪除任何剩餘進行中的工作。在大多數情況下，當佇列中沒有任何工作時，就會發生這種情況，不過，在此頁面前一節所述的特定情況下，正在進行縮減規模時，工作可能會排入佇列。

當您建立環境時，Amazon MVAA 會在不同可用區域中的兩個私有子網路中，建立一個 AWS 管理的 Amazon Aurora PostgreSQL 中繼資料資料庫和一個 Fargate 容器。例如，中繼資料資料庫和容器，以us-east-1a及區域us-east-1b可用區域中的中繼資料資料庫和容器。us-east-1

- 亞馬遜 MWAA 環境上的 Apache 氣流工作者使用芹菜執行程序排隊和分發任務從 Apache 氣流平台多個芹菜工作人員。芹菜執行人在一個AWS Fargate容器中運行。如果一個可用區域中的 Fargate 容

器發生故障，Amazon MWAA 會切換到不同可用區域中的另一個容器以執行 Celery 執行程式，而 Apache 氣流排程器則會在 Amazon Aurora PostgreSQL 中繼資料資料庫中建立新的任務執行個體。

- 根據預設，Amazon MWAA 會設定一個環境，以 parallel (在中) 執行數百個任務，並同時執行工作者 (在中`core.parallelism`)。`core.dag_concurrency`任務排入佇列時，Amazon MWAA 會新增工作者以滿足需求，直到達到您在最大工作者計數中定義的數量為止。
- 例如，如果您指定的值為`10`，Amazon MWAA 會增加最多 9 個額外的工作程式來滿足需求。此自動調度資源機制會繼續執行其他 Worker，直到沒有其他工作可執行為止。當沒有其他任務執行中或佇列中的任務時，Amazon MWAA 會處置工作者並縮減為單一工作者。

使用亞馬遜 MWAA 主控台

您可以選擇在 Amazon MWAA 主控台上同時在環境上執行的最大工作者數量。依預設，您可以指定最大值 25。

若要設定 Worker 數目

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇 編輯 。
4. 選擇下一步。
5. 在 [環境類別] 窗格中，輸入 [Worker 計數上限] 中的值。
6. 選擇 儲存 。

> **Note**
>
> 變更可能需要幾分鐘才會對您的環境生效。

範例高效能使用案例

下節說明可用來在環境中啟用高效能和平行處理原則的組態類型。

本地阿帕奇氣流

一般而言，在內部部署 Apache Airflow 平台中，您可以在檔案中設定工作平行處理、自動調度資源和並行設定：`airflow.cfg`

- `core.parallelism`— 每個排程器可同時執行的工作執行個體數目上限。
- `core.dag_concurrency`— DAG 的最大並行性 (不是工作者)。
- `celery.worker_autoscale`— 可在任何 Worker 上同時執行的工作數目上限與最小數目。

例如，如果設定`core.parallelism`為`100`且設定`core.dag_concurrency`為`7`，您仍然只能在有 2 個 DAG 的情況下同時執行總`14`工作。指定的是，即使整體平行處理原則設定為 (in`core.dag_concurrency`)，每個 DAG 都只能同時執行七項工作。`100` `core.parallelism`

在亞馬遜 MWAA 環境上

在 Amazon MWAA 環境中，您可以使用、和最大工作者計數自動調度機制在 Amazon MWAA 上使用阿帕奇氣流配置選項，設定 Amazon MWAA 環境類別直接在 Amazon MWAA 主控台上設定這些設定。雖然 `core.dag_concurrency` Amazon MWAA 主控台上的 Apache 氣流組態選項無法在下拉式清單中使用，但您可以將其新增為自訂 Apache 氣流組態選項。

比方說，當您建立環境時，您選擇了下列設定：

1. mw1.small 環境類別，可控制每個背景工作預設可執行的最大並行工作數目，以及容器的 vCPU。
2. [上限] Worker 計數中 [Worker] 的`10`預設設定。
3. 適用於每個工作者`5`,`5`任務`celery.worker_autoscale`的 Apache 氣流組態選項。

這表示您可以在環境中執行 50 個並行工作。任何超過 50 的任務都將被排入佇列，並等待正在運行的任務完成。

執行更多並行工作。您可以使用下列組態修改環境以同時執行更多工作：

1. 透過選擇 `mw1.medium` (預設為 10 個並行工作) 環境類別，增加每個 Worker 可執行的並行工作數目上限，以及容器的 vCPU。
2. 新增`celery.worker_autoscale`為 Ap ache 氣流組態選項。
3. 增加最大工作者計數。在此範例中，將最大 Worker 數從增加`10`到`20`會使環境可執行的並行作業數目增加一倍。

指定最小工作者。您也可以使用 AWS Command Line Interface (AWS CLI) 指定在您環境中執行的 Apache 氣流工作程式的最小和最大數目。例如：

```
aws mwaa update-environment --max-workers 10 --min-workers 10 --
name YOUR_ENVIRONMENT_NAME
```

若要深入了解，請參閱中的更新環境命令。AWS CLI

疑難排解卡在執行中狀態的工作

在極少數情況下，Apache 氣流可能會認為有工作仍在執行中。若要解決此問題，您需要在 Apache 氣流使用者介面中清除擱淺的工作。如需詳細資訊，請參閱我看到我的任務卡住或未完成疑難排解主題。

後續步驟？

- 進一步了解我們建議您調整環境效能的最佳實務Amazon MWAA 上阿帕奇氣流的性能調整。

在 Amazon MWAA 上使用阿帕奇氣流配置選項

Apache 氣流組態選項可以連接至您的 Amazon 管理工作流程，以作為環境變數的 Apache 氣流環境。您可以從建議的下拉式清單中選擇，或在 Amazon MWAA 主控台為 Apache 氣流版本指定自訂組態選項。本頁說明可用的 Apache 氣流組態選項，以及如何使用這些選項覆寫環境中的 Apache 氣流組態設定。

內容

- 範例 DAG
- 電子郵件通知設定範
- 後續步驟？

必要條件

您需要下列項目，才能完成此頁面上的步驟。

- 權限 — 您的 AWS 帳戶必須已被管理員授與您環境的 AmazonmWAA FullConsoleAccess 存取控制原則的存取權限。此外，您的執行角色必須允許 Amazon MWAA 環境，才能存取環境使用的 AWS 資源。
- 存取 — 如果您需要存取公用儲存庫，才能直接在 Web 伺服器上安裝相依性，則您的環境必須設定公用網路 Web 伺服器存取權。如需詳細資訊，請參閱 the section called “阿帕奇氣流存取模式”。
- Amazon S3 組態 — 用於存放 DAG、自訂外掛程式和 Python 相依性的 Amazon S3 儲存貯體`requirements.txt`必須設定為封鎖公用存取並啟用版本控制。`plugins.zip`

運作方式

當您建立環境時，Amazon MWAA 會將您在氣流組態選項中 Amazon MWAA 主控台上指定的組態設定做為環境變數附加到您環境的 AWS Fargate 容器。如果您在中使用相同名稱的設定`airflow.cfg`，您在 Amazon MWAA 主控台上指定的選項會覆寫中的值。`airflow.cfg`

雖然我們不會`airflow.cfg`在 Amazon MWAA 環境的 Apache 氣流使用者介面中公開，但您可以直接在 Amazon MWAA 主控台上變更 Apache 氣流組態選項，然後繼續使用中的所有其他設定。`airflow.cfg`

使用配置選項在 Apache 氣流 V2 中加載插件

默認情況下，在 Apache 氣流 v2 中，插件被配置為使用該`core.lazy_load_plugins : True`設置「懶惰」加載。如果您在 Apache Airflow v2 中使用自訂外掛程式，則必須新增`core.lazy_load_plugins : False`為 Apache 氣流組態選項，才能在每個氣流程開始時載入外掛程式，以覆寫預設設定。

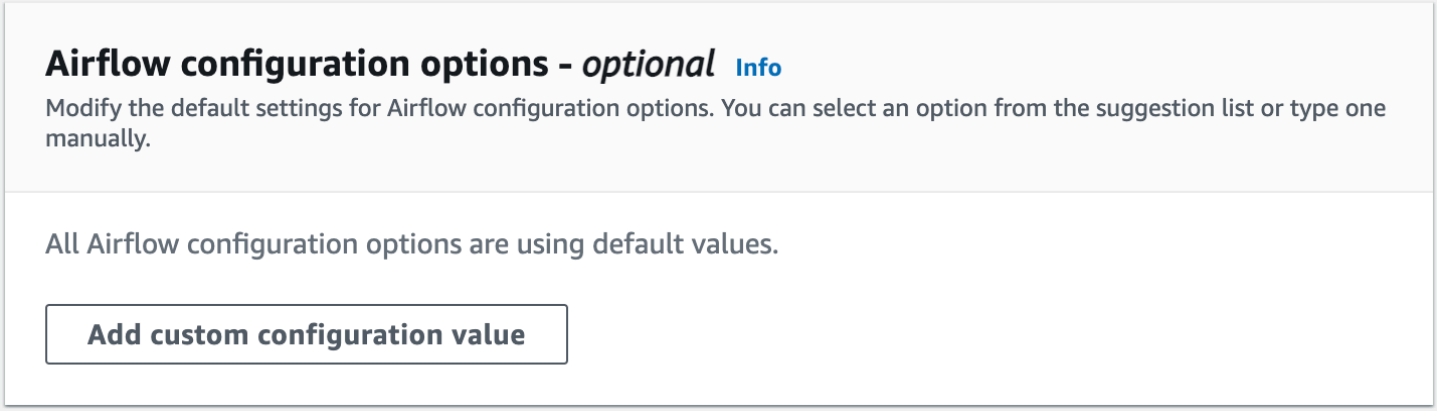
組態選項概觀

在 Amazon MWAA 主控台上新增組態時，Amazon MWAA 會將組態寫入為環境變數。

- 列出的選項。您可以從下拉式清單中選擇適用於 Apache Airflow 版本的其中一個組態設定。例如`dag_concurrency`：16。配置設置轉換為您環境的 Fargate 容器 `AIRFLOW__CORE__DAG_CONCURRENCY : 16`
- 自定義選項。您也可以在下拉式清單中指定未列出 Apache 氣流版本的氣流組態選項。例如`foo.user`：YOUR_USER_NAME。配置設置轉換為您環境的 Fargate 容器 `AIRFLOW__FOO__USER : YOUR_USER_NAME`

阿帕奇氣流配置選項

下圖顯示您可以在 Amazon MWAA 主控台上自訂 Apache 氣流組態選項的位置。



阿帕奇氣流參考

如需 Apache 氣流支援的組態選項清單，請參閱 Apache 氣流參考指南中的組態參考資料。若要檢視您在 Amazon MWAA 上執行的 Apache 氣流版本的選項，請從下拉式清單中選取版本。

使用 Amazon MWAA 主控台

下列程序會逐步引導您將 Airflow 組態選項新增至您的環境。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯。
4. 選擇下一步。
5. 在氣流組態選項窗格中選擇新增自訂組態。
6. 從下拉式清單中選擇組態並輸入值，或輸入自訂組態並輸入值。
7. 為您要新增的每個組態選擇新增自訂組態。

8. 選擇儲存。

組態參考

下一節包含 Amazon MWAA 主控台上下拉式清單中可用的 Apache 氣流組態清單。

電郵配置

下列清單顯示 Amazon MWAA 上可用的氣流電子郵件通知組態選項。

建議您針對 SMTP 流量使用連接埠 587。依預設，會在所有 Amazon EC2 執行個體的連接埠 25 上 AWS 封鎖輸出 SMTP 流量。如果您想要在通訊埠 25 上傳送輸出流量，您可以要求移除此限制。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
v2	電子郵件後端	Apache 氣流公用程式用於電子郵件後端的電子郵件通知。	氣流. 實用程序. 電子郵件發送電子郵件
v2	SMT.SMT 主機	smt p_host 中用於電子郵件地址的輸出伺服器名稱。	localhost
v2	SMT.SMTP-明星網	傳輸層安全性 (TLS) 是用來在 smt p_starttls 中透過網際網路加密電子郵件。	False
v2	短信網絡安全網絡	安全套接字層（SSL）用於在 smt p_ssl 中連接服務器和電子郵件客戶端。	True
v2	SMT.SMTP 端口	在 smt p_port 中指定給伺服器的傳輸控制通訊協定 (TCP) 連接埠。	587

氣流版本	氣流組態選項	描述	範例值
v2	短信電子郵件來自	寄件者中的輸出電子郵件地址。	myemail@domain.com

任務配置

下列清單顯示 Amazon MWAA 氣流任務下拉式清單中可用的組態。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
v2	核心. 默認任務重試	在預設值中重試阿帕奇氣流工作的次數。	3
v2	核心. 平行	可在整個環境中 parallel (平行處理原則) 同時執行的工作執行個體數目上限。	40

排程器組態

下列清單顯示 Amazon MWAA 下拉式清單中可用的 Apache 氣流排程器組態。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
v2	排程程式. 依預設值	告訴排程器建立 DAG 執行，以便「catch」在 catch up_by_default 中的特定時間間隔。	False
v2	排程器. 排程器 _ 殭屍任務閾值	告訴排程器是否要將作業執行個體標示為	300

氣流版本	氣流組態選項	描述	範例值
		失敗，並在排程器_zombie_task_threshold 中重新排程工作。	

工作者組態

下列清單顯示 Amazon MWAA 下拉式清單中可用的氣流工作者組態。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
v2	天然. 工人 _ 自動縮放	可以在 Worker_autoscale 中使用 Celly 執行程序的任何工作站上同時運行的任務的最大和最小數量。值必須以逗號分隔，順序如下：`max_concurrency,min_concurrency`	16,12

網頁伺服器組態

下列清單顯示 Amazon MWAA 下拉式清單中可用的氣流網路伺服器組態。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
v2	網絡服務器. 默認值	默認的阿帕奇氣流 UI 日期時間設置默認的 UI_時區/時區。	America/New_York

氣流版本	氣流組態選項	描述	範例值
		Note 設定此`default_ui_timezone`選項並不會變更排程執行 DAG 的時區。要更改 DAG 的時區，您可以使用自定義插件。如需詳細資訊，請參閱 the section called “變更 DAG 的時區”。	

耐用配置

下列清單顯示 Amazon MWAA 上可用的 Apache 氣流觸發器組態。

Apache Airflow v2

氣流版本	氣流組態選項	描述	範例值
第 2.7 版	已啟用多個特殊功能	用於在 Amazon MWAA 上激活和停用觸發器。依預設，此值是設為`True`。如果設定為`False`，Amazon MWAA 將不會在排程	True

氣流版本	氣流組態選項	描述	範例值
		器上啟動任何觸發程式程序。	
第 2.7 版	特雷格爾。默認容量	定義每個觸發器可以並行運 parallel 的觸發器的數量。在 Amazon MWAA 上，由於兩個元件彼此並排執行，因此每個觸發器和每個排程器都會設定此容量。每個排程器的預設值分別設為60125、250、、500以及1000針對小型、中型和大型、超大型和 2xlarge 執行個體。	125

範例和範例程式碼

範例 DAG

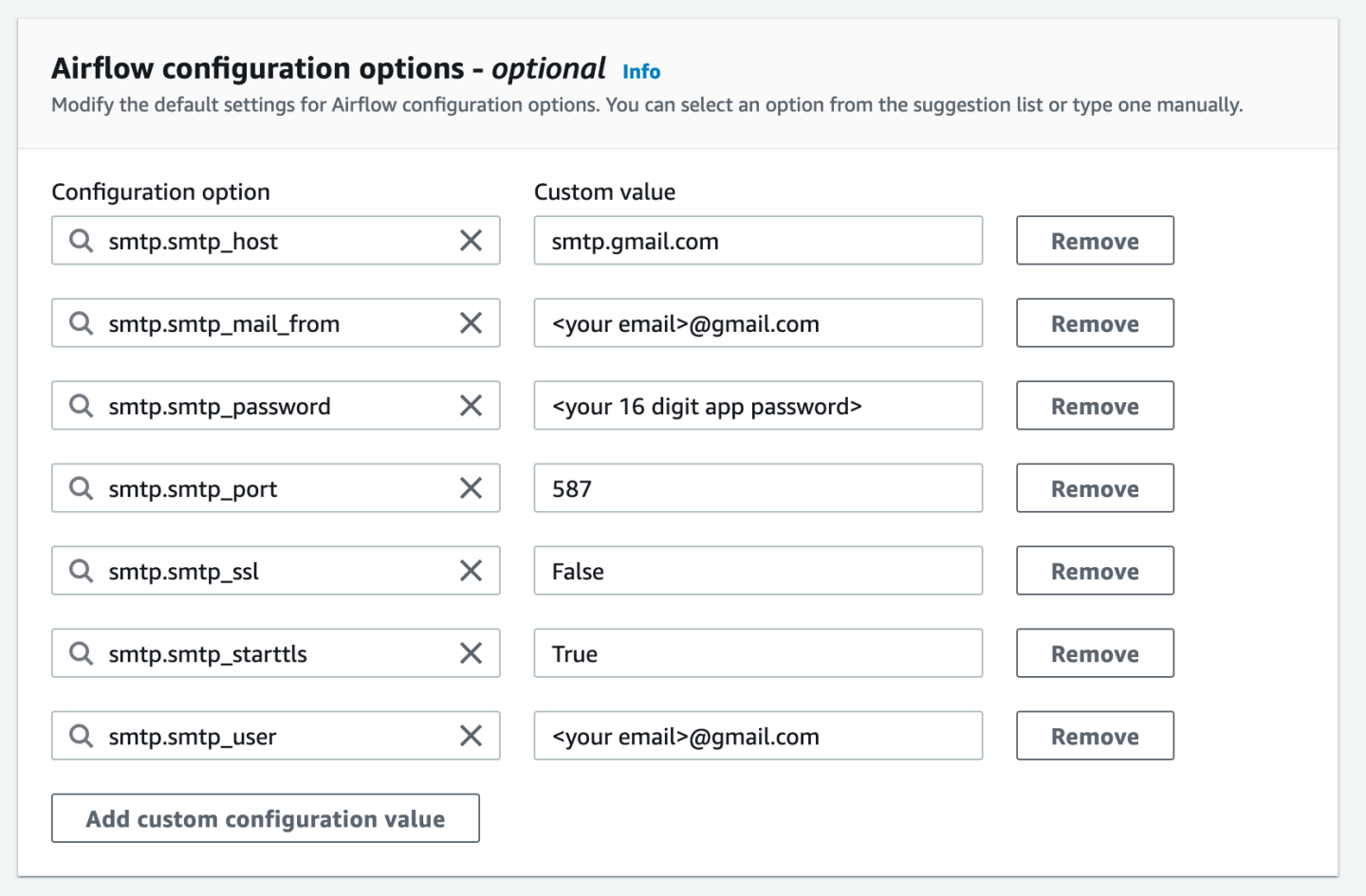
您可以使用下面的 DAG 打印您的 email_backend Apache 氣流配置選項。若要執行以回應 Amazon MWAA 事件，請將程式碼複製到 Amazon S3 儲存貯體上環境的 DAG 資料夾。

```
from airflow.decorators import dag
from datetime import datetime

def print_var(**kargs):
  email_backend = kwargs['conf'].get(section='email', key='email_backend')
  print("email_backend")
  return email_backend

@dag(
  dag_id="print_env_variable_example",
  schedule_interval=None,
```

```
  start_date=datetime(yyyy, m, d),
  catchup=False,
)
def print_variable_dag():
  email_backend_test = PythonOperator(
  task_id="email_backend_test",
  python_callable=print_var,
  provide_context=True

print_variable_test = print_variable_dag()
```

電子郵件通知設定範

下列 Apache 氣流設定選項可用於使用應用程式密碼的 Gmail.com 電子郵件帳戶。如需詳細資訊，請參閱 Gmail 說明參考指南中的使用應用程式密碼登入。



後續步驟？

- 了解如何將您的 DAG 資料夾上傳到您的 Amazon S3 儲存貯體新增或更新 DAG。

升級阿帕奇氣流版本

亞馬遜 MWAA 支援次要版本升級。這表示您可以將環境從版本升級`x.4.z`到`x.5.z`。若要執行主要版本升級 (例如從版本`1.y.z`到版本)`2.y.z`，您必須建立新環境並移轉資源。如需升級至新主要版本 Apache 氣流的詳細資訊，請參閱亞馬遜 MWAA 移轉指南中的遷移至新的亞馬遜 MWAA 環境。

在升級程序期間，Amazon MWAA 會擷取環境中繼資料的快照，將工作者、排程器、Web 伺服器升級至新的 Apache Airflow 版本，最後使用快照還原中繼資料資料庫。

> **Note**
>
> 您無法降級您環境的 Apache 氣流版本。

在升級之前，請確定您的 DAG 和其他工作流程資源與您要升級的新 Apache Airflow 版本相容。如果您使`requirements.txt`用管理相依性，也必須確定您在需求中指定的相依性與新版本相容。

主題

- 升級工作流程資源
- 指定新版本

升級工作流程資源

每當您要變更 Apache 氣流版本時，請確定您在`requirements.txt`. `--constraint`

> **Warning**
>
> 在升級期間指定與目標 Apache Airflow 版本不相容的需求，可能會導致對舊版 Apache Airflow 進行冗長的復原程序，並符合先前需求版本。

若要移轉工作流程資源

1. 建立aws-mwaa-local-runner儲存庫的分支，然後複製 Amazon MWAA 本機執行器的副本。
2. 簽出到與您要升級到的版本相匹配的 aws-mwaa-local-runner 存儲庫分支。
3. 使用亞馬遜 MWAA 本地運行器 CLI 工具來構建碼頭映像並在本地運行 Apache 氣流。有關更多信息，請參閱 GitHub 存儲庫中的本地運行器 README。

4. 若要更新您的資訊`requirements.txt`，請遵循 Amazon MWAA 使用者指南中的管理 Python 相依性中建議的最佳實務。
5. (選擇性) 若要加速升級程序，請清除環境的中繼資料資料庫。具有大量中繼資料的環境可能需要更長的時間來升級。
6. 成功測試工作流程資源後，請將 DAG 和外掛程式複製到環境的 Amazon S3 儲存貯體。`requirements.txt`

您現在可以編輯環境、指定新的 Apache Airflow 版本，然後開始更新程序。

指定新版本

完成更新工作流程資源以確保與新 Apache Airflow 版本的相容性之後，請執行下列動作來編輯環境詳細資料，並指定您要升級至的 Apache Airflow 版本。

> **Note**
>
> 當您執行升級時，程序期間會終止目前在環境上執行的所有工作。更新程序最多可能需要兩個小時，在此期間，您的環境將無法使用。

使用控制台指定新版本

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 從「環境」清單中，選擇您要升級的環境。
3. 在「環境」頁面上，選擇 「編輯」以編輯環境。
4. 在「環境詳細資料」區段中，對於 Airflow 版本，請從下拉式清單中選擇要將環境升級至的新 Apache Airflow 版本號碼。
5. 選擇 [下一步]，直到您進入 [檢閱並儲存] 頁面。
6. 在「檢閱並儲存」頁面上，檢閱您的變更，然後選擇「儲存」。

當您套用變更時，您的環境會開始升級程序。在此期間，您的環境狀態會指出 Amazon MWAA 正在採取的動作，以及程序是否成功。

在成功升級案例中，狀態會顯示`UPDATING`，然後`CREATING_SNAPSHOT`當 Amazon MWAA 擷取中繼資料的備份時。最後，狀態將首先返回`UPDATING`，然後返回到該過程完成`AVAILABLE`時。

如果環境升級失敗，則會顯示您的環境狀態ROLLING_BACK。如果復原成功，則會先顯示狀態UPDATE_FAILED，指出更新失敗，但環境可用。如果復原失敗，則會顯示狀態UNAVAILABLE，表示您無法存取環境。

搭配 Amazon MWAA 使用啟動指令碼

啟動指令碼是您在環境的 Amazon S3 儲存貯體中託管的 shell (.sh) 指令碼，類似於您的 DAG、需求和外掛程式。Amazon MWAA 會在啟動期間在每個個別的 Apache Airflow 元件 (工作者、排程器和網頁伺服器) 上執行此指令碼，然後再安裝需求並初始化 Apache 氣流程。使用啟動指令碼執行下列作業：

- 安裝執行階段 — 安裝工作流程和連線所需的 Linux 執行階段。
- 設定環境變數 — 為每個 Apache 氣流元件設定環境變數。覆寫常見變數PATH，例如PYTHONPATH、和LD_LIBRARY_PATH。
- 管理金鑰和權杖 — 將自訂儲存庫的存取權杖傳遞給requirements.txt並設定安全金鑰。

下列主題說明如何使用 CloudWatch 記錄設定啟動指令碼來安裝 Linux 執行階段、設定環境變數，以及疑難排解相關問題。

主題

- 設定啟動指令碼
- 使用啟動指令碼安裝 Linux 執行階段
- 使用啟動指令碼設定環境變數

設定啟動指令碼

若要將啟動指令碼與現有的 Amazon MWAA 環境搭配使用，請將.sh檔案上傳到環境的 Amazon S3 儲存貯體。然後，若要將指令碼與環境產生關聯，請在您的環境詳細資料中指定下列項目：

- 指令碼的 Amazon S3 URL 路徑 — 儲存貯體中託管之指令碼的相對路徑，例如，s3://mwaa-environment/*startup.sh*
- 指令碼的 Amazon S3 版本識別碼 — Amazon S3 儲存貯體中的啟動殼層指令碼版本。每次更新指令碼時，您都必須指定 Amazon S3 指派給檔案的版本識別碼。版本識別碼為 Unicode、UTF-8 編碼、網址就緒、不透明的字串，長度不超過 1,024 個位元組，例如。3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo

若要完成本節中的步驟，請使用下列範例指令碼。該腳本輸出分配給的值`MWAA_AIRFLOW_COMPONENT`。此環境變數可識別執行指令碼的每個 Apache 氣流元件。

複製代碼並將其保存在本地`startup.sh`。

```
#!/bin/sh

echo "Printing Apache Airflow component"
echo $MWAA_AIRFLOW_COMPONENT
```

接下來，將指令碼上傳到您的 Amazon S3 儲存貯體。

AWS Management Console

若要上傳殼層指令碼 (主控台)

1. 登入 AWS Management Console，並開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 從「值區」清單中，選擇與您環境相關聯的值區名稱。
3. 在 Objects (物件)標籤上，選擇 Upload (上傳)。
4. 在「上傳」頁面上，拖放您建立的 shell 指令碼。
5. 選擇上傳。

指令碼會出現在物件清單中。Amazon S3 為該文件創建一個新的版本 ID。如果您更新指令碼並使用相同的檔案名稱再次上傳，則會為檔案指定新的版本 ID。

AWS CLI

若要建立並上傳命令介面指令碼 (CLI)

1. 開啟新的命令提示字元，然後執行 Amazon S3 `ls` 命令以列出並識別與您的環境相關聯的儲存貯體。

   ```
   $ aws s3 ls
   ```

2. 導覽至您儲存殼層指令碼的資料夾。`cp`在新的提示視窗中使用，將指令碼上傳至您的值區。用*`######## S3 #`*。

   ```
   $ aws s3 cp startup.sh s3://your-s3-bucket/startup.sh
   ```

如果成功，Amazon S3 會將 URL 路徑輸出至物件：

```
upload: ./startup.sh to s3://your-s3-bucket/startup.sh
```

3. 使用下列命令擷取指令碼的最新版本 ID。

```
$ aws s3api list-object-versions --bucket your-s3-bucket --prefix startup --
query 'Versions[?IsLatest].[VersionId]' --output text
```

```
BbdVMmBRjtestta1EsVnbybZp1Wqh1J4
```

當您將指令碼與環境產生關聯時，您可以指定此版本識別碼。

現在，將指令碼與您的環境相關聯。

AWS Management Console

建立指令碼與環境 (主控台) 的關聯

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選取您要更新之環境的列，然後選擇「編輯」。
3. 在 [指定詳細資料] 頁面上，對於啟動指令碼檔案-選用，輸入指令碼的 Amazon S3 URL，例如：`s3://your-mwaa-bucket/startup-sh.`。
4. 從下拉式清單中選擇最新版本，或瀏覽S3以尋找指令碼。
5. 選擇「下一步」，然後前往「檢閱並儲存」頁面。
6. 檢閱變更，然後選擇儲存。

環境更新可能需要 10 到 30 分鐘。Amazon MWAA 會在環境中的每個元件重新啟動時執行啟動指令碼。

AWS CLI

將指令碼與環境 (CLI) 產生關聯的步驟

- 開啟命令提示字元，並`update-environment`使用指定指令碼的 Amazon S3 URL 和版本識別碼。

```
$ aws mwaa update-environment \
    --name your-mwaa-environment \
    --startup-script-s3-path startup.sh \
    --startup-script-s3-object-version BbdVMmBRjtestta1EsVnbybZp1Wqh1J4
```

如果成功，Amazon MWAA 返回環境的 Amazon 資源名稱（ARN）：

```
arn:aws::airflow:us-west-2:123456789012:environment/your-mwaa-environment
```

環境更新可能需要 10 到 30 分鐘。Amazon MWAA 會在環境中的每個元件重新啟動時執行啟動指令碼。

最後，擷取記錄事件以驗證指令碼是否如預期般運作。當您為每個 Apache 氣流元件啟動記錄時，Amazon MWAA 會建立新的日誌群組和日誌串流。如需詳細資訊，請參閱 Apache 氣流記錄檔類型。

AWS Management Console

若要檢查 Apache 氣流記錄資料流 (主控台)

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇您的環境。
3. 在 [監視] 窗格中，選擇要檢視其記錄檔的記錄群組，例如 Airflow 排程器記錄群組。
4. 在 CloudWatch 主控台的「記錄串流」清單中，選擇具有下列前置詞的串流：`startup_script_exection_ip`
5. 在 [記錄事件] 窗格中，您將看到列印值的命令輸出`MWAA_AIRFLOW_COMPONENT`。例如，對於排程器記錄，您將會執行下列動作：

```
Printing Apache Airflow component
scheduler
Finished running startup script. Execution time: 0.004s.
Running verification
Verification completed
```

您可以重複上述步驟來檢視 Worker 和 Web 伺服器記錄。

使用啟動指令碼安裝 Linux 執行階段

使用啟動指令碼更新 Apache Airflow 元件的作業系統，並安裝其他執行階段程式庫以搭配您的工作流程使用。例如，執行下列指令碼`yum update`來更新作業系統。

在啟動指令碼`yum update`中執行時，您必須使用`--exclude=python*`範例中所示排除 Python。為了讓您的環境能夠執行，Amazon MWAA 會安裝與您的環境相容的特定 Python 版本。因此，您無法使用啟動指令碼更新環境的 Python 版本。

```
#!/bin/sh

echo "Updating operating system"
sudo yum update -y --exclude=python*
```

若要在特定的 Apache Airflow 元件上安裝執行階段，請使用`MWAA_AIRFLOW_COMPONENT`和`if`條件陳述式。`fi`此範例會執行單一命令，在排程器和 Worker 上安裝程式`libaio`庫，但不會在 Web 伺服器上安裝程式庫。

> **Important**
>
> - 如果您已設定私人 Web 伺服器，則必須使用下列條件，或在本機提供所有安裝檔案，以避免安裝逾時。
> - 用`sudo`於執行需要管理權限的作業。

```
#!/bin/sh

if [[ "${MWAA_AIRFLOW_COMPONENT}" != "webserver" ]]
then
     sudo yum -y install libaio
fi
```

您可以使用啟動腳本來檢查 Python 版本。

```
#!/bin/sh

export PYTHON_VERSION_CHECK=`python -c 'import sys; version=sys.version_info[:3];
 print("{0}.{1}.{2}".format(*version))'`
```

```
echo "Python version is $PYTHON_VERSION_CHECK"
```

Amazon MWAA 不支援覆寫預設 Python 版本，因為這可能會導致與已安裝的 Apache 氣流程式庫不相容。

使用啟動指令碼設定環境變數

使用啟動指令碼來設定環境變數並修改 Apache 氣流組態。下面定義了一個新的變量，ENVIRONMENT_STAGE。您可以在 DAG 或自訂模組中參考此變數。

```
#!/bin/sh

export ENVIRONMENT_STAGE="development"
echo "$ENVIRONMENT_STAGE"
```

使用啟動指令碼覆寫常見的 Apache 氣流或系統變數。例如，您設定LD_LIBRARY_PATH為指示 Python 在您指定的路徑中尋找二進位檔案。這可讓您使用外掛程式為工作流程提供自訂二進位檔案：

```
#!/bin/sh

export LD_LIBRARY_PATH=/usr/local/airflow/plugins/your-custom-binary
```

預留環境變數

Amazon MWAA 保留一組關鍵環境變數。如果覆寫保留變數，Amazon MWAA 會將其還原為預設值。以下列出了保留變量：

- MWAA__AIRFLOW__COMPONENT— 用來識別具有下列其中一個值的 Apache 氣流元件：schedulerworker、或webserver。
- AIRFLOW__WEBSERVER__SECRET_KEY— 用於在 Apache 氣流網頁伺服器中安全簽署工作階段 Cookie 的密鑰。
- AIRFLOW__CORE__FERNET_KEY— 用於加密和解密中繼資料資料庫中儲存之敏感資料的金鑰，例如連線密碼。
- AIRFLOW_HOME— Apache 氣流主目錄的路徑，其中組態檔和 DAG 檔案儲存在本機。
- AIRFLOW__CELERY__BROKER_URL— 用於 Apache 氣流排程器和 Celly 工作者節點之間通訊的訊息代理程式 URL。
- AIRFLOW__CELERY__RESULT_BACKEND— 用來儲存 Celery 工作結果的資料庫 URL。

- AIRFLOW__CORE__EXECUTOR-阿帕奇氣流應該使用的執行程序類。在 Amazon MWAA，這是一個 CeleryExecutor
- AIRFLOW__CORE__LOAD_EXAMPLES— 用於啟動或停用範例 DAG 的載入。
- AIRFLOW__METRICS__METRICS_BLOCK_LIST— 用於管理 Amazon MWAA 中發出和捕獲的 Apache 氣流指標。 CloudWatch
- SQL_ALCHEMY_CONN— RDS 適用於 PostgreSQL 資料庫的連接字串，用於將阿帕奇氣流中繼資料儲存在 Amazon MWAA 中。
- AIRFLOW__CORE__SQL_ALCHEMY_CONN— 用於與之相同的用途SQL_ALCHEMY_CONN，但遵循新的 Apache 氣流命名慣例。
- AIRFLOW__CELERY__DEFAULT_QUEUE— Apache 氣流中芹菜工作的預設佇列。
- AIRFLOW__OPERATORS__DEFAULT_QUEUE— 使用特定 Apache 氣流運算子之工作的預設佇列。
- AIRFLOW_VERSION— 在 Amazon MWAA 環境中安裝的 Apache 氣流版本。
- AIRFLOW_CONN_AWS_DEFAULT— 用來與中的其他AWS服務整合的預設認AWS證。
- AWS_DEFAULT_REGION— 設置與默認憑據一起使用的默認AWS區域以與其他AWS服務集成。
- AWS_REGION— 如果已定義，此環境變數會覆寫環境變數AWS_DEFAULT_REGION和設定檔設定區域中的值。
- PYTHONUNBUFFERED-用於發送stdout和stderr流式傳輸到容器日誌。
- AIRFLOW__METRICS__STATSD_ALLOW_LIST— 用於配置逗號分隔前綴的允許列表，以發送以列表元素開頭的指標。
- AIRFLOW__METRICS__STATSD_ON— 啟動傳送量度至StatsD。
- AIRFLOW__METRICS__STATSD_HOST-用於連接到StatSD守護進程。
- AIRFLOW__METRICS__STATSD_PORT-用於連接到StatSD守護進程。
- AIRFLOW__METRICS__STATSD_PREFIX-用於連接到StatSD守護進程。
- AIRFLOW__CELERY__WORKER_AUTOSCALE— 設定最大和最小並行。
- AIRFLOW__CORE__DAG_CONCURRENCY— 設定一個 DAG 中排程器可同時執行的工作執行個體數目。
- AIRFLOW__CORE__MAX_ACTIVE_TASKS_PER_DAG— 設定每個 DAG 的使用中工作數目上限。
- AIRFLOW__CORE__PARALLELISM— 定義可同時執行的工作實例數目上限。
- AIRFLOW__SCHEDULER__PARSING_PROCESSES— 設定排程器剖析以排程 DAG 的最大處理程序數目。
- AIRFLOW__CELERY_BROKER_TRANSPORT_OPTIONS__VISIBILITY_TIMEOUT— 定義 Worker 在將訊息重新傳遞給另一個 Worker 之前等待確認工作的秒數。

- AIRFLOW__CELERY_BROKER_TRANSPORT_OPTIONS__REGION— 設置AWS區域的基礎芹菜運輸.
- AIRFLOW__CELERY_BROKER_TRANSPORT_OPTIONS__PREDEFINED_QUEUES— 為基礎 Celery 傳輸設定佇列。
- AIRFLOW_SCHEDULER_ALLOWED_RUN_ID_PATTERN— 用於在觸發 DAG 時驗證run_id參數輸入的有效性。
- AIRFLOW__WEBSERVER__BASE_URL— 用於託管 Apache 氣流使用者介面的網頁伺服器 URL。

未預留的環境變數

您可以使用啟動指令碼覆寫未保留的環境變數。下面列出了一些這些常見的變量：

- PATH— 指定作業系統在其中搜尋可執行檔和程序檔的目錄清單。當指令在指令行中執行時，系統會檢查目錄以尋找並執行指令。PATH當您在 Apache Airflow 中建立自訂運算子或工作時，您可能需要仰賴外部指令碼或可執行檔。如果包含這些檔案的目錄不在PATH變數中指定的，則當系統找不到工作時，工作將無法執行。透過將適當的目錄新增至PATH，Apache Airflow 工作可以尋找並執行所需的可執行檔。
- PYTHONPATH— 由 Python 解譯器用於確定要搜索導入模塊和包的目錄。這是您可以新增至預設搜尋路徑的目錄清單。這使解釋器可以查找並加載未包含在標準庫中或安裝在系統目錄中的 Python 庫。使用此變數可新增您的模組和自訂 Python 套件，並將它們與 DAG 搭配使用。
- LD_LIBRARY_PATH— Linux 中動態鏈接器和加載器用於查找和加載共享庫的環境變量。它指定包含共享庫的目錄列表，這些目錄在默認系統庫目錄之前進行搜索。使用此變數可指定您的自訂二進位檔案。
- CLASSPATH— 由 Java 執行階段環境 (JRE) 和 Java 開發套件 (JDK) 用來在執行階段尋找和載入 Java 類別、程式庫和資源。它是包含編譯的 Java 代碼的目錄，JAR 文件和 ZIP 存檔的列表。

在亞馬遜 MWAA 與 DAG 工作

若要在適用於 Apache 氣流環境的 Amazon 受管工作流程上執行定向無環圖 (DAG)，請將檔案複製到連接到環境的 Amazon S3 儲存貯體，然後讓 Amazon MWAA 知道您的 DAG 和支援檔案位於 Amazon MWAA 主控台上的位置。Amazon MWAA 負責同步工作者、排程器和 Web 伺服器之間的 DAG。本指南說明如何在 Amazon MWAA 環境中新增或更新您的 DAG，以及如何安裝自訂外掛程式和 Python 相依性。

主題

Amazon S3 存儲桶概述

亞馬遜 MWAA 環境的 Amazon S3 儲存貯體必須封鎖公開存取。根據預設，所有 Amazon S3 資源 (儲存貯體、物件和相關子資源 (例如生命週期組態) 都是私有的。

- 只有資源擁有者 (建立值區的AWS帳號) 可以存取資源。資源擁有者 (例如，您的管理員) 可以透過撰寫存取控制原則，將存取權限授與其他人。
- 您設定的存取政策必須具有將 DAG、自訂外掛程式和 Python 相依性新增`requirements.txt`至 Amazon S3 儲存貯體的權限。`plugins.zip`如需包含所需權限的範例政策，請參閱 Amazon FullConsoleAccess mWAA。

亞馬遜 MWAA 環境的 Amazon S3 儲存貯體必須啟用版本控制。啟用 Amazon S3 儲存貯體版本控制後，只要建立新版本，就會建立新副本。

- 已為 a 中的自訂外掛程式啟用版本控制`plugins.zip`，並在 Amazon S3 儲存貯體`requirements.txt`上啟用 Python 相依性。
- 每次在 Amazon S3 儲存貯體`requirements.txt`上更新這些檔案時`plugins.zip`，您都必須在 Amazon MWAA 主控台上指定和版本。

新增或更新 DAG

有向無環圖 (DAG) 是在定義 DAG 結構為程式碼的 Python 檔案中定義的。您可以使用AWS CLI、或 Amazon S3 主控台將 DAG 上傳至您的環境。本頁說明如何使用 Amazon S3 儲存貯體中的dags資料夾，在亞馬遜管理的 Apache 氣流環境中新增或更新 Apache 氣流 DAG 的步驟。

章節

先決條件

您需要下列項目，才能完成此頁面上的步驟。

- 權限 — 您的AWS帳戶必須已被管理員授與您環境的 AmazonmWAAFullConsoleAccess 存取控制原則的存取權限。此外，您的執行角色必須允許 Amazon MWAA 環境，才能存取環境使用的AWS資源。
- 存取 — 如果您需要存取公用儲存庫，才能直接在 Web 伺服器上安裝相依性，則您的環境必須設定公用網路 Web 伺服器存取權。如需詳細資訊，請參閱the section called “阿帕奇氣流存取模式”。
- Amazon S3 組態 — 用於存放 DAG、自訂外掛程式和 Python 相依性的`plugins.zip` Amazon S3 儲存貯體`requirements.txt`必須設定為封鎖公用存取並啟用版本控制。

運作方式

有向非循環圖 (DAG) 是在定義 DAG 結構為程式碼的單一 Python 檔案中定義的。它由以下各項：

- 一個 DAG 定義。
- 說明如何執行 DAG 和要執行之工作的運算子。

- 描述工作執行順序的運算子關係。

若要在 Amazon MWAA 環境上執行 Apache 氣流平台，您需要將 DAG 定義複製到儲存貯體中的dags資料夾。例如，儲存值區中的 DAG 資料夾可能如下所示：

Example DAG 資料夾

```
dags/
# dag_def.py
```

Amazon MWAA 每 30 秒自動將新的和變更的物件從 Amazon S3 儲存貯體同步到 Amazon MWAA 排程器和工作者容器的/usr/local/airflow/dags資料夾，不論檔案類型為何，都能保留 Amazon S3 來源的檔案階層。新的 DAG 出現在 Apache 氣流使用者介面中的時間由控制scheduler.dag_dir_list_interval。對現有 DAG 的變更將會在下一個 D AG 處理迴圈中取得。

> **Note**
>
> 您不需要將airflow.cfg設定檔案包含在 DAG 資料夾中。您可以從亞馬遜 MWAA 主控台覆寫預設的 Apache 氣流組態。如需詳細資訊，請參閱在 Amazon MWAA 上使用阿帕奇氣流配置選項。

v2 中有什麼變化

- 新功能：運營商，鉤子和執行者。DAG 中的匯入陳述式，以及您在亞馬遜 MWAA 中指定的自訂外掛程式已plugins.zip在 Apache 氣流 v1 和 Apache 氣流 v2 之間變更。例如，`from airflow.contrib.hooks.aws_hook import AwsHook`在阿帕奇氣流 V1 已經改變為`from airflow.providers.amazon.aws.hooks.base_aws import AwsBaseHook`在阿帕奇氣流 V2。若要進一步了解，請參閱 Apache 氣流參考指南中的 Python API 參考。

使用亞馬遜 MWAA CLI 公用程式測試 DAG

- 命令列介面 (CLI) 公用程式會針對 Apache Airflow 環境複寫 Amazon 受管工作流程。
- CLI 會在本機建立類似於 Amazon MWAA 生產映像的碼頭容器映像。這可讓您執行本機 Apache 氣流環境來開發和測試 DAG、自訂外掛程式和相依性，然後再部署到 Amazon MWAA。
- 要運行 CLI，請參閱（詳見）GitHub。aws-mwaa-local-runner

將 DAG 程式碼上傳至 Amazon S3

您可以使用 Amazon S3 主控台或AWS Command Line Interface (AWS CLI) 將 DAG 程式碼上傳至 Amazon S3 儲存貯體。下列步驟假設您將程式碼 (.py) 上傳到 Amazon S3 儲存貯體dags中名稱的資料夾。

使用 AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面上的步驟，您需要以下項目：

- AWS CLI— 安裝第 2 版。
- AWS CLI— 快速配置`aws configure`.

若要使用AWS CLI

1. 請用下列命令列出所有的 Amazon S3 儲存貯體。

   ```
   aws s3 ls
   ```

2. 請用下列命令列出您環境的 Amazon S3 儲存貯體中的檔案和資料夾。

   ```
   aws s3 ls s3://YOUR_S3_BUCKET_NAME
   ```

3. 下列指令會將`dag_def.py`檔案上傳至資`dags`料夾。

   ```
   aws s3 cp dag_def.py s3://YOUR_S3_BUCKET_NAME/dags/
   ```

   如果 Amazon S3 儲存貯體中尚`dags`未存在名為的資料夾，此命令會建立`dags`資料夾並`dag_def.py`將名為的檔案上傳到新資料夾。

使用 Amazon S3 主控台

Amazon S3 主控台是可用來管理 Amazon S3 儲存貯體的 Web 型使用者介面。下列步驟假設您有名為的 DAGs 資料夾`dags`。

使用 Amazon S3 主控台上傳

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。

2. 選擇一個環境。
3. 在 S3 窗格的 DAG 程式碼中選取 S3 儲存貯體連結，以在 Amazon S3 主控台上開啟儲存貯體。
4. 選擇 dags 資料夾。
5. 選擇 Upload (上傳)。
6. 選擇 [新增檔案]。
7. 選擇您的本地副本dag_def.py，選擇上傳。

在亞馬遜 MWAA 主控台上指定 DAG 資料夾的路徑 (第一次)

以下步驟假設您指定的是 Amazon S3 儲存貯體上名為的資料夾的路徑dags。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇您要執行 DAG 的環境。
3. 選擇 編輯 。
4. 在 Amazon S3 中的 DAG 程式碼上，選擇 D AG 資料夾欄位旁邊的 [瀏覽 S3]。
5. 選取您的dags資料夾。
6. 選擇 Choose (選擇)。
7. 選擇下一步，更新環境。

檢視您的 Apache Airflow UI 上的變更

登錄到阿帕奇氣流

您需要AWS Identity and Access Management (IAM) 中的AWS帳戶阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess許可，才能檢視您的 Apache 氣流使用者介面。

存取您的 Apache Airflow UI

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇「開啟氣流 UI」。

後續步驟？

- 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。

安裝自定義插件

適用於 Apache 氣流的 Amazon 受管工作流程支援 Apache Airflow 的內建外掛程式管理器，可讓您使用自訂的 Apache 氣流操作員、掛鉤、感應器或介面。本頁說明使用檔案在 Amazon MWAA 環境上安裝 Apache 氣流自訂外掛程式的步驟。`plugins.zip`

內容

必要條件

您需要下列項目，才能完成此頁面上的步驟。

- 權限 — 您的AWS帳戶必須已被管理員授與您環境的 AmazonmWAA FullConsoleAccess 存取控制原則的存取權限。此外，您的執行角色必須允許 Amazon MWAA 環境，才能存取環境使用的AWS資源。
- 存取 — 如果您需要存取公用儲存庫，才能直接在 Web 伺服器上安裝相依性，則您的環境必須設定公用網路 Web 伺服器存取權。如需詳細資訊，請參閱the section called “阿帕奇氣流存取模式”。
- Amazon S3 組態 — 用於存放 DAG、自訂外掛程式和 Python 相依性的 Amazon S3 儲存貯體`requirements.txt`必須設定為封鎖公用存取並啟用版本控制。`plugins.zip`

運作方式

若要在您的環境中執行自訂外掛程式，您必須執行以下三項作業：

1. 在本機建立`plugins.zip`檔案。
2. 將本機`plugins.zip`檔案上傳到您的 Amazon S3 儲存貯體。
3. 在 Amazon MWAA 主控台的外掛程式檔案欄位中指定此檔案的版本。

> **Note**
>
> 如果這是您第一次上傳`plugins.zip`到 Amazon S3 儲存貯體，您還需要在 Amazon MWAA 主控台上指定檔案的路徑。您只需要完成此步驟一次。

v2 中有什麼變化

- 新功能：運營商，鉤子和執行者。DAG 中的匯入陳述式，以及您在 Amazon MWAA 中指定的自訂外掛程式已`plugins.zip`在 Apache 氣流 v1 和 Apache 氣流 v2 之間變更。例如，`from airflow.contrib.hooks.aws_hook import AwsHook`在阿帕奇氣流 V1 已經改變為`from airflow.providers.amazon.aws.hooks.base_aws import AwsBaseHook`在阿帕奇氣流 V2。若要進一步了解，請參閱 Apache 氣流參考指南中的 Python API 參考。
- 新：進口插件。不再支持導入運算符，傳感器，插件中添加的掛鉤使用`airflow.{operators,sensors,hooks}.<plugin_name>`。這些擴展應該作為常規 Python 模塊導入。

在 v2 及更高版本中，建議的方法是將它們放置在 DAG 目錄中，並創建並使用 .airflowignore 文件以將其排除為 DAG 進行解析。若要深入了解，請參閱 Apache Airflow 參考指南中的模組管理和建立自訂操作員。

自定義插件概述

Apache Airflow 的內置插件管理器可以通過簡單地將文件放在文件$AIRFLOW_HOME/plugins夾中將外部功能集成到其核心。它允許您使用自定義的 Apache 氣流操作員，掛鉤，傳感器或接口。下節提供本機開發環境中平坦與巢狀目錄結構的範例，以及產生的 import 陳述式，這些陳述式會決定 plugins.zip 中的目錄結構。

自定義插件目錄和大小限制

Apache 氣流排程器和工作人員會在啟動期間，在AWS受管理的 Fargate 容器上尋找自訂外掛程式，適用於您的環境。/usr/local/airflow/plugins/*

- 目錄結構。目錄結構 (at/*) 以plugins.zip檔案的內容為基礎。例如，如果您將operators目錄plugins.zip包含為頂層目錄，則該目錄將被解壓縮到您/usr/local/airflow/plugins/*operators*的環境中。
- 大小限制。我們建議使用小於 1 GB 的plugins.zip檔案。plugins.zip檔案大小越大，環境上的啟動時間就越長。雖然 Amazon MWAA 不會明確限制plugins.zip檔案的大小，但是如果無法在十分鐘內安裝相依性，Fargate 服務會逾時並嘗試將環境回復到穩定狀態。

> **Note**
>
> 對於使用 Apache 氣流 v1.10.12 或 Apache 氣流 v2.0.2 的環境，Amazon MWAA 會限制 Apache 氣流網頁伺服器上的輸出流量，且不允許您直接在網頁伺服器上安裝外掛程式或 Python 相依性。從 Apache 氣流 v2.2.2 開始，Amazon MWAA 可以直接在 Web 服務器上安裝插件和依賴關係。

自定義插件的例子

以下章節使用 Apache Airflow 參考指南中的範例程式碼，說明如何建構本機開發環境。

在 plugins.zip 中使用平面目錄結構的示例

Apache Airflow v2

下面的例子顯示了 Apache 氣流 V2 的扁平目錄結構的`plugins.zip`文件。

Example 帶有 PythonVirtualenvOperator plugins.zip 的平面目錄

下列範例顯示中 PythonVirtualenvOperator 自訂外掛程式之 plugins.zip 檔案的頂層樹狀結構為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator。

```
### virtual_python_plugin.py
```

Example plugins/virtual_python_plugin.py

下面的例子顯示了 PythonVirtualenvOperator 自定義插件。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow.plugins_manager import AirflowPlugin
import airflow.utils.python_virtualenv
from typing import List

def _generate_virtualenv_cmd(tmp_dir: str, python_bin: str, system_site_packages:
 bool) -> List[str]:
    cmd = ['python3','/usr/local/airflow/.local/lib/python3.7/site-packages/
virtualenv', tmp_dir]
    if system_site_packages:
```

```
        cmd.append('--system-site-packages')
    if python_bin is not None:
        cmd.append(f'--python={python_bin}')
    return cmd

airflow.utils.python_virtualenv._generate_virtualenv_cmd=_generate_virtualenv_cmd

class VirtualPythonPlugin(AirflowPlugin):
    name = 'virtual_python_plugin'
```

Apache Airflow v1

下面的例子顯示了 Apache 氣流 v1 一個扁平目錄結構的`plugins.zip`文件。

Example 帶有 PythonVirtualenvOperator plugins.zip 的平面目錄

下列範例顯示中 PythonVirtualenvOperator 自訂外掛程式之 plugins.zip 檔案的頂層樹狀結構為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator。

```
### virtual_python_plugin.py
```

Example plugins/virtual_python_plugin.py

下面的例子顯示了 PythonVirtualenvOperator 自定義插件。

```
from airflow.plugins_manager import AirflowPlugin
from airflow.operators.python_operator import PythonVirtualenvOperator

def _generate_virtualenv_cmd(self, tmp_dir):
    cmd = ['python3','/usr/local/airflow/.local/lib/python3.7/site-packages/
virtualenv', tmp_dir]
    if self.system_site_packages:
        cmd.append('--system-site-packages')
    if self.python_version is not None:
        cmd.append('--python=python{}'.format(self.python_version))
    return cmd
PythonVirtualenvOperator._generate_virtualenv_cmd=_generate_virtualenv_cmd

class EnvVarPlugin(AirflowPlugin):
    name = 'virtual_python_plugin'
```

在 plugins.zip 中使用嵌套目錄結構的示例

Apache Airflow v2

下列範例顯示的plugins.zip檔案具有不同目錄的hooksoperators、和 Apache 氣流 v2 的目sensors錄。

Example plugins.zip

```
__init__.py
my_airflow_plugin.py
hooks/
|-- __init__.py
|-- my_airflow_hook.py
operators/
|-- __init__.py
|-- my_airflow_operator.py
|-- hello_operator.py
sensors/
|-- __init__.py
|-- my_airflow_sensor.py
```

下列範例顯示 DAG (DAGs 資料夾) 中使用自訂外掛程式的匯入陳述式。

Example dags/your_dag.py

```
from airflow import DAG
from datetime import datetime, timedelta
from operators.my_airflow_operator import MyOperator
from sensors.my_airflow_sensor import MySensor
from operators.hello_operator import HelloOperator

default_args = {
 'owner': 'airflow',
 'depends_on_past': False,
 'start_date': datetime(2018, 1, 1),
 'email_on_failure': False,
 'email_on_retry': False,
 'retries': 1,
 'retry_delay': timedelta(minutes=5),
}
```

```
with DAG('customdag',
   max_active_runs=3,
   schedule_interval='@once',
   default_args=default_args) as dag:

 sens = MySensor(
  task_id='taskA'
 )

 op = MyOperator(
  task_id='taskB',
  my_field='some text'
 )

 hello_task = HelloOperator(task_id='sample-task', name='foo_bar')



 sens >> op >> hello_task
```

Example plugins/my_airflow_plugin.py

```
from airflow.plugins_manager import AirflowPlugin
from hooks.my_airflow_hook import *
from operators.my_airflow_operator import *

class PluginName(AirflowPlugin):

    name = 'my_airflow_plugin'

    hooks = [MyHook]
    operators = [MyOperator]
    sensors = [MySensor]
```

下列範例顯示自訂外掛程式檔案中所需的每個 import 陳述式。

Example hooks/my_airflow_hook.py

```
from airflow.hooks.base import BaseHook


class MyHook(BaseHook):
```

```
    def my_method(self):
        print("Hello World")
```

Example sensors/my_airflow_sensor.py

```
from airflow.sensors.base import BaseSensorOperator
from airflow.utils.decorators import apply_defaults


class MySensor(BaseSensorOperator):

    @apply_defaults
    def __init__(self,
                 *args,
                 **kwargs):
        super(MySensor, self).__init__(*args, **kwargs)

    def poke(self, context):
        return True
```

Example operators/my_airflow_operator.py

```
from airflow.operators.bash import BaseOperator
from airflow.utils.decorators import apply_defaults
from hooks.my_airflow_hook import MyHook


class MyOperator(BaseOperator):

    @apply_defaults
    def __init__(self,
                 my_field,
                 *args,
                 **kwargs):
        super(MyOperator, self).__init__(*args, **kwargs)
        self.my_field = my_field

    def execute(self, context):
        hook = MyHook('my_conn')
        hook.my_method()
```

Example operators/hello_operator.py

```
from airflow.models.baseoperator import BaseOperator
from airflow.utils.decorators import apply_defaults

class HelloOperator(BaseOperator):

    @apply_defaults
    def __init__(
            self,
            name: str,
            **kwargs) -> None:
        super().__init__(**kwargs)
        self.name = name

    def execute(self, context):
        message = "Hello {}".format(self.name)
        print(message)
        return message
```

請遵循使用 Amazon MWAA CLI 公用程式測試自訂外掛程式中的步驟，然後建立 plugins.zip 檔案以壓縮目錄中的**plugins**內容。例如 `cd plugins`。

Apache Airflow v1

下列範例顯示的`plugins.zip`檔案具有不同目錄的`hooks`、`operators`、以及 Apache 氣流 v1.10.12 的目`sensors`錄。

Example plugins.zip

```
__init__.py
my_airflow_plugin.py
hooks/
    |-- __init__.py
    |-- my_airflow_hook.py
operators/
    |-- __init__.py
    |-- my_airflow_operator.py
    |-- hello_operator.py
sensors/
    |-- __init__.py
    |-- my_airflow_sensor.py
```

下列範例顯示 DAG (DAGs 資料夾) 中使用自訂外掛程式的匯入陳述式。

Example dags/your_dag.py

```
from airflow import DAG
from datetime import datetime, timedelta
from operators.my_operator import MyOperator
from sensors.my_sensor import MySensor
from operators.hello_operator import HelloOperator

default_args = {
 'owner': 'airflow',
 'depends_on_past': False,
 'start_date': datetime(2018, 1, 1),
 'email_on_failure': False,
 'email_on_retry': False,
 'retries': 1,
 'retry_delay': timedelta(minutes=5),
}


with DAG('customdag',
   max_active_runs=3,
   schedule_interval='@once',
   default_args=default_args) as dag:

 sens = MySensor(
  task_id='taskA'
 )

 op = MyOperator(
  task_id='taskB',
  my_field='some text'
 )

 hello_task = HelloOperator(task_id='sample-task', name='foo_bar')



 sens >> op >> hello_task
```

Example plugins/my_airflow_plugin.py

```
from airflow.plugins_manager import AirflowPlugin
from hooks.my_airflow_hook import *
from operators.my_airflow_operator import *
from utils.my_utils import *

class PluginName(AirflowPlugin):

    name = 'my_airflow_plugin'

    hooks = [MyHook]
    operators = [MyOperator]
    sensors = [MySensor]
```

下列範例顯示自訂外掛程式檔案中所需的每個 import 陳述式。

Example hooks/my_airflow_hook.py

```
from airflow.hooks.base_hook import BaseHook


class MyHook(BaseHook):

    def my_method(self):
        print("Hello World")
```

Example sensors/my_airflow_sensor.py

```
from airflow.sensors.base_sensor_operator import BaseSensorOperator
from airflow.utils.decorators import apply_defaults


class MySensor(BaseSensorOperator):

    @apply_defaults
    def __init__(self,
                 *args,
                 **kwargs):
        super(MySensor, self).__init__(*args, **kwargs)

    def poke(self, context):
```

```
        return True
```

Example operators/my_airflow_operator.py

```
from airflow.operators.bash_operator import BaseOperator
from airflow.utils.decorators import apply_defaults
from hooks.my_hook import MyHook


class MyOperator(BaseOperator):

    @apply_defaults
    def __init__(self,
                 my_field,
                 *args,
                 **kwargs):
        super(MyOperator, self).__init__(*args, **kwargs)
        self.my_field = my_field

    def execute(self, context):
        hook = MyHook('my_conn')
        hook.my_method()
```

Example operators/hello_operator.py

