An Introduction To SageMaker Pipelines

How To Get Started With Your MLOps Journey on Amazon SageMaker

[Ram Vegiraju](#)
Published in

Towards Data Science

·
6 min read
·
Mar 14, 2022
23

Image from Unsplash by JJ Ying
MLOps is a critical avenue that is often overlooked on the path to production. It's easy to isolate a singular model that you want to train and deploy. In reality, there's going to be a multitude of models that you will work with. You need to make sure you have the proper infrastructure to manage training, model tracking, and deployment. MLOps helps build and simplify these processes into a clear workflow that is tailored for Machine Learning.

SageMaker Pipelines is the premier MLOps feature that Amazon SageMaker supports. Using Pipelines you can create and manage end to end ML workflows at a large scale. With today's article we'll take an introductory look into how you can setup SageMaker Pipelines.

NOTE: For those of you new to AWS, make sure you make an account at the following **link** if you want to follow along. **This article will**

assume a novice to intermediate level of knowledge with AWS and SageMaker.

Table of Contents

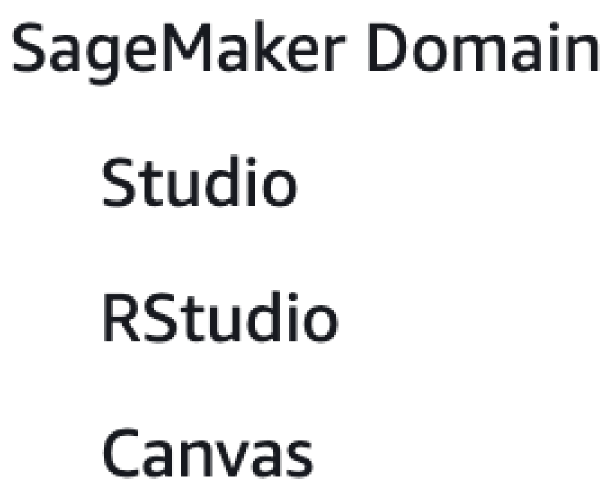
1. How Does SageMaker Pipelines Work?

At a high level a SageMaker Pipeline is built through a series of orchestrated **steps**. You can define different step types such as a **Training Step** or a **RegisterModel Step**. Using the SageMaker SDK you can build these steps out in Python and once executed a visual DAG is created that displays your workflow.

The other key portion of a SageMaker Pipeline is **Pipeline Parameters**. Through parameters you can inject variables into your Pipeline. By variables we can range from Instance Type to Instance Count to name a few. Using these you can call these parameters at different parts of your workflow.

