Mastering Amazon SageMaker

MLOps

Name

Role

Email

Agenda

- Amazon Sagemaker MLOps Capabilities
    - SageMaker Pipelines
    - SageMaker Model Registry
    - SageMaker Projects
- Demo
- Q&A

aws machine learning

Amazon SageMaker MLOps capabilities

aws machine learning

Operational challenges with managing the ML lifecycle

Manual iterative processes slow down ML innovation

Difficult to scale the number of models in production

CI/CD for ML requires writing custom code

Compliance requirements are difficult to meet

aws machine learning

Amazon SageMaker MLOps

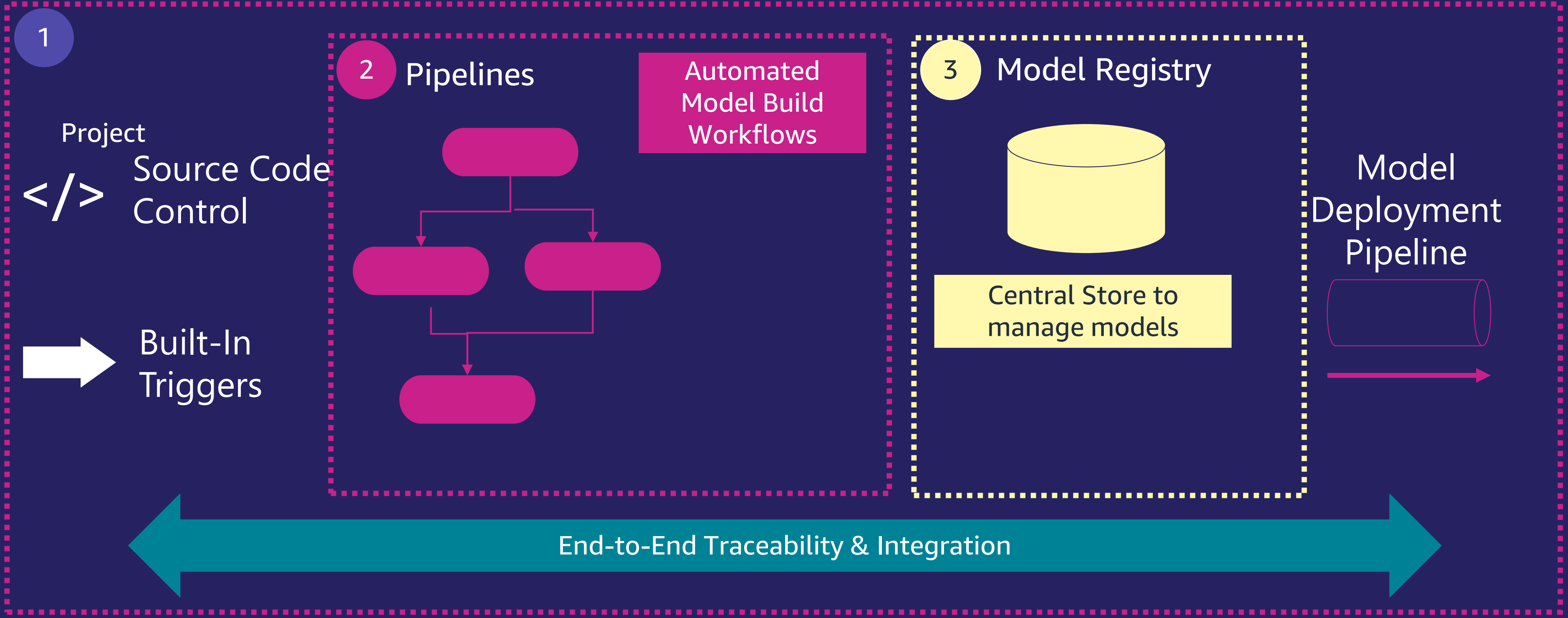
Streamline the ML lifecycle

- Automate ML workflows to scale model development
- Build CI/CD pipelines for ML to accelerate model deployment
- Catalog model versions, metadata, metrics, and approvals for traceability and reusability
- Track lineage for troubleshooting and compliance
- Maintain accuracy of predictions after models are deployed
- Enhance governance and security

aws machine learning

Amazon SageMaker MLOps

Components



1

Project

</>

Source Code Control

Built-In Triggers

2 Pipelines

Automated Model Build Workflows

3 Model Registry

Central Store to manage models

Model Deployment Pipeline

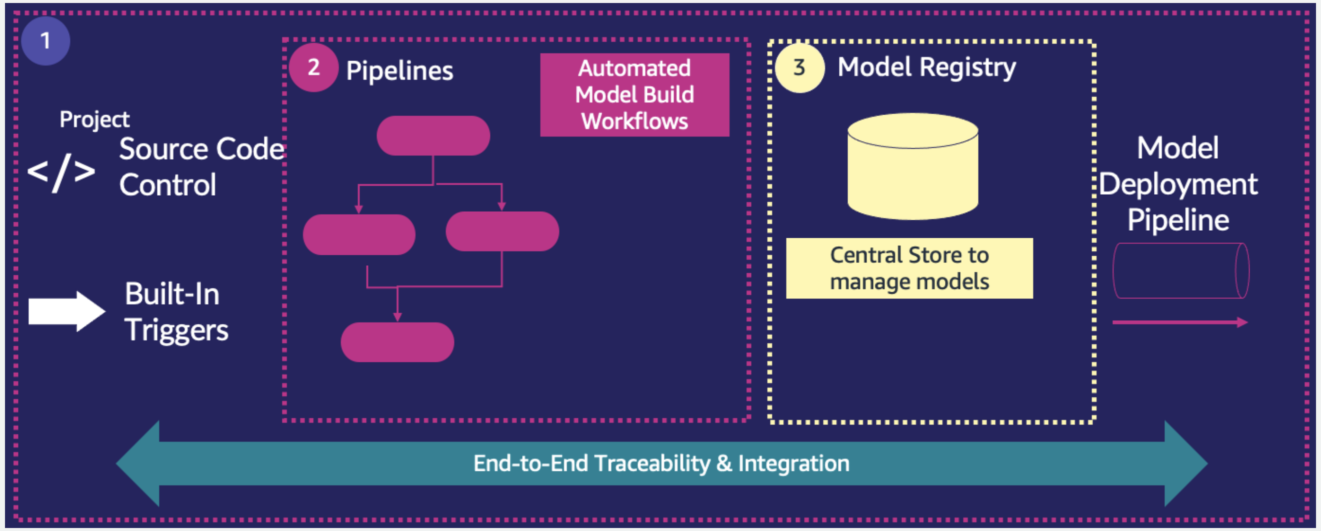
End-to-End Traceability & Integration

aws machine learning

Amazon SageMaker Pipelines

Components

- **Flexible:**
  - ✓ Use Projects for end-to-end pipelines that incorporate CI/CD practices
  - ✓ Optionally, use Pipelines or Model Registry without Projects to meet the needs of your use case


1
Project
</>
Source Code Control
Built-In Triggers
2
Pipelines
Automated Model Build Workflows
3
Model Registry
Central Store to manage models
Model Deployment Pipeline
End-to-End Traceability & Integration