```
from airflow.models.baseoperator import BaseOperator
from airflow.utils.decorators import apply_defaults

class HelloOperator(BaseOperator):

    @apply_defaults
    def __init__(
            self,
            name: str,
            **kwargs) -> None:
        super().__init__(**kwargs)
        self.name = name

    def execute(self, context):
        message = "Hello {}".format(self.name)
        print(message)
        return message
```

請遵循使用 Amazon MWAA CLI 公用程式測試自訂外掛程式中的步驟，然後建立 plugins.zip 檔案以壓縮目錄中的**plugins**內容。例如 `cd plugins`。

創建一個 plugins.zip 文件

下列步驟說明我們建議在本機建立 plugins.zip 檔案的步驟。

步驟一：使用 Amazon MWAA CLI 公用程式測試自訂外掛程式

- 命令列介面 (CLI) 公用程式可在本機複寫 Apache 氣流環境的 Amazon 受管工作流程。
- CLI 會在本機建立類似於 Amazon MWAA 生產映像的碼頭容器映像。這可讓您執行本機 Apache 氣流環境來開發和測試 DAG、自訂外掛程式和相依性，然後再部署到 Amazon MWAA。
- 要運行 CLI，請參閱（詳見） GitHub。aws-mwaa-local-runner

第二步：建立 plugins.zip 檔案

您可以使用內建的 ZIP 封存公用程式或任何其他 ZIP 公用程式 (例如 7zip) 來建立 .zip 檔案。

> **Note**
>
> 當您建立 .zip 檔案時，Windows 作業系統的內建 zip 公用程式可能會新增子資料夾。我們建議您在上傳到 Amazon S3 儲存貯體之前先驗證 plugins.zip 檔案的內容，以確保沒有新增其他目錄。

1. 將目錄變更為您的本機 Airflow 外掛程式目錄。例如：

```
myproject$ cd plugins
```

2. 執行下列命令，以確保內容具有可執行權限 (僅限 macOS 和 Linux)。

```
plugins$ chmod -R 755 .
```

3. 壓縮文plugins件夾中的內容。

```
plugins$ zip -r plugins.zip .
```

上傳`plugins.zip`到 Amazon S3

您可以使用 Amazon S3 主控台或 AWS Command Line Interface (AWS CLI) 將`plugins.zip`檔案上傳到 Amazon S3 儲存貯體。

使用 AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面上的步驟，您需要下列項目：

- AWS CLI— 安裝版本 2.
- AWS CLI— 快速配置 `aws configure`.

若要使用 AWS CLI

1. 在命令提示字元中，導覽至儲存`plugins.zip`檔案的目錄。例如：

```
cd plugins
```

2. 使用下列命令列出所有 Amazon S3 儲存貯體。

```
aws s3 ls
```

3. 使用下列命令列出您環境之 Amazon S3 儲存貯體中的檔案和資料夾。

```
aws s3 ls s3://YOUR_S3_BUCKET_NAME
```

4. 使用下列命令將`plugins.zip`檔案上傳到您環境的 Amazon S3 儲存貯體。

```
aws s3 cp plugins.zip s3://YOUR_S3_BUCKET_NAME/plugins.zip
```

使用 Amazon S3 主控台

Amazon S3 主控台是基於 Web 的使用者界面，可讓您建立和管理 Amazon S3 儲存貯體中的資源。

使用 Amazon S3 主控台上傳

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。

3. 在 S3 窗格的 DAG 程式碼中選取 S3 儲存貯體連結，以在 Amazon S3 主控台上開啟儲存貯體。
4. 選擇上傳。
5. 選擇 [新增檔案]。
6. 選擇您的本地副本`plugins.zip`，選擇上傳。

在環境中安裝自訂外掛程式

本節說明如何安裝您上傳到 Amazon S3 儲存貯體的自訂外掛程式，方法是指定 plugins.zip 檔案的路徑，並在每次更新 zip 檔案時指定 plugins.zip 檔案的版本。

在 Amazon MWAA 主控台**`plugins.zip`**上指定路徑 (第一次)

如果這是您第一次上傳`plugins.zip`到 Amazon S3 儲存貯體，您還需要在 Amazon MWAA 主控台上指定檔案的路徑。您只需要完成此步驟一次。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯 。
4. 在 Amazon S3 的 DAG 程式碼窗格中，選擇外掛程式檔案-選用欄位旁邊的瀏覽 S3。
5. 選取您的 Amazon S3 儲存貯體上的`plugins.zip`檔案。
6. 選擇 Choose (選擇)。
7. 選擇下一步，更新環境。

在 Amazon MWAA 主控台上指定**`plugins.zip`**版本

每次在 Amazon Amazon S3 上傳新版本時，都需要在 Amazon MWAA 主控台上指定`plugins.zip`檔案`plugins.zip`的版本。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯 。
4. 在 Amazon S3 中的 DAG 程式碼上，從下拉式清單中選擇一個`plugins.zip`版本。
5. 選擇下一步。

plugins.zip 的範例使用案例

- 瞭解如何在中建立自訂外掛程式自定義插件與阿帕奇蜂巢和 Hadoop。
- 瞭解如何在中建立自訂外掛程式自定義插件來修補PythonVirtualenvOperator 。
- 瞭解如何在中建立自訂外掛程式自定義插件與甲骨文。
- 瞭解如何在中建立自訂外掛程式the section called “變更 DAG 的時區”。

後續步驟？

- 使用 aws-mwaa-local-runneron GitHub 在本機測試您的 DAG、自訂外掛程式和 Python 相依性。

安裝 Python 的依賴

Python 相依性是指任何未包含在 Apache 氣流基本安裝中的套件或散發，適用於 Apache 氣流環境的 Amazon 管理工作流程中，適用於 Apache 氣流版本。本頁說明使用 Amazon Amazon S3 中的`requirements.txt`檔案在您的亞馬遜 MWAA 環境上安裝 Apache 氣流 Python 相依性的步驟。

內容

- 後續步驟？

必要條件

您需要下列項目，才能完成此頁面上的步驟。

- 權限 — 您的AWS帳戶必須已被管理員授與您環境的 AmazonmWAA FullConsoleAccess 存取控制原則的存取權限。此外，您的執行角色必須允許 Amazon MWAA 環境，才能存取環境使用的AWS資源。
- 存取 — 如果您需要存取公用儲存庫，才能直接在 Web 伺服器上安裝相依性，則您的環境必須設定公用網路 Web 伺服器存取權。如需詳細資訊，請參閱 the section called “阿帕奇氣流存取模式”。
- Amazon S3 組態 — 用於存放 DAG、自訂外掛程式和 Python 相依性的 Amazon S3 儲存貯體`requirements.txt`必須設定為封鎖公用存取並啟用版本控制。`plugins.zip`

運作方式

在 Amazon MWAA 上，您可以安裝所有 Python 相依性，方法是將`requirements.txt`檔案上傳到 Amazon S3 儲存貯體，然後在每次更新檔案時在 Amazon MWAA 主控台上指定檔案的版本。Amazon MWAA 運`pip3 install -r requirements.txt`行在 Apache 氣流調度程序和每個工作程序上安裝 Python 依賴關係。

要在您的環境上運行 Python 依賴關係，您必須執行以下三件事：

1. 在本機建立`requirements.txt`檔案。
2. `requirements.txt`將本機上傳到您的 Amazon S3 儲存貯體。
3. 在 Amazon MWAA 主控台的 [需求檔案] 欄位中指定此檔案的版本。

> **Note**
>
> 如果這是您第一次建立並上傳`requirements.txt`到 Amazon S3 儲存貯體，您還需要在 Amazon MWAA 主控台上指定檔案的路徑。您只需要完成此步驟一次。

Python 依賴關係概述

您可以從 Python Package 索引（PyPi.org），Python 輪子（`.whl`）或託管在私有 PyPi /PEP-503 兼容存儲庫上的 Python 依賴項安裝 Apache 氣流附加功能和其他 Python 依賴關係。

Python 依賴關係的位置和大小限制

Apache 氣流排程器和工作人員會在啟動期間，在AWS受管理的 Fargate 容器上尋找自訂外掛程式，適用於您的環境。`/usr/local/airflow/plugins`

- 大小限制。我們建議使`requirements.txt`用參考組合大小小小於 1 GB 的程式庫的檔案。Amazon MWAA 需要安裝的程式庫越多，環境上的啟動時間就越長。雖然 Amazon MWAA 並未明確限制已安裝程式庫的大小，但是如果無法在十分鐘內安裝相依性，Fargate 服務會逾時並嘗試將環境回復到穩定狀態。

創建一個 requirements.txt 文件

下列步驟說明我們建議在本機建立 requirements.txt 檔案的步驟。

步驟一：使用 Amazon MWAA CLI 公用程式測試 Python 相依性

- 命令列介面 (CLI) 公用程式可在本機複寫 Apache 氣流環境的 Amazon 受管工作流程。
- CLI 會在本機建立類似於 Amazon MWAA 生產映像的泊塢視窗容器映像。這可讓您執行本機 Apache 氣流環境來開發和測試 DAG、自訂外掛程式和相依性，然後再部署到 Amazon MWAA。
- 要運行 CLI，請參閱（詳見）GitHub。aws-mwaa-local-runner

第二步：創建 **`requirements.txt`**

以下部分介紹如何從`requirements.txt`文件中的 Python P ackage 索引指定 Python 依賴關係。

Apache Airflow v2

1. 在本地測試。在建立檔案之前，以反覆方式新增其他程式庫，以尋找套件及其版本的正確組合。`requirements.txt`若要執行 Amazon MWAA CLI 公用程式，請參閱（詳見）。aws-mwaa-local-runner GitHub
2. 檢閱 Apache 氣流套件附加功能。若要檢視 Amazon MWAA 上為 Apache 氣流 v2 安裝的套件清單，請參閱網站上的 Amazon MWAA 本地運`requirements.txt`行器。 GitHub

3. 新增條件約束陳述式。在檔案頂端新增 Apache 氣流 v2 環境的限制`requirements.txt`檔案。Apache 氣流限制檔案會指定 Apache 氣流發行時可用的提供者版本。

   從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一`--constraint`份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。

   在下面的例子中，將 *{####}* 替換為您環境的版本號，並將 *{Python ##}* 替換為與您的環境兼容的 Python 版本。

   如需與 Apache 氣流環境相容的 Python 版本的相關資訊，請參閱 Apache 氣流版本。

   ```
   --constraint "https://raw.githubusercontent.com/apache/airflow/
   constraints-{Airflow-version}/constraints-{Python-version}.txt"
   ```

   如果條件約束檔案判斷`xyz==1.0`套件與您環境中的其他套件不相容，`pip3 install`將無法防止不相容的程式庫安裝到您的環境中。如果任何套件的安裝失敗，您可以在記錄檔上的對應記錄資料流中檢視每個 Apache Airflow 元件 (排程器、Worker 和 Web 伺服器) 的錯誤 CloudWatch 記錄。如需記錄檔類型的詳細資訊，請參閱the section called “檢視氣流記錄”。

4. 阿帕奇氣流包。添加包附加功能和版本（ == ）。這有助於防止在您的環境中安裝相同名稱但版本不同的套件。

   ```
   apache-airflow[package-extra]==2.5.1
   ```

5. Python 庫。在文件中添加軟`requirements.txt`件包名稱和版本（ == ）。這有助於防止將 future 自 PyPi.org 的重大更新被自動套用。

   ```
   library == version
   ```

   Example 肉毒桿菌毒素 3 和精神 2-二進制

   此範例是為了展示目的而提供。博托和 psycopg2 二進制庫包含在 Apache 氣流 v2 基本安裝中，不需要在文件中指定。`requirements.txt`

   ```
   boto3==1.17.54
   boto==2.49.0
   botocore==1.20.54
   psycopg2-binary==2.8.6
   ```

如果指定的套件沒有版本，Amazon MWAA 會從 PyPi .org 安裝最新版本的套件。此版本可能會與`requirements.txt`.

Apache Airflow v1

1. 在本地測試。在建立檔案之前，以反覆方式新增其他程式庫，以尋找套件及其版本的正確組合。`requirements.txt`若要執行 Amazon MWAA CLI 公用程式，請參閱（詳見）。aws-mwaa-local-runner GitHub
2. 檢閱氣流套件附加功能。檢閱可用於阿帕奇氣流 v1.10.12 的套件清單，網址為 https://raw.githubusercontent.com/apache/airflow/constraints-1.10.12/constraints-3.7.txt。
3. 加入約束檔案。將 Apache 氣流 v1.10.12 的限制文件添加到文件的頂部。`requirements.txt`如果條件約束檔案判斷`xyz==1.0`套件與您環境中的其他套件不相容，`pip3 install`將無法防止不相容的程式庫安裝到您的環境中。

```
--constraint "https://raw.githubusercontent.com/apache/airflow/
constraints-1.10.12/constraints-3.7.txt"
```

4. 阿帕奇氣流 1.10.12 版軟件包。添加氣流包附加功能和阿帕奇氣流 v1.10.12 版本（）。==這有助於防止在您的環境中安裝相同名稱但版本不同的套件。

```
apache-airflow[package]==1.10.12
```

Example 安全殼層

下面的示例`requirements.txt`文件安裝 SSH 阿帕奇氣流 v1.10.12。

```
apache-airflow[ssh]==1.10.12
```

5. Python 庫。在文件中添加軟`requirements.txt`件包名稱和版本（==）。這有助於防止將 future 自 PyPi.org 的重大更新被自動套用。

```
library == version
```

Example Boto3

下面的示例`requirements.txt`文件安裝的博托 3 庫阿帕奇氣流 v1.10.12。

```
boto3 == 1.17.4
```

如果指定的套件沒有版本，Amazon MWAA 會從 PyPi .org 安裝最新版本的套件。此版本可能會與requirements.txt.

上傳requirements.txt到 Amazon S3

您可以使用 Amazon S3 主控台或 AWS Command Line Interface (AWS CLI) 將requirements.txt檔案上傳到您的 Amazon S3 儲存貯體。

使用 AWS CLI

AWS Command Line Interface (AWS CLI) 是開放原始碼工具，可讓您在命令列 Shell 中使用命令來與 AWS 服務互動。若要完成此頁面上的步驟，您需要下列項目：

- AWS CLI— 安裝版本 2.
- AWS CLI— 快速配置 aws configure.

若要使用 AWS CLI

1. 使用下列命令列出您所有的 Amazon S3 儲存貯體。

```
aws s3 ls
```

2. 使用下列命令列出您環境之 Amazon S3 儲存貯體中的檔案和資料夾。

```
aws s3 ls s3://YOUR_S3_BUCKET_NAME
```

3. 下列命令會將requirements.txt檔案上傳到 Amazon S3 儲存貯體。

```
aws s3 cp requirements.txt s3://YOUR_S3_BUCKET_NAME/requirements.txt
```

使用 Amazon S3 主控台

Amazon S3 主控台是一個以網路為基礎的使用者界面，可讓您建立和管理 Amazon S3 儲存貯體中的資源。

使用 Amazon S3 主控台上傳

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 S3 窗格的 DAG 程式碼中選取 S3 儲存貯體連結，以在 Amazon S3 主控台上開啟儲存貯體。
4. 選擇上傳。
5. 選擇 [新增檔案]。
6. 選擇您的本地副本`requirements.txt`，選擇上傳。

在您的環境中安裝 Python 相依性

本節說明如何透過指定 requirements.txt 檔案的路徑，並在每次更新 requirements.txt 檔案時指定檔案的版本，以安裝您上傳到 Amazon S3 儲存貯體的相依性。

在 Amazon MWAA 主控台**`requirements.txt`**上指定路徑 (第一次)

如果這是您第一次建立並上傳`requirements.txt`到 Amazon S3 儲存貯體，您還需要在 Amazon MWAA 主控台上指定檔案的路徑。您只需要完成此步驟一次。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯。
4. 在 Amazon S3 窗格中的 DAG 程式碼上，選擇需求檔案-選用欄位旁邊的瀏覽 S3。
5. 選取您的 Amazon S3 儲存貯體上的`requirements.txt`檔案。
6. 選擇 Choose (選擇)。
7. 選擇下一步，更新環境。

您可以在環境完成更新後立即開始使用新套件。

在 Amazon MWAA 主控台上指定**`requirements.txt`**版本

每次在 Amazon Amazon S3 上傳新版本時，都需要在 Amazon MWAA 主控台上指定`requirements.txt`檔案`requirements.txt`的版本。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。

2. 選擇一個環境。
3. 選擇編輯。
4. 在 Amazon S3 的 DAG 程式碼窗格中，從下拉式清單中選擇一個`requirements.txt`版本。
5. 選擇下一步，更新環境。

您可以在環境完成更新後立即開始使用新套件。

檢視您的記錄 **requirements.txt**

您可以檢視排程器排程工作流程和剖析`dags`資料夾的 Apache Airflow 記錄。下列步驟說明如何在 Amazon MWAA 主控台上開啟排程器的日誌群組，以及如何在日誌主控台上檢視 Apache 氣流 CloudWatch 日誌。

若要檢視 **requirements.txt**

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [監視] 窗格中選擇 Airflow 排程器記錄群組。
4. 在「`requirements_install_ip`記錄資料流」中選擇記錄檔。
5. 您應該會看到已安裝在環境中的套件清單，位於`/usr/local/airflow/.local/bin`。例如：

```
Collecting appdirs==1.4.4 (from -r /usr/local/airflow/.local/bin (line 1))
Downloading https://files.pythonhosted.org/
packages/3b/00/2344469e2084fb28kjdsfiuyweb47389789vxbmnbjhsdgf5463acd6cf5e3db69324/
appdirs-1.4.4-py2.py3-none-any.whl
Collecting astroid==2.4.2 (from -r /usr/local/airflow/.local/bin (line 2))
```

6. 檢閱套件清單，以及這些套件是否在安裝過程中遇到錯誤。如果發生錯誤，您可能會看到類似下列內容的錯誤：

```
2021-03-05T14:34:42.731-07:00
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
```

後續步驟？

- 使用 aws-mwaa-local-runneron GitHub 在本機測試您的 DAG、自訂外掛程式和 Python 相依性。

刪除 Amazon S3 上的文件

本頁說明在適用於 Apache 氣流環境的 Amazon S3 儲存貯體中版本控制如何運作，以及刪除 DAG 或`requirements.txt`檔案的步驟。`plugins.zip`

內容

先決條件

您需要下列項目，才能完成此頁面上的步驟。

- 權限 — 您的AWS帳戶必須已被管理員授與您環境的 AmazonmWAA FullConsoleAccess 存取控制原則的存取權限。此外，您的執行角色必須允許 Amazon MWAA 環境，才能存取環境使用的AWS資源。
- 存取 — 如果您需要存取公用儲存庫，才能直接在 Web 伺服器上安裝相依性，則您的環境必須設定公用網路 Web 伺服器存取權。如需詳細資訊，請參閱the section called “阿帕奇氣流存取模式”。
- Amazon S3 組態 — 用於存放 DAG、自訂外掛程式和 Python 相依性的 Amazon S3 儲存貯體`requirements.txt`必須設定為封鎖公用存取並啟用版本控制。`plugins.zip`

版本概觀

requirements.txt和您plugins.zip的 Amazon S3 存儲桶中的版本控制。當某個物件啟用 Amazon S3 儲存貯體版本控制，且從 Amazon S3 儲存貯體刪除成品 (例如 plugins.zip) 時，檔案不會完全刪除。每當 Amazon S3 上刪除工件時，就會創建一個新的文件副本，該文件副本是 404（未找到對象）錯誤/0k 文件，顯示「我不在這裡」。Amazon S3 稱此為刪除標記。刪除標記是文件的「空」版本，其密鑰名稱（或密鑰）和版本 ID 與任何其他對象一樣。

建議您定期刪除檔案版本並刪除標記，以降低 Amazon S3 儲存貯體的儲存成本。要完全刪除「非當前」（以前的）文件版本，您必須刪除文件的版本，然後刪除該版本的刪除標記。

運作方式

亞馬遜 MWAA 每三十秒在您的 Amazon S3 儲存貯體上執行一次同步操作。這會導致 Amazon S3 儲存貯體中的任何 DAG 刪除同步到 Fargate 容器的氣流影像。

對於plugins.zip和requirements.txt檔案，只有在 Amazon MWAA 使用自訂外掛程式和 Python 相依性建立 Fargate 容器的新氣流影像時，才會在環境更新後發生變更。如果您刪除任何requirements.txt或plugins.zip檔案的目前版本，然後在未為已刪除檔案提供新版本的情況下更新環境，則更新將會失敗並顯示錯誤訊息，例如「無法讀取檔案版本 {version}{file}」。

刪除 Amazon S3 上的 DAG

DAG 檔案 (.py) 未進行版本控制，可以直接在 Amazon S3 主控台上刪除。下列步驟說明如何刪除 Amazon S3 儲存貯體上的 DAG。

若要刪除 DAG

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 S3 窗格的 DAG 程式碼中選取 S3 儲存貯體連結，以在 Amazon S3 主控台上開啟儲存貯體。
4. 選擇 dags 資料夾。
5. 選取 DAG，刪除。
6. 在刪除物件？ ，鍵入delete。
7. 選擇 Delete objects (刪除物件)。

> **Note**
>
> 阿帕奇氣流保留歷史 DAG 運行。在 Apache 氣流中執行 DAG 之後，無論檔案狀態為何，它都會保留在氣流 DAG 清單中，直到您在 Apache 氣流中將其刪除為止。若要刪除 Apache 氣流中的 DAG，請選擇 [連結] 欄下的紅色 [刪除] 按鈕。

從環境中移除「目前的」requirements.txt 或 plugins.zip

目前，沒有辦法在添加 plugins.zip 或 requirements.txt 後從環境中刪除它們，但我們正在處理這個問題。在過渡期間，因應措施是分別指向空白文字或 zip 檔案。

刪除「非目前」(先前的) requirements.txt 或 plugins.zip 版本

Amazon S3 儲存貯體中的`requirements.txt`和`plugins.zip`檔案是在亞馬遜 MWAA 上進行版本控制的。如果您想要完全刪除 Amazon S3 儲存貯體上的這些檔案，您必須擷取物件的目前版本 (121212) (例如 plugins.zip)、刪除版本，然後移除檔案版本的刪除標記。

您也可以在 Amazon S3 主控台上刪除「非目前」(先前) 檔案版本；不過，您仍然需要使用下列其中一個選項刪除刪除標記。

- 若要擷取物件版本，請參閱 Amazon S3 指南中的從啟用版本控制的儲存貯體擷取物件版本。
- 若要刪除物件版本，請參閱 Amazon S3 指南中的從啟用版本控制的儲存貯體刪除物件版本。
- 若要移除刪除標記，請參閱 Amazon S3 指南中的管理刪除標記。

使用生命週期刪除「非目前」(舊版) 並自動刪除標識

您可以為 Amazon S3 儲存貯體設定生命週期政策，在特定天數後刪除 Amazon S3 儲存貯體中的「非目前」(先前) 版本的 plugins.zip 和 requirements.txt 檔案，或移除過期物件的刪除標記。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 Amazon S3 的 DAG 代碼下，選擇您的 Amazon S3 存儲桶。
4. 選擇建立生命週期規則。

範例生命週期原則：刪除 requirements.txt「非目前」版本並自動刪除標記

下列範例顯示如何建立生命週期規則，該規則會在三十天後永久刪除 requirements.txt 檔案的「非目前」版本及其刪除標記。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 Amazon S3 的 DAG 代碼下，選擇您的 Amazon S3 存儲桶。
4. 選擇建立生命週期規則。
5. 在生命週期規則名稱中，鍵入`Delete previous requirements.txt versions and delete markers after thirty days`。
6. 在前綴中，要求。
7. 在「生命週期規則動作」 中，選擇「永久刪除舊版物件」和「刪除過期的刪除標記」或「不完整的分段上傳
8. 在物件變成先前版本後的天數中，鍵入`30`。
9. 在過期物件刪除標記中，選擇刪除過期的物件刪除標記，物件會在 30 天後永久刪除。

後續步驟？

- 若要進一步了解 Amazon S3 刪除標記，請參閱管理刪除標記。
- 在即將到期的物件中進一步了解 Amazon S3 生命週期。

聯網

本指南說明 Amazon MWAA 環境所需的 Amazon 虛擬私人 VPC 網路設定。

章節

關於 Amazon MWAA 上的聯網

Amazon VPC 是連結到您 AWS 帳戶的虛擬網路。它提供對虛擬基礎架構和網路流量分段的精細控制，為您提供雲端安全性和動態擴展的能力。本頁說明具有公有路由或私有路由的 Amazon VPC 基礎設施，這些基礎設施需要支援 Apache Airflow 環境的 Amazon 受管工作流程。

內容

條款

公共路由

可存取網際網路的 Amazon VPC 人雲端網路。

私人路由

無法存取網際網路的 Amazon VPC 人雲端網路。

支援的項目

下表說明 Amazon VPC Amazon MWAA 支援的類型。

Amazon VPC 類型	支援	
嘗試建立環境的帳戶所擁有的 Amazon VPC。	是	
共用的 Amazon VPC，可讓多個 AWS 帳戶建立其 AWS 資源。	是	

VPC 基礎架構概觀

當您建立 Amazon MWAA 環境時，Amazon MWAA 會根據您為環境選擇的 Apache 氣流存取模式，在您的環境中建立一到兩個 VPC 端點。這些端點在您的 Amazon VPC 中顯示為具有私有 IP 的彈性網路界面 (ENI)。建立這些端點之後，傳往這些 IP 的任何流量都會以私密方式或公開路由到您的環境所使用的對應 AWS 服務。

下節說明透過網際網路公開路由或在 Amazon VPC 內私下路由流量所需的 Amazon V PC 基礎設施。

網際網路上的公用路由

本節說明具有公有路由之環境的 Amazon VPC 基礎設施。您將需要以下 VPC 基礎結構：

- 一個 VPC 安全群組。虛擬私人 VPC 安全群組充當虛擬防火牆，以控制執行個體上的輸入 (輸入) 和輸出 (輸出) 網路流量。
  - 最多可以指定 5 個安全群組。
  - 安全性群組必須為自己指定自我參照的輸入規則。
  - 安全性群組必須為所有流量 (`0.0.0.0/0`) 指定輸出規則。
  - 安全性群組必須允許自我參照規則中的所有流量。例如 (建議) 範例所有存取自我參照安全性群組。

  - 安全群組可以選擇性地透過指定 HTTPS 連接埠範圍和 TCP 連接埠範圍的連接埠範圍443，進一步限制流量5432。例如，(選擇性) 限制連接埠 5432 輸入存取權的安全性群組範例 和 (選擇性) 限制連接埠 443 輸入存取權的安全性群組範例。
- 兩個公用子網路。公有子網路是一種子網路，其與具有網際網路閘道路由的路由表相關聯。
  - 需要兩個公用子網路。這可讓 Amazon MWAA 在其他可用區域中的環境建立新的容器映像 (如果其中一個容器發生故障)。
  - 子網路必須位於不同的可用區域中。例如 `us-east-1a` 和 `us-east-1b`。
  - 子網路必須路由至具有彈性 IP 位址 (EIP) 的 NAT 閘道 (或 NAT 執行個體)。
  - 子網路必須有一個路由表，可將連結網際網路的流量導向至網際網路閘道。
- 兩個私人子網路。私有子網路是與具有網際網路閘道路由之路由表沒有關聯的子網路。
  - 需要兩個私人子網路。這可讓 Amazon MWAA 在其他可用區域中的環境建立新的容器映像 (如果其中一個容器發生故障)。
  - 子網路必須位於不同的可用區域中。例如 `us-east-1a` 和 `us-east-1b`。
  - 子網路必須有 NAT 裝置 (閘道或執行個體) 的路由表。
  - 子網路不得路由到網際網路閘道。
- 網路存取控制清單 (ACL)。NACL 會在子網路層級管理 (透過允許或拒絕規則) 入站和輸出流量。
  - NACL 必須具有允許所有流量 (`0.0.0.0/0`) 的輸入規則。
  - NACL 必須具有拒絕所有流量的輸出規則 () `0.0.0.0/0`。
  - 例如 (建議使用) ACL 範例。
- 兩個 NAT 閘道 (或稱 NAT 執行個體)。NAT 裝置會將流量從私有子網路中的執行個體轉送至網際網路或其他 AWS 服務，然後將回應路由傳回執行個體。
  - NAT 裝置必須附加至公用子網路。(每個公有子網路一個 NAT 裝置。)
  - NAT 裝置必須有一個彈性 IPv4 位址 (EIP) 連接至每個公用子網路。
- 網際網路閘道。網際網路閘道會將 Amazon VPC 連線到網際網路和其他 AWS 服務。
  - 必須將網際網路閘道連接至 Amazon VPC。

沒有網際網路存取的私人

本節說明具有私有路由之環境的 Amazon VPC 基礎設施。您將需要以下 VPC 基礎結構：

- 一個 VPC 安全群組。虛擬私人 VPC 安全群組充當虛擬防火牆，以控制執行個體上的輸入 (輸入) 和輸出 (輸出) 網路流量。
  - 最多可以指定 5 個安全群組。

  - 安全性群組必須為自己指定自我參照的輸入規則。
  - 安全性群組必須為所有流量 (`0.0.0.0/0`) 指定輸出規則。
  - 安全性群組必須允許自我參照規則中的所有流量。例如 (建議) 範例所有存取自我參照安全性群組。
  - 安全群組可以選擇性地透過指定 HTTPS 連接埠範圍和 TCP 連接埠範圍的連接埠範圍443，進一步限制流量5432。例如，(選擇性) 限制連接埠 5432 輸入存取權的安全性群組範例 和 (選擇性) 限制連接埠 443 輸入存取權的安全性群組範例。
- 兩個私人子網路。私有子網路是與具有網際網路閘道路由之路由表沒有關聯的子網路。
  - 需要兩個私人子網路。這可讓 Amazon MWAA 在其他可用區域中的環境建立新的容器映像 (如果其中一個容器發生故障)。
  - 子網路必須位於不同的可用區域中。例如 `us-east-1a` 和 `us-east-1b`。
  - 子網路必須具有通往 VPC 端點的路由表。
  - 子網路不得有 NAT 裝置 (閘道或執行個體) 的路由表，也不能有網際網路閘道。
- 網路存取控制清單 (ACL)。NACL 會在子網路層級管理 (透過允許或拒絕規則) 入站和輸出流量。
  - NACL 必須具有允許所有流量 (`0.0.0.0/0`) 的輸入規則。
  - NACL 必須具有拒絕所有流量的輸出規則 () `0.0.0.0/0`。
  - 例如 (建議使用) ACL 範例。
- 本地路由表。本機路由表是 VPC 內通訊的預設路由。
  - 本機路由表必須與您的私有子網路相關聯。
  - 本機路由表必須讓 VPC 中的執行個體能夠與您自己的網路通訊。例如，如果您使用存取 Apache Airflow 網頁伺服器的虛擬私人雲端介面端點，則路由表必須路由至 VPC 端點。 AWS Client VPN
- 您的環境所使用之每項 AWS 服務的 VPC 端點，以及位於相同 AWS 區域的 Apache 氣流 VPC 端點，以及作為 Amazon MWAA 環境的 Amazon VPC 端點。
  - 環境所使用之每項 AWS 服務的 VPC 端點，以及 Apache 氣流的 VPC 端點。例如 (必要) VPC 端點。
  - VPC 端端點必須啟用私人 DNS。
  - VPC 端點必須與您環境的兩個私有子網路相關聯。
  - VPC 端點必須與環境的安全性群組相關聯。
  - 應將每個端點的 VPC 端點原則設定為允許存取環境所使用的 AWS 服務。例如 (建議) 允許所有存取的 VPC 端點政策範。
  - Amazon S3 的 VPC 私人雲端節點政策應設定為允許儲存貯體存取。例如 (建議使用) 允許儲存貯體存取的 Amazon S3 閘道端點政策範例。

Amazon VPC 和 Apache 氣流存取模式的範例使用案例

本節介紹 Amazon VPC 中網路存取的不同使用案例，以及您應該在 Amazon MWAA 主控台上選擇的 Apache 氣流網路伺服器存取模式。

允許網際網路存取-新的 Amazon VPC 網路

如果您的組織允許在 VPC 中存取網際網路，且您希望使用者透過網際網路存取 Apache Airflow 網頁伺服器：

1. 建立可存取網際網路的 Amazon VPC 人雲端網路。
2. 為您的 Apache 氣流網頁伺服器建立具有公用網路存取模式的環境。
3. 我們建議使用 AWS CloudFormation 快速入門範本，以同時建立 Amazon VPC 基礎設施、Amazon S3 儲存貯體和 Amazon MWAA 環境。如需進一步了解，請參閱 Amazon Apache 氣流管理工作流程的快速入門教學。

如果您的組織允許虛擬私人雲端中的網際網路存取，而且您想要將 Apache Airflow Web 伺服器存取限制在 VPC 中的使用者：

1. 建立可存取網際網路的 Amazon VPC 人雲端網路。
2. 建立一個機制，從您的 VPC 存取 Apache 氣流網頁伺服器的虛擬私人雲端介面端點。
3. 為您的 Apache 氣流網頁伺服器建立具有私人網路存取模式的環境。
4. 我們推薦什麼：
    a. 我們建議您在中使用 Amazon MWAA 主控台選項一：在亞馬遜 MWAA 主控台上建立虛擬私人雲端網路，或使用中的 AWS CloudFormation 範本。選項二：建立亞馬遜虛擬私人雲端網路與互聯網接入
    b. 我們建議您使用中的 Apache 氣流網頁伺服器 AWS Client VPN 來設定存取教學課程：使用 AWS Client VPN。

不允許網際網路存取-新的 Amazon VPC 網路

如果您的組織不允許在 VPC 中存取網際網路：

1. 建立不存取網際網路的 Amazon VPC 人雲端網路。
2. 建立一個機制，從您的 VPC 存取 Apache 氣流網頁伺服器的虛擬私人雲端介面端點。
3. 為您的環境使用的每個 AWS 服務建立 VPC 端點。

4. 為您的 Apache 氣流網頁伺服器建立具有私人網路存取模式的環境。
5. 我們推薦什麼：

    a. 我們建議您使用 AWS CloudFormation 範本建立不具網際網路存取權的 Amazon VPC，以及針對 Amazon MWAA 使用的每個 AWS 服務建立 VPC 端點。選項三：建立亞馬遜虛擬私人雲端網路無互聯網接入

    b. 我們建議您使用中的 Apache 氣流網頁伺服器 AWS Client VPN 來設定存取教學課程：使用 AWS Client VPN。

不允許網際網路存取-現有的 Amazon VPC 網路

如果您的組織不允許在您的虛擬私人雲端中存取網際網路，而且您已經擁有必要的 Amazon VPC 網路，而且沒有網際網路存取：

1. 為您的環境使用的每個 AWS 服務建立 VPC 端點。
2. 為 Apache 氣流建立 VPC 端端點。
3. 建立一個機制，從您的 VPC 存取 Apache 氣流網頁伺服器的虛擬私人雲端介面端點。
4. 為您的 Apache 氣流網頁伺服器建立具有私人網路存取模式的環境。
5. 我們推薦什麼：

    a. 我們建議您建立並連接 Amazon MWAA 所使用之每項 AWS 服務所需的 VPC 端點，以及 Apache 氣流所需的 VPC 端點。使用私有路由在 Amazon VPC 中建立所需的虛擬私人雲端服務端點

    b. 我們建議您使用中的 Apache 氣流網頁伺服器 AWS Client VPN 來設定存取教學課程：使用 AWS Client VPN。

Amazon MWAA 上 VPC 的安全政策

本頁說明用於保護 Apache 氣流環境之 Amazon 受管工作流程的 Amazon VPC 元件，以及這些元件所需的組態。

內容

  - (建議使用) ACL 範例
- VPC security groups (VPC 安全群組)
  - (建議) 範例所有存取自我參照安全性群組
  - (選擇性) 限制連接埠 5432 輸入存取權的安全性群組範例
  - (選擇性) 限制連接埠 443 輸入存取權的安全性群組範例
- VPC 端點原則 (僅限私人路由)
  - (建議) 允許所有存取的 VPC 端點政策範
  - (建議使用) 允許儲存貯體存取的 Amazon S3 閘道端點政策範例

條款

公共路由

可存取網際網路的 Amazon VPC 人雲端網路。

私人路由

無法存取網際網路的 Amazon VPC 人雲端網路。

安全政策

安全群組和存取控制清單 (ACL) 提供了使用您指定的規則控制 Amazon VPC 中子網路和執行個體之間的網路流量的方法。

- 來自子網的網路流量可由存取控制清單 (ACL) 來控制。您只需要一個 ACL，並且相同的 ACL 可以在多個環境中使用。
- 進出執行個體的網路流量可由 Amazon VPC 安全群組控制。您可以在每個環境中使用一到五個安全群組。
- 進出執行個體的網路流量也可以由 VPC 端點原則控制。如果您的組織不允許在 Amazon VPC 內存取網際網路，而且您使用的是具有私有路由的 Amazon 虛擬私人雲端網路，則 VPC 端點和 Apache AirflowAWS VPC 擬私人雲端節點需要 VPC 端點政策。

網路存取控制清單 (ACL)

網路存取控制清單 (ACL) 可管理子網層級的傳入和傳出流量 (透過允許或拒絕規則) 傳入和傳出流量。ACL 是無狀態的，這意味著輸入和輸出規則必須單獨且明確地指定。它可用來指定虛擬私人雲端網路中執行個體允許進出的網路流量類型。

每個 Amazon VPC 都有預設的 ACL，可允許所有傳入和傳入和傳出流量。您可以編輯預設 ACL 規則，或建立自訂 ACL 並將其附加至您的子網路。一個子網路在任何時候都只能附加一個 ACL，但一個 ACL 可以附加到多個子網路。

(建議使用) ACL 範例

以下範例顯示可用於具有公有路由或私有路由之 Amazon VPC 的 Amazon VPC 的入站和輸出 ACL 規則。

規則編號	類型	通訊協定	連接埠範圍	來源	允許/拒絕
100	所有 IPv4 流量	全部	全部	0.0.0.0/0	允許
*	所有 IPv4 流量	全部	全部	0.0.0.0/0	拒絕

VPC security groups (VPC 安全群組)

VPC 安全群組可做為控制執行個體層級的網路流量的虛擬防火牆。安全群組是可設定狀態的，也就是說，當允許輸入連線時，就可以回覆。它用於指定 VPC 網路中執行個體允許進入的網路流量類型。

每個 Amazon VPC 都有預設安全群組。預設情況下，它沒有傳入規則。它具有允許所有傳出流量的規則。您可以編輯預設安全群組規則，或建立自訂安全群組並將其附加到 Amazon VPC。在 Amazon MWAA 上，您需要設定輸入和輸出規則，以引導 NAT 閘道上的流量。

(建議) 範例所有存取自我參照安全性群組

下列範例顯示輸入安全群組規則，該規則允許使用公有路由或私有路由的 Amazon VPC 適用於 Amazon VPC 的所有流量。此範例中的安全性群組是本身的自我參照規則。

類型	通訊協定	來源類型	來源		
所有流量	全部	全部	政治機構-0909 第 8E81919 年/my-mwaa-vpc-security-組		

下列範例顯示輸出安全性群組規則。

類型	通訊協定	來源類型	來源		
所有流量	全部	全部	0.0.0.0/0		

(選擇性) 限制連接埠 5432 輸入存取權的安全性群組範例

下列範例顯示允許您環境使用 Amazon Aurora PostgreSQL 中繼資料資料庫 (由 Amazon MWAA 擁有) 連接埠 5432 上所有 HTTPS 流量的輸入安全群組規則。

> **Note**
>
> 如果您選擇使用此規則限制流量，則需要新增另一個規則，以允許連接埠 443 上的 TCP 流量。

類型	通訊協定	連接埠範圍	Source type (來源類型)	來源	
自訂 TCP	TCP	5432	自訂	政治機構-0909 第 8E81919 年/my-mwaa-vpc-security-組	

(選擇性) 限制連接埠 443 輸入存取權的安全性群組範例

下列範例顯示允許 Apache 氣流網頁伺服器連接埠 443 上所有 TCP 流量的輸入安全性群組規則。

類型	通訊協定	連接埠範圍	Source type (來源類型)	來源	
HTTPS	TCP	443	自訂	政治機構 -0909 第 8E81919 年/ my-mwaa-vpc-security-組	

VPC 端點原則 (僅限私人路由)

VPC 端點 (AWS PrivateLink) 原則可控制您私有子網路中AWS服務的存取。VPC 端點政策是您連接至 VPC 閘道或介面端點的 IAM 資源政策。本節說明每個 VPC 端點的 VPC 端點原則所需的權限。

我們建議針對您建立的每個允許完整存取所有AWS服務的 VPC 端點使用 VPC 介面端點原則，並使用專門針對AWS權限的執行角色。

(建議) 允許所有存取的 VPC 端點政策範

下列範例顯示具有私有路由之 Amazon VPC 的 VPC 介面端點政策。

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

(建議使用) 允許儲存貯體存取的 Amazon S3 閘道端點政策範例

下方範例會說明 VPC 閘道端點政策，提供 Amazon VPC 操作所需的 Amazon S3 儲存貯體存取 Amazon ECR 操作所需的 Amazon S3 儲存貯體存取權限。除了存放 DAG 和支援檔案的儲存貯體之外，還需要擷取 Amazon ECR 映像。

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::prod-region-starport-layer-bucket/*"]
    }
  ]
}
```

在 Amazon MWAA 上管理服務特定 Amazon VPC 端點的存取

VPC 端點 (AWS PrivateLink) 可讓您將 VPC 以私密方式連線至託管於其上的服務，AWS而不需要網際網路閘道、NAT 裝置、VPN 或防火牆代理伺服器。這些端點可水平擴充且具有高可用性的虛擬裝置，可讓您在 VPC 和AWS服務中的執行個體之間進行通訊。本頁說明 Amazon MWAA 建立的 VPC 私人雲端節點，以及如果您在 Apache 氣流的 Amazon 受管工作流程上選擇私有網路存取模式，如何存取 Apache 氣流網頁伺服器的 VPC 擬私人雲端端點。

內容

- 存取 Apache 氣流網頁伺服器的 VPC 私人雲端端點 (私人網路存取)
  - 使用 AWS Client VPN
  - 使用 Linux 防禦主機
  - 使用 Load Balancer (進階)

定價

- AWS PrivateLink 定價

VPC 端點概觀

當您建立 Amazon MWAA 環境時，Amazon MWAA 會為您的環境建立一到兩個 VPC 端點之間。這些端點在您的 Amazon VPC 中顯示為具有私有 IP 的彈性網路界面 (ENI)。建立這些端點之後，傳往這些 IP 的任何流量都會以私密方式或公開路由到您的環境所使用的對應AWS服務。

公共網路存取模式

如果您為 Apache Airflow 網頁伺服器選擇公用網路存取模式，網路流量會透過網際網路公開路由。

- Amazon MWAA 為您的 Amazon Aurora PostgreSQL 中繼資料資料庫建立 VPC 界面端點。端點是在對應至私人子網路的可用區域中建立，且獨立於其他AWS帳戶。
- 然後，Amazon MWAA 會將 IP 位址從您的私有子網路繫結到介面端點。這是為了支援從 Amazon VPC 的每個可用區域繫結單一 IP 的最佳實務而設計。

私人網路存取模式

如果您為 Apache Airflow 網頁伺服器選擇私人網路存取模式，網路流量會在您的 Amazon VPC 中以私密方式路由。

- Amazon MWAA 為您的 Apache 氣流網頁伺服器建立 VPC 界面端點，並為您的 Amazon Aurora PostgreSQL 中繼資料資料庫建立一個介面端點。端點是在對應至私人子網路的可用區域中建立，且獨立於其他AWS帳戶。
- 然後，Amazon MWAA 會將 IP 位址從您的私有子網路繫結到介面端點。這是為了支援從 Amazon VPC 的每個可用區域繫結單一 IP 的最佳實務而設計。

使用其他AWS服務的許可

介面端點會使用 AWS Identity and Access Management (IAM) 中環境的執行角色來管理環境所使用AWS資源的權限。隨著為環境啟用更多AWS服務，每個服務都會要求您使用環境的執行角色來設定權限。若要新增權限，請參閱Amazon MWAA 執行角色。

如果您已為 Apache Airflow Web 伺服器選擇私人網路存取模式，您也必須在每個端點的 VPC 端點原則中允許權限。如需進一步了解，請參閱 the section called “VPC 端點原則 (僅限私人路由)”。

檢視 VPC 端點

本節說明如何檢視 Amazon MWAA 建立的虛擬私人雲端端點，以及如何識別 Apache 氣流 VPC 端點的私有 IP 地址。

在 Amazon VPC 主控台上檢視 VPC 端點

以下部分顯示了檢視 Amazon MWAA 建立的 VPC 端點的步驟，以及如果您在 Amazon VPC 使用私有路由，則可能已建立的任何 VPC 端點。

若要檢視 VPC 端點

1. 在 Amazon VPC 主控台上開啟「端點」頁面。
2. 使用「AWS地區」選取器選取您的地區。
3. 您應該會看到 Amazon MWAA 建立的 VPC 界面端點，以及如果您在 Amazon VPC 中使用私有路由，可能已建立的任何 VPC 端點。

若要進一步了解具有私有路由的 Amazon VPC 所需的 VPC 服務端點，請參閱。使用私有路由在 Amazon VPC 中建立所需的虛擬私人雲端服務端點

識別您的 Apache 氣流網頁伺服器及其虛擬私人雲端端點的私有 IP 位址

下列步驟說明如何擷取 Apache Airflow 網頁伺服器及其虛擬私人雲端介面端點的主機名稱，以及其私人 IP 位址。

1. 使用下列 AWS Command Line Interface (AWS CLI) 命令擷取 Apache 氣流網頁伺服器的主機名稱。

```
aws mwaa get-environment --name YOUR_ENVIRONMENT_NAME --query
 'Environment.WebserverUrl'
```

您應該會看到類似下列回應的內容：

```
"99aa99aa-55aa-44a1-a91f-f4552cf4e2f5-vpce.c10.us-west-2.airflow.amazonaws.com"
```

2. 對上一個命令回應中傳回的主機名稱執行 dig 命令。例如：

```
dig CNAME +short 99aa99aa-55aa-44a1-a91f-f4552cf4e2f5-vpce.c10.us-
west-2.airflow.amazonaws.com
```

您應該會看到類似下列回應的內容：

```
vpce-0699aa333a0a0a0-bf90xjtr.vpce-svc-00bb7c2ca2213bc37.us-
west-2.vpce.amazonaws.com.
```

3. 使用下列 AWS Command Line Interface (AWS CLI) 命令擷取上一個命令回應中傳回的 VPC 端點 DNS 名稱。例如：

```
aws ec2 describe-vpc-endpoints | grep vpce-0699aa333a0a0a0-bf90xjtr.vpce-
svc-00bb7c2ca2213bc37.us-west-2.vpce.amazonaws.com.
```

您應該會看到類似下列回應的內容：

```
"DnsName": "vpce-066777a0a0a0-bf90xjtr.vpce-svc-00bb7c2ca2213bc37.us-
west-2.vpce.amazonaws.com",
```

4. 在您的 Apache 氣流主機名稱及其虛擬私人雲端端點 DNS 名稱上執行 nslookup 或挖掘命令，以擷取 IP 位址。例如：

```
dig +short YOUR_AIRFLOW_HOST_NAME YOUR_AIRFLOW_VPC_ENDPOINT_DNS
```

您應該會看到類似下列回應的內容：

```
10.199.11.111
  10.999.11.33
```

存取 Apache 氣流網頁伺服器的 VPC 私人雲端端點 (私人網路存取)

如果您已為 Apache 氣流網頁伺服器選擇私人網路存取模式，則需要建立一個機制來存取 Apache 氣流網頁伺服器的 VPC 私人雲端介面端點。對於這些資源，您必須使用與 Amazon MWAA 環境相同的 Amazon VPC、VPC 安全群組和私有子網路。

使用 AWS Client VPN

AWS Client VPN 是以用戶端為基礎的受管 VPN 服務，能讓您安全地存取您的 AWS 資源，以及您的現場部署網路中的資源。它提供了一個安全的 TLS 連接從任何位置使用 OpenVPN 客戶端。

我們建議您遵循 Amazon MWAA 教學課程來設定 Client VPN：。教學課程：使用AWS Client VPN

使用 Linux 防禦主機

防禦主機是一種伺服器，其目的是提供從外部網路 (例如透過電腦透過網際網路) 存取私人網路的伺服器。Linux 執行個體位於公有子網路中，並使用安全群組進行設定，該群組允許從連接至執行防禦主機的基礎 Amazon EC2 執行個體的安全群組進行 SSH 存取。

我們建議您按照 Amazon MWAA 教學課程來設定 Linux 防禦主機：教學課程：使用 Linux 防禦主機設定私人網路存取

使用 Load Balancer (進階)

以下部分顯示套用至應用 Ap plication Load Balancer 所需的組態。

1. 目標群組。您必須使用指向 Apache Airflow 網頁伺服器及其虛擬私人雲端介面端點之私有 IP 位址的目標群組。我們建議將兩個私有 IP 位址指定為註冊目標，因為只使用一個會降低可用性。如需如何識別私人 IP 位址的詳細資訊，請參閱the section called “識別您的 Apache 氣流網頁伺服器及其虛擬私人雲端端點的私有 IP 位址”。
2. 狀態碼。建議您在目標群組設定中使用200和302狀態碼。否則，如果 Apache Airflow Web 伺服器的虛擬私人雲端端點回應錯誤，則目標可能會被標記為狀況不良。`302 Redirect`
3. 監聽程式。您需要指定 Apache 氣流網頁伺服器的目標連接埠。例如：

通訊協定	連線埠
HTTPS	443

4. ACM 新網域。如果您想要在中建立 SSL/TLS 憑證的關聯AWS Certificate Manager，則必須為負載平衡器的 HTTPS 接聽程式建立新網域。

5. ACM 憑證區域。如果您想要在中建立 SSL/TLS 憑證的關聯AWS Certificate Manager，則需要上傳到與環境相同的AWS區域。例如：

   - Example 上傳憑證的區域

     ```
     aws acm import-certificate --certificate fileb://Certificate.pem --certificate-chain fileb://CertificateChain.pem --private-key fileb://PrivateKey.pem --region us-west-2
     ```

使用私有路由在 Amazon VPC 中建立所需的虛擬私人雲端服務端點

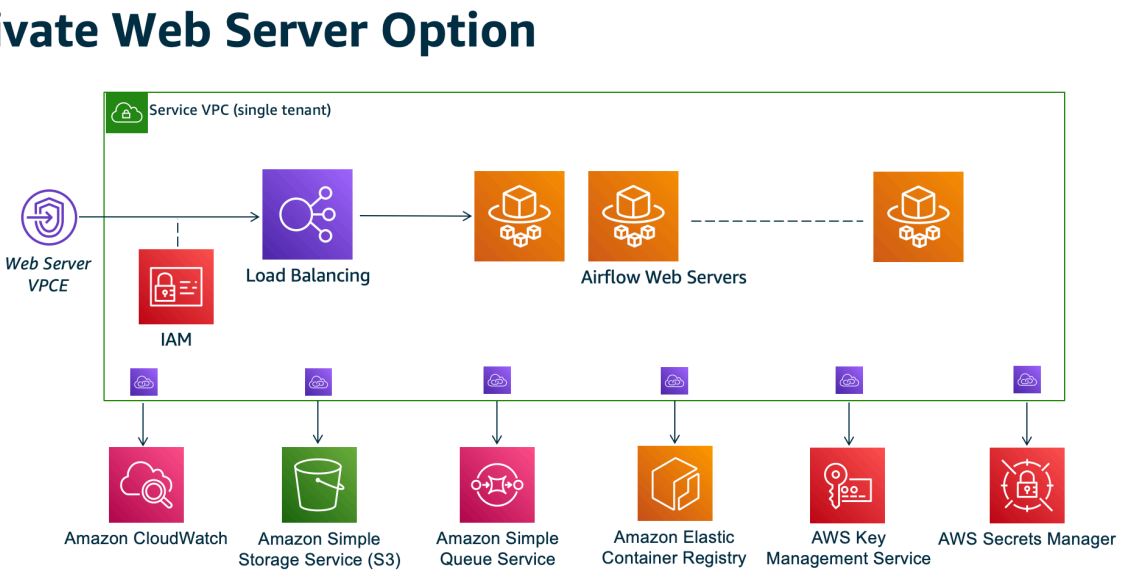
沒有網際網路存取的現有 Amazon VPC 人雲端網路需要額外的 VPC 人雲端服務端點 (AWS PrivateLink)，才能在亞馬遜受管工作流程上使用 Apache 氣流進行 本頁說明 Amazon MWAA 所使用之AWS服務所需的虛擬私人雲端端點、Apache 氣流所需的虛擬私人雲端端點，以及如何使用私有路由建立 VPC 端點並將其連接到現有的 Amazon VPC。

內容

定價

- AWS PrivateLink 定價

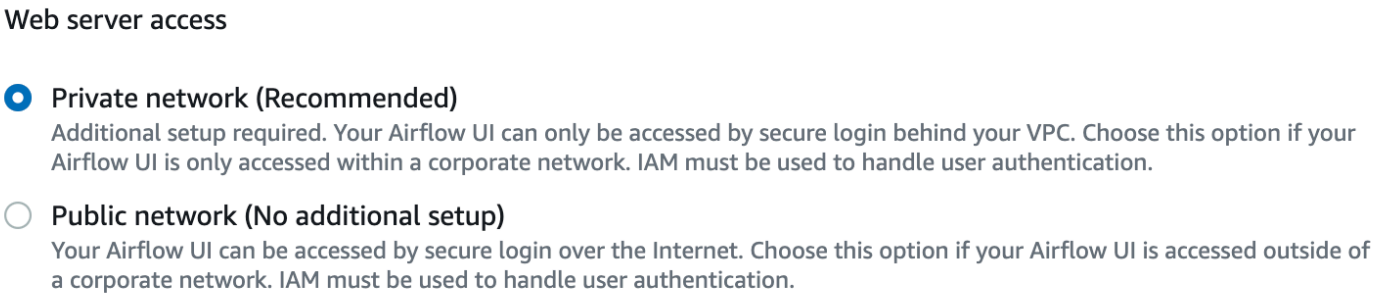
私人網路和私人路由



私有網路存取模式可將 Apache Airflow UI 的存取權限限制為 Amazon VPC 中已授予您環境 IAM 政策存取權的使用者。

當您創建具有私有 Web 服務器訪問權限的環境時，必須將所有依賴項打包到 Python wheel 存檔（`.whl`）中，然後`.whl`在`requirements.txt`. 有關使用 wheel 打包和安裝依賴項的說明，請參閱使用 Python wheel 管理依賴關係。

下圖顯示 Amazon MWAA 主控台上哪裡可以找到私人網路選項。



- 私人路由。沒有網際網路存取權的 Amazon VPC 會限制 VPC 內的網路流量。此頁面假設您的 Amazon VPC 沒有網際網路存取權限，且您的環境所使用的每個AWS服務都需要 VPC 端點，以及 Amazon MWAA 環境位於相同AWS區域和 Amazon VPC 中的 Apache 氣流的 VPC 端點。

(必要) VPC 端點

以下部分顯示無法存取網際網路的 Amazon VPC 所需的必要 VPC 端點。它會列出 Amazon MWAA 所使用之每個AWS服務的虛擬私人雲端節點，包括 Apache 氣流所需的 VPC 端點。

```
com.amazonaws.YOUR_REGION.s3
com.amazonaws.YOUR_REGION.monitoring
com.amazonaws.YOUR_REGION.ecr.dkr
com.amazonaws.YOUR_REGION.ecr.api
com.amazonaws.YOUR_REGION.logs
com.amazonaws.YOUR_REGION.sqs
com.amazonaws.YOUR_REGION.kms
com.amazonaws.YOUR_REGION.airflow.api
com.amazonaws.YOUR_REGION.airflow.env
com.amazonaws.YOUR_REGION.airflow.ops
```

附加必要的 VPC 端點

本節說明使用私有路由連接 Amazon VPC 所需的 VPC 端點的步驟。

AWS服務所需的 VPC 端點

以下部分說明將環境使用之AWS服務的 VPC 端點連接到現有 Amazon VPC 的步驟。

將 VPC 端點附加到私有子網路

1. 開啟 Amazon VPC 主控台上的端點頁面。
2. 使用「AWS地區」選取器選取您的地區。
3. 為 Amazon S3 建立端點：
   a. 選擇 Create Endpoint (建立端點)。
   b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.s3**，輸入:，然後按鍵盤上的 Enter。
   c. 建議您選擇針對閘道類型列出的服務端點。

      例如：**com.amazonaws.us-west-2.s3 amazon Gateway**
   d. 在 VPC 中選擇您環境的亞馬遜 VPC。
   e. 確定已選取位於不同可用區域中的兩個私人子網路，並透過選取啟用 DNS 名稱啟用該私人 DNS。
   f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

4. 為亞馬遜 ECR 建立第一個端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.ecr.dkr**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

5. 為亞馬遜 ECR 建立第二個端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.ecr.api**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

6. 建立記CloudWatch錄檔的端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.logs**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

7. 建立用於CloudWatch監控的端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.monitoring**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

8. 為 Amazon SQS 建立端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.sqs**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

9. 為以下項目建立端點AWS KMS：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.kms**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

Apache Airflow 所需 VPC 端點

下節說明將 Apache 氣流的虛擬私人雲端端點連接到現有的 Amazon VPC 的步驟。

將 VPC 端點附加到私有子網路

1. 開啟 Amazon VPC 主控台上的端點頁面。
2. 使用「AWS地區」選取器選取您的地區。
3. 為 Apache 氣流 API 建立端點：
   a. 選擇 Create Endpoint (建立端點)。
   b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.airflow.api**，輸入:，然後按鍵盤上的 Enter。
   c. 選取服務端點。
   d. 在 VPC 中選擇您環境的亞馬遜 VPC。
   e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。
   f. 選擇您環境的 Amazon VPC 安全群組。
   g. 在策略中選擇「完整存取」。
   h. 選擇 Create endpoint (建立端點)。
4. 為 Apache 氣流環境建立第一個端點：
   a. 選擇 Create Endpoint (建立端點)。
   b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.airflow.env**，輸入:，然後按鍵盤上的 Enter。
   c. 選取服務端點。
   d. 在 VPC 中選擇您環境的亞馬遜 VPC。
   e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。
   f. 選擇您環境的 Amazon VPC 安全群組。
   g. 在策略中選擇「完整存取」。
   h. 選擇 Create endpoint (建立端點)。
5. 為 Apache 氣流作業建立第二個端點：

a. 選擇 Create Endpoint (建立端點)。

b. 在「依屬性篩選」或「依關鍵字搜尋」文字欄位中**.airflow.ops**，輸入:，然後按鍵盤上的 Enter。

c. 選取服務端點。

d. 在 VPC 中選擇您環境的亞馬遜 VPC。

e. 確定已選取位於不同可用區域中的兩個私人子網路，並已啟用「啟用 DNS 名稱」。

f. 選擇您環境的 Amazon VPC 安全群組。

g. 在策略中選擇「完整存取」。

h. 選擇 Create endpoint (建立端點)。

(選擇性) 為您的 Amazon S3 虛擬私人雲端介面端點啟用私有 IP 地址

Amazon S3 界面端點不支援私有 DNS。S3 端點請求仍會解析為公用 IP 位址。若要將 S3 位址解析為私有 IP 位址，您需要在 Route 53 中為 S3 區域端點新增私有託管區域。

使用 Route 53

本節說明使用 Route 53 為 S3 介面端點啟用私有 IP 位址的步驟。

1. 為 Amazon S3 VPC 介面端點 (例如 s3.eu-west-1.amazonaws.com) 建立私有託管區域，並將其與 Amazon VPC 建立關聯。
2. 為 Amazon S3 VPC 介面端點端點 (例如 s3.eu-west-1.amazonaws.com) 建立 Alias As3.eu-west-1.amazonaws.com.
3. 為您的 Amazon S3 介面端點建立別名萬用字元記錄 (例如，*.s3.eu-西部 1.amazonaws.com)，以解析為 VPC 界面端點 DNS 名稱。