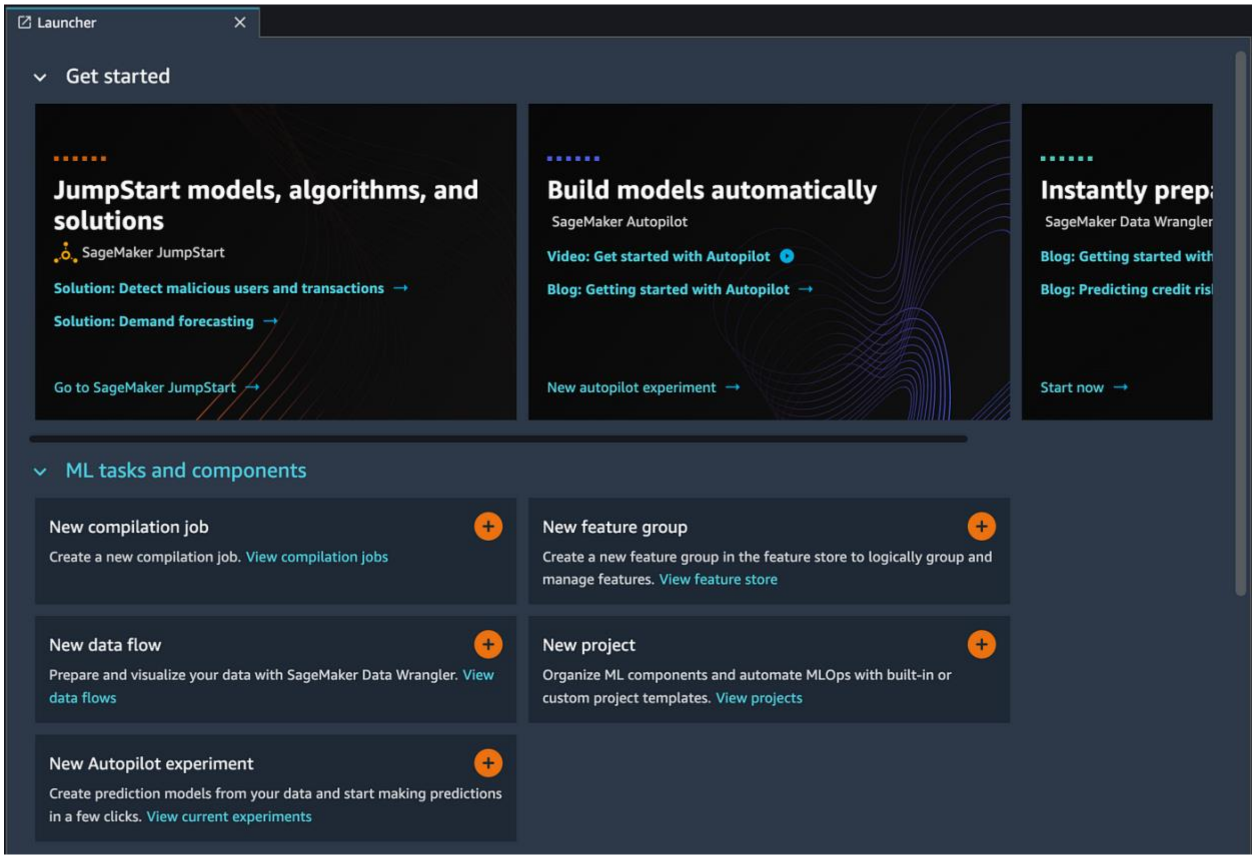
2. Setup

SageMaker Pipelines is most efficiently orchestrated from SageMaker Studio. SageMaker Studio is an IDE provided by SageMaker, here you can work in an environment that is very similar to JupyterLab but powered with all SageMaker capabilities. You should be able to access Studio on the left side of the SageMaker Console.



SageMaker Studio (Screenshot by Author)

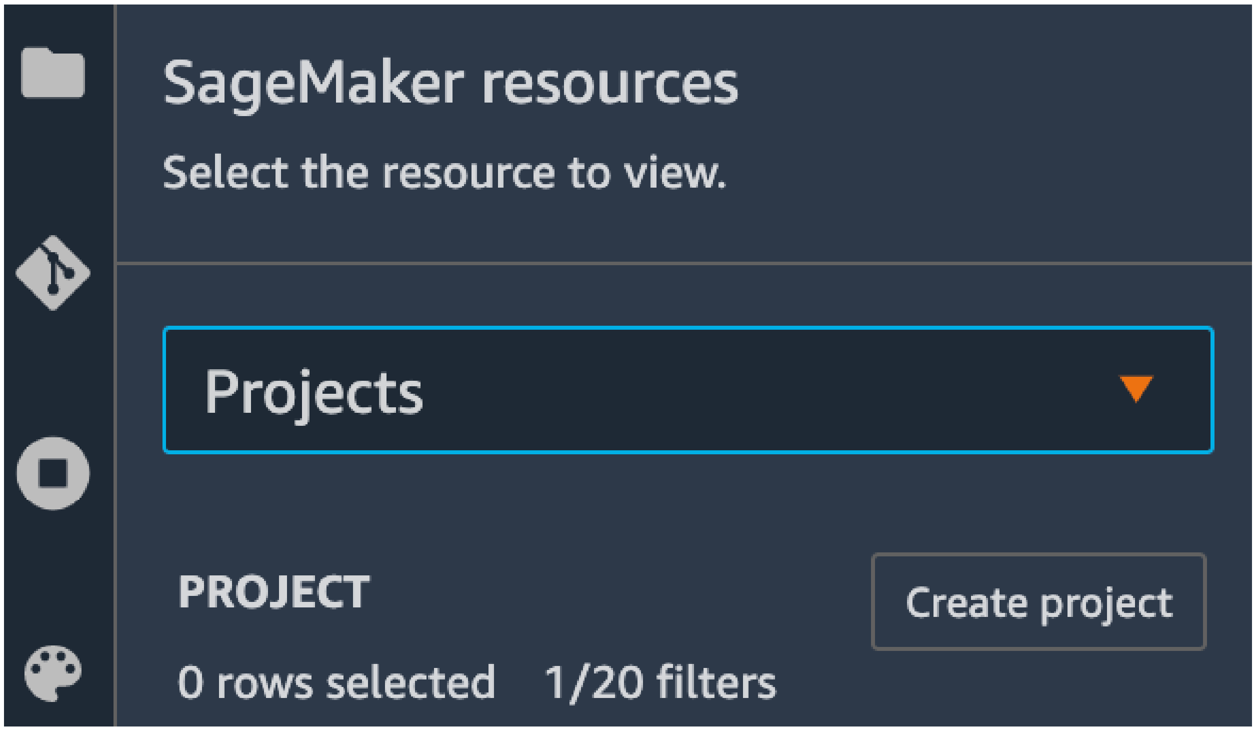
Once on the Studio Console you should be able to click on **Add User**, where you can create a SageMaker Domain. After you've clicked on the User that you have created you should be able to launch a Studio App. Here you should be able to see the general setup for where you can get started with working on Studio.