Amazon SageMaker Pipelines

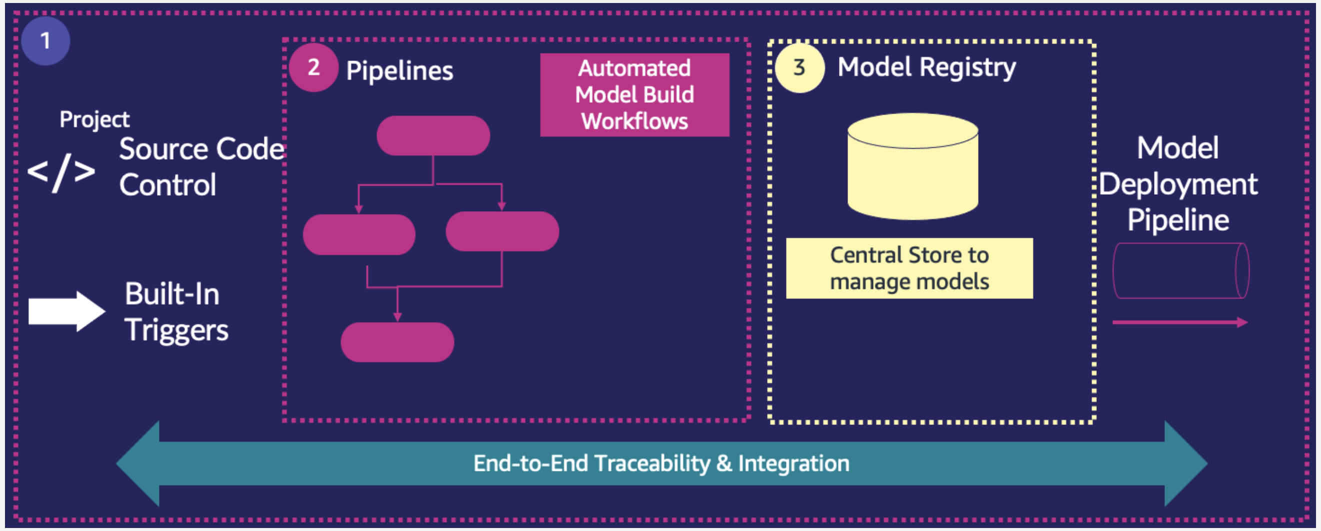
Components

Extensible:

- ✓ Take advantage of built-in MLOps Project Templates

or

- ✓ Create custom MLOps Project Templates to meet the unique demands of your corporate or regulatory requirements


1
Project
Source Code Control
Built-In Triggers
2
Pipelines
Automated Model Build Workflows
3
Model Registry
Central Store to manage models
Model Deployment Pipeline
End-to-End Traceability & Integration

SageMaker Pipelines

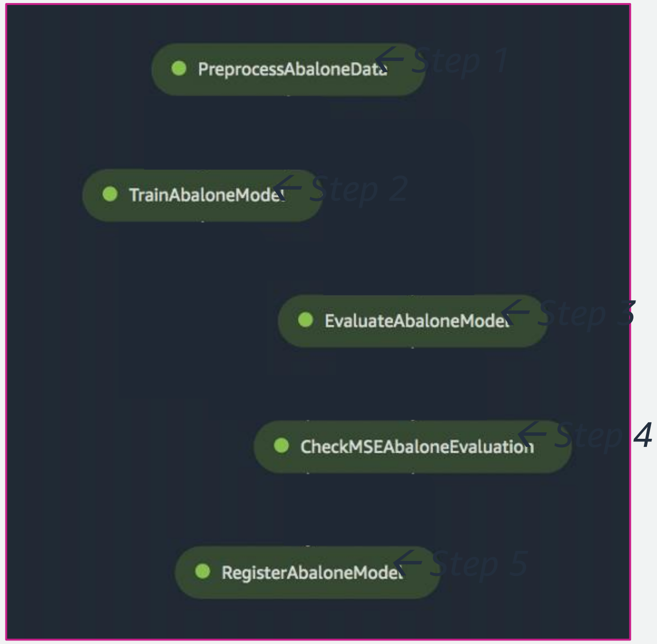
Creating Amazon SageMaker Automated Pipelines

