



Implementing MLOps practices with Amazon SageMaker

Shelbee Eigenbrode

AI/ML Specialist Solutions Architect

Agenda

300 Level



- MLOps: Setting the stage
- SageMaker MLOps-ready capabilities
- SageMaker Pipelines
- Wrap-up

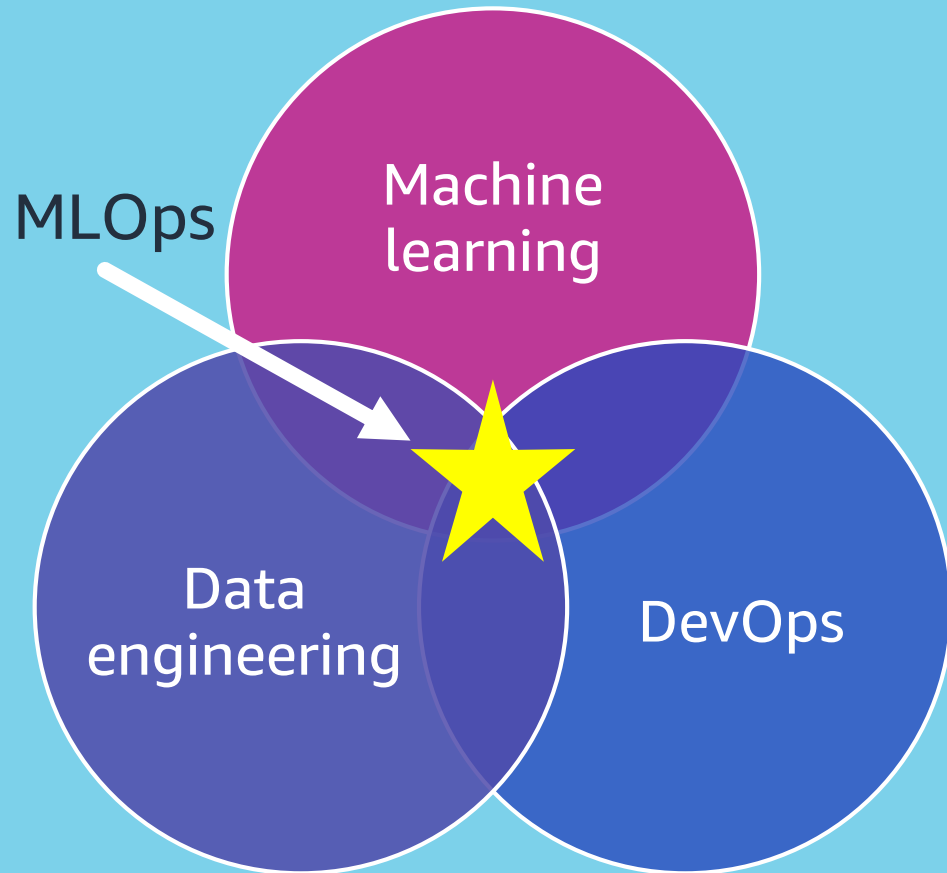
MLOps: Setting the stage



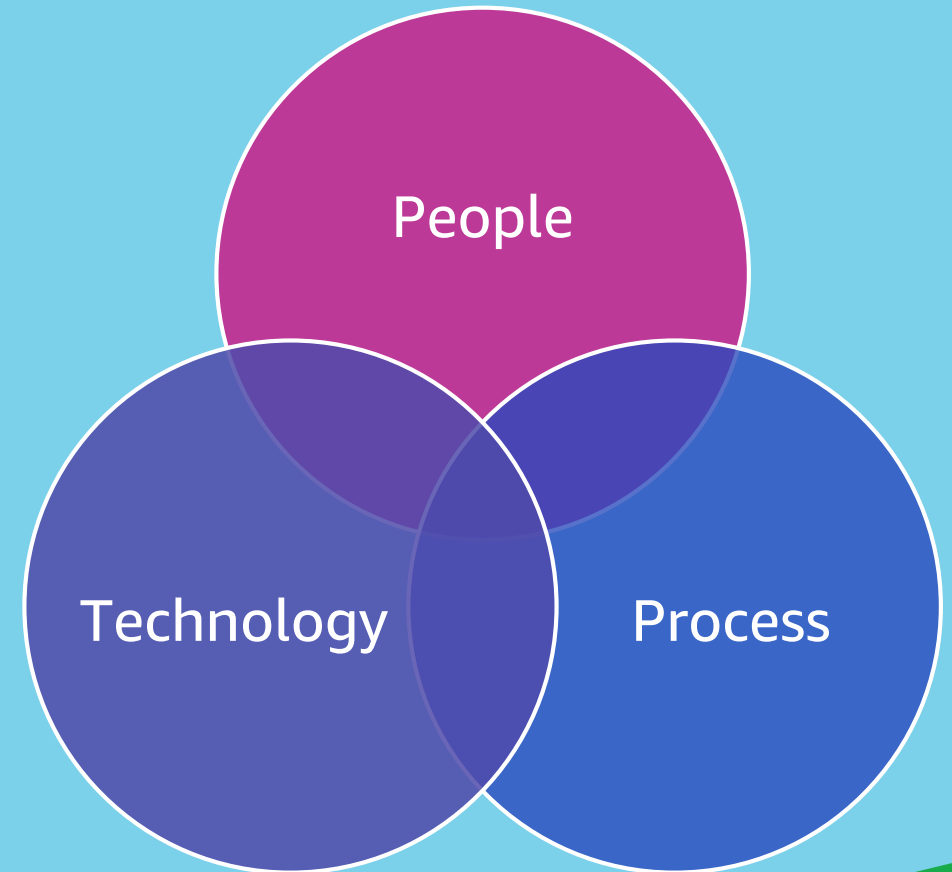
MLOps: High-level view

Intersection of Domains with Enabling Practices

Domains

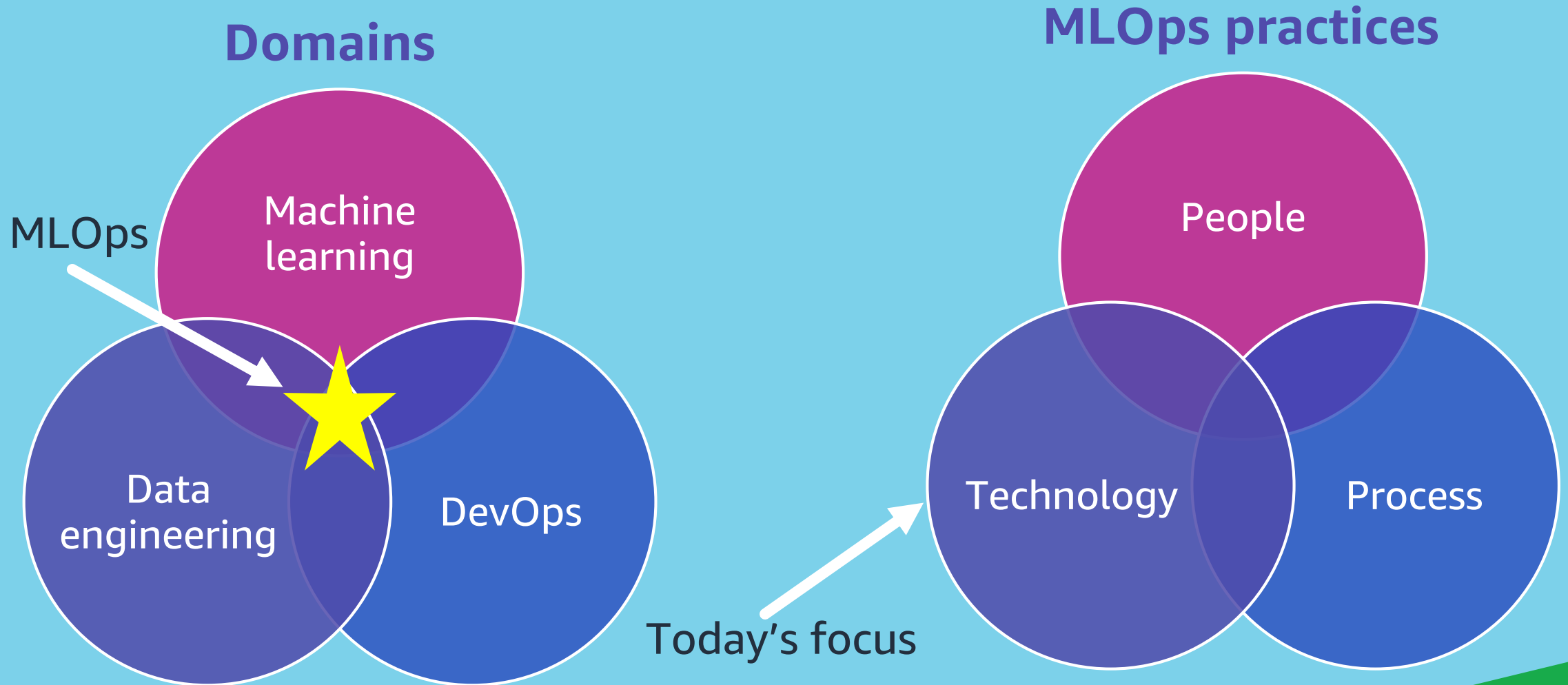


MLOps practices



MLOps: High-level view

Intersection of Domains with Enabling Practices



MLOps: The golden pipeline

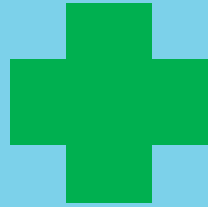
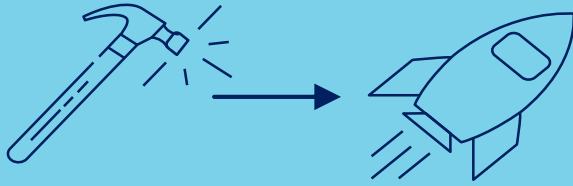


While the technical implementation can vary, the core practices remain the same.

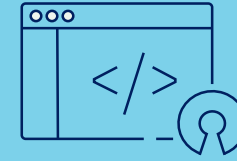
Machine learning pipelines

Technical Implementations: Services & Tooling

Purpose-built services for
automating machine learning tasks



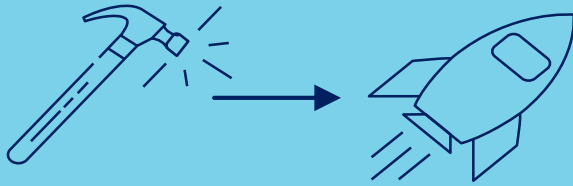
Purpose-built services for incorporating
everything-as-code and **CI/CD practices**



Machine learning pipelines

Technical Implementations: Services & Tooling

Purpose-built services for
automating machine learning tasks



AWS-native examples:



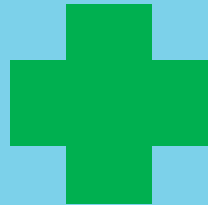
Amazon SageMaker
Pipelines



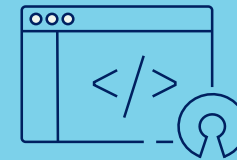
Amazon Managed
Workflows
for Apache Airflow
(Amazon MWAA)



AWS
Step Functions



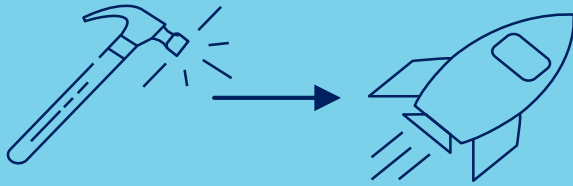
Purpose-built services for incorporating
everything-as-code and **CI/CD practices**



Machine learning pipelines

Technical Implementations: Services & Tooling

Purpose-built services for
automating machine learning tasks



AWS-native examples:



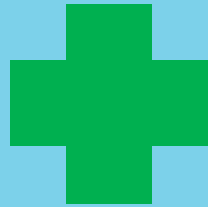
Amazon SageMaker
Pipelines



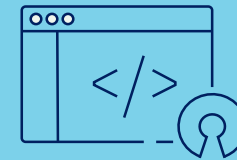