Launcher
Get started
JumpStart models, algorithms, and solutions
SageMaker JumpStart
Solution: Detect malicious users and transactions →
Solution: Demand forecasting →
Go to SageMaker JumpStart →
Build models automatically
SageMaker Autopilot
Video: Get started with Autopilot
Blog: Getting started with Autopilot →
New autopilot experiment →
Instantly prep
SageMaker Data Wrangler
Blog: Getting started with
Blog: Predicting credit ris
Start now →
ML tasks and components
New compilation job
Create a new compilation job. View compilation jobs
New feature group
Create a new feature group in the feature store to logically group and manage features. View feature store
New data flow
Prepare and visualize your data with SageMaker Data Wrangler. View data flows
New project
Organize ML components and automate MLOps with built-in or custom project templates. View projects
New Autopilot experiment
Create prediction models from your data and start making predictions in a few clicks. View current experiments

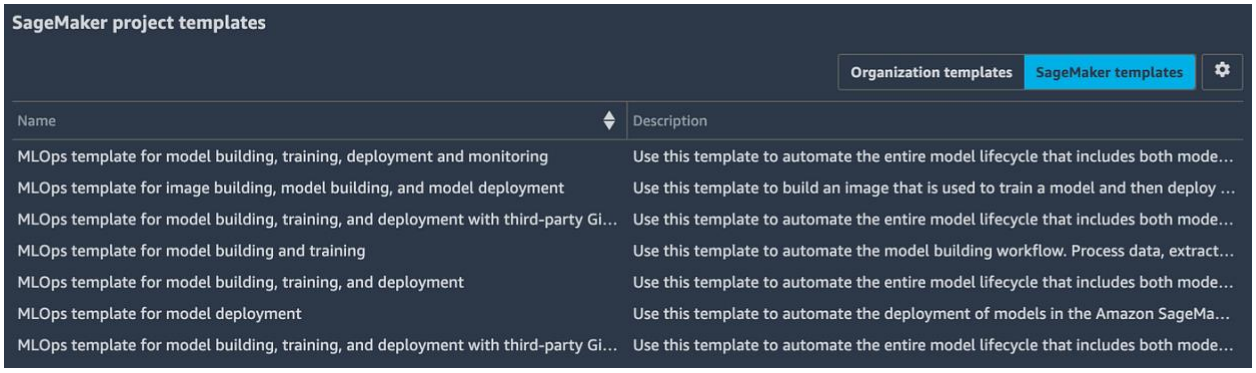
Studio (Screenshot by Author)

Now we could get to work immediately and start building a Pipeline from scratch using the SageMaker Python SDK. However **building a Pipeline from ground up can be a tedious** and a lot of the base setup can be automated. SageMaker Projects help provide **ready made templates** that you can alter and build upon for your own specific ML use cases. We can find Projects in the SageMaker Studio UI.