VPC 具有自訂 DNS

如果您的 Amazon VPC 使用自訂 DNS 路由，則需要透過建立 CNAME 記錄在 DNS 解析器中進行變更 (不是 Route 53，通常是執行 DNS 伺服器的 EC2 執行個體)。例如：

```
Name: s3.us-west-2.amazonaws.com
Type: CNAME
Value:  *.vpce-0f67d23e37648915c-e2q2e2j3.s3.eu-west-1.vpce.amazonaws.com
```

在 Amazon MWAA 上管理您自己的 Amazon VPC 端點

Amazon MWAA 使用 Amazon VPC 端點與設定 Apache 氣流環境所需的各種AWS服務整合。管理您自己的端點有兩個主要使用案例：

1. 這表示當您使用AWS Organizations來管理多個AWS帳戶和共用資源時，您可以在共用的 Amazon VPC 中建立 Apache 氣流環境。
2. 它可讓您透過將權限縮小到使用端點的特定資源，以使用限制性更高的存取政策。

如果您選擇管理自己的 VPC 端點，則必須負責為環境 RDS for PostgreSQL 資料庫和環境網頁伺服器建立自己的端點。

如需 Amazon MWAA 如何在雲端部署 Apache 氣流的詳細資訊，請參閱 Amazon M WAA 架構圖表。

在共用的 Amazon VPC 中建立環境

如果您使用管理共用資源的多個AWS帳戶，可以AWS Organizations將客戶受管的 VPC 端點與 Amazon MWAA 搭配使用，與組織中的其他帳戶共用環境資源。

設定共用 VPC 存取時，擁有主要 Amazon VPC (擁有者) 的帳戶會與屬於同一組織的其他帳戶 (參與者) 共用 Amazon MWAA 所需的兩個私有子網路。共用這些子網路的參與者帳戶可以檢視、建立、修改和刪除共用 Amazon VPC 中的環境。

假設您擁有一個帳戶 (與組織中的`Root`帳戶一樣)`0wner`，並擁有 Amazon VPC 資源，以及一個參與者帳戶 (同一組織的成員)。`Participant`在與之共用的 Amazon VPC 中建`Participant`立新的 Amazon MWAA 時`0wner`，Amazon MWAA 會先建立服務 VPC 資源，然後進入最多 72 小時的`PENDING`狀態。

環境狀態從變更`CREATING`為後`PENDING`，代表作業的主參與者`0wner`會建立所需的端點。為此，Amazon MWAA 會在 Amazon MWAA 主控台中列出資料庫和網路伺服器端點。您也可以呼叫`GetEnvironment`API 動作來取得服務端點。

> **Note**
>
> 如果您用來共用資源的 Amazon VPC 是私有的 Amazon VPC，您仍然必須完成中所述的步驟。the section called “管理 VPC 端點的存取”本主題涵蓋設定一組與其他AWS整合的AWS服務相關的不同 Amazon VPC 端點，例如 Amazon ECR、Amazon ECS 和 Amazon SQS。這些服務對於在雲端中操作和管理 Apache Airflow 環境至關重要。

必要條件

在共用 VPC 中建立 Amazon MWAA 環境之前，您需要下列資源：

- 一個AWS帳0wner戶，用作擁有 Amazon VPC 的帳戶。
- 以根*MyOrganization*建立的AWS Organizations組織單位。
- 第二個AWS帳戶Participant，在MyOrganization為建立新環境的參與者帳戶提供服務。

此外，我們建議您熟悉擁有者和參與者在 Amazon VPC 中共用資源時的責任和許可。

創建 Amazon VPC

首先，建立擁有者和參與者帳戶將共用的新 Amazon VPC：

1. 使用登入主控台0wner，然後開啟主AWS CloudFormation控台。使用下列範本建立堆疊。此堆疊佈建許多網路資源，包括 Amazon VPC，以及兩個帳戶在此案例中將共用的子網路。

```
AWSTemplateFormatVersion: "2010-09-09"
Description: >-
  This template deploys a VPC, with a pair of public and private subnets spread
  across two Availability Zones. It deploys an internet gateway, with a default
  route on the public subnets. It deploys a pair of NAT gateways (one in each
  AZ), and default routes for them in the private subnets.
Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to resource names
    Type: String
    Default: mwaa-
  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16
  PublicSubnet1CIDR:
    Description: >-
      Please enter the IP range (CIDR notation) for the public subnet in the
      first Availability Zone
    Type: String
    Default: 10.192.10.0/24
  PublicSubnet2CIDR:
    Description: >-
      Please enter the IP range (CIDR notation) for the public subnet in the
      second Availability Zone
```

```
    Type: String
    Default: 10.192.11.0/24
  PrivateSubnet1CIDR:
    Description: >-
      Please enter the IP range (CIDR notation) for the private subnet in the
      first Availability Zone
    Type: String
    Default: 10.192.20.0/24
  PrivateSubnet2CIDR:
    Description: >-
      Please enter the IP range (CIDR notation) for the private subnet in the
      second Availability Zone
    Type: String
    Default: 10.192.21.0/24
Resources:
  VPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName
  InternetGateway:
    Type: 'AWS::EC2::InternetGateway'
    Properties:
      Tags:
        - Key: Name
          Value: !Ref EnvironmentName
  InternetGatewayAttachment:
    Type: 'AWS::EC2::VPCGatewayAttachment'
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
  PublicSubnet1:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - 0
        - !GetAZs ''
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
```

```
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Public Subnet (AZ1)'
  PublicSubnet2:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - 1
        - !GetAZs ''
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Public Subnet (AZ2)'
  PrivateSubnet1:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - 0
        - !GetAZs ''
      CidrBlock: !Ref PrivateSubnet1CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Private Subnet (AZ1)'
  PrivateSubnet2:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      AvailabilityZone: !Select
        - 1
        - !GetAZs ''
      CidrBlock: !Ref PrivateSubnet2CIDR
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Private Subnet (AZ2)'
  NatGateway1EIP:
    Type: 'AWS::EC2::EIP'
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc
```

```
  NatGateway2EIP:
    Type: 'AWS::EC2::EIP'
    DependsOn: InternetGatewayAttachment
    Properties:
      Domain: vpc
  NatGateway1:
    Type: 'AWS::EC2::NatGateway'
    Properties:
      AllocationId: !GetAtt NatGateway1EIP.AllocationId
      SubnetId: !Ref PublicSubnet1
  NatGateway2:
    Type: 'AWS::EC2::NatGateway'
    Properties:
      AllocationId: !GetAtt NatGateway2EIP.AllocationId
      SubnetId: !Ref PublicSubnet2
  PublicRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Public Routes'
  DefaultPublicRoute:
    Type: 'AWS::EC2::Route'
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway
  PublicSubnet1RouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet1
  PublicSubnet2RouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnet2
  PrivateRouteTable1:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref VPC
      Tags:
```

```
        - Key: Name
          Value: !Sub '${EnvironmentName} Private Routes (AZ1)'
  DefaultPrivateRoute1:
    Type: 'AWS::EC2::Route'
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway1
  PrivateSubnet1RouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable1
      SubnetId: !Ref PrivateSubnet1
  PrivateRouteTable2:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref VPC
      Tags:
        - Key: Name
          Value: !Sub '${EnvironmentName} Private Routes (AZ2)'
  DefaultPrivateRoute2:
    Type: 'AWS::EC2::Route'
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway2
  PrivateSubnet2RouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable2
      SubnetId: !Ref PrivateSubnet2
  SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupName: mwaa-security-group
      GroupDescription: Security group with a self-referencing inbound rule.
      VpcId: !Ref VPC
  SecurityGroupIngress:
    Type: 'AWS::EC2::SecurityGroupIngress'
    Properties:
      GroupId: !Ref SecurityGroup
      IpProtocol: '-1'
      SourceSecurityGroupId: !Ref SecurityGroup
Outputs:
```

```
VPC:
  Description: A reference to the created VPC
  Value: !Ref VPC
PublicSubnets:
  Description: A list of the public subnets
  Value: !Join
    - ','
    - - !Ref PublicSubnet1
      - !Ref PublicSubnet2
PrivateSubnets:
  Description: A list of the private subnets
  Value: !Join
    - ','
    - - !Ref PrivateSubnet1
      - !Ref PrivateSubnet2
PublicSubnet1:
  Description: A reference to the public subnet in the 1st Availability Zone
  Value: !Ref PublicSubnet1
PublicSubnet2:
  Description: A reference to the public subnet in the 2nd Availability Zone
  Value: !Ref PublicSubnet2
PrivateSubnet1:
  Description: A reference to the private subnet in the 1st Availability Zone
  Value: !Ref PrivateSubnet1
PrivateSubnet2:
  Description: A reference to the private subnet in the 2nd Availability Zone
  Value: !Ref PrivateSubnet2
SecurityGroupIngress:
  Description: Security group with self-referencing inbound rule
  Value: !Ref SecurityGroupIngress
```

2. 佈建新的 Amazon VPC 資源後，導覽至AWS Resource Access Manager主控台，然後選擇建立資源共用。
3. 從可共用的可用子網路清單中選擇您在第一個步驟中建立的子網路。`Participant`

建立 環境

完成下列步驟，以使用客戶管理的 Amazon VPC 端點建立 Amazon MWAA 環境。

1. 使用登入`Participant`，然後開啟 Amazon MWAA 主控台。完成步驟 1：指定詳細資訊以指定新環境的 Amazon S3 儲存貯體、DAG 資料夾和相依性。如需詳細資訊，請參閱入門。
2. 在「設定進階設定」 頁面的「聯網」下，從共用的 Amazon VPC 中選擇子網路。

3. 在端點管理下，從下拉式清單中選擇「客戶」。
4. 保留頁面上其餘選項的預設值，然後在「複查並建立」頁面上選擇建立環境。

環境從CREATING狀態開始，然後變更為PENDING。當環境為時PENDING，請使用主控台寫下 Database 端點服務名稱和 Web 伺服器端點服務名稱（如果您設定私人 Web 伺服器）。

當您使用 Amazon MWAA 主控台建立新環境時。Amazon MWAA 會建立具有所需輸入和輸出規則的新安全群組。記下該安全群組 ID。

在下一節中，0wner將使用服務端點和安全群組 ID 在共用的 Amazon VPC 中建立新的 Amazon VPC 端點。

建立 Amazon VPC 端點

完成以下步驟，為您的環境建立所需的 Amazon VPC 端點。

1. 登錄到使AWS Management Console用0wner，打開 https://console.aws.amazon.com/vpc/。
2. 從左側導覽面板中選擇安全群組，然後使用下列輸入和輸出規則在共用 Amazon VPC 中建立新的安全群組：

	Type	通訊協定	來源類型	來源
傳入	所有流量	全部	全部	您的環境安全性群組
傳出	所有流量	全部	全部	0.0.0.0/0

> **Warning**
>
> 該0wner帳戶必須在0wner帳戶中設定安全群組，以允許從新環境傳輸到共用 Amazon VPC 的流量。您可以在中建立新的安全性群組0wner，或編輯現有的安全性群組來執行此操作。

3. 選擇「端點」，然後使用先前步驟的端點服務名稱為環境資料庫和 Web 伺服器（如果處於私人模式）建立新端點。選擇共用的 Amazon VPC、您用於環境的子網路，以及環境的安全群組。

如果成功，環境就會從PENDING回到CREATING，然後最後變為AVAILABLE。如果是AVAILABLE，您可以登入 Apache 氣流主控台。

共用 Amazon VPC 疑難排解

使用下列參考來解決在共用 Amazon VPC 中建立環境時遇到的問題。

環境CREATE_FAILED狀PENDING態

- 驗證Owner正在與使用共Participant用AWS Resource Access Manager子網路。
- 確認資料庫和 Web 伺服器的 Amazon VPC 端點是在與環境關聯的相同子網路中建立。
- 確認與您的端點搭配使用的安全群組是否允許來自環境所使用之安全群組的流量。Owner帳戶會建立將安全性群組參照Participant為*account-number*/*security-group-id*：的規則。

Type	通訊協定	來源類型	來源
所有流量	全部	全部	*12345 6789012/ SG-0909e8E 81919*

如需詳細資訊，請參閱擁有者與參與者的責任與權限

環境卡在PENDING狀態

驗證每個 VPC 端點狀態，以確保其狀態為Available。如果您使用私人 Web 伺服器設定環境，則還必須為 Web 伺服器建立端點。如果環境卡在中PENDING，這可能表示私人 Web 伺服器端點遺失。

收到The Vpc Endpoint Service '*vpce-service-name*' does not exist錯誤

如果看到下列錯誤，請確認帳戶是在擁有共用 VPC 的Owner帳戶中建立端點：

```
ClientError: An error occurred (InvalidServiceName) when calling the
 CreateVpcEndpoint operation:

The Vpc Endpoint Service 'vpce-service-name' does not exist
```

適用於 Apache 氣流的 Amazon 受管工作流程

本指南包含使用和設定適用於 Apache 氣流環境之 Amazon 受管工作流程的 step-by-step 教學課程。

主題

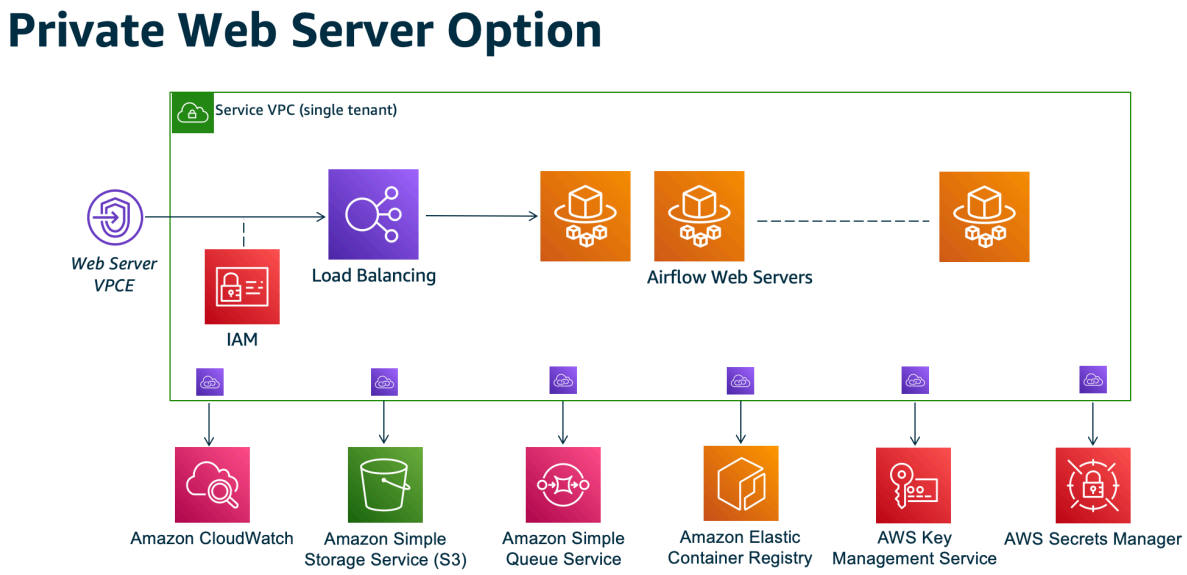
教學課程：使用AWS Client VPN

本教學將逐步引導您完成從電腦建立 VPN 通道到 Apache 氣流網頁伺服器的步驟，以適用於 Apache 氣流環境的亞馬遜管理工作流程。若要透過 VPN 通道連線到網際網路，您首先需要建立AWS Client VPN端點。設定完成後，Client VPN 端點就像 VPN 伺服器一樣，允許從您的電腦與 VPC 中的資源進行安全連線。然後，您將使用桌面版從電腦連線到 Client VPN。AWS Client VPN

章節

私有網路

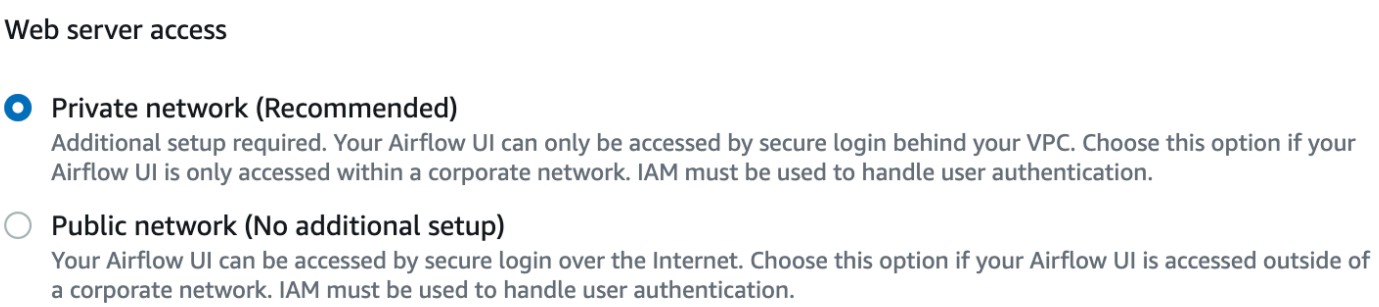
本教學課程假設您已為 Apache Airflow 網頁伺服器選擇私人網路存取模式。



私有網路存取模式可將 Apache Airflow UI 的存取權限限制為 Amazon VPC 中已授予您環境 IAM 政策存取權的使用者。

當您創建具有私有 Web 服務器訪問權限的環境時，必須將所有依賴項打包到 Python wheel 存檔（.whl）中，然後.whl在requirements.txt. 有關使用 wheel 打包和安裝依賴項的說明，請參閱使用 Python wheel 管理依賴關係。

下圖顯示 Amazon MWAA 主控台上哪裡可以找到私人網路選項。



使用案例

您可以在建立 Amazon MWAA 環境之前或之後使用本教學課程。您必須使用與環境相同的 Amazon VPC、VPC 安全群組和私有子網路。如果您在建立 Amazon MWAA 環境後使用本教學課程，在完成這

些步驟後，您可以返回 Amazon MWAA 主控台，並將 Apache 氣流網頁伺服器存取模式變更為私人網路。

開始之前

1. 檢查使用者權限。請確定您在AWS Identity and Access Management (IAM) 中的帳戶具有足夠的許可來建立和管理 VPC 資源。
2. 使用您的亞馬遜 MWAA VPC。本教學假設您將 Client VPN 與現有 VPC 產生關聯到現有 VPC。Amazon VPC 必須與 Amazon MWAA 環境位於相同的AWS區域，並且具有兩個私有子網路。如果您尚未建立 Amazon VPC，請使用中的AWS CloudFormation範本選項三：建立亞馬遜虛擬私人雲端網路無互聯網接入。

目標

在本教學中，您將執行下列作業：

1. 使用現有 Amazon VPC 的AWS CloudFormation範本建立AWS Client VPN端點。
2. 產生伺服器和用戶端憑證和金鑰，然後將伺服器憑證和金鑰上傳到AWS Certificate Manager Amazon MWAA 環境相同AWS區域。
3. 下載並修改 Client VPN 的 Client VPN 端點設定檔，並使用該檔案建立 VPN 設定檔，以便使用桌面 Client VPN 進行連線。

(選擇性) 步驟一：識別您的 VPC、CIDR 規則和虛擬私人雲端安全性

以下部分說明如何尋找 Amazon VPC、VPC 安全群組的 ID，以及識別在後續步驟中建立 Client VPN 所需的 CIDR 規則的方法。

識別您的 CIDR 規則

以下部分說明如何識別 CIDR 規則，您需要建立 Client VPN。

若要識別 Client VPN 的 CIDR

1. 在亞馬遜 VPC 主控台上開啟您的亞馬遜 VPC 頁面。
2. 使用導覽列中的區域選擇器，選擇與 Amazon MWAA 環境相同的AWS區域。
3. 選擇您的 Amazon VPC。

4. 假設您的私有子網路的 CIDR 是：

   - 私有子網路 1：/24
   - 私有子網路 2：/24

   如果您的亞馬遜 VPC 的 CIDR 是 10.192.0.0/16，那麼您為客戶端 VPN 指定的用戶端 IPv4 CIDR 將是 10.192.0.0/22。
5. 儲存此 CIDR 值，以及 VPC ID 的值，以供後續步驟使用。

識別您的 VPC 和安全群組

以下部分說明如何尋找 Amazon VPC 和安全群組的 ID，以及建立 Client VPN 所需的 ID。

> **Note**
> 您可能正在使用多個安全群組。您需要在後續步驟中指定所有 VPC 的安全群組。

若要識別安全性群組

1. 在 Amazon VPC 主控台上開啟「安全群組」頁面。
2. 使用導覽列中的區域選擇器選擇AWS區域。
3. 在 VPC ID 中尋找 Amazon VPC，並識別與虛擬私人 VPC 相關聯的安全群組。
4. 儲存安全群組和 VPC 的 ID，以便後續步驟使用。

步驟 2：建立伺服器和用戶端憑證

Client VPN 端點僅支援 1024 位元和 2048 位元 RSA 金鑰大小。下節介紹如何使用 OpenVPN 產生伺服器和用戶端憑證及金鑰，然後使用AWS Command Line Interface (AWS CLI) 將憑證上傳到 ACM。

建立用戶端憑證

1. 依照下列快速步驟，透過 [用戶端驗證與授權：相互驗證]AWS CLI 中的 [建立憑證並上傳至 ACM]。
2. 在這些步驟中，您必須在上傳伺服器和用戶端憑證時，在AWS CLI命令中指定與 Amazon MWAA 環境相同的AWS區域。以下是如何在這些指令中指定區域的範例：

a. Example 伺服器憑證的區域

```
aws acm import-certificate --certificate fileb://server.crt --private-key
 fileb://server.key --certificate-chain fileb://ca.crt --region us-west-2
```

b. Example 用戶端憑證的區域

```
aws acm import-certificate --certificate fileb://client1.domain.tld.crt
 --private-key fileb://client1.domain.tld.key --certificate-chain fileb://
ca.crt --region us-west-2
```

c. 完成這些步驟後，請儲存在回AWS CLI應中針對伺服器憑證和用戶端憑證 ARN 所傳回的值。您將在AWS CloudFormation範本中指定這些 ARN 以建立 Client VPN。

3. 在這些步驟中，用戶端憑證和私密金鑰會儲存到您的電腦。以下是在哪裡找到這些憑證的範例：

a. Example macOS

在 macOS 上，內容會儲存在/Users/*youruser*/custom_folder。如果您列出此目錄的所有 (ls -a) 內容，您應該會看到類似下列內容的內容：

```
.
..
ca.crt
client1.domain.tld.crt
client1.domain.tld.key
server.crt
server.key
```

b. 完成這些步驟後，請儲存內容或記下用戶端憑證的位置client1.domain.tld.crt，以及私密金鑰client1.domain.tld.key。您將會將這些值新增至 Client VPN 的設定檔中。

步驟三：在本機儲存AWS CloudFormation範本

以下部分包含用於創建 Client VPN 的AWS CloudFormation模板。您必須指定與 Amazon MWAA 環境相同的 Amazon VPC、VPC 安全群組和私有子網路。

- 複製下列範本機內容並在本機儲存為mwaa_vpn_client.yaml。您也可以下載模板。

  以下列值取代：

- **YOUR_CLIENT_ROOT_CERTIFICATE_ARN**— ARN 為您的用戶端 1. 網域 .tld 憑證中ClientRootCertificateChainArn。
- **YOUR_SERVER_CERTIFICATE_ARN**— 中伺服器憑證的 ARNServerCertificateArn。
- 中的用戶端 IPv4 CIDR 規則ClientCidrBlock。提供的10.192.0.0/22 CIDR 規則。
- 您的亞馬遜 VPC ID 在VpcId. 本發明提供一種的 VPC 人vpc-010101010101雲端。
- 您在中的 VPC 安全群組識別碼SecurityGroupIds。提供的安全sg-0101010101性群組。

```
AWSTemplateFormatVersion: 2010-09-09
Description: This template deploys a VPN Client Endpoint.
Resources:
  ClientVpnEndpoint:
    Type: 'AWS::EC2::ClientVpnEndpoint'
    Properties:
      AuthenticationOptions:
        - Type: "certificate-authentication"
          MutualAuthentication:
            ClientRootCertificateChainArn: "YOUR_CLIENT_ROOT_CERTIFICATE_ARN"
      ClientCidrBlock: 10.192.0.0/22
      ClientConnectOptions:
        Enabled: false
      ConnectionLogOptions:
        Enabled: false
      Description: "MWAA Client VPN"
      DnsServers: []
      SecurityGroupIds:
        - sg-0101010101
      SelfServicePortal: ''
      ServerCertificateArn: "YOUR_SERVER_CERTIFICATE_ARN"
      SplitTunnel: true
      TagSpecifications:
        - ResourceType: "client-vpn-endpoint"
          Tags:
          - Key: Name
            Value: MWAA-Client-VPN
      TransportProtocol: udp
      VpcId: vpc-010101010101
      VpnPort: 443
```

> **Note**
>
> 如果您的環境使用多個安全性群組，您可以使用下列格式指定多個安全性群組：
>
> ```
> SecurityGroupIds:
>       - sg-0112233445566778b
>       - sg-0223344556677889f
> ```

步驟四：建立 Client VPNAWS CloudFormation 堆疊

建立 AWS Client VPN

1. 開啟 AWS CloudFormation 主控台。
2. 選擇模板已準備就緒，上傳模板文件。
3. 選擇 [選擇檔案]，然後選取您的`mwaa_vpn_client.yaml`檔案。
4.
5. 選擇下一步，下一步。
6. 選取確認，然後選擇 [建立堆疊]。

步驟五：將子網路與 Client VPN 建立關聯

若要將私人子網路關聯至AWS Client VPN

1. 打開亞馬遜 VPC 控制台。
2. 選擇「Client VPN 端點」頁面。
3. 選取您的 Client VPN，然後選擇 [關聯] 索引標籤 [關聯]。
4. 在下拉式清單中選擇下列項目：
   - 您在 VPC 中的亞馬遜 VPC。
   - 您在 [選擇要關聯的子網路] 中的其中一個私人子網路。
5. 選擇 Associate (關聯)。

> **Note**
>
> 將 VPC 和子網路與 Client VPN 相關聯到 Client VPN。

步驟六：將授權輸入規則新增至 Client VPN

您需要使用 VPC 的 CIDR 規則將授權輸入規則新增至 Client VPN。如果您想要從您的作用中目錄群組或 SAML 型身分識別提供者 (IdP) 授權特定使用者或群組，請參閱 Client VPN 指南中的授權規則。

若要將 CIDR 新增至AWS Client VPN

1. 打開亞馬遜 VPC 控制台。
2. 選擇「Client VPN 端點」頁面。
3. 選取您的 Client VPN，然後選擇 [授權] 索引標籤 [授權輸入]。
4. 指定下列內容：
   - 您要在目的地網路中啟用的 Amazon VPC 的 CIDR 規則。例如：

     ```
     10.192.0.0/16
     ```

   - 在授與存取權限中選擇允許所有使用者。
   - 在說明中輸入描述性名稱。
5. 選擇「新增授權規則」。

> **Note**
>
> 視 Amazon VPC 的聯網元件而定，您可能還需要將此授權輸入規則納入網路存取控制清單 (NACL)。

步驟 7：下載 Client VPN 端點組態檔案

下載組態檔

1. 請依照下列快速步驟，在下載 Client VPN 端點設定檔案中下載 Client VPN 組態檔案。
2. 在這些步驟中，系統會要求您在 Client VPN 端點 DNS 名稱前面加上字串。範例如下：

- Example 端點 DNS 名稱

  如果您的 Client VPN 端點 DNS 名稱如下所示：

  ```
  remote cvpn-endpoint-0909091212aaee1.prod.clientvpn.us-west-1.amazonaws.com 443
  ```

  您可以新增字串來識別 Client VPN 端點，如下所示：

  ```
  remote mwaavpn.cvpn-endpoint-0909091212aaee1.prod.clientvpn.us-
  west-1.amazonaws.com 443
  ```

3. 在這些步驟中，系統會要求您在新標籤之間新增用戶端憑證的內容與新<cert></cert>標籤組的<key></key>私有金鑰內容。範例如下：

   a. 開啟命令提示字元，並將目錄變更為用戶端憑證和私密金鑰的位置。

   b. Example macOS 用戶端 1. 網域名稱

      要在 macOS 上顯示client1.domain.tld.crt文件的內容，您可以使用cat client1.domain.tld.crt。

      從終端複製值並downloaded-client-config.ovpn像這樣粘貼：

      ```
      ZZZ1111dddaBBB
      -----END CERTIFICATE-----
      </ca>
      <cert>
      -----BEGIN CERTIFICATE-----
      YOUR client1.domain.tld.crt
      -----END CERTIFICATE-----
      </cert>
      ```

   c. Example macOS 用戶端 1. 網域名稱

      若要顯示的內容client1.domain.tld.key，您可以使用cat client1.domain.tld.key。

      從終端複製值並downloaded-client-config.ovpn像這樣粘貼：

      ```
      ZZZ1111dddaBBB
      -----END CERTIFICATE-----
      ```

```
</ca>
<cert>
-----BEGIN CERTIFICATE-----
YOUR client1.domain.tld.crt
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN CERTIFICATE-----
YOUR client1.domain.tld.key
-----END CERTIFICATE-----
</key>
```

第八步：Connect 到AWS Client VPN

的用戶端AWS Client VPN是免費提供的。您可將您的電腦直接連線到AWS Client VPN端到端 VPN 體驗。

連線到 Client VPN

1. 下載並安裝AWS Client VPN桌面版。
2. 開啟 AWS Client VPN。
3. 在 VPN 用戶端功能表中選擇檔案、受管理的設定檔。
4. 選擇 [新增設定檔]，然後選擇`downloaded-client-config.ovpn`。
5. 在顯示名稱中輸入描述性名稱。
6. 選擇新增設定檔、完成。
7. 選擇 Connect (連線)。

連線到 Client VPN 後，您需要中斷與其他 VPN 的連線，才能檢視 Amazon VPC 中的任何資源。

> **Note**
>
> 您可能需要結束用戶端，然後重新開始才能連線。

後續步驟？

- 了解如何在中建立 Amazon MWAA 環境開始使用 Amazon Managed Workflows。您必須在與 Client VPN 相同的AWS區域中建立環境，並使用與 Client VPN 相同的 VPC、私有子網路和安全性群組。

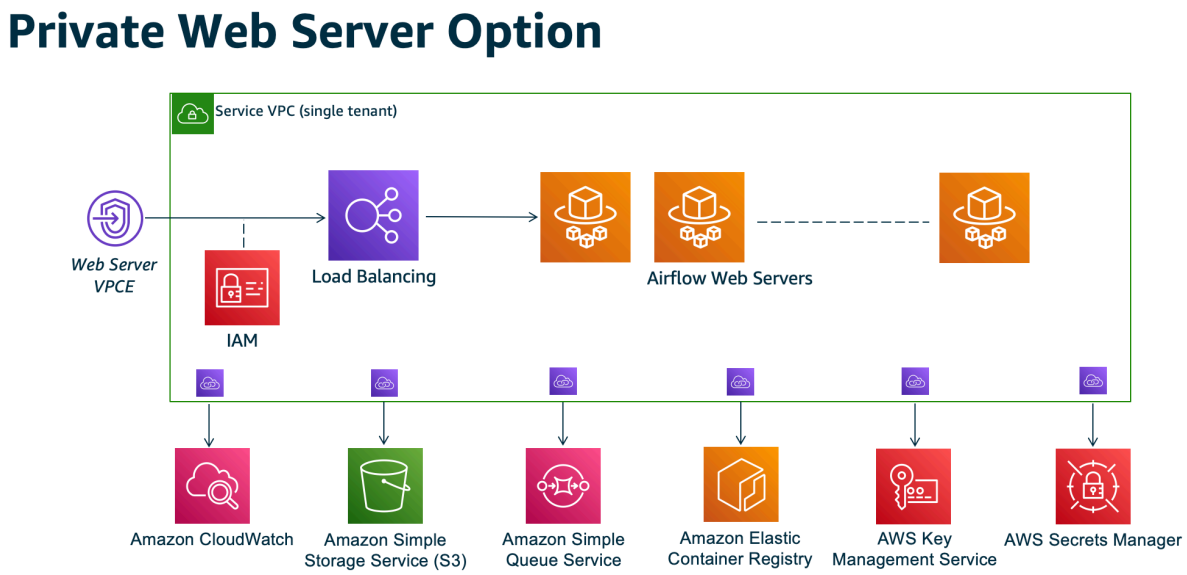
教學課程：使用 Linux 防禦主機設定私人網路存取

本教學將逐步引導您如何建立 SSH 通道，從您的電腦到 Apache 氣流網頁伺服器的步驟，以便為您的亞馬遜管理的 Apache 氣流環境的工作流程。它假設您已經建立了一個亞馬遜 MWAA 環境。一旦設定完成，Linux 防禦主機就像跳躍伺服器一樣，允許從您的電腦與 VPC 中的資源進行安全連線。然後，您將使用 SOCKS 代理管理附加元件來控制瀏覽器中的代理設定，以存取 Apache 氣流使用者介面。

章節

私有網路

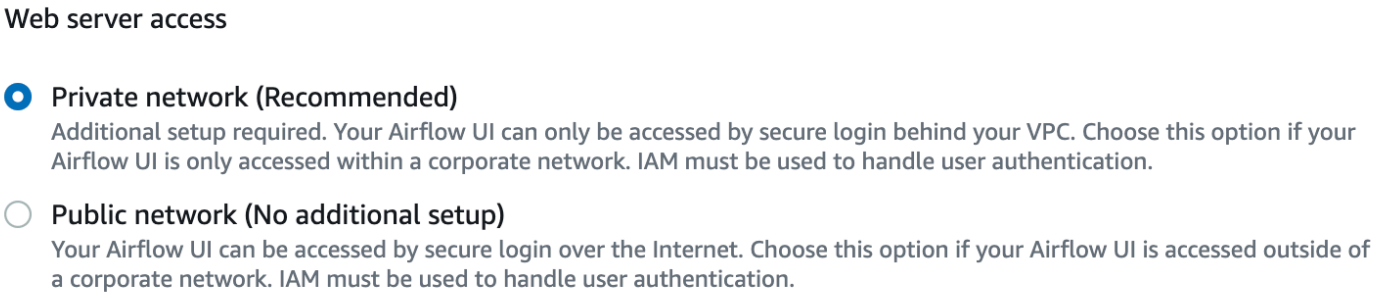
本教學課程假設您已為 Apache Airflow 網頁伺服器選擇私人網路存取模式。



私有網路存取模式可將 Apache Airflow UI 的存取權限限制為 Amazon VPC 中已授予您環境 IAM 政策存取權的使用者。

當您創建具有私有 Web 服務器訪問權限的環境時，必須將所有依賴項打包到 Python wheel 存檔（.whl）中，然後.whl在`requirements.txt`. 有關使用 wheel 打包和安裝依賴項的說明，請參閱使用 Python wheel 管理依賴關係。

下圖顯示 Amazon MWAA 主控台上哪裡可以找到私人網路選項。



使用案例

您可以在建立 Amazon MWAA 環境之後使用本教學課程。您必須使用與環境相同的 Amazon VPC、VPC 安全群組和公有子網路。

開始之前

1. 檢查使用者權限。請確定您在AWS Identity and Access Management (IAM) 中的帳戶具有足夠的許可來建立和管理 VPC 資源。
2. 使用您的亞馬遜 MWAA VPC。本教學假設您要將防禦主機與現有 VPC 相關聯。Amazon VPC 必須與您的 Amazon MWAA 環境位於相同的區域，並且具有兩個私有子網路 (如中所定義)建立虛擬私人雲端網路。
3. 建立 SSH 金鑰。您必須在與 Amazon MWAA 環境相同的區域中建立 Amazon EC2 安全殼層金鑰 (.pem)，才能連線到虛擬伺服器。如果您沒有 SSH 金鑰，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的建立或匯入 key pair。

目標

在本教學中，您將執行下列作業：

1. 使用現有 VPC 的AWS CloudFormation範本建立 Linux 防禦主機執行個體。
2. 使用連接埠上的輸入規則，授權堡壘執行個體安全性群組的輸入流量22。
3. 將 Amazon MWAA 環境安全群組的傳入流量授權至防禦執行個體安全群組。
4. 建立防禦執行個體的 SSH 通道。
5. 安裝並配置火狐瀏覽器的FoxyProxy附加元件，以查看 Apache 氣流使用者介面。

步驟 1：建立堡壘執行個體

下節說明使用AWS CloudFormation主控台上現有 VPC 的AWS CloudFormation範本建立 linux 堡壘執行個體的步驟。

若要建立 Linux 防禦主機

1. 在主控台上開啟 [部署快速入門] 頁AWS CloudFormation面。
2. 使用導覽列中的區域選擇器，選擇與 Amazon MWAA 環境相同的AWS區域。
3. 選擇 下一步。
4. 在「堆疊名稱」文字欄位中輸入名稱，例如`mwaa-linux-bastion`。
5. 在 [參數] 的 [網路組態] 窗格中，選擇下列選項：

    a. 選擇您的亞馬遜 MWAA 環境的虛擬私人雲端識別碼。

b. 選擇您的 Amazon MWAA 環境的公有子網路 1 識別碼。

c. 選擇您的 Amazon MWAA 環境的公有子網路 2 識別碼。

d. 在允許的堡壘外部存取 CIDR 中輸入最小的可能位址範圍 (例如，內部 CIDR 範圍)。

> **Note**
>
> 識別範圍的最簡單方法是使用與公用子網路相同的 CIDR 範圍。例如，[建立虛擬私人 VPC 網路] 頁面上AWS CloudFormation範本中的公用子網路為`10.192.10.0/24`和`10.192.11.0/24`。

6. 在 Amazon EC2 組態窗格上，選擇下列項目：

a. 在金鑰配對名稱的下拉式清單中選擇您的 SSH 金鑰。

b. 在防禦主機名稱中輸入名稱。

c.

d. 為 TCP 轉送選擇真。

> **Warning**
>
> 在此步驟中，必須將 TCP 轉送設定為 true。否則，您將無法在下一步中建立 SSH 通道。

e.

7. 選擇下一步，下一步。

8. 選取確認，然後選擇 [建立堆疊]。

若要深入了解 Linux 防禦主機的架構，請參閱AWS雲端上的 Linux 防禦主機：架構。

步驟二：建立 SSH 隧道

下列步驟說明如何建立 Linux 堡壘的 SSH 隧道。SSH 通道會從您的本地 IP 地址接收請求到 linux 堡壘，這就是為什麼 Linux 堡壘的 TCP 轉發在之前的步驟`true`中設置為的原因。

macOS/Linux

若要透過指令行建立隧道

1. 開啟 Amazon EC2 主控台上的執行個體頁面。
2. 選擇執行個體。
3. 複製在公共 IPv4 DNS 中的地址。例如：ec2-4-82-142-1.compute-1.amazonaws.com。
4. 在命令提示中，導覽至儲存 SSH 金鑰的目錄。
5. 執行下列命令，以使用 ssh 連接至堡壘執行個體。以中的 SSH 金鑰名稱取代範例值mykeypair.pem。

```
ssh -i mykeypair.pem -N -D 8157 ec2-user@YOUR_PUBLIC_IPV4_DNS
```

Windows (PuTTY)

若要使用 PuTTY 建立隧道

1. 開啟 Amazon EC2 主控台上的執行個體頁面。
2. 選擇執行個體。
3. 複製在公共 IPv4 DNS 中的地址。例如：ec2-4-82-142-1.compute-1.amazonaws.com。
4. 打開 PuTTY，選擇會話。
5. 在主機名稱中輸入主機名稱，作為#### _IPV4_DNS 使用者，並將連接埠輸入為22。
6. 展開 SSH 索引標籤，選取驗證。在用於身份驗證的私鑰文件中，選擇您的本地「ppk」文件。
7. 在 [SSH] 下，選擇 [通道] 索引標籤，然後選取 [動態] 和 [自動] 選項。
8. 在 [來源通訊埠] 中，新增8157連接埠 (或任何其他未使用的連接埠)，然後將 [目的地通訊埠] 保留空白。選擇 Add (新增)。
9. 選擇工作階段標籤，然後輸入工作階段名稱。例如：SSH Tunnel。
10. 選擇儲存，開啟。

> **Note**
>
> 您可能需要輸入公開金鑰的密語。

> **Note**
>
> 如果您收到`Permission denied (publickey)`錯誤訊息，我們建議您使用 AWSSupport-疑難排解工具，然後選擇執行此自動化 (主控台) 來疑難排解 SSH 設定。

步驟三：將堡壘安全群組設定為輸入規則

透過附加到這些伺服器的特殊維護安全性群組，允許從伺服器存取伺服器並定期存取網際網路。下列步驟說明如何將堡壘安全群組設定為環境 VPC 安全性群組的輸入流量來源。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [網路] 窗格中，選擇 [VPC 安全性群組]。
4. 選擇 Edit inbound Rules (編輯傳入規則)。
5. 選擇 Add rule (新增規則)。
6. 在「來源」下拉式清單中選擇您的 VPC 安全群組 ID。
7. 將其餘選項保留空白，或設定為其預設值。
8. 選擇 Save rules (儲存規則)。

第四步：複製 Apache 氣流網址

下列步驟說明如何開啟 Amazon MWAA 主控台，並將 URL 複製到 Apache 氣流使用者介面。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 複製氣流使用者介面中的 URL 以進行後續步驟。

步驟五：設定代理伺服器設定

如果您使用 SSH 通道搭配動態連接埠轉送，您必須使用 SOCKS 代理管理附加元件，以控制在瀏覽器中的代理設定。例如，您可以使用 Chromium 的`--proxy-server`功能啟動瀏覽器會話，或者在 MozillaFireFox 瀏覽器中使用FoxyProxy擴展程序。

選項一：使用本地端口轉發設置 SSH 隧道

如果您不想使用 SOCKS 代理伺服器，您可以使用本機連接埠轉送來設定 SSH 通道。下列範例命令會透過轉送本機連接埠 8157 上的流量來存取 Amazon EC2 ResourceManagerWeb 介面。

1. 開啟新的命令提示視窗。
2. 輸入下列命令以開啟 SSH 通道。

```
ssh -i mykeypair.pem -N -L 8157:YOUR_VPC_ENDPOINT_ID-
vpce.YOUR_REGION.airflow.amazonaws.com:443
 ubuntu@YOUR_PUBLIC_IPV4_DNS.YOUR_REGION.compute.amazonaws.com
```

   -L表示使用本地端口轉發，它允許您指定用於將數據轉發到節點本地 Web 服務器上標識的遠程端口的本地端口。

3. `http://localhost:8157/`在您的瀏覽器中輸入。

> **Note**
>
> 您可能需要使用`https://localhost:8157/`。

選項二：通過命令行代理

大多數 Web 瀏覽器允許您通過命令行或配置參數配置代理。例如，使用 Chromium，您可以使用下列命令啟動瀏覽器：

```
chromium --proxy-server="socks5://localhost:8157"
```

這將啟動一個瀏覽器會話，該會使用您在先前步驟中創建的 ssh 隧道來代理其請求。您可以打開您的私人亞馬遜 MWAA 環境網址（使用 https：//），如下所示：

```
https://YOUR_VPC_ENDPOINT_ID-vpce.YOUR_REGION.airflow.amazonaws.com/home.
```

選項三：使用FoxyProxy火狐瀏覽器代理

下面的例子演示了一個FoxyProxy標準（版本 7.5.1）配置火狐瀏覽器。FoxyProxy提供了一組代理管理工具。它可讓您針對符合 Apache 氣流 UI 所使用之網域對應之網域模式的 URL 使用代理伺服器。

1. 在中FireFox，開啟 [標FoxyProxy準擴充功能] 頁面。

2. 選擇新增至火狐瀏覽器。
3. 選擇 Add (新增)。
4. 選擇瀏覽器工具欄中的FoxyProxy圖標，然後選擇「選項」。
5. 複製以下代碼並在本地保存為mwaa-proxy.json。用您的阿帕奇氣流網址替換######中的樣本值。

```
{
  "e0b7kh1606694837384": {
    "type": 3,
    "color": "#66cc66",
    "title": "airflow",
    "active": true,
    "address": "localhost",
    "port": 8157,
    "proxyDNS": false,
    "username": "",
    "password": "",
    "whitePatterns": [
      {
        "title": "airflow-ui",
        "pattern": "YOUR_HOST_NAME",
        "type": 1,
        "protocols": 1,
        "active": true
      }
    ],
    "blackPatterns": [],
    "pacURL": "",
    "index": -1
  },
  "k20d21508277536715": {
    "active": true,
    "title": "Default",
    "notes": "These are the settings that are used when no patterns match a URL.",
    "color": "#0055E5",
    "type": 5,
    "whitePatterns": [
      {
        "title": "all URLs",
        "active": true,
        "pattern": "*",
        "type": 1,
```

```
        "protocols": 1
      }
    ],
    "blackPatterns": [],
    "index": 9007199254740991
  },
  "logging": {
    "active": true,
    "maxSize": 500
  },
  "mode": "patterns",
  "browserVersion": "82.0.3",
  "foxyProxyVersion": "7.5.1",
  "foxyProxyEdition": "standard"
}
```

6. 在「從FoxyProxy 6.0+ 匯入設定」窗格中，選擇「匯入設定」，然後選取`mwaa-proxy.json`檔案。
7. 選擇 OK (確定)。

第六步：打開 Apache 氣流用戶界面

下列步驟說明如何開啟您的 Apache 氣流使用者介面。

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 選擇「開啟氣流 UI」。

後續步驟？

- 了解如何在中的防禦主機的 SSH 通道上執行 Airflow CLI 命令阿帕奇氣流 CLI 命令參考。
- 了解如何在中將 DAG 程式碼上傳到您的 Amazon S3 儲存貯體新增或更新 DAG。

教學課程：限制 Amazon MWAA 使用者對 DAG 子集的存取

Amazon MWAA 會將您的 IAM 主體對應到一或多個 Apache 氣流的預設角色，以管理您對環境的存取。下列教學課程說明如何限制個別 Amazon MWAA 使用者僅檢視特定 DAG 或一組 DAG 並與之互動。

> **Note**
>
> 只要可以使用 IAM 角色，就可以使用聯合存取來完成本教學課程中的步驟。

主題

先決條件

若要完成本教學課程的步驟，您將需要執行下列操作：

- 具有多個 DAG 的亞馬遜 MWAA 環境
- Admin具有AdministratorAccess許可的 IAM 主體和 IAM 使用者MWAAUser，做為您可以限制其 DAG 存取權的主體。如需管理員角色的詳細資訊，請參閱《IAM 使用者指南》中的管理員工作職能

  > **Note**
  >
  > 請勿將權限政策直接附加到 IAM 使用者。我們建議您設定使用者可假設的 IAM 角色，以暫時存取您的 Amazon MWAA 資源。

- AWS Command Line Interface版本 2 已安裝。

步驟一：使用預設的**Public** Apache 氣流角色，為您的 IAM 主體提供 Amazon MWAA 網路伺服器存取權。

若要使用授與權限AWS Management Console

1. 使用Admin角色登入您的AWS帳戶，然後開啟 IAM 主控台。

2. 在左側導覽窗格中，選擇「使用者」，然後從「使用者」表格中選擇您的 Amazon MWAA IAM 使用者。
3. 在使用者詳細資料頁面的 [摘要] 下，選擇 [權限] 索引標籤，然後選擇 [權限] 原則以展開卡片並選擇 [新增權限]。
4. 在 [授與權限] 區段中，選擇 [直接附加現有原則]，然後選擇 [建立原則] 以建立並附加您自己的自訂權限原則。
5. 在 [建立原則] 頁面上，選擇 [JSON]，然後在原則編輯器中複製並貼上下列 JSON 權限原則。Tha原則會將網頁伺服器存取權授與具有預設Public Apache 氣流角色的使用者。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Action": "airflow:CreateWebLoginToken",
        "Resource": [

 "arn:aws:airflow:YOUR_REGION:YOUR_ACCOUNT_ID:role/YOUR_ENVIRONMENT_NAME/Public"
            ]
        }
    ]
 }
```

步驟二：建立新的 Apache 氣流自訂角色

若要使用 Apache 氣流使用者介面來執行下列動作：

1. 使用您的管理員 IAM 角色，開啟 Amazon MWAA 主控台，然後啟動環境的 Apache 氣流使用者介面。
2. 在頂端的導覽窗格中，將游標暫留在 [安全性] 上以開啟下拉式清單，然後選擇 [列出角色] 以檢視預設的 Apache Airflow 角色。
3. 從角色清單中選取 [使用者]，然後在頁面頂端選擇 [動作] 以開啟下拉式清單。選擇「複製角色」並確認 「確定」

> **Note**
>
> 複製 Ops 或檢視者角色以分別授予或多或少存取權。

4. 找出您在表格中建立的新角色，然後選擇 [編輯記錄]。
5. 在編輯角色頁面上，執行下列動作：

   - 在 [名稱] 中，在文字欄位中輸入角色的新名稱。例如：**Restricted**。
   - 對於 Perimssions 清單，請移除`can read on DAGs`和`can edit on DAGs`，然後為您要提供存取權的 DAG 集新增讀取和寫入權限。例如，對於 DAG`example_dag.py`，新增**can read on DAG:*example_dag***和**can edit on DAG:*example_dag***。

   選擇 Save (儲存)。您現在應該有一個新角色，限制對 Amazon MWAA 環境中可用 DAG 子集的存取。您現在可以將此角色指派給任何現有的 Apache 氣流使用者。

步驟三：將您建立的角色指派給 Amazon MWAA 使用者

若要指派新角色

1. 使用存取登入資料來執行下列 CLI 命令`MWAAUser`，以擷取環境的 Web 伺服器 URL。

   ```
   $ aws mwaa get-environment --name YOUR_ENVIRONMENT_NAME | jq
    '.Environment.WebserverUrl'
   ```

   如果成功登入，您將會看到下列輸出結果：

   ```
   "ab1b2345-678a-90a1-a2aa-34a567a8a901.c13.us-west-2.airflow.amazonaws.com"
   ```

2. `MWAAUser`登入後AWS Management Console，開啟新的瀏覽器視窗並存取下列 URL。`Webserver-URL`以您的資訊取代。

   ```
   https://<Webserver-URL>/home
   ```

   如果成功，您會看到`Forbidden`錯誤頁面，因為尚`MWAAUser`未獲得存取 Apache Airflow UI 的權限。

3. `Admin`登入後AWS Management Console，再次開啟 Amazon MWAA 主控台，然後啟動您環境的 Apache 氣流使用者介面。
4. 在 UI 儀表板中，展開 [安全性] 下拉式清單，這次選擇 [列出使用者]。
5. 在使用者表格中，尋找新的 Apache 氣流使用者，然後選擇編輯記錄。使用者的名字會與您的 IAM 使用者名稱相符，模式如下：`user/`*`mwaa-user`*。

6. 在 [編輯使用者] 頁面的 [角色] 區段中，新增您建立的新自訂角色，然後選擇 [儲存]。

> **Note**
>
> 「姓氏」欄位是必填欄位，但空格可滿足需求。

IAM`Public` 主體授與存取 Apache Airflow 使用者介面的`MWAAUser`權限，而新角色則提供查看其 DAG 所需的其他權限。

> **Important**
>
> 使用 Apache Airflow UI 新增的未經 IAM 授權的 5 個預設角色 (例如`Admin`) 中的任何一個，都會在下次使用者登入時移除。

後續步驟

- 若要進一步了解如何管理 Amazon MWAA 環境的存取權，以及查看可用於環境使用者的 JSON IAM 政策範例，請參閱the section called “存取 Amazon MWAA 環境”

相關資源

- 存取控制 (Apache 氣流文件) — 進一步了解 Apache 氣流文件網站上的預設 Apache 氣流角色。

教學課程：在 Amazon MWAA 上自動管理您自己的環境端點

如果您用AWS Organizations來管理共用資源的多個AWS帳戶，Amazon MWAA 可讓您建立和管理自己的 Amazon VPC 端點。這表示您可以使用更嚴格的安全性原則，只允許存取環境所需的資源。

當您在共用的 Amazon VPC 中建立環境時，擁有主要 Amazon VPC (擁有者) 的帳戶會與屬於同一組織的其他帳戶 (參與者) 共用 Amazon MWAA 所需的兩個私有子網路。然後，共用這些子網路的參與者帳戶可以檢視、建立、修改和刪除共用 VPC 中的環境。

當您在共用或受政策限制的 Amazon VPC 中建立環境時，Amazon MWAA 會先建立服務 VPC 資源，然後進入最多 72 小時的`PENDING`狀態。

當環境狀態從變更CREATING為時PENDING，Amazon MWAA 會傳送有關狀態變更的 Amazon EventBridge 通知。這可讓擁有者帳戶根據來自 Amazon MWAA 主控台或 API 的端點服務資訊代表參與者建立所需的端點，或以程式設計方式在接下來，我們使用 Lambda 函數和監聽 Amazon MWAA 狀態變更通知的 EventBridge 規則建立新的 Amazon VPC 端點。

在這裡，我們在與環境相同的 Amazon VPC 中建立新端點。若要設定共用的 Amazon VPC，請在擁有者帳戶中建立 EventBridge 規則，並在參與者帳戶中建立 Amazon MWAA 環境中的 Lambda 函數。

主題

必要條件

若要完成本教學課程中的步驟，您將需要下列項目：

- ...

創建 Amazon VPC

使用下列AWS CloudFormation範本和AWS CLI命令建立新的 Amazon VPC。範本會設定 Amazon VPC 資源並修改端點政策，以限制對特定佇列的存取。

1. 下載AWS CloudFormation範本，然後解壓縮.yml檔案。
2. 在新指令提示視窗中，導覽至儲存樣板的資料夾，然後使create-stack用建立堆疊。該--template-body標誌指定模板的路徑。

```
$ aws cloudformation create-stack --stack-name stack-name --template-body file://
cfn-vpc-private-network.yml
```

在下一節中，您將建立 Lambda 函數。

建立 Lambda 函數

使用下列 Python 程式碼和 IAM JSON 政策來建立新的 Lambda 函數和執行角色。此函數會為私有 Apache 氣流網路伺服器和 Amazon SQS 佇列建立亞馬遜 VPC 端點。擴展您的環境時，Amazon MWAA 會使用 Amazon SQS 將多個工作者與 Celly 的任務排入佇列。

1. 下載 Python 函數代碼。
2. 下載 IAM 權限政策，然後解壓縮檔案。
3. 開啟命令提示字元，然後瀏覽至儲存 JSON 權限原則的資料夾。使用 IAM `create-role`命令建立新角色。

   ```
   $ aws iam create-role --role-name function-role \
   --assume-role-policy-document file://lambda-mwaa-vpce-policy.json
   ```

   請注意來自AWS CLI回應的角色 ARN。在下一步中，我們使用其 ARN 將此新角色指定為函數的執行角色。
4. 導覽至儲存函數代碼的資料夾，然後使用`create-function`指令建立新函數。

   ```
   $ aws lambda create-function --function-name mwaa-vpce-lambda \
   --zip-file file://mwaa-lambda-shared-vpc.zip --runtime python3.8 --role
    arn:aws:iam::123456789012:role/function-role --handler lambda_handler
   ```

   請注意來自AWS CLI響應的函數 ARN。在下一步中，我們指定 ARN 以將函數配置為新 EventBridge 規則的目標。

在下一節中，您將建立當環境進入`PENDING`狀態時叫用此函數的 EventBridge 規則。

建立規 EventBridge 則

執行下列動作以建立接聽 Amazon MWAA 通知並鎖定新 Lambda 函數的新規則。

1. 使用 EventBridge `put-rule`指令建立新 EventBridge 規則。

   ```
   $ aws events put-rule --name "mwaa-lambda-rule" \
   --event-pattern "{\"source\":[\"aws.airflow\"],\"detail-type\":[\"MWAA Environment
    Status Change\"]}"
   ```

   事件模式會在環境狀態變更時接聽 Amazon MWAA 傳送的通知。

```
{
  "source": ["aws.airflow"],
  "detail-type": ["MWAA Environment Status Change"]
}
```

2. 使用put-targets命令將 Lambda 函數新增為新規則的目標。

```
$ aws events put-targets --rule "mwaa-lambda-rule" \
--targets "Id"="1","Arn"="arn:aws::lambda:region:123456789012:function:mwaa-vpce-lambda"
```

您已準備好使用客戶管理的 Amazon VPC 端點建立新的 Amazon MWAA 環境。

建立 Amazon MWAA 環境

使用 Amazon MWAA 主控台建立具有客戶管理的 Amazon VPC 端點的新環境。

1. 開啟 Amazon MWAA 主控台，然後選擇 [建立環境]。
2. 在「名稱」中，輸入唯一的名稱。
3. 對於氣流版本，請選擇最新版本。
4. 選擇 Amazon S3 儲存貯體和 DAG 資料夾，例如與環境搭配dags/使用，然後選擇「下一步」。
5. 在 [設定進階設定] 頁面上，執行下列動作：
   a. 對於虛擬私有雲，請選擇您在上一步中建立的 Amazon VPC。
   b. 對於 Web 伺服器存取，請選擇 [公用網路 (可存取網際網路)]
   c. 在「安全性群組」中，選擇您使用建立的安全性群組AWS CloudFormation。由於先前步驟中AWS PrivateLink端點的安全性群組是自我參照的，因此您必須為環境選擇相同的安全性群組。
   d. 對於端點管理，請選擇「客戶管理端點」。
6. 保留剩餘的預設設定，然後選擇「下一步」。
7. 檢閱您的選取項目，然後選擇建立環境。

> **Tip**
>
> 如需有關設定新環境的詳細資訊，請參閱開始使用 Amazon MWAA。

當環境為時PENDING，Amazon MWAA 會傳送符合您為規則設定的事件模式的通知。此規則會叫用您的 Lambda 函數。此函數會剖析通知事件，並取得 Web 伺服器和 Amazon SQS 佇列的必要端點資訊。然後，它會在您的 Amazon VPC 中建立端點。

當端點可用時，Amazon MWAA 會繼續建立您的環境。準備就緒後，環境狀態會變更為AVAILABLE，您可以使用 Amazon MWAA 主控台存取 Apache 氣流網路伺服器。

Amazon Managed Workflows

本指南包含程式碼範例，包括 DAG 和自訂外掛程式，您可以在適用於 Apache Airflow 環境的 Amazon 受管工作流程中使用。如需將 Apache 氣流與AWS服務搭配使用的更多範例，請參閱 Apache 氣流 GitHub儲存庫中的`example_dags`目錄。

範例

- 使用 DAG 在 CLI 中匯入變數
- 使用建立 SSH 連線 SSHOperator
- 使用密鑰AWS Secrets Manager阿帕奇氣流雪花連接
- 使用 DAG 寫入自訂指標CloudWatch
- Amazon MWAA 環境上的 Aurora 資料庫清理
- 將環境中繼資料匯出到 Amazon S3 上的 CSV 檔案
- 使用密鑰AWS Secrets Manager對於一個阿帕奇氣流變量
- 使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接
- 使用 Oracle 創建一個自定義插件
- 建立產生執行階段環境變數的自訂外掛程式
- 在亞馬遜 MWAA 上更改 DAG 的時區
- 刷新一個 CodeArtifact 代幣
- 創建一個自定義插件與阿帕奇蜂巢和 Hadoop
- 為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator
- 使用拉姆達函數調用 DAG
- 在不同的亞馬遜 MWAA 環境中叫用 DAG
- 使用亞馬遜 MWAA 與亞馬遜 RDS 微軟 SQL 服務器
- 與 Amazon EMR 搭配 Amazon EMR 與 Amazon EMR 搭配 Amazon EMR R
- 使用亞馬遜 MWAA 與亞馬遜 EKS
- 使用連接到亞馬遜 ECSECSOperator
- 使用 dbt 與亞馬遜 MWAA
- AWS博客和教程

使用 DAG 在 CLI 中匯入變數

下列範例程式碼使用亞馬遜上的 CLI 匯入變數，適用於 Apache 氣流的受管工作流程。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

- 使用此頁面上的程式碼範例不需要其他權限。

許可

您的AWS帳戶需要存取AmazonMWAAAirflowCliAccess政策。如需進一步了解，請參閱 阿帕奇氣流 CLI 政策:亞馬遜 AirflowCliAccess。

相依性

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V2 基地安裝在您的環境中。

程式碼範例

下列範例程式碼需要三個輸入：您的 Amazon MWAA 環境名稱 (mwaa_env), 該AWS您環境的區域 (位於aws_region)，以及包含您要匯入之變數的本機檔案 (var_file).