Amazon Managed
Workflows
for Apache Airflow
(Amazon MWAA)



AWS
Step Functions



Purpose-built services for incorporating
everything-as-code and **CI/CD practices**



AWS-native examples:



Amazon
SageMaker
Projects



AWS CodeBuild



AWS CodePipeline



AWS CodeCommit

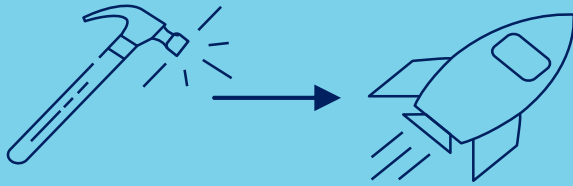


AWS CloudFormation

Machine learning pipelines

Technical Implementations: Services & Tooling

Purpose-built services for
automating machine learning tasks



AWS-native examples:



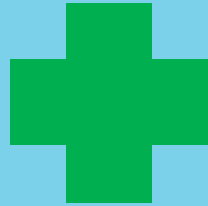
Amazon SageMaker
Pipelines



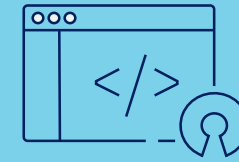
Amazon Managed
Workflows
for Apache Airflow
(Amazon MWAA)



AWS
Step Functions



Purpose-built services for incorporating
everything-as-code and **CI/CD practices**



AWS-native examples:



Amazon
SageMaker
Projects



AWS CodeBuild



AWS CodePipeline



AWS CodeCommit



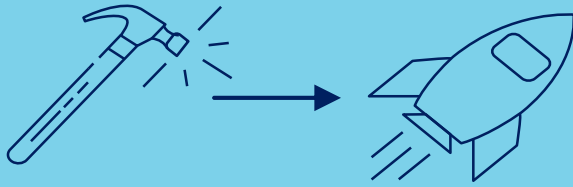
AWS CloudFormation

Do-it-yourself, 3rd Party & Open Source

Machine learning pipelines

Technical Implementations: Services & Tooling

Purpose-built services for
automating machine learning tasks



AWS-native examples:



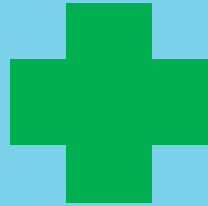
Amazon SageMaker
Pipelines



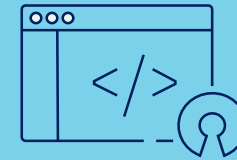
Amazon Managed
Workflows
for Apache Airflow
(Amazon MWAA)



AWS
Step Functions



Purpose-built services for incorporating
everything-as-code and **CI/CD practices**



AWS-native examples:



Amazon
SageMaker
Projects



AWS CodeBuild



AWS CodePipeline



AWS CodeCommit



AWS CloudFormation

Do-it-yourself, 3rd Party & Open Source

SageMaker MLOps-Ready Capabilities



MLOps: Core practices & benefits

Common challenges

Enabling innovation
while balancing
governance needs

MLOps practice

Everything as Code
(DevOps)