Creating a SageMaker Project (Screenshot by Author)

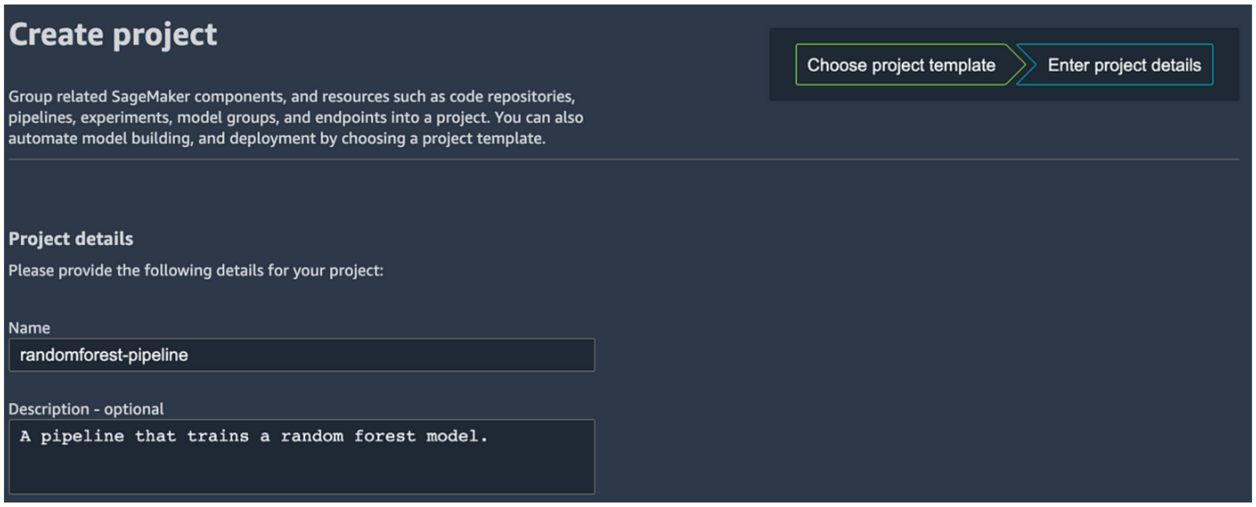
If we click **Create Project** we can see the different templates that are offered.



Name	Description
MLOps template for model building, training, deployment and monitoring	Use this template to automate the entire model lifecycle that includes both mode…
MLOps template for image building, model building, and model deployment	Use this template to build an image that is used to train a model and then deploy …
MLOps template for model building, training, and deployment with third-party Gi…	Use this template to automate the entire model lifecycle that includes both mode…
MLOps template for model building and training	Use this template to automate the model building workflow. Process data, extract…
MLOps template for model building, training, and deployment	Use this template to automate the entire model lifecycle that includes both mode…
MLOps template for model deployment	Use this template to automate the deployment of models in the Amazon SageMa…
MLOps template for model building, training, and deployment with third-party Gi…	Use this template to automate the entire model lifecycle that includes both mode…

Project Templates (Screenshot by Author)

Here we can take the simple model building and training template to get started. Once selected we can name and create the Project, this process should take a few minutes.



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.

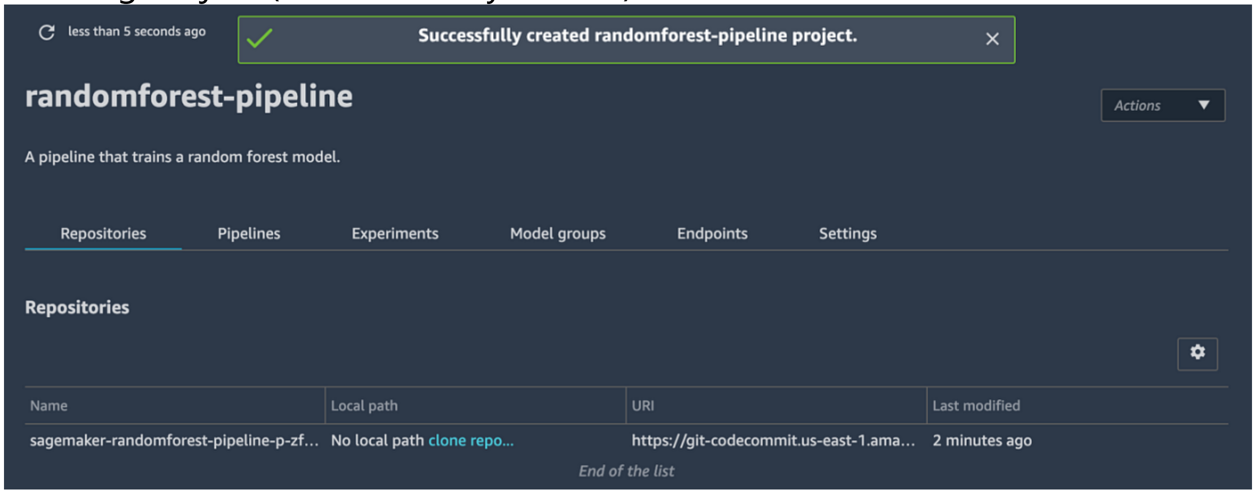
Project details

Please provide the following details for your project:

Name: randomforest-pipeline

Description - optional: A pipeline that trains a random forest model.