```
import boto3
import json
import requests
import base64
import getopt
import sys

argv = sys.argv[1:]
mwaa_env=''
aws_region=''
var_file=''

try:
    opts, args = getopt.getopt(argv, 'e:v:r:', ['environment', 'variable-
file','region'])
    #if len(opts) == 0 and len(opts) > 3:
    if len(opts) != 3:
        print ('Usage: -e MWAA environment -v variable file location and filename -r
 aws region')
    else:
        for opt, arg in opts:
            if opt in ("-e"):
                mwaa_env=arg
            elif opt in ("-r"):
                aws_region=arg
            elif opt in ("-v"):
                var_file=arg

        boto3.setup_default_session(region_name="{}".format(aws_region))
        mwaa_env_name = "{}".format(mwaa_env)

        client = boto3.client('mwaa')
        mwaa_cli_token = client.create_cli_token(
            Name=mwaa_env_name
        )

        with open ("{}".format(var_file), "r") as myfile:
            fileconf = myfile.read().replace('\n', '')

        json_dictionary = json.loads(fileconf)
        for key in json_dictionary:
            print(key, " ", json_dictionary[key])
            val = (key + " " + json_dictionary[key])
```

```
            mwaa_auth_token = 'Bearer ' + mwaa_cli_token['CliToken']
            mwaa_webserver_hostname = 'https://{0}/aws_mwaa/
cli'.format(mwaa_cli_token['WebServerHostname'])
            raw_data = "variables set {0}".format(val)
            mwaa_response = requests.post(
                mwaa_webserver_hostname,
                headers={
                    'Authorization': mwaa_auth_token,
                    'Content-Type': 'text/plain'
                    },
                data=raw_data
                )
            mwaa_std_err_message = base64.b64decode(mwaa_response.json()
['stderr']).decode('utf8')
            mwaa_std_out_message = base64.b64decode(mwaa_response.json()
['stdout']).decode('utf8')
            print(mwaa_response.status_code)
            print(mwaa_std_err_message)
            print(mwaa_std_out_message)

except:
    print('Use this script with the following options: -e MWAA environment -v variable
 file location and filename -r aws region')
    print("Unexpected error:", sys.exc_info()[0])
    sys.exit(2)
```

後續步驟？

- 了解如何將此範例中的 DAG 程式碼上傳至dags您的亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。

使用建立 SSH 連線 **SSHOperator**

以下範例說明如何使用定向非循環圖 (DAG) SSHOperator 中的，從適用於 Apache 氣流環境的 Amazon 受管工作流程連接到遠端 Amazon EC2 執行個體。您可以使用類似的方法連接到任何具有 SSH 訪問權限的遠程實例。

在下列範例中，您將 SSH 秘密金鑰 (.pem) 上傳到 Amazon S3 上環境的dags目錄。然後，您可以使用安裝必要的依賴關係，requirements.txt並在 UI 中創建一個新的 Apache 氣流連接。最後，您撰寫的 DAG 會建立與遠端執行個體的 SSH 連線。

主題

版本

- 您可以使用此頁面上的代碼示例與 Apache 氣流 v2 及更高版本在 Python 3.10 中。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。
- 一個安全殼層密鑰。程式碼範例假設您有一個 Amazon EC2 執行個體，以及與 Amazon MWAA 環境.pem位於相同區域中的執行個體。如果您沒有金鑰，請參閱 Amazon EC2 Linux 執行個體使用者指南中的建立或匯入 key pair。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

需求

添加以下參數requirements.txt以在 Web 服務器上安裝apache-airflow-providers-ssh軟件包。一旦您的環境更新並且 Amazon MWAA 成功安裝相依性，您就會在使用者介面中看到新的 SSH 連線類型。

```
-c https://raw.githubusercontent.com/apache/airflow/constraints-Airflow-version/
constraints-Python-version.txt
```

```
apache-airflow-providers-ssh
```

> **Note**
>
> `-c`定義中的條件約束 URL `requirements.txt`。如此可確保 Amazon MWAA 為您的環境安裝正確的套件版本。

將您的密鑰複製到 Amazon S3

使用下列AWS Command Line Interface命令將`.pem`金鑰複製到 Amazon S3 中環境的`dags`目錄。

```
$ aws s3 cp your-secret-key.pem s3://your-bucket/dags/
```

亞馬遜 MWAA 將中`dags`的內容 (包括`.pem`金鑰) 複製到本機`/usr/local/airflow/dags/`目錄，這樣一來，Apache 氣流可以存取金鑰。

創建一個新的 Apache 氣流連接

若要使用 Apache 氣流使用者介面建立新的 SSH 連線

1. 在亞馬遜 MWAA 主控台上開啟「環境」頁面。
2. 從環境清單中，為您的環境選擇開啟 Airflow UI。
3. 在 Apache Airflow UI 頁面上，從頂端導覽列選擇 [管理員] 以展開下拉式清單，然後選擇 [連線]。
4. 在 [列出連線] 頁面上，選擇 [+] 或 [新增記錄] 按鈕以新增連線。
5. 在 [新增連線] 頁面上，新增下列資訊：

   a. 對於「連線 ID」，輸入**ssh_new**。

   b. 在 [連線類型] 中，從下拉式清單中選擇 [SSH]。

   > **Note**
   >
   > 如果清單中沒有 SSH 連線類型，表示 Amazon MWAA 尚未安裝所需`apache-airflow-providers-ssh`的套件。請更新您的`requirements.txt`檔案以包含此套件，然後再試一次。

   c. 對於主機，請輸入要連接到的 Amazon EC2 執行個體的 IP 地址。例如：**12.345.67.89**。

d. 對於使用者名稱，請輸入是**ec2-user**否要連線到 Amazon EC2 執行個體。您的使用者名稱可能會有所不同，視您希望 Apache Airflow 連線的遠端執行個體類型而定。

e. 在「額外」中，輸入以下 JSON 格式的機碼值組：

```
{ "key_file": "/usr/local/airflow/dags/your-secret-key.pem" }
```

此機碼值配對會指示 Apache 氣流尋找本機/dags目錄中的秘密金鑰。

程式碼範例

下面的 DAG 使用連接SSHOperator到您的目標 Amazon EC2 實例，然後運行 hostname Linux 命令來打印安裝的名稱。您可以修改 DAG 以在遠端執行個體上執行任何命令或指令碼。

1. 開啟終端機，然後瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為ssh.py。

```
from airflow.decorators import dag
from datetime import datetime
from airflow.providers.ssh.operators.ssh import SSHOperator

@dag(
    dag_id="ssh_operator_example",
    schedule_interval=None,
    start_date=datetime(2022, 1, 1),
    catchup=False,
    )
def ssh_dag():
    task_1=SSHOperator(
        task_id="ssh_task",
        ssh_conn_id='ssh_new',
        command='hostname',
    )

my_ssh_dag = ssh_dag()
```

3. 執行下列AWS CLI命令，將 DAG 複製到您環境的儲存貯體，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果成功，您會在 ssh_operator_example DAG 中的工作記錄中看到類似下列的ssh_task輸出：

```
[2022-01-01, 12:00:00 UTC] {{base.py:79}} INFO - Using connection to: id: ssh_new.
 Host: 12.345.67.89, Port: None,
Schema: , Login: ec2-user, Password: None, extra: {'key_file': '/usr/local/airflow/
dags/your-secret-key.pem'}
[2022-01-01, 12:00:00 UTC] {{ssh.py:264}} WARNING - Remote Identification Change is
 not verified. This won't protect against Man-In-The-Middle attacks
[2022-01-01, 12:00:00 UTC] {{ssh.py:270}} WARNING - No Host Key Verification. This
 won't protect against Man-In-The-Middle attacks
[2022-01-01, 12:00:00 UTC] {{transport.py:1819}} INFO - Connected (version 2.0,
 client OpenSSH_7.4)
[2022-01-01, 12:00:00 UTC] {{transport.py:1819}} INFO - Authentication (publickey)
 successful!
[2022-01-01, 12:00:00 UTC] {{ssh.py:139}} INFO - Running command: hostname
[2022-01-01, 12:00:00 UTC]{{ssh.py:171}} INFO - ip-123-45-67-89.us-
west-2.compute.internal
[2022-01-01, 12:00:00 UTC] {{taskinstance.py:1280}} INFO - Marking task as SUCCESS.
 dag_id=ssh_operator_example, task_id=ssh_task, execution_date=20220712T200914,
 start_date=20220712T200915, end_date=20220712T200916
```

使用密鑰AWS Secrets Manager阿帕奇氣流雪花連接

下列範例呼叫AWS Secrets Manager為 Apache 氣流亞馬遜管理工作流阿帕奇氣流雪花連接獲取秘密密鑰。它假設您已完成中的步驟使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 秘密管理器後端作為 Apache 氣流配置選項，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。
- 秘密管理器中的 Apache 氣流連接字符串，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

許可

- 密碼管理員權限，如所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

請求

若要使用此頁面上的範例程式碼，請將下列相依性新增至`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴。

```
apache-airflow-providers-snowflake==1.3.0
```

程式碼範例

下列步驟說明如何建立呼叫秘密管理員以取得密碼的 DAG 程式碼。

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

   ```
   cd dags
   ```

2. 複製下列程式碼範例的內容，並在本機儲存為`snowflake_connection.py`。

   ```
   """
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

   Permission is hereby granted, free of charge, to any person obtaining a copy of
   ```

```
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG
from airflow.providers.snowflake.operators.snowflake import SnowflakeOperator
from airflow.utils.dates import days_ago

snowflake_query = [
    """use warehouse "MY_WAREHOUSE";""",
    """select * from "SNOWFLAKE_SAMPLE_DATA"."WEATHER"."WEATHER_14_TOTAL" limit
 100;""",
]

with DAG(dag_id='snowflake_test', schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    snowflake_select = SnowflakeOperator(
        task_id="snowflake_select",
        sql=snowflake_query,
        snowflake_conn_id="snowflake_conn",
    )
```

後續步驟？

- 了解如何將此範例中的 DAG 程式碼上傳至dags亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。

使用 DAG 寫入自訂指標CloudWatch

您可以使用下列程式碼範例來撰寫執行PythonOperator擷取 Amazon MWAA 環境的作業系統層級指標。然後，DAG 將數據作為自定義指標發布到亞馬遜CloudWatch。

自訂作業系統層級指標可讓您進一步瞭解環境工作者如何利用虛擬記憶體和 CPU 等資源。您可以使用此資訊來選取環境類最適合您的工作負載。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的程式碼範例，您需要下列項目：

- 一個亞馬遜 MWAA 環境。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

相依性

- 使用此頁面上的程式碼範例不需要其他相依性。

程式碼範例

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的資料夾。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並將其儲存為本機`dag-custom-metrics.py`。取代`MWAA-ENV-NAME`使用您的環境名稱。

```
from airflow import DAG
```

```
from airflow.operators.python_operator import PythonOperator
from airflow.utils.dates import days_ago
from datetime import datetime
import os,json,boto3,psutil,socket

def publish_metric(client,name,value,cat,unit='None'):
    environment_name = os.getenv("MWAA_ENV_NAME")
    value_number=float(value)
    hostname = socket.gethostname()
    ip_address = socket.gethostbyname(hostname)
    print('writing value',value_number,'to metric',name)
    response = client.put_metric_data(
        Namespace='MWAA-Custom',
        MetricData=[
            {
                'MetricName': name,
                'Dimensions': [
                    {
                        'Name': 'Environment',
                        'Value': environment_name
                    },
                    {
                        'Name': 'Category',
                        'Value': cat
                    },
                    {
                        'Name': 'Host',
                        'Value': ip_address
                    },
                ],
                'Timestamp': datetime.now(),
                'Value': value_number,
                'Unit': unit
            },
        ]
    )
    print(response)
    return response

def python_fn(**kwargs):
    client = boto3.client('cloudwatch')

    cpu_stats = psutil.cpu_stats()
    print('cpu_stats', cpu_stats)
```

```
    virtual = psutil.virtual_memory()
    cpu_times_percent = psutil.cpu_times_percent(interval=0)

    publish_metric(client=client, name='virtual_memory_total',
 cat='virtual_memory', value=virtual.total, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_available',
 cat='virtual_memory', value=virtual.available, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_used', cat='virtual_memory',
 value=virtual.used, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_free', cat='virtual_memory',
 value=virtual.free, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_active',
 cat='virtual_memory', value=virtual.active, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_inactive',
 cat='virtual_memory', value=virtual.inactive, unit='Bytes')
    publish_metric(client=client, name='virtual_memory_percent',
 cat='virtual_memory', value=virtual.percent, unit='Percent')

    publish_metric(client=client, name='cpu_times_percent_user',
 cat='cpu_times_percent', value=cpu_times_percent.user, unit='Percent')
    publish_metric(client=client, name='cpu_times_percent_system',
 cat='cpu_times_percent', value=cpu_times_percent.system, unit='Percent')
    publish_metric(client=client, name='cpu_times_percent_idle',
 cat='cpu_times_percent', value=cpu_times_percent.idle, unit='Percent')

    return "OK"


with DAG(dag_id=os.path.basename(__file__).replace(".py", ""),
 schedule_interval='*/5 * * * *', catchup=False, start_date=days_ago(1)) as dag:
    t = PythonOperator(task_id="memory_test", python_callable=python_fn,
 provide_context=True)
```

3. 運行以下AWS CLI命令，將 DAG 複製到您環境的值區，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果 DAG 成功執行，您應該會在 Apache 氣流記錄中看到類似下列內容的內容：

```
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO -
 cpu_stats scpustats(ctx_switches=3253992384, interrupts=1964237163,
 soft_interrupts=492328209, syscalls=0)
```

```
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - writing value
 16024199168.0 to metric virtual_memory_total
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - {'ResponseMetadata':
 {'RequestId': 'fad289ac-aa51-46a9-8b18-24e4e4063f4d', 'HTTPStatusCode': 200,
 'HTTPHeaders': {'x-amzn-requestid': 'fad289ac-aa51-46a9-8b18-24e4e4063f4d',
 'content-type': 'text/xml', 'content-length': '212', 'date': 'Tue, 16 Aug 2022
 17:54:45 GMT'}, 'RetryAttempts': 0}}
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - writing value
 14356287488.0 to metric virtual_memory_available
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - {'ResponseMetadata':
 {'RequestId': '6ef60085-07ab-4865-8abf-dc94f90cab46', 'HTTPStatusCode': 200,
 'HTTPHeaders': {'x-amzn-requestid': '6ef60085-07ab-4865-8abf-dc94f90cab46',
 'content-type': 'text/xml', 'content-length': '212', 'date': 'Tue, 16 Aug 2022
 17:54:45 GMT'}, 'RetryAttempts': 0}}
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - writing value
 1342296064.0 to metric virtual_memory_used
[2022-08-16, 10:54:46 UTC] {{logging_mixin.py:109}} INFO - {'ResponseMetadata':
 {'RequestId': 'd5331438-5d3c-4df2-bc42-52dcf8d60a00', 'HTTPStatusCode': 200,
 'HTTPHeaders': {'x-amzn-requestid': 'd5331438-5d3c-4df2-bc42-52dcf8d60a00',
 'content-type': 'text/xml', 'content-length': '212', 'date': 'Tue, 16 Aug 2022
 17:54:45 GMT'}, 'RetryAttempts': 0}}
...
[2022-08-16, 10:54:46 UTC] {{python.py:152}} INFO - Done. Returned value was: OK
[2022-08-16, 10:54:46 UTC] {{taskinstance.py:1280}} INFO - Marking task as SUCCESS.
 dag_id=dag-custom-metrics, task_id=memory_test, execution_date=20220816T175444,
 start_date=20220816T175445, end_date=20220816T175446
[2022-08-16, 10:54:46 UTC] {{local_task_job.py:154}} INFO - Task exited with return
 code 0
```

Amazon MWAA 環境上的 Aurora 資料庫清理

適用於 Apache 氣流的 Amazon 受管工作流程使用 Aurora PostgreSQL 資料庫做為 Apache 氣流中繼資料庫，在此資料庫執行 DAG 執行和任務執行個體 下列範例程式碼會定期清除 Amazon MWAA 環境專用 Aurora PostgreSQL 資料庫中的項目。

主題

版本

- 您可以使用此頁面上的代碼示例與 Apache 氣流 v2 及更高版本在 Python 3.10 中。

必要條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- Amazon MWAA 環境。

相依性

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用 Apache 氣流 v2 基本安裝在您的環境。

範例程式碼

下列 DAG 會清除中指定之表格的中繼資料資料庫TABLES_TO_CLEAN。這個範例會刪除指定資料表中過去七天的資料。若要調整項目的刪除距離，請將其設定MAX_AGE_IN_DAYS為不同的值。

Apache Airflow v2

```
from airflow import settings
from airflow.utils.dates import days_ago
from airflow.models import DagTag, DagModel, DagRun, ImportError, Log, SlaMiss,
 RenderedTaskInstanceFields, TaskInstance, TaskReschedule, XCom
from airflow.decorators import dag, task
from airflow.utils.dates import days_ago
from time import sleep

from airflow.version import version
major_version, minor_version = int(version.split('.')[0]), int(version.split('.')
[1])
if major_version >= 2 and minor_version >= 6:
    from airflow.jobs.job import Job
else:
    # The BaseJob class was renamed as of Apache Airflow v2.6
    from airflow.jobs.base_job import BaseJob as Job
```

```
# Delete entries for the past seven days. Adjust MAX_AGE_IN_DAYS to set how far back
 this DAG cleans the database.
MAX_AGE_IN_DAYS = 7
MIN_AGE_IN_DAYS = 0
DECREMENT = -7

# This is a list of (table, time) tuples.
# table = the table to clean in the metadata database
# time  = the column in the table associated to the timestamp of an entry
#         or None if not applicable.
TABLES_TO_CLEAN = [[Job, Job.latest_heartbeat],
    [TaskInstance, TaskInstance.execution_date],
    [TaskReschedule, TaskReschedule.execution_date],
    [DagTag, None],
    [DagModel, DagModel.last_parsed_time],
    [DagRun, DagRun.execution_date],
    [ImportError, ImportError.timestamp],
    [Log, Log.dttm],
    [SlaMiss, SlaMiss.execution_date],
    [RenderedTaskInstanceFields, RenderedTaskInstanceFields.execution_date],
    [XCom, XCom.execution_date],
]

@task()
def cleanup_db_fn(x):
    session = settings.Session()

    if x[1]:
        for oldest_days_ago in range(MAX_AGE_IN_DAYS, MIN_AGE_IN_DAYS, DECREMENT):
            earliest_days_ago = max(oldest_days_ago + DECREMENT, MIN_AGE_IN_DAYS)
            print(f"deleting {str(x[0])} entries between {earliest_days_ago} and
 {oldest_days_ago} days old...")
            earliest_date = days_ago(earliest_days_ago)
            oldest_date = days_ago(oldest_days_ago)
            query = session.query(x[0]).filter(x[1] >= oldest_date).filter(x[1] <=
 earliest_date)
            query.delete(synchronize_session= False)
            session.commit()
            sleep(5)
    else:
        # No time column specified for the table. Delete all entries
        print("deleting", str(x[0]), "...")
        query = session.query(x[0])
        query.delete(synchronize_session= False)
```

```
        session.commit()

    session.close()


@dag(
    dag_id="cleanup_db",
    schedule_interval="@weekly",
    start_date=days_ago(7),
    catchup=False,
    is_paused_upon_creation=False
)

def clean_db_dag_fn():
    t_last=None
    for x in TABLES_TO_CLEAN:
        t=cleanup_db_fn(x)
        if t_last:
            t_last >> t
        t_last = t

clean_db_dag = clean_db_dag_fn()
```

將環境中繼資料匯出到 Amazon S3 上的 CSV 檔案

下列程式碼範例顯示如何建立定向無環圖 (DAG)，以查詢資料庫中的一系列 DAG 執行資訊，並將資料寫入 Amazon S3 上存放的`.csv`檔案。

您可能想要從環境的 Aurora PostgreSQL 資料庫匯出資訊，以便在本機檢查資料、將資料存檔到物件儲存中，或將它們與 Amazon S3 等工具結合到 Amazon Redshift 操作員和資料庫清理，以便將 Amazon MWAA 中繼資料移出環境，但保留它們以供將 future 分析之用。

您可以在資料庫中查詢 Apache 氣流模型中列出的任何物件。此程式碼範例使用三個模型、`DagRun`、和 `TaskFailTaskInstance`，這些模型提供與 DAG 執行相關的資訊。

主題

- 版本
- 先決條件
- 許可
- 需求

- 程式碼範例

版本

- 您可以使用此頁面上的代碼示例與 Apache 氣流 v2 及更高版本在 Python 3.10 中。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 亞馬遜 MWAA 環境。
- 您想要匯出中繼資料資訊的新 Amazon S3 儲存貯體。

許可

Amazon MWAA 需要 Amazon S3 動作的許可，才能`s3:PutObject`將查詢的中繼資料資訊寫入 Amazon S3。將下列原則陳述式新增至環境的執行角色。

```
{
  "Effect": "Allow",
  "Action": "s3:PutObject*",
  "Resource": "arn:aws:s3:::your-new-export-bucket"
}
```

此原則將寫入存取權限限制為僅限*your-new-export-bucket*。

需求

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用 Apache 氣流 v2 基本安裝在您的環境。

程式碼範例

下列步驟說明如何建立可查詢 Aurora PostgreSQL 的 DAG，並將結果寫入新的 Amazon S3 儲存貯體。

1. 在終端機中，導覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並將其儲存為本機metadata_to_csv.py。您可以變更指派給的值，MAX_AGE_IN_DAYS以控制從中繼資料資料庫 DAG 查詢的最舊記錄的保留期限。

```
from airflow.decorators import dag, task
from airflow import settings
import os
import boto3
from airflow.utils.dates import days_ago
from airflow.models import DagRun, TaskFail, TaskInstance
import csv, re
from io import StringIO

DAG_ID = os.path.basename(__file__).replace(".py", "")

MAX_AGE_IN_DAYS = 30
S3_BUCKET = '<your-export-bucket>'
S3_KEY = 'files/export/{0}.csv'

# You can add other objects to export from the metadatabase,
OBJECTS_TO_EXPORT = [
    [DagRun,DagRun.execution_date],
    [TaskFail,TaskFail.execution_date],
    [TaskInstance, TaskInstance.execution_date],
]

@task()
def export_db_task(**kwargs):
    session = settings.Session()
    print("session: ",str(session))

    oldest_date = days_ago(MAX_AGE_IN_DAYS)
    print("oldest_date: ",oldest_date)

    s3 = boto3.client('s3')

    for x in OBJECTS_TO_EXPORT:
        query = session.query(x[0]).filter(x[1] >= days_ago(MAX_AGE_IN_DAYS))
        print("type",type(query))
        allrows=query.all()
        name=re.sub("[<>']", "", str(x[0]))
```

```
        print(name,": ",str(allrows))

        if len(allrows) > 0:
            outfileStr=""
            f = StringIO(outfileStr)
            w = csv.DictWriter(f, vars(allrows[0]).keys())
            w.writeheader()
            for y in allrows:
                w.writerow(vars(y))
            outkey = S3_KEY.format(name[6:])
            s3.put_object(Bucket=S3_BUCKET, Key=outkey, Body=f.getvalue())

@dag(
    dag_id=DAG_ID,
    schedule_interval=None,
    start_date=days_ago(1),
    )
def export_db():
    t = export_db_task()

metadb_to_s3_test = export_db()
```

3. 執行下列AWS CLI命令，將 DAG 複製到您環境的儲存貯體，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果成功，您將在任務的任務日誌中輸出類似以下內export_db容：

```
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - type <class
 'sqlalchemy.orm.query.Query'>
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - class
 airflow.models.dagrun.DagRun : [your-tasks]
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - type <class
 'sqlalchemy.orm.query.Query'>
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - class
 airflow.models.taskfail.TaskFail :  [your-tasks]
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - type <class
 'sqlalchemy.orm.query.Query'>
[2022-01-01, 12:00:00 PDT] {{logging_mixin.py:109}} INFO - class
 airflow.models.taskinstance.TaskInstance :  [your-tasks]
[2022-01-01, 12:00:00 PDT] {{python.py:152}} INFO - Done. Returned value was: OK
```

```
[2022-01-01, 12:00:00 PDT] {{taskinstance.py:1280}} INFO - Marking task as
 SUCCESS. dag_id=metadb_to_s3, task_id=export_db, execution_date=20220101T000000,
 start_date=20220101T000000, end_date=20220101T000000
[2022-01-01, 12:00:00 PDT] {{local_task_job.py:154}} INFO - Task exited with return
 code 0
[2022-01-01, 12:00:00 PDT] {{local_task_job.py:264}} INFO - 0 downstream tasks
 scheduled from follow-on schedule check
```

您現在可以在中存取和下載新 Amazon S3 儲存貯體中匯出的`.csv`檔案`/files/export/`。

使用密鑰AWS Secrets Manager對於一個阿帕奇氣流變量

下列範例呼叫AWS Secrets Manager在亞馬遜管理的 Apache 氣流變量上獲取 Apache 氣流變量的秘密密鑰。它假設您已完成中的步驟使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

主題

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 秘密管理器後端作為 Apache 氣流配置選項，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

- 秘密管理器中的 Apache 氣流變量字符串，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

許可

- 密碼管理員權限，如所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

請求

- 若要將此程式碼範例與 Apache Airflow v1 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V1 基本安裝在您的環境中。

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V2 基地安裝在您的環境中。

程式碼範例

下列步驟說明如何建立呼叫秘密管理員以取得密碼的 DAG 程式碼。

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為`secrets-manager-var.py`。

```
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.models import Variable
from airflow.utils.dates import days_ago
from datetime import timedelta
import os
DAG_ID = os.path.basename(__file__).replace(".py", "")
DEFAULT_ARGS = {
    'owner': 'airflow',
    'depends_on_past': False,
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
```

```
}
def get_variable_fn(**kwargs):
    my_variable_name = Variable.get("test-variable", default_var="undefined")
    print("my_variable_name: ", my_variable_name)
    return my_variable_name
with DAG(
    dag_id=DAG_ID,
    default_args=DEFAULT_ARGS,
    dagrun_timeout=timedelta(hours=2),
    start_date=days_ago(1),
    schedule_interval='@once',
    tags=['variable']
) as dag:
    get_variable = PythonOperator(
        task_id="get_variable",
        python_callable=get_variable_fn,
        provide_context=True
    )
```

後續步驟？

- 了解如何將此範例中的 DAG 程式碼上傳至dags亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。

使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接

下列範例呼叫AWS Secrets Manager在亞馬遜 Apache 氣流管理的工作流程上獲取 Apache 氣流連接的秘密密鑰。它假設您已完成中的步驟使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

主題

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。
- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 秘密管理器後端作為 Apache 氣流配置選項，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。
- 秘密管理器中的 Apache 氣流連接字符串，如圖所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

許可

- 密碼管理員權限，如所示使用AWS Secrets Manager機密來設定 Apache Airflow 連線。

請求

- 若要將此程式碼範例與 Apache Airflow v1 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V1 基本安裝在您的環境中。
- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V2 基地安裝在您的環境中。

程式碼範例

下列步驟說明如何建立呼叫秘密管理員以取得密碼的 DAG 程式碼。

Apache Airflow v2

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為secrets-manager.py。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG, settings, secrets
from airflow.operators.python import PythonOperator
from airflow.utils.dates import days_ago
from airflow.providers.amazon.aws.hooks.base_aws import AwsBaseHook

from datetime import timedelta
import os

### The steps to create this secret key can be found at: https://
docs.aws.amazon.com/mwaa/latest/userguide/connections-secrets-manager.html
sm_secretId_name = 'airflow/connections/myconn'

default_args = {
    'owner': 'airflow',
    'start_date': days_ago(1),
    'depends_on_past': False
}


### Gets the secret myconn from Secrets Manager
def read_from_aws_sm_fn(**kwargs):
    ### set up Secrets Manager
```

```
    hook = AwsBaseHook(client_type='secretsmanager')
    client = hook.get_client_type('secretsmanager')
    response = client.get_secret_value(SecretId=sm_secretId_name)
    myConnSecretString = response["SecretString"]

    return myConnSecretString

### 'os.path.basename(__file__).replace(".py", "")' uses the file name secrets-
manager.py for a DAG ID of secrets-manager
with DAG(
        dag_id=os.path.basename(__file__).replace(".py", ""),
        default_args=default_args,
        dagrun_timeout=timedelta(hours=2),
        start_date=days_ago(1),
        schedule_interval=None
) as dag:
    write_all_to_aws_sm = PythonOperator(
        task_id="read_from_aws_sm",
        python_callable=read_from_aws_sm_fn,
        provide_context=True
    )
```

Apache Airflow v1

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為secrets-manager.py。

```
from airflow import DAG, settings, secrets
from airflow.operators.python_operator import PythonOperator
from airflow.utils.dates import days_ago
from airflow.contrib.hooks.aws_hook import AwsHook

from datetime import timedelta
import os

### The steps to create this secret key can be found at: https://
docs.aws.amazon.com/mwaa/latest/userguide/connections-secrets-manager.html
sm_secretId_name = 'airflow/connections/myconn'
```

```
default_args = {
    'owner': 'airflow',
    'start_date': days_ago(1),
    'depends_on_past': False
}


### Gets the secret myconn from Secrets Manager
def read_from_aws_sm_fn(**kwargs):
    ### set up Secrets Manager
    hook = AwsHook()
    client = hook.get_client_type('secretsmanager')
    response = client.get_secret_value(SecretId=sm_secretId_name)
    myConnSecretString = response["SecretString"]

    return myConnSecretString

### 'os.path.basename(__file__).replace(".py", "")' uses the file name secrets-
manager.py for a DAG ID of secrets-manager
with DAG(
        dag_id=os.path.basename(__file__).replace(".py", ""),
        default_args=default_args,
        dagrun_timeout=timedelta(hours=2),
        start_date=days_ago(1),
        schedule_interval=None
) as dag:
    write_all_to_aws_sm = PythonOperator(
        task_id="read_from_aws_sm",
        python_callable=read_from_aws_sm_fn,
        provide_context=True
    )
```

後續步驟？

- 了解如何將此範例中的 DAG 程式碼上傳至dags您的亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。

使用 Oracle 創建一個自定義插件

以下範例會引導您完成使用 Oracle 為 Amazon MWAA 建立自訂外掛程式的步驟，並且可以與 plugins.zip 檔案中的其他自訂外掛程式和二進位檔案結合使用。

內容

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。
- 工人在任何記錄層級啟用記錄，`CRITICAL`或以上，適用於您的環境。如需 Amazon MWAA 日誌類型以及如何管理日誌群組的詳細資訊，請參閱the section called “檢視氣流記錄”

許可

- 使用此頁面上的程式碼範例不需要其他權限。

請求

若要使用此頁面上的範例程式碼，請將下列相依性新增至`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴。

Apache Airflow v2

```
-c https://raw.githubusercontent.com/apache/airflow/constraints-2.0.2/
constraints-3.7.txt
cx_Oracle
apache-airflow-providers-oracle
```

Apache Airflow v1

```
cx_Oracle==8.1.0
apache-airflow[oracle]==1.10.12
```

程式碼範例

下列步驟說明如何建立將測試自訂外掛程式的 DAG 程式碼。

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為`oracle.py`。

```
from airflow import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.utils.dates import days_ago
import os
import cx_Oracle

DAG_ID = os.path.basename(__file__).replace(".py", "")
```

```
def testHook(**kwargs):
    cx_Oracle.init_oracle_client()
    version = cx_Oracle.clientversion()
    print("cx_Oracle.clientversion",version)
    return version

with DAG(dag_id=DAG_ID, schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    hook_test = PythonOperator(
        task_id="hook_test",
        python_callable=testHook,
        provide_context=True
    )
```

創建自定義插件

本節介紹如何下載依賴關係，創建自定義插件和 plugins.zip。

下載相依性

亞馬遜 MWAA 會將 plugins.zip 的內容提取到/usr/local/airflow/plugins在每個亞馬遜 MWAA 排程器和工作者容器上。這是用來將二進製文件添加到您的環境中。下列步驟說明如何組合自訂外掛程式所需的檔案。

拉亞馬遜 Linux 容器映像

1. 在命令提示字元中，提取 Amazon Linux 容器映像，然後在本機執行容器。例如：

   ```
   docker pull amazonlinux
   docker run -it amazonlinux:latest /bin/bash
   ```

   您的命令提示符應調用 bash 命令行。例如：

   ```
   bash-4.2#
   ```

2. 安裝 Linux 原生的非同步 I/O 設施 (程式碼)。

   ```
   yum -y install libaio
   ```

3. 保持此視窗開啟，以便後續步驟使用。我們將在本地複製以下文件：lib64/libaio.so.1,lib64/libaio.so.1.0.0,lib64/libaio.so.1.0.1。

下載客戶資料夾

1. 在本機安裝解壓縮套件。例如：

```
sudo yum install unzip
```

2. 建立 oracle_plugin 目錄。例如：

```
mkdir oracle_plugin
cd oracle_plugin
```

3. 使用下面的 curl 命令下載instantclient-basic-linux拉鍊.從甲骨文即時客戶端下載。

```
curl https://download.oracle.com/otn_software/linux/instantclient/185000/
instantclient-basic-linux.x64-18.5.0.0.0dbru.zip > client.zip
```

4. 解壓縮 client.zip 檔案。例如：

```
unzip *.zip
```

從碼頭提取文件

1. 在新的命令提示字元中，顯示並記下您的 Docker 容器 ID。例如：

```
docker container ls
```

您的命令提示符應返回所有容器及其 ID。例如：

```
debc16fd6970
```

2. 在您的oracle_plugin目錄，解壓縮lib64/libaio.so.1,lib64/libaio.so.1.0.0,lib64/libaio.so.1.0.1檔案至本機instantclient_18_5資料夾。例如：

```
docker cp debc16fd6970:/lib64/libaio.so.1 instantclient_18_5/
docker cp debc16fd6970:/lib64/libaio.so.1.0.0 instantclient_18_5/
docker cp debc16fd6970:/lib64/libaio.so.1.0.1 instantclient_18_5/
```

自定義插件

阿帕奇氣流將在啟動時執行插件文件夾中的 Python 文件的內容。這是用來設置和修改環境變量。下列步驟說明自訂外掛程式的範例程式碼。

- 複製下列程式碼範例的內容，並在本機儲存為env_var_plugin_oracle.py。

```
from airflow.plugins_manager import AirflowPlugin
import os

os.environ["LD_LIBRARY_PATH"]='/usr/local/airflow/plugins/instantclient_18_5'
os.environ["DPI_DEBUG_LEVEL"]="64"

class EnvVarPlugin(AirflowPlugin):
    name = 'env_var_plugin'
```

Plugins.zip

下列步驟說明如何建立plugins.zip。這個例子的內容可以與您的其他插件和二進製文件組合成一個單一的plugins.zip文件。

壓縮插件目錄的內容

1. 在命令提示字元中，導覽至oracle_plugin目錄。例如：

```
cd oracle_plugin
```

2. 拉鍊instantclient_18_5plugins.zip 中的目錄。例如：

```
zip -r ../plugins.zip ./
```

3. 您應該在命令提示符中看到以下內容：

```
oracle_plugin$ ls
client.zip  instantclient_18_5
```

4. 移除client.zip文件。例如：

```
rm client.zip
```

壓縮 env_var_plugin_oracle.py 文件

1. 添加`env_var_plugin_oracle.py`文件到 plugins.zip 的根目錄下。例如：

```
zip plugins.zip env_var_plugin_oracle.py
```

2. 您的 plugins.zip 現在應該包含以下內容：

```
env_var_plugin_oracle.py
instantclient_18_5/
```

氣流組態選項

如果您使用的是阿帕奇氣流 V2，添加`core.lazy_load_plugins : False`作為 Apache 氣流配置選項。若要深入瞭解，請參閱使用配置選項在 2 中加載插件。

後續步驟？

- 了解如何上傳`requirements.txt`在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝 Python 的依賴。
- 了解如何將此範例中的 DAG 程式碼上傳至`dags`亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。
- 進一步了解如何上傳`plugins.zip`在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝自定義插件。

建立產生執行階段環境變數的自訂外掛程式

以下範例將逐步引導您完成建立自訂外掛程式的步驟，以便在適用於 Apache Airflow 環境的 Amazon 受管工作流程上，在執行時期產生環境變數。

主題

- 版本
- 先決條件
- 許可
- 請求
- 自定義

- Plugins.zip
- 氣流
- 後續步驟？

版本

- 此頁面上的示例代碼可以與 Python 3.7 中的阿帕奇氣流 V1 一起使用。

先決條件

若要使用此頁面上的範例程式碼，您必須準備好以下事項：

- 一個亞馬遜 MWAA 環境。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

請求

- 若要將此程式碼範例與 Apache Airflow v1 搭配使用，不需要額外的相依性。此程式碼會在您的環境中使用 Apache 氣流 v1 基本安裝。

自定義

阿帕奇氣流將在啟動時執行插件文件夾中的 Python 文件的內容。這是用來設置和修改環境變量。下列步驟說明下列步驟。

1. 在命令提示中，導航到外掛程式的目錄。例如：

```
cd plugins
```

2. 複製下列程式碼範例的內容，並在上述資料夾`env_var_plugin.py`中本機儲存為。

```
from airflow.plugins_manager import AirflowPlugin
import os
```

```
os.environ["PATH"] = os.getenv("PATH") + ":/usr/local/airflow/.local/lib/python3.7/
site-packages"
os.environ["JAVA_HOME"]="/usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.272.b10-1.amzn2.0.1.x86_64"

class EnvVarPlugin(AirflowPlugin):
     name = 'env_var_plugin'
```

Plugins.zip

下列步驟顯示如何建立plugins.zip。這個例子的內容可以與其他插件和二進製文件組合成一個plugins.zip文件。

1. 在命令提示中，導覽hive_plugin至上一個步驟。例如：

```
cd plugins
```

2. 壓縮文plugins件夾中的內容。

```
zip -r ../plugins.zip ./
```

氣流

如果您使用的是 Apache 氣流 v2，請添加core.lazy_load_plugins : False為 Apache 氣流配置選項。若要深入瞭解，請參閱使用設定選項載入外掛程式 2。

後續步驟？

- 在中了解如何將此範例中的requirements.txt檔案上傳到您的 Amazon S3 儲存貯體安裝 Python 的依賴。
- 了解如何將此範例中的 DAG 程式碼上傳到的 Amazon S3 儲存貯體中的dags資料夾新增或更新 DAG。
- 在本範例中進一步了解如何將plugins.zip檔案上傳到中的 Amazon S3 儲存貯體安裝自定義插件。

在亞馬遜 MWAA 上更改 DAG 的時區

默認情況下，阿帕奇氣流以 UTC+0 計劃您的定向非環圖（DAG）。以下步驟顯示了如何更改亞馬遜 MWAA 運行 DAG 的時區鐘擺。或者，本主題示範如何建立自訂外掛程式，以變更環境 Apache Airflow 記錄的時區。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

建立外掛程式以變更氣流記錄中的時區

阿帕奇氣流將運行在 Python 文件`plugins`啟動時的目錄。使用以下插件，您可以覆蓋執行程序的時區，該時區修改 Apache Airflow 寫入日誌的時區。

1. 創建一個名為的目錄`plugins`為您的自定義插件，並導航到目錄。例如：

```
$ mkdir plugins
$ cd plugins
```

2. 複製下列程式碼範例的內容，並在本機儲存為dag-timezone-plugin.py在plugins資料夾。

```
import time
import os

os.environ['TZ'] = 'America/Los_Angeles'
time.tzset()
```

3. 在plugins目錄中，創建一個名為的空 Python 文件__init__.py。您的plugins目錄應類似於以下內容：

```
plugins/
|-- __init__.py
|-- dag-timezone-plugin.py
```

建立plugins.zip

以下步驟顯示如何創建plugins.zip。這個例子的內容可以與其他插件和二進製文件組合成一個單一的plugins.zip文件。

1. 在命令提示字元中，導覽至plugins上一步的目錄。例如：

```
cd plugins
```

2. 壓縮您的內容plugins目錄。

```
zip -r ../plugins.zip ./
```

3. 上传plugins.zip到您的 S3 儲存貯體

```
$ aws s3 cp plugins.zip s3://your-mwaa-bucket/
```

程式碼範例

若要變更 DAG 執行的預設時區 (UTC+0)，我們將使用名為的程式庫鐘擺，一個 Python 庫，用於處理時區感知日期時間。

1. 在命令提示字元中，瀏覽至儲存 DAG 的目錄。例如：

```
$ cd dags
```

2. 複製下列範例的內容並另存新檔tz-aware-dag.py。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from datetime import datetime, timedelta
# Import the Pendulum library.
import pendulum

# Instantiate Pendulum and set your timezone.
local_tz = pendulum.timezone("America/Los_Angeles")

with DAG(
    dag_id = "tz_test",
    schedule_interval="0 12 * * *",
    catchup=False,
    start_date=datetime(2022, 1, 1, tzinfo=local_tz)
) as dag:
    bash_operator_task = BashOperator(
        task_id="tz_aware_task",
        dag=dag,
        bash_command="date"
    )
```

3. 運行以下命令AWS CLI命令，將 DAG 複製到您環境的值區，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果成功，您將在任務日誌中輸出類似以下內容tz_aware_task在tz_test天:

```
[2022-08-01, 12:00:00 PDT] {{subprocess.py:74}} INFO - Running command: ['bash', '-
c', 'date']
```

```
[2022-08-01, 12:00:00 PDT] {{subprocess.py:85}} INFO - Output:
[2022-08-01, 12:00:00 PDT] {{subprocess.py:89}} INFO - Mon Aug  1 12:00:00 PDT 2022
[2022-08-01, 12:00:00 PDT] {{subprocess.py:93}} INFO - Command exited with return
 code 0
[2022-08-01, 12:00:00 PDT] {{taskinstance.py:1280}} INFO - Marking task as
 SUCCESS. dag_id=tz_test, task_id=tz_aware_task, execution_date=20220801T190033,
 start_date=20220801T190035, end_date=20220801T190035
[2022-08-01, 12:00:00 PDT] {{local_task_job.py:154}} INFO - Task exited with return
 code 0
[2022-08-01, 12:00:00 PDT] {{local_task_job.py:264}} INFO - 0 downstream tasks
 scheduled from follow-on schedule check
```

後續步驟？

- 進一步了解如何上傳`plugins.zip`在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝自定義插件。

刷新一個 CodeArtifact 代幣

如果您正在使用 CodeArtifact 要安裝 Python 依賴關係，亞馬遜 MWAA 需要一個活動令牌。若要允許亞馬遜 MWAA 存取 CodeArtifact 在運行時存儲庫，您可以使用啟動腳本並設置`PIP_EXTRA_INDEX_URL`與令牌。

下列主題說明如何建立使用`get_authorization_token` CodeArtifact API 操作可在每次環境啟動或更新時檢索新令牌。

主題

- 版本
- 先決條件
- 許可
- 程式碼範例
- 後續步驟？

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。
- 一個CodeArtifact 儲存庫您可以在其中儲存環境的相依性。

許可

若要重新整理 CodeArtifact 權杖並將結果寫入 Amazon S3 Amazon MWAA 必須具有執行角色中的下列許可。

- 該`codeartifact:GetAuthorizationToken`動作允許亞馬遜 MWAA 從中擷取新令牌 CodeArtifact。以下政策授予每個人的權限 CodeArtifact 您建立的網域。您可以修改陳述式中的資源值，並僅指定您希望環境存取的網域，進一步限制對網域的存取。

```
{
  "Effect": "Allow",
  "Action": "codeartifact:GetAuthorizationToken",
  "Resource": "arn:aws:codeartifact:us-west-2:*:domain/*"
}
```

- 該`sts:GetServiceBearerToken`需要採取行動來調用 CodeArtifact `GetAuthorizationToken`API 操作。此操作返回使用包管理器時必須使用的令牌，例如`pip`與 CodeArtifact。若要搭配使用套件管理員 CodeArtifact 儲存庫，您環境的執行角色角色必須允許`sts:GetServiceBearerToken`如以下政策聲明所示。

```
{
  "Sid": "AllowServiceBearerToken",
  "Effect": "Allow",
  "Action": "sts:GetServiceBearerToken",
  "Resource": "*"
}
```

程式碼範例

下列步驟說明如何建立啟動指令碼以更新 CodeArtifact 令牌。

1. 複製下列程式碼範例的內容，並在本機儲存為`code_artifact_startup_script.sh`。

```
#!/bin/sh

# Startup script for MWAA, see https://docs.aws.amazon.com/mwaa/latest/userguide/
using-startup-script.html

set -eu

# setup code artifact endpoint and token
# https://pip.pypa.io/en/stable/cli/pip_install/#cmdoption-0
# https://docs.aws.amazon.com/mwaa/latest/userguide/samples-code-artifact.html
DOMAIN="amazon"
DOMAIN_OWNER="112233445566"
REGION="us-west-2"
REPO_NAME="MyRepo"
echo "Getting token for CodeArtifact with args: --domain $DOMAIN --region $REGION
 --domain-owner $DOMAIN_OWNER"
TOKEN=$(aws codeartifact get-authorization-token --domain $DOMAIN --region $REGION
 --domain-owner $DOMAIN_OWNER | jq -r '.authorizationToken')
echo "Setting Pip env var for '--index-url' to point to CodeArtifact"
export PIP_EXTRA_INDEX_URL="https://aws:$TOKEN@$DOMAIN-
$DOMAIN_OWNER.d.codeartifact.$REGION.amazonaws.com/pypi/$REPO_NAME/simple/"
echo "CodeArtifact startup setup complete"
```

2. 導覽至您儲存指令碼的資料夾。使用cp在新的提示窗口中將腳本上傳到您的存儲桶。取代*## S3-##*與您的信息。

```
$ aws s3 cp code_artifact_startup_script.sh s3://your-s3-bucket/
code_artifact_startup_script.sh
```

如果成功，Amazon S3 會將 URL 路徑輸出至物件：

```
upload: ./code_artifact_startup_script.sh to s3://your-s3-bucket/
code_artifact_startup_script.sh
```

上傳指令碼之後，您的環境會在啟動時更新並執行指令碼。

後續步驟？

- 瞭解如何使用啟動指令碼來自訂您的環境the section called “使用啟動腳本”。

- 了解如何將此範例中的 DAG 程式碼上傳至dags您的亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。
- 進一步了解如何上傳plugins.zip在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝自定義插件。

創建一個自定義插件與阿帕奇蜂巢和 Hadoop

亞馬遜 MWAA 提取的內容plugins.zip至/usr/local/airflow/plugins。這可用於將二進製文件添加到容器中。此外，阿帕奇氣流執行 Python 文件的內容plugins資料夾位於啟動— 可讓您設定和修改環境變數。以下範例將引導您完成在適用於 Apache Airflow 環境的 Amazon 受管工作流程上使用 Apache Hive 和 Hadoop 建立自訂外掛程式的步驟，並可與其他自訂外掛程式和二進位檔案結合使用。

主題

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

請求

若要使用此頁面上的範例程式碼，請將下列相依性新增至`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴。

Apache Airflow v2

```
-c https://raw.githubusercontent.com/apache/airflow/constraints-2.0.2/
constraints-3.7.txt
apache-airflow-providers-amazon[apache.hive]
```

Apache Airflow v1

```
apache-airflow[hive]==1.10.12
```

下載相依性

亞馬遜 MWAA 會將 plugins.zip 的內容提取到`/usr/local/airflow/plugins`在每個亞馬遜 MWAA 排程器和工作者容器上。這是用來將二進製文件添加到您的環境。下列步驟說明如何組合自訂外掛程式所需的檔案。

1. 在命令提示符中，導航到要創建插件的目錄。例如：

   ```
   cd plugins
   ```

2. 下載Hadoop從一個鏡子，例如：

   ```
   wget https://downloads.apache.org/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz
   ```

3. 下載蜂巢從一個鏡子，例如：

```
wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
```

4. 建立目錄。例如：

```
mkdir hive_plugin
```

5. 提取哈達。

```
tar -xvzf hadoop-3.3.0.tar.gz -C hive_plugin
```

6. 提取蜂巢。

```
tar -xvzf apache-hive-3.1.2-bin.tar.gz -C hive_plugin
```

自定義插件

阿帕奇氣流將在啟動時執行插件文件夾中的 Python 文件的內容。這是用來設置和修改環境變量。下列步驟說明自訂外掛程式的範例程式碼。

1. 在命令提示字元中，導覽至hive_plugin目錄。例如：

```
cd hive_plugin
```

2. 複製下列程式碼範例的內容，並在本機儲存為hive_plugin.py在hive_plugin目錄。

```
from airflow.plugins_manager import AirflowPlugin
import os
os.environ["JAVA_HOME"]="/usr/lib/jvm/jre"
os.environ["HADOOP_HOME"]='/usr/local/airflow/plugins/hadoop-3.3.0'
os.environ["HADOOP_CONF_DIR"]='/usr/local/airflow/plugins/hadoop-3.3.0/etc/hadoop'
os.environ["HIVE_HOME"]='/usr/local/airflow/plugins/apache-hive-3.1.2-bin'
os.environ["PATH"] = os.getenv("PATH") + ":/usr/local/airflow/plugins/
hadoop-3.3.0:/usr/local/airflow/plugins/apache-hive-3.1.2-bin/bin:/usr/local/
airflow/plugins/apache-hive-3.1.2-bin/lib"
os.environ["CLASSPATH"] = os.getenv("CLASSPATH") + ":/usr/local/airflow/plugins/
apache-hive-3.1.2-bin/lib"
class EnvVarPlugin(AirflowPlugin):
    name = 'hive_plugin'
```

3. 應付以下文本的內容並在本地保存為.airflowignore在hive_plugin目錄。

```
hadoop-3.3.0
apache-hive-3.1.2-bin
```

Plugins.zip

以下步驟顯示如何創建plugins.zip。這個例子的內容可以與其他插件和二進製文件組合成一個單一的plugins.zip文件。

1. 在命令提示字元中，導覽至hive_plugin上一步的目錄。例如：

```
cd hive_plugin
```

2. 壓縮您的內容plugins資料夾。

```
zip -r ../hive_plugin.zip ./
```

程式碼範例

下列步驟說明如何建立將測試自訂外掛程式的 DAG 程式碼。

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為hive.py。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago

with DAG(dag_id="hive_test_dag", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    hive_test = BashOperator(
        task_id="hive_test",
        bash_command='hive --help'
    )
```

氣流組態選項

如果您使用的是阿帕奇氣流 V2，添加`core.lazy_load_plugins : False`作為 Apache 氣流配置選項。若要深入瞭解，請參閱使用配置選項加載插件 2。

後續步驟？

- 了解如何上傳`requirements.txt`在此示例中將文件文件到您的亞馬遜 S3 存儲桶中安裝 Python 的依賴。
- 了解如何將此範例中的 DAG 程式碼上傳至`dags`亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。
- 進一步了解如何上傳`plugins.zip`在此示例中將文件文件到您的亞馬遜 S3 存儲桶中安裝自定義插件。

為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator

以下範例顯示如何修補 Apache 氣流PythonVirtualenvOperator在亞馬遜的 Apache 氣流管理工作流程上使用自定義插件。

主題

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。

許可

- 使用此頁面上的程式碼範例不需要其他權限。

請求

若要使用此頁面上的範例程式碼，請將下列相依性新增至`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴。

```
virtualenv
```

自定義插件示例代碼

阿帕奇氣流將在啟動時執行插件文件夾中的 Python 文件的內容。這個插件將修補內置`PythonVirtualenvOperator`在該啟動過程中，以使其與亞馬遜 MWAA 兼容。下列步驟顯示自訂外掛程式的範例程式碼。

Apache Airflow v2

1. 在命令提示字元中，導覽至`plugins`上面的目錄。例如：

   ```
   cd plugins
   ```

2. 複製下列程式碼範例的內容，並在本機儲存為`virtual_python_plugin.py`。

   ```
   """
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

   Permission is hereby granted, free of charge, to any person obtaining a copy of
   this software and associated documentation files (the "Software"), to deal in
   the Software without restriction, including without limitation the rights to
   use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
   the Software, and to permit persons to whom the Software is furnished to do so.
   ```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow.plugins_manager import AirflowPlugin
import airflow.utils.python_virtualenv
from typing import List

def _generate_virtualenv_cmd(tmp_dir: str, python_bin: str,
 system_site_packages: bool) -> List[str]:
    cmd = ['python3','/usr/local/airflow/.local/lib/python3.7/site-packages/
virtualenv', tmp_dir]
    if system_site_packages:
        cmd.append('--system-site-packages')
    if python_bin is not None:
        cmd.append(f'--python={python_bin}')
    return cmd

airflow.utils.python_virtualenv._generate_virtualenv_cmd=_generate_virtualenv_cmd

class VirtualPythonPlugin(AirflowPlugin):
    name = 'virtual_python_plugin'
```

Apache Airflow v1

1. 在命令提示字元中，導覽至plugins上面的目錄。例如：

```
cd plugins
```

2. 複製下列程式碼範例的內容，並在本機儲存為virtual_python_plugin.py。

```
from airflow.plugins_manager import AirflowPlugin
from airflow.operators.python_operator import PythonVirtualenvOperator

def _generate_virtualenv_cmd(self, tmp_dir):
    cmd = ['python3','/usr/local/airflow/.local/lib/python3.7/site-packages/
virtualenv', tmp_dir]
    if self.system_site_packages:
```

```
        cmd.append('--system-site-packages')
    if self.python_version is not None:
        cmd.append('--python=python{}'.format(self.python_version))
    return cmd
PythonVirtualenvOperator._generate_virtualenv_cmd=_generate_virtualenv_cmd

class EnvVarPlugin(AirflowPlugin):
    name = 'virtual_python_plugin'
```

Plugins.zip

下列步驟說明如何建立plugins.zip。

1. 在命令提示符中，導航到包含的目錄virtual_python_plugin.py以上。例如：

```
cd plugins
```

2. 壓縮您的內容plugins資料夾。

```
zip plugins.zip virtual_python_plugin.py
```

程式碼範例

下列步驟說明如何建立自訂外掛程式的 DAG 程式碼。

Apache Airflow v2

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為virtualenv_test.py。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
```

```
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG
from airflow.operators.python import PythonVirtualenvOperator
from airflow.utils.dates import days_ago
import os

os.environ["PATH"] = os.getenv("PATH") + ":/usr/local/airflow/.local/bin"

def virtualenv_fn():
    import boto3
    print("boto3 version ",boto3.__version__)

with DAG(dag_id="virtualenv_test", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    virtualenv_task = PythonVirtualenvOperator(
        task_id="virtualenv_task",
        python_callable=virtualenv_fn,
        requirements=["boto3>=1.17.43"],
        system_site_packages=False,
        dag=dag,
    )
```

Apache Airflow v1

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為`virtualenv_test.py`。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG
from airflow.operators.python_operator import PythonVirtualenvOperator
from airflow.utils.dates import days_ago
import os

os.environ["PATH"] = os.getenv("PATH") + ":/usr/local/airflow/.local/bin"

def virtualenv_fn():
    import boto3
    print("boto3 version ",boto3.__version__)

with DAG(dag_id="virtualenv_test", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    virtualenv_task = PythonVirtualenvOperator(
        task_id="virtualenv_task",
        python_callable=virtualenv_fn,
        requirements=["boto3>=1.17.43"],
        system_site_packages=False,
        dag=dag,
    )
```

氣流組態選項

如果您使用的是阿帕奇氣流 V2，添加`core.lazy_load_plugins : False`作為一個 Apache 氣流配置選項。若要深入瞭解，請參閱使用配置選項在 2 中加載插件。

後續步驟？

- 了解如何上傳`requirements.txt`在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝 Python 的依賴。
- 了解如何將此範例中的 DAG 程式碼上傳至`dags`您的亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。
- 進一步了解如何上傳`plugins.zip`在這個例子中將文件文件到您的亞馬遜 S3 存儲桶安裝自定義插件。

使用拉姆達函數調用 DAG

下列程式碼範例使用AWS Lambda用於獲取 Apache 氣流 CLI 令牌並在亞馬遜 MWAA 環境中調用有向無環圖（DAG）的函數。

主題

- 版本
- 先決條件
- 許可
- 相依性
- 程式碼範例

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此程式碼範例，您必須：

- 使用公用網路存取模式為您的亞馬遜 MWAA 環境。
- 有一個拉姆達函數使用最新的 Python 運行時間。

> **Note**
>
> 如果 Lambda 函數和您的 Amazon MWAA 環境位於相同的 VPC 中，您可以在私有網路上使用此程式碼。對於此組態，Lambda 函數的執行角色需要有權限才能呼叫亞馬遜彈性運算雲端 (Amazon EC2)CreateNetworkInterfaceAPI 操作。您可以使用AWSLambdaVPCAccessExecutionRole AWS受管理的策略。

許可

若要使用此頁面上的程式碼範例，Amazon MWAA 環境的執行角色需要存取權才能執行airflow:CreateCliToken動作。您可以使用AmazonMWAAAirflowCliAccess AWS受管理的策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "airflow:CreateCliToken"
            ],
            "Resource": "*"
        }
    ]
```

如需詳細資訊，請參閱阿帕奇氣流 CLI 政策:亞馬遜 AirflowCliAccess。

相依性

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V2 基地安裝在您的環境中。

程式碼範例

1. 在 https://console.aws.amazon.com/lambda/ 開啟 AWS Lambda 主控台。
2. 從中選擇您的 Lambda 函數函数列表。
3. 在功能頁面上，複製下列程式碼，並將下列程式碼取代為您的資源名稱：

- YOUR_ENVIRONMENT_NAME— 您的亞馬遜 MWAA 環境的名稱。
- YOUR_DAG_NAME— 您要叫用的 DAG 的名稱。

```
import boto3
import http.client
import base64
import ast
mwaa_env_name = 'YOUR_ENVIRONMENT_NAME'
dag_name = 'YOUR_DAG_NAME'
mwaa_cli_command = 'dags trigger'

client = boto3.client('mwaa')

def lambda_handler(event, context):
    # get web token
    mwaa_cli_token = client.create_cli_token(
        Name=mwaa_env_name
    )

    conn = http.client.HTTPSConnection(mwaa_cli_token['WebServerHostname'])
    payload = mwaa_cli_command + " " + dag_name
    headers = {
      'Authorization': 'Bearer ' + mwaa_cli_token['CliToken'],
      'Content-Type': 'text/plain'
    }
    conn.request("POST", "/aws_mwaa/cli", payload, headers)
    res = conn.getresponse()
    data = res.read()
    dict_str = data.decode("UTF-8")
    mydata = ast.literal_eval(dict_str)
    return base64.b64decode(mydata['stdout'])
```

4. 選擇 部署 。
5. 選擇測試以使用 Lambda 主控台叫用您的函數。
6. 若要確認您的 Lambda 是否成功叫用 DAG，請使用 Amazon MWAA 主控台導覽至環境的 Apache 氣流使用者介面，然後執行下列動作：

   a. 在「」DAG頁面上的 DAG 清單中尋找新的目標 DAG。

   b. 下上次執行，檢查最新 DAG 執行的時間戳記。此時間戳記應與最新的時間戳記密切相符invoke_dag在您的其他環境中。

c. 下最近的工作，檢查上次執行是否成功。

在不同的亞馬遜 MWAA 環境中叫用 DAG

下列程式碼範例會建立 Apache 氣流 CLI 權杖。然後，程式碼會在一個 Amazon MWAA 環境中使用有向無環圖 (DAG)，在不同的 Amazon MWAA 環境中叫用 DAG。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的程式碼範例，您需要下列項目：

- 二亞馬遜 MWAA 環境與公眾網Web 伺服器存取，包括您目前的環境。
- 上傳到目標環境的亞馬遜簡單儲存服務 (Amazon S3) 儲存貯體的範例 DAG。

許可

若要使用此頁面上的程式碼範例，您環境的執行角色必須具有建立 Apache Airflow CLI 權杖的權限。您可以附加AWS受管理政策`AmazonMWAAAirflowCliAccess`授予此權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "airflow:CreateCliToken"
            ],
            "Resource": "*"
        }
    ]
```

如需詳細資訊，請參閱阿帕奇氣流 CLI 政策:亞馬遜 AirflowCliAccess。

相依性

- 若要將此程式碼範例與 Apache Airflow v2 搭配使用，不需要額外的相依性。該代碼使用阿帕奇氣流 V2 基地安裝在您的環境中。

程式碼範例

下列程式碼範例假設您在目前環境中使用 DAG 來叫用另一個環境中的 DAG。

1. 在終端機中，導覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並將其儲存為本機`invoke_dag.py`。將下列值取代為您的資訊。

   - `your-new-environment-name`— 您要呼叫 DAG 之其他環境的名稱。
   - `your-target-dag-id`— 您要呼叫的其他環境中 DAG 的識別碼。

```
from airflow.decorators import dag, task
import boto3
from datetime import datetime, timedelta
import os, requests

DAG_ID = os.path.basename(__file__).replace(".py", "")

@task()
def invoke_dag_task(**kwargs):
    client = boto3.client('mwaa')
    token = client.create_cli_token(Name='your-new-environment-name')
    url = f"https://{token['WebServerHostname']}/aws_mwaa/cli"
    body = 'dags trigger your-target-dag-id'
    headers = {
```

```
        'Authorization' : 'Bearer ' + token['CliToken'],
        'Content-Type': 'text/plain'
        }
    requests.post(url, data=body, headers=headers)

@dag(
    dag_id=DAG_ID,
    schedule_interval=None,
    start_date=datetime(2022, 1, 1),
    dagrun_timeout=timedelta(minutes=60),
    catchup=False
    )
def invoke_dag():
    t = invoke_dag_task()

invoke_dag_test = invoke_dag()
```

3. 運行以下AWS CLI命令，將 DAG 複製到您環境的值區，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果 DAG 成功執行，您會在工作記錄中看到類似下列的輸出invoke_dag_task。

```
[2022-01-01, 12:00:00 PDT] {{python.py:152}} INFO - Done. Returned value was: None
[2022-01-01, 12:00:00 PDT] {{taskinstance.py:1280}} INFO - Marking task as SUCCESS.
 dag_id=invoke_dag, task_id=invoke_dag_task, execution_date=20220101T120000,
 start_date=20220101T120000, end_date=20220101T120000
[2022-01-01, 12:00:00 PDT] {{local_task_job.py:154}} INFO - Task exited with return
 code 0
[2022-01-01, 12:00:00 PDT] {{local_task_job.py:264}} INFO - 0 downstream tasks
 scheduled from follow-on schedule check
```

若要確認您的 DAG 是否已成功呼叫，請瀏覽至新環境的 Apache Airflow UI，然後執行下列動作：

a. 在「」DAG頁面上的 DAG 清單中尋找新的目標 DAG。

b. 下上次執行，檢查最新 DAG 執行的時間戳記。此時間戳記應與最新的時間戳記密切相符invoke_dag在您的其他環境中。

c. 下最近的工作，檢查上次執行是否成功。

使用亞馬遜 MWAA 與亞馬遜 RDS 微軟 SQL 服務器

您可以使用亞馬遜管理的 Apache 氣流工作流程來連接到適用於 SQL 伺服器的 RDS。下列範例程式碼使用亞馬遜受管工作流程上的 DAG，以便在適用於微軟 SQL 伺服器的亞馬遜 RDS 上連接和執行查詢。

主題

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。
- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。
- 亞馬遜 MWAA 和適用於 SQL 服務器的 RDS 在同一個亞馬遜 VPC 中運行/
- Amazon MWAA 和伺服器的虛擬私人雲端安全群組使用下列連線進行設定：
  - 連接埠的輸入規則1433在亞馬遜 MWAA 的安全組中為亞馬遜 RDS 開放
  - 或端口的出站規則1433從亞馬遜 MWAA 開放到 RDS
- 適用於 SQL 伺服器 RDS 的 Apache 氣流連線會反映在先前程序中建立的 Amazon RDS SQL 伺服器資料庫的主機名稱、連接埠、使用者名稱和密碼。

相依性

要使用本節中的示例代碼，請將以下依賴項添加到您的`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴

Apache Airflow v2

```
apache-airflow-providers-microsoft-mssql==1.0.1
apache-airflow-providers-odbc==1.0.1
pymssql==2.2.1
```

Apache Airflow v1

```
apache-airflow[mssql]==1.10.12
```

阿帕奇氣流 V2 連接

如果您在 Apache Airflow v2 中使用連線，請確定氣流連線物件包含下列索引鍵值組：

1. 連接埠識別碼：默認值
2. 連接器類型：亞馬遜網絡服務
3. 主持人： YOUR_DB_HOST
4. 綱要：
5. 登入:管理
6. 密碼：
7. 連接埠：1433
8. 額外：

程式碼範例

1. 在命令提示字元中，瀏覽至儲存 DAG 程式碼的目錄。例如：

   ```
   cd dags
   ```

2. 複製下列程式碼範例的內容，並在本機儲存為`sql-server.py`。

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
import pymssql
import logging
import sys
from airflow import DAG
from datetime import datetime
from airflow.operators.mssql_operator import MsSqlOperator
from airflow.operators.python_operator import PythonOperator

default_args = {
    'owner': 'aws',
    'depends_on_past': False,
    'start_date': datetime(2019, 2, 20),
    'provide_context': True
}

dag = DAG(
    'mssql_conn_example', default_args=default_args, schedule_interval=None)

drop_db = MsSqlOperator(
   task_id="drop_db",
   sql="DROP DATABASE IF EXISTS testdb;",
   mssql_conn_id="mssql_default",
   autocommit=True,
   dag=dag
)

create_db = MsSqlOperator(
   task_id="create_db",
   sql="create database testdb;",
```

```
   mssql_conn_id="mssql_default",
   autocommit=True,
   dag=dag
)

create_table = MsSqlOperator(
   task_id="create_table",
   sql="CREATE TABLE testdb.dbo.pet (name VARCHAR(20), owner VARCHAR(20));",
   mssql_conn_id="mssql_default",
   autocommit=True,
   dag=dag
)

insert_into_table = MsSqlOperator(
   task_id="insert_into_table",
   sql="INSERT INTO testdb.dbo.pet VALUES ('Olaf', 'Disney');",
   mssql_conn_id="mssql_default",
   autocommit=True,
   dag=dag
)

def select_pet(**kwargs):
   try:
        conn = pymssql.connect(
            server='sampledb.<xxxxxx>.<region>.rds.amazonaws.com',
            user='admin',
            password='<yoursupersecretpassword>',
            database='testdb'
        )

        # Create a cursor from the connection
        cursor = conn.cursor()
        cursor.execute("SELECT * from testdb.dbo.pet")
        row = cursor.fetchone()

        if row:
            print(row)
   except:
      logging.error("Error when creating pymssql database connection: %s",
 sys.exc_info()[0])

select_query = PythonOperator(
    task_id='select_query',
    python_callable=select_pet,
```

```
    dag=dag,
)

drop_db >> create_db >> create_table >> insert_into_table >> select_query
```

後續步驟？

- 了解如何上傳`requirements.txt`在此示例中將文件文件到您的亞馬遜 S3 存儲桶中安裝 Python 的依賴。
- 了解如何將此範例中的 DAG 程式碼上傳至`dags`亞馬遜 S3 存儲桶中的文件夾新增或更新 DAG。
- 探索範例指令碼和其他模塊示例。
- 進一步了解如何在特定微軟 SQL 資料庫中執行 SQL 程式碼微型運算子在阿帕奇氣流參考指南。

與 Amazon EMR 搭配 Amazon EMR 與 Amazon EMR 搭配 Amazon EMR R

下列程式碼範例示範如何使用 Amazon EMR 和 Amazon 管理的 Apache 氣流工作流程啟用整合。

主題

- 版本
- 程式碼範本

版本

- 此頁面上的示例代碼可以與 Python 3.7 中的阿帕奇氣流 V1 一起使用。

程式碼範本

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
```

```
    the Software, and to permit persons to whom the Software is furnished to do so.