Pipeline-as-Code

How it Works …

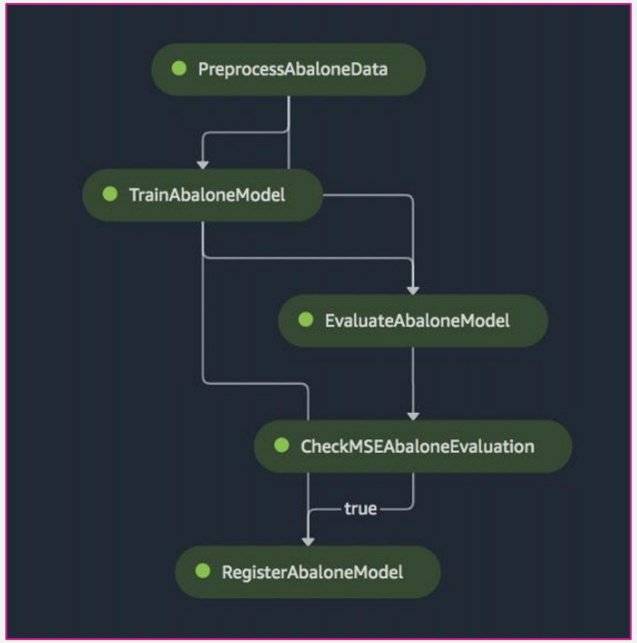
1 Create Steps →

Define & configure each step



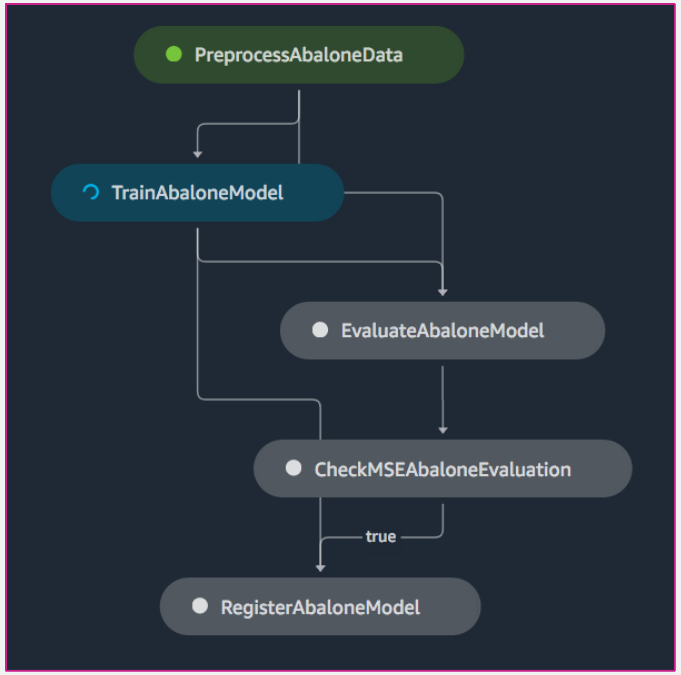
2 Define Pipeline →

Define & configure the workflow



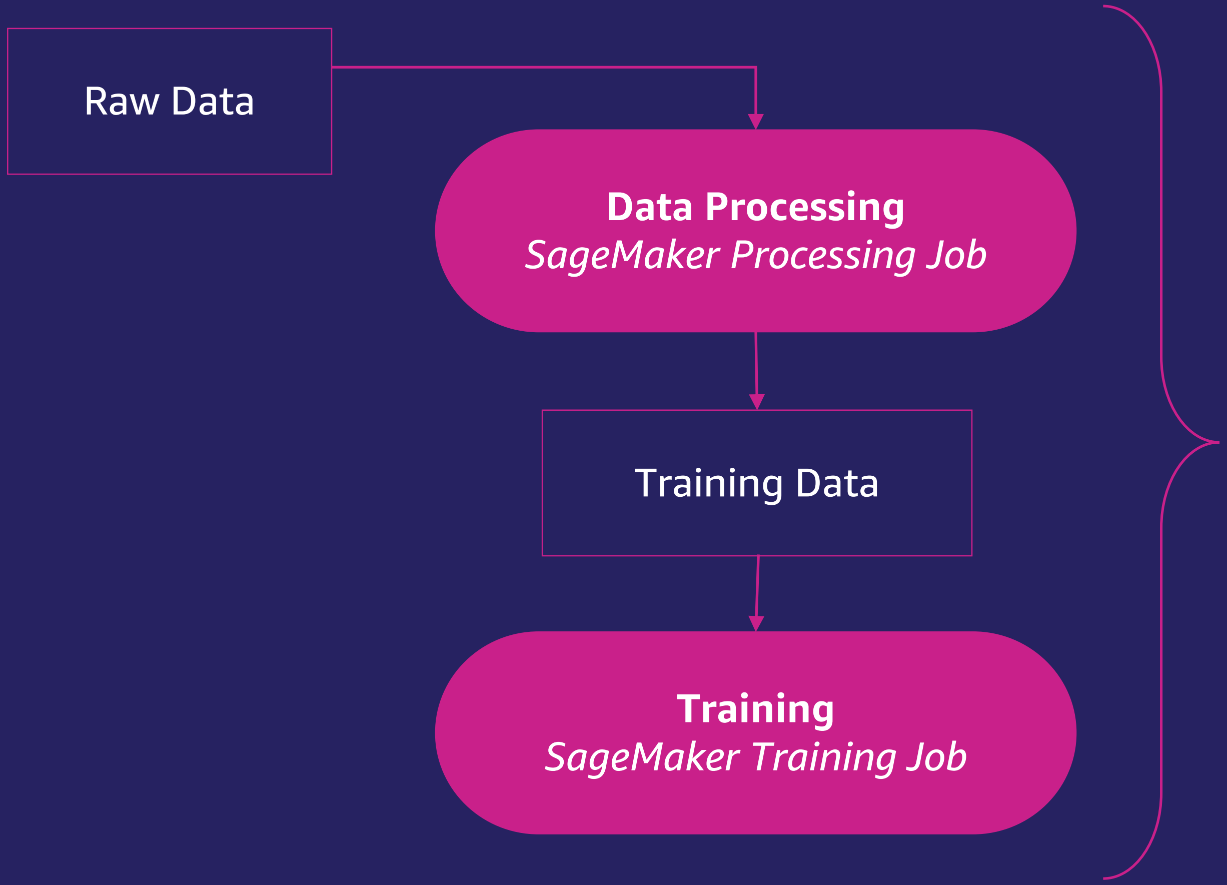
3 Start Pipeline →

Execute the pipeline



Amazon SageMaker Pipelines

Support for Step Caching →



What if you want to:

- tweak hyperparameters?
- modify training code?

Amazon SageMaker Pipelines

Support for Pipeline Parameters

1 Configure your parameter →

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

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

2 Pass in parameter on pipeline create →

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

3 Optionally, pass a non-default value for pipeline execution →

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

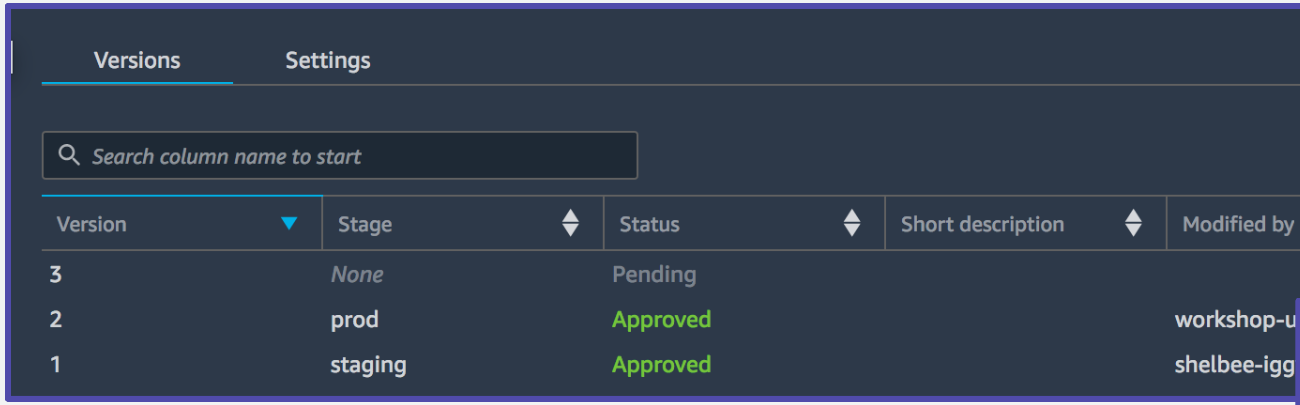
aws machine learning

SageMaker Model Registry

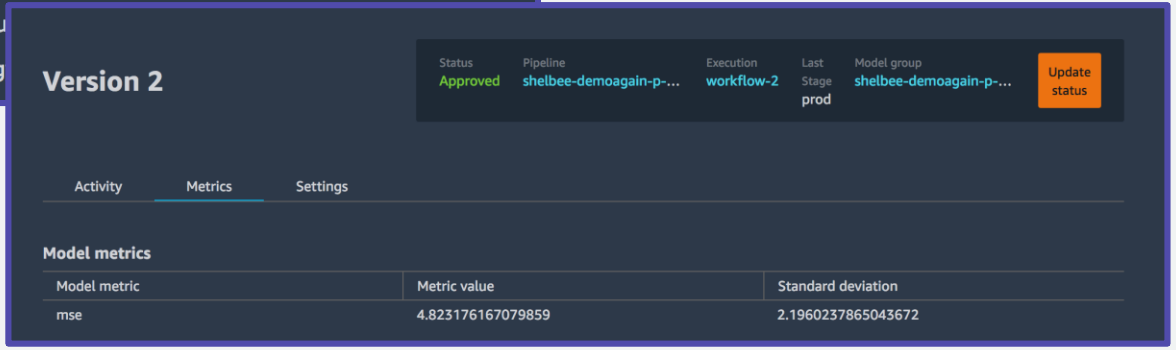
aws machine learning

Amazon SageMaker Pipelines

Model Registry



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



Version 2

Status: Approved
Pipeline: shelbee-demoagain-p-...
Execution: workflow-2
Last Stage: prod
Model group: shelbee-demoagain-p-...

