  
Picture  
  
  
  
  
  
  
MLOPS Pipeline Runbook

Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Version ID | Date | Author | Remark |
| 1 | 1.0 |  | Manoj P | First Version |
|  |  |  |  |  |

## Runbook: Create GitHub Connection using CloudFormation

### Goal

Set up a GitHub connection in AWS using a CloudFormation template, then complete the GitHub authorization process to make the connection usable in services like CodePipeline using a console.

### Steps

1. **Install AWS CLI**
   1. [Installation Guide](https://word-edit.officeapps.live.com/we/%22https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)
2. **Configure AWS CLI (One-Time Setup)/Launch using Cloud shell in AWS environment.**
   1. If you want to launch resources locally.
      1. Before deploying the CloudFormation stack, ensure your AWS CLI is configured with the necessary credentials and region.
      2. Type bash command given below.

aws configure

* + 1. You will be prompted to enter:
       1. **AWS Access Key ID**
       2. **AWS Secret Access Key**
       3. **Default region name** (e.g., us-west-2)
       4. **Default output format** (you can press Enter to leave it as None or type json)
    2. Make sure the credentials have sufficient permissions to deploy CloudFormation stacks and manage CodeStar Connections.
  1. If you want to launch resources from cloudshell in an AWS account, then directly run the command given below.

1. **Deploy the CloudFormation Stack**
   1. Run the following AWS CLI command to launch the stack:

aws cloudformation create-stack\  
 --template-url <https://mlops-pipeline-template-files.s3.us-east-1.amazonaws.com/create-github-connection.yaml> \  
 --stack-name github-connection-stack \  
 --parameter \  
 ParameterKey=GitHubConnectionName,ParameterValue=mlops-pipeline-github-connection \  
 --capabilities CAPABILITY\_NAMED\_IAM

* + 1. In template-url: add url of create-github-connection.yaml template which present in S3 bucket.

### Step 4-6: Launch resources using AWS Console

1. **Provide Parameter Value**
   1. Open [AWS CloudFormation Console](https://console.aws.amazon.com/cloudformation/).
   2. Upload the template you created (create-github-connection.yaml).
   3. When prompted, provide a value for stack name and GitHubConnectionName (e.g., mlops-pipeline-github-connection) and click **Next**.
2. **Configure Stack Options**
   1. Leave the default options or add tags if required.
   2. Click **Next**.
3. **Review and Submit**
   1. Review the stack configuration.
   2. Click **Submit** to launch the stack.
4. **Navigate to CodePipeline Settings**
   1. After the stack is created, go to the AWS Console.
   2. Open the **CodePipeline** service.
   3. Go to the **Settings** tab on the left.
5. **Access Connection Settings**
   1. In the settings panel, click on the **Connections** section.
6. **Check Connection State**
   1. You will see the newly created connection listed.
   2. Its status will be **Pending**.
7. **Update Pending Connection**
   1. Click on the connection name.
   2. Then click on the **Update pending connection** button.
8. **Install GitHub App**
   1. In the popup, select **Install a new app**.
9. **Authenticate with GitHub**
   1. A GitHub login window will open.
   2. Enter your **GitHub username and password**.
10. **Authorize Repository Access**
    1. Choose the **GitHub repository** you want to allow access to.
    2. Click **Save**.
11. **Connect and Finalize**
    1. Back in the AWS Console, click **Connect**.
    2. The connection will now move to **Available** state.

### Result

The GitHub connection is now successfully created and authorized. You can reference this connection ARN in your CodePipeline stages.

## Runbook: Create AWS Service Catalog Portfolio, AWS Service Catalog Product and Sagemaker custom project using single CloudFormation template.

### Steps

1. **Install AWS CLI**
   1. [Installation Guide](https://word-edit.officeapps.live.com/we/%22https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)
   2. Skip this step if you have already installed the AWS CLI.
2. **Configure AWS CLI (One-Time Setup)/Launch using Cloudshell in AWS environment.**
   1. If you want to launch resources locally.
      1. Before deploying the CloudFormation stack, ensure your AWS CLI is configured with the necessary credentials and region:
      2. Type bash command given below.

aws configure

* + 1. You will be prompted to enter:
       1. **AWS Access Key ID**
       2. **AWS Secret Access Key**
       3. **Default region name** (e.g., us-west-1)
       4. **Default output format** (you can press Enter to leave it as None or type json)
    2. Make sure the credentials have sufficient permissions to deploy CloudFormation stacks and manage CodeStar Connections.
  1. If you want to launch resources from cloudshell in an AWS account, then directly run the command given below.

