



Amazon SageMaker Pipelines

최영준
AI/ML Expert SA
AWS

Overview

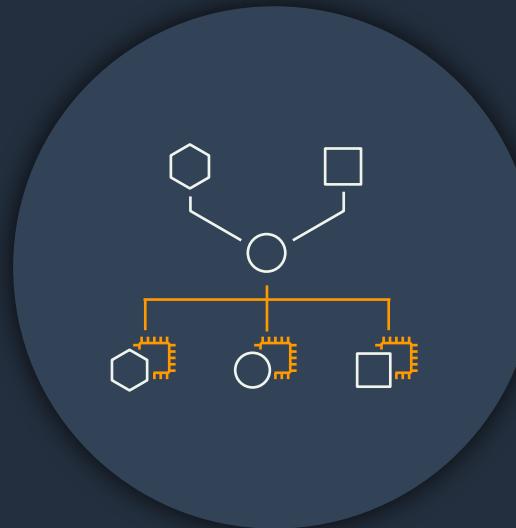
SageMaker Pipelines



NEW

Amazon SageMaker Pipelines

Managed machine learning CI/CD service



워크플로우의
각 단계를 중앙에서 관리



워크플로우 공유 및
재실행



built-in 템플릿에서
선택

Amazon SageMaker Pipelines

규모에 맞게 완전히
자동화된 머신러닝
워크플로우 구축



ML 워크플로우를 빠르고 쉽게 구축할 수 있는 Python SDK

실행을 위한 매개변수, 파이프라인 단계(데이터 처리, 교육, 튜닝, 배치 추론, 모델 등록) 및 파이프라인 DAG를 정의합니다. 파이프라인 단계는 데이터를 인식하며 한 단계의 출력이 다음 단계의 입력이 될 수 있습니다.

SageMaker 관리되는 workflow 실행

파이프라인 실행, 관련 메트릭 및 로그는 파이프라인을 쉽게 디버깅할 수 있도록 SageMaker Studio에서 실시간으로 유지되고 볼 수 있습니다. 재사용 및 문제 해결을 위해 파이프라인 실행을 캐시합니다.

Model Registry의 모델 버전 카탈로그

훈련된 모델을 중앙 모델 레지스트리에 등록합니다. 기본 제공 모델 버전 관리 및 교차 계정 지원. 모델 메트릭을 보고 모델 배포를 승인합니다.

CI/CD에 대해 built-in 지원을 통한 Fully managed MLOps

모델 구축 및 구축을 위한 built-in MLOps 템플릿은 AWS Code* 서비스를 사용하여 ML 워크플로우를 자동화합니다. DevOps는 AWS 서비스 카탈로그에 게시된 사용자 지정 템플릿을 가져올 수 있습니다. Model Monitor를 사용하여 예약된 이벤트, 코드 및 데이터 변경, 드리프트 감지를 기반으로 ML 파이프라인을 자동화합니다.

governance and audits를 위한 End to end Lineage Tracking

데이터, 코드, 메트릭 및 모델 아티팩트에 대한 lineage 추적을 위한 built-in 지원합니다.



Amazon SageMaker Pipelines

Amazon SageMaker
서비스 통합을 통한
워크플로우 구축



Amazon SageMaker Studio Integration

Amazon SageMaker Studio에서 직접 워크플로우 구축 및 시각화합니다.

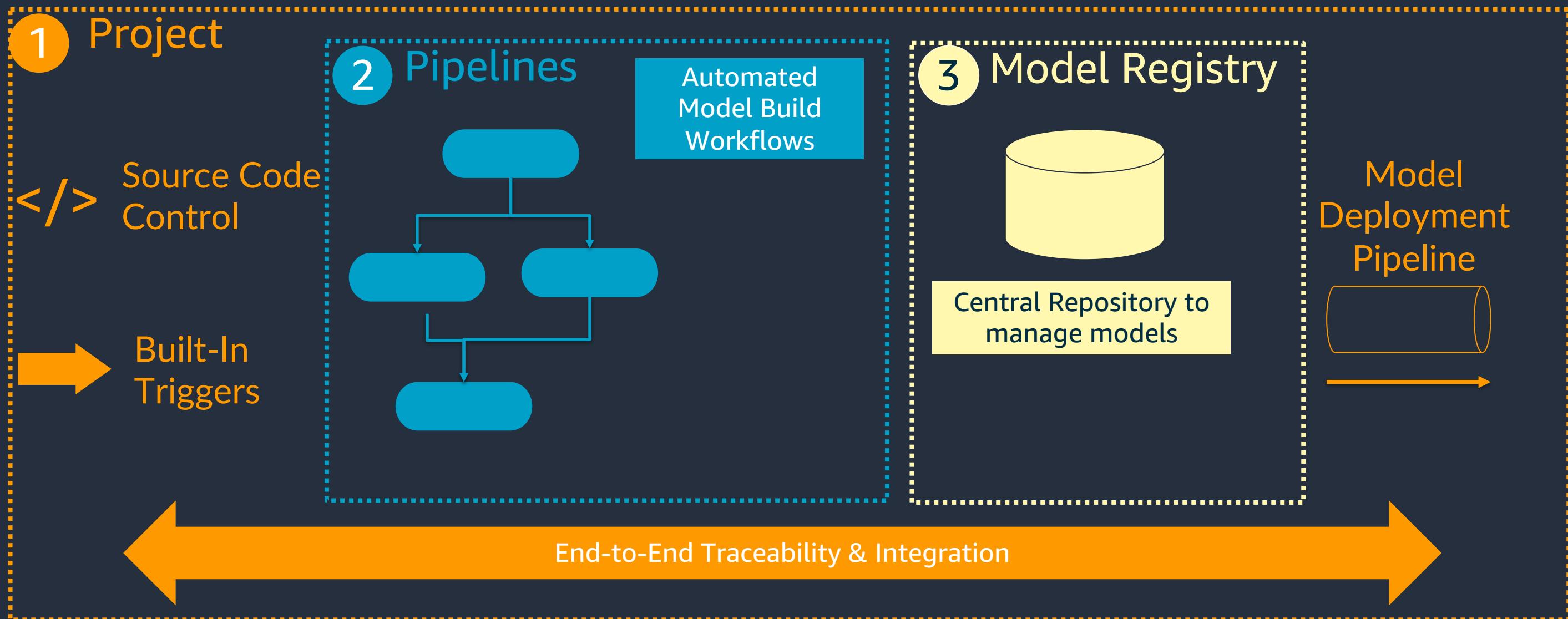


Amazon SageMaker Feature Integration

SageMaker Training 및 SageMaker Processing과 같은 서비스와 built-in 통합 기능을 통해 다른 SageMaker 기능을 활용합니다.

Amazon SageMaker Pipelines

Components

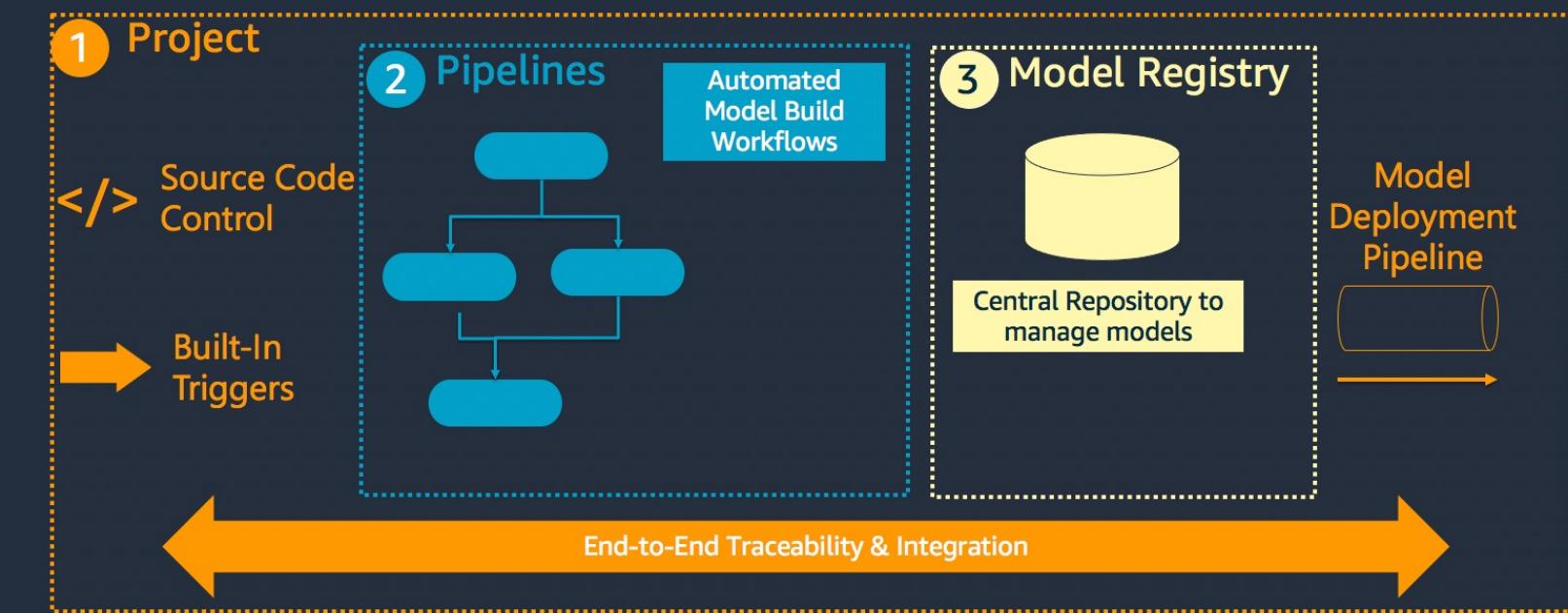


Amazon SageMaker Pipelines

Components

- 유연성 (Flexible):

- ✓ CI/CD 모범 사례를 통합한 엔드 투 엔드 파이프라인에 대한 **Projects** 사용
- ✓ 선택적으로, 요구사항을 충족하기 위한 **Projects** 없이 **Pipelines** 또는 **Model Registry** 사용도 가능