    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
    IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
    FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
    COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
    IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
    CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
    """
    from airflow import DAG

    from airflow.contrib.operators.emr_add_steps_operator import EmrAddStepsOperator
    from airflow.contrib.operators.emr_create_job_flow_operator import
EmrCreateJobFlowOperator
    from airflow.contrib.sensors.emr_step_sensor import EmrStepSensor

    from airflow.utils.dates import days_ago
    from datetime import timedelta
    import os

    DAG_ID = os.path.basename(__file__).replace(".py", "")

    DEFAULT_ARGS = {
        'owner': 'airflow',
        'depends_on_past': False,
        'email': ['airflow@example.com'],
        'email_on_failure': False,
        'email_on_retry': False,
    }

    SPARK_STEPS = [
        {
            'Name': 'calculate_pi',
            'ActionOnFailure': 'CONTINUE',
            'HadoopJarStep': {
                'Jar': 'command-runner.jar',
                'Args': ['/usr/lib/spark/bin/run-example', 'SparkPi', '10'],
            },
        }
    ]

    JOB_FLOW_OVERRIDES = {
        'Name': 'my-demo-cluster',
        'ReleaseLabel': 'emr-5.30.1',
```

```
        'Applications': [
            {
                'Name': 'Spark'
            },
        ],
        'Instances': {
            'InstanceGroups': [
                {
                    'Name': "Master nodes",
                    'Market': 'ON_DEMAND',
                    'InstanceRole': 'MASTER',
                    'InstanceType': 'm5.xlarge',
                    'InstanceCount': 1,
                },
                {
                    'Name': "Slave nodes",
                    'Market': 'ON_DEMAND',
                    'InstanceRole': 'CORE',
                    'InstanceType': 'm5.xlarge',
                    'InstanceCount': 2,
                }
            ],
            'KeepJobFlowAliveWhenNoSteps': False,
            'TerminationProtected': False,
            'Ec2KeyName': 'mykeypair',
        },
        'VisibleToAllUsers': True,
        'JobFlowRole': 'EMR_EC2_DefaultRole',
        'ServiceRole': 'EMR_DefaultRole'
    }

    with DAG(
        dag_id=DAG_ID,
        default_args=DEFAULT_ARGS,
        dagrun_timeout=timedelta(hours=2),
        start_date=days_ago(1),
        schedule_interval='@once',
        tags=['emr'],
    ) as dag:

        cluster_creator = EmrCreateJobFlowOperator(
            task_id='create_job_flow',
            job_flow_overrides=JOB_FLOW_OVERRIDES
        )
```

```
        step_adder = EmrAddStepsOperator(
            task_id='add_steps',
            job_flow_id="{{ task_instance.xcom_pull(task_ids='create_job_flow',
key='return_value') }}",
            aws_conn_id='aws_default',
            steps=SPARK_STEPS,
        )

        step_checker = EmrStepSensor(
            task_id='watch_step',
            job_flow_id="{{ task_instance.xcom_pull('create_job_flow',
key='return_value') }}",
            step_id="{{ task_instance.xcom_pull(task_ids='add_steps',
key='return_value')[0] }}",
            aws_conn_id='aws_default',
        )

        cluster_creator >> step_adder >> step_checker
```

使用亞馬遜 MWAA 與亞馬遜 EKS

以下範例示範如何使用亞馬遜受管工作流程搭配 Amazon EKS 進行 Apache 氣流。

主題

- 版本
- 先決條件
- 為亞馬遜 EC2 創建一個公鑰
- 建立叢集
- 創建一個mwaa命名空間
- 建立角色mwaa命名空間
- 為亞馬遜 EKS 叢集建立和附加 IAM 角色
- 建立 requirements.txt 檔案
- 為亞馬遜 EKS 創建身份映射
- 建立 kubeconfig
- 建立一個 DAG
- 新增 DAG 和kube_config.yaml到亞馬遜 S3 桶

- 啟用並觸發範例

版本

- 此頁面上的範例程式碼可搭配使用阿帕奇氣流 V1在蟒蛇 3.7。
- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用本主題中的範例，您需要下列項目：

- 一個亞馬遜 MWAA 環境。
- 埃克特爾. 若要深入瞭解，請參閱安裝外掛。
- 庫貝克特爾。若要深入瞭解，請參閱安裝和設置設置。在某些情況下，這是與 eksctl 一起安裝的。
- 位於您建立 Amazon MWAA 環境的區域中的 EC2 金鑰配對。若要深入瞭解，請參閱建立或匯入金鑰配對。

> **Note**
>
> 當您使用`eksctl`指令，您可以包括`--profile`以指定預設值以外的紀要。

為亞馬遜 EC2 創建一個公鑰

使用下面的命令從您的私鑰對創建一個公鑰。

```
ssh-keygen -y -f myprivatekey.pem > mypublickey.pub
```

若要深入瞭解，請參閱擷取金鑰組的公開金鑰。

建立叢集

使用下列指令建立叢集。如果您想要叢集的自訂名稱，或在不同的區域中建立名稱，請取代名稱和 Region 值。您必須在建立 Amazon MWAA 環境的相同區域中建立叢集。取代子網路的值，以符合您用於 Amazon MWAA 的 Amazon 虛擬私人雲端網路中的子網路。取代的值`ssh-public-key`匹配您

使用的密鑰。您可以使用位於相同區域的 Amazon EC2 現有金鑰，或在建立 Amazon MWAA 環境的相同區域中建立新金鑰。

```
eksctl create cluster \
--name mwaa-eks \
--region us-west-2 \
--version 1.18 \
--nodegroup-name linux-nodes \
--nodes 3 \
--nodes-min 1 \
--nodes-max 4 \
--with-oidc \
--ssh-access \
--ssh-public-key MyPublicKey \
--managed \
--vpc-public-subnets "subnet-11111111111111111, subnet-2222222222222222" \
--vpc-private-subnets "subnet-33333333333333333, subnet-44444444444444444"
```

完成叢集建立需要一些時間。完成後，您可以使用下列命令驗證叢集是否已成功建立，並設定 IAM OIDC 提供者：

```
eksctl utils associate-iam-oidc-provider \
--region us-west-2 \
--cluster mwaa-eks \
--approve
```

創建一個mwaa命名空間

確認已成功建立叢集之後，請使用下列命令為網繭建立命名空間。

```
kubectl create namespace mwaa
```

建立角色mwaa命名空間

建立命名空間後，請為可在 MWAA 命名空間中執行網繭的 EKS 上的 Amazon MWAA 使用者建立角色和角色繫結。如果您為命名空間使用了不同的名稱，請取代 mwaa-n *mwaa*使用您使用的名稱。

```
cat << EOF | kubectl apply -f - -n mwaa
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
```

```
  name: mwaa-role
rules:
  - apiGroups:
      - ""
      - "apps"
      - "batch"
      - "extensions"
    resources:
      - "jobs"
      - "pods"
      - "pods/attach"
      - "pods/exec"
      - "pods/log"
      - "pods/portforward"
      - "secrets"
      - "services"
    verbs:
      - "create"
      - "delete"
      - "describe"
      - "get"
      - "list"
      - "patch"
      - "update"
---
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: mwaa-role-binding
subjects:
- kind: User
  name: mwaa-service
roleRef:
  kind: Role
  name: mwaa-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

執行下列命令，確認新角色可存取 Amazon EKS 叢集。如果沒有使用，請務必使用正確的名稱*Mwaa*:

```
kubectl get pods -n mwaa --as mwaa-service
```

您應該會看到傳回的訊息，上面寫著：

```
No resources found in mwaa namespace.
```

為亞馬遜 EKS 叢集建立和附加 IAM 角色

您必須建立 IAM 角色，然後將其繫結至 Amazon EKS (k8s) 叢集，以便透過 IAM 將其用於身分驗證。角色僅用於登入叢集，並且沒有任何主控台或 API 呼叫的權限。

使用以下步驟為 Amazon MWAA 環境建立新角色Amazon MWAA 執行角色。不過，請不要建立並附加該主題中描述的原則，而是附加下列原則：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "airflow:PublishMetrics",
            "Resource": "arn:aws:airflow:${MWAA_REGION}:${ACCOUNT_NUMBER}:environment/
${MWAA_ENV_NAME}"
        },
        {
            "Effect": "Deny",
            "Action": "s3:ListAllMyBuckets",
            "Resource": [
                "arn:aws:s3:::{MWAA_S3_BUCKET}",
                "arn:aws:s3:::{MWAA_S3_BUCKET}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:GetBucket*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::{MWAA_S3_BUCKET}",
                "arn:aws:s3:::{MWAA_S3_BUCKET}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "logs:GetLogRecord",
                "logs:GetLogGroupFields",
                "logs:GetQueryResults",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:${MWAA_REGION}:${ACCOUNT_NUMBER}:log-group:airflow-
${MWAA_ENV_NAME}-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "sqs:ChangeMessageVisibility",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:${MWAA_REGION}:*:airflow-celery-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey*",
                "kms:Encrypt"
            ],
            "NotResource": "arn:aws:kms:*:${ACCOUNT_NUMBER}:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "sqs.${MWAA_REGION}.amazonaws.com"
```

```
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "eks:DescribeCluster"
            ],
            "Resource": "arn:aws:eks:${MWAA_REGION}:${ACCOUNT_NUMBER}:cluster/
${EKS_CLUSTER_NAME}"
        }
    ]
}
```

建立角色後，請編輯 Amazon MWAA 環境，以使用您建立的角色做為環境的執行角色。若要變更角色，請編輯要使用的環境。您選取下方的執行角色權限。

已知問題：

- 子路徑無法透過 Amazon EKS 進行驗證的角色 ARN 存在已知問題。解決方法是手動建立服務角色，而不是使用 Amazon MWAA 本身建立的服務角色。若要深入瞭解，請參閱當路徑包含在 aws-auth 配置映射中的 ARN 中時，具有路徑的角色不起作用
- 如果 IAM 中無法使用 Amazon MWAA 服務清單，您需要選擇替代服務政策 (例如 Amazon EC2)，然後更新角色的信任政策以符合下列項目：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "airflow-env.amazonaws.com",
          "airflow.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

若要深入瞭解，請參閱如何搭配 IAM 角色使用信任政策。

建立 requirements.txt 檔案

若要使用本節中的範例程式碼，請確定您已將下列其中一個資料庫選項新增至`requirements.txt`。如需進一步了解，請參閱 安裝 Python 的依賴。

Apache Airflow v2

```
kubernetes
apache-airflow[cncf.kubernetes]==3.0.0
```

Apache Airflow v1

```
awscli
kubernetes==12.0.1
```

為亞馬遜 EKS 創建身份映射

使用 ARN 作為您在以下命令中建立的角色，為 Amazon EKS 建立身分對應。變更地區######到您建立環境的區域。替換角色的 ARN，最後更換*mwaa-execution-role*與您環境的執行角色。

```
eksctl create iamidentitymapping \
--region your-region \
--cluster mwaa-eks \
--arn arn:aws:iam::111222333444:role/mwaa-execution-role \
--username mwaa-service
```

建立 **kubeconfig**

使用下面的命令來創建kubeconfig:

```
aws eks update-kubeconfig \
--region us-west-2 \
--kubeconfig ./kube_config.yaml \
--name mwaa-eks \
--alias aws
```

如果您在跑步時使用了特定的配置文件update-kubeconfig你需要刪除env:區段已新增至 kube_config.yaml 檔案，以便與亞馬遜 MWAA 正常運作。若要這麼做，請從檔案中刪除下列項目，然後儲存：

```
env:
- name: AWS_PROFILE
  value: profile_name
```

建立一個 DAG

使用下面的代碼示例來創建一個 Python 文件，如mwaa_pod_example.py對於 DAG。

Apache Airflow v2

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG
from datetime import datetime
from airflow.providers.cncf.kubernetes.operators.kubernetes_pod import
 KubernetesPodOperator

default_args = {
   'owner': 'aws',
   'depends_on_past': False,
   'start_date': datetime(2019, 2, 20),
   'provide_context': True
}

dag = DAG(
   'kubernetes_pod_example', default_args=default_args, schedule_interval=None)
```

```
#use a kube_config stored in s3 dags folder for now
kube_config_path = '/usr/local/airflow/dags/kube_config.yaml'

podRun = KubernetesPodOperator(
                        namespace="mwaa",
                        image="ubuntu:18.04",
                        cmds=["bash"],
                        arguments=["-c", "ls"],
                        labels={"foo": "bar"},
                        name="mwaa-pod-test",
                        task_id="pod-task",
                        get_logs=True,
                        dag=dag,
                        is_delete_operator_pod=False,
                        config_file=kube_config_path,
                        in_cluster=False,
                        cluster_context='aws'
                        )
```

Apache Airflow v1

```
"""
Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
"""
from airflow import DAG
from datetime import datetime
from airflow.contrib.operators.kubernetes_pod_operator import KubernetesPodOperator

default_args = {
   'owner': 'aws',
   'depends_on_past': False,
```

```
    'start_date': datetime(2019, 2, 20),
    'provide_context': True
}

dag = DAG(
    'kubernetes_pod_example', default_args=default_args, schedule_interval=None)

#use a kube_config stored in s3 dags folder for now
kube_config_path = '/usr/local/airflow/dags/kube_config.yaml'

podRun = KubernetesPodOperator(
                        namespace="mwaa",
                        image="ubuntu:18.04",
                        cmds=["bash"],
                        arguments=["-c", "ls"],
                        labels={"foo": "bar"},
                        name="mwaa-pod-test",
                        task_id="pod-task",
                        get_logs=True,
                        dag=dag,
                        is_delete_operator_pod=False,
                        config_file=kube_config_path,
                        in_cluster=False,
                        cluster_context='aws'
                        )
```

新增 DAG 和**kube_config.yaml**到亞馬遜 S3 桶

放置您創建的 DAG 和kube_config.yaml歸檔到亞馬遜 S3 存儲桶的亞馬遜 MWAA 環境。您可以使用 Amazon S3 主控台或AWS Command Line Interface。

啟用並觸發範例

在 Apache 氣流中，啟用範例，然後觸發它。

順利執行並完成之後，請使用下列命令來驗證網繭：

```
kubectl get pods -n mwaa
```

您應該會看到類似下列的輸出：

```
NAME READY STATUS RESTARTS AGE
```

```
mwaa-pod-test-aa11bb22cc3344445555666677778888 0/1 Completed 0 2m23s
```

然後，您可以使用以下命令驗證網繭的輸出。將 name 值替換為從上一個命令返回的值：

```
kubectl logs -n mwaa mwaa-pod-test-aa11bb22cc3344445555666677778888
```

使用連接到亞馬遜 ECS**ECSOperator**

本主題說明如何使用ECSOperator從亞馬遜 MWAA 連接到亞馬遜彈性容器服務（亞馬遜 ECS）容器。在以下步驟中，您將將所需的權限添加到環境的執行角色中，使用AWS CloudFormation用於建立 Amazon ECS 遠門叢集的範本，最後建立並上傳連線到新叢集的 DAG。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

若要使用此頁面上的範例程式碼，您需要下列項目：

- 一個亞馬遜 MWAA 環境。

許可

- 您環境的執行角色需要許可才能在 Amazon ECS 中執行任務。您可以附上亞馬遜FullAccess AWS-託管策略到您的執行角色，或者創建以下策略並將其附加到您的執行角色。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask",
                "ecs:DescribeTasks"
            ],
            "Resource": "*"
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        }
    ]
}
```

- 除了新增必要的預先設定以在 Amazon ECS 中執行任務之外，您還必須修改CloudWatch在 Amazon MWAA 執行角色中記錄政策聲明，以允許存取 Amazon ECS 任務日誌群組，如下所示。亞馬遜 ECS 日誌群組是由AWS CloudFormation範本the section called “建立亞馬遜 ECS 叢集”。

```
{
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:CreateLogGroup",
        "logs:PutLogEvents",
        "logs:GetLogEvents",
        "logs:GetLogRecord",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
    ],
    "Resource": [
        "arn:aws:logs:region:account-id:log-group:airflow-environment-name-*",
```

```
        "arn:aws:logs:*:*:log-group:ecs-mwaa-group:*"
    ]
}
```

如需 Amazon MWAA 執行角色以及如何附加政策的詳細資訊，請參閱執行角色。

建立亞馬遜 ECS 叢集

使用以下AWS CloudFormation範本中，您將建立一個亞馬遜 ECS 遠門叢集，以搭配您的亞馬遜 MWAA 工作流程使用。如需詳細資訊，請參閱建立任務定義在Amazon 彈性容器服務開發人員指南。

1. 使用以下代碼創建 JSON 文件並將其另存為ecs-mwaa-cfn.json。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "This template deploys an ECS Fargate cluster with an Amazon
 Linux image as a test for MWAA.",
    "Parameters": {
        "VpcId": {
            "Type": "AWS::EC2::VPC::Id",
            "Description": "Select a VPC that allows instances access to ECR, as
 used with MWAA."
        },
        "SubnetIds": {
            "Type": "List<AWS::EC2::Subnet::Id>",
            "Description": "Select at two private subnets in your selected VPC, as
 used with MWAA."
        },
        "SecurityGroups": {
            "Type": "List<AWS::EC2::SecurityGroup::Id>",
            "Description": "Select at least one security group in your selected
 VPC, as used with MWAA."
        }
    },
    "Resources": {
        "Cluster": {
            "Type": "AWS::ECS::Cluster",
            "Properties": {
                "ClusterName": {
                    "Fn::Sub": "${AWS::StackName}-cluster"
                }
```

```
            }
        },
        "LogGroup": {
            "Type": "AWS::Logs::LogGroup",
            "Properties": {
                "LogGroupName": {
                    "Ref": "AWS::StackName"
                },
                "RetentionInDays": 30
            }
        },
        "ExecutionRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": "ecs-tasks.amazonaws.com"
                            },
                            "Action": "sts:AssumeRole"
                        }
                    ]
                },
                "ManagedPolicyArns": [
                    "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy"
                ]
            }
        },
        "TaskDefinition": {
            "Type": "AWS::ECS::TaskDefinition",
            "Properties": {
                "Family": {
                    "Fn::Sub": "${AWS::StackName}-task"
                },
                "Cpu": 2048,
                "Memory": 4096,
                "NetworkMode": "awsvpc",
                "ExecutionRoleArn": {
                    "Ref": "ExecutionRole"
                },
                "ContainerDefinitions": [
```

```
                    {
                        "Name": {
                            "Fn::Sub": "${AWS::StackName}-container"
                        },
                        "Image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/
amazonlinux:latest",
                        "PortMappings": [
                            {
                                "Protocol": "tcp",
                                "ContainerPort": 8080,
                                "HostPort": 8080
                            }
                        ],
                        "LogConfiguration": {
                            "LogDriver": "awslogs",
                            "Options": {
                                "awslogs-region": {
                                    "Ref": "AWS::Region"
                                },
                                "awslogs-group": {
                                    "Ref": "LogGroup"
                                },
                                "awslogs-stream-prefix": "ecs"
                            }
                        }
                    }
                ],
                "RequiresCompatibilities": [
                    "FARGATE"
                ]
            }
        },
        "Service": {
            "Type": "AWS::ECS::Service",
            "Properties": {
                "ServiceName": {
                    "Fn::Sub": "${AWS::StackName}-service"
                },
                "Cluster": {
                    "Ref": "Cluster"
                },
                "TaskDefinition": {
                    "Ref": "TaskDefinition"
                },
```

```
                "DesiredCount": 1,
                "LaunchType": "FARGATE",
                "PlatformVersion": "1.3.0",
                "NetworkConfiguration": {
                    "AwsvpcConfiguration": {
                        "AssignPublicIp": "ENABLED",
                        "Subnets": {
                            "Ref": "SubnetIds"
                        },
                        "SecurityGroups": {
                            "Ref": "SecurityGroups"
                        }
                    }
                }
            }
        }
    }
}
```

2. 在命令提示符下，使用以下命令AWS CLI用於建立新堆疊的指令。您必須取代這些值`SecurityGroups`和`SubnetIds`包含 Amazon MWAA 環境安全群組和子網路的值。

```
$ aws cloudformation create-stack \
--stack-name my-ecs-stack --template-body file://ecs-mwaa-cfn.json \
--parameters ParameterKey=SecurityGroups,ParameterValue=your-mwaa-security-group \
ParameterKey=SubnetIds,ParameterValue=your-mwaa-subnet-1\\,your-mwaa-subnet-1 \
--capabilities CAPABILITY_IAM
```

或者，您可以使用下面的 shell 腳本。此指令碼會擷取您環境安全性群組的必要值，以及使用[get-environment](#) AWS CLI命令，然後相應地創建堆棧。若要執行指令碼，請執行下列動作。

a. 複製，並將腳本另存為`ecs-stack-helper.sh`在與您的相同目錄中AWS CloudFormation範本。

```
#!/bin/bash

joinByString() {
  local separator="$1"
  shift
  local first="$1"
  shift
```

```
  printf "%s" "$first" "${@/#/$separator}"
}

response=$(aws mwaa get-environment --name $1)

securityGroupId=$(echo "$response" | jq -r
 '.Environment.NetworkConfiguration.SecurityGroupIds[]')
subnetIds=$(joinByString '\,' $(echo "$response" | jq -r
 '.Environment.NetworkConfiguration.SubnetIds[]'))

aws cloudformation create-stack --stack-name $2 --template-body file://ecs-
cfn.json \
--parameters ParameterKey=SecurityGroups,ParameterValue=$securityGroupId \
ParameterKey=SubnetIds,ParameterValue=$subnetIds \
--capabilities CAPABILITY_IAM
```

b. 使用下列命令執行指令碼。取代`environment-name`和`stack-name`與您的信息。

```
$ chmod +x ecs-stack-helper.sh
$ ./ecs-stack-helper.bash environment-name stack-name
```

如果成功，你會看到下面的輸出顯示你的新AWS CloudFormation堆疊識別碼。

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-ecs-
stack/123456e7-8ab9-01cd-b2fb-36cce63786c9"
}
```

之後你AWS CloudFormation堆疊已完成，AWS已佈建您的 Amazon ECS 資源，您就可以建立和上傳您的 DAG 了。

程式碼範例

1. 開啟命令提示字元，然後瀏覽至儲存 DAG 程式碼的目錄。例如：

```
cd dags
```

2. 複製下列程式碼範例的內容，並在本機儲存為`mwaa-ecs-operator.py`，然後將您的新 DAG 上傳到亞馬遜 S3。

```
from http import client
from airflow import DAG
from airflow.providers.amazon.aws.operators.ecs import ECSOperator
from airflow.utils.dates import days_ago
import boto3

CLUSTER_NAME="mwaa-ecs-test-cluster" #Replace value for CLUSTER_NAME with your
 information.
CONTAINER_NAME="mwaa-ecs-test-container" #Replace value for CONTAINER_NAME with
 your information.
LAUNCH_TYPE="FARGATE"

with DAG(
    dag_id = "ecs_fargate_dag",
    schedule_interval=None,
    catchup=False,
    start_date=days_ago(1)
) as dag:
    client=boto3.client('ecs')
    services=client.list_services(cluster=CLUSTER_NAME,launchType=LAUNCH_TYPE)

 service=client.describe_services(cluster=CLUSTER_NAME,services=services['serviceArns'])

    ecs_operator_task = ECSOperator(
        task_id = "ecs_operator_task",
        dag=dag,
        cluster=CLUSTER_NAME,
        task_definition=service['services'][0]['taskDefinition'],
        launch_type=LAUNCH_TYPE,
        overrides={
            "containerOverrides":[
                {
                    "name":CONTAINER_NAME,
                    "command":["ls", "-l", "/"],
                },
            ],
        },

        network_configuration=service['services'][0]['networkConfiguration'],
        awslogs_group="mwaa-ecs-zero",
        awslogs_stream_prefix=f"ecs/{CONTAINER_NAME}",
    )
```

> **Note**
>
> 在範例 DAG 中，awslogs_group時，您可能需要使用 Amazon ECS 任務日誌群組的名稱修改日誌群組。此範例假設名為的記錄群組mwaa-ecs-zero。對於awslogs_stream_prefix，請使用亞馬遜 ECS 任務日誌串流前綴。此範例假設記錄資料流前置詞，ecs。

3. 運行以下命令AWS CLI命令，將 DAG 複製到您環境的值區，然後使用 Apache 氣流 UI 觸發 DAG。

```
$ aws s3 cp your-dag.py s3://your-environment-bucket/dags/
```

4. 如果成功，您會在工作記錄中看到類似下列內容的輸出ecs_operator_task在ecs_fargate_dag天:

```
[2022-01-01, 12:00:00 UTC] {{ecs.py:300}} INFO - Running ECS Task -
Task definition: arn:aws:ecs:us-west-2:123456789012:task-definition/mwaa-ecs-test-
task:1 - on cluster mwaa-ecs-test-cluster
[2022-01-01, 12:00:00 UTC] {{ecs-operator-test.py:302}} INFO - ECSOperator
 overrides:
{'containerOverrides': [{'name': 'mwaa-ecs-test-container', 'command': ['ls', '-l',
 '/']}]}
.
.
.
[2022-01-01, 12:00:00 UTC] {{ecs.py:379}} INFO - ECS task ID is:
 e012340b5e1b43c6a757cf012c635935
[2022-01-01, 12:00:00 UTC] {{ecs.py:313}} INFO - Starting ECS Task Log Fetcher
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] total
 52
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC]
 lrwxrwxrwx   1 root root    7 Jun 13 18:51 bin -> usr/bin
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] dr-xr-
xr-x   2 root root 4096 Apr  9  2019 boot
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   5 root root  340 Jul 19 17:54 dev
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   1 root root 4096 Jul 19 17:54 etc
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Apr  9  2019 home
```

```
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC]
 lrwxrwxrwx   1 root root    7 Jun 13 18:51 lib -> usr/lib
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC]
 lrwxrwxrwx   1 root root    9 Jun 13 18:51 lib64 -> usr/lib64
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Jun 13 18:51 local
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Apr  9  2019 media
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Apr  9  2019 mnt
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Apr  9  2019 opt
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] dr-xr-
xr-x 103 root root    0 Jul 19 17:54 proc
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] dr-xr-
x-\-\-   2 root root 4096 Apr  9  2019 root
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Jun 13 18:52 run
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC]
 lrwxrwxrwx   1 root root    8 Jun 13 18:51 sbin -> usr/sbin
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x   2 root root 4096 Apr  9  2019 srv
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] dr-xr-
xr-x  13 root root    0 Jul 19 17:54 sys
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC]
 drwxrwxrwt   2 root root 4096 Jun 13 18:51 tmp
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x  13 root root 4096 Jun 13 18:51 usr
[2022-01-01, 12:00:00 UTC] {{ecs.py:119}} INFO - [2022-07-19, 17:54:03 UTC] drwxr-
xr-x  18 root root 4096 Jun 13 18:52 var
.
.
.
[2022-01-01, 12:00:00 UTC] {{ecs.py:328}} INFO - ECS Task has been successfully
 executed
```

使用 dbt 與亞馬遜 MWAA

本主題示範如何將 dbt 和 Postgres 與亞馬遜 MWAA 搭配使用。在以下步驟中，您將將所需的依賴項添加到`requirements.txt`，然後將範例 dbt 專案上傳到您環境的 Amazon S3 儲存貯體。然後，您

將使用範例 DAG 來驗證 Amazon MWAA 是否已安裝相依性，最後使用BashOperator來運行 dbt 項目。

主題

版本

- 您可以使用此頁面上的程式碼範例阿帕奇氣流 v2 及以上在蟒蛇。

先決條件

您需要下列項目，才能完成下列步驟：

- 一個亞馬遜 MWAA 環境使用阿帕奇氣流 v2.2.2。此範例已撰寫完畢，並使用 v2.2.2 進行測試。您可能需要修改範例，才能與其他 Apache 氣流版本搭配使用。
- 一個範例 dbt 專案。要開始使用 dbt 與亞馬遜 MWAA，您可以創建一個分支並克隆dbt 入門項目從數據庫實驗室GitHub儲存庫。

相依性

若要搭配 dbt 使用亞馬遜 MWAA，請將下列相依性新增至您的requirements.txt。如需進一步了解，請參閱 安裝 Python 的依賴

當您的環境完成更新時，Amazon MWAA 會安裝所需的 dbt 程式庫和其他相依性，例如psycopg2。

> **Note**
>
> Apache 氣流 v2.2.2 提供的預設條件約束檔案具有衝突的版本jsonschema本指南中使用的 dbt 版本不支持該版本。因此，將 Amazon MWAA 與 dbt 搭配使用時，您可以將 Apache 氣

流限制檔案下載並修改到您的 Amazon S3 DAG 資料夾中，然後在`requirements.txt`檔案為`--constraint /usr/local/airflow/dags/my-updated-constraint.txt`，或省略`--constraint`從`requirements.txt`如下所示。

```
json-rpc==1.13.0
minimal-snowplow-tracker==0.0.2
packaging==20.9
networkx==2.6.3
mashumaro==2.5
sqlparse==0.4.2

logbook==1.5.3
agate==1.6.1
dbt-extractor==0.4.0

pyparsing==2.4.7
msgpack==1.0.2
parsedatetime==2.6
pytimeparse==1.1.8
leather==0.3.4
pyyaml==5.4.1

# Airflow constraints are jsonschema==3.2.0
jsonschema==3.1.1
hologram==0.0.14
dbt-core==0.21.1

psycopg2-binary==2.8.6
dbt-postgres==0.21.1
dbt-redshift==0.21.1
```

在以下各節中，您會將 dbt 專案目錄上傳到 Amazon S3，並執行 DAG 來驗證 Amazon MWAA 是否已成功安裝所需的 dbt 相依性。

將 dbt 項目上傳到亞馬遜 S3

若要能夠將 dbt 專案與 Amazon MWAA 環境搭配使用，您可以將整個專案目錄上傳到您的環境`dags`資料夾。當環境更新時，亞馬遜 MWAA 會將 dbt 目錄下載到本機`usr/local/airflow/dags/`資料夾。

將 dbt 專案上傳到亞馬遜 S3

1. 瀏覽至您複製 dbt 入門專案的目錄。
2. 運行以下亞馬遜 S3AWS CLI命令以遞歸方式將項目的內容複製到您的環境dags資料夾使用--recursive參數。該命令創建一個名為的子目錄dbt你可以用於所有的 dbt 項目。如果子目錄已存在，則專案檔案會複製到現有目錄中，而不會建立新目錄。該命令還會在dbt此特定入門專案的目錄。

   ```
   $ aws s3 cp dbt-starter-project s3://mwaa-bucket/dags/dbt/dbt-starter-project --
   recursive
   ```

   您可以為項目子目錄使用不同的名稱來組織父項目中的多個 dbt 項目dbt目錄。

使用 DAG 來驗證 dbt 相依性安裝

下面的 DAG 使用BashOperator和一個 bash 命令來驗證亞馬遜 MWAA 是否已成功安裝在中指定的 dbt 依賴關係requirements.txt。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago

with DAG(dag_id="dbt-installation-test", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    cli_command = BashOperator(
        task_id="bash_command",
        bash_command="/usr/local/airflow/.local/bin/dbt --version"
    )
```

請執行下列動作以檢視工作記錄，並確認 dbt 及其相依性是否已安裝。

1. 瀏覽至亞馬遜 MWAA 主控台，然後選擇開啟氣流 UI從可用環境清單中。
2. 在 Apache 氣流使用者介面上，找到dbt-installation-test從清單中選擇 DAG，然後選擇Last Run列打開最後一個成功的任務。
3. 使用圖表檢視，選擇bash_command任務打開任務實例詳細信息。
4. 選擇日誌打開任務日誌，然後驗證日誌是否成功列出了我們在其中指定的 dbt 版本requirements.txt。

使用 DAG 執行 dbt 專案

下面的 DAG 使用BashOperator複製您從本地上傳到亞馬遜 S3 的 dbt 項目usr/local/airflow/dags/寫入可存取的目錄/tmp目錄中，然後運行 dbt 項目。bash 命令假定一個名為標題的初學者 dbt 項目dbt-starter-project。根據專案目錄的名稱修改目錄名稱。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago
import os
DAG_ID = os.path.basename(__file__).replace(".py", "")
with DAG(dag_id=DAG_ID, schedule_interval=None, catchup=False, start_date=days_ago(1))
 as dag:
    cli_command = BashOperator(
        task_id="bash_command",
        bash_command="cp -R /usr/local/airflow/dags/dbt /tmp;\
cd /tmp/dbt/dbt-starter-project;\
/usr/local/airflow/.local/bin/dbt  run --project-dir /tmp/dbt/dbt-starter-project/ --
profiles-dir ..;\
cat /tmp/dbt_logs/dbt.log"
    )
```

AWS博客和教程

- 與亞馬遜 EKS 和亞馬遜 MWAA 阿帕奇氣流 V2.x 的工作

Amazon 管理的 Apache 氣流工作流程的最佳實務

本指南說明在 Apache 氣流使用 Amazon 受管工作流程時，我們建議的最佳實務。

主題

Amazon MWAA 上阿帕奇氣流的性能調整

本頁說明我們建議使用的最佳實務來調整 Apache 氣流環境之 Amazon 受管工作流程的效能在 Amazon MWAA 上使用阿帕奇氣流配置選項。

內容

新增 Apache 氣流組態選項

下列程序會逐步引導您將 Airflow 組態選項新增至您的環境。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯。
4. 選擇下一步。

5. 在氣流組態選項窗格中選擇新增自訂組態。
6. 從下拉式清單中選擇組態並輸入值，或輸入自訂組態並輸入值。
7. 為您要新增的每個組態選擇新增自訂組態。
8. 選擇儲存。

如需進一步了解，請參閱在 Amazon MWAA 上使用阿帕奇氣流配置選項。

阿帕奇氣流調度

阿帕奇氣流調度程序是 Apache 氣流的核心組成部分。排程器的問題可能會導致 DAG 無法剖析和排程工作。如需有關 Apache 氣流排程器調整的詳細資訊，請參閱 Apache Airflow 文件網站中的微調排程器效能。

參數

本節說明 Apache 氣流排程器可用的組態選項及其使用案例。

Apache Airflow v2

版本	組態選項	預設	描述	使用案例
v2	提琴. 同步平行度	1	Celery 執行程式用來同步處理工作狀態的處理序數目。	您可以使用此選項，藉由限制 Celery 執行程式使用的處理程序來防止佇列衝突。依預設，會設定值以防止將1工作記錄傳送至 CloudWatch 記錄時發生錯誤。將此值設定為0表示使用最大處理序數目，但在傳送工作記錄時可能會導致錯誤。

版本	組態選項	預設	描述	使用案例
v2	排程器. 處理器 _ 波爾 _ 間隔	1	排程器「迴圈」中連續 DAG 檔案處理之間的等待秒數。	您可以使用此選項來釋放排程器上的 CPU 使用率，方法是增加排程器完成擷取 DAG 剖析結果、尋找和佇列工作，以及在 Executor 中執行佇列工作之後的休眠時間。增加此值會消耗在 Apache 氣流 v2 和 `scheduler .max_thre ads` Apache 氣流 v1 `scheduler .parsing_ processes` 的環境上執行的排程器執行緒數目。這可能會減少排程器剖析 DAG 的容量，並增加 DAG 在 Web 伺服器中顯示所需的時間。

版本	組態選項	預設	描述	使用案例
v2	排程程式最大值至建立循環	10	每個排程器「迴圈」要建立DagRuns的 DAG 數目上限。	您可以使用此選項，藉由減少排程器「迴圈」的最大數DagRuns目，以釋放資源來排定工作。
v2	排程器. 解析程序	2	排程器可以 parallel 執行以排程 DAG 的執行緒數目。	您可以使用此選項，藉由減少排程器 parallel 執行以剖析 DAG 的處理序數目來釋放資源。如果 DAG 剖析影響工作排程，我們建議您保持較低的數字。您必須指定小於環境中 vCPU 計數的值。如需進一步了解，請參閱限制。

限制

本節說明調整排程器預設參數時應考量的限制。

調度程序. 解析進程, 調度程序. 最大線程

一個環境類別的每個 vCPU 允許兩個執行緒。至少必須為環境類別的排程器保留一個執行緒。如果您發現排程的工作有延遲，您可能需要增加環境類別。例如，大型環境的排程器具有 4 個 vCPU Fargate 容器執行個體。這意味著最多的7總線程可用於其他進程。也就是說，兩個執行

緒乘以四個 vCPUs，減去排程器本身的一個執行緒。您在中指定`scheduler.max_threads`且不`scheduler.parsing_processes`得超過環境類別可用的執行緒數目 (如下所示：

- mw1.small-不得超過其他進1程的線程。剩餘的執行緒會保留給排程器使用。
- mw1.medium — 不得超過其他處理3程序的執行緒。剩餘的執行緒會保留給排程器使用。
- mw1.large — 不得超過其他處理7程序的執行緒。剩餘的執行緒會保留給排程器使用。

DAG 資料夾

Apache 氣流排程器會持續掃描環境中的 DAG 資料夾。任何包含的`plugins.zip`檔案，或包含「氣流」匯入陳述式的 Python (`.py`) 檔案。然後會將任何產生的 Python DAG 物件放入DagBag該檔案中，以便由排程器處理，以決定需要排程的工作 (如果有的話)。無論檔案是否包含任何可行的 DAG 物件，都會發生 DAG 檔案剖析。

參數

本節說明 DAGs 資料夾可用的組態選項及其使用案例。

Apache Airflow v2

版本	組態選項	預設	描述	使用案例
v2	排程器. DAG 目錄清單間隔	300 秒	應掃描 DAGs 資料夾中是否有新檔案的秒數。	您可以使用此選項，藉由增加剖析 DAGs 資料夾的秒數來釋放資源。如果您在中看到較長的剖析時間，建議您增加此值`total_parse_time` `metrics`，這可能是因為 DAG 資料夾中有大量檔案所致。

版本	組態選項	預設	描述	使用案例
v2	排程程式 .min 檔案 _ 處理間隔	30 秒	排程器剖析 DAG 和 DAG 更新之後的秒數會反映出來。	您可以使用此選項來釋放資源，方法是增加排程器在剖析 DAG 之前等待的秒數。例如，如果您指定的值為30，則會在每隔 30 秒剖析一次 DAG 檔案。建議您將此數字保持較高，以減少環境中的 CPU 使用率。

DAG 檔案

作為 Apache 氣流排程器迴圈的一部分，個別的 DAG 檔案會剖析以擷取 DAG Python 物件。在 Apache Airflow v2 及更新版本中，排程器會同時剖析最多數目的剖析程序。在重新剖析相同檔案之前，`scheduler.min_file_process_interval`必須經過中指定的秒數。

參數

本節說明 Apache 氣流 DAG 檔案的可用組態選項及其使用案例。

Apache Airflow v2

版本	組態選項	預設	描述	使用案例
v2	核心檔案處理器逾時	五十秒	處理 DAG 檔案逾DagFileProcessor時之前的秒數。	您可以使用此選項，藉由增加逾時前所需的時間來釋放資源。DagFileProcessor如果您

版本	組態選項	預設	描述	使用案例
				在 DAG 處理記錄檔中看到逾時而導致無法載入可行的 DAG，建議您增加此值。
v2	核心匯入逾時	30 秒	匯入 Python 檔案前的秒數逾時。	您可以使用此選項來釋放資源，方法是在匯入 Python 檔案以擷取 DAG 物件時，增加排程器逾時所需的時間。此選項會作為排程器「迴圈」的一部分進行處理，且必須包含小於中指定值的值`core.dag_file_processor_timeout` 。

版本	組態選項	預設	描述	使用案例
v2	核心. 分鐘序列化的 DAG 更新間隔	30	更新資料庫中序列化 DAG 之後的秒數下限。	您可以使用此選項，藉由增加資料庫中序列化 DAG 更新的秒數來釋放資源。如果您有大量 DAG 或複雜的 DAG，建議您增加此值。當 DAG 序列化時，增加此值可減少排程器和資料庫的負載。
v2	核心. 分鐘序列化 _DAG_ 擷取間隔	10	已在中載入時，從資料庫中重新擷取序列化 DAG 的秒數。DagBag	您可以使用此選項，藉由增加重新擷取序列化 DAG 的秒數來釋放資源。此值必須大於中指定的值，`core.min_serialized_dag_update_interval` 以降低資料庫「寫入」率。當 DAG 序列化時，增加此值可減少 Web 伺服器和資料庫的負載。

任務

Apache Airflow 排程器和工作者都參與佇列和取消佇列工作。排程器會將已剖析的工作準備好從「無」狀態排程為「已排程」狀態。執行程序也在 Fargate 的調度程序容器上運行，將這些任務排入隊列並將其狀態設置為「已排入佇列」。當 Worker 有容量時，它會從佇列取得工作，並將狀態設定為「執行中」，然後根據工作是成功還是失敗，將狀態變更為「成功」或「失敗」。

參數

本節說明 Apache 氣流工作的可用組態選項及其使用案例。

Amazon MWAA ################

Apache Airflow v2

版本	組態選項	預設	描述	使用案例
v2	核心. 平行	10000	狀態為「執行中」的作業執行個體數目上限。	您可以使用此選項，藉由增加可同時執行的工作執行個體數目來釋放資源。指定的值應該是「工作者」工作密度的可用 Worker 數目「次」。我們建議您只有在看到大量工作停留在「執行中」或「已佇列」狀態時才變更此值。
v2	核心并行性	10000	允許針對每個 DAG 同時執行的工作執行個體數目。	您可以使用此選項，藉由增加允許同時執行的作業執行個體數目來釋放資源。例

版本	組態選項	預設	描述	使用案例
				如，如果您有一百個 DAG 具有十個 parallel 工作，而且您希望所有 DAG 同時執行，您可以計算最大平行處理原則，做為可用 Worker 工作者工作密度的「乘以」的數目`celery.worker_concurrency` ，除以 DAG 數目 (例如 100)。
v2	核心. 執行任務 _ 新的 Python 解釋器	`True`	判斷 Apache 氣流是否透過分叉父處理序或建立新的 Python 程序來執行工作。	設定為時`True`，Apache 氣流會將您對外掛程式所做的變更識別為新的 Python 處理程序，以便為執行工作而建立。
v2	天然工人并行	N/A	Amazon MWAA 會覆寫此選項的 Airflow 基本安裝，以便將員工擴展為其自動調度資源元件的一部分。	########## #####

版本	組態選項	預設	描述	使用案例
v2	天然. 工人 _ 自動縮放	兆瓦 1. 小-5,0 兆瓦 1. 中-10，0 兆瓦 1. 大-20，0 米瓦特大-40,0 兆瓦 1.2 克大-80,0	工作者的工作並行處理。	您可以使用此選項`minimum`，藉由減少 Worker 的工*`maximum`*作並行處理來釋放資源。無論是否有足夠的資源可執`maximum`行，Worker 都會接受最多已設定的並行作業。如果排程的工作沒有足夠的資源，工作會立即失敗。我們建議您針對資源密集型工作變更此值，方法是將值減少為小於預設值，以允許每項工作擁有更多容量。

在 requirements.txt Python 管理依賴項

本頁說明我們建議在適用於 Apache 氣流環境之 Amazon 受管工作流程的`requirements.txt`檔案中安裝和管理 Python 相依性的最佳實務。

內容

使用 Amazon MWAA CLI 公用程式測試 DAG

- 命令列介面 (CLI) 公用程式可在本機複寫 Apache 氣流環境的 Amazon 受管工作流程。
- CLI 會在本機建立類似於 Amazon MWAA 生產映像的碼頭容器映像。這可讓您執行本機 Apache 氣流環境來開發和測試 DAG、自訂外掛程式和相依性，然後再部署到 Amazon MWAA。
- 要運行 CLI，請參閱（詳見） GitHub。aws-mwaa-local-runner

使用 PyPi .org 要求文件格式安裝 Python 依賴關係

以下部分描述了根據 PyPi .org 要求文件格式安裝 Python 依賴關係的不同方法。

選項一：Python Package 索引中的依賴關係

以下部分介紹如何從`requirements.txt`文件中的 Python P ackage 索引指定 Python 依賴關係。

Apache Airflow v2

1. 在本地測試。在建立檔案之前，以反覆方式新增其他程式庫，以尋找套件及其版本的正確組合。`requirements.txt`若要執行 Amazon MWAA CLI 公用程式，請參閱（詳見）。aws-mwaa-local-runner GitHub
2. 檢閱 Apache 氣流套件附加功能。若要檢視 Amazon MWAA 上為 Apache 氣流 v2 安裝的套件清單，請參閱網站上的 Amazon MWAA 本地運`requirements.txt`行器。 GitHub
3. 新增條件約束陳述式。在檔案頂端新增 Apache 氣流 v2 環境的限制`requirements.txt`檔案。Apache 氣流限制檔案會指定 Apache 氣流發行時可用的提供者版本。

從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一--constraint份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。

在下面的例子中，將 *{####}* 替換為您環境的版本號，並將 *{Python ##}* 替換為與您的環境兼容的 Python 版本。

如需與 Apache 氣流環境相容的 Python 版本的相關資訊，請參閱 Apache 氣流版本。

```
--constraint "https://raw.githubusercontent.com/apache/airflow/
constraints-{Airflow-version}/constraints-{Python-version}.txt"
```

如果條件約束檔案判斷xyz==1.0套件與您環境中的其他套件不相容，pip3 install將無法防止不相容的程式庫安裝到您的環境中。如果任何套件的安裝失敗，您可以在記錄檔上的對應記錄資料流中檢視每個 Apache Airflow 元件 (排程器、Worker 和 Web 伺服器) 的錯誤 CloudWatch 記錄。如需記錄類型的詳細資訊，請參閱the section called “檢視氣流記錄”。

4. 阿帕奇氣流包。添加包附加功能和版本（==）。這有助於防止在您的環境中安裝相同名稱但版本不同的套件。

```
apache-airflow[package-extra]==2.5.1
```

5. Python 庫。在文件中添加軟requirements.txt件包名稱和版本（==）。這有助於防止將future 自 PyPi.org 的重大更新被自動套用。

```
library == version
```

Example 肉毒桿菌毒素 3 和心理 2-二進制

此範例是為了展示目的而提供。博托和 psycopg2 二進制庫包含在 Apache 氣流 v2 基本安裝中，不需要在文件中指定。requirements.txt

```
boto3==1.17.54
boto==2.49.0
botocore==1.20.54
psycopg2-binary==2.8.6
```

如果指定的套件沒有版本，Amazon MWAA 會從 PyPi .org 安裝最新版本的套件。此版本可能會與您的requirements.txt.

Apache Airflow v1

1. 在本地測試。在建立檔案之前，以反覆方式新增其他程式庫，以尋找套件及其版本的正確組合。`requirements.txt`若要執行 Amazon MWAA CLI 公用程式，請參閱（詳見）。aws-mwaa-local-runner GitHub
2. 檢閱氣流套件附加功能。檢閱可用於阿帕奇氣流 v1.10.12 的套件清單，網址為 https://raw.githubusercontent.com/apache/airflow/constraints-1.10.12/constraints-3.7.txt。
3. 加入約束檔案。將 Apache 氣流 v1.10.12 的限制文件添加到文件的頂部。`requirements.txt`如果條件約束檔案判斷`xyz==1.0`套件與您環境中的其他套件不相容，`pip3 install`將無法防止不相容的程式庫安裝到您的環境中。

   ```
   --constraint "https://raw.githubusercontent.com/apache/airflow/
   constraints-1.10.12/constraints-3.7.txt"
   ```

4. 阿帕奇氣流 1.10.12 版軟件包。添加氣流包附加功能和阿帕奇氣流 v1.10.12 版本（）。==這有助於防止在您的環境中安裝相同名稱但版本不同的套件。

   ```
   apache-airflow[package]==1.10.12
   ```

   Example 安全殼層

   下面的示例`requirements.txt`文件安裝 SSH 阿帕奇氣流 v1.10.12。

   ```
   apache-airflow[ssh]==1.10.12
   ```

5. Python 庫。在文件中添加軟`requirements.txt`件包名稱和版本（==）。這有助於防止將 future 自 PyPi.org 的重大更新被自動套用。

   ```
   library == version
   ```

   Example Boto3

   下面的示例`requirements.txt`文件安裝博托 3 庫阿帕奇氣流 v1.10.12。

   ```
   boto3 == 1.17.4
   ```

   如果指定的套件沒有版本，Amazon MWAA 會從 PyPi .org 安裝最新版本的套件。此版本可能會與您的`requirements.txt`.

選項二：Python 輪（.whl）

Python 滾輪是一種包格式，旨在將庫與編譯的工件一起運送。輪子套件在 Amazon MWAA 中安裝相依性的方法有幾個好處：

- 更快的安裝 — WHL 檔案會以單一 ZIP 的形式複製到容器中，然後在本機安裝，而不必下載每個檔案。
- 減少衝突 — 您可以事先判斷套件的版本相容性。因此，不需要遞歸地計pip算出兼容的版本。
- 彈性更高 — 使用外部託管的程式庫，下游需求可能會改變，導致 Amazon MWAA 環境上容器之間的版本不相容。由於不依賴於依賴關係的外部源，無論每個容器何時實例化，每個容器都具有相同的庫。

我們建議您使用以下方法來安裝 Python 的依賴關係從 Python 輪子存檔（`.whl`）在您的`requirements.txt`.

方法

- 使用 Amazon S3 存儲桶上的plugins.zip文件
- 使用託管在網址上的 WHL 文件
- 從 DAG 建立 WHL 檔案

使用 Amazon S3 存儲桶上的**plugins.zip**文件

Apache 氣流排程器、工作者和網頁伺服器 (適用於 Apache Airflow v2.2.2 及更新版本) 會在您 AWS環境的受管理 Fargate 容器上啟動期間尋找自訂外掛程式。`/usr/local/airflow/plugins/*`此程序開始於 Amazon MWAA 的 `pip3 install -r requirements.txt` Python 相依性和 Apache 氣流服務啟動之前。`plugins.zip`檔案可用於您不希望在環境執行期間持續變更的任何檔案，或是您可能不想要授與寫入 DAG 的使用者存取權。例如，Python 庫輪盤文件，證書 PEM 文件和配置 YAML 文件。

下節說明如何在 Amazon S3 儲存貯體上安裝`plugins.zip`檔案中的輪子。

1. 下載必要的 WHL 檔案您可以`pip download``requirements.txt`在 Amazon MWAA 本機執行器或其他 Amazon Linux 2 容器上使用，以解決並下載必要的 Python 輪盤檔案。

```
$ pip3 download -r "$AIRFLOW_HOME/dags/requirements.txt" -d "$AIRFLOW_HOME/plugins"
$ cd "$AIRFLOW_HOME/plugins"
$ zip "$AIRFLOW_HOME/plugins.zip" *
```

2. 指定您的 **`requirements.txt`**. 使用指定 requirements.txt 頂部的插件目錄，`--find-links`並指示pip不要使用從其他來源安裝 `--no-index`，如下所示

```
--find-links /usr/local/airflow/plugins
--no-index
```

Example 在 requirements.txt 中的車輪

下列範例假設您已將輪子上傳到 Amazon S3 儲存貯體根目錄的`plugins.zip`檔案中。例如：

```
--find-links /usr/local/airflow/plugins
--no-index

numpy
```

Amazon MWAA 會從`plugins`資料夾中取出`numpy-1.20.1-cp37-cp37m-manylinux1_x86_64.whl`滾輪，並將其安裝在您的環境中。

使用託管在網址上的 WHL 文件

以下部分說明如何安裝裝載在 URL 上的操控盤。該 URL 必須可公開存取，或可從您為 Amazon MWAA 環境指定的自訂 Amazon VPC 中存取。

- 提供網址。將 URL 提供給您的`requirements.txt`.

  Example 公共 URL 上的車輪歸檔

  下列範例會從公用網站下載操控盤。

```
--find-links https://files.pythonhosted.org/packages/
--no-index
```

  Amazon MWAA 會從您指定的 URL 擷取滾輪，然後將它們安裝在您的環境中。

> **Note**
>
> 無法從 Amazon MWAA v2.2.2 及更新版本中安裝需求的私有網頁伺服器存取 URL。

從 DAG 建立 WHL 檔案

如果您擁有使用 Apache Airflow v2.2.2 或更新版本的私有網頁伺服器，且因為您的環境無法存取外部儲存庫而無法安裝需求，則可以使用下列 DAG 取得您現有的 MamaZon MVAA 需求，並將它們封裝在 Amazon S3 上：

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago

S3_BUCKET = 'my-s3-bucket'
S3_KEY = 'backup/plugins_whl.zip'

with DAG(dag_id="create_whl_file", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    cli_command = BashOperator(
        task_id="bash_command",
        bash_command=f"mkdir /tmp/whls;pip3 download -r /usr/local/airflow/
requirements/requirements.txt -d /tmp/whls;zip -j /tmp/plugins.zip /tmp/whls/*;aws s3
 cp /tmp/plugins.zip s3://{S3_BUCKET}/{S3_KEY}"
    )
```

執行 DAG 之後，請使用此新檔案做為 Amazon MWAA`plugins.zip`，選擇性地與其他外掛程式一起封裝。然後，更新您的`requirements.txt`先前`--find-links /usr/local/airflow/plugins`和`--no-index`不添加`--constraint`。

此方法可讓您離線使用相同的程式庫。

選項三：託管在私有 PyPi /PEP-503 兼容存儲庫上的 Python 依賴關係

以下部分說明如何安裝 Apache 氣流額外託管在具有驗證的私人 URL 上。

1. 將您的使用者名稱和密碼新增為 Apache 氣流組態選項。例如：
   - `foo.user` : *YOUR_USER_NAME*
   - `foo.pass` : *YOUR_PASSWORD*
2. 建立您的`requirements.txt`檔案。將下列範例中的預留位置替換為您的私人 URL，以及您新增為 Apache Airflow 組態選項的使用者名稱和密碼。例如：

   ```
   --index-url https://${AIRFLOW__FOO__USER}:${AIRFLOW__FOO__PASS}@my.privatepypi.com
   ```

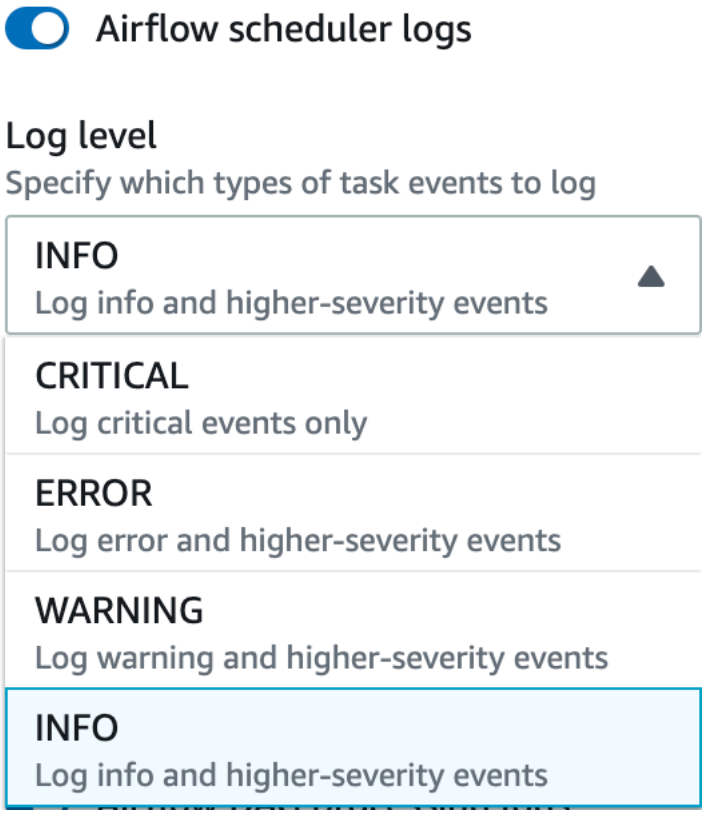
3. 將任何其他程式庫新增至您的requirements.txt檔案。例如：

```
--index-url https://${AIRFLOW__FOO__USER}:${AIRFLOW__FOO__PASS}@my.privatepypi.com
my-private-package==1.2.3
```

在 Amazon MWAA 主控台上啟用日誌

Amazon MWAA 環境的執行角色需要將日誌傳送到 CloudWatch 日誌的權限。若要更新執行角色的權限，請參閱Amazon MWAA 執行角色。

您可以在INFO、WARNING或CRITICAL層級啟用 Apache 氣流記錄。ERROR當您選擇記錄層級時，Amazon MWAA 會傳送該層級和所有較高嚴重性層級的日誌。例如，如果您在INFO層級啟用日誌，Amazon MWAA 會將日誌WARNING、ERROR、和INFO日誌層級傳送到 CloudWatch 日CRITICAL誌。我們建議您在INFO層級啟用 Apache Airflow 記錄，讓排程器檢視為requirements.txt.