Creating Project (Screenshot by Author)



Successfully created randomforest-pipeline project.

randomforest-pipeline

A pipeline that trains a random forest model.

Repositories

Name	Local path	URI	Last modified
sagemaker-randomforest-pipeline-p-zf…	No local path clone repo…	https://git-codecommit.us-east-1.ama…	2 minutes ago

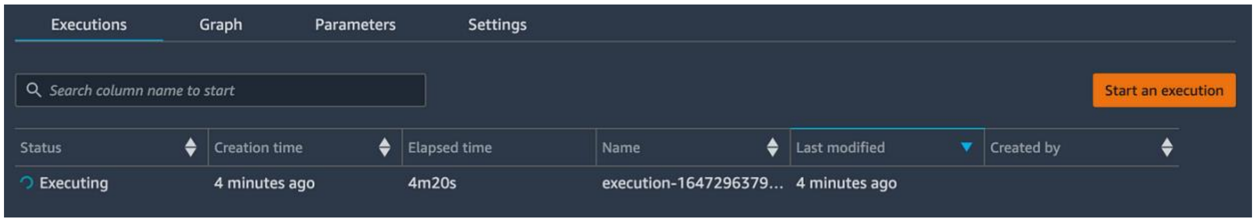
End of the list

Project Created (Screenshot by Author)

Now we can take a deeper look into what the Pipeline is actually composed off.

3. Visualizing SageMaker Pipelines

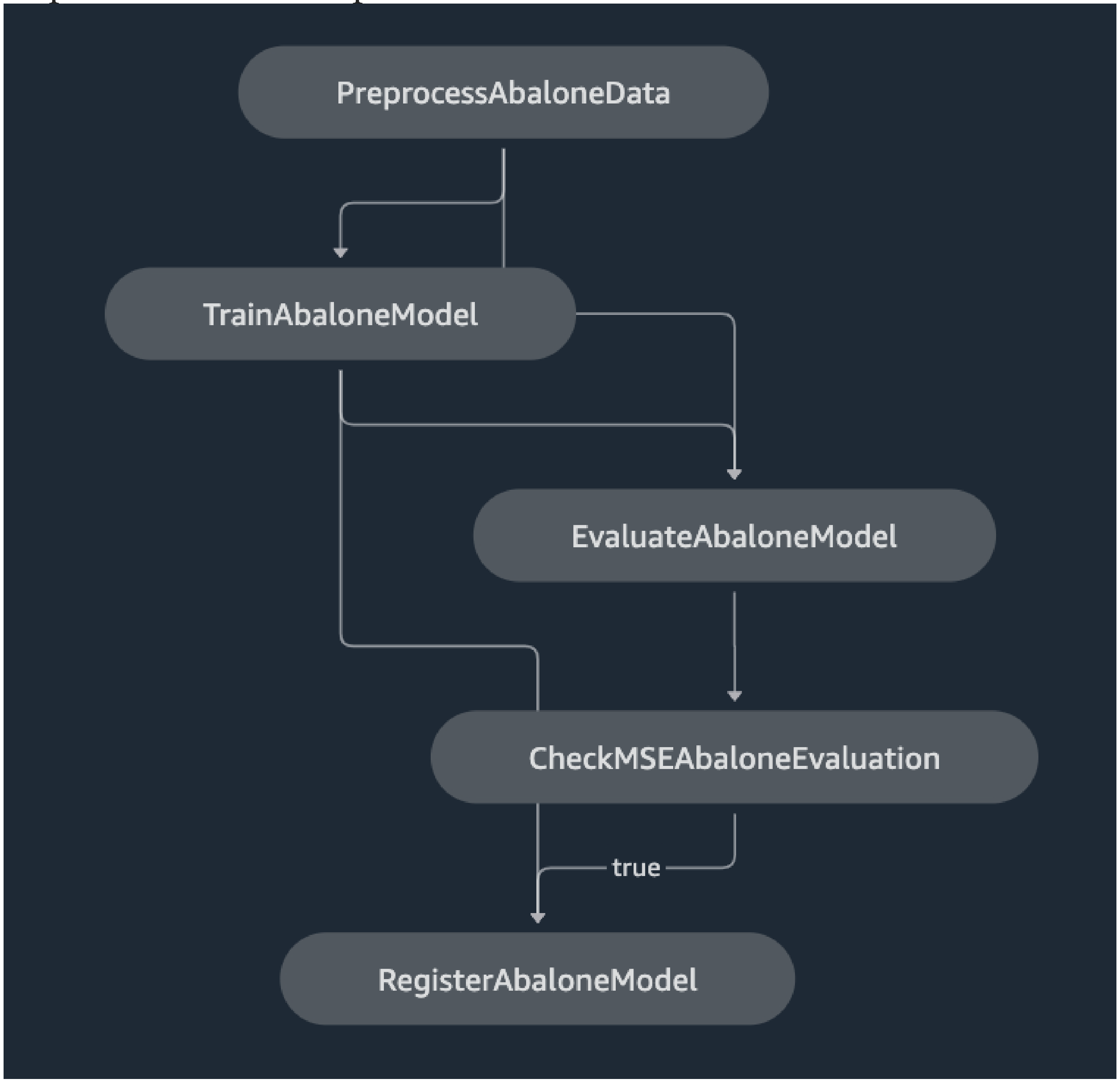
Right next to the Repositories tab you should be able to see a Pipelines tab. This is where we can see our Project template has created a pre-built Pipeline for us. If we click on the Pipeline we should be able to see the execution in process.