Benefit

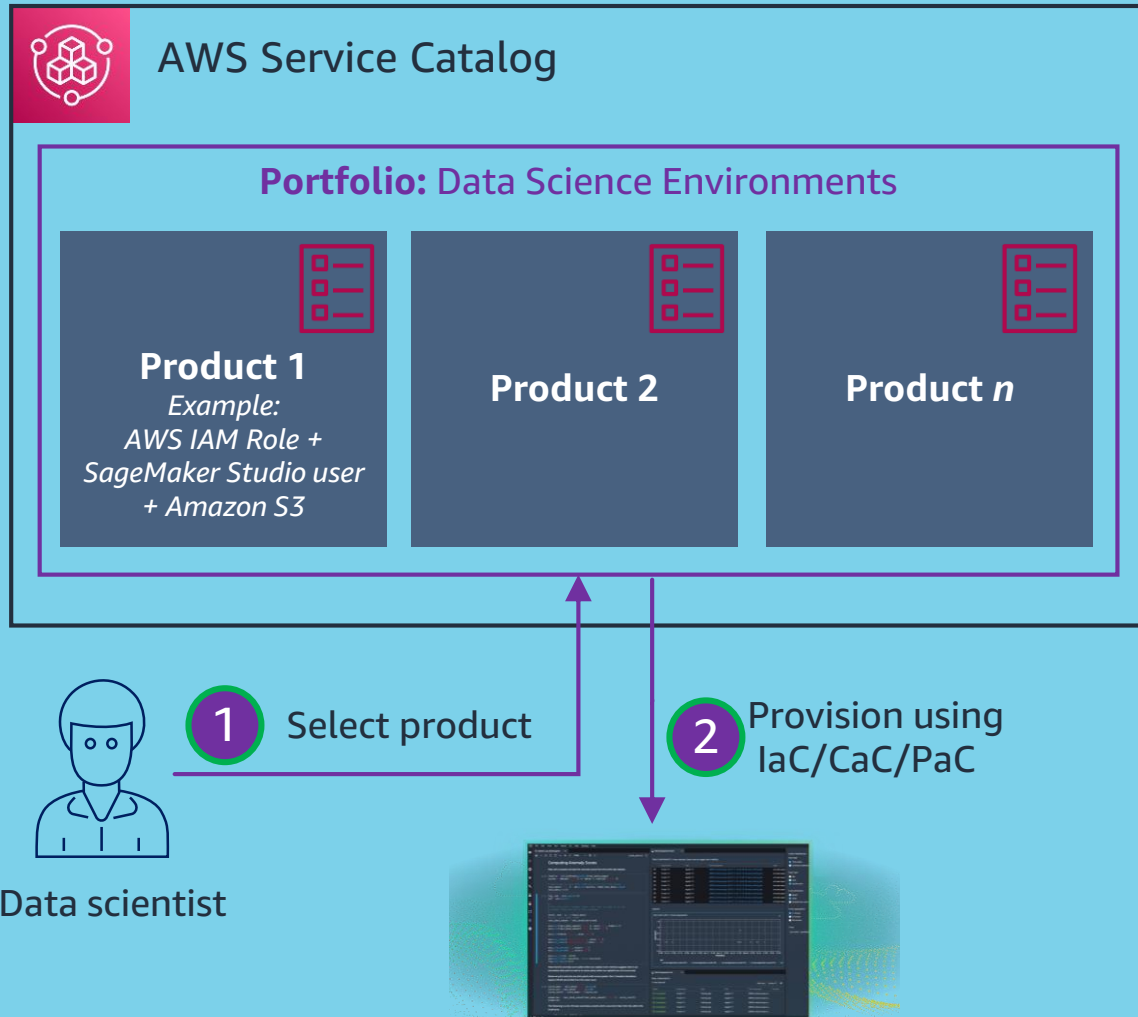
Self-service governed data
science environments

Solution

AWS
Service Catalog

Data science environments

Challenge: Enabling innovation while balancing governance needs

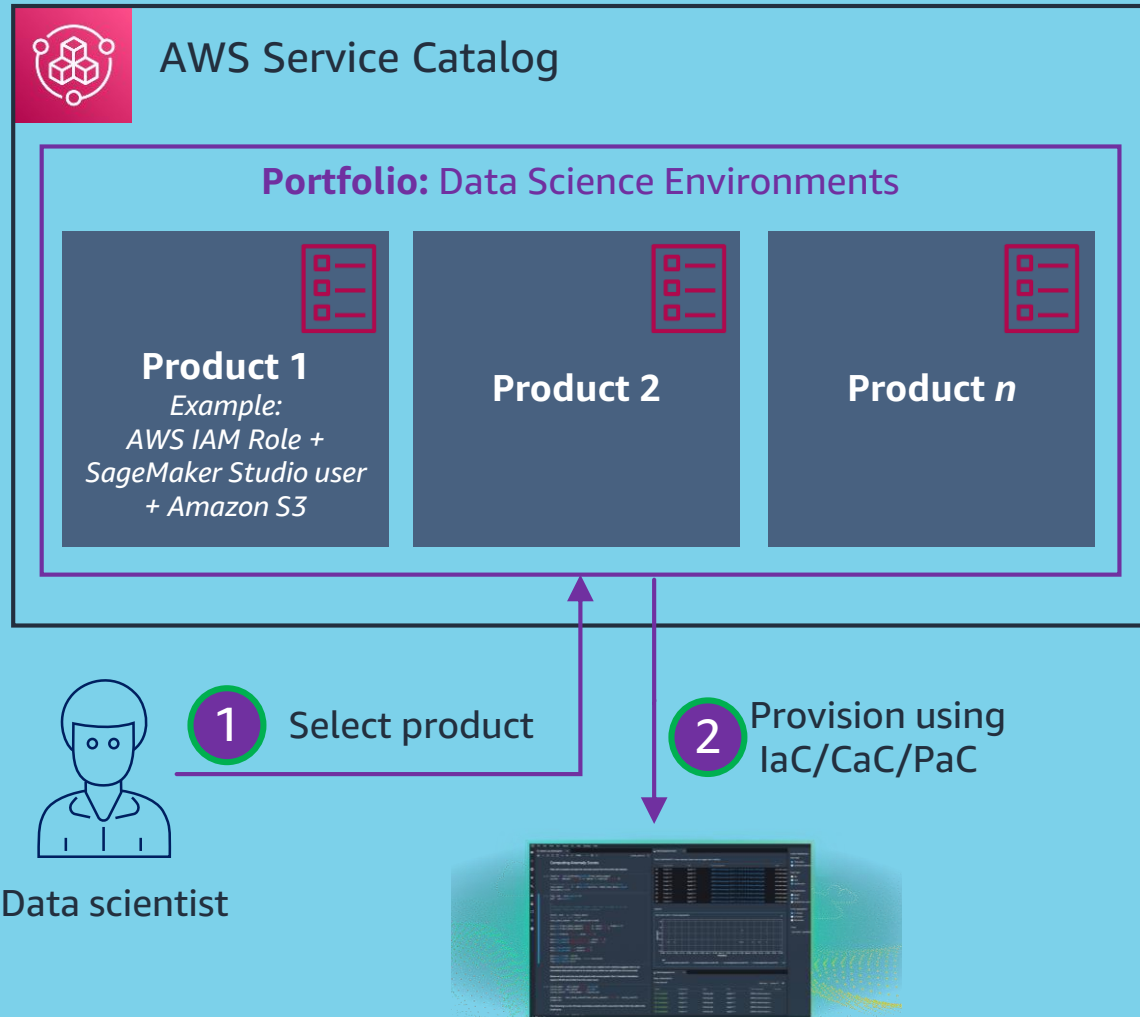


Solution

Utilize AWS Service Catalog to provision environments using infrastructure, configuration, or policy as code

Data science environments

Challenge: Enabling innovation while balancing governance needs



Solution

Utilize AWS Service Catalog to provision environments using infrastructure, configuration, or policy as code

Benefits:

- Ensure repeatability and consistency
- Standardize data science environments
- Implement guardrails via code

MLOps: Core practices & benefits

Common challenges

Enabling innovation
while balancing
governance needs

Bridging the gap between
model build and model
deploy tasks

MLOps practice

Everything as Code
(DevOps)

Lineage tracking
and configuration
management

Benefit

Self-service governed data
science environments

Repeatable process to
create workflows
across personas

Solution

AWS
Service Catalog

Amazon SageMaker
Model Registry

MLOps: Core practices & benefits

Common challenges

Enabling innovation while balancing governance needs

Bridging the gap between model build and model deploy tasks

Providing end-to-end traceability

MLOps practice

Everything as Code (DevOps)

Lineage tracking and configuration management

CI/CD

Benefit

Self-service governed data science environments

Repeatable process to create workflows across personas

Auditable ML pipeline

Solution

AWS Service Catalog

Amazon SageMaker Model Registry

Amazon SageMaker projects

MLOps: Core practices & benefits

Common challenges

Enabling innovation while balancing governance needs

Bridging the gap between model build and model deploy tasks

Providing end-to-end traceability

Building a repeatable and consistent process for deploying models

MLOps practice

Everything as Code (DevOps)