在記錄主控台上檢視 CloudWatch 記錄檔

您可以檢視排程器排程工作流程和剖析dags資料夾的 Apache Airflow 記錄。下列步驟說明如何在 Amazon MWAA 主控台上開啟排程器的日誌群組，以及如何在日誌主控台上檢視 Apache 氣流 CloudWatch 日誌。

若要檢視 **`requirements.txt`**

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [監視] 窗格中選擇 Airflow 排程器記錄群組。
4. 在「requirements_install_ip記錄資料流」中選擇記錄檔。
5. 您應該會看到已安裝在環境中的套件清單，位於/usr/local/airflow/.local/bin。例如：

```
Collecting appdirs==1.4.4 (from -r /usr/local/airflow/.local/bin (line 1))
Downloading https://files.pythonhosted.org/
packages/3b/00/2344469e2084fb28kjdsfiuyweb47389789vxbmnbjhsdgf5463acd6cf5e3db69324/
appdirs-1.4.4-py2.py3-none-any.whl
Collecting astroid==2.4.2 (from -r /usr/local/airflow/.local/bin (line 2))
```

6. 檢閱套件清單，以及這些套件是否在安裝期間遇到錯誤。如果發生錯誤，您可能會看到類似下列內容的錯誤：

```
2021-03-05T14:34:42.731-07:00
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
```

檢視 Apache 氣流使用者介面中的錯誤

您也可以檢查 Apache 氣流使用者介面，以識別錯誤是否可能與其他問題有關。在 Amazon MWAA 上，您可能會遇到的阿帕奇氣流的最常見錯誤是：

```
Broken DAG: No module named x
```

如果您在 Apache Airflow 使用者介面中看到此錯誤，表示您可能會遺漏requirements.txt檔案中必要的相依性。

登錄到阿帕奇氣流

您需要 AWS Identity and Access Management (IAM) 中的 AWS 帳戶阿帕奇氣流 UI 訪問政策:亞馬遜 WebServerAccess許可，才能檢視您的 Apache 氣流使用者介面。

存取您的 Apache 氣流使用者介面

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇「開啟氣流 UI」。

範例`requirements.txt`案例

您可以在`requirements.txt`. 下列範例使用不同方式的組合來安裝額外功能。

Example PyPi.org 和公共網址上的其他內容

除了公開 URL 上的套件之外，您還需要在指定 PyPi .org 的套件時使用此`--index-url`選項，例如自訂 PEP 503 相容的存放庫 URL。

```
aws-batch == 0.6
phoenix-letter >= 0.3

--index-url http://dist.repoze.org/zope2/2.10/simple
    zopelib
```

針對 Apache 氣流的 Amazon 受管工作流程監控和指標

監控是維護適用於 Apache Airflow 的 Amazon 受管工作流程和您的 AWS 解決方案的可靠性、可用性和效能的重要組成部分。我們建議您從 AWS 解決方案的所有部分收集監控資料，以便在發生多點故障時更輕鬆地對多點失敗進行偵錯。本主題說明哪些資源 AWS 可用於監控 Amazon MWAA 環境和回應潛在事件。

> **Note**
>
> Apache 氣流指標和記錄需遵守標準 Amazon CloudWatch 定價。

如需監控 Apache 氣流的詳細資訊，請參閱 Apache 氣流文件網站中的記錄與監控。

章節

在 Amazon MWAA 上監控概觀

本頁說明用於 AWS 監控 Apache 氣流環境之 Amazon 受管工作流程的服務。

內容

Amazon CloudWatch 概述

CloudWatch 是 AWS 服務的測量結果儲存庫，可讓您根據服務所發佈的測量結果和維度擷取統計資料。您可以使用這些指標來設定警示、計算統計資料，然後在儀表板中顯示資料，以協助您在 Amazon CloudWatch 主控台評估環境的運作狀態。

Apache 氣流已經設置為將 Apache 氣流環境的 Amazon 管理工作流程的 StatsD 指標發送到 Amazon CloudWatch。

要進一步了解，請參閱什麼是 Amazon CloudWatch？ 。

AWS CloudTrail 概述

CloudTrail 是一項稽核服務，提供 Amazon MWAA 中使用者、角色或 AWS 服務所採取之動作的記錄。使用收集的資訊 CloudTrail，您可以判斷向 Amazon MWAA 發出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及稽核日誌中提供的其他詳細資訊。

若要深入瞭解，請參閱什麼是 AWS CloudTrail？ 。

檢視稽核記錄 AWS CloudTrail

AWS CloudTrail 在您建立 AWS 帳戶時，已在您的帳戶上啟用。 CloudTrail 記錄 IAM 實體或 AWS 服務所採取的活動，例如 Apache 氣流的 Amazon 受管工作流程，這會記錄為 CloudTrail 事件。您可以在主控台中檢視、搜尋和下載過去 90 天的事件歷史記 CloudTrail 錄。 CloudTrail 擷取 Amazon MWAA 主控台上的所有事件，以及對 Amazon MWAA API 的所有呼叫。它不會擷取唯讀動作，例如`GetEnvironment`，或`PublishMetrics`動作。本頁說明如何用 CloudTrail 來監控 Amazon MWAA 的事件。

內容

在中建立軌跡 CloudTrail

您需要建立追蹤，以檢視 AWS 帳戶中持續的事件記錄，包括 Amazon MWAA 的事件。追蹤可 CloudTrail 將日誌檔交付到 Amazon S3 儲存貯體。如果您不建立追蹤，您仍然可以在 CloudTrail 主控台中檢視可用的事件歷史記錄。例如，使用收集的資訊 CloudTrail，您可以判斷向 Amazon MWAA 發出的請求、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。若要深入瞭解，請參閱為您的 AWS 帳戶建立追蹤。

以 CloudTrail 事件歷史記錄檢視事件

您可以檢視事件歷程記錄，在 CloudTrail 主控台中疑難排解過去 90 天的操作和安全性事件。例如，您可以針對每個區域檢視與建立、修改或刪除 AWS 帳戶中資源 (例如 IAM 使用者或其他 AWS 資源) 相關的事件。若要深入瞭解，請參閱檢視具有事 CloudTrail 件歷史記錄的事件。

1. 開啟 CloudTrail 主控台。
2. 選擇事件歷史記錄。
3. 選取您要檢視的事件，然後選擇 [比較事件詳細資料]。

範例軌跡 `CreateEnvironment`

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 記錄檔包含一或多個記錄項目。事件代表來自任何來源的單一請求，並包含所請求動作的相關資訊，例如動作的日期和時間，或請求參數。 CloudTrail 日誌文件不是公共 API 調用的有序堆棧跟踪，並且不會以任何特定順序顯示。下列範例為因缺少權限而遭拒絕之`CreateEnvironment`動作的記錄項目。中的值為`AirflowConfigurationOptions`了保護隱私而被編輯。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "00123456ABC7DEF8HIJK",
        "arn": "arn:aws:sts::012345678901:assumed-role/root/myuser",
        "accountId": "012345678901",
        "accessKeyId": "",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "00123456ABC7DEF8HIJK",
                "arn": "arn:aws:iam::012345678901:role/user",
                "accountId": "012345678901",
                "userName": "user"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-10-07T15:51:52Z"
            }
```

```
        }
    },
    "eventTime": "2020-10-07T15:52:58Z",
    "eventSource": "airflow.amazonaws.com",
    "eventName": "CreateEnvironment",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.178",
    "userAgent": "PostmanRuntime/7.26.5",
    "errorCode": "AccessDenied",
    "requestParameters": {
        "SourceBucketArn": "arn:aws:s3:::my-bucket",
        "ExecutionRoleArn": "arn:aws:iam::012345678901:role/AirflowTaskRole",
        "AirflowConfigurationOptions": "***",
        "DagS3Path": "sample_dag.py",
        "NetworkConfiguration": {
            "SecurityGroupIds": [
                "sg-01234567890123456"
            ],
            "SubnetIds": [
                "subnet-01234567890123456",
                "subnet-65432112345665431"
            ]
        },
        "Name": "test-cloudtrail"
    },
    "responseElements": {
        "message": "Access denied."
    },
    "requestID": "RequestID",
    "eventID": "EventID",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "012345678901"
}
```


後續步驟？

- 了解如何針對CloudTrail 支援的 AWS 服務和整合的 CloudTrail 記錄檔中收集的事件資料設定其他服務。
- 請參閱設定的 Amazon SNS 通知，了解如何在將新的日誌檔 CloudTrail 發佈到 Amazon S3 儲存貯體時收到通知 CloudTrail。

在 Amazon 中查看氣流日誌 CloudWatch

Amazon MWAA 可以將阿帕奇氣流日誌發送到 Amazon。 CloudWatch您可以從單一位置檢視多個環境的記錄，輕鬆識別 Apache Airflow 工作延遲或工作流程錯誤，而不需要其他協力廠商工具。您必須在 Apache 氣流主控台的 Amazon 受管工作流程上啟用 Apache 氣流日誌，才能檢視 Apache 氣流 DAG 處理、任務、網頁伺服器、工作者登入 CloudWatch。

內容

定價

- 需支付標準 CloudWatch 記錄費用。如需詳細資訊，請參閱CloudWatch 定價。

開始之前

- 您必須擁有可以在中檢視記錄的角色 CloudWatch。如需詳細資訊，請參閱 存取 Amazon MWAA 環境。

記錄檔類型

Amazon MWAA 會為您啟用的每個 Airflow 記錄選項建立一個日誌群組，並將日誌推送到與環境關聯的 CloudWatch 日誌群組。記錄群組的命名格式如下：`YourEnvironmentName-`**`LogType`**例如，如果您的環境命名為`Airflow-v202-Public`，則 Apache 氣流工作記錄會傳送至`Airflow-v202-Public-`**`Task`**。

日誌類型	描述
YourEnvironmentName-**DAGProcessing**	DAG 處理器管理員 (處理 DAG 檔案的排程器部分) 的記錄。
YourEnvironmentName-**Scheduler**	氣流排程器產生的記錄。
YourEnvironmentName-**Task**	DAG 產生的工作記錄檔。
YourEnvironmentName-**WebServer**	氣流網頁介面所產生的記錄檔。
YourEnvironmentName-**Worker**	作為工作流程和 DAG 執行的一部分產生的記錄檔。

啟用 Apache 氣流記錄

您可以在INFO、WARNING或CRITICAL層級啟用 Apache 氣流記錄。ERROR當您選擇記錄層級時，Amazon MWAA 會傳送該層級和所有較高嚴重性層級的日誌。例如，如果您在INFO層級啟用日誌，Amazon MWAA 會將日誌WARNING、ERROR、和INFO日誌層級傳送到 CloudWatch 日CRITICAL誌。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 選擇編輯。
4. 選擇下一步。
5. 選擇下列一或多個記錄選項：
    a. 在 [監視] 窗格中選擇 Airflow 排程器記錄群組。
    b. 在 [監視] 窗格中選擇 Airflow Web 伺服器記錄群組。
    c. 在 [監視] 窗格中選擇氣流工作者記錄群組。
    d. 在 [監視] 窗格上選擇氣流 DAG 處理記錄群組。
    e. 選擇 [監視] 窗格上的 [氣流] 工作記錄群組。
    f. 在記錄層級中選擇記錄層級。
6. 選擇下一步。
7. 選擇儲存。

檢視阿帕奇氣流記錄

下節說明如何在 CloudWatch 主控台中檢視 Apache 氣流記錄檔。

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [監視] 窗格中選擇記錄群組。
4. 在記錄串流中選擇一個登入。

排程器記錄範例

您可以檢視排程器排程工作流程和剖析dags資料夾的 Apache Airflow 記錄。下列步驟說明如何在 Amazon MWAA 主控台上開啟排程器的日誌群組，以及如何在日誌主控台上檢視 Apache 氣流 CloudWatch 日誌。

若要檢視 **`requirements.txt`**

1. 在 Amazon MWAA 主控台上開啟「環境」頁面。
2. 選擇一個環境。
3. 在 [監視] 窗格中選擇 Airflow 排程器記錄群組。
4. 在「requirements_install_ip記錄資料流」中選擇記錄檔。
5. 您應該會看到已安裝在環境中的套件清單，位於/usr/local/airflow/.local/bin。例如：

```
Collecting appdirs==1.4.4 (from -r /usr/local/airflow/.local/bin (line 1))
Downloading https://files.pythonhosted.org/
packages/3b/00/2344469e2084fb28kjdsfiuyweb47389789vxbmnbjhsdgf5463acd6cf5e3db69324/
appdirs-1.4.4-py2.py3-none-any.whl
Collecting astroid==2.4.2 (from -r /usr/local/airflow/.local/bin (line 2))
```

6. 檢閱套件清單，以及這些套件是否在安裝期間遇到錯誤。如果發生錯誤，您可能會看到類似下列內容的錯誤：

```
2021-03-05T14:34:42.731-07:00
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
No matching distribution found for LibraryName==1.0.0 (from -r /usr/local/
airflow/.local/bin (line 4))
```

後續步驟？

- 在使用鬧鐘中了解如何設定 CloudWatch 鬧 AWS CloudTrail 鐘。
- 在使用儀表板中了解如何建立 CloudWatch 儀表板。 CloudWatch

監控 Amazon MWAA 上的儀表板和警報

您可以在 Amazon 中建立自訂儀表板， CloudWatch 並為特定指標新增警示，以監控 Apache Airflow 環境之 Amazon 受管工作流程的運作狀態。當警示顯示在儀表板上時，警示處於ALARM狀態時會變成紅色，讓您更輕鬆地主動監控 Amazon MWAA 環境的健康狀態。

Apache Airflow 會公開許多處理序的指標，包括 DAG 處理序數目、DAG 包大小、目前執行中的工作、工作失敗和成功。當您建立環境時，氣流會設定為自動將 Amazon MWAA 環境的指標傳送到。CloudWatch本頁說明如何為 Amazon MWAA 環境的氣流指標建立健康狀態儀表板。 CloudWatch

內容

指標

您可以為 Apache Airflow 版本的任何可用指標建立自訂儀表板和警示。每個量度都對應於一個 Apache 氣流關鍵效能指標 (KPI)。若要檢視測量結果清單，請參閱：

- 阿帕奇氣流 v2 環境指標 CloudWatch

警報狀態概述

警示擁有以下可能的狀態：

- OK – 指標或表達式在定義的閾值內。
- ALARM – 指標或表達式在定義的閾值外。
- INSUFFICIENT_DATA – 警示剛開始無法使用指標，或資料不足無法讓指標判斷警示狀態。

自訂儀表板和警示範例

您可以建立自訂的監控儀表板，以顯示 Amazon MWAA 環境所選指標的圖表。

關於這些量度

下列清單說明本節中的教學課程和範本定義在自訂儀表板中建立的每個量度。

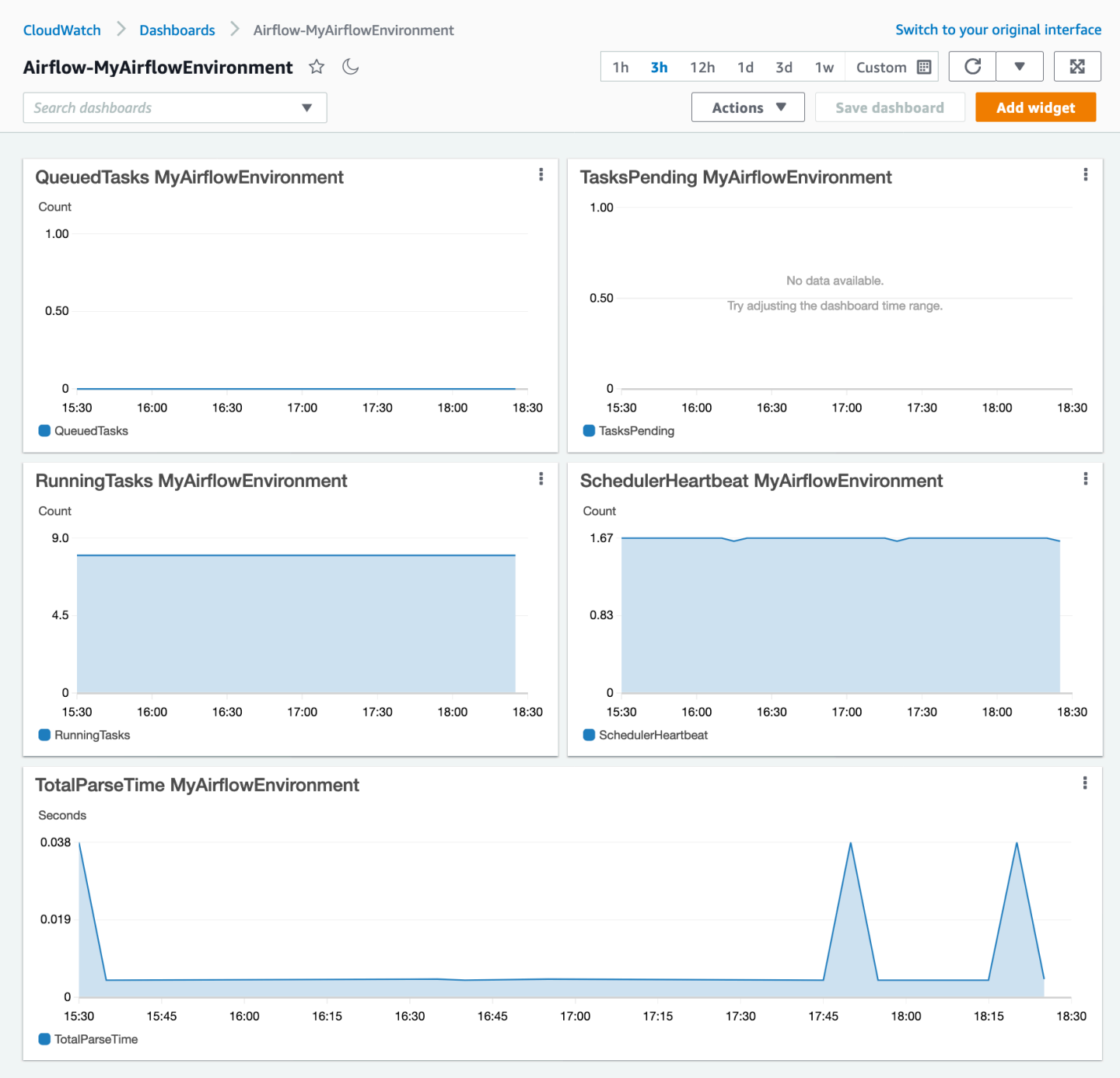
- QueuedTasks-處於佇列狀態的工作數目。對應至 Ap `executor.queued_tasks` ache 氣流量度。
- TasksPending-執行者中待處理的任務數量。對應至 Ap `scheduler.tasks.pending` ache 氣流量度。

> **Note**
>
> 不適用於阿帕奇氣流 v2.2 及以上版本。

- RunningTasks-執行程序中運行的任務數量。對應至 Ap `executor.running_tasks` ache 氣流量度。
- SchedulerHeartbeat-Apache 氣流在排程器工作上執行的簽入次數。對應至 `scheduler_heartbeat` Apache 氣流量度量。
- TotalParseTime-掃描和匯入所有 DAG 檔案一次所需的秒數。對應至 Ap `dag_processing.total_parse_time` ache 氣流量度。

關於儀表板

下圖展示了本節中教學課程和範本定義所建立的監督儀表板。



使用 AWS 教程

您可以使用以下 AWS 教學為目前部署的任何 Amazon MWAA 環境自動建立健康狀態儀表板。它還會針對所有 Amazon MWAA 環境中健康狀態不良的工作者和排程器活動訊號故障建立 CloudWatch 警示。

- CloudWatch Amazon MWAA 的儀表板自動化

使用 AWS CloudFormation

您可以使用此段落中的 AWS CloudFormation 樣板定義在中建立監督儀表板 CloudWatch，然後在 CloudWatch 主控台新增警示，以便在測量結果超過特定臨界值時接收通知。若要使用此範本定義建立堆疊，請參閱在AWS CloudFormation 主控台上建立堆疊。若要將警示新增至儀表板，請參閱使用警示。

```
AWSTemplateFormatVersion: "2010-09-09"
Description: Creates MWAA Cloudwatch Dashboard
Parameters:
  DashboardName:
    Description: Enter the name of the CloudWatch Dashboard
    Type: String
  EnvironmentName:
    Description: Enter the name of the MWAA Environment
    Type: String
Resources:
  BasicDashboard:
    Type: AWS::CloudWatch::Dashboard
    Properties:
      DashboardName: !Ref DashboardName
      DashboardBody:
        Fn::Sub: '{
              "widgets": [
                  {
                      "type": "metric",
                      "x": 0,
                      "y": 0,
                      "width": 12,
                      "height": 6,
                      "properties": {
                          "view": "timeSeries",
                          "stacked": true,
                          "metrics": [
                              [
                                  "AmazonMWAA",
                                  "QueuedTasks",
                                  "Function",
                                  "Executor",
                                  "Environment",
                                  "${EnvironmentName}"
                              ]
                          ],
```

```
                "region": "${AWS::Region}",
                "title": "QueuedTasks ${EnvironmentName}",
                "period": 300
            }
        },
        {
            "type": "metric",
            "x": 0,
            "y": 6,
            "width": 12,
            "height": 6,
            "properties": {
                "view": "timeSeries",
                "stacked": true,
                "metrics": [
                    [
                        "AmazonMWAA",
                        "RunningTasks",
                        "Function",
                        "Executor",
                        "Environment",
                        "${EnvironmentName}"
                    ]
                ],
                "region": "${AWS::Region}",
                "title": "RunningTasks ${EnvironmentName}",
                "period": 300
            }
        },
        {
            "type": "metric",
            "x": 12,
            "y": 6,
            "width": 12,
            "height": 6,
            "properties": {
                "view": "timeSeries",
                "stacked": true,
                "metrics": [
                    [
                        "AmazonMWAA",
                        "SchedulerHeartbeat",
                        "Function",
                        "Scheduler",
```

```
                        "Environment",
                        "${EnvironmentName}"
                    ]
                ],
                "region": "${AWS::Region}",
                "title": "SchedulerHeartbeat ${EnvironmentName}",
                "period": 300
            }
        },
        {
            "type": "metric",
            "x": 12,
            "y": 0,
            "width": 12,
            "height": 6,
            "properties": {
                "view": "timeSeries",
                "stacked": true,
                "metrics": [
                    [
                        "AmazonMWAA",
                        "TasksPending",
                        "Function",
                        "Scheduler",
                        "Environment",
                        "${EnvironmentName}"
                    ]
                ],
                "region": "${AWS::Region}",
                "title": "TasksPending ${EnvironmentName}",
                "period": 300
            }
        },
        {
            "type": "metric",
            "x": 0,
            "y": 12,
            "width": 24,
            "height": 6,
            "properties": {
                "view": "timeSeries",
                "stacked": true,
                "region": "${AWS::Region}",
                "metrics": [
```

```
                        [
                            "AmazonMWAA",
                            "TotalParseTime",
                            "Function",
                            "DAG Processing",
                            "Environment",
                            "${EnvironmentName}"
                        ]
                    ],
                    "title": "TotalParseTime  ${EnvironmentName}",
                    "period": 300
                }
            }
        ]
    }'
```

刪除指標和儀表板

如果您刪除 Amazon MWAA 環境，對應的儀表板也會一併刪除。 CloudWatch 指標會儲存十五 (15) 個月，無法刪除。主 CloudWatch 控台會將指標搜尋限制為上次擷取指標後的兩 (2) 週，以確保針對 Amazon MWAA 環境顯示最新的執行個體。要進一步了解，請參閱 Amazon CloudWatch 常見問答集。

後續步驟？

- 了解如何為您的環境建立可查詢 Amazon Aurora PostgreSQL 中繼資料庫的 DAG，並將自訂指標發佈到 CloudWatch 中。使用 DAG 寫入自訂指標CloudWatch

阿帕奇氣流 v2 環境指標 CloudWatch

Apache 氣流 v2 已經設定為收集 Amazon 管理工作流程的 Apache 氣流環境的 StatsD 指標，並將其傳送給 Amazon CloudWatch。Apache 氣流傳送量度的完整清單可在 Apache 氣流參考指南的「測量結果」頁面上找到。本頁說明中可用的 Apache Airflow 測量結果 CloudWatch，以及如何存取 CloudWatch 主控台中的測量結果。

內容

條款

命名空間

命名空間是 AWS 服務指 CloudWatch 標的容器。對於 Amazon MWAA，命名空間是亞馬遜 MWAA。

CloudWatch 度量

CloudWatch 量度代表特定於的一組時間順序的資料點。 CloudWatch

阿帕奇氣流指標

Apache 氣流特定的指標。

維度

維度是一組名稱值對，是指標身分的一部分。

單位

統計資料具有測量單位。對於 Amazon MWAA，單位包括計數、秒數和毫秒。對於 Amazon MWAA，系統會根據原始氣流指標中的單位來設定單位。

維度

本節說明中 Apache 氣流量度的 CloudWatch 維度群組 CloudWatch。

維度	描述			
天	表示特定的 Apache 氣流 DAG 名稱。			

維度	描述
DAG 檔案名	指出特定的 Apache 氣流 DAG 檔案名稱。
函式	此維度可用來改善中量度的分組 CloudWatch。
任務	指出排程器執行的 Apache 氣流 Job。始終具有 Job 的值。
運算子	表示特定的 Apache 氣流操作員。
集區	表示特定的 Apache 氣流工作者集區。
任務	表示特定的 Apache 氣流工作。
HostName	指出特定執行中 Apache 氣流程序的主機名稱。

在 CloudWatch 主控台中存取指標

本節說明如何存取特定 DAG 的 CloudWatch 效能測量結果。

檢視維度的效能測量結果

1. 在主控台上開啟「測量結果」頁 CloudWatch 面。
2. 使用「 AWS 地區」選取器選取您的地區。
3. 選擇亞馬遜名稱空間。
4. 在「所有量度」標籤中，選取維度。例如，DAG，環境。
5. 選擇維 CloudWatch 度的量度。例如，TaskInstanceSuccesses或TaskInstanceDuration。選擇圖表所有搜尋結果。

6. 選擇圖形測量結果頁籤，即可檢視 Apache Airflow 測量結果的效能統計資料，例如 DAG、環境、工作。

阿帕奇氣流指標可用於 CloudWatch

本節說明傳送至的 Apache 氣流量度和維度 CloudWatch。

Apache 氣流計數器

本節中的 Apache 氣流量度包含 Apache 氣流計數器的相關資料。

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
猛烈抨擊 Note 適用於阿帕奇氣流 v2.4.3 及以上版本。	斯拉 _ 錯過	計數	函數, 排程器	
失敗回復全部回復 Note 適用於阿帕奇氣流 v2.4.3 及以上版本。	回呼通知失敗 (__R)	計數	函數, 排程器	
更新 Note 適用於阿帕奇氣流 v2.6.3 及以上版本。	資料集. 更新	計數	函數, 排程器	
孤兒	資料集. 孤立	計數	函數, 排程器	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
Note 適用於阿帕奇氣流 v2.6.3 及以上版本。				
FailedCeleryTaskExecution Note 適用於阿帕奇氣流 v2.4.3 及以上版本。	特徵. 執行命令失敗	計數	功能, 芹菜	
FilePathQueueUpdateCount Note 適用於阿帕奇氣流 v2.6.3 及以上版本。	DAG_ 處理. 檔案路徑佇列更新計數	計數	函數, 排程器	
CriticalSectionBusy	排程器. 批判 _ 區段 _ 忙碌	計數	函數, 排程器	
DagBagSize	短劍尺寸	計數	功能, DAG 處理	
DagCallbackExceptions	異常回調	計數	星期日, 全部	
失敗的 SLA EmailAttempts	電子郵件通知失敗	計數	函數, 排程器	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
TaskInstanceFinished	完成。{匕首}{任務識別碼}。{狀態}	計數	日，{日} 任務，{任務 ID} 狀態，{狀態}	
JobEnd	{工作名稱} _ 結束	計數	Job，{工作名稱}	
JobHeartbeatFailure	{工作名稱} _ 心跳失敗	計數	Job，{工作名稱}	
JobStart	{工作名稱} _ 開始	計數	Job，{工作名稱}	
ManagerStalls	DAG_ 處理. 管理員檔	計數	功能, DAG 處理	
OperatorFailures	操作員失敗 _ {運營商名稱}	計數	運算子，{運算子名稱}	
OperatorSuccesses	運營商 _ 成功 _{運營商名稱}	計數	運算子，{運算子名稱}	
OtherCallbackCount Note 可在阿帕奇氣流 v2.6.3 及以上版本。	處理. 其他回呼計數	計數	函數, 排程器	
Processes	加工. 流程	計數	功能, DAG 處理	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
SchedulerHeartbeat	排程器心跳 (_)	計數	函數, 排程器	
StartedTaskInstances	開始。{匕首} {任務識別碼}	計數	星期日, 全部 工作、全部	
SlaCallbackCount	DAG 處理. 回呼計數 **Note** 適用於阿帕奇氣流 v2.6.3 及以上版本。	計數	函數, 排程器	
TasksKilledExternally	排程器. 任務. 殺死 _ 外部	計數	函數, 排程器	
TaskTimeoutError	提琴. 任務超時 _ 錯誤	計數	功能, 芹菜	
TaskInstanceCreatedUsingOperator	創建任務-{運算符名稱}	計數	運算子，{運算子名稱}	
TaskInstancePreviouslySucceeded	之前 (_ 成功)	計數	星期日, 全部 工作、全部	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
TaskInstanceFailures	TI_ 失敗	計數	星期日, 全部 工作、全部	
TaskInstanceSuccesses	成功	計數	星期日, 全部 工作、全部	
TaskRemovedFromDAG	從星期日移除工作。 {匕首}	計數	日, {日}	
TaskRestoredToDAG	工作 _ 還原到 _ 日。 {匕首}	計數	日, {日}	
TriggersSucceeded Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	觸發器. 成功	計數	功能、觸發	
TriggersFailed Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	觸發器失敗	計數	功能、觸發	
TriggersBlockedMainThread Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	觸發器. 封鎖執行緒	計數	功能、觸發	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
TriggerHeartbeat Note 適用於阿帕奇氣流 v2.8.1 及以上版本。	特里格爾心跳	計數	功能、加固型	
TaskInstanceCreatedUsingOperator	氣流. 任務 _ 實例 _ 創建 _ {operator _name} Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	計數	運營商，{operator _name}	
ZombiesKilled	殭屍殺死	計數	星期日，全部 工作、全部	

阿帕奇氣流計

本節中的 Apache 氣流量度包含 Apache 氣流量計的相關資料。

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
DAG FileRefreshError	DAG 檔案更新錯誤	計數	功能, DAG 處理	
ImportErrors	DAG 處理. 匯入錯誤	計數	功能, DAG 處理	
Exception Failures	智慧型感應器操作員. 例外失敗	計數	功能智慧感測器操作員	
ExecutedTasks	智能傳感器操作員. 執行任務	計數	功能智慧感測器操作員	
InfraFailures	智能傳感器操作員. 主機故障	計數	功能智慧感測器操作員	
LoadedTasks	智能傳感器操作員. 加載任務	計數	功能智慧感測器操作員	
TotalParseTime	DAG 處理. 總分析時間	秒鐘	功能, DAG 處理	
TriggeredDagRuns **Note** 可在阿帕奇氣流 v2.6.3 及以上版本。	資料集. 已觸發的日常運行	計數	函數, 排程器	
TriggersRunning	觸發器。運行。*{####}*	計數	功能、觸發 HostName, *{####}*	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
Note 可在阿帕奇氣流 v2.7.2 及更高版本。				
PoolDeferredSlots Note 可在阿帕奇氣流 v2.7.2 及更高版本。	池延期插槽。{pool_name}	計數	池，{池名稱}	
DAG FileProcessingLastRunSecondsAgo	最後一次執行第二個之前的 DAG 處理。{匕首檔案名稱}	秒鐘	DAG 檔案名稱，{DAG 檔案名稱}	
OpenSlots	執行程式. 開啟插槽	計數	函數，執行人	
OrphanedTasksAdopted	排程器. 孤兒 _ 任務.	計數	函數, 排程器	
OrphanedTasksCleared	排程器. 孤兒 _ 任務. 已清除	計數	函數, 排程器	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
PokedExceptions	智能傳感器操作員. 口袋 _ 異常	計數	功能智慧感測器操作員	
PokedSuccess	智能感應器操作員. 精靈 _ 成功	計數	功能智慧感測器操作員	
PokedTasks	智能傳感器操作員. 口袋妖怪任務	計數	功能智慧感測器操作員	
PoolFailures	池. 打開插槽。{池名稱}	計數	池，{池名稱}	
PoolStarvingTasks	池啟動任務。 {池名稱}	計數	池，{池名稱}	
PoolOpenSlots	池. 打開插槽。{池名稱}	計數	池，{池名稱}	
PoolQueuedSlots	池. 佇列插槽。{池名稱}	計數	池，{池名稱}	
PoolRunningSlots	池. 運行 _ 槽。{池名稱}	計數	池，{池名稱}	
Processor Timeouts	處理. 處理器逾時	計數	功能, DAG 處理	
QueuedTasks	執行器. 佇列工作	計數	函數，執行人	
RunningTasks	執行程式. 執行工作	計數	函數，執行人	
TasksExecutable	調度程序. 任務. 可執行	計數	函數, 排程器	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
TasksPending Note 不適用於阿帕奇氣流 v2.2 及以上版本。	調度程序. 任務. 擱置	計數	函數, 排程器	
TasksRunning	調度程序. 任務. 運行	計數	函數, 排程器	
TasksStarving	調度程序. 任務. 挨餓	計數	函數, 排程器	
TasksWithoutDagRun	調度程序. 任務. 沒有	計數	函數, 排程器	

阿帕奇氣流計時器

本節中的 Apache 氣流量度包含 Apache 氣流計時器的相關資料。

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
收集資料庫	收集資料庫	毫秒	功能, DAG 處理	
CriticalSectionDuration	排程器. 臨界 _ 區段持續時間	毫秒	函數, 排程器	
CriticalSectionQueryDuration	排程器. 關鍵區段查詢持續時間	毫秒	函數, 排程器	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
Note 適用於阿帕奇氣流 v2.5.1 及以上版本。				
DAG DependencyCheck	每日依賴檢查。{匕首}	毫秒	日，{日}	
DAG DurationFailed	持續時間。失敗。{匕首}	毫秒	日，{日}	
DAG DurationSuccess	持續時間。成功。{匕首}	毫秒	日，{日}	
DAG FileProcessingLastDuration	DAG 處理最後一個持續時間。{匕首檔案名稱}	秒鐘	DAG 檔案名稱，{DAG 檔案名稱}	
DAG ScheduleDelay	每日排程 _ 延遲。{匕首}	毫秒	日，{日}	
FirstTaskSchedulingDelay	每日。 第一個任務排程 _ 延遲	毫秒	日，{日}	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
Scheduler LoopDuration Note 適用於阿帕奇氣流 v2.5.1 及以上版本。	排程器. 排程器 _ 迴圈持續時間	毫秒	函數, 排程器	
TaskInstanceDuration	天。 {匕首} {任務 ID}. 持續時間	毫秒	日, {日} 任務 , {任務 ID}	
TaskInstanceQueuedDuration	天。 `{dag_id}`。 `{task_id}` . 佇列持續時間 Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	毫秒	日, {日} 任務 , {任務 ID}	

CloudWatch 公制	阿帕奇氣流公制	單位	維度	
TaskInstanceScheduledDuration Note 適用於阿帕奇氣流 v2.7.2 及以上版本。	天。{dag_id}。{task_id} . 排程持續時間 (_)	毫秒	日, {日} 任務，{任務 ID}	

選擇要報告的量度

您可以使用下列 Amazon MWAA 組態選項 CloudWatch，選擇要發送至 Apache 氣流或封鎖哪些 Apache 氣流指標：

- **metrics.metrics_allow_list**— 逗號分隔的前置詞清單，您可以用來選取您的環境發出哪些量度。 CloudWatch 如果您希望 Apache Airflow 不傳送所有可用的量度，而是選取元素子集，請使用此選項。例如 `scheduler,executor,dagrun`。
- **metrics.metrics_block_list**— 逗號分隔的前置字元清單，可篩選出以清單元素開頭的量度。例如 `scheduler,executor,dagrun`。

如果同時配置`metrics.metrics_allow_list`和`metrics.metrics_block_list`，Apache 氣流會忽略`metrics.metrics_block_list`。如果您設定`metrics.metrics_block_list`但未進行設定`metrics.metrics_allow_list`，Apache 氣流會過濾掉您在中指定的元素`metrics.metrics_block_list`。

> **Note**
>
> `metrics.metrics_allow_list`和組`metrics.metrics_block_list`態選項僅適用於 Apache 氣流 v2.6.3 及以上版本。對於以前版本的 Apache 氣流使用`metrics.statsd_allow_list`和`metrics.statsd_block_list`代替。

後續步驟？

- 探索用於在發佈環境運作狀態指標的 Amazon MWAA API 操作。PublishMetrics

適用於 Amazon MWAA 的容器、佇列和資料庫指標

除了 Apache Airflow 指標之外，您還可以使用以下方式監控適用於 Apache Airflow 環境的 Amazon 受管工作流程的基礎元件 CloudWatch，這些元件會收集原始資料，並將資料處理為可讀且接近即時的指標。有了這些環境指標，您將可以更清楚地瞭解關鍵效能指標，以協助您適當調整環境的大小，並針對工作流程進行問題偵錯。這些指標適用於 Amazon MWAA 上所有支援的 Apache 氣流版本。

Amazon MWAA 將為每個 Amazon 彈性容器服務 (Amazon ECS) 容器和 Amazon Aurora PostgreSQL 執行個體提供 CPU 和記憶體使用率，以及針對訊息數量和最早訊息的存留時間、資料庫連線的 Amazon 關聯式資料庫服務 (Amazon RDS) 指標、磁碟佇列深度、寫入操作、延遲和輸送量以及 Amazon RDS 代理指標提供亞馬遜簡單佇列服務 (Amazon SQS) 指標。這些指標也包括基礎背景工作、其他背景工作、排程器和 Web 伺服器的數量。

這些統計資料會保留 15 個月，因此您可以存取歷史資訊，並更好地瞭解排程失敗的原因，並針對潛在問題進行疑難排解。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

主題

- 條款
- 維度
- 在 CloudWatch 主控台中存取指標
- 指標清單

條款

命名空間

命名空間是 AWS 服務指 CloudWatch 標的容器。對於 Amazon MWAA，命名空間是. AWS/MWAA

CloudWatch 度量

CloudWatch 量度代表特定於的一組時間順序的資料點。 CloudWatch

維度

維度是一組名稱值對，是指標身分的一部分。

單位

統計資料具有測量單位。對於 Amazon MWAA，單位包括計數。

維度

本節說明中 Amazon MWAA 指標的 CloudWatch 維度分組。 CloudWatch

維度	描述
叢集	Amazon MWAA 環境用來執行 Apache 氣流元件的至少三個 Amazon ECS 容器的指標：排程器、工作者和網頁伺服器。
佇列	Amazon SQS 佇列的指標，用於將排程器與工作者分離。當工作人員閱讀這些消息時，它們被視為在飛行中，並且不適用於其他工作人員。如果其他 Worker 未在 12 小時可見性逾時之前刪除訊息，則訊息將可供其他 Worker 讀取。
資料庫	指標 Amazon MWAA 使用的 Aurora 叢集。這包括主要資料庫執行處理的測量結果和僅供讀取複本，以支援讀取作業。Amazon MWAA 會針對讀取器和寫入器執行個體發佈資料庫指標。

在 CloudWatch 主控台中存取指標

本節說明如何在中存取您的 Amazon MWAA 指標。 CloudWatch

檢視維度的效能測量結果

1. 在主控台上開啟「測量結果」頁 CloudWatch 面。
2. 使用「 AWS 地區」選取器選取您的地區。
3. 選擇 AWS/MW AA 命名空間。
4. 在「所有量度」標籤中，選擇維度。例如，叢集。
5. 選擇維 CloudWatch 度的量度。例如，NumSchedulers或 CPU 使用率。然後，選擇圖表所有搜索結果。
6. 選擇「圖形測量結果」頁籤以檢視效能測量結果。

指標清單

下表列出 Amazon MWAA 的叢集、佇列和資料庫服務指標。若要檢視直接從 Amazon ECS、Amazon SQS 或 Amazon RDS 發出的指標說明，請選擇相應的文件連結。

主題

- 叢集指標
- 資料庫指標
- 適用於 Amazon RDS 代理伺服器的資料庫指標 (如果有)
- 佇列指標

叢集指標

下列量度適用於每個排程器、基礎 Worker、其他 Worker 和 Web 伺服器。如需每個叢集指標的詳細資訊和說明，請參閱 Amazon ECS 開發人員指南中的可用指標和維度。

命名空間	指標	單位
AWS/MWAA	CPUUtilization	百分比
AWS/MWAA	MemoryUtilization	百分比

評估其他工作者實例的數量

您可以使用「叢集」維度下提供的元件測量結果 (如下列程序所述)，評估環境在指定時間點使用的其他 Worker。您可以透過繪製 CPU 利用率或MemoryUtilization量度的圖形，並將統計資料類型設定為「樣本計數」來執行此操作。結果值是AdditionalWorker元件的RUNNING工作總數。瞭解您的環境使用的其他 Worker 執行個體數量，可協助您評估環境如何 auto 擴展，並讓您最佳化其他 Worker 的數量。

1. 選擇 AWS/MW AA 命名空間。
2. 在所有測量結果頁籤中，選擇叢集維度。
3. 在「叢集」維度下 AdditionalWorker，選擇「CPU 使用率」或測量結MemoryUtilization果。
4. 在「圖形量度」標籤上，將「期間」設定為 1 分鐘，將「統計值」設定為「樣本計數」。

如需詳細資訊，請參閱 Amazon 彈性容器服*RUNNING*務開發人員指南中的服務任務計數。

資料庫指標

下列指標適用於每個資料庫執行個體，直到它被 Amazon RDS 代理取代為止。如需下列資料庫指標的詳細資訊和說明，請參閱 Amazon 關聯式資料庫服務使用者指南中的 Amazon RDS 指CloudWatch標。

命名空間	指標	單位
AWS/MWAA	CPUUtilization	百分比
AWS/MWAA	DatabaseConnections	計數
AWS/MWAA	DiskQueueDepth	計數
AWS/MWAA	FreeableMemory	位元組
AWS/MWAA	VolumeWriteIOPS	每五分鐘計數
AWS/MWAA	WriteIOPS	每秒計數
AWS/MWAA	WriteLatency	秒鐘
AWS/MWAA	WriteThroughput	每秒位元組數

適用於 Amazon RDS 代理伺服器的資料庫指標 (如果有)

如需下列資料庫代理指標的詳細資訊說明，請參閱 Amazon 關聯式資料庫服務使用者指南 CloudWatch 中的使用監控 Amazon RDS 代理指標。

命名空間	指標	單位
AWS/MWAA	ClientConnections	計數
AWS/MWAA	ClientConnectionsClosed	計數
AWS/MWAA	ClientConnectionsReceived	計數
AWS/MWAA	AvailabilityPercentage	百分比
AWS/MWAA	DatabaseConnection sCurrentlyInTransaction	計數
AWS/MWAA	DatabaseConnection sSetupFailed	計數
AWS/MWAA	DatabaseConnection sSetupSucceeded	計數
AWS/MWAA	DatabaseConnection Requests	計數
AWS/MWAA	DatabaseConnections	計數
AWS/MWAA	QueryDatabaseRespo nseLatency	微秒
AWS/MWAA	QueryRequests	計數
AWS/MWAA	QueryResponseLatency	微秒

佇列指標

如需下列佇列指標的單位和說明的詳細資訊，請參閱 Amazon 簡單佇列服務開發人員指南中的 Amazon SQS 可用指 CloudWatch 標。

命名空間	指標	單位
AWS/MWAA	ApproximateAgeOfOldestMessage	秒鐘
AWS/MWAA	ApproximateNumberOfMessagesNotVisible （正在運行的任務）	計數
AWS/MWAA	ApproximateNumberOfMessagesVisible （已排入佇列的工作）	計數

適用於 Apache 氣流的 Amazon 受管工作流程

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您（客戶）之間的共同責任。共同責任模型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 雲端中執行 AWS 服務的基礎架構。 AWS 還為您提供可以安全使用的服務。若要了解適用於 Amazon MWAA 的合規計劃，請參閱合規計劃AWS 服務範圍內的合規計劃AWS服務。
- 雲端安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您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 Apache 氣流使用 Amazon 受管工作流程時，如何套用共同的責任模型。它說明如何設定 Amazon MWAA 以符合安全性和合規目標。您也會學到如何使用其他可 AWS 協助您監控和保護 Amazon MWAA 資源的服務。

在本節中：

- 適用於 Apache 氣流程的 Amazon 受管工作流程中
- AWS Identity and Access Management
- 適用於 Apache 氣流的 Amazon 受管工作流程合規
- 適用於 Apache 氣流的 Amazon 受管工作流程
- Amazon MWAA 中的基礎設施安全性
- Amazon MWAA 中的組態和漏洞分析
- Amazon MWAA 上的安全最佳實務

適用於 Apache 氣流程的 Amazon 受管工作流程中

AWS 共同責任模型適用於 Apache Airflow 的 Amazon 受管工作流程中的資料保護。如此模型中所述，AWS 負責保護執行所有 AWS 雲端. 您負責維護在此基礎設施上託管內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的更多相關資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您使用 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者帳戶。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。建議使用 TLS 1.2 或更新版本。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及 AWS 服務中的所有預設安全性控制。
- 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 Simple Storage Service (Amazon Simple Storage Service (Amazon S3)) 的個人資料。

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欄位中，例如名稱欄位。這包括當您使用主控台、API 或開發套件使用 Amazon MWAA 或 AWS 其他 AWS 服務時。 AWS CLI您在標籤或自由格式欄位中輸入的任何資料都可能用於計費或診斷記錄。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。

Amazon MWAA 上的加密

下列主題說明 Amazon MWAA 如何保護您的靜態資料和傳輸中的資料。使用此資訊來了解 Amazon MWAA 如何與靜態資料整合以 AWS KMS 加密靜態資料，以及如何使用傳輸中的傳輸層安全性 (TLS) 通訊協定加密資料。

主題

靜態加密

在 Amazon MWAA 上，靜態資料是服務儲存到持續性媒體的資料。

您可以使用AWS 擁有的金鑰進行靜態資料加密，或選擇性地在建立環境時提供客戶管理的金鑰以進行額外的加密。如果您選擇使用客戶受管 KMS 金鑰，則該金鑰必須與您在環境中使用的其他 AWS 資源和服務位於相同的帳戶中。

若要使用客戶管理的 KMS 金鑰，您必須附加必要的原則聲明，才能 CloudWatch 存取您的金鑰原則。當您在環境中使用客戶受管 KMS 金鑰時，Amazon MWAA 會代表您附加四個授權。如需 Amazon MWAA 附加至客戶受管 KMS 金鑰的授權的詳細資訊，請參閱資料加密的客戶受管金鑰。

如果您未指定客戶受管 KMS 金鑰，Amazon MWAA 預設會使用 AWS 擁有的 KMS 金鑰來加密和解密您的資料。我們建議您使用 AWS 擁有的 KMS 金鑰來管理 Amazon MWAA 上的資料加密。

> **Note**
>
> 您需要支付 Amazon MWAA 上 AWS 擁有或客戶受管 KMS 金鑰的儲存和使用費用。如需詳細資訊，請參閱 AWS KMS 定價。

加密成品

您可以在建立 Amazon MWAA 環境時指定AWS 擁有的金鑰或客戶受管金鑰，以指定用於靜態加密的加密成品。Amazon MWAA 會將所需的授權新增至您指定的金鑰。

Amazon S3 — Amazon S3 資料會使用伺服器端加密 (SSE) 在物件層級加密。Amazon S3 加密和解密會在儲存 DAG 程式碼和支援檔案的 Amazon S3 儲存貯體上進行。物件上傳到 Amazon S3 時會加密，並在物件下載到您的 Amazon MWAA 環境時進行解密。根據預設，如果您使用的是客戶受管 KMS 金鑰，Amazon MWAA 會使用該金鑰來讀取和解密 Amazon S3 儲存貯體上的資料。

CloudWatch 記錄 — 如果您使用 AWS 擁有的 KMS 金鑰，傳送至記錄的 Apache Airflow CloudWatch 記錄會使用伺服器端加密 (SSE) 與 CloudWatch 記錄 AWS 擁有的 KMS 金鑰加密。如果您使用客戶管理的 KMS 金鑰，則必須將金鑰原則新增至 KMS 金鑰，以允許 CloudWatch 記錄使用您的金鑰。

Amazon SQS — Amazon MWAA 為您的環境建立一個 Amazon SQS 佇列。Amazon MWAA 使用 AWS 擁有的 KMS 金鑰或您指定的客戶管理 KMS 金鑰，使用伺服器端加密 (SSE) 來處理傳入佇列和傳出佇列的資料。無論您使用的是 AWS 擁有的還是客戶受管 KMS 金鑰，都必須將 Amazon SQS 許可新增至執行角色。

Aurora — Amazon MWAA 為您的環境建立一個 PostgreSQL 叢集。Aurora PostgreSQL 使用伺服器端加密 (SSE) 使用 AWS 擁有的或客戶受管的 KMS 金鑰來加密內容。如果您使用客戶受管 KMS 金鑰，Amazon RDS 至少會向金鑰新增兩個授權：一個用於叢集，另一個用於資料庫執行個體。如果您選擇在多個環境中使用客戶受管 KMS 金鑰，Amazon RDS 可能會建立其他授權。如需詳細資訊，請參閱 Amazon RDS 中的資料保護。

傳輸中加密

傳輸中的資料稱為在傳輸網路時可能會遭到攔截的資料。

傳輸層安全性 (TLS) 會加密環境 Apache 氣流元件和其他與 Amazon MWA 整合的 AWS 服務之間傳輸中的 Amazon MWAA 物件。例如 Amazon S3。如需 Amazon S3 加密的詳細資訊，請參閱使用加密保護資料。

使用客戶管理的金鑰進行加密

您可以選擇性地提供客戶管理的金鑰，以便在環境中進行資料加密。您必須在 Amazon MWAA 環境執行個體的相同區域中建立客戶受管 KMS 金鑰，以及用來存放工作流程資源的 Amazon S3 儲存貯體。如果您指定的客戶管理 KMS 金鑰與用來設定環境的帳戶位於不同的帳戶中，則您必須使用其 ARN 來指定金鑰以進行跨帳戶存取。如需有關建立金鑰的詳細資訊，請參閱AWS Key Management Service 開發人員指南中的建立金鑰。

支援的項目

AWS KMS 特徵	支援	
AWS KMS 金鑰識別碼或 ARN。	是	
AWS KMS 索引鍵別名。	否	
AWS KMS 多區域金鑰。	否	

使用授權進行加密

本主題說明 Amazon MWAA 代表您附加至客戶受管 KMS 金鑰的授權，以加密和解密您的資料。

運作方式

客戶管理的 KMS 金鑰支援兩種以資源 AWS KMS 為基礎的存取控制機制：金鑰原則和授權。

當權限大部分是靜態且在同步服務模式下使用時，就會使用金鑰原則。當需要更動態和細微的權限時，例如當服務需要為本身或其他帳戶定義不同的存取權限時，會使用授權。

Amazon MWAA 會使用四個授權政策，並將其附加至您的客戶受管 KMS 金鑰。這是因為環境從 CloudWatch 日誌、Amazon SQS 佇列、Aurora PostgreSQL 資料庫資料庫、秘密管理員機密、Amazon S3 儲存貯體和 DynamoDB 表格加密靜態資料所需的精細許可。

當您建立 Amazon MWAA 環境並指定客戶受管 KMS 金鑰時，Amazon MWAA 會將授權政策附加到您的客戶受管 KMS 金鑰。這些政策可讓 Amazon MWAA 中使`airflow.`*`region`*`}.amazonaws.com`用您的客戶受管 KMS 金鑰代表您加密 Amazon MWAA 擁有的資源。

Amazon MWAA 會代表您建立並附加其他授權至指定的 KMS 金鑰。這包括在您刪除環境時淘汰授權、使用客戶管理的 KMS 金鑰進行用戶端加密 (CSE)，以及針對需要在 Secrets Manager 中存取受客戶管理金鑰保護之密鑰的 AWS Fargate 執行角色的政策。

授予政策

Amazon MWAA 會代表您將以下資源型政策授權新增至客戶受管的 KMS 金鑰。這些政策允許受權者和主體 (Amazon MWAA) 執行政策中定義的動作。

授予 1：用於創建數據平面資源

```
{
            "Name": "mwaa-grant-for-env-mgmt-role-environment name",
            "GranteePrincipal": "airflow.region.amazonaws.com",
            "RetiringPrincipal": "airflow.region.amazonaws.com",
            "Operations": [
              "kms:Encrypt",
              "kms:Decrypt",
              "kms:ReEncrypt*",
              "kms:GenerateDataKey*",
              "kms:CreateGrant",
              "kms:DescribeKey",
              "kms:RetireGrant"
            ]
          }
```

授予 2：用於**ControllerLambdaExecutionRole**訪問

```
{
            "Name": "mwaa-grant-for-lambda-exec-environment name",
            "GranteePrincipal": "airflow.region.amazonaws.com",
            "RetiringPrincipal": "airflow.region.amazonaws.com",
            "Operations": [
              "kms:Encrypt",
              "kms:Decrypt",
              "kms:ReEncrypt*",
              "kms:GenerateDataKey*",
              "kms:DescribeKey",
              "kms:RetireGrant"
            ]
          }
```

授予 3：用於**CfnManagementLambdaExecutionRole**訪問

```
{
              "Name": " mwaa-grant-for-cfn-mgmt-environment name",
              "GranteePrincipal": "airflow.region.amazonaws.com",
              "RetiringPrincipal": "airflow.region.amazonaws.com",
              "Operations": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
              ]
            }
```

授予 4：用於 Fargate 執行角色以訪問後端機密

```
{
                "Name": "mwaa-fargate-access-for-environment name",
                "GranteePrincipal": "airflow.region.amazonaws.com",
                "RetiringPrincipal": "airflow.region.amazonaws.com",
                "Operations": [
                  "kms:Encrypt",
                  "kms:Decrypt",
                  "kms:ReEncrypt*",
                  "kms:GenerateDataKey*",
                  "kms:DescribeKey",
                  "kms:RetireGrant"
                ]
              }
```

將重要政策附加至客戶管理的金鑰

如果您選擇將自己的客戶受管 KMS 金鑰與 Amazon MWAA 搭配使用，則必須在金鑰上附加下列政策，以允許 Amazon MWAA 使用該金鑰來加密您的資料。

如果您用於 Amazon MWAA 環境的客戶受管 KMS 金鑰尚未設定為可使用 CloudWatch，您必須更新金鑰政策以允許加密 CloudWatch 記錄。如需詳細資訊，請參閱 < CloudWatch 使用 AWS Key Management Service 服務加密記錄檔資料 > 。

下列範例代表記 CloudWatch 錄的金鑰原則。取代為區域提供的範例值。