1. **Deploy the CloudFormation Stack**
   1. Run the following AWS CLI command to launch the stack:

aws cloudformation create-stack \  
 --template-url <https://axcess-devst-sagemaker.s3.us-east-1.amazonaws.com/mlops-pipeline-resource-templates/Complete_Mlops_Package.yaml> \  
 --stack-name complete-mlops-resources-stack \  
 --parameters \  
 ParameterKey=PortfolioName,ParameterValue=mlops-service-catalog-portfolio \  
 ParameterKey=SageMakerExecutionRoleArn,ParameterValue=arn:aws:iam::345951:role/service-role/AmazonSageMaker-ExecutionRole-20250325T120134 \  
 ParameterKey=PrincipalArn,ParameterValue=arn:aws:iam::345951:role/aws-reserved/sso.amazonaws.com/AWSReservedSSO\_AWSAdministratorAccess\_834ce3a515a6f00d \  
 ParameterKey=ProductName,ParameterValue=mlops-service-catalog-product \  
 ParameterKey=GithubUrl,ParameterValue="https://github.com/<user-name>/<repo-name>" \  
 ParameterKey=ZipUrlGithubRepo,ParameterValue="https://github.com/<user-name>/<repo-name>/archive/refs/heads/<brach-name>.zip" \  
 ParameterKey=RepoBranchName,ParameterValue="<repo-name>-<branch-name>" \  
 ParameterKey=MainFolderName,ParameterValue="<main-folder-name-from-repo>" \  
 ParameterKey=ServiceCatalogPortfolioTemplateURL,ParameterValue=https://axcess-devst-sagemaker.s3.us-east-1.amazonaws.com/mlops-pipeline-resource-templates/create-service-catalog-portfolio.yaml \  
 ParameterKey=ServiceCatalogProductTemplateURL,ParameterValue=https://axcess-devst-sagemaker.s3.us-east-1.amazonaws.com/mlops-pipeline-resource-templates/create-product.yaml \  
 ParameterKey=SageMakerProjectName,ParameterValue="axcess-sagemaker-project-8" \  
 ParameterKey=GitHubOwner,ParameterValue="<github-user-name>" \  
 ParameterKey=GitHubRepo,ParameterValue="<github-repo-name>" \  
 ParameterKey=GitHubBranch,ParameterValue="<branch-name>" \  
 ParameterKey=GithubConnArn,ParameterValue="arn:aws:codeconnections:us-east-1:345951:connection/0cdb21f1-6062-433b-bdea-f4bde8ea6f07" \  
 ParameterKey=UseCase,ParameterValue="trip-prediction" \  
 ParameterKey=S3Prefix,ParameterValue="NA" \  
 ParameterKey=ProcessingInstanceType,ParameterValue="ml.t3.large" \  
 ParameterKey=TrainingInstanceType,ParameterValue="ml.m5.large" \  
 ParameterKey=ProcessingInstanceCount,ParameterValue="1" \  
 ParameterKey=TrainingInstanceCount,ParameterValue="1" \  
 ParameterKey=ModelPackageGroupName,ParameterValue="trip-prediction" \  
 ParameterKey=ModelVersion,ParameterValue="1" \  
 ParameterKey=DeployStackName,ParameterValue="trip-prediction-1-deploy-staging" \  
 ParameterKey=S3Bucket,ParameterValue="axcess-devst-sagemaker-bucket-1" \  
 --capabilities CAPABILITY\_NAMED\_IAM \  
 --region us-east-1

* + 1. In template-url: add url of Complete\_Mlops\_Package.yaml template which present in S3 bucket. (using this template will launch all resources)
    2. In SageMakerExecutionRoleArn: add arn of sagemaker domain execution role.
    3. In PrincipalArn: add arn of user or sso user role.
    4. In ZipUrlGithubRepo: add url of zipped repo.
    5. In RepoBranchName: add "RepoName"-"BrachName"
    6. In ServiceCatalogPortfolioTemplateURL: add url of create-service-catalog-portfolio.yaml template present in S3 bucket.
    7. In ServiceCatalogProductTemplateURL: add url of create-product.yaml template which present in S3 bucket.
    8. In GithubConnArn: add arn of github connection which created first step.

## Runbook: If you want Create an AWS Service Catalog Portfolio separately for SageMaker MLOps

This runbook helps you deploy a Service Catalog Portfolio using CloudFormation to support Amazon SageMaker MLOps workflows.