Model metrics

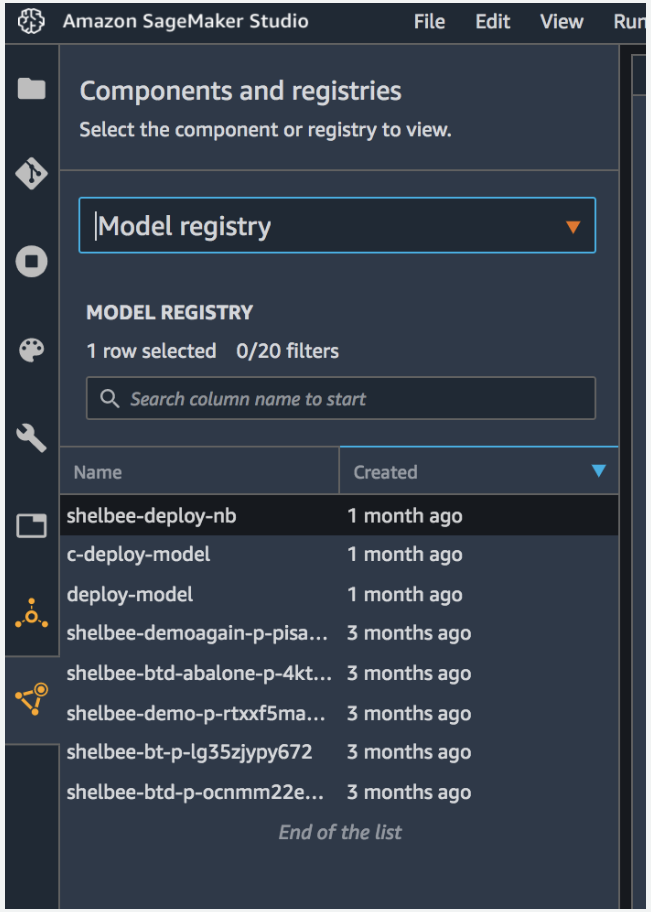
Model metric	Metric value	Standard deviation
mse	4.823176167079859	2.1960237865043672

- 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
- Rollback model versions

aws machine learning

Amazon SageMaker Pipelines

Model Registry



Components and registries

Select the component or registry to view.

Model registry ▾

MODEL REGISTRY

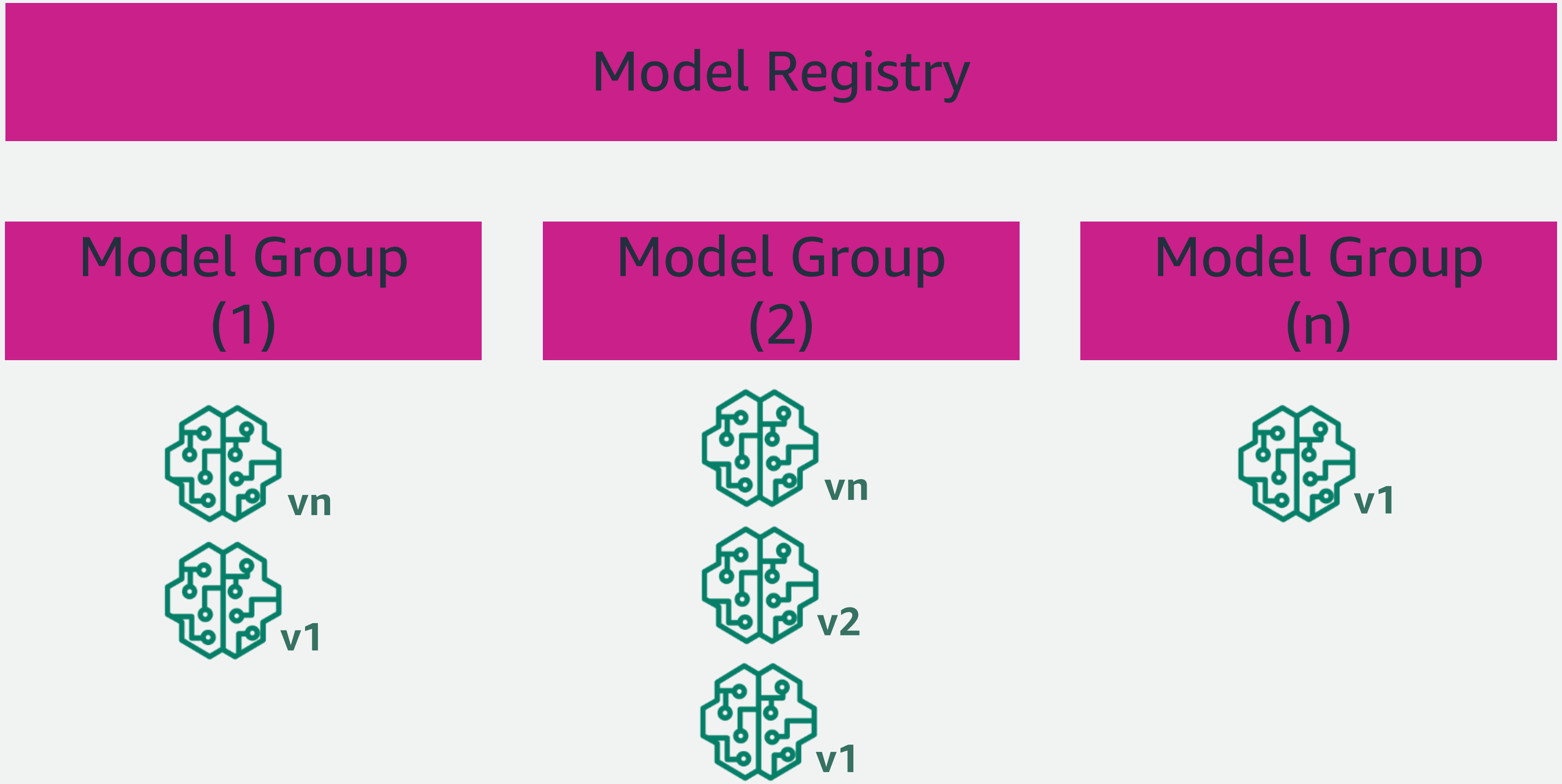
1 row selected 0/20 filters

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 (1)

Model Group (2)

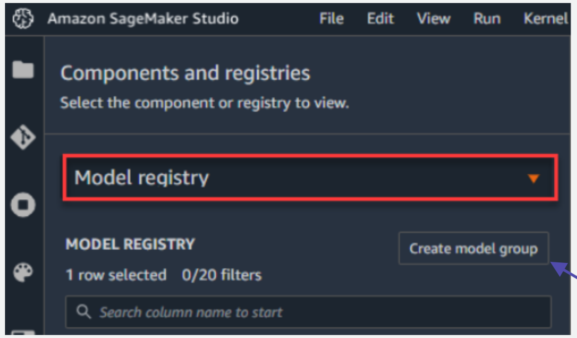
Model Group (n)

Amazon SageMaker Pipelines

Model Registry

How it Works …

1 Create Model Group →



Amazon SageMaker Studio File Edit View Run Kernel

Components and registries

Select the component or registry to view.