Lineage tracking and configuration management

CI/CD

Continuous delivery and deployment

Benefit

Self-service governed data science environments

Repeatable process to create workflows across personas

Auditable ML pipeline

Improve time-to-market

Solution

AWS Service Catalog

Amazon SageMaker Model Registry

Amazon SageMaker projects

Amazon SageMaker projects and Pipelines

MLOps: Core practices & benefits

| Common challenges | | | | |
|--|---|-----------------------------------|---|--------------------------------------|
| Enabling innovation while balancing governance needs | Bridging the gap between model build and model deploy tasks | Providing end-to-end traceability | Building a repeatable and consistent process for deploying models | Monitoring models |
| MLOps practice | | | | |
| Everything as Code (DevOps) | Lineage tracking and configuration management | CI/CD | Continuous delivery and deployment | Continuous monitoring |
| Benefit | | | | |
| Self-service governed data science environments | Repeatable process to create workflows across personas | Auditable ML pipeline | Improve time-to-market | Maintain model performance over time |
| Solution | | | | |
| AWS Service Catalog | Amazon SageMaker Model Registry | Amazon SageMaker projects | Amazon SageMaker projects and Pipelines | Amazon SageMaker Model Monitor |

MLOps-ready features and capabilities

Data preparation

Prepare data

SageMaker
Data Wrangler

SageMaker
Processing Job

MLOps-ready features and capabilities

Data preparation

Prepare data

SageMaker
Data Wrangler

SageMaker
Processing Job

Curate features

SageMaker
Feature Store

MLOps-ready features and capabilities

Data preparation

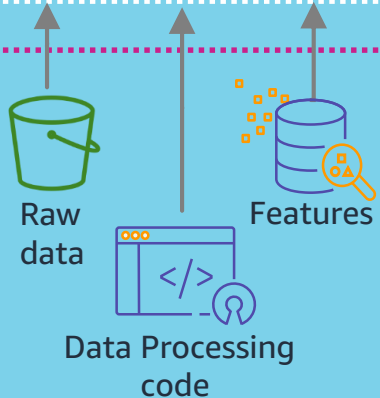
Prepare data

SageMaker
Data Wrangler

SageMaker
Processing Job

Curate features

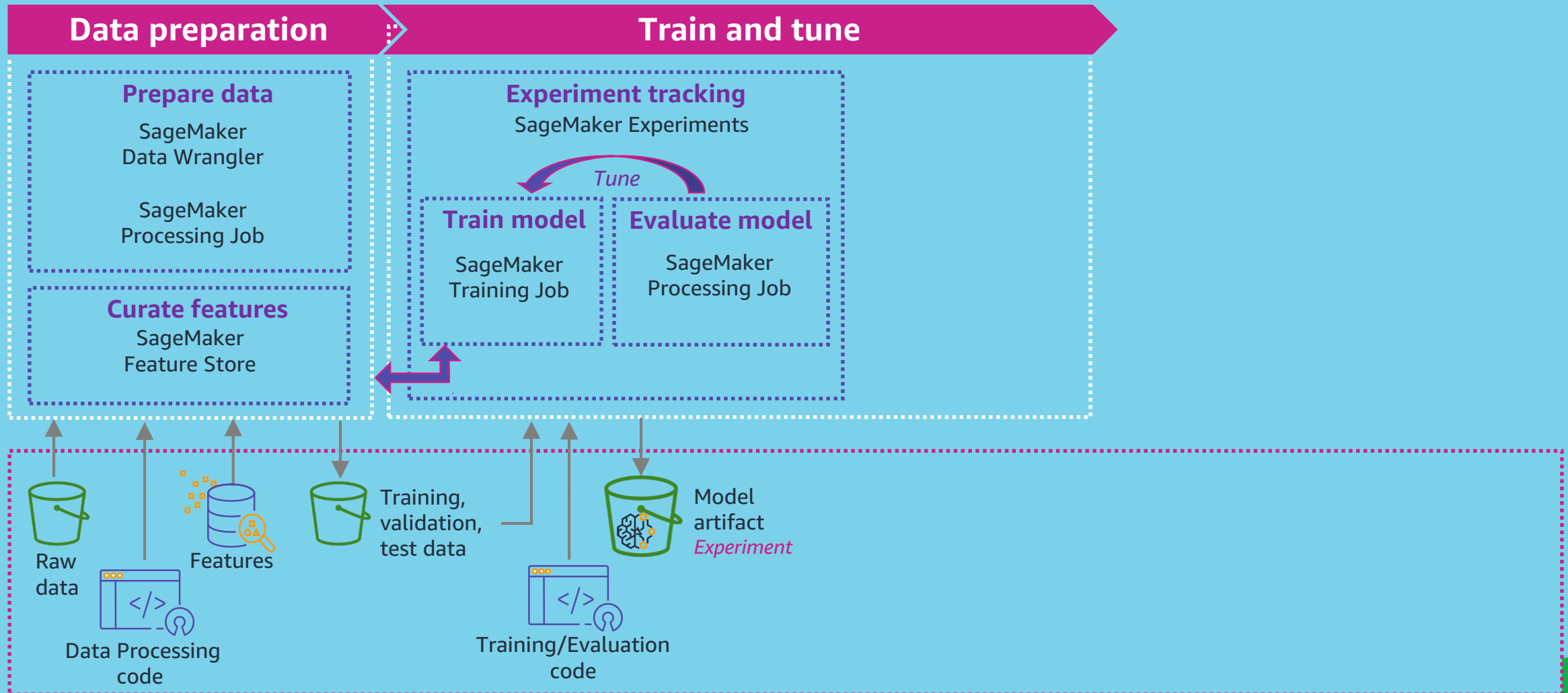
SageMaker
Feature Store



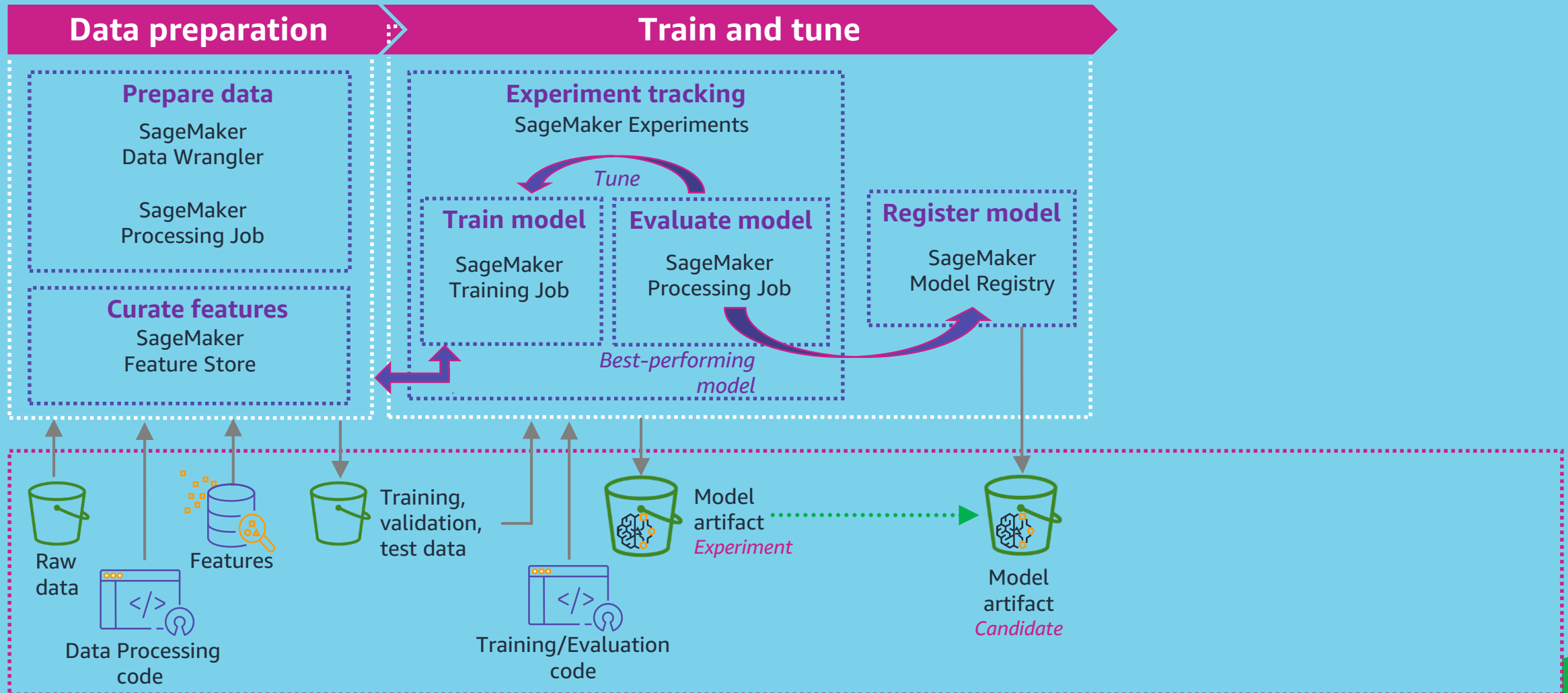
MLOps-ready features and capabilities



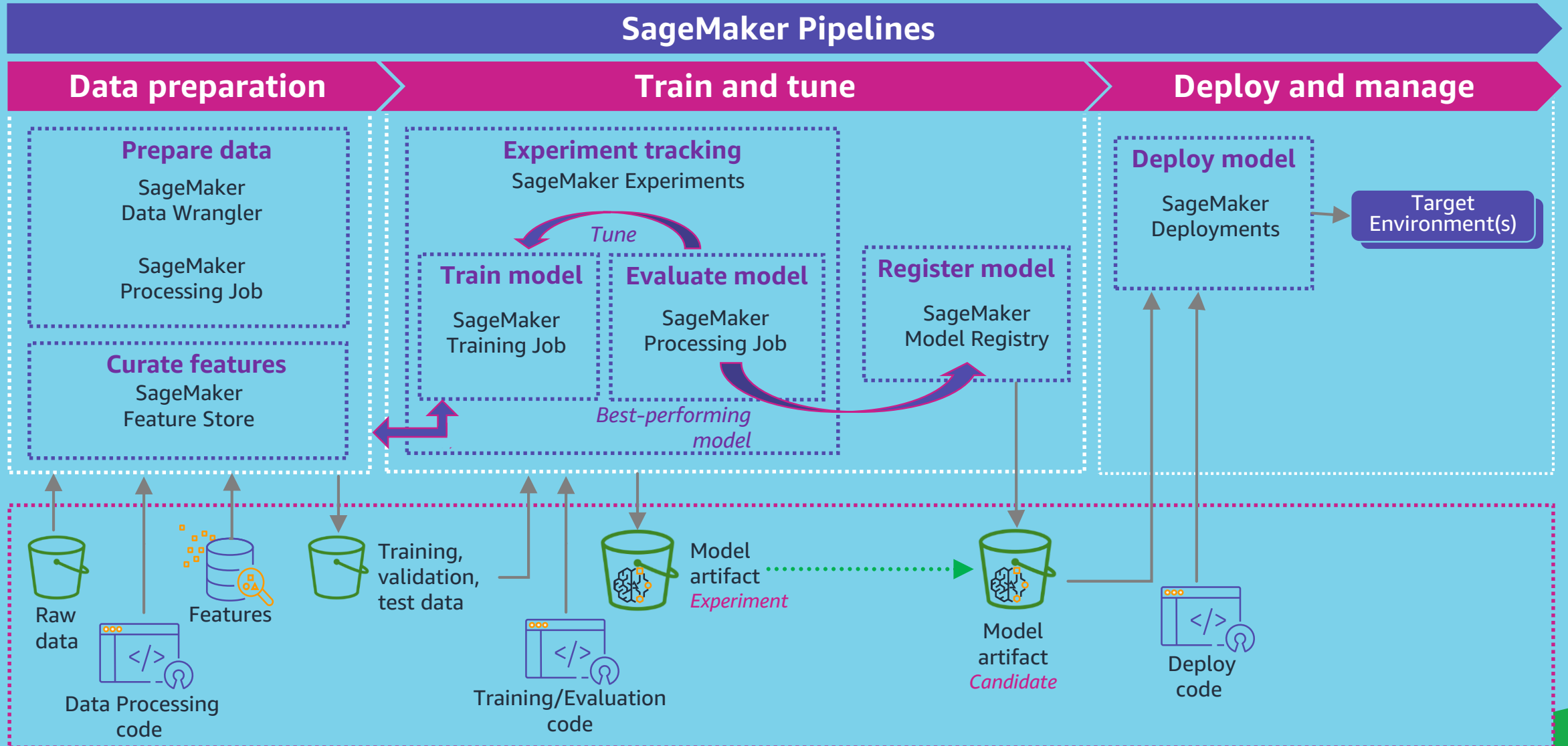
MLOps-ready features and capabilities



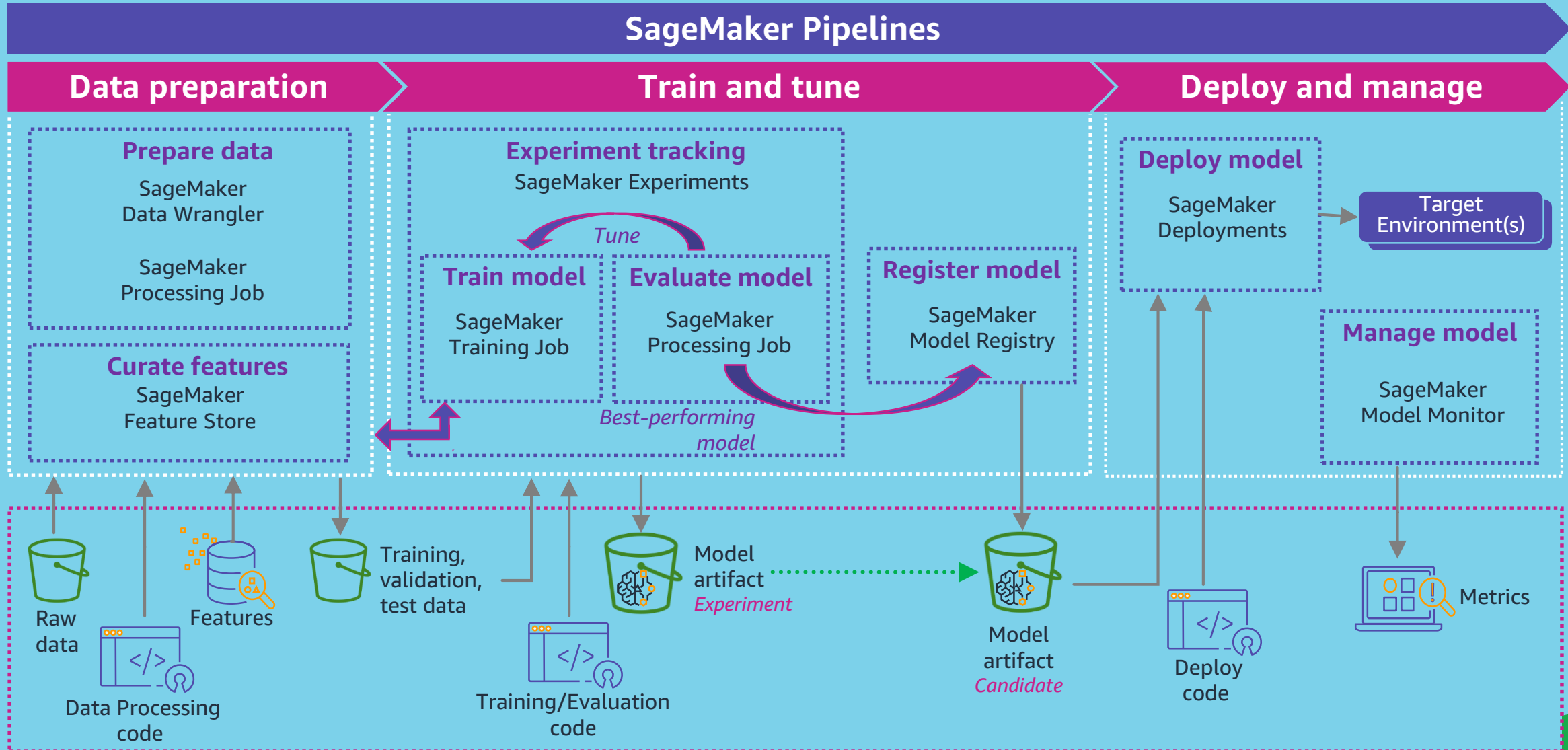
MLOps-ready features and capabilities



MLOps-ready features and capabilities



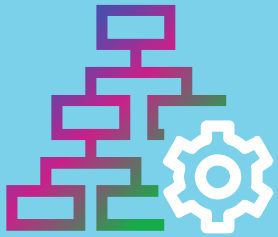
MLOps-ready features and capabilities



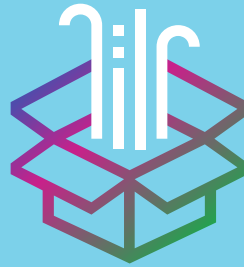
SageMaker Pipelines



SageMaker Pipelines



Python SDK for quickly
and easily **building ML
workflows**

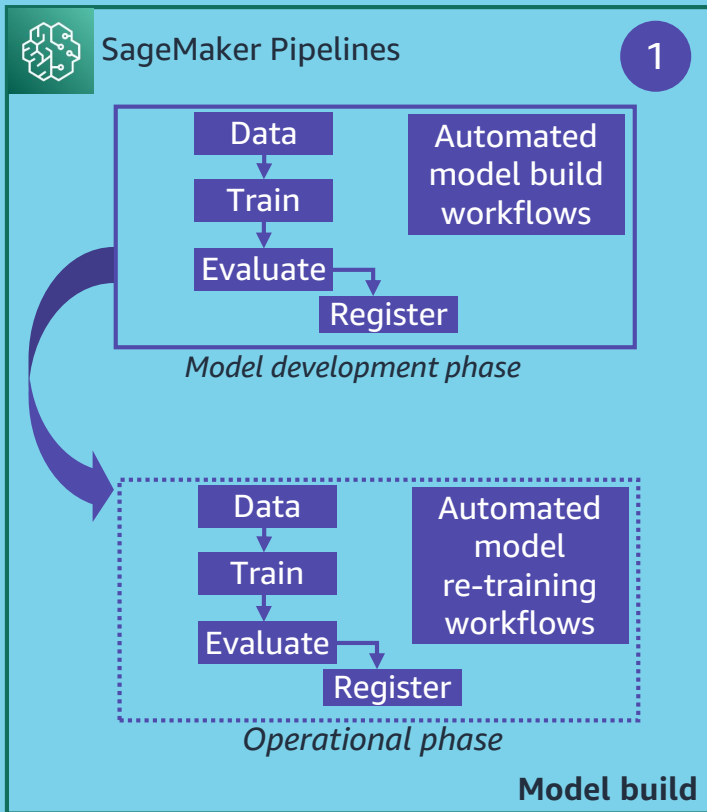


Catalog models to
manage models at scale
and trigger automated
deployment workflows