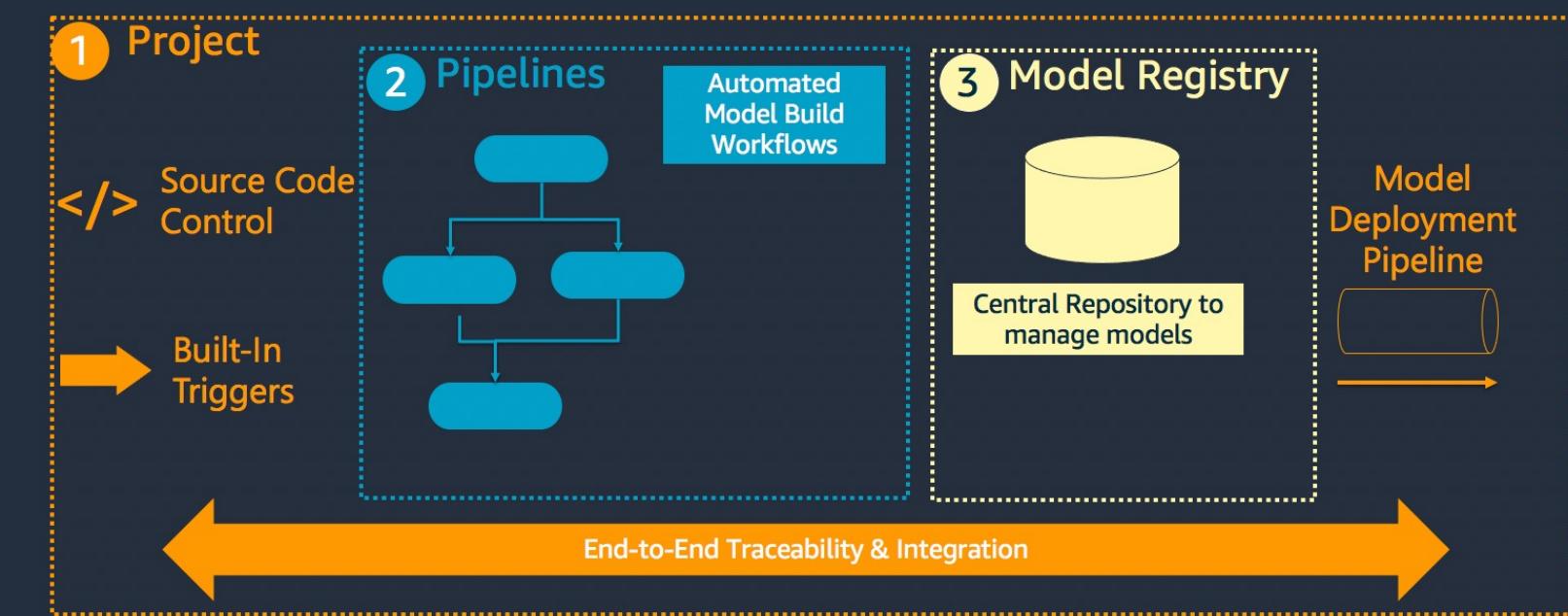


Amazon SageMaker Pipelines

Components

- 확장성 (Extensible):

- ✓ built-in MLOps Project Templates 활용
~OR~
- ✓ 회사 또는 규제 요구사항을 충족하기 위한 custom MLOps Project Templates 생성



Amazon SageMaker Pipelines

Cross Account Deployment

custom MLOps Project Templates을 사용하여 여러 계정에 배포할 수 있는 기능



Reference: <https://aws.amazon.com/blogs/machine-learning/multi-account-model-deployment-with-amazon-sagemaker-pipelines/>

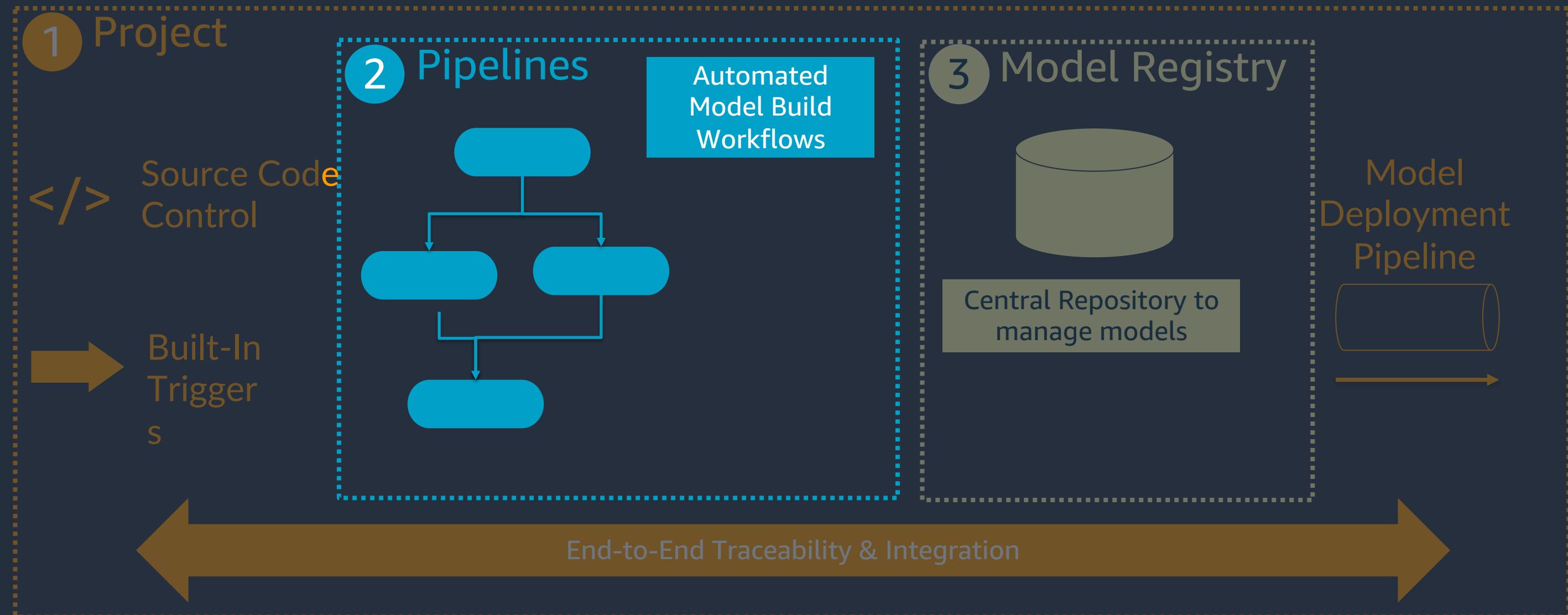
Pipelines Deep Dive

SageMaker Pipelines - Pipelines



Amazon SageMaker Pipelines

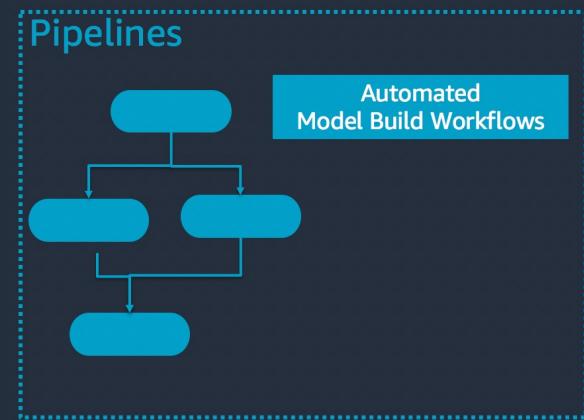
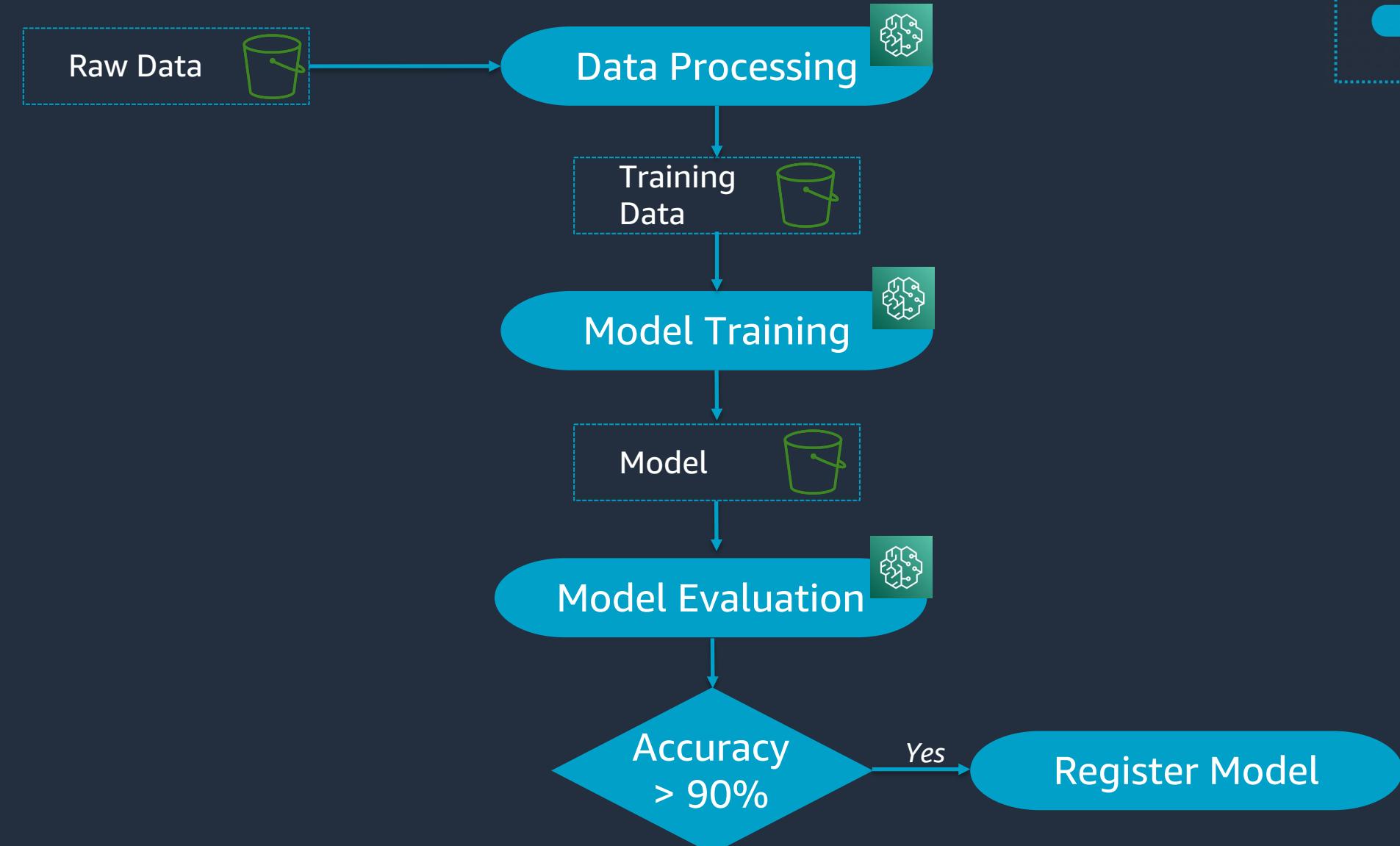
Components – Pipelines