Model registry

MODEL REGISTRY

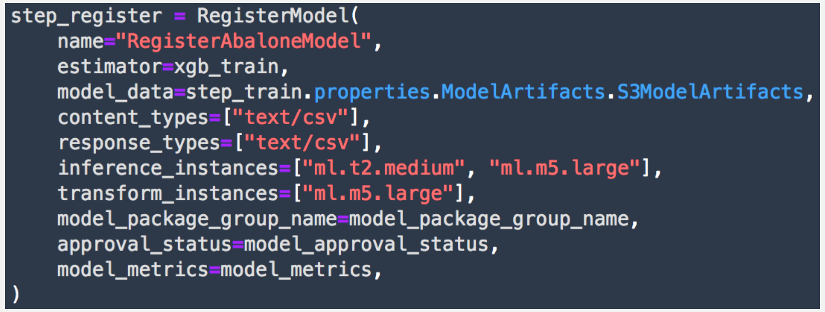
Create model group

1 row selected 0/20 filters

Search column name to start

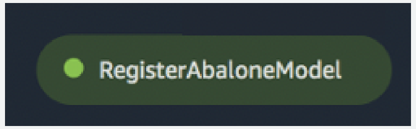
2 Register a Model Version →

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_mode_package_response["ModelPackageArn"]
print('ModelPackage Version ARN : {}'.format(model_package_arn))
```

aws machine learning

SageMaker Projects

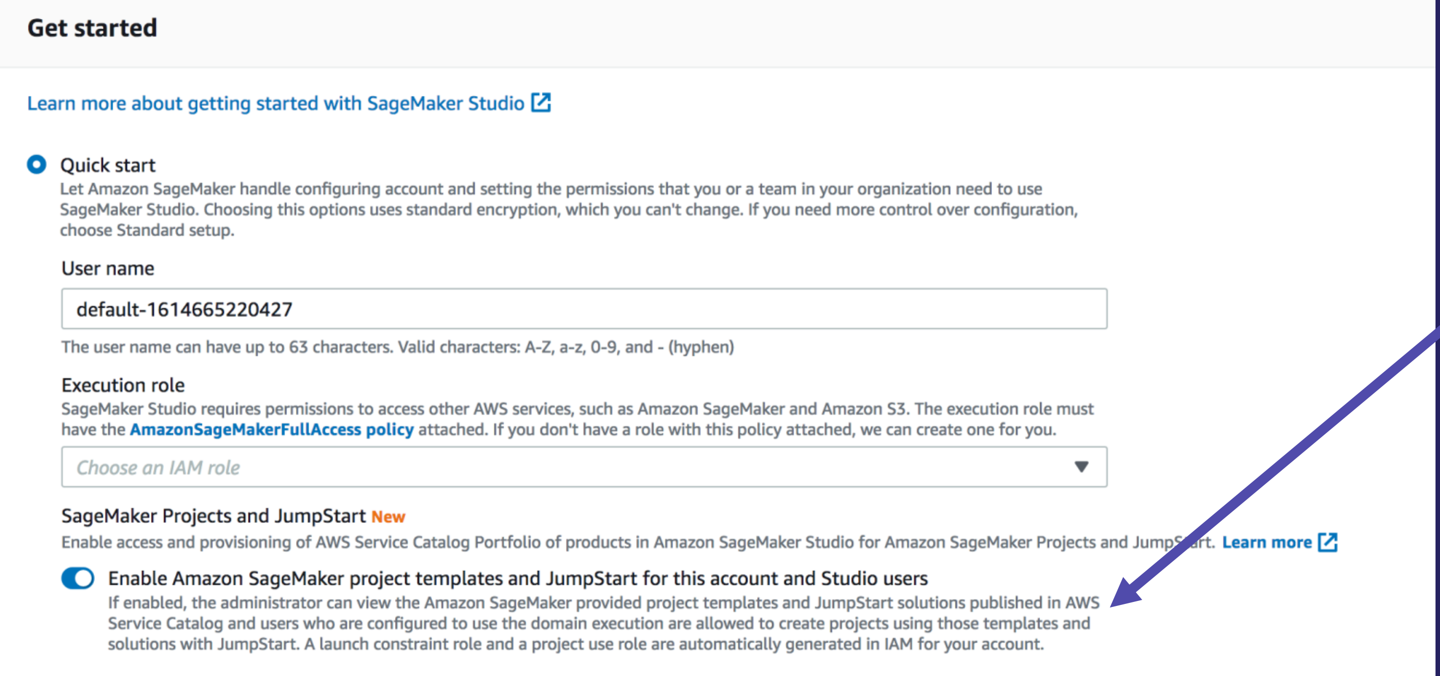
aws machine learning

Amazon SageMaker Pipelines - Projects

Getting Started

One-Time Setup

1 *Enable Projects for your Studio Domain*



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 options uses standard encryption, which you can't change. If you need more control over configuration, choose Standard setup.

User name

default-1614665220427

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.

Enable on a new domain or update an existing domain

Amazon SageMaker - Projects

Features

- Utilize Built-In MLOps Project Templates:
    1. Build, Train, Deploy
    2. Build, Train
    3. Deploy

- Create Custom MLOps Project Templates

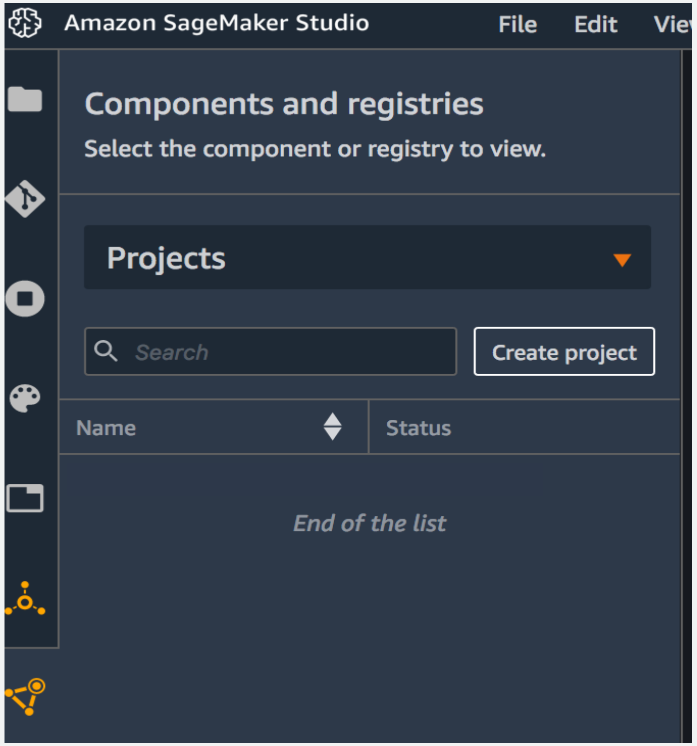
aws machine learning

Amazon SageMaker Pipelines - Projects

Built-In ML-Ops Project Templates

How it Works …

1 *Select Create Project*



Components and registries

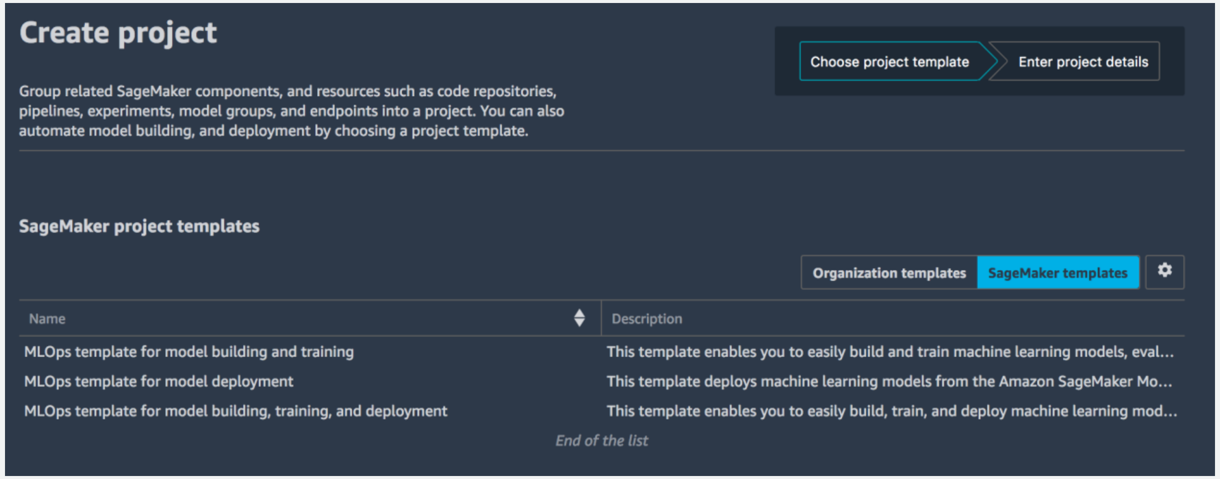
Select the component or registry to view.

Projects

Name	Status

End of the list

2 *Select Template*



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.

SageMaker project templates

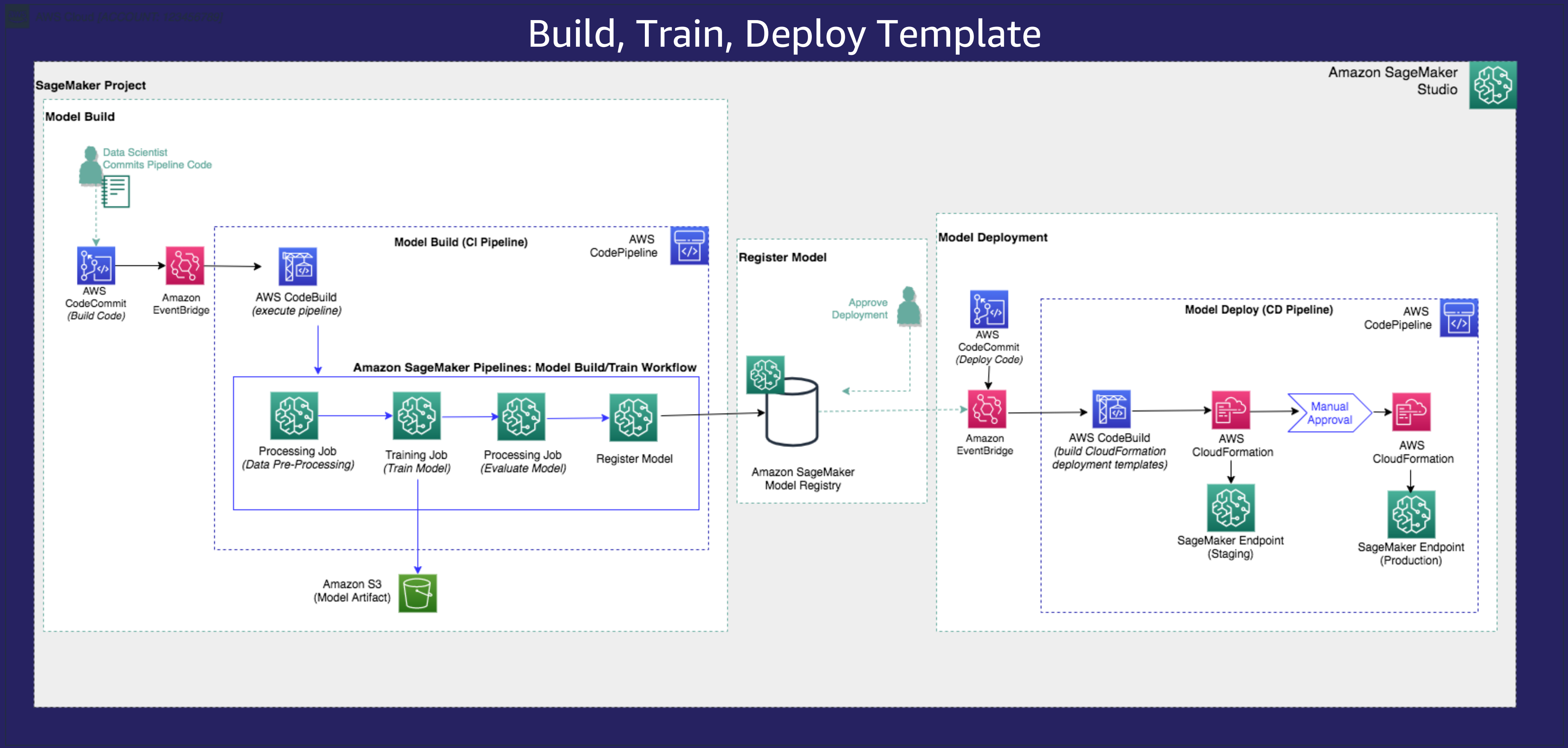
Name	Description
MLOps template for model building and training	This template enables you to easily build and train machine learning models, eval…