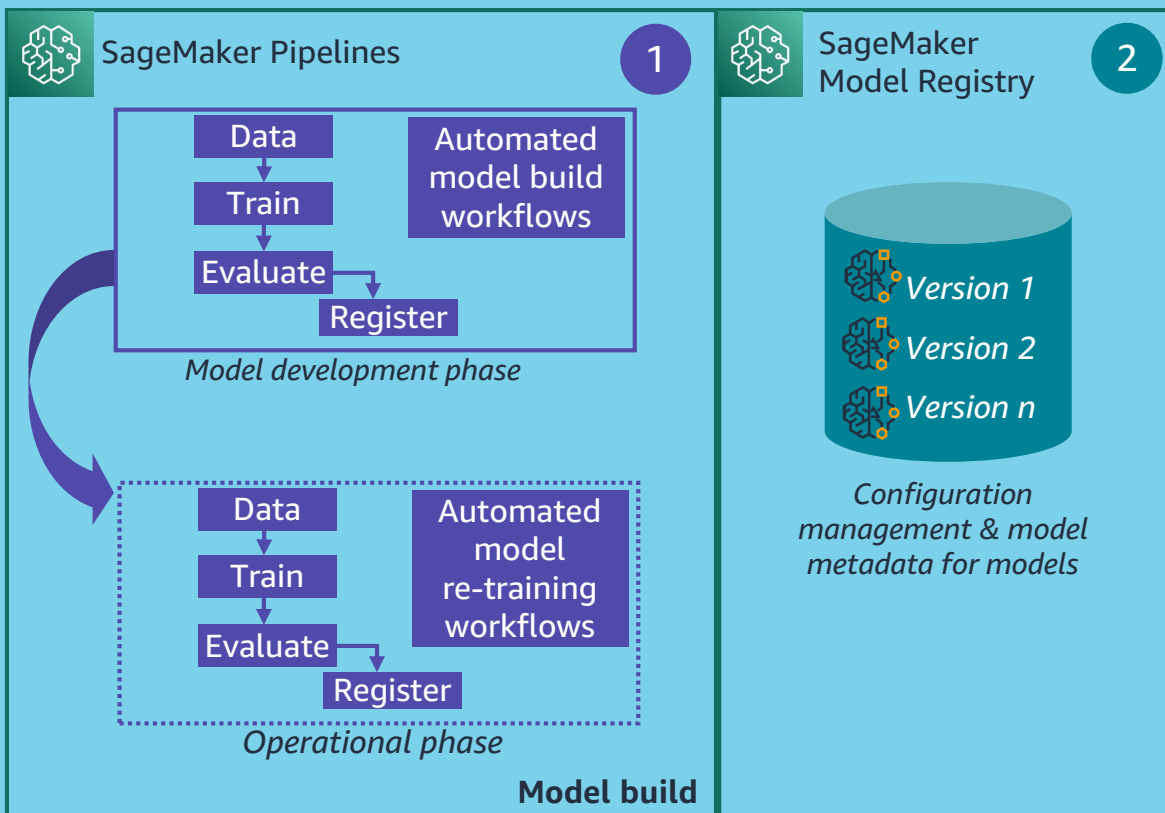


Built-in support for
CI/CD and end-to-end
lineage tracking

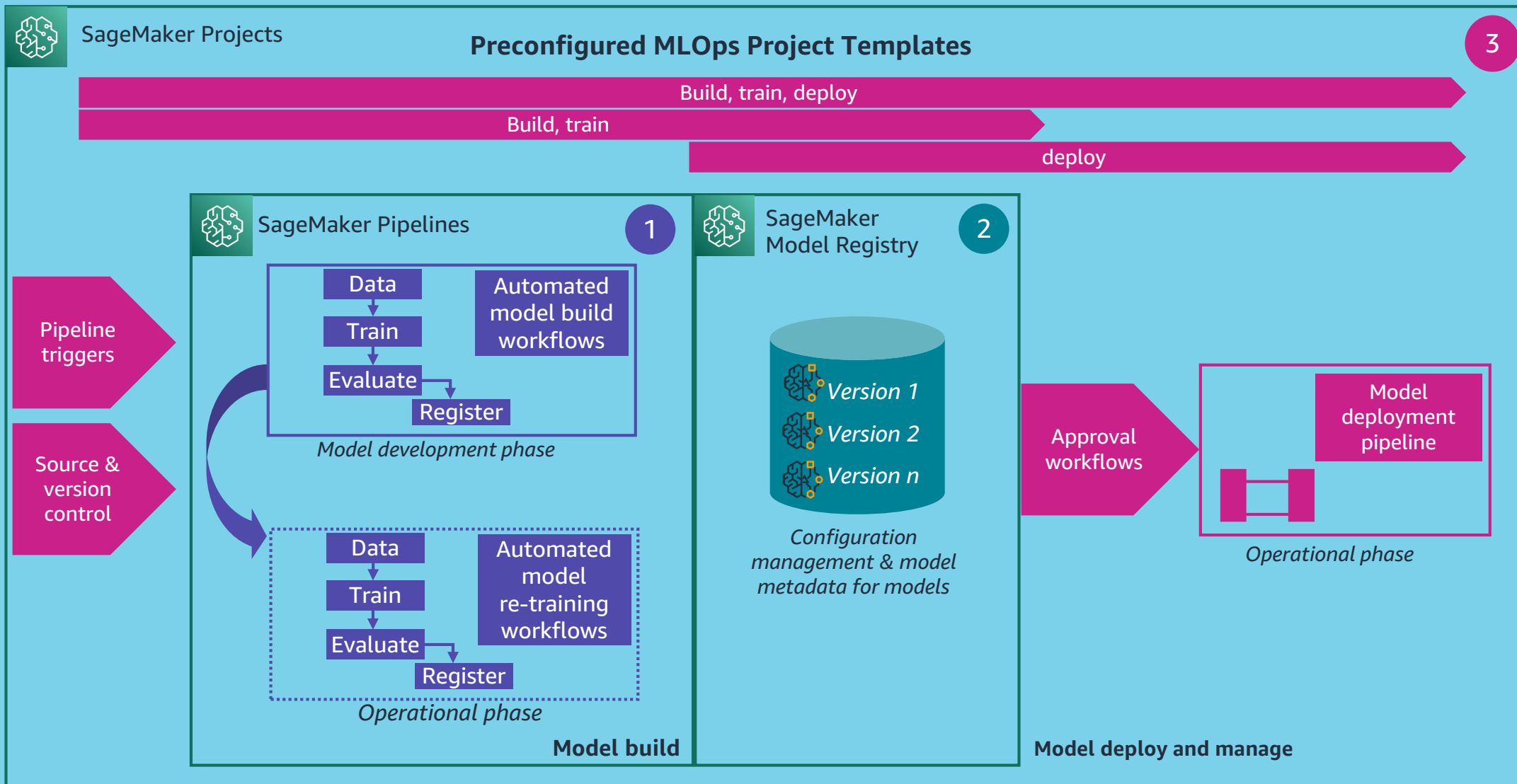
SageMaker Pipelines: components



SageMaker Pipelines: components



SageMaker Pipelines: components



SageMaker Pipelines - Demo

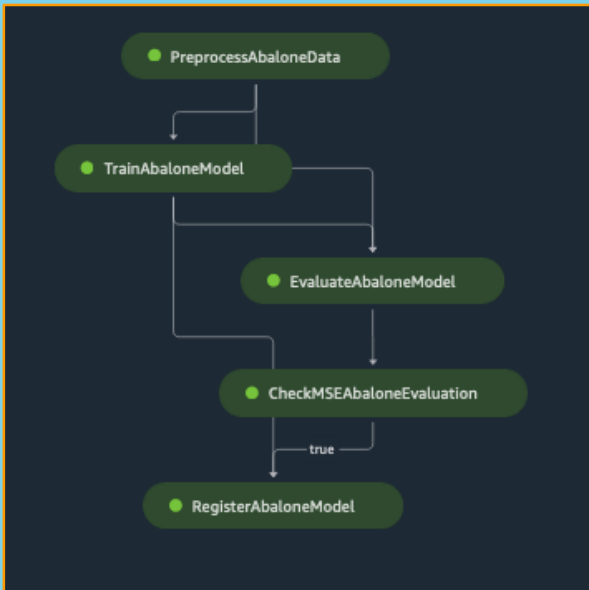


<<Insert Recorded Demo>>

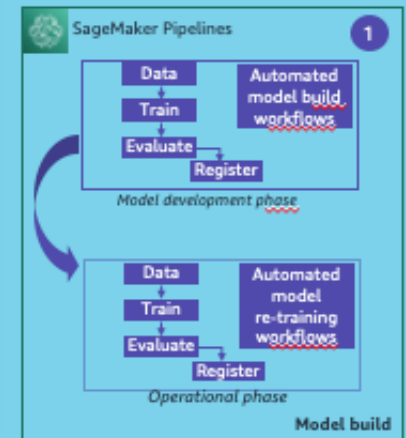
SageMaker Pipelines: Model build pipelines

High Level Features

Built-in steps to support model build activities

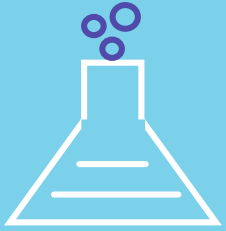


- **SageMaker native:** Processing, Training, Tuning, Create Model, Register Model, Batch Transform, Clarify Check, Quality Check
- **AWS services:** Lambda, EMR
- **Extend to other AWS services & tasks using Callback step**
- **Conditional steps to assess next pipeline steps**



SageMaker Pipelines: Model build pipelines

High Level Features



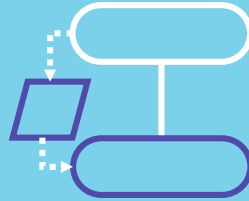
Integration with SageMaker Experiments

Automatically track pipeline executions – no setup required!