```
{
          "Effect": "Allow",
          "Principal": {
          "Service": "logs.us-west-2.amazonaws.com"
        },
        "Action": [
          "kms:Encrypt*",
          "kms:Decrypt*",
          "kms:ReEncrypt*",
          "kms:GenerateDataKey*",
          "kms:Describe*"
        ],
        "Resource": "*",
        "Condition": {
          "ArnLike": {
            "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-west-2:*:*"
            }
          }
        }
```

AWS Identity and Access Management

AWS Identity and Access Management (IAM) 是協助管理員安全地控制 AWS 資源存取的 AWS 服務。IAM 管理員控制哪些人可以通過身份驗證 (登入) 和授權 (具有許可)，以便將 Amazon 受管工作流程用於 Apache Airflow 資源。IAM 是一項無需額外付費即可使用的 AWS 服務。

本主題提供 Amazon MWAA 如何使用 AWS Identity and Access Management (IAM) 的基本概觀。若要進一步了解如何管理 Amazon MWAA 的存取權，請參閱。管理對 Amazon MWAA 環境的存取

目錄

- 針對 Apache 氣流身分識別和存取的 Amazon 受管工作流程
- Amazon MWAA 如何與 IAM 搭配使用

物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在 Amazon MWAA 中執行的工作。

服務使用者 — 如果您使用 Amazon MWAA 服務執行工作，則管理員會為您提供所需的登入資料和許可。當您使用更多 Amazon MWAA 功能完成工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 Amazon MWAA 中的功能，請參閱。針對 Apache 氣流身分識別和存取的 Amazon 受管工作流程

服務管理員 — 如果您負責公司的 Amazon MWAA 資源，您可能擁有完整的 Amazon MWAA 存取權。您的任務是判斷服務使用者應存取哪些 Amazon MWAA 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何將 IAM 與 Amazon MWAA 搭配使用，請參閱。Amazon MWAA 如何與 IAM 搭配使用

IAM 管理員 — 如果您是 IAM 管理員，您可能想要了解如何撰寫政策以管理 Amazon MWAA 存取權的詳細資訊。若要檢視可在 IAM 中使用的 Amazon MWAA 身分型政策範例，請參閱。Amazon MWAA 身分識別型政策範例

使用身分來驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。 AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。 AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 使用者指南中的如何登入您 AWS 帳戶的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，在中暫時擔任 IAM 角色。 AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。

- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源（而不是使用角色作為代理）。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源類型政策的差異。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務。
  - 服務連結角色 — 服務連結角色是連結至. AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中， AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS ，當與身分識別或資源相關聯時，會定義其權限。 AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶. 受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。主參與者可以包括帳戶、使用者、角色、同盟使用者或。 AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。 AWS WAF若要進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 實體許可範圍。
- 服務控制策略（SCP）— SCP 是 JSON 策略，用於指定中組織或組織單位（OU）的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的 SCP 運作方式。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
```

```
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

針對 Apache 氣流身分識別和存取的 Amazon 受管工作流程

使用下列資訊協助您診斷和修正使用 Amazon MWAA 和 IAM 時可能遇到的常見問題。

我沒有授權在 Amazon MWAA 中執行操作

如果 AWS Management Console 告訴您您沒有執行動作的授權，則您必須聯絡管理員以尋求協助。您的管理員是提供您使用者名稱和密碼的人員。

我沒有授權執行 iam：PassRole

如果您收到未獲授權執行`iam:PassRole`動作的錯誤訊息，則必須更新您的政策以允許您將角色傳遞給 Amazon MWAA。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者`marymajor`嘗試使用主控台在 Amazon MWAA 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

我想要允許我 AWS 帳戶以外的人員存取我的 Amazon MWAA 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 Amazon MWAA 是否支援這些功能，請參閱。Amazon MWAA 如何與 IAM 搭配使用
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱《IAM 使用者指南》中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的提供第三方 AWS 帳戶 擁有的存取權。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 IAM 角色與資源型政策的差異。

Amazon MWAA 如何與 IAM 搭配使用

Amazon MWAA 使用以身分識別為基礎的政策，將許可授與 Amazon MWAA 動作和資源。如需可用來控制 Amazon MWAA 資源存取權限之自訂 IAM 政策的建議範例，請參閱。the section called “存取 Amazon MWAA 環境”

若要取得 Amazon MWAA 和其他 AWS 服務如何與 IAM 搭配運作的高階檢視，請參閱 IAM 使用者指南中的與 IAM 搭配使用的AWS 服務。

Amazon MWAA 身分識別型政策

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。Amazon MWAA 支援特定動作、資源和條件金鑰。

下列步驟說明如何使用 IAM 主控台建立新的 JSON 政策。此政策提供 Amazon MWAA 資源的唯讀存取權。

若要使用 JSON 政策編輯器來建立政策

1. 登入 AWS Management Console 並開啟身分與存取權管理主控台，網址為 https://console.aws.amazon.com/iam/。
2. 在左側的導覽窗格中，選擇 Policies (政策)。

   如果這是您第一次選擇 Policies (政策)，將會顯示 Welcome to Managed Policies (歡迎使用受管政策) 頁面。選擇 Get Started (開始使用)。
3. 在頁面頂端，選擇 Create policy (建立政策)。
4. 在政策編輯器中，選擇 JSON 選項。
5. 輸入下列 JSON 政策文件：

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "airflow:ListEnvironments",
                "airflow:GetEnvironment",
                "airflow:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

6. 選擇下一步。

> **Note**
>
> 您可以隨時切換視覺化與 JSON 編輯器選項。不過，如果您進行變更或在視覺化編輯器中選擇下一步，IAM 就可能會調整您的政策結構，以便針對視覺化編輯器進行最佳化。如需詳細資訊，請參閱 IAM 使用者指南中的調整政策結構。

7. 在檢視與建立頁面上，為您正在建立的政策輸入政策名稱與描述 (選用)。檢視此政策中定義的許可，來查看您的政策所授予的許可。

8. 選擇 Create policy (建立政策) 儲存您的新政策。

若要了解您在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的 JSON 政策元素參考。

動作

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

政策陳述式必須包含 `Action` 或 `NotAction` 元素。`Action` 元素會列出政策允許的動作。`NotAction` 元素會列出不允許的動作。

針對 Amazon MWAA 定義的動作反映了您可以使用 Amazon MWAA 執行的任務。Detective 中的政策動作具有以下前綴：`airflow:`。

您可以使用萬用字元 (*) 來指定多個動作。您可以將所有以字結尾的動作授與存取權，而不是個別列出這些動作，例如，`environment`。

若要查看 Amazon MWAA 動作清單，請參閱 IAM 使用者指南中的 Amazon 管理工作流程針對 Apache 氣流定義的動作。

Amazon MWAA 身分識別型政策範例

若要檢視 Amazon MWAA 政策，請參閱。管理對 Amazon MWAA 環境的存取

依預設，IAM 使用者和角色沒有建立或修改 Amazon MWAA 資源的權限。他們也無法使用 AWS Management Console AWS CLI、或 AWS API 執行工作。

IAM 管理員必須建立 IAM 政策，授予使用者和角色在指定資源上執行特定 API 作業的所需許可。管理員接著必須將此類政策附加至需要此類許可的 IAM 使用者或群組。

> **Important**
>
> 我們建議您使用 IAM 角色和臨時登入資料來提供 Amazon MWAA 資源的存取權。避免將許可點直接附加到 IAM 使用者。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱 IAM 使用者指南中的在 JSON 索引標籤上建立政策。

主題

- 政策最佳實務
- 使用 Amazon MWAA 主控台
- 允許使用者檢視自己的許可

政策最佳實務

以身分識別為基礎的政策決定某人是否可以在您的帳戶中建立、存取或刪除 Amazon MWAA 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始授與使用者和工作負載的權限，請使用可授與許多常見使用案例權限的AWS 受管理原則。它們在您的 AWS 帳戶. 建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

使用 Amazon MWAA 主控台

若要使用 Amazon MWAA 主控台，使用者或角色必須能夠存取相關動作，這些動作與 API 中的對應動作相符。

若要檢視 Amazon MWAA 政策，請參閱。管理對 Amazon MWAA 環境的存取

允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

適用於 Apache 氣流的 Amazon 受管工作流程合規

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱AWS 服務 遵循規範計劃方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱AWS 規範計劃AWS。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載中的報告中的 AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- 安全性與合規性快速入門指南 — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- 在 Amazon Web Services 上架構 HIPAA 安全性與合規性 — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 用程式。

> **Note**
>
> 並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 HIPAA 資格服務參照。

- AWS 合規資源AWS — 此工作簿和指南集合可能適用於您的產業和所在地。
- AWS 客戶合規指南 — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- 使用AWS Config 開發人員指南中的規則評估資源 — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 Security Hub controls reference。
- AWS Audit Manager— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

適用於 Apache 氣流的 Amazon 受管工作流程

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需區域和可用區域的相關 AWS 資訊，請參閱AWS 全域基礎結構。

Amazon MWAA 中的基礎設施安全性

作為一項受管服務，適用於 Apache Airflow 的 Amazon 受管工作流程受到 AWS 全球網路安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱AWS 雲端安全 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱安全性支柱架構良 AWS 好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫透過網路存取 Amazon MWAA。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

Amazon MWAA 中的組態和漏洞分析

配置和 IT 控制是與您（我們的客戶）AWS 之間的共同責任。

Amazon 管理的 Apache 氣流工作流程會定期修補和升級您環境中的 Apache 氣流。您應確定 VPC 已使用適當的存取原則。

如需詳細資訊，請參閱以下 資源：

- 適用於 Apache 氣流的 Amazon 受管工作流程合規
- 共同的責任模型
- Amazon Web Services：安全程序概觀

- Amazon MWAA 中的基礎設施安全性
- Amazon MWAA 上的安全最佳實務

Amazon MWAA 上的安全最佳實務

Amazon MWAA 提供許多安全功能，可在您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。

- 使用最低權限的權限原則。僅授與使用者執行工作所需的資源或動作的權限。
- 用 AWS CloudTrail 於監視您帳戶中的使用者活動。
- 確保 Amazon S3 儲存貯體政策和物件 ACL 從關聯的 Amazon MWAA 環境向使用者授與許可，以便將物件放入儲存貯體。這可確保具有將工作流程新增至值區之權限的使用者也具有在 Airflow 中執行工作流程的權限。
- 使用與 Amazon MWAA 環境關聯的 Amazon S3 儲存貯體。您的 Amazon S3 存儲桶可以是任何名稱。請勿將其他物件儲存在值區中，或將值區與其他服務搭配使用。

Apache 氣流中的安全性最佳做法

阿帕奇氣流不是多租戶。雖然有存取控制措施可將某些功能限制為 Amazon MWAA 實作的特定使用者，但 DAG 建立者確實可以撰寫 DAG 來變更 Apache Airflow 使用者權限，並與基礎中繼資料庫互動。

在 Amazon MWAA 上使用 Apache Airflow 時，我們建議您執行下列步驟，以確保您環境的中繼資料庫和 DAG 安全無虞。

- 假設可以寫入環境的使用者也可以存取 Amazon MWAA 執行角色或 Apache Airflow 連線存取任何可存取的項目，對具有 DAG 寫入存取權的不同團隊使用不同的環境，或將檔案新增至 Amazon S3 `/dags` 資料夾的功能。
- 不提供直接的 Amazon S3 DAG 資料夾存取。而是使用 CI/CD 工具將 DAG 寫入 Amazon S3，並透過驗證步驟確保 DAG 程式碼符合您團隊的安全準則。
- 防止使用者存取您環境的 Amazon S3 儲存貯體。而是使用 DAG 工廠，該工廠根據存放 DAG 的亞馬遜 MWAA Amazon S3 儲存貯體存放在不同位置的 YAML、JSON 或其他定義檔案來產生 DAG。
- 將密碼儲存在 Secrets Manager 中。雖然這不會阻止可以寫入 DAG 的使用者讀取密碼，但會防止他們修改您環境使用的密碼。

偵測 Apache 氣流使用者權限的變更

您可以使用 CloudWatch 記錄深入解析來偵測 DAG 變更 Apache 氣流使用者權限的發生次數。為此，您可以在其中一個 DAG 變更 Apache Airflow 使用者權限時，使用 EventBridge 排程規則、Lambda 函數和 CloudWatch 日誌深入解析來傳遞通知給 CloudWatch 指標。

必要條件

要完成以下步驟，您將需要以下內容：

- 具有在日誌層級啟用所有 Apache 氣流日誌類型的 Amazon MWAA 環境。INFO如需詳細資訊，請參閱 the section called “檢視氣流記錄”。

若要設定 Apache 氣流使用者權限變更的通知

1. 建立 Lambda 函數，針對五個 Amazon MWAA 環境 CloudWatch 日誌群組 (DAGProcessing、、Scheduler和Worker) 執行下列日誌見解查詢字串。Task WebServer

   ```
   fields @log, @timestamp, @message | filter @message like "add-role" | stats count()
    by @log
   ```

2. 建立依排程執行的 EventBridge 規則，並將您在上一步中建立的 Lambda 函數做為規則的目標。使用 Cron 或速率運算式設定您的排程，以定期執行。

適用於 Apache 氣流的 Amazon 氣流管理工作流程

本頁說明 Apache 氣流支援的 Amazon 氣流管理工作流程版本，以及我們建議升級至最新版本的策略。

主題

關於 Amazon MWAA 版本

Amazon MWAA 會建置容器映像，將 Apache 氣流發行版本與其他常見的二進位檔案和 Python 程式庫結合在一起。此映像會針對您指定的版本使用 Apache 氣流基本安裝。建立環境時，請指定要使用的映像版本。建立環境後，它會繼續使用指定的映像版本，直到您將其升級到更新的版本為止。

最新版本

Amazon MWAA 支持多個阿帕奇氣流版本。如果您在建立環境時未指定映像版本，Amazon MWAA 會使用最新支援的 Apache 氣流版本建立環境。

阿帕奇氣流版本

Amazon 管理的 Apache 氣流程支援以下 Apache 氣流版本。

> **Note**
>
> - 從 Apache 氣流 v2.2.2 開始，Amazon MWAA 支援直接在 Apache 氣流網頁伺服器上安裝 Python 需求、供應商套件和自訂外掛程式。

- 從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一`--constraint`份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。

  如需在需求檔案中設定條件約束的詳細資訊，請參閱安裝 Python 相依性。

阿帕奇氣流版	阿帕奇氣流指南	阿帕奇氣流限制	Python 版本
v2.8.1	阿帕奇氣流 v2.8.1 參考指南	阿帕奇氣流 v2.8.1 約束文件	Python
v2.7.2	阿帕奇氣流 v2.7.2 參考指南	阿帕奇氣流 v2.7.2 約束文件	Python
v2.6.3	阿帕奇氣流 v2.6.3 參考指南	阿帕奇氣流 v2.6.3 約束文件	Python
v2.5.1	阿帕奇氣流 v2.5.1 參考指南	阿帕奇氣流 v2.5.1 約束文件	Python
v2.4.3	阿帕奇氣流 v2.4.3 參考指南	阿帕奇氣流 v2.4.3 約束文件	Python
v2.2.2	阿帕奇氣流 v2.2.2 參考指南	阿帕奇氣流 v2.2.2 約束文件	Python 3.7
V2.0.2	阿帕奇氣流 v2.0.2 參考指南	阿帕奇氣流 v2.0.2 約束文件	Python 3.7

如需有關遷移自我管理的 Apache Airflow 部署或遷移現有 Amazon MWAA 環境的詳細資訊 (包括備份中繼資料資料庫的說明)，請參閱 Amazon M WAA 遷移指南。

阿帕奇氣流組件

本節說明 Amazon MWAA 上每個 Apache 氣流版本可用的 Apache 氣流排程器和工作程式數量，並提供 Apache 氣流的主要功能清單，指出支援各項功能的版本。

排程器

阿帕奇氣流版	排程器 (預設值)	排程器 (分鐘)	排程器 (最大值)	
阿帕奇氣流 v2 及以上	2	2	5	

工作程序

氣流版本	工人 (分鐘)	工人 (最大值)	工作者 (預設值)	
阿帕奇氣流 V2	1	25	10	

升級阿帕奇氣流版本

Amazon MWAA 支援次要版本升級。這意味著您可以將環境從版本升級`x.1.z`到`x.2.z`，但不能升級到新的主要版本，例如從`1.y.z`到`2.y.z`。

> **Note**
>
> 您無法降級您環境的 Apache 氣流版本。

如需更新工作流程資源以及將環境升級至新版本的詳細資訊和詳細指示，請參閱the section called “升級版本”。

阿帕奇氣流棄用版本

下表列出 Amazon MWAA 中已淘汰的 Apache Airflow 版本，以及每個版本的初始版本和終止支援日期。如需有關移轉至較新版本的詳細資訊，請參閱 Amazon MWAA 移轉指南。

阿帕奇氣流版	阿帕奇氣流發布日期	Amazon MWAA 可用性日期	Amazon MWAA 有限的支援日期	Amazon MWAA 支援結束日期
V1.10.12	2020 年 8 月 25 日	2020 年 11 月 24 日	2023 年 8 月 21 日	2024年2月21 日
V2.0.2	2021 年 4 月 19 日	2021 年 5 月 25 日	2023年11月23日	2024年4月29 日
v2.2.2	2021 年 11 月 15 日	2022 年一月二十七日	2024年1月25日	2024年6月27日

Apache 氣流版本支持和常見問題

根據 Apache 氣流社群發行程序和版本政策，Amazon MWAA 致力於在任何給定時間支援至少三個次要版本的 Apache 氣流。我們將在支援結束日期前至少 90 天宣布指定 Apache Airflow 次要版本的支援結束日期。

常見問答集

問：Amazon MWAA 支援 Apache 氣流版本有多長時間？

答：Amazon MWAA 在首次推出後支援至少 12 個月的 Apache 氣流次要版本。

問：當 Amazon MWAA 上 Apache 氣流版本的支援終止時，是否會通知我？

答案：是。如果您帳戶中 AWS Health Dashboard 有任何 Amazon MWAA 環境執行即將結束支援的版本，Amazon MWAA 會在支援結束日期之後寄出通知。

問：在有限的支援日期會發生什麼事？

答：在有限的支援日期，您無法再使用相關版本建立新的 Amazon MWAA 環境。您現有的環境將繼續可用，直到支援日期結束為止。

問：終止支援日期會發生什麼事？

答：在支援結束日期後，您將繼續能夠存取現有的 Amazon MWAA 環境，這些環境可執行相關的、已取代的 Apache Airflow 版本，風險由您自行承擔。如需在 Amazon MWAA 上升級到較新版本的 Apache 氣流的說明，請參閱 Amazon MWAA 遷移指南。

Important

您有責任將您的 Amazon MWAA 版本保持在最新狀態。 AWS 敦促所有客戶將 Amazon MWAA 環境升級至最新版本，以便享有最新的安全性、隱私權和可用性保護措施。如果您在不受支援的版本或軟體上操作環境超過棄用日期 (稱為舊版)，則您將面臨更大的安全性、隱私權和作業風險 (包括停機事件) 的可能性。透過在舊版本上操作 Amazon MWAA 環境，即表示您確認瞭解並明知承擔這些風險，並且同意盡快完成升級至最新版本。在舊版本上繼續運作您的環境需遵守有關您使用 AWS 服務的協議。

舊版本不被視為一般可用，也 AWS 不再提供對舊版本的支援。因此，如果 AWS 確定舊版本對服務、其關聯公司或任何其他第三方構成安全或責任風險，或有傷害風險，則 AWS 可能隨時對任何舊版本的訪問或使用設置限制。 AWS您決定繼續在舊版本上執行工作負載可能會導致您的內容無法使用、損毀或無法復原。在舊版本上執行的環境受服務等級協定 (SLA) 例外規範。

在舊版本上執行的環境和相關軟體可能包含錯誤、錯誤、瑕疵和有害元件。因此，儘管協議中有任何相反的信息，或服務條款，仍按原樣 AWS 提供舊版本。

如需共同責任模型 AWS的詳細資訊，請參閱 AWS Well-Architected 的架構中的共同責任。

Amazon Managed Work(Amazon Managed Work與配額

Amazon Managed Work(Amazon Managed Work) 具有下列服務配額和端配額 服務配額或也稱為限制，即您AWS的服務資源或操作數目最大值。

內容

服務端點

若要檢視 Amazon MWAA 的端點清單，請參閱適用於 Apache 氣流端點和配額的亞馬遜受管工作流程。

Service Quotas

配額名稱	描述	預設配額	可調整
環境	每個區域的 Amazon MWAA 環境數目上限。	10	是
每個環境	每個 Amazon MWAA 環境的工作者數目上限。	25	是

增加配額

您可以透過提交配額增加要求來要求提高可調配額。

Amazon MWAA 常見問題

本頁說明使用 Amazon 受管工作流程進行 Apache 氣流時可能遇到的常見問題。

內容

- 為什麼我的 DAG 文件沒有被 Apache 氣流拿起？
- 我可以從環境requirements.txt中移除plugins.zip或嗎？
- 為什麼我在 Apache 氣流 v2.0.2 管理插件菜單中看不到我的插件？
- 我可以使用 AWS Database Migration Service (DMS) 操作員嗎？

支援的版本

Amazon MWAA 對阿帕奇氣流 V2 的支持是什麼？

若要了解 Amazon MWAA 支援的功能，請參閱。適用於 Apache 氣流的 Amazon 氣流管理工作流程

為什麼不支援舊版本的 Apache 氣流？

我們只支持最新的（截至發射）Apache 氣流版本 Apache 氣流 v1.10.12 由於舊版本的安全問題。

我應該使用什麼版本？

Amazon 管理的 Apache 氣流程支援以下 Apache 氣流版本。

> **Note**
>
> - 從 Apache 氣流 v2.2.2 開始，Amazon MWAA 支援直接在 Apache 氣流網頁伺服器上安裝 Python 需求、供應商套件和自訂外掛程式。
> - 從 Apache 氣流 v2.7.2 開始，您的需求文件必須包含一`--constraint`份聲明。如果您未提供限制，Amazon MWAA 會為您指定一個限制，以確保需求中列出的套件與您正在使用的 Apache Airflow 版本相容。
>
>   如需在需求檔案中設定條件約束的詳細資訊，請參閱安裝 Python 相依性。

阿帕奇氣流版	阿帕奇氣流指南	阿帕奇氣流限制	Python 版本
v2.8.1	阿帕奇氣流 v2.8.1 參考指南	阿帕奇氣流 v2.8.1 約束文件	Python
v2.7.2	阿帕奇氣流 v2.7.2 參考指南	阿帕奇氣流 v2.7.2 約束文件	Python

阿帕奇氣流版	阿帕奇氣流指南	阿帕奇氣流限制	Python 版本
v2.6.3	阿帕奇氣流 v2.6.3 參考指南	阿帕奇氣流 v2.6.3 約束文件	Python
v2.5.1	阿帕奇氣流 v2.5.1 參考指南	阿帕奇氣流 v2.5.1 約束文件	Python
v2.4.3	阿帕奇氣流 v2.4.3 參考指南	阿帕奇氣流 v2.4.3 約束文件	Python
v2.2.2	阿帕奇氣流 v2.2.2 參考指南	阿帕奇氣流 v2.2.2 約束文件	Python 3.7
V2.0.2	阿帕奇氣流 v2.0.2 參考指南	阿帕奇氣流 v2.0.2 約束文件	Python 3.7

如需有關遷移自我管理的 Apache Airflow 部署或遷移現有 Amazon MWAA 環境的詳細資訊 (包括備份中繼資料資料庫的說明)，請參閱 Amazon M WAA 遷移指南。

Amazon MWAA 使用什麼版本？**pip**

對於執行阿帕奇氣流 v1.10.12 的環境，Amazon MWAA 會安裝 21.1.2 版。`pip`

> **Note**
>
> Amazon MWAA 不會`pip`針對阿帕奇氣流 v1.10.12 環境升級。

對於執行 Apache 氣流 v2 及以上版本的環境，Amazon MWAA 會安裝 21.3.1 `pip` 版。

使用案例

我應該什麼時候使用 AWS Step Functions vs. Amazon MWAA？

1. 您可以使用「Step Functions」來處理個別客戶訂單，因為「Step Functions」可以調整以滿足一筆訂單或一百萬個訂單的需求。

2. 如果您正在執行隔夜工作流程來處理前一天的訂單，則可以使用 Step Functions 或 Amazon MWAA。Amazon MWAA 可讓您使用開放原始碼選項，從您正在使用的 AWS 資源中抽象化工作流程。

環境規格

每個環境有多少工作儲存空間？

任務存儲限制為 10 GB，並由 Amazon ECS Fargate 1.3 指定。RAM 的容量取決於您指定的環境類別。如需環境類別的詳細資訊，請參閱設定 Amazon MWAA 環境類別。

Amazon MWAA 環境使用的預設作業系統為何？

Amazon MWAA 環境是在執行 Amazon Linux AMI 的執行個體上建立的。

我可以在我的 Amazon MWAA 環境中使用自訂映像檔嗎？

不支援自訂影像。Amazon MWAA 使用在 Amazon Linux AMI 上構建的圖像。Amazon MWAA 會根據您`pip3 -r install`為環境新增至 Amazon S3 儲存貯體的 requirements.txt 檔案中指定的需求執行，以安裝其他需求。

Amazon MWAA HIPAA 是否符合規定？

Amazon MVAA 符合 Health 保險可攜性和責任法案 (HIPAA) 資格。如果您有一份 HIPAA 商業夥伴增補合約 (BAA) AWS，則可以使用 Amazon MWAA 處理在 2022 年 11 月 14 日或之後建立的環境上處理受保護 Health 資訊 (PHI) 的工作流程。

Amazon MWAA 是否支援競價型執行個體？

Amazon MWAA 目前不支援 Apache 氣流的隨需 Amazon EC2 競價型執行個體類型。但是，Amazon MWAA 環境可以在上觸發競價型執行個體，例如 Amazon EMR 和 Amazon EC2。

Amazon MWAA 是否支援自訂網域？

若要能夠為 Amazon MWAA 主機名稱使用自訂網域，請執行下列其中一個動作：

- 對於具有公有網路伺服器存取權的 Amazon MWAA 部署，您可以使用 Amazon CloudFront 搭配 Lambda @Edge 將流量導向您的環境，並將自訂網域名稱對應至。 CloudFront如需為公用環境設

定自訂網域的詳細資訊和範例，請參閱 Amazon MWAA 範例儲存庫中公用 Web 伺服器的 Amazon MWAA 自訂網域範例。 GitHub

- 對於具有私有 Web 伺服器存取權的 Amazon MWAA 部署，您可以使用 Application Load Balancer 器 (ALB) 將流量導向 Amazon MWAA，並將自訂網域名稱對應至 ALB。如需詳細資訊，請參閱 the section called “使用 Load Balancer (進階)”。

我可以 SSH 進入我的環境嗎？

雖然 Amazon MWAA 環境不支援 SSH，但是可以使用 DAG 來執行 bash 命令，使用. `BashOperator` 例如：

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago
with DAG(dag_id="any_bash_command_dag", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
      cli_command = BashOperator(
          task_id="bash_command",
          bash_command="{{ dag_run.conf['command'] }}"
      )
```

若要在 Apache 氣流使用者介面中觸發 DAG，請使用：

```
{ "command" : "your bash command"}
```

為什麼 VPC 安全性群組需要自我參照規則？

透過建立自我參考規則，您將來源限制在 VPC 中的相同安全群組，而且並非對所有網路開放。如需進一步了解，請參閱 the section called “您的 VPC 安全政策”。

我可以在 IAM 中隱藏不同群組的環境嗎？

您可以透過在中指定環境名稱來限制存取 AWS Identity and Access Management，但是可見性篩選在 AWS 主控台中無法使用 — 如果使用者可以看到一個環境，則可以看到所有環境。

我可以在 Apache 氣流工作者上存儲臨時數據嗎？

您的 Apache 氣流操作員可以將臨時數據存儲在工作人員上。Apache 氣流工作者可以針對您的環境存/tmp取 Fargate 容器中的暫存檔案。

> **Note**
>
> 根據 Amazon ECS Fargate 1.3，總任務存儲空間限制為 10 GB。不保證後續任務將在相同的 Fargate 容器實例上運行，該實例可能使用不同的/tmp文件夾。

我可以指定超過 25 個 Apache 氣流工作者嗎？

是。雖然您可以在 Amazon MWAA 主控台上指定最多 25 個 Apache 氣流工作者，但您可以透過要求增加配額，在一個環境中設定最多 50 個。如需詳細資訊，請參閱請求增加配額。

Amazon MWAA 是否支援共用的 Amazon VPC 或共用子網路？

Amazon MWAA 不支援共用的 Amazon VPC 或共用子網路。您在建立環境時選取的 Amazon VPC 應由嘗試建立環境的帳戶所擁有。不過，您可以將流量從 Amazon MWAA 帳戶中的 Amazon VPC 路由到共用 VPC。如需詳細資訊，並查看將流量路由到共用 Amazon VPC 的範例，請參閱 Amazon VPC 傳輸閘道指南中的集中式傳出路由至網際網路。

指標

使用哪些指標來決定是否要擴展 Worker？

Amazon MWAA 會監控QueuedTasks和輸入， CloudWatch 以判斷是否要RunningTasks在您的環境中擴展 Apache 氣流工作者。如需進一步了解，請參閱 監控和指標。

我可以在中建立自訂指標 CloudWatch嗎？

不在 CloudWatch 主控台上。不過，您可以建立在中寫入自訂指標的 DAG CloudWatch。如需詳細資訊，請參閱 the section called “使用 DAG 寫入自訂指標”。

DAG，操作員，連接和其他問題

我可以使用`PythonVirtualenvOperator`嗎？

Amazon MWAA 並未明確支援，但您可以建立使用. `PythonVirtualenvOperator` `PythonVirtualenvOperator` 如需程式碼範例，請參閱 the section called “自定義插件來修補 PythonVirtualenvOperator ”。

Amazon MWAA 識別新的 DAG 文件需要多長時間？

DAG 會定期從 Amazon S3 儲存貯體同步到您的環境。如果您新增一個新的 DAG 檔案，Amazon MWAA 大約需要 300 秒才能開始使用新檔案。如果您更新現有的 DAG，Amazon MWAA 大約需要 30 秒才能辨識您的更新。

這些值 (新 DAG 為 300 秒)，對於現有 DAG 的更新 30 秒，對應於 Apache 氣流組態選項，分別對應於 Apache 氣流組態選項dag_dir_list_interval。min_file_process_interval

為什麼我的 DAG 文件沒有被 Apache 氣流拿起？

以下是此問題的可能解決方案：

1. 檢查您的執行角色是否具有足夠的 Amazon S3 儲存貯體許可。如需進一步了解，請參閱 Amazon MWAA 執行角色。
2. 檢查 Amazon S3 儲存貯體是否已設定區塊公開存取並啟用版本控制。如需進一步了解，請參閱 為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯。
3. 驗證 DAG 檔案本身。例如，請確定每個 DAG 都有唯一的 DAG 識別碼。

我可以從環境`requirements.txt`中移除`plugins.zip`或嗎？

目前，一旦添加了 plugins.zip 或 requirements.txt，就無法從環境中刪除它們，但我們正在處理這個問題。在過渡期間，因應措施是分別指向空白文字或 zip 檔案。如需進一步了解，請參閱 刪除 Amazon S3 上的文件。

為什麼我在 Apache 氣流 v2.0.2 管理插件菜單中看不到我的插件？

基於安全理由，Amazon MWAA 上的 Apache 氣流網頁伺服器的網路輸出量有限，而且不會直接在 2.0.2 版環境的 Apache 氣流網頁伺服器上安裝外掛程式或 Python 相依性。顯示的外掛程式可讓 Amazon MWAA 在 AWS Identity and Access Management (IAM) 中驗證您的 Apache 氣流使用者。

若要能夠直接在網頁伺服器上安裝外掛程式和 Python 相依性，我們建議您使用 Apache 氣流 v2.2 及更新版本建立新的環境。Amazon MWAA 直接在阿帕奇氣流 v2.2 及更高版本的網頁伺服器上安裝 Python 相依性和自訂外掛程式。

我可以使用 AWS Database Migration Service (DMS) 操作員嗎？

Amazon MWAA 支援 D MS 運營商。不過，這個運算子無法用來在與 Amazon MWAA 環境關聯的 Amazon Aurora PostgreSQL 中繼資料資料庫上執行動作。

Amazon Managed Workflows Managed Workflows Managed Workflows

本主題說明在適用於 Apache 氣流的 Amazon 受管工作流程上使用 Apache Airflow 時可能遇到的常見問題和錯誤，以及解決這些錯誤的建議步驟。

內容

疑難排解：在 Apache 氣流 v2 中的 DAG、操作員、連線和其他問題

本頁的主題說明 Apache Airflow v2 Python 相依性、自訂外掛程式、DAG、操作員、連線、任務和 Web 伺服器問題的解決方案，您可能會遇到的 Apache 氣流環境的 Amazon 受管工作流程。

內容

連線

下列主題說明您在使用 Apache Airflow 連線或使用其他AWS資料庫時可能收到的錯誤。

我無法連線至 Secrets Manager 式

建議下列步驟：

1. 瞭解如何在中為 Apache 氣流連線和變數建立密鑰the section called “設定 Secrets Manager”。
2. 瞭解如何在中針對 Apache 氣流變數 (`test-variable`) 使用密鑰使用密鑰AWS Secrets Manager對於一個阿帕奇氣流變量。
3. 在中了解如何使用密鑰進行 Apache 氣流連接 (`myconn`)使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接。

如何在我的執行角色策略中配置`secretsmanager:ResourceTag/<tag-key>` Secret 管理器條件或資源限制？

> **Note**
>
> 適用於 Apache 氣流 2.0 版及更早版本。

目前，您無法使用環境執行角色中的條件金鑰或其他資源限制來限制 Secrets Manager 密碼的存取，這是因為 Apache Airflow 中存在已知問題所致。

我無法連線至雪花

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將下列項目新增至您環境的 requirements.txt。

```
apache-airflow-providers-snowflake==1.3.0
```

3. 將下列匯入新增至您的 DAG：

```
from airflow.providers.snowflake.operators.snowflake import SnowflakeOperator
```

確保 Apache Airflow 連線物件包含下列索引鍵/值組：

1. 連接埠 ID：雪花 _ 連接
2. 連接器類型：雪花
3. 主持人：<my account>. <my region if not us-west-2>. 雪花計算 .com
4. 綱要：<my schema>
5. 登入:<my user name>
6. 密碼:
7. 連接埠：<port, if any>
8. 額外：

```
{
    "account": "<my account>",
    "warehouse": "<my warehouse>",
    "database": "<my database>",
    "region": "<my region if not using us-west-2 otherwise omit this line>"
}
```

例如：

```
>>> import json
```

```
>>> from airflow.models.connection import Connection
>>> myconn = Connection(
...     conn_id='snowflake_conn',
...     conn_type='Snowflake',
...     host='YOUR_ACCOUNT.YOUR_REGION.snowflakecomputing.com',
...     schema='YOUR_SCHEMA'
...     login='YOUR_USERNAME',
...     password='YOUR_PASSWORD',
...     port='YOUR_PORT'
...     extra=json.dumps(dict(account='YOUR_ACCOUNT', warehouse='YOUR_WAREHOUSE',
 database='YOUR_DB_OPTION', region='YOUR_REGION')),
... )
```

我在氣流使用者介面中看不到我的連線

阿帕奇氣流提供了 Apache 氣流用戶界面中的連接模板。它使用它來生成連接 URI 字符串，而不管連接類型如何。如果 Apache Airflow UI 中沒有連線範本，則可以使用替代連線範本來產生連線 URI 字串，例如使用 HTTP 連線範本。

建議下列步驟：

1. 在 Apache 氣流使用者介面中檢視亞馬遜 MWAA 提供的連線類型，位於Amazon MWAA 環境上安裝的 Apache 氣流供應商套件。
2. 檢視在 CLI 中建立 Apache 氣流連線的命令阿帕奇氣流 CLI 命令參考。
3. 了解如何在 Apache 氣流使用者介面中互換使用連線範本，以便在 Amazon MWAA 上的 Apache 氣流使用者介面中無法使用的連線類型連線類型概觀。

Web 伺服器

下列主題說明您在 Amazon MWAA 上針對 Apache 氣流網頁伺服器可能收到的錯誤。

我看到存取網頁伺服器時發生 5xx 錯誤

建議下列步驟：

1. 檢查 Apache Airflow 組態選項。確認您指定為 Apache 氣流組態選項的索引鍵值配對 (例如AWS Secrets Manager) 已正確設定。如需進一步了解，請參閱 the section called “我無法連線至 Secrets Manager 式”。
2. 檢查`requirements.txt`。確認您的氣流「額外」套件和其他程式庫與`requirements.txt`您的 Apache 氣流版本相容。

3. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

我看到「調度程序似乎沒有運行」錯誤

如果排程器似乎沒有在執行中，或數小時前收到最後一個「心跳」，您的 DAG 可能不會出現在 Apache Airflow 中，而且不會排定新工作。

建議下列步驟：

1. 確認您的 VPC 安全群組允許對連接埠進行輸入存取5432。需要此連接埠才能連接到您環境的 Amazon Aurora PostgreSQL 中繼資料資料庫。添加此規則後，請給 Amazon MWAA 幾分鐘，錯誤應該消失。如需進一步了解，請參閱 the section called “您的 VPC 安全政策”。

   > **Note**
   >
   > - Aurora PostgreSQL 中繼資料庫是亞馬遜 MWAA 服務架構的一部分，在您的AWS 帳戶.
   > - 資料庫相關錯誤通常是排程器失敗的徵兆，而不是根本原因。

2. 如果排程器未執行，可能是由於許多因素，例如相依性安裝失敗或排程器過載。在記錄檔中檢視對應的CloudWatch記錄群組，確認您的 DAG、外掛程式和需求是否正常運作。如需進一步了解，請參閱 監控和指標。

任務

下列主題說明您在環境中針對 Apache Airflow 工作可能收到的錯誤。

我看到我的任務卡住或沒有完成

如果您的 Apache 氣流任務「卡住」或未完成，我們建議您執行以下步驟：

1. 可能有大量的定義 DAG。減少 DAG 的數目，並執行環境的更新 (例如變更記錄層級) 以強制重設。

   a. 氣流會剖析 DAG 是否已啟用。如果您使用的環境容量超過 50%，您可能會開始壓倒 Apache 氣流排程器。這會導致CloudWatch量度中的「總剖析時間」較長，或是CloudWatch記錄檔中的 DAG 處理時間過長。還有其他方法可將 Apache Airflow 組態設定最佳化，這些組態不在本指南的討論範圍內。

b. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。

2. 佇列中可能有大量工作。這通常會顯示為處於「無」狀態的大量工作 (而且正在增加)，或在中排入佇列的工作和/或擱置中的工作中顯示為大量工作CloudWatch。這種情況可能是由於下列原因而發生：

   a. 如果要執行的工作數量超過環境所能執行的容量，和/或在自動調度資源有時間偵測工作並部署其他 Worker 之前排入佇列的大量工作。

   b. 如果要執行的工作數量超過具有執行容量的環境，建議您減少 DAG 同時執行的工作數量，和/或增加 Apache Airflow Worker 的最小值。

   c. 如果在自動調度資源有足夠時間偵測和部署其他工作者之前，有大量工作已排入佇列中，我們建議您進行驚人的工作部署和/或增加 Apache Airflow Worker 的最小值。

   d. 您可以使用AWS Command Line Interface (AWS CLI) 中的更新環境命令來變更在您環境中執行的 Worker 數目下限或上限。

   ```
   aws mwaa update-environment --name MyEnvironmentName --min-workers 2 --max-
   workers 10
   ```

   e. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。

3. 執行中可能會刪除某些工作，這些工作會顯示為工作記錄，而在 Apache Airflow 中沒有進一步指示的情況下停止。這種情況可能是由於下列原因而發生：

   a. 如果有一個短暫的時刻，其中 1）當前任務超過當前的環境容量，然後是 2）幾分鐘沒有任務執行或排隊，那麼 3）新的任務被排隊。

   b. Amazon MWAA 透過新增其他工作程式，自動調度資源回應第一個案例。在第二個案例中，它會移除其他 Worker。某些正在排入佇列的工作可能會導致 Worker 在移除程序中，而且會在刪除容器時結束。

   c. 我們建議您增加環境中的最少員工數量。另一個選項是調整 DAG 和工作的時間，以確保這些案例不會發生。

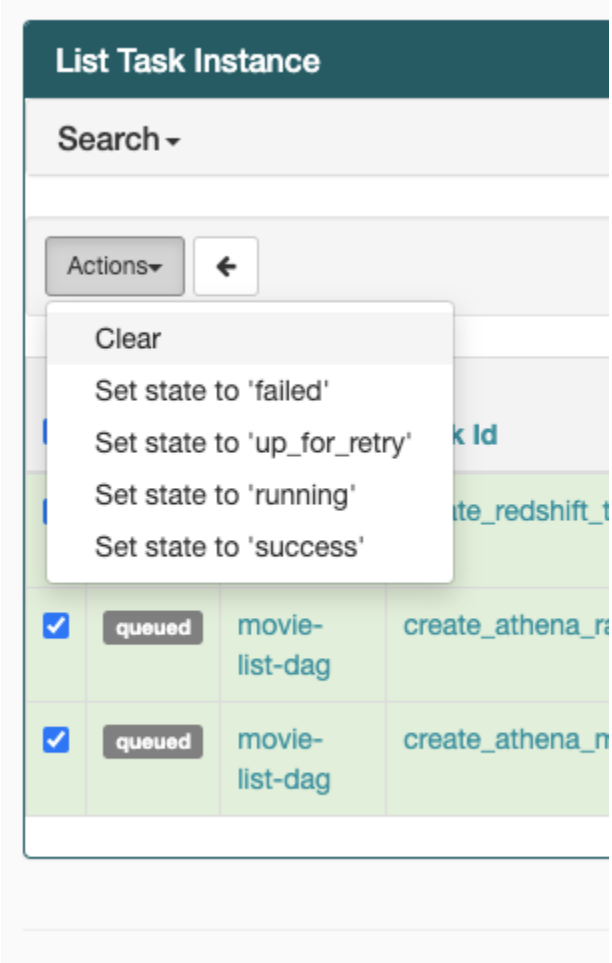
   d. 您還可以將最小工作人員設置為等於環境中最大工作人員，以有效地禁用自動調度資源。使用AWS Command Line Interface (AWS CLI) 中的更新環境命令，藉由將 Worker 的最小和最大數目設定為相同，以停用自動調度資源。

   ```
   aws mwaa update-environment --name MyEnvironmentName --min-workers 5 --max-
   workers 5
   ```

e. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。

4. 如果您的任務處於「正在運行」狀態，您也可以清除任務或將其標記為成功或失敗。這可讓您的環境自動調度資源元件縮減環境中執行的 Worker 數目。以下圖片顯示了擱淺工作的範例。

- 選擇擱淺工作的圓形，然後選取 [清除] (如圖所示)。這可讓 Amazon MWAA 縮減工作者的規模；否則，Amazon MWAA 無法判斷哪些 DAG 已啟用或停用，如果仍有佇列的任務，也無法縮小規模。



5. 若要深入瞭解 Apache 氣流工作生命週期，請參閱 Apache 氣流參考指南中的概念。

CLI

下列主題說明您在中執行 Airflow CLI 命令時可能收到的錯誤AWS Command Line Interface。

在 CLI 中觸發 DAG 時，我看到「503」錯誤

氣流 CLI 在 Apache 氣流網頁伺服器上執行，該伺服器的並行性有限。通常最多可同時執行 4 個 CLI 命令。

為什麼**dags backfill**阿帕奇氣流 CLI 命令失敗？ 有沒有解決方法？

> **Note**
>
> 以下內容僅適用於 Apache Airflow v2.0.2 環境。

與其他 Apache 氣流 CLI`backfill` 命令一樣，命令會在處理任何 DAG 之前在本機剖析所有 DAG，而不論 CLI 作業適用於哪個 DAG。在使用 Apache Airflow v2.0.2 的 Amazon MWAA 環境中，由於 CLI 命令執行時尚未在網頁伺服器上安裝外掛程式和需求，因此剖析作業會失敗，且不會叫用`backfill`作業。如果您的環境中沒有任何要求或插件，則`backfill`操作將成功。

為了能夠運行`backfill` CLI 命令，我們建議在 bash 運算符中調用它。在 bash 運算符中，`backfill`從 Worker 啟動，允許 DAG 成功解析，因為所有必要的需求和 PLGUIN 都可用並已安裝。以下範例顯示如何使用建立 DAG，`BashOperator`以執行`backfill`。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago

with DAG(dag_id="backfill_dag", schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    cli_command = BashOperator(
        task_id="bash_command",
        bash_command="airflow dags backfill my_dag_id"
    )
```

電信業者

下列主題說明使用運算子時可能收到的錯誤。

我使用 S3 轉換運算符收到PermissionError: [Errno 13] Permission denied錯誤

如果您嘗試使用 S3Transform 運算子執行 shell 指令碼，並且收到PermissionError: [Errno 13] Permission denied錯誤訊息，我們建議您執行下列步驟。下列步驟假設您有現有的 plugins.zip 檔案。如果您要建立新的 plugins.zip，請參閱安裝自定義插件。

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 創建您的「轉換」腳本。

```
#!/bin/bash
cp $1 $2
```

3. (可選) macOS 和 Linux 使用者可能需要執行下列命令，以確保指令碼可執行。

```
chmod 777 transform_test.sh
```

4. 將指令碼新增到您的 plugins.zip 中。

```
zip plugins.zip transform_test.sh
```

5. 請按照將 plugins.zip 上傳到 Amazon S3 中的步驟進行操作。
6. 請依照在亞馬遜 MWAA 主控台上指定 plugins.zip 版本中的步驟進行。
7. 建立下列 DAG。

```
from airflow import DAG
from airflow.providers.amazon.aws.operators.s3_file_transform import
 S3FileTransformOperator
from airflow.utils.dates import days_ago
import os

DAG_ID = os.path.basename(__file__).replace(".py", "")

with DAG (dag_id=DAG_ID, schedule_interval=None, catchup=False,
 start_date=days_ago(1)) as dag:
    file_transform = S3FileTransformOperator(
        task_id='file_transform',
        transform_script='/usr/local/airflow/plugins/transform_test.sh',
        source_s3_key='s3://YOUR_S3_BUCKET/files/input.txt',
        dest_s3_key='s3://YOUR_S3_BUCKET/files/output.txt'
```

```
)
```

8. 請依照將 DAG 程式碼上傳至 Amazon S3 中的步驟進行。

疑難排解：DAG、操作員、連線和 Apache 氣流 v1 中的其他問題

本頁的主題包含 Apache 氣流 v1.10.12 Python 相依性、自訂外掛程式、DAG、操作員、連線、任務和 Web 伺服器問題的解決方案，您可能會在 Apache 氣流環境的 Amazon 受管工作流程中遇到的問題。

內容

- 我在 CLI 中觸發 DAG 時看到「503」錯誤

更新 requirements.txt 文件

下列主題說明您在更新`requirements.txt`.

新增**apache-airflow-providers-amazon**會導致我的環境失敗

`apache-airflow-providers-`**`xyz`**僅與阿帕奇氣流 v2 兼容。 `apache-airflow-backport-providers-`**`xyz`**與阿帕奇氣流 1.10.12 兼容。

破碎的天

下列主題說明您在執行 DAG 時可能收到的錯誤。

我在使用 Amazon DynamoDB 運算子時收到「損壞的 DAG」錯誤訊息

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將以下軟件包添加到您的`requirements.txt`.

   ```
   boto
   ```

3. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

我收到 '損壞的 DAG：沒有名為 psycopg2 的模塊' 錯誤

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將以下內容添加到您`requirements.txt`的 Apache 氣流版本中。例如：

   ```
   apache-airflow[postgres]==1.10.12
   ```

3. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

我在使用 Slack 運算子時收到「破碎的 DAG」錯誤訊息

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將下列套件新增至您的，`requirements.txt`並指定您的 Apache 氣流版本。例如：

```
apache-airflow[slack]==1.10.12
```

3. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

我收到了安裝谷歌/GCP/ 的各種錯誤BigQuery

亞馬遜 MWAA 使用亞馬遜 Linux，它需要特定版本的 Cython 和加密庫。建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將以下軟件包添加到您的`requirements.txt`.

```
grpcio==1.27.2
cython==0.29.21
pandas-gbq==0.13.3
cryptography==3.3.2
apache-airflow-backport-providers-amazon[google]
```

3. 如果您不使用反向移植提供程序，則可以使用：

```
grpcio==1.27.2
cython==0.29.21
pandas-gbq==0.13.3
cryptography==3.3.2
apache-airflow[gcp]==1.10.12
```

4. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

我收到「損壞的 DAG：沒有名為 Cython on 的模塊」錯誤

亞馬遜 MWAA 使用亞馬遜 Linux，這需要特定版本的 Cython。建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將以下軟件包添加到您的`requirements.txt`.

```
cython==0.29.21
```

3. Cython 庫有各種所需的 pip 依賴版本。例如，使用`awswrangler==2.4.0`需要`pyarrow<3.1.0,>=2.0.0`，因此 pip3 嘗試安裝`pyarrow==3.0.0`，導致 DAG 損壞錯誤。我們建議明確指定最舊的可接受版本。例如，如果您`pyarrow==2.0.0`在之前指定了最小值，`awswrangler==2.4.0`則錯誤消失，並且`requirements.txt`安裝正確。最終要求應如下所示：

```
cython==0.29.21
pyarrow==2.0.0
awswrangler==2.4.0
```

4. 探索在文件中指定 Python 依賴關係的方`requirements.txt`法，請參閱在 requirements.txt Python 管理依賴項。

電信業者

下列主題說明使用運算子時可能收到的錯誤。

我使用BigQuery運算子收到錯誤訊息

亞馬遜 MWAA 不支援具有 UI 擴充功能的運算子。建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 因應措施是在匯入問題運算子`<operator name>.operator_extra_links = None`之後，在 DAG 中新增要設定的行來覆寫延伸模組。例如：

```
from airflow.contrib.operators.bigquery_operator import BigQueryOperator
BigQueryOperator.operator_extra_links = None
```

3. 您可以通過將上述內容添加到插件中來為所有 DAG 使用此方法。如需範例，請參閱 the section called “自定義插件來修補PythonVirtualenvOperator ”。

連線

下列主題說明您在使用 Apache Airflow 連線或使用其他AWS資料庫時可能收到的錯誤。

我無法連線至雪花

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 將下列項目新增至您環境的 requirements.txt。

```
asn1crypto == 0.24.0
snowflake-connector-python == 1.7.2
```

3. 將下列匯入新增至您的 DAG：

```
from airflow.contrib.hooks.snowflake_hook import SnowflakeHook
from airflow.contrib.operators.snowflake_operator import SnowflakeOperator
```

確定 Apache Airflow 連線物件包含下列索引鍵/值組：

1. 康涅狄格 ID：雪花 _ 連接器
2. 連接器類型：雪花
3. 主持人：<my account>. <my region if not us-west-2>. 雪花計算 .com
4. 綱要：<my schema>
5. 登入:<my user name>
6. 密碼:
7. 連接埠：<port, if any>
8. 額外：

```
{
    "account": "<my account>",
    "warehouse": "<my warehouse>",
    "database": "<my database>",
    "region": "<my region if not using us-west-2 otherwise omit this line>"
}
```

例如：

```
>>> import json
```

```
>>> from airflow.models.connection import Connection
>>> myconn = Connection(
...     conn_id='snowflake_conn',
...     conn_type='Snowflake',
...     host='YOUR_ACCOUNT.YOUR_REGION.snowflakecomputing.com',
...     schema='YOUR_SCHEMA'
...     login='YOUR_USERNAME',
...     password='YOUR_PASSWORD',
...     port='YOUR_PORT'
...     extra=json.dumps(dict(account='YOUR_ACCOUNT', warehouse='YOUR_WAREHOUSE',
 database='YOUR_DB_OPTION', region='YOUR_REGION')),
... )
```

我無法連線至 Secrets Manager 式

建議下列步驟：

1. 瞭解如何在中為 Apache 氣流連線和變數建立密鑰the section called “設定 Secrets Manager”。
2. 瞭解如何在中針對 Apache 氣流變數 (`test-variable`) 使用密鑰使用密鑰AWS Secrets Manager對於一個阿帕奇氣流變量。
3. 在中了解如何使用密鑰進行 Apache 氣流連接 (`myconn`)使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接。

我無法在 '<DB-identifier-name>.cluster-id 上連接到我的 MySQL 服務器。<region>.rds. 亞馬遜. COM」

Amazon MWAA 的安全群組和 RDS 安全群組需要輸入規則，才能允許彼此往來的流量。建議下列步驟：

1. 修改 RDS 安全群組，以允許來自 Amazon MWAA 虛擬私人雲端安全群組的所有流量。
2. 修改 Amazon MWAA 的虛擬私人雲端安全群組，以允許來自 RDS 安全群組的所有流量。
3. 再次重新執行您的工作，並檢查記錄中CloudWatch的 Apache 氣流記錄，以確認 SQL 查詢是否成功。

Web 伺服器

下列主題說明您在 Amazon MWAA 上針對 Apache 氣流網頁伺服器可能收到的錯誤。

我正在使用BigQueryOperator它導致我的 Web 服務器崩潰

建議下列步驟：

1. Apache 氣流操作員 (例如BigQueryOperator和包QuboleOperator含)operator_extra_links 可能會導致 Apache 氣流網頁伺服器當機。這些操作員嘗試將代碼加載到您的 Web 服務器，出於安全原因，這是不允許的。建議您在匯入陳述式之後新增下列程式碼，以修補 DAG 中的運算子：

   ```
   BigQueryOperator.operator_extra_links = None
   ```

2. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。

我看到存取網頁伺服器時發生 5xx 錯誤

建議下列步驟：

1. 檢查 Apache Airflow 組態選項。確認您指定為 Apache 氣流組態選項的索引鍵值配對 (例如AWS Secrets Manager) 已正確設定。如需進一步了解，請參閱 the section called “我無法連線至 Secrets Manager 式”。
2. 檢查requirements.txt。確認您的氣流「額外」套件和其他程式庫與requirements.txt您的 Apache 氣流版本相容。
3. 探索在文件中指定 Python 依賴關係的方requirements.txt法，請參閱在 requirements.txt Python 管理依賴項。

我看到「調度程序似乎沒有運行」錯誤

如果排程器似乎沒有在執行中，或數小時前收到最後一個「心跳」，您的 DAG 可能不會出現在 Apache Airflow 中，而且不會排定新工作。

建議下列步驟：

1. 確認您的 VPC 安全群組允許對連接埠進行輸入存取5432。需要此連接埠才能連接到您環境的 Amazon Aurora PostgreSQL 中繼資料資料庫。添加此規則後，請給 Amazon MWAA 幾分鐘，錯誤應該消失。如需進一步了解，請參閱 the section called “您的 VPC 安全政策”。

> **Note**
>
> - Aurora PostgreSQL 中繼資料庫是亞馬遜 MWAA 服務架構的一部分，在您的AWS 帳戶.
> - 資料庫相關錯誤通常是排程器失敗的徵兆，而不是根本原因。

2. 如果排程器未執行，可能是由於許多因素，例如相依性安裝失敗或排程器過載。在記錄檔中檢視對應的CloudWatch記錄群組，確認您的 DAG、外掛程式和需求是否正常運作。如需進一步了解，請參閱 監控和指標。

任務

下列主題說明您在環境中針對 Apache Airflow 工作可能收到的錯誤。

我看到我的任務卡住或未完成

如果您的 Apache 氣流任務「卡住」或未完成，我們建議您執行以下步驟：

1. 可能有大量的定義 DAG。減少 DAG 的數目，並執行環境的更新 (例如變更記錄層級) 以強制重設。
   a. 氣流會剖析 DAG 是否已啟用。如果您使用的環境容量超過 50%，您可能會開始壓倒 Apache 氣流排程器。這會導致CloudWatch量度中的「總剖析時間」較長，或是CloudWatch記錄檔中的 DAG 處理時間過長。還有其他方法可將 Apache Airflow 組態最佳化，不在本指南的討論範圍內。
   b. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。
2. 佇列中可能有大量工作。這通常會顯示為處於「無」狀態的大量工作 (而且正在增加)，或在中排入佇列的工作和/或擱置中的工作中顯示為大量工作CloudWatch。這種情況可能是由於下列原因而發生：
   a. 如果要執行的工作數量超過環境所能執行的容量，和/或在自動調度資源有時間偵測工作並部署其他 Worker 之前排入佇列的大量工作。
   b. 如果要執行的工作數量超過具有執行容量的環境，建議您減少 DAG 同時執行的工作數量，和/或增加 Apache Airflow Worker 的最小值。
   c. 如果在自動調度資源有時間偵測和部署其他工作者之前，有大量工作已排入佇列，建議您進行驚人的工作部署和/或增加 Apache Airflow Worker 的最小值。

d. 您可以使用AWS Command Line Interface (AWS CLI) 中的更新環境命令來變更在您環境中執行的 Worker 數目下限或上限。

```
aws mwaa update-environment --name MyEnvironmentName --min-workers 2 --max-
workers 10
```

e. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。

3. 執行中可能會刪除某些工作，這些工作會顯示為工作記錄，而在 Apache Airflow 中沒有進一步指示的情況下停止。這種情況可能是由於下列原因而發生：

a. 如果有一個短暫的時刻，其中 1）當前任務超過當前的環境容量，然後是 2）幾分鐘沒有任務執行或排隊，那麼 3）新的任務被排隊。

b. Amazon MWAA 透過新增其他工作程式，自動調度資源回應第一個案例。在第二個案例中，它會移除其他 Worker。某些正在排入佇列的工作可能會導致 Worker 在移除程序中，而且會在刪除容器時結束。

c. 我們建議您增加環境中的最少員工數量。另一個選項是調整 DAG 和工作的時間，以確保這些案例不會發生。

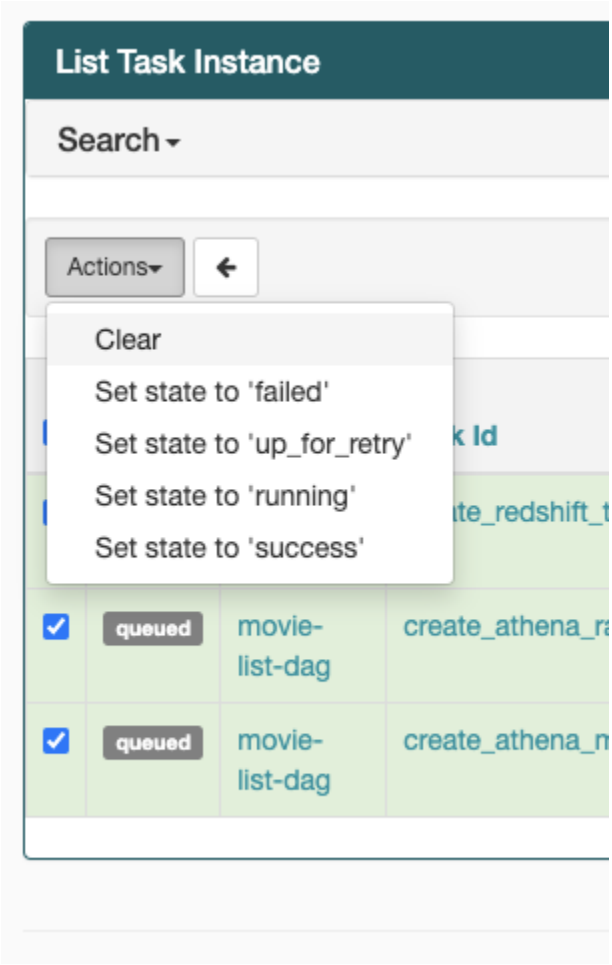
d. 您還可以將最小工作人員設置為等於環境中最大工作人員，以有效地禁用自動調度資源。使用AWS Command Line Interface (AWS CLI) 中的更新環境命令，藉由將 Worker 的最小和最大數目設定為相同，以停用自動調度資源。