MLOps template for model deployment	This template deploys machine learning models from the Amazon SageMaker Mo…
MLOps template for model building, training, and deployment	This template enables you to easily build, train, and deploy machine learning mod…

End of the list

Everything needed to create a CI/CD Pipeline for Machine Learning gets automatically provisioned & configured for you……

Amazon SageMaker Pipelines - Projects

High Level Services View


AWS Cloud [ACCOUNT: 123456789]
Build, Train, Deploy Template
SageMaker Project
Amazon SageMaker Studio
Model Build
Data Scientist Commits Pipeline Code
AWS CodeCommit (Build Code)
Amazon EventBridge
Model Build (CI Pipeline)
AWS CodePipeline
AWS CodeBuild (execute pipeline)
Amazon SageMaker Pipelines: Model Build/Train Workflow
Processing Job (Data Pre-Processing)
Training Job (Train Model)
Processing Job (Evaluate Model)
Register Model
Amazon S3 (Model Artifact)
Register Model
Approve Deployment
Amazon SageMaker Model Registry
Model Deployment
AWS CodeCommit (Deploy Code)
Amazon EventBridge
Model Deploy (CD Pipeline)
AWS CodePipeline
AWS CodeBuild (build CloudFormation deployment templates)
AWS CloudFormation
Manual Approval
AWS CloudFormation
SageMaker Endpoint (Staging)
SageMaker Endpoint (Production)

aws machine learning

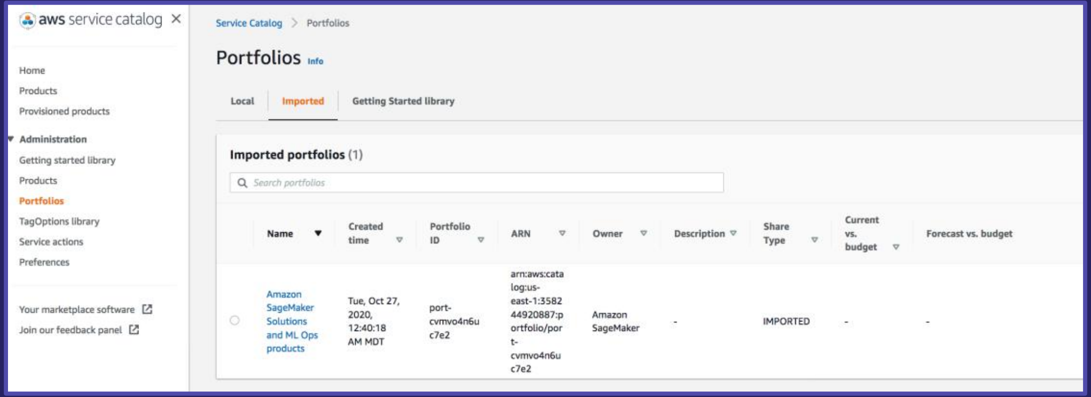
Amazon SageMaker Pipelines - Projects

Underlying Services

AWS Service Catalog

Description: *AWS Service Catalog allows organizations to create and manage catalogs of IT services that are approved for use on AWS.*

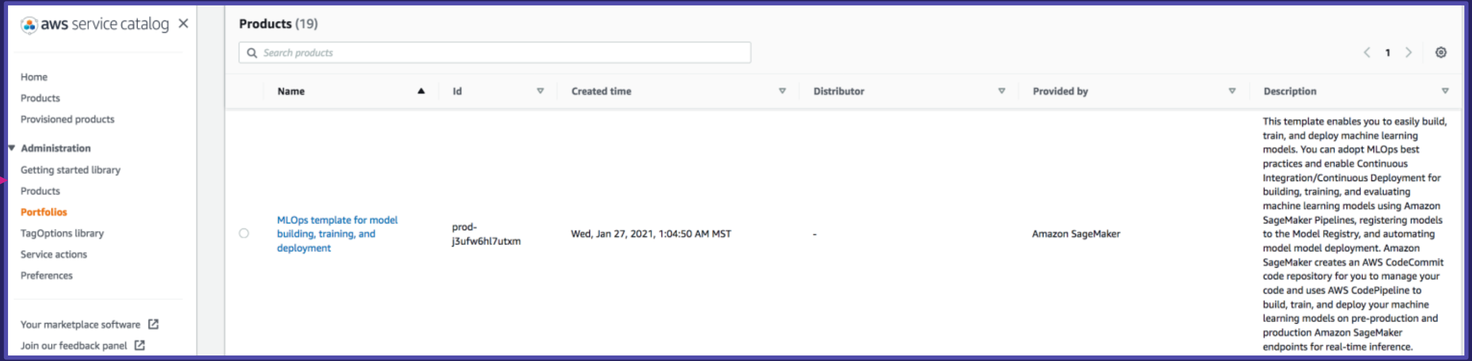
- Built-In Project templates are offered through Products contained in a managed AWS Service Catalog Portfolio



Portfolios Info

Imported portfolios (1)