Amazon SageMaker Pipelines

Model Build Workflow

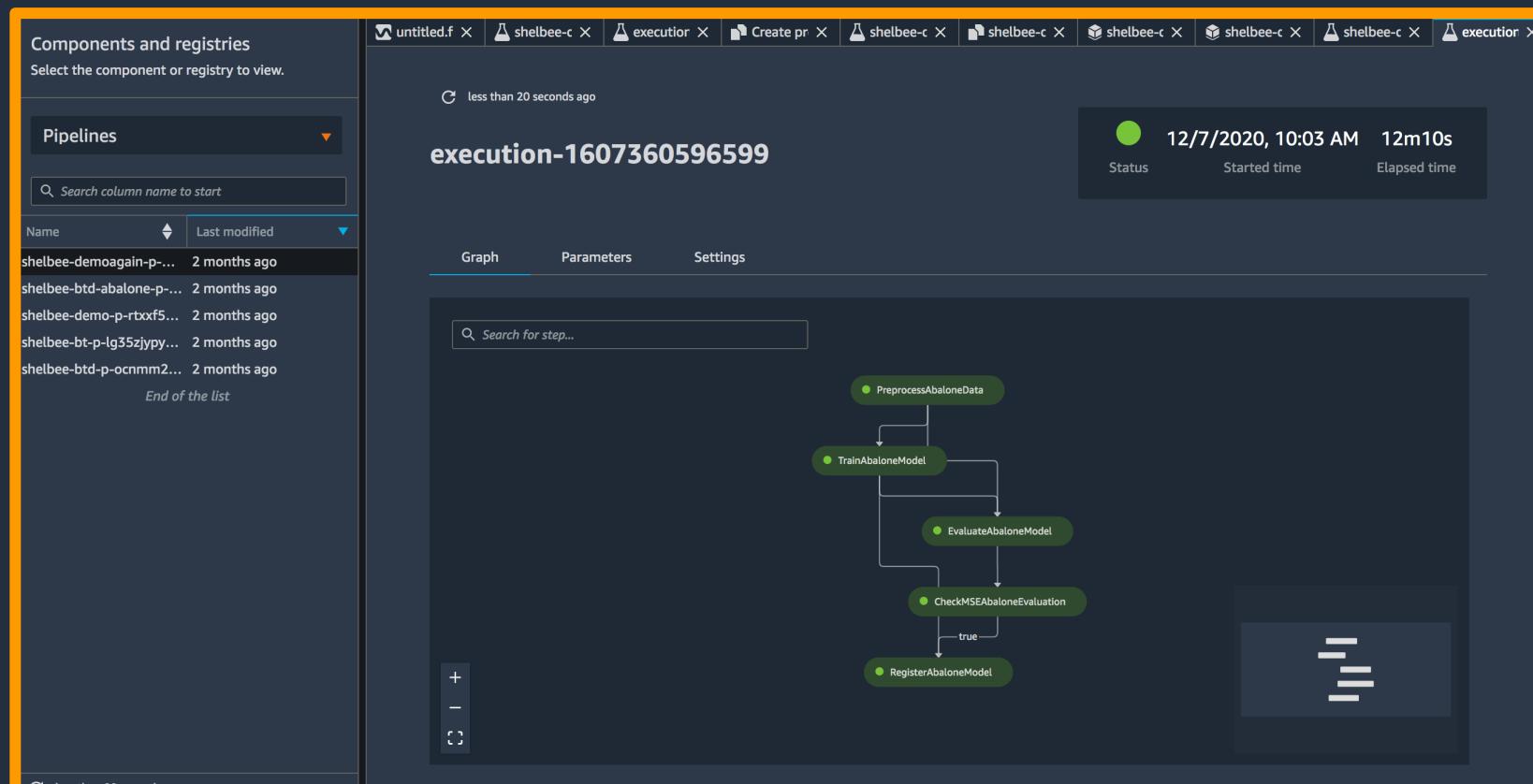
Example Workflow



Amazon SageMaker Pipelines

Pipelines

Supported Steps:



- **Processing:**

Data Processing & Model Evaluation

- **Training:**

Model Training using SageMaker Training Jobs

- **Conditional:**

Conditional execution of branches

- **BatchTransform:**

Batch predictions

- **RegisterModel:**

Create a model package resource

- **CreateModel:**

Package the model for deployment

Amazon SageMaker Pipelines

Supported Steps:

- **Processing** : Data Processing & Model Evaluation
- **Training**: Model Training using SageMaker Training Jobs
- **BatchTransform**: Batch predictions
- **Tuning** : create a hyperparameter tuning job
- **AutoML** : create an AutoML job to automatically train a model
- **Model** : create or register a SageMaker model
- **CreateModel** : Package the model for deployment
- **RegisterModel** : Create a model package resource
- **Transform** : Batch predictions
- **Condition** : Conditional execution of branches
- **Callback** : incorporate additional processes and AWS services
- **Lambda** : run an AWS Lambda function
- **ClarifyCheck** : conduct baseline drift checks against previous baselines for bias analysis and model explainability
- **QualityCheck** : conduct baseline suggestions and drift checks
- **EMR** : process Amazon EMR
- **Fail** : stop an Amazon SageMaker Model Building Pipelines execution

Creating Amazon SageMaker Automated Pipelines

Pipeline-as-Code

How it Works...

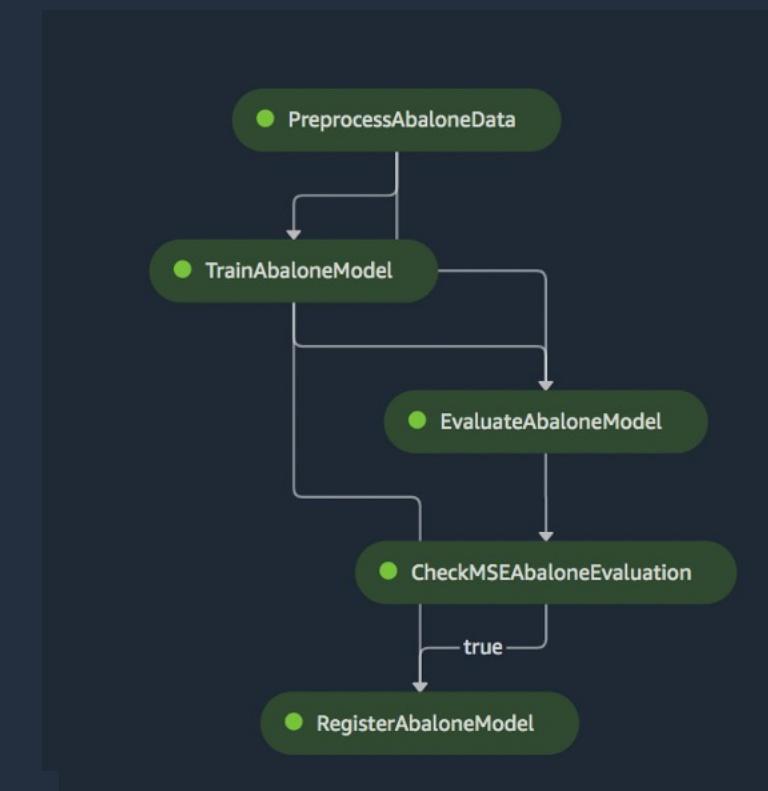
1 Create Steps →

코드를 통해 각 단계 정의 및 구성



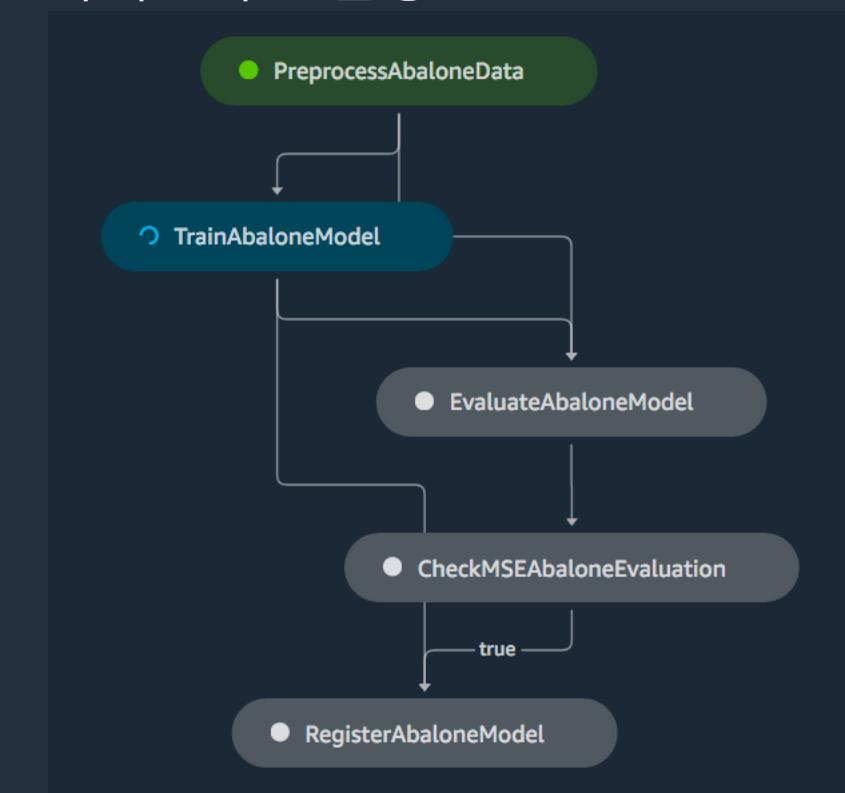
2 Define Pipeline →

워크플로우 정의 및 구성



3 Start Pipeline →

파이프라인 실행



Creating Amazon SageMaker Automated Pipelines

Pipeline-as-Code

How it Works...

1 Create Steps →

코드를 통해 각 단계 정의 및 구성

↳ PreprocessAbaloneData

← Step 1

● TrainAbaloneModel

← Step 2

● EvaluateAbaloneModel

← Step 3

● CheckMSEAbaloneEvaluation

← Step 4

● RegisterAbaloneModel

← Step 5

2 Define Pipeline →

워크플로우 정의 및 구성

↳ PreprocessAbaloneData

● TrainAbaloneModel

● EvaluateAbaloneModel

● CheckMSEAbaloneEvaluation

● RegisterAbaloneModel

↳ PreprocessAbaloneData

● TrainAbaloneModel

● EvaluateAbaloneModel

● CheckMSEAbaloneEvaluation

● RegisterAbaloneModel

3 Start Pipeline →

파이프라인 실행

● PreprocessAbaloneData

● TrainAbaloneModel

● EvaluateAbaloneModel

● CheckMSEAbaloneEvaluation

● RegisterAbaloneModel

Creating Amazon SageMaker Automated Pipelines

Pipelines

1 Create Steps →

```
from sagemaker.inputs import TrainingInput
from sagemaker.workflow.steps import TrainingStep

step_train = TrainingStep(
    name="TrainAbaloneModel",
    estimator=xgb_train,
    inputs={
        "train": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "train"
            ].S3Output.S3Uri,
            content_type="text/csv"
        ),
        "validation": TrainingInput(
            s3_data=step_process.properties.ProcessingOutputConfig.Outputs[
                "validation"
            ].S3Output.S3Uri,
            content_type="text/csv"
        )
    }
)
```

Example:
Training Step

Creating Amazon SageMaker Automated Pipelines

Pipelines

2 Define Pipeline →

```
from sagemaker.workflow.pipeline import Pipeline

pipeline_name = f"AbalonePipeline"
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_type,
        processing_instance_count,
        training_instance_type,
        model_approval_status,
        input_data,
        batch_data,
    ],
    steps=[step_process, step_train, step_eval, step_cond],
)

#Submit the pipeline definition
pipeline.upsert(role_arn=role)
```