### Prerequisites

* AWS CLI installed and configured
* Required IAM permissions to create Service Catalog resources

### Step 1: Deploy CloudFormation Stack

* Run below command to create Service Catalog Portfolio

aws cloudformation create-stack \  
--template-url <https://mlops-pipeline-template-files.s3.us-east-1.amazonaws.com/create-service-catalog-portfolio.yaml> \  
--stack-name mlops-service-catalog-portfolio-stack \  
--parameters \  
 ParameterKey=AllowDirectUserAccessParameter,ParameterValue=false \  
 ParameterKey=MakePorfolioAccessibleFromSageMakerStudioParameter,ParameterValue=true \  
 ParameterKey=PortfolioDescriptionParameter,ParameterValue="Custom project templates for MLOps" \  
 ParameterKey=PortfolioNameParameter,ParameterValue=mlops\_pipeline\_service\_catalog\_portfolio \  
 ParameterKey=PortfolioOwnerParameter,ParameterValue=owner \ ParameterKey=SageMakerStudioExecutionRoleParameter,ParameterValue=arn:aws:iam::317046:role/service-role/AmazonSageMaker-ExecutionRole-20240228T142935 \  
 ParameterKey=UserAccessPrincipalsParameter,ParameterValue= \  
--capabilities CAPABILITY\_NAMED\_IAM \  
--region us-east-1

* In SageMakerStudioExecutionRoleParameter section add role of Sagemaker Domain.
* Keep UserAccessPrincipalsParameter section blank.
* In PortfolioNameParameter section, add the name for Service Catalog Portfolio.

### Step 2: Post Deployment

After successful deployment:

1. Navigate to **AWS Console > Service Catalog > Portfolios**
2. You will find a new portfolio named mlops\_pipeline\_service\_catalog\_portfolio
3. Confirm:
   1. The execution role is associated
   2. In tag section sagemaker:studio-visibility should be true

### Step 3: Grant Access

1. To create custom projects using a Service Catalog product in SageMaker Studio, you need to add Group,Users,Roles as principals in the portfolio access section.
   1. AWS Console > Service Catalog > Portfolio > Access > Grant Access

## Runbook: If you want Create Service Catalog Product separately for Multi-model Training and Batch Inference Pipeline

### Overview

This CloudFormation template deploys a **Service Catalog product** that can be used from within **Amazon SageMaker Studio** to launch a custom project for MLOps pipelines.

### Prerequisites

* An existing **Service Catalog Portfolio**
* Portfolio ID (from CloudFormation stack output if you used a template to create it)
* AWS CLI configured (or use the AWS Console)

### Deployment Using AWS Console

1. **Open AWS CloudFormation Console**
   1. [CloudFormation Console](https://console.aws.amazon.com/cloudformation)
2. **Create Stack**
   1. Click **"Create stack" → "With new resources (standard)"**
3. **Specify Template**
   1. Choose your template file (.yaml or .json)
   2. If the template is in S3, select **Amazon S3 URL** and paste the full path (e.g., <https://s3.amazonaws.com/your-bucket/template.yaml>)
4. **Provide Parameters**
   1. Fill out values as per the parameter.
5. **Stack Options**
   1. Leave default or add tags if needed
6. **Permissions**
   1. Ensure the deploying IAM user/role has access to servicecatalog:\* and cloudformation:\*
7. **Review and Submit**
   1. Click **Create stack**

### AWS CLI Deployment

* In PortfolioIDParameter add Service Catalog Portfolio ID in which we have to create Service Catalog Product.
* In ProductNameParameter add name for Product.
* In ProductSupportURLParameter add url of Github Repo where custom project template is present.
* In SageMakerProjectRepoZipParameter url should like ProductSupportURLParameter/archive/refs/heads/main.zip
* In SageMakerProjectRepoNameBranchParameter add Name/Branch of the SageMaker Projects Examples GitHub Repo. (should like RepoName-Branch)
* In SageMakerProjectsProjectNameParameter add Project folder inside of the GitHub repo for this project.