	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-cvmvo4n6uc7e2	arn:aws:catalog:us-east-1:358244920887:portfolio/port-cvmvo4n6uc7e2	Amazon SageMaker	-	IMPORTED	-	-



Products (19)

	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 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.

aws machine learning

Amazon SageMaker Pipelines - Projects

Underlying Services

AWS CloudFormation

Description: *AWS CloudFormation templates allow for the ability to consistently provision, configure, and manage resources using Infrastructure-as-Code*

- SageMaker Pipelines uses AWS CloudFormation to provision & configure:

    1. Project resources needed to build an end-to-end pipeline

    2. Model endpoint deployments for an instantiated Project

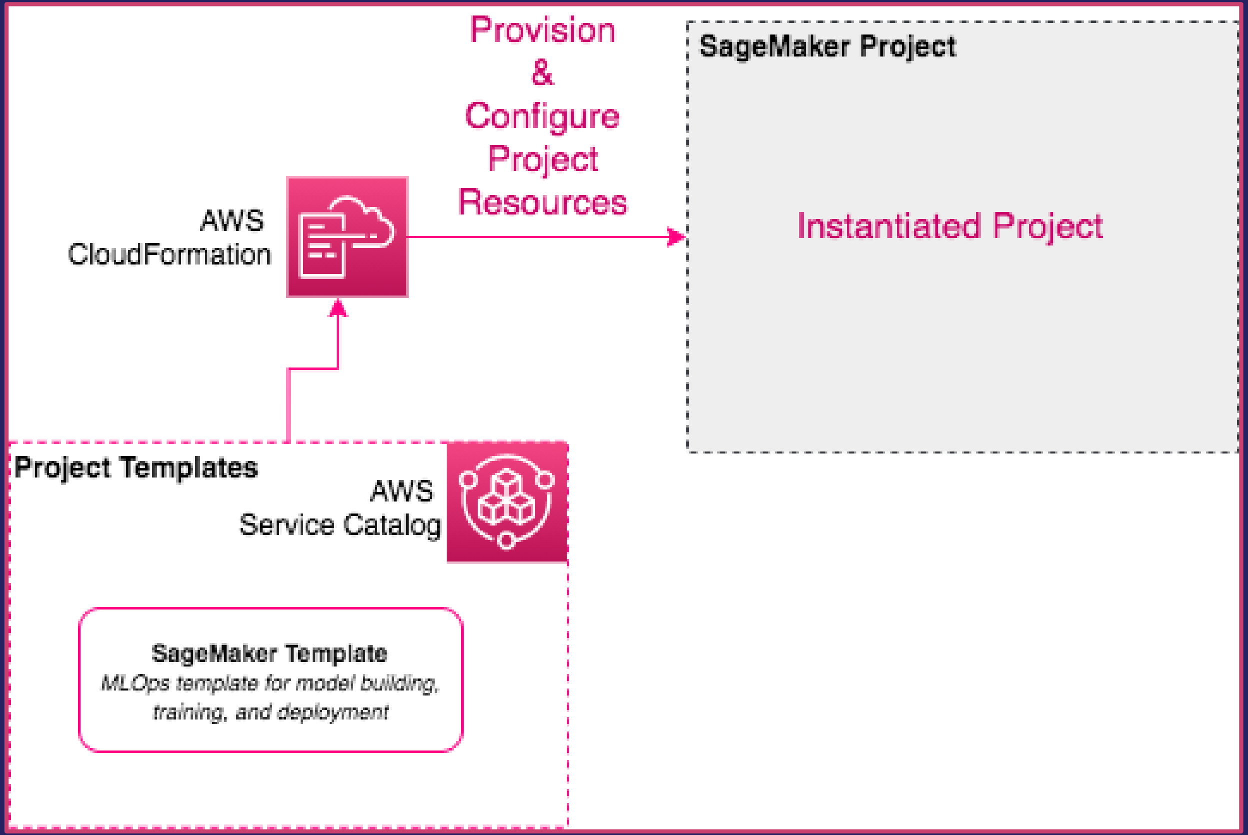
aws machine learning

Amazon SageMaker Pipelines - Projects

Underlying Services

AWS CloudFormation

1. Project resources needed to build an end-to-end pipeline



aws machine learning

Amazon SageMaker Pipelines - Projects

Underlying Services

AWS CodePipeline

Description: *AWS CodePipeline is a fully managed Continuous Delivery (CD) service*

- SageMaker Pipelines uses AWS CodePipeline to orchestrate model build and model deploy activities

1. Model Build:

2. Model endpoint deployments for an instantiated Project

Amazon SageMaker Pipelines - Projects Features

- Utilize Built-In MLOps Project Templates:
  1. Build, Train, Deploy
  2. Build, Train
  3. Deploy

- Create Custom MLOps Project Templates

aws machine learning

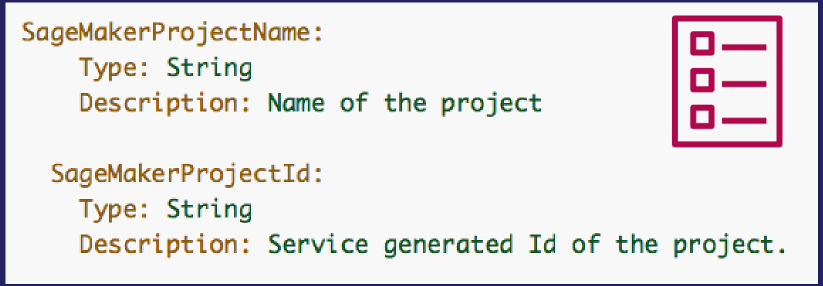
Amazon SageMaker Pipelines - Projects

Building Custom Templates

1 *Create Your Custom Template*

AWS
CloudFormation

Required parameters to include:

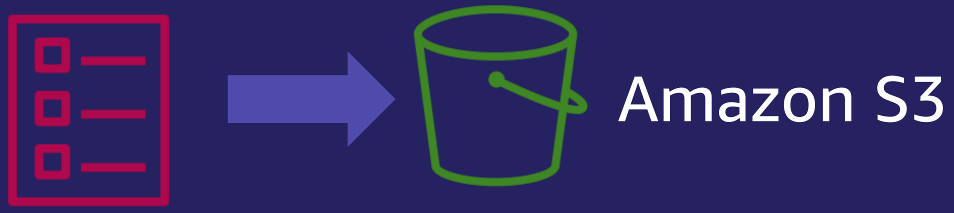