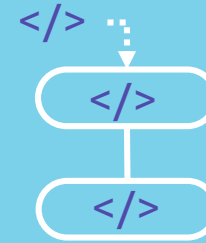
Pipeline step retry policies

Retry failed steps in the pipeline without re-running the full pipeline



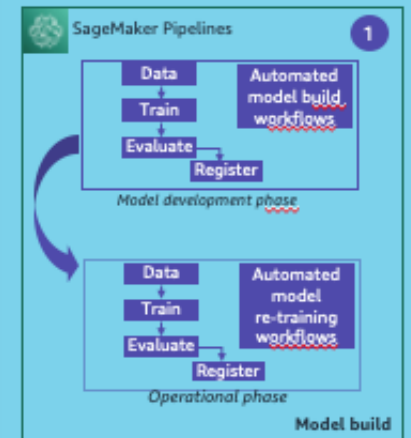
Step caching

Enable step caching to avoid unnecessary runs of pipeline steps



Pipeline parameterization

Use pipeline parameters to define configurations specific to a pipeline execution



SageMaker Pipelines: Model Registry

High Level Features

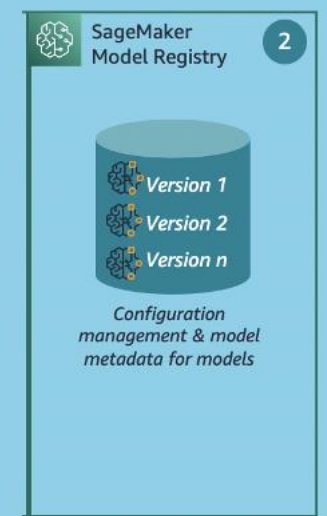
Versions

Settings

Search column name to start

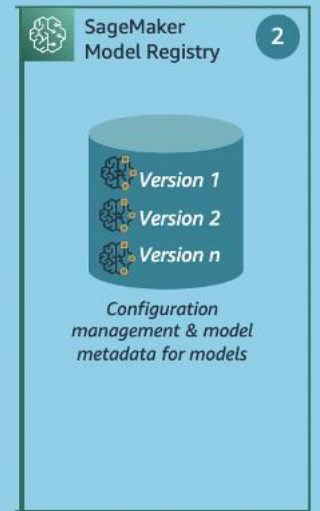
| Version | Stage | Status | Short description | Modified by | Last modified |
|---------|---------|----------|-------------------|---------------|---------------|
| 3 | None | Pending | | | |
| 2 | prod | Approved | | workshop-user | 1 day ago |
| 1 | staging | Approved | | shelbee-iggy | 2 months ago |

- Manage model versions & associated metadata
- Manage the approval status of a model
- Deploy models to production (with Projects)
- Support for cross-account model registration & deployments