```
aws mwaa update-environment --name MyEnvironmentName --min-workers 5 --max-
workers 5
```

e. 若要深入瞭解我們建議調整環境效能的最佳做法，請參閱the section called “Apache 氣流的效能微調”。

4. 如果您的任務處於「正在運行」狀態，您也可以清除任務或將其標記為成功或失敗。這可讓您的環境自動調度資源元件縮減環境中執行的 Worker 數目。下圖顯示了擱淺工作的範例。

- 選擇擱淺工作的圓形，然後選取 [清除] (如圖所示)。這可讓 Amazon MWAA 縮減工作者的規模；否則，Amazon MWAA 無法判斷哪些 DAG 已啟用或停用，如果仍有佇列的任務，也無法縮小規模。



5. 若要深入瞭解 Apache 氣流工作生命週期，請參閱 Apache 氣流參考指南中的概念。

CLI

下列主題說明您在中執行 Airflow CLI 命令時可能收到的錯誤AWS Command Line Interface。

我在 CLI 中觸發 DAG 時看到「503」錯誤

氣流 CLI 在 Apache 氣流網頁伺服器上執行，該伺服器的並行性有限。通常最多可同時執行 4 個 CLI 命令。

故障排除：建立和更新 Amazon MWAA 環境

本頁的主題包含建立和更新 Apache Airflow 環境的 Amazon 受管工作流程時可能遇到的錯誤，以及如何解決這些錯誤。

內容

更新 **requirements.txt**

下列主題說明您在更新`requirements.txt`.

我指定了我的一個新版本**requirements.txt**，更新我的環境需要超過 20 分鐘

如果您的環境需要二十分鐘以上才能安裝新版本的`requirements.txt`檔案，則環境更新會失敗，而 Amazon MWAA 正在復原至容器映像的最新穩定版本。

1. 檢查套件版本。我們建議您始終指定一個特定的版本（==>=）或最大版本（ ）為您的`requirements.txt`.
2. 檢查阿帕奇氣流日誌。如果您已啟用 Apache Airflow 記錄檔，請在CloudWatch主控台的 [記錄群組] 頁面上確認您的記錄群組已成功建立。如果您看到空白日誌，最常見的原因是在寫入日誌的

Amazon S3CloudWatch 或 Amazon S3 的執行角色中缺少許可。如需進一步了解，請參閱 執行角色。

3. 檢查 Apache Airflow 組態選項。如果您使用的是 Secrets Manager，請確認您指定為 Apache 氣流組態選項的索引鍵值配對是否已正確設定。如需進一步了解，請參閱 the section called “設定 Secrets Manager”。
4. 檢查 VPC 網路組態。如需進一步了解，請參閱 the section called “環境卡住”。
5. 檢查執行角色許可。執行角色是具有許可政策的AWS Identity and Access Management (IAM) 角色，可授予 Amazon MWAA 權限，以代表您叫用其他AWS服務 (例如 Amazon S3CloudWatch、Amazon SQS、Amazon ECR) 的資源。您的客戶管理金鑰或AWS擁有的金鑰也需要被允許存取。如需進一步了解，請參閱 執行角色。
6. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。

外掛程式

下列主題說明您在設定或更新 Apache Airflow 外掛程式時可能遇到的問題。

亞馬遜 MWAA 是否支援實作自訂使用者介面？

從 Apache 氣流 v2.2.2 開始，亞馬遜 MWAA 支持在 Apache 氣流網絡服務器上安裝插件，並實現自定義用戶界面。如果您的亞馬遜 MWAA 環境正在執行 Apache 氣流 v2.0.2 或更早版本，您將無法實作自訂使用者介面。

如需有關版本管理和升級現有環境的詳細資訊，請參閱版本。

我能夠通過插件在 Amazon MWAA 本地運行器上實現自定義 UI 更改，但是當我嘗試在 Amazon MWAA 上執行相同的操作時，我看不到我的更改也沒有任何錯誤。為什麼會發生這種情況？

亞馬遜 MWAA 本地運行器將所有 Apache 氣流組件捆綁在一個映像中，允許您應用自定義 UI 插件更改。

建立儲存貯

下列主題說明您在建立 Amazon S3 儲存貯體時可能收到的錯誤。

我無法選擇 S3 封鎖公有存取設定

Amazon MWAA 環境的執行角色需要 Amazon S3 儲存貯體上GetBucketPublicAccessBlock動作的許可，才能驗證儲存貯體封鎖的公共存取。建議下列步驟：

1. 請依照下列步驟將 JSON 原則附加至您的執行角色。
2. 附加下列 JSON 政策：

```
{
   "Effect":"Allow",
   "Action":[
      "s3:GetObject*",
      "s3:GetBucket*",
      "s3:List*"
   ],
   "Resource":[
      "arn:aws:s3:::YOUR_S3_BUCKET_NAME",
      "arn:aws:s3:::YOUR_S3_BUCKET_NAME/*"
   ]
}
```

   以*## Amazon S3 ######*取代您的範例預留位置，例如 *my-mwaa-unique-s3 #######*。
3. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。

建立 環境

下列主題說明您在建立環境時可能會收到的錯誤。

我試圖創建一個環境，它停留在「創建」狀態

建議下列步驟：

1. 使用公共路由檢查 VPC 網路。如果您使用具有網際網路存取權的 Amazon VPC，請確認以下事項：
   - 您的 Amazon VPC 設定為允許 Amazon MWAA 環境使用的不同AWS資源之間的網路流量(如中所定義)the section called “關於網路”。例如，您的 VPC 安全性群組必須允許自我參照規則中的所有流量，或選擇性地指定 HTTPS 連接埠範圍 443 的連接埠範圍和 TCP 連接埠範圍 5432。

2. 使用私有路由檢查 VPC 私人雲端網路。如果您在沒有網際網路存取的情況下使用 Amazon VPC，請確認以下事項：

    - 您的 Amazon VPC 設定為允許 Amazon MWAA 環境的不同AWS資源之間的網路流量 (如中所定義)the section called “關於網路”。例如，您的兩個私有子網路不得有 NAT 閘道 (或 NAT 執行個體) 的路由表，也不能有網際網路閘道。

3. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。

我試圖創建一個環境，但它顯示狀態為「創建失敗」

建議下列步驟：

1. 檢查 VPC 網路組態。如需進一步了解，請參閱 the section called “環境卡住”。
2. 檢查使用者權限。Amazon MWAA 在建立環境之前，對使用者的登入資料執行乾式執行。您的AWS帳戶可能沒有AWS Identity and Access Management (IAM) 中的許可，無法為環境建立部分資源。例如，如果您選擇私人網路 Apache Airflow 存取模式，您的AWS帳戶必須已被管理員授與您環境的 AmazonmWAAFullConsoleAccess 存取控制原則的存取權限，這可讓您的帳戶建立 VPC 端點。
3. 檢查執行角色許可。執行角色是具有許可政策的AWS Identity and Access Management (IAM) 角色，可授予 Amazon MWAA 權限，以代表您叫用其他AWS服務 (例如 Amazon S3CloudWatch、Amazon SQS、Amazon ECR) 的資源。您的客戶管理金鑰或AWS擁有的金鑰也需要被允許存取。如需進一步了解，請參閱 執行角色。
4. 檢查阿帕奇氣流日誌。如果您已啟用 Apache Airflow 記錄檔，請在CloudWatch主控台的 [記錄群組] 頁面上確認您的記錄群組已成功建立。如果您看到空白日誌，最常見的原因是在寫入日誌的 Amazon S3CloudWatch 或 Amazon S3 的執行角色中缺少許可。如需進一步了解，請參閱 執行角色。
5. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。
6. 如果您在沒有網際網路存取的情況下使用 Amazon VPC，請確保您已建立 Amazon S3 閘道端點，並將所需的最低權限授予 Amazon ECR 以存取 Amazon S3。若要進一步了解建立 Amazon S3 閘道端點，請參閱以下內容：

    - 建立沒有網際網路存取權的 Amazon VPC 網路
    - 在 Amazon E lastic Container Registry 使用者指南中建立 Amazon S3 閘道端點

我嘗試選擇一個 VPC 並收到「網絡故障」錯誤

建議下列步驟：

- 如果在建立環境時嘗試選取 Amazon VPC 時看到「網路故障」錯誤，請關閉任何正在執行的瀏覽器內 Proxy，然後再試一次。

我嘗試創建一個環境並收到服務，分區或資源「必須傳遞」錯誤

建議下列步驟：

- 您可能會收到此錯誤，因為您為 Amazon S3 儲存貯體指定的 URI 在 URI 末尾包含「/」。我們建議移除路徑中的「/」。該值應採用以下格式：

```
s3://your-bucket-name
```

我嘗試創建一個環境，它將狀態顯示為「可用」，但是當我嘗試訪問 Airflow UI 時，會顯示「來自服務器的空回复」或「502 錯誤網關」錯誤

建議下列步驟：

1. 檢查 VPC 安全群組組態。如需進一步了解，請參閱 the section called “環境卡住”。
2. 確認您在列出的任何 Apache 氣流套件都`requirements.txt`對應於您在亞馬遜 MWAA 上執行的 Apache 氣流版本。如需進一步了解，請參閱 安裝 Python 的依賴。
3. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。

我試圖創建一個環境，我的用戶名是一堆隨機字符名

- Apache Airflow 最多有 64 個字符的使用者名稱。如果您的AWS Identity and Access Management (IAM) 角色超過此長度，則會使用雜湊演算法來減少它，同時保持唯一性。

更新環境

下列主題說明您在更新環境時可能會收到的錯誤。

我嘗試更改環境類，但更新失敗

如果您將環境更新為不同的環境類別 (例如將環境變更為mw1.small)，且更新環境的要求失敗，則環境狀態會進入UPDATE_FAILED狀態，而且環境會復原至環境，並根據環境的先前穩定版本計費。mw1.medium

建議下列步驟：

1. 使用 aws-mwaa-local-runneron 在本機測試您的 DAG、自訂外掛程式和 Python 相依性GitHub。
2. 若要執行疑難排解指令碼來檢查 Amazon VPC 網路設定和設定的 Amazon MWAA 環境，請參閱上的 SupAWS port 工具中的驗證環境指令碼GitHub。

存取環境

下列主題說明存取環境時可能會收到的錯誤。

我無法訪問 Apache Airflow UI

建議下列步驟：

1. 檢查使用者權限。您可能沒有被授與權限原則的存取權，該原則可讓您檢視 Apache 氣流使用者介面。如需進一步了解，請參閱 the section called “存取 Amazon MWAA 環境”。
2. 檢查網路存取。這可能是因為您選擇了私人網路存取模式。如果您的 Apache 氣流使用者介面的 URL 採用下列格式387fbcn-8dh4-9hfj-0dnd-834jhdfb-vpce.c10.us-west-2.airflow.amazonaws.com，表示您正在使用 Apache 氣流網頁伺服器的私人路由。您可以將 Apache 氣流存取模式更新為公用網路存取模式，或建立機制來存取 Apache 氣流網頁伺服器的 VPC 人雲端端點。如需進一步了解，請參閱the section called “管理 VPC 端點的存取”。

疑難排解： CloudWatch 記錄檔和 CloudTrail 錯誤

本頁的主題包含 Amazon CloudWatch 日誌的解決方案，以及您在 Apache 氣流環境的 Amazon 受管工作流程中可能遇到的AWS CloudTrail錯誤。

內容

- 我看到一個 ResourceAlreadyExistsException " 錯誤 CloudTrail
- 我看到「無效請求」錯誤 CloudTrail
- 我在 Apache 氣流日誌中看到「找不到 64 位元的 Oracle 用戶端程式庫：「libclntsh.so：無法開啟共享物件檔案：沒有這樣的檔案或目錄」
- 我在我的調度程序日誌中看到 psycopg2「服務器意外關閉了連接」
- 我看到 '執行程序報告任務實例 %s 已完成（ %s ），儘管任務在我的 DAG 處理日誌中顯示其 %s
- 我看到「無法從 log_group 讀取遠程日誌：氣流-* {* 環境名稱}-任務日誌流：* {* DAG_ID} /* {* TASK_ID} /* {* 時間} /* {* n}. 日誌。」 在我的任務日誌中

日誌

下列主題說明您在檢視 Apache 氣流記錄檔時可能收到的錯誤。

我看不到我的任務日誌，或者我收到「從 Cloudwatch 日誌組讀取遠端日誌」錯誤

亞馬遜 MWAA 已將 Apache 氣流設定為直接從亞馬遜 CloudWatch 日誌讀取和寫入日誌。如果 Worker 無法啟動工作，或無法寫入任何記錄，您將會看到以下錯誤：

```
*** Reading remote log from Cloudwatch log_group: airflow-environmentName-Task
 log_stream: DAG_ID/TASK_ID/timestamp/n.log.Could not read remote logs from log_group:
 airflow-environmentName-Task log_stream: DAG_ID/TASK_ID/time/n.log.
```

- 建議下列步驟：

  a. 確認您已在環境的INFO層級啟用工作記錄。如需詳細資訊，請參閱在 Amazon 中查看氣流日誌 CloudWatch。

  b. 確認環境執行角色具有正確的權限原則。

  c. 請確認您的運算子或工作是否正常運作、有足夠的資源來剖析 DAG，並具有適當的 Python 程式庫來載入。要驗證您是否具有正確的依賴關係，請嘗試刪除導入，直到找到導致問題的導入。我們建議您使用 Amazon MWAA 本機執行器工具來測試您的 Python 相依性。

任務失敗，沒有任何日誌

如果工作流程中的工作失敗，而您找不到失敗工作的任何記錄檔，請檢查您是否要在預設引數中設定queue參數，如下所示。

```
from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from airflow.utils.dates import days_ago

# Setting queue argument to default.
default_args = {
 "start_date": days_ago(1),
 "queue": "default"
}

with DAG(dag_id="any_command_dag", schedule_interval=None, catchup=False,
 default_args=default_args) as dag:
    cli_command = BashOperator(
        task_id="bash_command",
        bash_command="{{ dag_run.conf['command'] }}"
    )
```

若要解決此問題，請從程式碼queue中移除，然後再次叫用 DAG。

我看到一個 ResourceAlreadyExistsException " 錯誤 CloudTrail

```
"errorCode": "ResourceAlreadyExistsException",
    "errorMessage": "The specified log stream already exists",
    "requestParameters": {
        "logGroupName": "airflow-MyAirflowEnvironment-DAGProcessing",
        "logStreamName": "scheduler_cross-account-eks.py.log"
    }
```

某些 Python 需求，例如apache-airflow-backport-providers-amazon將 Amazon MWAA 用來與之通訊的程式watchtower庫復原 CloudWatch 至舊版本。建議下列步驟：

- 將下列程式庫新增至 `requirements.txt`

  ```
  watchtower==1.0.6
  ```

我看到「無效請求」錯誤 CloudTrail

```
Invalid request provided: Provided role does not have sufficient permissions for s3
 location airflow-xxx-xxx/dags
```

如果您要使用相同的範本建立 Amazon MWAA 環境和 Amazon S3 儲存貯體，則需要在AWS CloudFormation範本中新增DependsOn區段。AWS CloudFormation這兩個資源 (MWAA 環境和 MWAA 執行原則) 在. AWS CloudFormation 建議下列步驟：

- 將下列**DependsOn**陳述式新增至您的AWS CloudFormation範本。

```
...
      MaxWorkers: 5
      NetworkConfiguration:
        SecurityGroupIds:
          - !GetAtt SecurityGroup.GroupId
        SubnetIds: !Ref subnetIds
      WebserverAccessMode: PUBLIC_ONLY
    DependsOn: MwaaExecutionPolicy

    MwaaExecutionPolicy:
    Type: AWS::IAM::ManagedPolicy
    Properties:
      Roles:
        - !Ref MwaaExecutionRole
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action: airflow:PublishMetrics
            Resource:
...
```

如需範例，請參閱 Amazon Apache 氣流管理工作流程的快速入門教學。

我在 Apache 氣流日誌中看到「找不到 64 位元的 Oracle 用戶端程式庫：「libclntsh.so：無法開啟共享物件檔案：沒有這樣的檔案或目錄」

- 建議下列步驟：
  - 如果您使用的是 Apache 氣流 v2，請添加`core.lazy_load_plugins : False`為 Apache 氣流配置選項。若要深入瞭解，請參閱使用設定選項載入外掛程式 2。

我在我的調度程序日誌中看到 psycopg2「服務器意外關閉了連接」

如果您看到類似下列內容的錯誤，表示您的 Apache 氣流排程器可能已耗盡資源。

```
2021-06-14T10:20:24.581-05:00    sqlalchemy.exc.OperationalError:
 (psycopg2.OperationalError) server closed the connection unexpectedly
2021-06-14T10:20:24.633-05:00    This probably means the server terminated abnormally
2021-06-14T10:20:24.686-05:00    before or while processing the request.
```

建議下列步驟：

- 考慮升級到 Apache 氣流 v2.0.2，它允許您指定最多 5 個排程器。

我看到 '執行程序報告任務實例 %s 已完成（%s），儘管任務在我的 DAG 處理日誌中顯示其 %s

如果您看到類似以下內容的錯誤，則長時間執行的任務可能已達到 Amazon MWAA 上的任務時間限制。Amazon MWAA 對於任何一項 Airflow 任務都有 12 小時的限制，以防止任務卡在佇列中並封鎖自動調度資源等活動。

```
Executor reports task instance %s finished (%s) although the task says its %s. (Info:
 %s) Was the task killed externally
```

建議下列步驟：

- 請考慮將工作分解為多個較短的執行工作。氣流通常具有一個模型，其中運營商是異步的。它調用外部系統上的活動，Apache 氣流傳感器輪詢以查看其完成時間。如果感測器故障，可以安全地重試，而不會影響操作員的功能。

我看到「無法從 log_group 讀取遠程日誌：氣流-* {* 環境名稱}-任務日誌流：* {* DAG_ID} /* {* TASK_ID} /* {* 時間} /* {* n}. 日誌。」 在我的任務日誌中

如果您看到類似下列內容的錯誤，則您環境的執行角色可能不包含建立工作記錄資料流的權限原則。

```
Could not read remote logs from log_group: airflow-*{*environmentName}-Task
 log_stream:* {*DAG_ID}/*{*TASK_ID}/*{*time}/*{*n}.log.
```

建議下列步驟：

- 使用的其中一個範例原則修改您環境的執行角色the section called “執行角色”。

您可能還在`requirements.txt`檔案中指定了與 Apache Airflow 版本不相容的提供者套件。例如，如果您使用的是 Apache 氣流 v2.0.2，您可能已經指定了一個套件，例如僅與氣流 2.1 + 相容的apache-airflow-providers-databricks套件。

建議下列步驟：

1. 如果您使用的是阿帕奇氣流 v2.0.2，修改`requirements.txt`文件並添加。`apache-airflow[databricks]`這會安裝與 Apache 氣流 v2.0.2 相容的正確版本的資料庫套件。
2. 使用 aws-mwaa-local-runneron GitHub 在本機測試您的 DAG、自訂外掛程式和 Python 相依性。

Amazon MWAA 文檔歷史記錄

下表說明從 2020 年 11 月開始，Amazon MWAA 服務文件的重要新增內容。若要接收有關此文件更新的通知，請訂閱 RSS 摘要。

變更	描述	日期
Support 較大的執行個體大小	Amazon MWAA 現在支援兩個較大的執行個體大小選項，適用於較大的工作負載：`mw1.xlarge` `mw1.2xlarge` • the section called “環境能力”	2024年4月16日
新的阿帕奇氣流版本	Amazon MWAA 現在支持阿帕奇氣流 v2.8.1。此更新包含有關更新的供應商套件的資訊，以及在 Amazon MWAA 上使用 Apache 氣流 v2.8.1 的詳細資訊。 • 版本 • the section called “阿帕奇氣流 v2.8.1 連接的提供程序包”	2024年2月22 日
Support 共用的 Amazon VPC	Amazon MWAA 支援使用 Amazon OpenSearch 服務的組織建立跨帳戶環境，以使用擁有者帳戶中的中央共用 Amazon VPC 來管理 Amazon MWAA 資源。作為此次推出的一部分，Amazon MWAA 可讓您選擇建立和管理自己的 Amazon VPC 端點。	2023 年 11 月 15 日

	• the section called “管理您自己的 Amazon VPC 端點”	
新的阿帕奇氣流版本	Amazon MWAA 現在支持阿帕奇氣流 v2.7.2。此更新包含有關更新的供應商套件的資訊，以及在 Amazon MWAA 上使用 Apache 氣流 v2.7.2 的詳細資訊。 • 版本 • the section called “阿帕奇氣流 v2.7.2 連線的提供者套件”	2023 年 11 月 6 日
新的阿帕奇氣流版本	Amazon MWAA 現在支持阿帕奇氣流 v2.6.3。此更新包含有關更新的供應商套件的資訊，以及有關在 Amazon MWAA 上使用 Apache 氣流 v2.6.3 的詳細資訊， • 版本 • the section called “阿帕奇氣流 v2.6.3 連接的提供程序包”	2023 年 8 月 9 日
版本棄用資訊	更新了有關版本棄用的主題，以包含 Apache 氣流 v2.0.2 和 Apache 氣流 v2.2.2 的棄用通知和時間表。 • the section called “阿帕奇氣流棄用版本”	2023 年 7 月 31 日

新主題和使用案例	Amazon MWAA 支援次要版本升級。此更新包含下列新主題，說明如何升級環境，並確定您的工作流程資源與您要升級至的 Apache Airflow 版本相容： • the section called “升級版本”	2023 年 6 月 5 日
更新主題	更新的客戶受管身分與存取權管理政策，可授與使用者對 Amazon MWAA 的完整主控台和 API 存取權。此更新說明為什麼您必須提供`iam:PassRole` 權限，才能允許使用者將角色傳遞給 Amazon MWAA。Amazon MWAA 使用這些許可代表使用者執行動作。 • the section called “存取 Amazon MWAA 環境”	2023 年 4 月 12 日
新指引	已更新有關設定 AWS Secrets Manager 為 Amazon MWAA 後端的主題，以提供使用查閱模式的指導。使用查閱模式可縮小 Apache 氣流搜尋的密碼範圍，並減少 Amazon MWAA 對 Secrets Manager 擷取連線和變數所做的 API 呼叫數量。這樣可以降低與使用 Secrets Manager 作為後端相關的成本。 • 將 Secrets Manager 後端建立為 Apache 氣流組態選項	2023 年 4 月 12 日

新的阿帕奇氣流版本	Amazon MWAA 現在支持阿帕奇氣流 v2.5.1。此更新包含有關更新的供應商套件的資訊，以及有關在 Amazon MWAA 上使用 Apache 氣流 v2.5.1 的詳細資訊， • 版本 • the section called “阿帕奇氣流 v2.5.1 連接的提供程序包”	2023 年 4 月 11 日
新主題和使用案例	已新增有關在 Amazon MWAA 環境中使用啟動指令碼的新主題。本主題說明為現有環境設定啟動指令碼、使用它來安裝 Linux 執行階段，以及設定環境變數。 • the section called “使用啟動腳本”	2023 年 4 月 3 日
更新了私人 Web 服務器訪問的部分	更新了以下有關私人 Web 服務器訪問的主題。此更新澄清了，在具有私有 Web 服務器訪問權限的環境中，您必須使用 Python wheel 歸檔（`.whl`）來打包和安裝依賴項。 • 私人網頁伺服器存取模式	2023 年 2 月 24 日

增加了對不推薦使用的 Apache 氣流版	更新了版本主題，其中包含 Amazon MWAA 如何管理取代 Apache 氣流版本的新資訊。刪除了有關升級到 Apache 氣流的新版本的部分，以及描述 Apache 氣流 v1 和 Apache 氣流 V2 之間更改的部分。如需有關移轉至新版 Apache 氣流的詳細資訊，請參閱 Amazon MWAA 移轉指南。 • the section called “阿帕奇氣流棄用版本” • the section called “Apache 氣流版本支持和常見問題”	2023 年 2 月 17 日
修正了 Amazon MWAA 容器指標	已更新容器量度主題，並移除維度下不存在的一組錯誤量度。Cluster新增額外段落，說明如何透過繪製AdditionalWorker 元件的CPUUtilization 或MemoryUtilization 測量結果，並將統計資料類型設為，來SampleCount評估環境在指定時間使用的其他 Worker 數目。 • the section called “評估其他工作者實例的數量”	2023 年 1 月 20 日

新的阿帕奇氣流版本	Amazon MWAA 現在支持阿帕奇氣流 v2.4.3。此更新包含有關更新的供應商套件的資訊、有關在 Amazon MWAA 上使用 Apache 氣流 v2.4.3 的詳細資訊，以及 Amazon MWAA 上每個 Apache 氣流版本支援哪些功能的整合資訊。 • 版本 • the section called “阿帕奇氣流 v2.4.3 連接的提供程序包”	2023 年 1 月 5 日
已更新服務連結角色的主題	已更新 Amazon MWAA 用來代表您建立和管理 AWS 資源的服務連結角色資訊，包括如何在不再需要服務連結角色時刪除該角色的相關資訊。這包括更新的服務連結角色權限政策，可讓 Amazon MWAA 在命名空間下發佈其他 CloudWatch 指標。`AWS/MWAA` • the section called “服務連結角色”	2022 年 11 月 18 日

有關服務指標的新主題	已新增新主題，說明 Amazon MWAA 在命名空間下發出的服務指標。AWS/MWAA其中包括 Amazon ECS 叢集指標排程器、工作者和網頁伺服器、可讓 Amazon MWAA 分離排程器和工作者之佇列的 Amazon SQS 指標，以及中繼資料資料庫的 Amazon RDS 指標。 • the section called “容器、佇列和資料庫測量結果”	2022 年 11 月 18 日
新主題	已新增修改條件約束檔案的新指引，以指定要與 Amazon MWAA 環境搭配使用的供應商套件的新版本。 • the section called “指定新的提供者套件”	2022 年 11 月 18 日
更新的常見問題	已更新與 Amazon MWAA HIPAA 資格相關的資訊。 • the section called “HIPAA 合規”	2022 年 11 月 15 日
新主題	新增有關在 Amazon MWAA 執行角色信任政策中使用aws:SourceArn 和aws:SourceAccount 全域條件上下文金鑰的新主題，以防止跨服務混淆的副手。 • the section called “預防跨服務混淆代理人”	2022 年 10 月 21 日

新的範例程式碼	已新增將自訂作業系統層級量度寫入的更新指示和 DAG 程式碼範例。 CloudWatch • the section called “使用 DAG 寫入自訂指標”	2022 年 9 月 13 日
新的範例程式碼	已新增更新指示和新的 AWS Lambda Python 程式碼範例，可擷取 Apache 氣流 CLI 權杖，然後在指定的 Amazon MWAA 環境中叫用 DAG。 • the section called “使用拉姆達調用 DAG”	2022 年 9 月 12 日
新建築圖	新增新的架構圖表，展示 Amazon MWAA 環境搭配公有和私有 Web 伺服器。 • the section called “阿帕奇氣流存取模式”	2022 年 9 月 12 日
新的範例程式碼	已新增更新指示和新的 DAG 程式碼範例，可擷取 Apache 氣流 CLI 權杖，然後在不同的 Amazon MWAA 環境中叫用另一個 DAG。 • the section called “在不同環境中叫用 DAG”	2022 年 8 月 16 日

新的範例程式碼	新增更新的指示和新的 DAG，可查詢環境的 Aurora PostgreSQL 以取得中繼資料資訊、將結果寫入 CSV 檔案，並將檔案存放在 Amazon S3 中。 • the section called “將環境中繼資料匯出到 Amazon S3”	2022 年 8 月 12 日
新的範例程式碼	新增更新的指示和新的 DAG，可在執行階段重新整理 AWS CodeArtifact 權杖，並將結果儲存在 Amazon S3 中。 • the section called “刷新AWS CodeArtifact運行時令牌”	2022 年 8 月 3 日
新的範例程式碼	已新增`ECSOperator` 在 Amazon MWAA 中使用的更新指示和 DAG 程式碼範例。 • the section called “使用 `ECSOperator` ”	2022 年 7 月 26 日
新的範例程式碼	已新增`SSHOperator` 在 Amazon MWAA 中使用的更新指示和 DAG 程式碼範例。 • the section called “使用 `SSHOperator` ”	2022 年 7 月 15 日
新的範例程式碼	添加了新的指示和 DAG 代碼示例，用於將 dbt Postgres 與 Amazon MWAA 一起使用。 • the section called “使用 dbt 與亞馬遜 MWAA”	2022 年 6 月 17 日

新主題和使用案例	已新增新指示和 DAG 程式碼範例，以使用 Python 輪子檔案在具有公開和私有存取權的 Amazon MWAA 環境中安裝相依性。 • 使用 Python 輪子管理依賴關係	2022 年 5 月 13 日
新主題和使用案例	已新增有關選擇 Amazon MWAA 傳送到哪些 Apache 氣流指標的新指引。 CloudWatch • 選擇要報告的 Apache 氣流量度	2022 年 4 月 19 日
新指南	Amazon MWAA 提供遷移指南，說明如何從自我管理部署和現有的 Amazon MWAA 環境遷移 Apache 氣流工作流程。 • Amazon MWAA 遷移指南	2022 年 3 月 7 日
新主題和使用案例	為使用 Apache Airflow 新增安全性最佳作法，包括偵測 Apache 氣流使用者權限變更的解決方案。 • the section called “Apache 氣流中的安全性最佳做法”	2022 年 2 月 18 日
新的範例程式碼	已新增使用擺錘建立時區感知 DAG 的新程式碼範例，並說明如何使用自訂外掛程式變更建立 Apache Airflow 記錄的時區。 • the section called “變更 DAG 的時區”	2022 年 2 月 11 日

阿帕奇氣流 v2.2.2 發射	Amazon 管理的阿帕奇氣流工作流程現在支持 Apache 氣流 v2.2.2。從 2.2 版開始，Amazon MWAA 將直接在 Apache 氣流網頁伺服器上安裝 Python 套件和自訂外掛程式，讓您更有彈性地管理環境。如需更多資訊，請參閱下列內容。 • 適用於 Apache 氣流的 Amazon 氣流管理工作流程. • the section called “阿帕奇氣流 v2.2.2 連接的提供程序包” . • 阿帕奇氣流 v2.2.2 更新日誌阿帕奇氣流文檔網站上。	2022 年 1 月 27 日
新教學課程	新增了一個新的教學課程，示範如何建立新的自訂 Apache Airflow 角色，並將角色指派給從 IAM 對應的 Apache Airflow 使用者，以限制使用者對指定 DAG 子集的存取權限。 • the section called “教學課程：將使用者限制為 DAG 的子集”	2021 年 12 月 8 日

修正項目	已修正設定值以最佳化 CPU 使用率`scheduler.min_file_process_interval` 的最佳作法建議。已新增 IAM 政策範例，以執行角色授與 Secrets Manager 資源的存取權。已新增使用 Secrets Manager 條件金鑰的疑難排解主題 • 排程器剖析 DAG 的效能調整 • 向 Amazon MWAA 提供存取 Secrets Manager 金鑰的權限 • 在 Secrets Manager 的 Amazon MWAA 執行角色中設定條件金鑰	2021 年 11 月 22 日
新的範例程式碼	已新增下列新程式碼範例，用於修改使用自訂外掛程式處理 DAG 的時區，以及從 bash 運算子中叫用 `dags backfill` Apache Airflow CLI 命令的新疑難排解主題。 • the section called “變更 DAG 的時區” • 使用 bash 運算符回填 CLI 命令	2021 年 11 月 1 日
修正項目	已修正 Amazon ECS 操作員程式碼範例中的問題，並澄清 Amazon MWAA 執行角色中允許環境存取日誌中 Amazon ECS 任務日誌群組所需的其他許可。 CloudWatch • Amazon ECS 操作員許可。	2021 年 10 月 26 日

新的範例程式碼	已新增新程式碼範例，用於查詢 Aurora PostgreSQL 資料庫以取得與 DAG 執行相關的資訊，並將結果寫入存放在 Amazon S3 上的CSV檔案。 • the section called “將環境中繼資料匯出到 Amazon S3”.	2021 年 10 月 1 日
修正項目	已更正 Amazon MWAA 如何自動將目標 Amazon S3 儲存貯體中新的和變更的物件同步到排程器和工作者的資訊。 • DAG 資料夾的運作方式。	2021 年 10 月 1 日
現已支援	Amazon MWAA 現在支援 Apache 氣流 2.0 + 的其他供應商套件。若要深入瞭解支援的套件，請參閱下列內容： • the section called “阿帕奇氣流 v2.0.2 連接的提供程序包” .	2021 年 9 月 24 日
新命令和程序	新增額外的指導和 AWS CLI 命令範例，以便在使用 Amazon VPC 時建立 Amazon S3 閘道端點而無法存取網際網路： • 建立不存取網際網路的 Amazon VPC 人雲端網路。	2021 年 9 月 24 日

新主題和使用案例	新增下列變更： • 在中新增使用 Amazon 彈性容器服務運算子的新程式碼範例the section called “使用 ECSOperator ”。 • 針對在中設定 Apache 氣流外掛程式的問題新增疑難排解主題the section called “外掛程式”。	2021年9月19日
新的支援地區	Amazon MWAA 現在可在下列區域使用： • 亞太區域 (孟買) – ap-south-1 • 亞太區域 (首爾) – ap-northeast-2 • 歐洲 (倫敦) – eu-west-2 • 歐洲 (巴黎) – eu-west-3 • 加拿大 (中部) – ca-central-1 • 南美洲 (聖保羅) – sa-east-1 如需區域可用性和服務端點的詳細資訊，請參閱下列內容： • Amazon MWAA 端點和. AWS 一般參考	2021 年 8 月 31 日

新主題和使用案例	新增下列變更： • 已更新範例政策，允許亞馬遜 MWAA 在中擷取帳戶層級的 Amazon S3 設定 ()。`s3:GetAccountPublicAccessBlock` Amazon MWAA 執行角色	2021 年 8 月 27 日
修正項目	新增下列變更： • 已修正 AWS CloudFormation 範本，以針對中建立虛擬私人雲端網路的安全性群組使用自我參照輸入規則。 • 已修正 AWS CloudFormation 範本，以針對中Amazon Apache 氣流管理工作流程的快速入門教學的安全性群組使用自我參照輸入規則。	2021 年 8 月 27 日
新主題和使用案例	新增下列變更： • 將 DAG 裝飾器添加到 Apache 氣流 v 適用於 Apache 氣流的 Amazon 氣流管理工作流程 2.0.2 支持的列表中。	2021 年 8 月 20 日

新主題和使用案例	新增下列變更： • 已將`celery.sync_parallelism` 使用案例新增至Amazon MWAA 上阿帕奇氣流的性能調整. • 已將服務端點新增至配額頁面，並將名稱變更為Amazon Managed Work(Amazon Managed Work與配額。 • 根據使用者的意見反應，釐清網路必要條件，位 開始使用 Amazon Managed Workflows • 已移`dags list-runs` `dags next-execution` 至中不支援的氣流 CLI 命令阿帕奇氣流 CLI 命令參考。	2021 年 8 月 13 日
新的範例程式碼	新增下列變更： • 添加 bash 示例來設置，獲取或刪除阿帕奇氣流 v2.0.2 變量。阿帕奇氣流 CLI 命令參考 • 將 Apache 氣流 v2.0.2 依賴關係和氣流連接示例添加到. 使用亞馬遜 MWAA 與亞馬遜 RDS 微軟 SQL 服務器	2021 年 8 月 13 日
修正項目	新增下列變更： • 修正了基於用戶反饋的 Python 代碼示例使用建立 SSH 連線 `SSHOperator` 。	2021 年 8 月 13 日

新主題和使用案例	新增下列變更： • 已移`variables set`至中支援的氣流 CLI 命令阿帕奇氣流 CLI 命令參考。 • 新增了 v2.0.2 版本從 Airflow 版本頁面變更為安裝 Python 的依賴根據使用者意見反應的摘要。 • 新增了 v2.0.2 版本從 Airflow 版本頁面變更為阿帕奇氣流 CLI 命令參考根據使用者意見反應的摘要。 • 新增了 v2.0.2 版本從 Airflow 版本頁面變更為連線類型概觀根據使用者意見反應的摘要。 • 新增了 v2.0.2 版本從 Airflow 版本頁面變更為安裝自定義插件根據使用者意見反應的摘要。 • 新增了 v2.0.2 版本從 Airflow 版本頁面變更為新增或更新 DAG根據使用者意見反應的摘要。	2021 年 8 月 6 日
新的範例程式碼	新增下列變更： • 增加了阿帕奇氣流 v2.0.2 示例代碼。使用 DAG 在 CLI 中匯入變數 • 增加了阿帕奇氣流 v2.0.2 示例代碼。使用拉姆達函數調用 DAG	2021 年 8 月 6 日

新主題和使用案例	新增下列變更： • 已新增「我在 Airflow UI 中看不到我的連線」的疑難排解主題，位Amazon Managed Workflows Managed Workflows Managed Workflows於。 • 添加了 Amazon VPC Amazon MWAA 支持的列表。關於 Amazon MWAA 上的聯網	2021 年 7 月 29 日
修正項目	新增下列變更： • 修復了基於用戶反饋的 Python 代碼示例，以打印 Web 登錄令牌創建一個 Apache 氣流網絡登錄令牌。 • 修正了雪花連接主題根據用戶反饋使用倉庫參數的單引號Amazon Managed Workflows Managed Workflows Managed Workflows。	2021 年 7 月 29 日
已移除或移動的主題	新增下列變更： • 重組現有頁面以包含中的所有監視和指標文件頁面。針對 Apache 氣流的 Amazon 受管工作流程監控和指標 • 移阿帕奇氣流 v2 環境指標 CloudWatch至監控和指標導覽功能表。	2021 年 7 月 23 日

新指南	新增下列變更： • 已建立Amazon MWAA 環境上安裝的 Apache 氣流供應商套件。 • 已建立在 Amazon MWAA 上監控概觀。 • 已建立檢視稽核記錄 AWS CloudTrail。 • 已建立在 Amazon 中查看氣流日誌 CloudWatch。	2021 年 7 月 23 日
修正項目	新增下列變更： • 已修正基於使用者意見反應的 Python 程式碼範例，以正確的順序產生 Airflow 連線字串，並在中新增連接埠參數使用AWS Secrets Manager機密來設定 Apache Airflow 連線。 • 已新增根據中的使用者意見反應在使用 Oracle 創建一個自定義插件本機安裝解壓縮套件的步驟。	2021 年 7 月 23 日

新主題和使用案例	新增下列變更： • 已在Amazon MWAA 常見問題中新增 AWS DMS 運算子的主題。 • 添加了遠程日誌錯誤的故障排除主題Amazon Managed Workflows Managed Workflows Managed Workflows。 • 已移`variables set`至中不支援的氣流 CLI 命令阿帕奇氣流 CLI 命令參考。	2021 年 7 月 16 日
新主題和使用案例	新增下列變更： • 已新增循序步驟，以根據使用者的意見反應建立 requirements.txt 檔案，位於安裝 Python 的依賴。 • 已新增循序步驟，以根據使用者的意見反應建立 plugins.zip 檔案，位於安裝自定義插件。 • 在 Amazon Apache 氣流 API 受管工作流程參考指南中的 API 參考指南中，在使用者指南中新增交互參考連結。 • 增加了主題為什麼插件沒有顯示在氣流 2.0 管理 > 插件菜單中Amazon MWAA 常見問題。	2021 年 7 月 9 日

新指南	新增下列變更： • 已建立刪除 Amazon S3 上的文件。	2021 年 7 月 9 日
新主題和使用案例	新增下列變更： • 已在新增支援值的清單使用客戶管理的金鑰進行加密。 • 根據中在 requirements.txt Python 管理依賴項的使用者意見，更新並澄清了私有存放庫 URL 的範例。	2021 年 7 月 2 日
新的範例程式碼	新增下列變更： • 增加了阿帕奇氣流 v1.10.12 示例代碼，以使用私鑰 AWS Secrets Manager 進行 SSH 連接。使用建立 SSH 連線 SSHOperator	2021 年 7 月 2 日
新主題和使用案例	新增下列變更： • 已新增 StartedTaskInstances 和 FinishedTaskInstances 指標阿帕奇氣流 v2 環境指標 CloudWatch。	2021 年 6 月 25 日
新的範例程式碼	新增下列變更： • 增加了阿帕奇氣流 v2.0.2 示例代碼在。使用亞馬遜 MWAA 與亞馬遜 EKS	2021 年 6 月 25 日
新指南	新增下列變更： • 已建立Amazon MWAA 上阿帕奇氣流的性能調整。	2021 年 6 月 25 日

新主題和使用案例	新增下列變更：	2021 年 6 月 18 日

- 添加`connections addconnections delete`到支持的阿帕奇氣流 v2.0.2 CLI 命令在. 阿帕奇氣流 CLI 命令參考
- 補充說，可用的最新版本 AWS CloudFormation 是阿帕奇氣流 v2.0.2 在. Amazon Apache 氣流管理工作流程的快速入門教學
- 已新增將 Apache 氣流工作者暫存資料儲存至Amazon MWAA 常見問題.
- 添加了「執行程序報告任務實例 %s 已完成」錯誤的主題。Amazon Managed Workflows Managed Workflows Managed Workflows
- 添加了「服務器意外關閉連接」日誌的主題。Amazon Managed Workflows Managed Workflows Managed Workflows
- 已新增範例，在 SSH 通道上執行 CLI 命令至防禦主機。創建一個阿帕奇氣流 CLI 令牌
- 已將隨機產生的使用者名稱主題新增至。Amazon Managed Workflows Managed Workflows Managed Workflows

- 已新增在 CLI 中執行 DAG 時發生 503 錯誤的主題Amazon Managed Workflows Managed Workflows Managed Workflows。
- 已新增 Apache Airflow v2.0.2 中自訂外掛程式的主題，這些外掛程式需要在每個氣流程開始時載入外掛程式的氣流組態選項，以覆寫版本的預設設定。`core.lazy_load_plugins : False` 在 Amazon MWAA 上使用阿帕奇氣流配置選項
- 增加了氣流配置選項步驟阿帕奇氣流 v2.0.2 插件示例代碼在. 創建一個自定義插件與阿帕奇蜂巢和 Hadoop
- 增加了氣流配置選項步驟阿帕奇氣流 v2.0.2 插件示例代碼在. 建立產生執行階段環境變數的自訂外掛程式
- 增加了氣流配置選項步驟阿帕奇氣流 v2.0.2 插件示例代碼在. 為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator
- 增加了氣流配置選項步驟阿帕奇氣流 v2.0.2 插件示例代碼在. 使用 Oracle 創建一個自定義插件

新的範例程式碼	新增下列變更： • 增加了示例代碼為 Apache 氣流雪花連接在使用密鑰 AWS Secrets Manager阿帕奇氣流雪花連接。	2021 年 6 月 18 日

新主題和使用案例	新增下列變更：	2021 年 6 月 2 日

- 將伺服器端加密指引新增至為 Amazon MWAA 儲存貯貯存貯貯貯存貯貯貯存貯貯貯.
- 增加了阿帕奇氣流 v2.0.2 的秘密後端。使用AWS Secrets Manager機密來設定 Apache Airflow 連線
- 已新增 Apache 氣流工作者配額增加要求的問題Amazon MWAA 常見問題。
- 已新增問題，說明使用哪些量度來判斷是否要將 Apache 氣流工作者擴展至Amazon MWAA 常見問題。
- 已新增在中建立自訂量度的 CloudWatch 問題Amazon MWAA 常見問題。
- 已新增步驟，針對具有私有路由的 VPC 啟用 Amazon S3 VPC 界面端點的私有 IP 地址。使用私有路由在 Amazon VPC 中建立所需的虛擬私人雲端服務端點
- 添加了使用本地端口轉發來設置 SSH 隧道的選項教學課程：使用 Linux 防禦主機設定私人網路存取。

新的範例程式碼	新增下列變更： • 已新增 DAG 的範例程式碼，該程式碼可查詢 Amazon Aurora PostgreSQL 中繼資料資料庫，並將自訂指標發佈至 Amazon CloudWatch 網站 使用 DAG 寫入自訂指標CloudWatch	2021 年 6 月 2 日
新指南	新增下列變更： • 在中建立了如何在 Apache 氣流使用者介面中互換使用連線範本的指南。連線類型概觀	2021 年 6 月 2 日
修正項目	新增下列變更： • 在選項三：建立沒有網際網路存取的 VPC 人雲端網路中將 Apache 氣流 VPC 人雲端端點新增至 AWS CloudFormation 範本。建立虛擬私人雲端網路	2021 年 6 月 2 日

阿帕奇氣流 v2.0.2 發射	一般可用性推出阿帕奇氣流 v2.0.2。 • 已建立適用於 Apache 氣流的 Amazon 氣流管理工作流程。 • 已建立阿帕奇氣流 v2 環境指標 CloudWatch。 • 增加了阿帕奇氣流 v2.0.2 到版本特定的鏈接。在 Amazon MWAA 上使用阿帕奇氣流配置選項 • 將 Apache 氣流 v2.0.2 版本特定的指導添加到. 安裝 Python 的依賴 • 將 Apache 氣流 v2.0.2 版本特定的指導添加到. 在 requirements.txt Python 管理依賴項 • 增加了阿帕奇氣流 v2.0.2 樣本插件。安裝自定義插件 • 增加了阿帕奇氣流 v2.0.2 示例代碼。Amazon MWAA 環境上的 Aurora 資料庫清理 • 增加了阿帕奇氣流 v2.0.2 示例代碼。使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接 • 增加了阿帕奇氣流 v2.0.2 示例代碼。為 Apache 氣流創建一個自定義插件PythonVirtualenvOperator • 添加了阿帕奇氣流 v2.0.2 命令到. 阿帕奇氣流 CLI 命令參考	2021 年 5 月 26 日

	• 增加了阿帕奇氣流 v2.0.2 腳本到. 創建一個阿帕奇氣流 CLI 令牌 • 添加了一個說明，Amazon MWAA 默認情況下使用最新的 Apache 氣流版本。建立一個 Amazon MWAA 環境	
新主題和使用案例	新增下列變更：	2021 年 5 月 14 日
	• 已新增疑難排解卡住或未執行的氣流工作的指引Amazon Managed Workflows Managed Workflows Managed Workflows。	
修正項目	新增下列變更：	2021 年 5 月 12 日
	• 我們已更新範例外掛程式程式碼，以便在中使用最新的 Java 版本創建一個自定義插件與阿帕奇蜂巢和 Hadoop。以前，它是`os.environ["JAVA_HOME"]="/usr/lib/jvm/jre-1.8.0-openjdk-1.8.0.272.b10-1.amzn2.0.1.x86_64"` 。	
已移除或移動的主題	新增下列變更：	2021 年 5 月 10 日
	• 依類別將主題移Amazon Managed Workflows Managed Workflows Managed Workflows至新頁面。	

新主題和使用案例	新增下列變更： • 將 Amazon S3 存儲桶概述添加到在亞馬遜 MWAA 與 DAG 工作.	2021 年 5 月 10 日
已移除或移動的主題	新增下列變更： • 移訪問阿帕奇氣流用戶界面至頂層導覽，並新增創建一個 Apache 氣流網絡登錄令牌創建一個阿帕奇氣流 CLI 令牌、和的頁面阿帕奇氣流 CLI 命令參考。	2021 年 5 月 7 日
新主題和使用案例	新增下列變更： • 針對中所有受支援和不支援的氣流 CLI 命令，新增了 Apache Airflow 參考指南的版本特定連結。阿帕奇氣流 CLI 命令參考 • 針對中的所有組態選項，新增了 Apache Airflow 參考指南的版本特定連結。在 Amazon MWAA 上使用阿帕奇氣流配置選項 • 已將 Amazon MWAA CLI 公用程式新增到. 在 requirements.txt Python 管理依賴項	2021 年 5 月 7 日

新主題和使用案例	新增下列變更： • 增加了如何在中構建 plugins.zip 的平坦和嵌套示例安裝自定義插件。 • 已將 Amazon MWAA CLI 公用程式新增至新增或更新 DAG安裝自定義插件、和安裝 Python 的依賴頁面。 • 根據中的使用者意見反應和頁面，將內容重新架構為概觀、上傳到 Amazon S3，然後在 安裝自定義插件 Amazon MWAA 區段上安裝。安裝 Python 的依賴 • 新增範例使用案例，可建立必要的 VPC 端點並將其連接到現有的 Amazon VPC，而無需存取網際網路。關於 Amazon MWAA 上的聯網	2021 年 4 月 30 日
新的範例程式碼	新增下列變更： • 已新增範例程式碼，在秘 Secrets Manager 中針對中的 Apache 氣流變數使用秘密金鑰 使用密鑰AWS Secrets Manager對於一個阿帕奇氣流變量	2021 年 4 月 30 日
新指南	新增下列變更： • 已建立使用私有路由在 Amazon VPC 中建立所需的虛擬私人雲端服務端點。	2021 年 4 月 30 日

修正項目	新增下列變更： • 哎呀！我們已更新`core.default_ui_timezone` 為`webserver.default_ui_timezone` 中在 Amazon MWAA 上使用阿帕奇氣流配置選項。	2021 年 4 月 30 日
新主題和使用案例	新增下列變更： • 為 SSH 通道添加了視窗（PuTTY）步驟。教學課程：使用 Linux 防禦主機設定私人網路存取 • 增加了主題`apache-airflow-providers-amazon` ，這是只與 Apache 氣流 2.0 兼容到Amazon Managed Workflows Managed Workflows Managed Workflows.	2021 年 4 月 23 日
新的範例程式碼	新增下列變更： • 已新增範例程式碼，在秘密管理員中使用秘密金鑰進行 Apache 氣流連線使用秘密金鑰AWS Secrets Manager對於一個阿帕奇氣流連接。	2021 年 4 月 23 日

新指南	新增下列變更：	2021 年 4 月 23 日

- 已建立關於 Amazon MWAA 上的聯網。
- 已建立Amazon MWAA 上 VPC 的安全政策。
- 已建立在 Amazon MWAA 上管理服務特定 Amazon VPC 端點的存取。

新主題和使用案例	新增下列變更：	2021 年 4 月 16 日

- 添加了一個新的 AWS CloudFormation 模板來創建一個沒有互聯網訪問的 Amazon VPC 網絡。建立虛擬私人雲端網路
- 添加了一個新的教程來創建一 AWS Client VPN 個教學課程：使用AWS Client VPN。
- 根據使用者意見反應，將 [網路存取] 頁面的名稱變更為 Apache Airflow 存取模式，並簡化中的文件阿帕奇氣流存取模式。
- 簡化文件，根據使用者意見反應，僅包含 Amazon VPC 入門資訊和範本。建立虛擬私人雲端網路
- 新增 BigQuery 運算子解決方法至Amazon Managed Workflows Managed Workflows Managed Workflows.
- 添加了一個阿帕奇氣流 v1.10.12 約束文件最佳實踐到. 安裝 Python 的依賴

新的範例程式碼	新增下列變更： • 添加了示例代碼來創建使用 Oracle 中的自定義插件使用 Oracle 創建一個自定義插件。 • 已新增範例程式碼以建立在中產生執行階段環境變數的自訂外掛程式建立產生執行階段環境變數的自訂外掛程式。 •	2021 年 4 月 16 日
新主題和使用案例	新增下列變更： • 已將 VPC 安全性群組上自我參照規則需求的主題新增至。Amazon MWAA 常見問題 • 增加了自定義插件目錄和大小限制安裝自定義插件。 • 已將需求目錄和大小限制新增至安裝 Python 的依賴. • 已釐清`foo.user`和`foo.pass`中的 Apache 氣流組態選項。在 requirements.txt Python 管理依賴項 • 將配置選項概述添加到在 Amazon MWAA 上使用阿帕奇氣流配置選項.	2021 年 4 月 9 日

新的範例程式碼	新增下列變更： • 添加示例代碼來創建使用 PythonVirtualenvOperator 中的自定義插件為 Apache 氣流創建一個自定義插件 PythonVirtualenvOperator。 • 添加示例代碼來創建一個自定義插件與阿帕奇蜂巢和 Hadoop 的創建一個自定義插件與阿帕奇蜂巢和 Hadoop。	2021 年 4 月 9 日
修正項目	新增下列變更： • 哎呀！我們已經更新了 requirements.txt 的格式，並新增了一個與 Apache 氣流 v1.10.12 相容的範例。安裝 Python 的依賴	2021 年 3 月 31 日
新主題和使用案例	新增下列變更： • 已新增將 requirements.txt 或 plugins.zip 移除至的解決方法Amazon MWAA 常見問題。 • 在環境中將 SSH 的 bash 解決方法添加到Amazon MWAA 常見問題. • 添加了 CloudTrail ResourceAlreadyExistsException 錯誤的主題Amazon Managed Workflows Managed Workflows Managed Workflows。	2021 年 3 月 26 日

新主題和使用案例	新增下列變更：	2021 年 3 月 19 日

- 添加了用於的 AWS 服務列表Amazon MWAA 執行角色。
- 添加了用於的 AWS 服務列表Amazon MWAA 的服務連結角色。
- 增加了對 Python 3.7 版本的問題 Amazon MWAA 到. Amazon MWAA 常見問題
- 已新增 PythonVirtualenvOperator 到的問題Amazon MWAA 常見問題。
- 已將疑難排解指令碼新增為與 VPC 和環境組態相關之所有主題的後續步驟，位Amazon Managed Workflows Managed Workflows Managed Workflows於。
- 澄清了 linux 堡壘必須與環境位於相同區域的文檔。教學課程：使用 Linux 防禦主機設定私人網路存取

新指南	新增下列變更： • 創建阿帕奇氣流連接指南 AWS Secrets Manager 在使用AWS Secrets Manager機密來設定 Apache Airflow 連線. • 使用 AWS CloudFormation 範本建立快速入門教學課程，以便在上建立 Amazon VPC 基礎設施、Amazon S3 儲存貯體和 Amazon MWAA 環境。Amazon Apache 氣流管理工作流程的快速入門教學	2021 年 3 月 19 日
新主題和使用案例	新增下列變更： • 已新增建立 Amazon S3 儲存貯體疑難排解主題Amazon Managed Workflows Managed Workflows Managed Workflows。 • 已新增建立和附加 JSON 政策的步驟Amazon MWAA 執行角色。	2021 年 3 月 12 日
新的範例程式碼	新增下列變更： • 已新增範例程式碼，以在觸發 DAG 時新增組態。訪問阿帕奇氣流用戶界面	2021 年 3 月 12 日
新指南	新增下列變更： • 在創建了最佳實踐指南在 requirements.txt Python 管理依賴項。	2021 年 3 月 12 日

新主題和使用案例	新增下列變更： • 添加谷歌/GCP /故障排除 BigQuery 主題。Amazon Managed Workflows Managed Workflows Managed Workflows • 將 Cython 疑難排解主題新增至Amazon Managed Workflows Managed Workflows Managed Workflows. • 將 MySQL 故障排除主題添加到Amazon Managed Workflows Managed Workflows Managed Workflows. • 將 5xx Web 服務器錯誤故障排除主題添加到Amazon Managed Workflows Managed Workflows Managed Workflows。	2021 年 3 月 5 日
現已支援	新增下列變更： • 先backend_kwargs 前不支援，而 AWS Secrets Manager 且您需要因應措施來覆寫 Secrets Manager 函數呼叫。現在backend_kwargs ，支持。請參閱中的 AWS Secrets Manager 疑難排解主題Amazon Managed Workflows Managed Workflows Managed Workflows。	2021 年 3 月 4 日

修正項目	新增下列變更： • 哎呀！我們已更新每個環境類別的大小，以反映中的實際 GB 設定 Amazon MWAA 環境類別。	2021 年 3 月 4 日
新主題和使用案例	新增下列變更： • 已將使用 VPC 端點策略的私人網路存取新增至阿帕奇氣流存取模式。 • 已將建立環境疑難排解主題的其他檢查新增至Amazon Managed Workflows Managed Workflows Managed Workflows。 • 已新增檢視至的記錄檔`requirements.txt` 的步驟安裝 Python 的依賴。	2021 年 2 月 26 日
新主題和使用案例	新增下列變更： • 已將 Apache 蜂巢使用案例新增至安裝 Python 的依賴. • 澄清了 Apache 氣流套件所需的相依性必須包含在檔案中的文`requirements.txt` 件，位安裝 Python 的依賴於. • 已將 requirements.txt 疑難排解主題更新至Amazon Managed Workflows Managed Workflows Managed Workflows.	2021 年 2 月 25 日

新教學課程	新增下列變更： • 將私人網路教學課程新增至教學課程：使用 Linux 防禦主機設定私人網路存取。	2021 年 2 月 22 日
新主題和使用案例	新增下列變更： • 已將私人和公用網路組態新增至阿帕奇氣流存取模式. • 已將開發群組使用案例和使用者案例新增至Amazon MWAA 執行角色.	2021 年 2 月 22 日
新的範例程式碼	新增下列變更： • 已將網頁登入權杖和 CLI 權杖的 Python 指令碼範例新增至訪問阿帕奇氣流用戶界面。 • 已新增範例程式碼，以在其他環境中觸發 DAG 至Amazon Managed Workflows. • 已新增範例程式碼，以使用 Lambda 函數來觸發 DAG 使用拉姆達函數調用 DAG。	2021 年 2 月 22 日
新命令和程序	新增下列變更： • 在的所有腳本中添加了逐步過程訪問阿帕奇氣流用戶界面。	2021 年 2 月 22 日

新的範例程式碼	新增下列變更： • 更新了網絡登錄令牌的 curl 示例訪問阿帕奇氣流用戶界面。 • 添加了示例代碼以連接到 Amazon RDS Microsoft SQL 服務器使用亞馬遜 MWAA 與亞馬遜 RDS 微軟 SQL 服務器。	2021 年 2 月 17 日
新命令和程序	新增下列變更： • 向在亞馬遜 MWAA 與 DAG 工作頁面添加了 AWS CLI 命令。 • 阿帕奇氣流不支援 CLI 命令中的序列化 DAG。由於 CLI 在 Web 伺服器上執行，基於安全性考量而沒有外掛程式或需求，因此任何具有 plugins.zip 或 requirements.txt 的 MWAA 環境都不支援這些命令。將 Apache 氣流`list_dags` 和`backfill`命令移至不支援的命令，位於訪問阿帕奇氣流用戶界面。	2021 年 2 月 17 日
GitHub 啟動	用戶指南文檔現在是開源的 GitHub。在任何頁面上選擇「編輯此頁面 GitHub」。	2021 年 2 月 17 日

新主題和使用案例	新增下列變更： • 已將 Step Functions 與 Amazon MWAA 使用案例的問題新增至. Amazon Managed Workflows Managed Workflows Managed Workflows • 已將 CLI 存取原則新增至存取 Amazon MWAA 環境. • 澄清任何支援 Apache 氣流組態選項的文件可以在在 Amazon MWAA 上使用阿帕奇氣流配置選項中指定。 • 澄清了文檔，如果一個可用區域中的 Fargate 容器發生故障，則 MWAA 切換到另一個容器位於的不同可用區域中。建立虛擬私人雲端網路	2021 年 2 月 12 日
新主題和使用案例	新增下列變更： • 新增了 設定 Amazon MWAA 環境類別。	2021 年 2 月 5 日
已移除或移動的主題	新增下列變更： • 刪除了 Amazon S3 存儲桶名稱開頭的`airflow-`要求開始使用 Amazon Managed Workflows。 • 移存取 Amazon MWAA 環境動Amazon MWAA 執行角色到管理對 Amazon MWAA 環境的存取。	2021 年 2 月 4 日

Amazon MWAA CloudFormation	更新參數以在 Amazon MWAA CloudFormation 上建立環境。 • 移除 SubnetList。 • 移除 TagList。 • 添加 NetworkConfiguration。 • 添加 TagMap。 • 新增建立環境要求範例。	2021 年 2 月 4 日
新主題和使用案例	新增下列變更： • 將電子郵件配置示例添加到在 Amazon MWAA 上使用阿帕奇氣流配置選項. • 將 PostgresHook 疑難排解主題新增至Amazon Managed Workflows Managed Workflows Managed Workflows. • 將 AWS Secrets Manager 疑難排解主題新增至Amazon Managed Workflows Managed Workflows Managed Workflows. • 將高性能用例添加到設定亞馬遜 MWAA 自動擴展.	2021 年 1 月 29 日
Amazon MWAA 推出	正式推出適用於 Apache 氣流的 Amazon 受管工作流程。 • 使用者指南文件 • AWS CloudFormation 文件	2020 年 11 月 24 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。