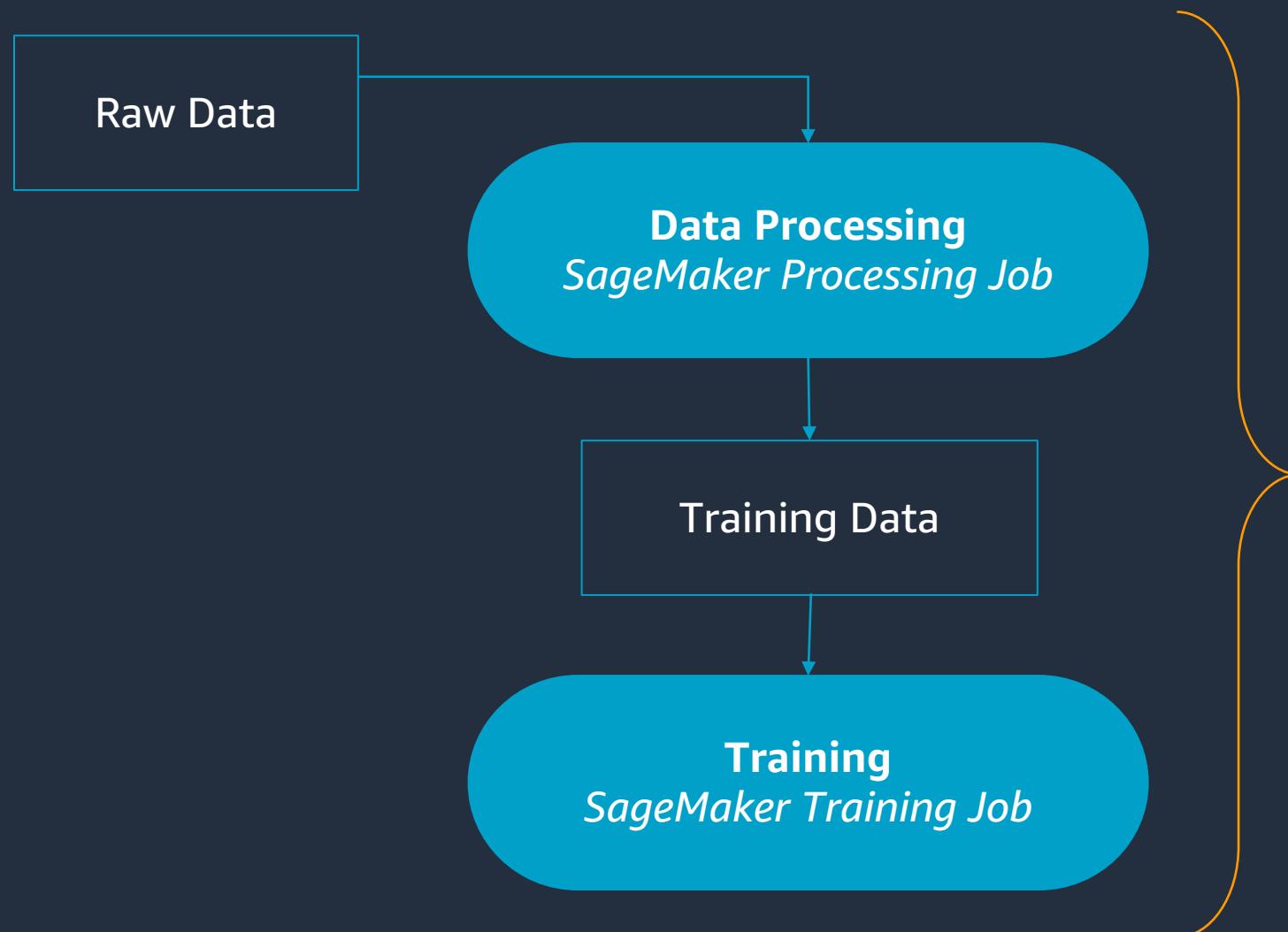
3 Start Pipeline →

```
execution = pipeline.start()
```

Amazon SageMaker Pipelines

Pipelines

Support for Step Caching →



다음을 수행하려면 어떻게 해야 하나요?:

- hyperparameters을 수정하시나요?
- training code 수정하나요?

Amazon SageMaker Pipelines

Pipelines

Support for Pipeline Parameters →

1 매개 변수 구성 →

```
from sagemaker.workflow.parameters import (
    ParameterInteger,
    ParameterString,
    ParameterFloat
)

processing_instance_count = ParameterInteger(
    name="ProcessingInstanceCount",
    default_value=1
)
```

2 파이프라인 생성 시 parameter 전달 →

```
pipeline = Pipeline(
    name=pipeline_name,
    parameters=[
        processing_instance_count
    ],
    steps=[step_process]
)
```

3 선택적으로, pipeline 실행을 위한 non-default 값 전달 →

```
execution = pipeline.start(
    parameters=dict(
        ProcessingInstanceType="ml.c5.xlarge",
        ModelApprovalStatus="Approved"
    )
)
```

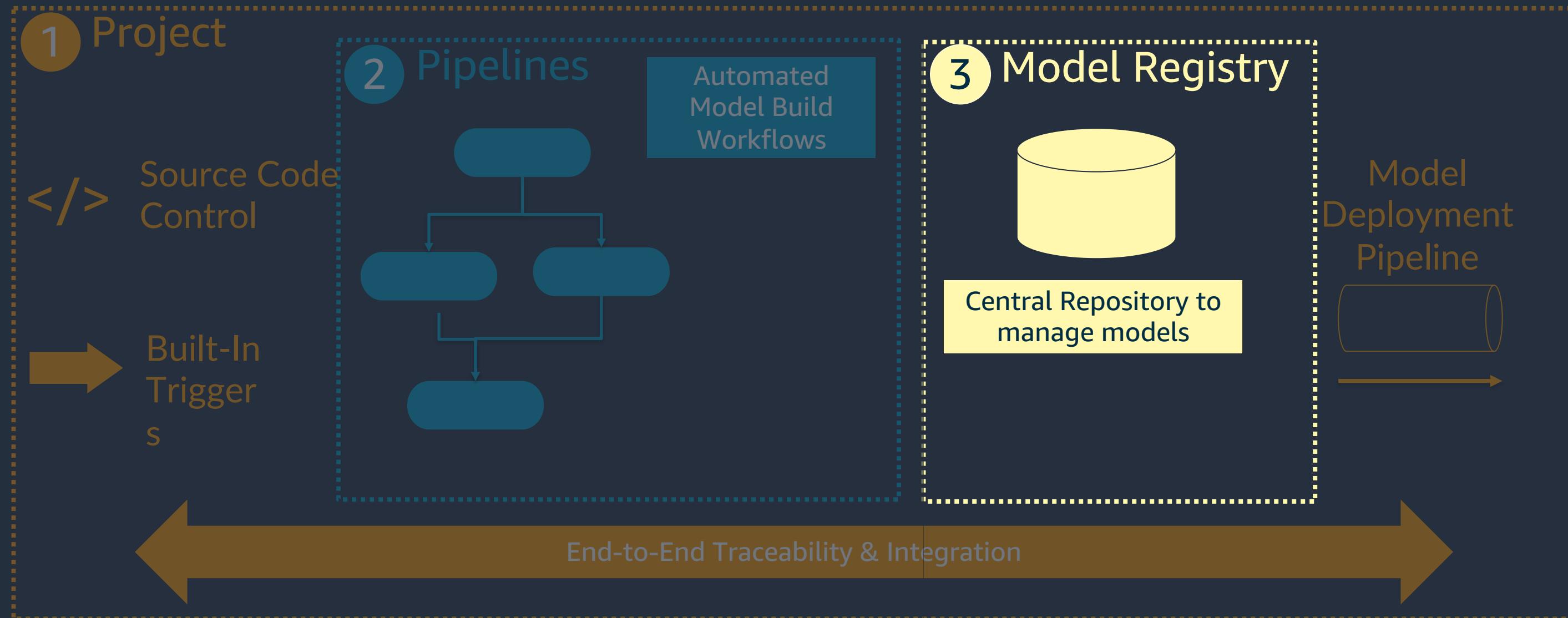
Model Registry Deep Dive

SageMaker Pipelines – Model Registry



Amazon SageMaker Pipelines

Components – Model Registry



Amazon SageMaker Pipelines

Model Registry

The screenshot shows the Amazon SageMaker Model Registry interface. At the top, there are tabs for 'Versions' and 'Settings'. Below is a search bar with placeholder text 'Search column name to start'. A table lists three model versions:

| Version | Stage | Status | Short description | Modified by | Last modified |
|---------|---------|----------|-------------------|----------------|---------------|
| 3 | None | Pending | | workshop-u... | |
| 2 | prod | Approved | | shelbee-igg... | |
| 1 | staging | Approved | | shelbee-igg... | |

A modal window for 'Version 2' is open, showing details like Status (Approved), Pipeline (shelbee-demoagain-p...), Execution workflow-2, Last Stage (prod), Model group (shelbee-demoagain-p...), and an 'Update status' button. The modal also includes tabs for Activity, Metrics, and Settings, and a section for Model metrics.

- Catalog models for production
- Manage model versions
- Associate metadata with a model
- Manage the approval status of a model
- Deploy models to production (with Projects)

- Track model performance metrics

Amazon SageMaker Pipelines

Model Registry

Amazon SageMaker Studio File Edit View Run

Components and registries
Select the component or registry to view.

| Model registry

MODEL REGISTRY
1 row selected 0/20 filters
Search column name to start

| Name | Created |
|------------------------------|--------------|
| shelbee-deploy-nb | 1 month ago |
| c-deploy-model | 1 month ago |
| deploy-model | 1 month ago |
| shelbee-demoagain-p-pisa... | 3 months ago |
| shelbee-btd-abalone-p-4kt... | 3 months ago |
| shelbee-demo-p-rtxxf5ma... | 3 months ago |
| shelbee-bt-p-lg35zjypy672 | 3 months ago |
| shelbee-btd-p-ocnmm22e... | 3 months ago |

End of the list

Hierarchy →

Model Registry

Model Group
ML Problem:
Abalone Age

Model Group
ML Problem:
Customer Churn

Model Group
ML Problem:
(n)

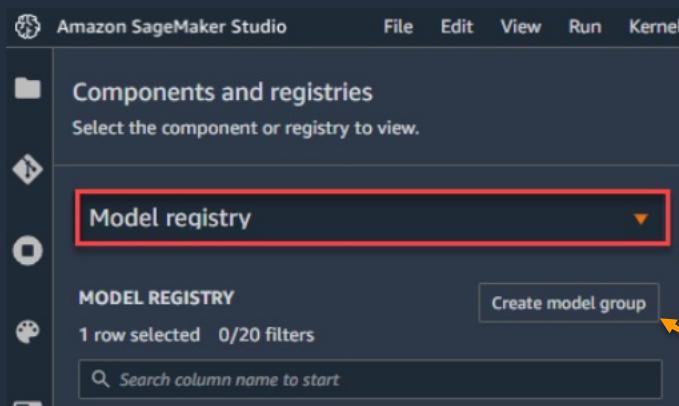


Amazon SageMaker Pipelines

Model Registry

How it Works...