SageMaker Pipelines: Model Registry

High Level Features



| Versions | | | | | | Settings |
|--|---------|----------|-------------------|---------------|---------------|----------|
| <input type="text" value="Search column name to start"/> | | | | | | |
| Version | Stage | Status | Short description | Modified by | Last modified | |
| 3 | None | Pending | | | | |
| 2 | prod | Approved | | workshop-user | 1 day ago | |
| 1 | staging | Approved | | shelbee-iggy | 2 months ago | |

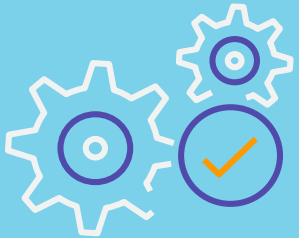
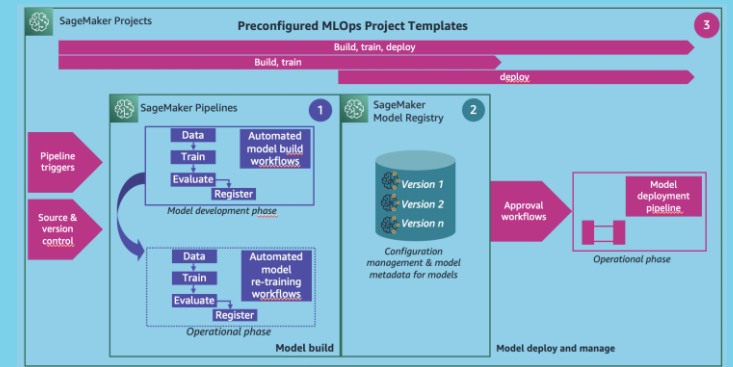
- Manage model versions & associated metadata
- Manage the approval status of a model
- Deploy models to production (with Projects)
- Support for cross-account model registration & deployments

| Version 2 | | | | | | Update status |
|---------------------------|-------------------------|-------------------|------------|-------------------------|--------------------|---------------|
| Status | Pipeline | Execution | Last Stage | Model group | | |
| Approved | shelbee-demoagain-p-... | workflow-2 | prod | shelbee-demoagain-p-... | | |
| Activity Metrics Settings | | | | | | |
| Model metrics | | | | | | |
| Model metric | | Metric value | | | Standard deviation | |
| mse | | 4.823176167079859 | | | 2.1960237865043672 | |

- Track model performance metrics

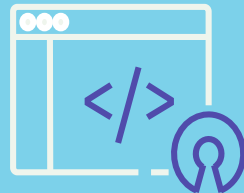
SageMaker Pipelines: Projects

High Level Features



Enable CI/CD practices

*source/version control,
pipeline traceability, and
automated triggers for
training & deployment*



Support for third- party tools

*Including: GitHub, GitHub
Enterprise, BitBucket
Cloud, and Jenkins*



Customizable MLOps Project templates

*Utilize built-in templates
or build custom templates
to meet your requirements*



Automatic rollback to previous model version

*Rollback deployment to
previous version of model*

Wrap-up



SageMaker MLOps summary



Start small
and iterate

SageMaker MLOps summary

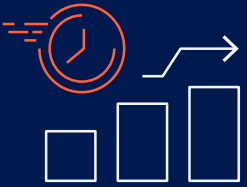


Start small
and iterate

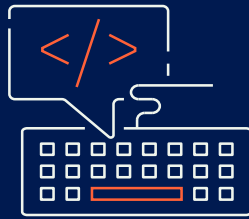


Practice
Everything-as-Code
for repeatability,
consistency, and
governance

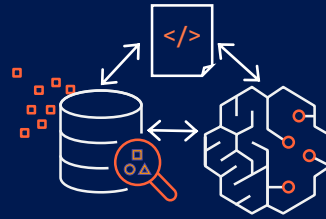
SageMaker MLOps summary



Start small
and iterate

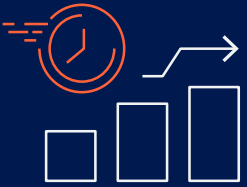


Practice
Everything-as-Code
for repeatability,
consistency, and
governance

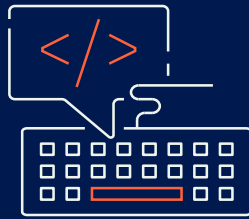


Consider ML
specific inputs,
outputs, and artifacts
for model and pipeline
lineage

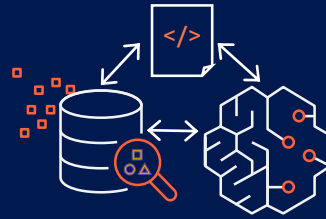
SageMaker MLOps summary



Start small
and iterate



Practice
Everything-as-Code
for repeatability,
consistency, and
governance

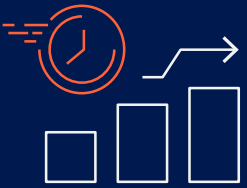


Consider ML
specific inputs,
outputs, and artifacts
for model and pipeline
lineage

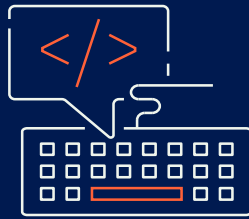


Use integrated
SageMaker services
when possible to
reduce management
and setup time

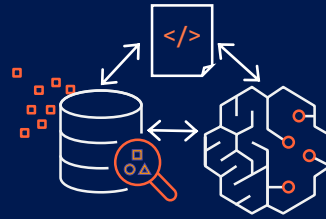
SageMaker MLOps summary



Start small
and iterate



Practice
Everything-as-Code
for repeatability,
consistency, and
governance



Consider ML
specific inputs,
outputs, and artifacts
for model and pipeline
lineage



Use integrated
SageMaker services
when possible to
reduce management
and setup time

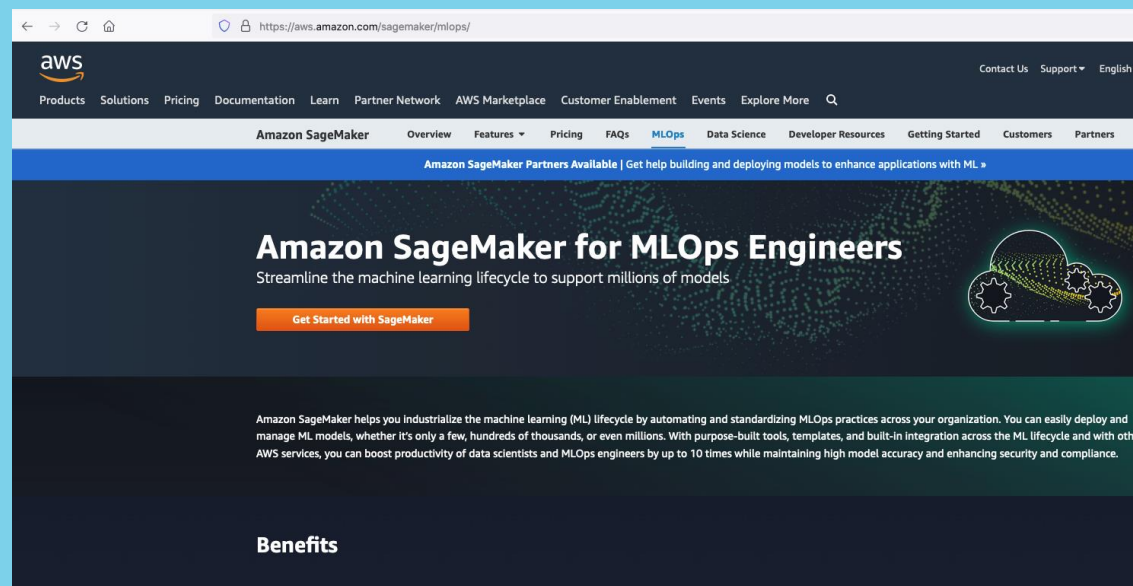


Implement a Model
Registry as part of
your ML pipeline for
managing models

SageMaker MLOps: Resources & Next Steps

Check out the new MLOps Page

<https://aws.amazon.com/sagemaker/mlops/>



Additional Resources

1. SageMaker Documentation: <https://docs.aws.amazon.com/sagemaker/latest/dg/pipelines.html>
2. Custom Project Template Examples: <https://github.com/aws-samples/sagemaker-custom-project-templates>



Thank you!