aws cloudformation create-stack \  
--template-url <https://mlops-pipeline-template-files.s3.us-east-1.amazonaws.com/create-product.yaml> \  
--stack-name mlops-service-catalog-product-stack \  
--parameters \  
 ParameterKey=PortfolioIDParameter,ParameterValue=port-474rj4x4tzpw2 \  
 ParameterKey=ProductNameParameter,ParameterValue=mlops-pipeline-service-catalog-product \  
 ParameterKey=ProductDescriptionParameter,ParameterValue="A pipeline project for training and batch inference" \  
 ParameterKey=ProductOwnerParameter,ParameterValue="Product Owner" \  
 ParameterKey=ProductDistributorParameter,ParameterValue="Product Distributor" \  
 ParameterKey=ProductSupportDescriptionParameter,ParameterValue="Support Description" \  
 [ParameterKey=ProductSupportEmailParameter,ParameterValue="support@example.com](mailto:ParameterKey=ProductSupportEmailParameter,ParameterValue="support@example.com)" \  
ParameterKey=ProductSupportURLParameter,ParameterValue="https://github.com/<user-name>/<repo-name>" \  
ParameterKey=SageMakerProjectRepoZipParameter,ParameterValue="https://github.com/mnpat/axcess-test/archive/refs/heads/main.zip" \  
 ParameterKey=SageMakerProjectRepoNameBranchParameter,ParameterValue="<repo-name>-<branch-name>" \  
 ParameterKey=SageMakerProjectsProjectNameParameter,ParameterValue="<main-folder-name-from-repo>" \  
--capabilities CAPABILITY\_NAMED\_IAM \  
--region us-east-1

### Validation Steps

1. Go to **Service Catalog > Products**
2. We should see a new product with the name provided.
3. Open **Amazon SageMaker Studio**
4. Go to **Projects > Create project**
5. We should see the custom project listed in the Organization Template section.

## Runbook: If you want to create SageMaker ML Project separately.

### Overview

This CloudFormation template provisions all required infrastructure for managing a CI/CD pipeline in **Amazon SageMaker**, targeting batch inference jobs. It uses GitHub as the source, SageMaker Pipelines for training, and Model Registry for model versioning and deployment.

### Pre-requisites

Ensure the following are in place before deployment:

* **GitHub connection** is created via *AWS CodeStar Connections*.
* AWS CLI configured:
* IAM role has permissions to deploy CloudFormation stacks and access SageMaker, S3, and CodePipeline.

### Template Description

This template:

* Sets up CodePipeline for CI/CD
* Uses SageMaker Pipelines for training
* Registers models in Model Registry
* Automates deployment to **staging** environments for **batch inference**

### Deploy Step (AWS CLI)

1. **Run AWS CLI Command**

* In the project-name add name of sagemaker project which want to create.
* In ProductId add id of product which created in above steps.
* In ProvisioningParameters add required values for given keys.

aws sagemaker create-project --project-name "sm-project" --service-catalog-provisioning-details '{  
 "ProductId": "prod-rcp2tgmmj32ck",  
 "ProvisioningParameters": [  
   
 {  
 "Key": "GitHubOwner",  
 "Value": "<github-user-name>"  
 },  
 {  
 "Key": "GitHubRepo",  
 "Value": "<github-remo-name>"  
 },  
 {  
 "Key": "GitHubBranch",  
 "Value": "<github-branch-name>"  
 },  
 {  
 "Key": "GithubConnArn",  
 "Value": "<github-connection-arn-created\_in\_first\_step>"  
 },  
 {  
 "Key": "UseCase",  
 "Value": "<usecase-name-which\_is\_folder\_in\_repo\_includes\_all\_codes\_like\_processing\_training\_and\_other>"  
 },  
 {  
 "Key": "s3Prefix",  
 "Value": "NA"  
 },  
 {  
 "Key": "ProcessingInstanceType",  
 "Value": "<Processing-Instance-Type>"  
 },  
 {  
 "Key": "TrainingInstanceType",  
 "Value": "<Training-Instance-Type>"  
 },  
 {  
 "Key": "ProcessingInstanceCount",  
 "Value": "<Processing-Instance-Count>"  
 },  
 {  
 "Key": "TrainingInstanceCount",  
 "Value": "<Training-Instance-Count>"  
 },  
 {  
 "Key": "ModelPackageGroupnameInput",  
 "Value": "<Model-Package-Groupname-Input>"  
 },  
 {  
 "Key": "ModelVersionInput",  
 "Value": "<Model-Version-Input>"  
 },  
 {  
 "Key": "StackName",  
 "Value": "<stack-name-while\_deploying\_model\_at\_staging>"  
 },  
 {  
 "Key": "mls3bucket",  
 "Value": "<name-of-s3-bucket-using\_for\_mlops\_pipeline>"  
 }  
 ]  
 }'

### Validation Steps

After stack creation:

* Go to **AWS CodePipeline** → Validate the pipeline stages
* Confirm **Source** connects to GitHub
* Go to **AWS Sagemaker > Domain > User Profile > Sagemaker Studio > Deployments > Project** where launched custom prject has created.

## Runbook: Auto-Stopping Initial CodePipeline Execution (Lambda + EventBridge Rule) launched by CreatePipeline API.

### Objective

* In codepipeline when we create a pipeline using cli or console it starts pipeline execution automatically.
* Prevent the **automatic execution** of an AWS **CodePipeline** triggered immediately after stack creation by CreatePipeline API.