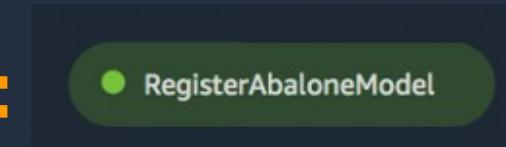
1 Create Model Group →



2 Register a Model Version →

Within a Pipeline, using RegisterModel Step:

```
step_register = RegisterModel(  
    name="RegisterAbaloneModel",  
    estimator=xgb_train,  
    model_data=step_train.properties.ModelArtifacts.S3ModelArtifacts,  
    content_types=["text/csv"],  
    response_types=["text/csv"],  
    inference_instances=["ml.t2.medium", "ml.m5.large"],  
    transform_instances=["ml.m5.large"],  
    model_package_group_name=model_package_group_name,  
    approval_status=model_approval_status,  
    model_metrics=model_metrics,  
)
```



~OR~

Using boto3:

```
create_model_package_response = sm_client.create_model_package(**create_model_package_input_dict)  
model_package_arn = create_model_package_response["ModelPackageArn"]  
print('ModelPackage Version ARN : {}'.format(model_package_arn))
```

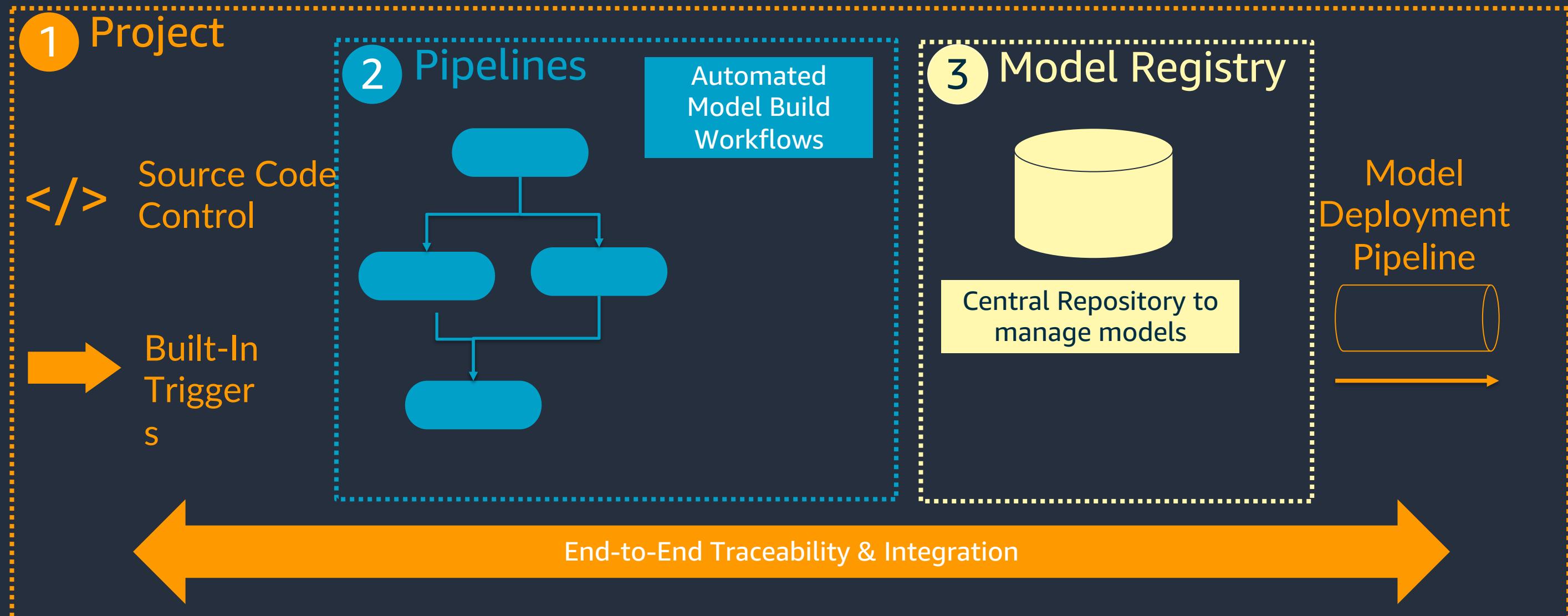
Projects Deep Dive

SageMaker Pipelines - Projects



Amazon SageMaker Pipelines

Components – Projects



Amazon SageMaker Pipelines - Projects

Getting Started

One-Time Setup

1

Studio 도메인에 대한 Projects 사용

Get started

[Learn more about getting started with SageMaker Studio](#)

Quick start
Let Amazon SageMaker handle configuring account and setting the permissions that you or a team in your organization need to use SageMaker Studio. Choosing this option uses standard encryption, which you can't change. If you need more control over configuration, choose Standard setup.

User name

The user name can have up to 63 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Execution role
SageMaker Studio requires permissions to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess policy](#) attached. If you don't have a role with this policy attached, we can create one for you.
[Choose an IAM role](#)

SageMaker Projects and JumpStart [New](#)
Enable access and provisioning of AWS Service Catalog Portfolio of products in Amazon SageMaker Studio for Amazon SageMaker Projects and JumpStart. [Learn more](#)

Enable Amazon SageMaker project templates and JumpStart for this account and Studio users
If enabled, the administrator can view the Amazon SageMaker provided project templates and JumpStart solutions published in AWS Service Catalog and users who are configured to use the domain execution are allowed to create projects using those templates and solutions with JumpStart. A launch constraint role and a project use role are automatically generated in IAM for your account.

새 도메인에서
사용하도록 설정하거나
기존 도메인을
업데이트

Amazon SageMaker Pipelines - Projects

Features



- Built-In MLOps Project Templates 활용:
 1. Build, Train, Deploy
 2. Build, Train
 3. Deploy
- Custom MLOps Project Templates 생성

Amazon SageMaker Pipelines - Projects

Features



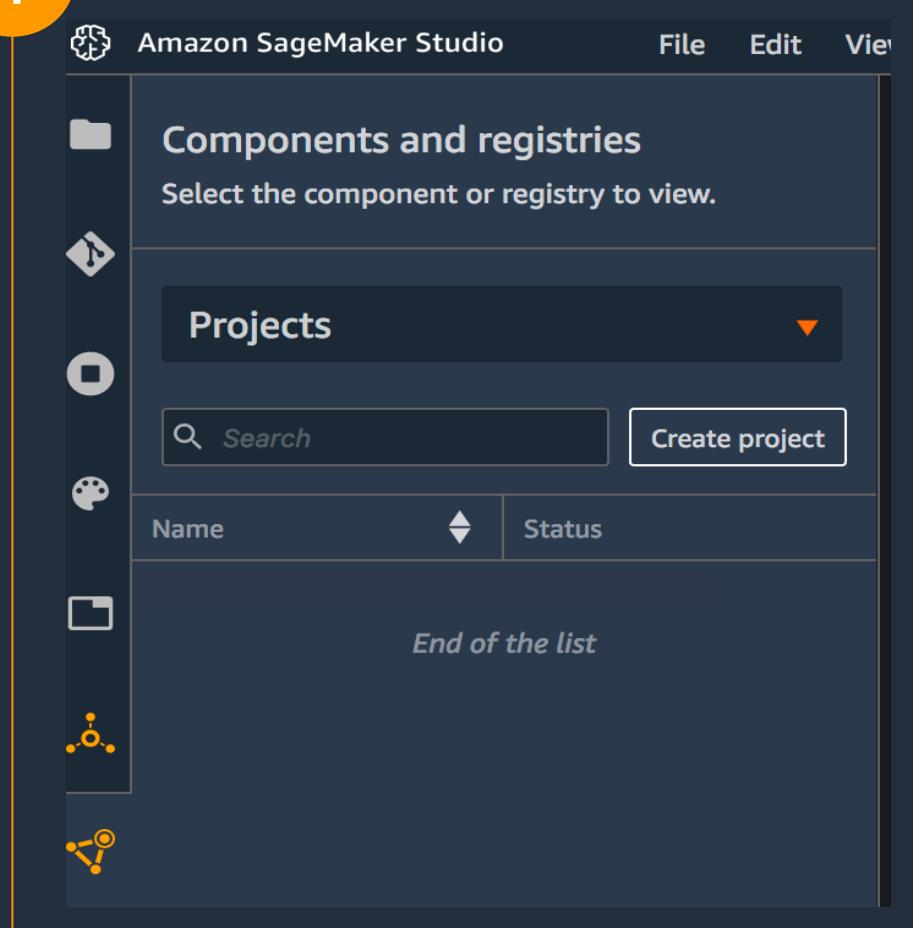
- Built-In MLOps Project Templates 활용:
 1. Build, Train, Deploy
 2. Build, Train
 3. Deploy
- Custom MLOps Project Templates 생성

Amazon SageMaker Pipelines - Projects

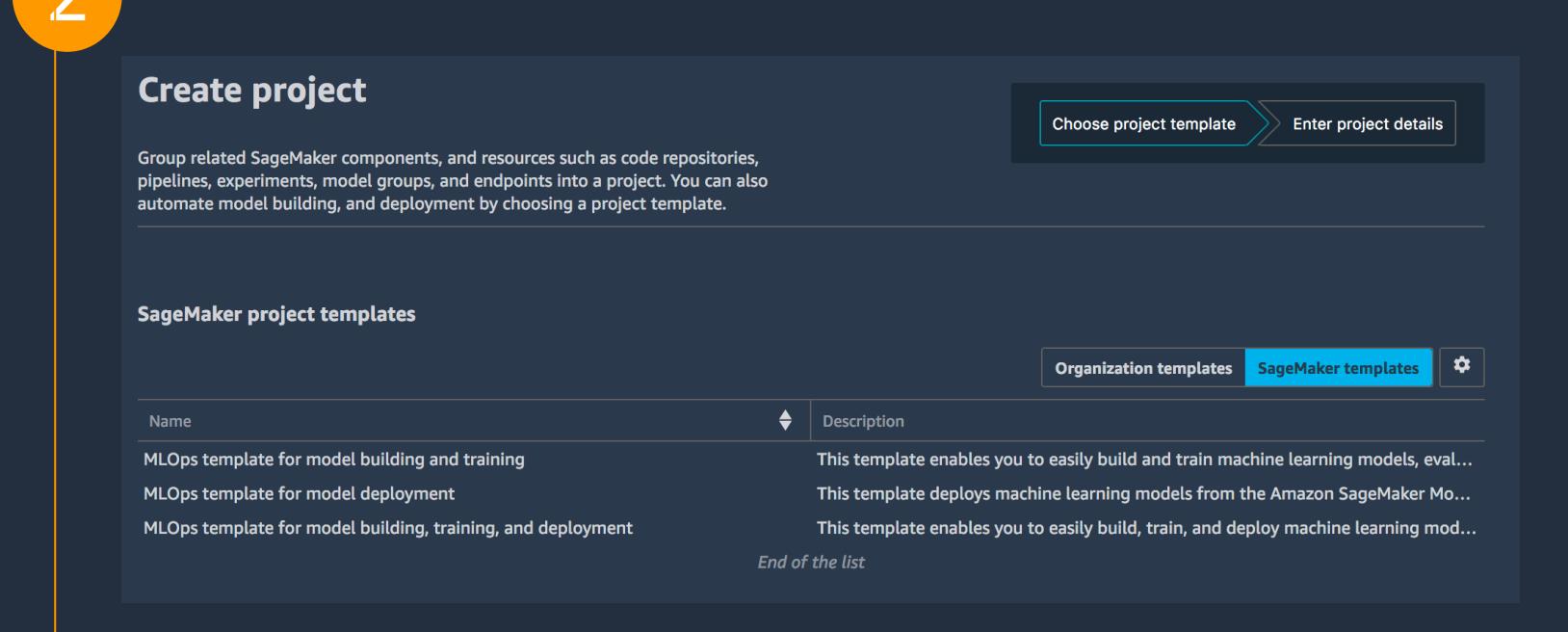
Built-In MLOps Project Templates

How it Works...