```
SageMakerProjectName:
    Type: String
    Description: Name of the project

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

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

2 *Upload CloudFormation Template to S3*

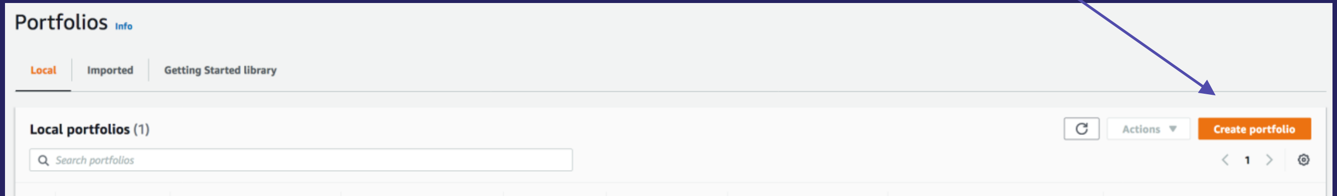


Amazon SageMaker Pipelines - Projects
Building Custom Templates

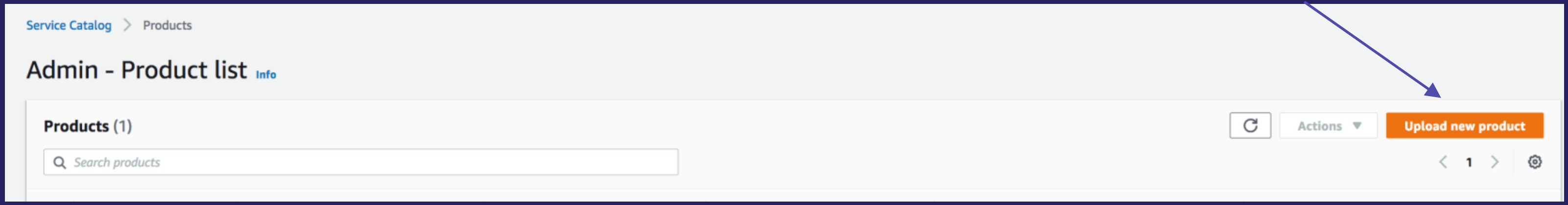
AWS
Service Catalog

Service Catalog Tasks

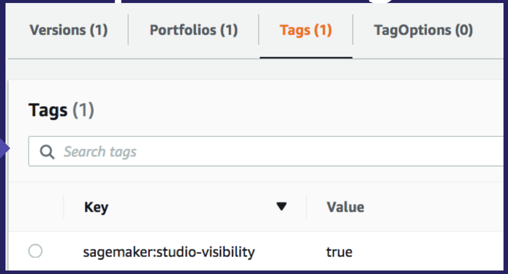
3 *Create Portfolio*



4 *Create Product*



Required Tags



	Key	Value
○	sagemaker:studio-visibility	true

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

Amazon SageMaker

Next Steps

Onboarding & Processing

- https://docs.aws.amazon.com/sagemaker/latest/dg/gs-studio-onboard.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/processing-job.html

Training

- https://docs.aws.amazon.com/sagemaker/latest/dg/train-model.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/distributed-training.html
- https://aws.amazon.com/sagemaker/debugger

Deployment

- https://docs.aws.amazon.com/sagemaker/latest/dg/realtime-endpoints.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/async-inference.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/batch-transform.html

https://github.com/aws/amazon-sagemaker-examples

https://sagemaker.readthedocs.io/en/stable/index.html

aws machine learning

Q & A

Please Complete the session Survey

Thank you!