### Pre-requisites

Ensure:

* **IAM permissions** to create Lambda functions, IAM roles, EventBridge rules, and manage CodePipeline.
* The **CloudFormation template** is available locally or accessible via S3.
* The Lambda function code either resides in the template or in a referenced S3 location.
* AWS CLI configured with access to the appropriate AWS account.

### Deployment Steps

#### Option 1: Using AWS CLI

1. Deploy the stack with AWS CLI:

aws cloudformation deploy \  
 --template-url <https://axcess-devst-sagemaker.s3.us-east-1.amazonaws.com/mlops-pipeline-resource-templates/Stop-codepipeline-pipeline-execution.yaml> \  
 --stack-name stop-initial-pipeline-execution \  
 --capabilities CAPABILITY\_NAMED\_IAM

#### Option 2: Using AWS Console

1. Go to **CloudFormation Console** > “Create Stack”.
2. Upload the template or provide S3 URL.
3. Acknowledge IAM resource creation and deploy.

### Validation Checklist

After deployment:

* Confirm the **Lambda function** is created and named accordingly (e.g., StopInitialPipelineExecution).
* Verify that the **EventBridge rule** is configured to trigger the Lambda CodePipeline Execution State Change for **STARTED** state.
* Confirm the **pipeline was stopped automatically** after stack creation.

## Runbook: Deploying Lambda + EventBridge Rule for cross account model deployment (Production).

### Overview

This CloudFormation template provisions:

* A **Lambda function** is designed to handle specific production tasks.
* An **Amazon EventBridge rule** that triggers the Lambda function based on predefined events.

This is designed for deployment in the **Production account** as part of the overall MLOps toolchain.

### Pre-requisites

Ensure the following before deployment:

* You are operating in the **Production AWS account**.
* You have **deployment permissions** for AWS Lambda, IAM, and EventBridge.
* AWS CLI or CloudFormation console access is available.
* IAM execution role for Lambda has necessary permissions.

### Deployment Steps

#### **Using AWS CLI**

1. **Execute the CloudFormation deployment**:

aws cloudformation deploy \  
 --template-url <https://axcess-devst-sagemaker.s3.us-east-1.amazonaws.com/mlops-pipeline-resource-templates/Production-account-setup.yaml> \  
 --stack-name mlops-pipeline-prod-account-setup \  
 --parameter-overrides \  
 SageMakerProjectName=mlops-project \  
 --capabilities CAPABILITY\_NAMED\_IAM

### Validation Checklist

After deployment:

* Go to **AWS Lambda Console** and confirm the Lambda function is created.
* Validate the **IAM Role** associated with the Lambda (includes logs, event access, etc.).
* Navigate to **Amazon EventBridge > Rules**, and verify the rule exists and is targeting the Lambda.
* Check the **event pattern** to confirm it matches your production trigger (e.g., SageMaker job status).

### Create one role for sagemaker tasks in production account

* Role name: AmazonSageMakerServiceCatalogProductsUseRoleMultiModelTB
* Policies:
  + AWS managed:
    - AmazonS3FullAccess
    - AmazonSageMakerFullAccess
  + Customer managed:
    - access\_ecr\_stage\_policy {  
       "Version": "2012-10-17",  
       "Statement": [  
       {  
       "Effect": "Allow",  
       "Action": [  
       "ecr:GetAuthorizationToken"  
       ],  
       "Resource": "\*"  
       },  
       {  
       "Effect": "Allow",  
       "Action": [  
       "ecr:BatchGetImage",  
       "ecr:GetDownloadUrlForLayer",  
       "ecr:BatchCheckLayerAvailability"  
       ],  
       "Resource": "arn:aws:ecr:us-east-1:345594592951:repository/sagemaker-inference"  
       }  
       ]  
       }
    - Below is trust relationship: {  
       "Version": "2012-10-17",  
       "Statement": [  
       {  
       "Sid": "",  
       "Effect": "Allow",  
       "Principal": {  
       "Service": "sagemaker.amazonaws.com"  
       },  
       "Action": "sts:AssumeRole"  
       }  
       ]  
       }

## In staging and prod add select create role for eventbridge event while cross account setup.

## Runbook: Permissions for cross account sagemaker model deployment.

* [User Guide](https://word-edit.officeapps.live.com/we/%22https://axcesstechsystems.sharepoint.com/:b:/s/MLDataTransformation/EaPk7pphVcNFvKMywztPjG4B3l_AzA5MZh4Xw-cepb0y0w?e=SrWJbT%22)