1 Select Create Project



2 Select Template



Amazon SageMaker Pipelines - Projects

Built-In MLOps Project Templates

How it Works...

3

Enter Project Details

Create project

Group related SageMaker components, and resources such as code repositories, pipelines, experiments, model groups, and endpoints into a project. You can also automate model building, and deployment by choosing a project template.

Project details

Please provide the following details for your project:

Name

Description - optional

Tags - optional

| Key | Value |
|------|-------------|
| team | product-xyz |

[Remove](#)

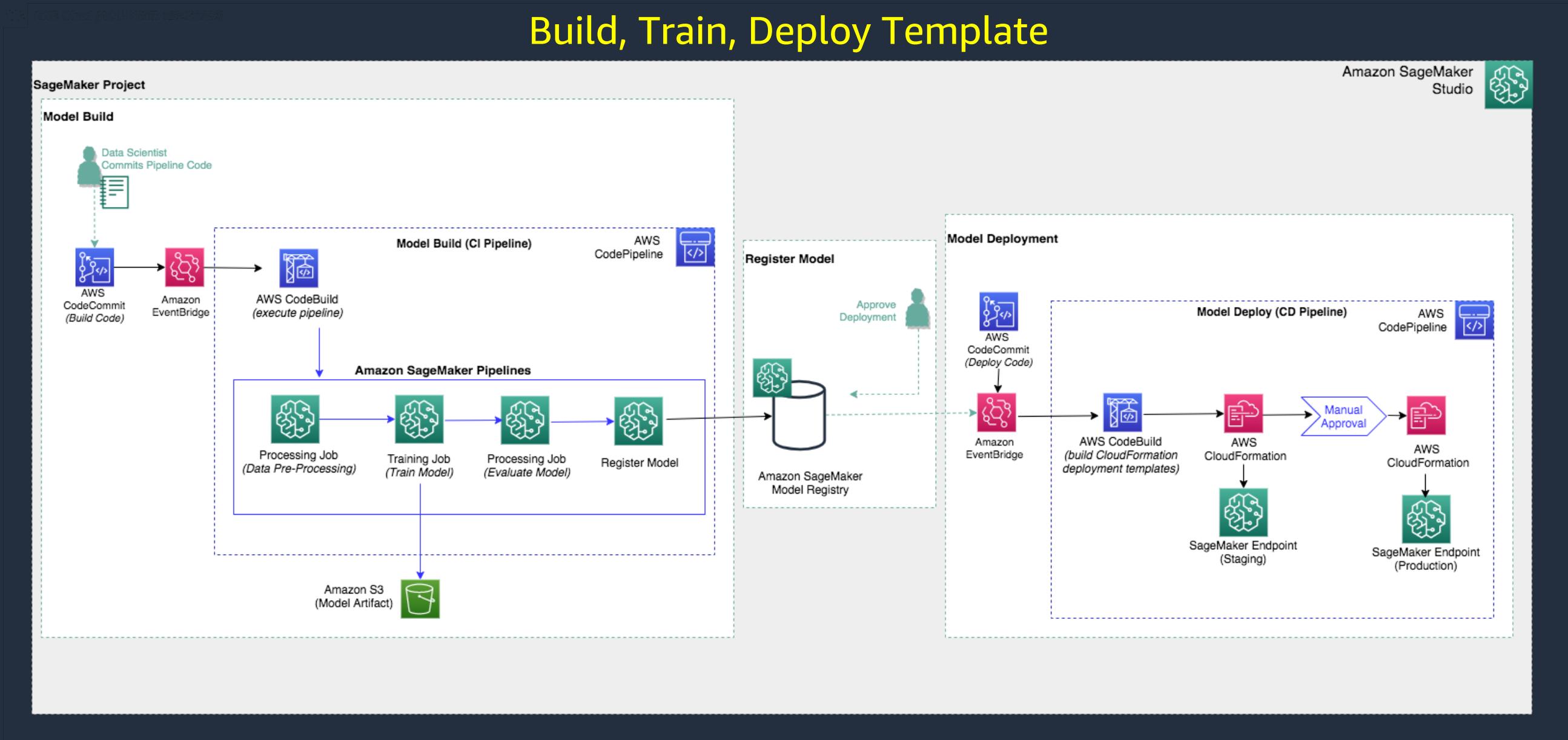
[Back](#) [Create project](#)

- custom tags 추가 가능
- built-in templates을 통해 배포된 리소스에는 project-name과 project-id 태그가 할당됨.
Example(s):
 - AWS CodePipeline → Pipelines
 - AWS CodeBuild → Build Projects
 - SageMaker Pipelines → Pipeline
 - → Training & Processing Jobs (via AWS CodeBuild)

머신 러닝을 위한 CI/CD 파이프라인을 생성하는데 필요한 모든 것이 자동으로 프로비저닝되고 구성됩니다.

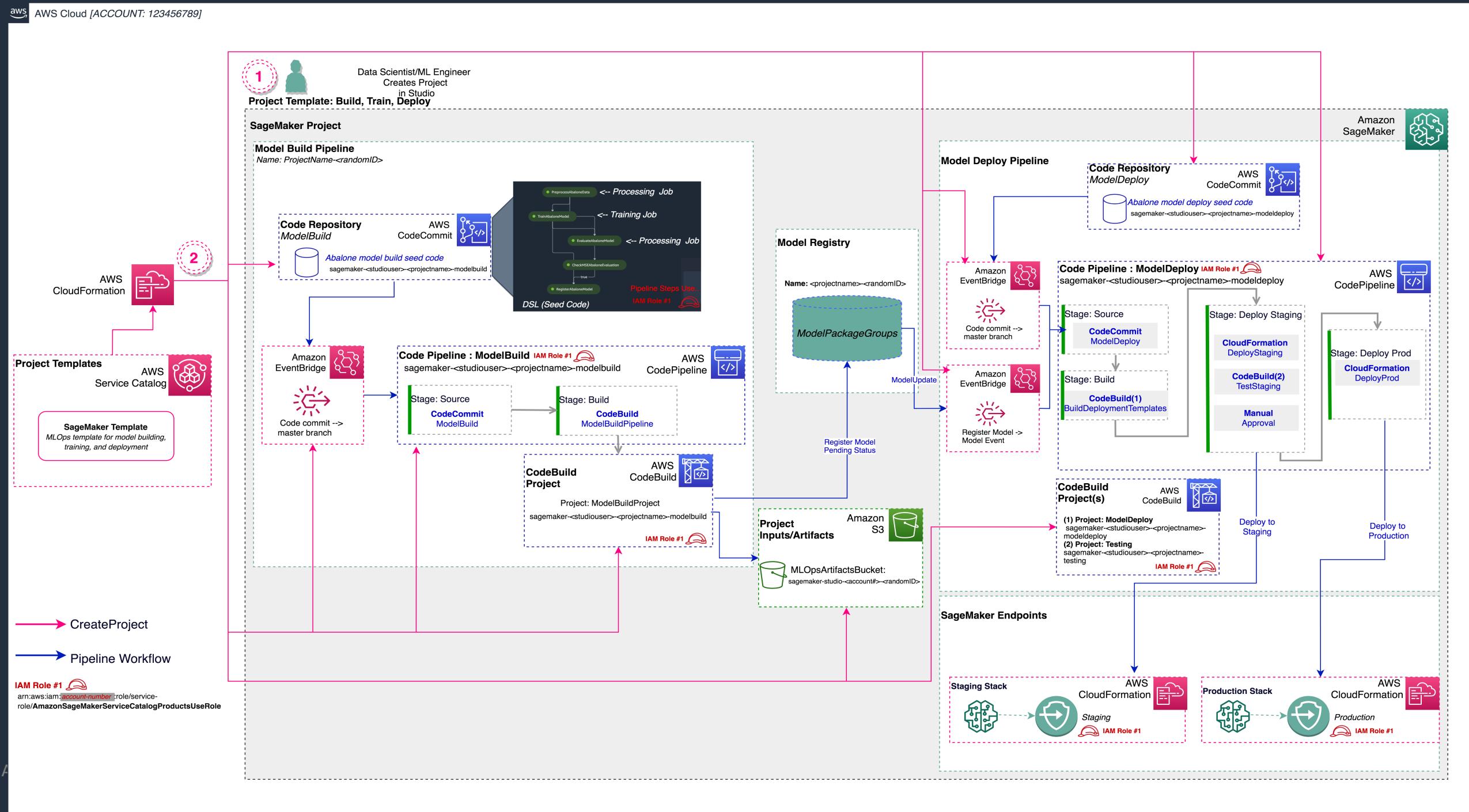
Amazon SageMaker Pipelines - Projects

High Level Services View



Amazon SageMaker Pipelines - Projects

Deep Dive Services View – Build, Train, Deploy



Amazon SageMaker Pipelines - Projects Underlying Services



AWS Service Catalog

Description: AWS Service Catalog는 organizations가 AWS에서 사용하도록 승인된 IT 서비스의 카탈로그를 생성하고 관리할 수 있도록 합니다.

- managed AWS Service Catalog Portfolio에 포함된 Products 통해 Built-In Project templates을 제공됩니다.

The screenshot shows the AWS Service Catalog interface under the 'Portfolios' tab. It displays a list of imported portfolios, with one entry visible:

| Name | Created time | Portfolio ID | ARN | Owner | Description | Share Type | Current vs. budget | Forecast vs. budget |
|--|------------------------------------|---------------------|--|------------------|-------------|------------|--------------------|---------------------|
| Amazon SageMaker Solutions and ML Ops products | Tue, Oct 27, 2020, 12:40:18 AM MDT | port-cvmmvo4n6uc7e2 | arn:aws:catalog:us-east-1:3582:44920887:portfolio/portfolio/por-t-cvmmvo4n6uc7e2 | Amazon SageMaker | - | IMPORTED | - | - |

The screenshot shows the AWS Service Catalog interface under the 'Products' tab. It displays a list of products, with one entry visible:

| Name | Id | Created time | Distributor | Provided by | Description |
|---|--------------------|-----------------------------------|-------------|------------------|---|
| MLOps template for model building, training, and deployment | prod-j3ufw6hl7utxm | Wed, Jan 27, 2021, 1:04:50 AM MST | - | Amazon SageMaker | This template enables you to easily build, train, and deploy machine learning models. You can adopt MLOps best practices and enable Continuous Integration/Continuous Deployment for building, training, and evaluating machine learning models using Amazon SageMaker Pipelines, registering models to the Model Registry, and automating model deployment. Amazon SageMaker creates an AWS CodeCommit code repository for you to manage your code and uses AWS CodePipeline to build, train, and deploy your machine learning models on pre-production and production Amazon SageMaker endpoints for real-time inference. |

Amazon SageMaker Pipelines - Projects

Underlying Services



AWS CloudFormation

Description: AWS CloudFormation templates을 사용하면 Infrastructure-as-Code를 사용하여 리소스를 일관되게 프로비저닝, 구성 및 관리할 수 있습니다.

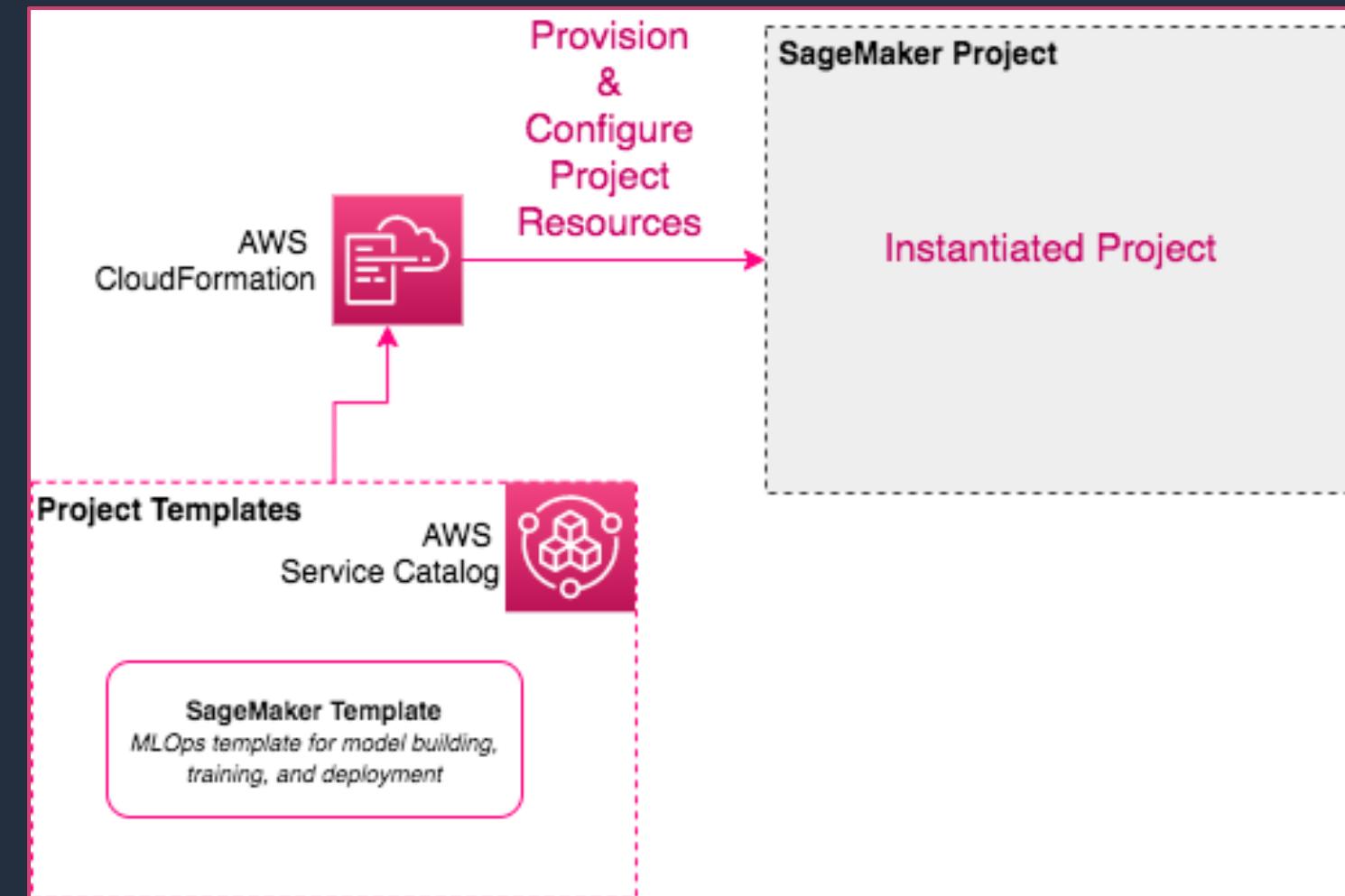
- SageMaker Pipelines는 AWS Cloud Formation을 사용하여 다음을 프로비저닝하고 구성합니다:
 - End-to-end pipeline를 구축하는데 필요한 프로젝트 리소스
 - Instantiated Project에 대한 모델 endpoint 배포

Amazon SageMaker Pipelines - Projects Underlying Services

AWS
CloudFormation



1. End-to-end pipeline를 구축하는데 필요한 프로젝트 리소스

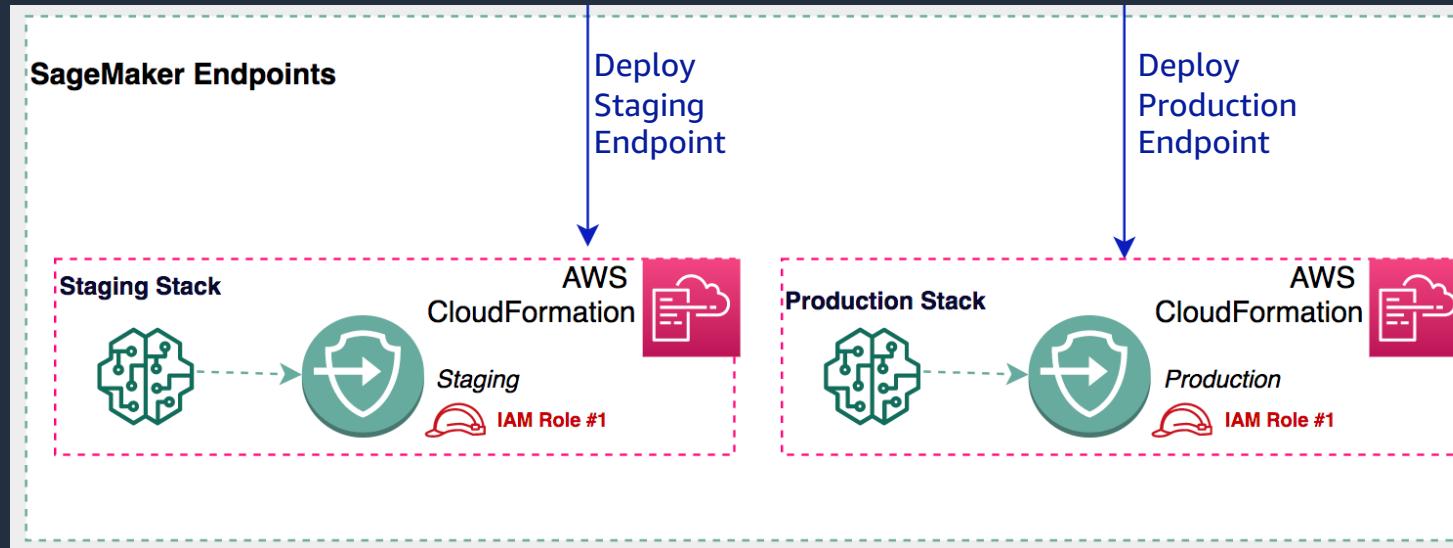


Amazon SageMaker Pipelines - Projects Underlying Services

AWS
CloudFormation

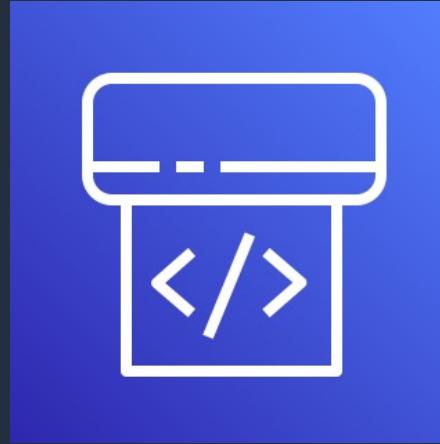


2. Instantiated Project에 대한 모델 endpoint 배포



Amazon SageMaker Pipelines - Projects

Underlying Services



AWS CodePipeline

Description: AWS CodePipeline는 fully managed Continuous Delivery (CD) service입니다.

- SageMaker Pipeline은 AWS CodePipeline을 사용하여 모델 빌드 및 모델 배포 작업을 조정합니다.

1. Model 생성

2. instantiated Project에 대한 Model endpoint 배포

Amazon SageMaker Pipelines - Projects Underlying Services



1. Model 생성

Developer Tools > CodePipeline > Pipelines > sagemaker-btd-abalone-p-7k5sogssnako-modelbuild

sagemaker-btd-abalone-p-7k5sogssnako-modelbuild

Source Succeeded
Pipeline execution ID: 9a7f0da7-0161-4b1a-9ac5-e99730b9e556

ModelBuildWorkflowCode
AWS CodeCommit
Succeeded - 7 days ago
08349aee

08349aee ModelBuildWorkflowCode: Initial commit by AWS CodeCommit

Disable transition

Build Succeeded
Pipeline execution ID: 9a7f0da7-0161-4b1a-9ac5-e99730b9e556

BuildAndExecuteSageMaker...
AWS CodeBuild
Succeeded - 7 days ago
Details

08349aee ModelBuildWorkflowCode: Initial commit by AWS CodeCommit



AWS
CodeCommit

Model Build Code



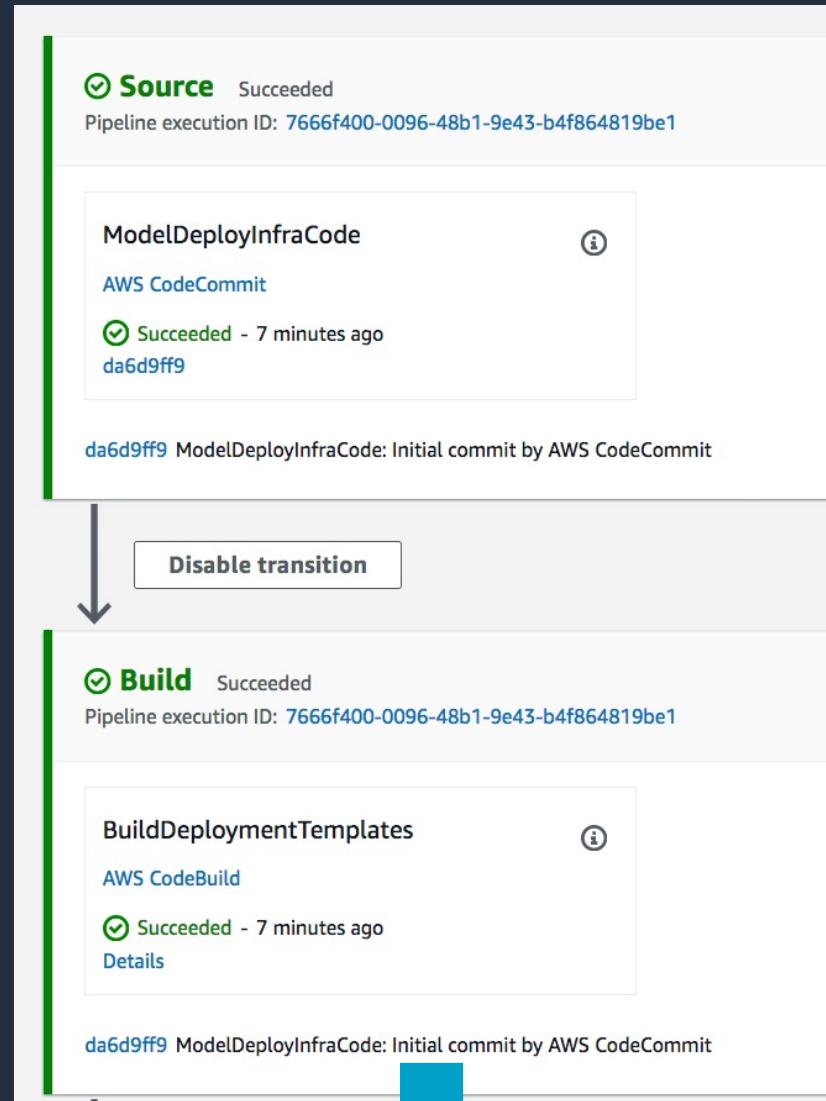
AWS
CodeBuild

SageMaker Pipelines 사용하여
Model Build Pipeline 실행

Amazon SageMaker Pipelines - Projects Underlying Services



2. instantiated Project에 대한 Model endpoint 배포



Model Deploy Code



AWS
CodeBuild

SageMaker Endpoints 생성하기 위한
CloudFormation templates 패키지

Amazon SageMaker Pipelines - Projects Underlying Services



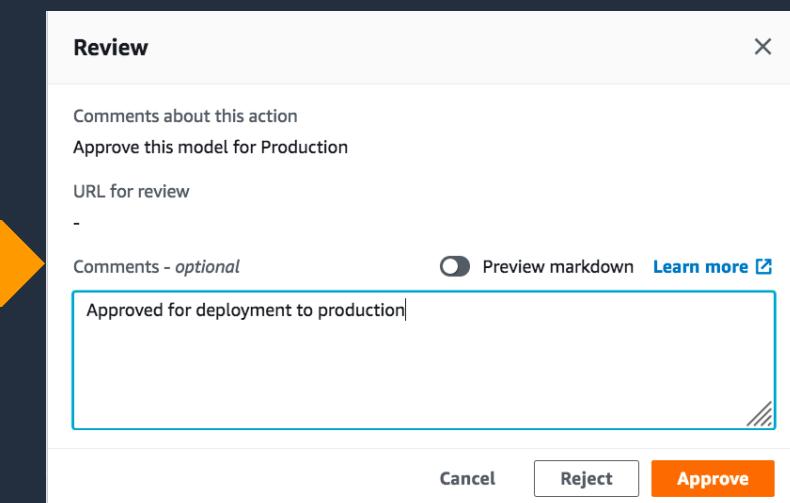
2. instantiated Project에 대한 Model endpoint 배포



A screenshot of the AWS CodePipeline console showing a pipeline execution. The stages are:

- DeployStaging**: Pending. Pipeline execution ID: 7666f400-0096-48b1-9e43-b4f864819be1. Sub-steps: DeployResourcesStaging (AWS CloudFormation), which succeeded 8 minutes ago.
- TestStaging**: Succeeded - 7 minutes ago. Sub-step: AWS CodeBuild, which succeeded 7 minutes ago.
- ApproveDeployment**: Waiting for approval. Sub-step: Manual approval. A "Review" button is present.

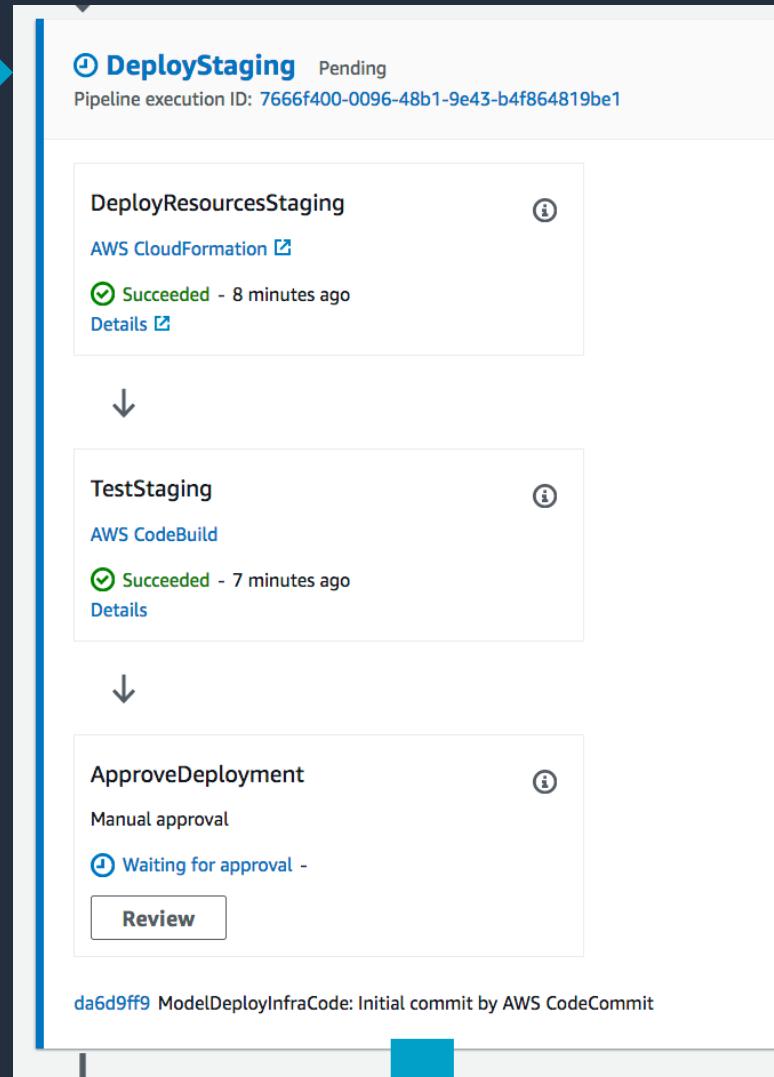
Commit message: da6d9ff9 ModelDeployInfraCode: Initial commit by AWS CodeCommit



Amazon SageMaker Pipelines - Projects Underlying Services



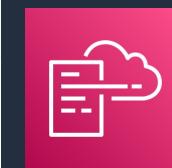
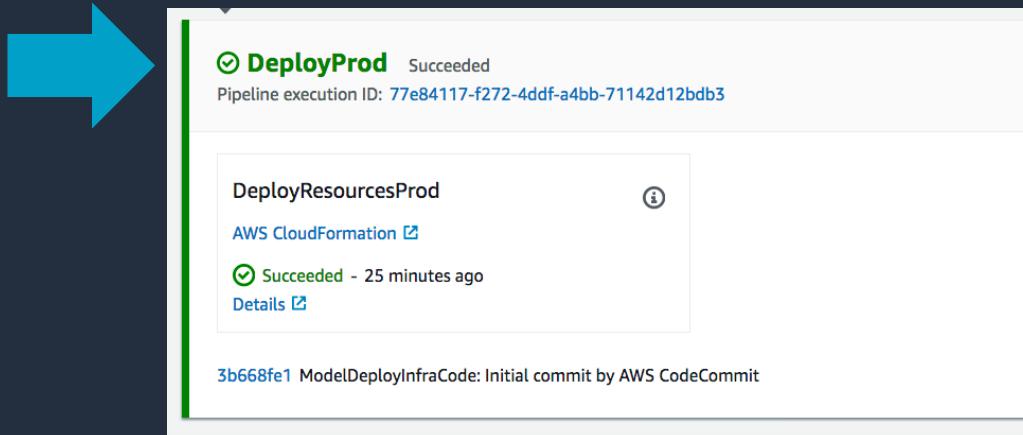
2. instantiated Project에 대한 Model endpoint 배포



Amazon SageMaker Pipelines - Projects Underlying Services



2. instantiated Project에 대한 Model endpoint 배포



AWS
CloudFormation

Create SageMaker Endpoint

Amazon SageMaker Pipelines - Projects

Features



- Built-In MLOps Project Templates 활용 :
 1. Build, Train, Deploy
 2. Build, Train
 3. Deploy
- Custom MLOps Project Templates 생성

Amazon SageMaker Pipelines - Projects

Building Custom Templates

1 Create Your Custom Template



AWS
CloudFormation

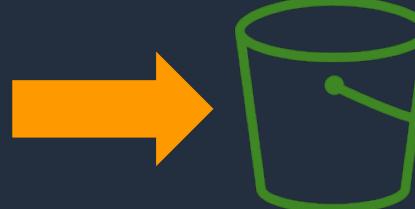
Required parameters to include:

SageMakerProjectName:
Type: String
Description: Name of the project



SageMakerProjectId:
Type: String
Description: Service generated Id of the project.

2 Upload CloudFormation Template to S3



Amazon S3

Tip: Use the SageMaker built-in templates as a starting point

Amazon SageMaker Pipelines - Projects

Building Custom Templates

Service Catalog Tasks

AWS
Service Catalog



3 Create Portfolio

The screenshot shows the 'Portfolios' page in the AWS Service Catalog. It has tabs for 'Local', 'Imported', and 'Getting Started library'. Under 'Local portfolios (1)', there is a single entry with a search bar below it. At the bottom right, there is a 'Create portfolio' button. An orange arrow points from the top right towards this button.

4 Create Product

The screenshot shows the 'Admin - Product list' page in the AWS Service Catalog. It has a search bar and a table showing one product. At the bottom right, there is a 'Upload new product' button. An orange arrow points from the top right towards this button. To the right of the product list, there is a large orange arrow pointing right, indicating the flow from step 3 to step 4.

Required Tags

The screenshot shows the 'Tags' section of the AWS Service Catalog. It has tabs for 'Versions (1)', 'Portfolios (1)', 'Tags (1)', and 'TagOptions (0)'. Under 'Tags (1)', there is a search bar and a table with one tag entry. The tag entry shows 'Key: sagemaker:studio-visibility' and 'Value: true'. A large orange arrow points from the 'Required Tags' section towards this tag entry.

Your template will now be available for use in SageMaker Studio!

Amazon SageMaker Pipelines

Resources

SageMaker Pipelines Documentation

- Pipelines: <https://docs.aws.amazon.com/sagemaker/latest/dg/pipelines.html>
- Projects: <https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-projects.html>

SageMaker Pipelines Blogs

- Launch: <https://aws.amazon.com/blogs/aws/amazon-sagemaker-pipelines-brings-devops-to-machine-learning-projects/>
- Modifying Project Code: <https://aws.amazon.com/blogs/machine-learning/building-automating-managing-and-scaling-ml-workflows-using-amazon-sagemaker-pipelines/>
- Cross-Account Deployment: <https://aws.amazon.com/blogs/machine-learning/multi-account-model-deployment-with-amazon-sagemaker-pipelines/>