Status	Creation time	Elapsed time	Name	Last modified	Created by
Executing	4 minutes ago	4m20s	execution-1647296379...	4 minutes ago	

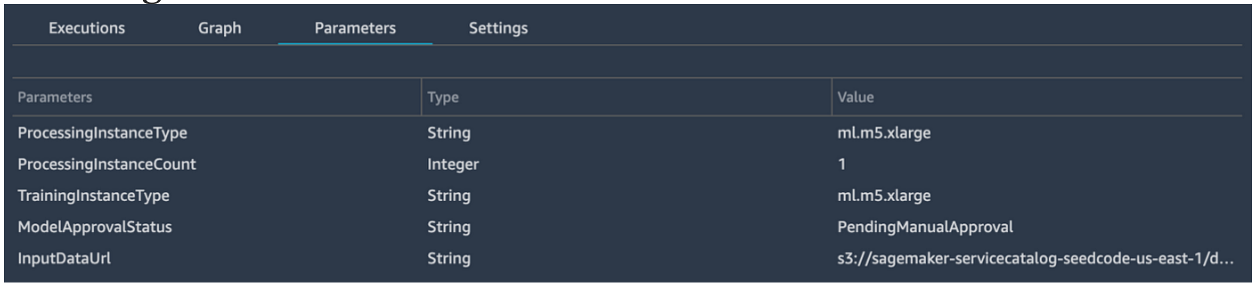
Pipeline Executing (Screenshot by Author)

To understand the workflow if we click Graph we can see the different steps that build this Pipeline.



DAG (Screenshot by Author)

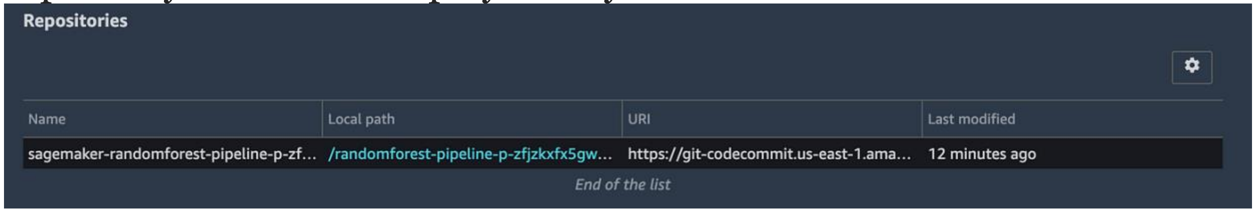
The other main part we talked about was the parameters or variables we are injecting into our Pipeline. This is visible in the Parameters tab to the right.



Parameters	Type	Value
ProcessingInstanceType	String	ml.m5.xlarge
ProcessingInstanceCount	Integer	1
TrainingInstanceType	String	ml.m5.xlarge
ModelApprovalStatus	String	PendingManualApproval
InputDataUrl	String	s3://sagemaker-servicecatalog-seedcode-us-east-1/d...

Parameters (Screenshot by Author)

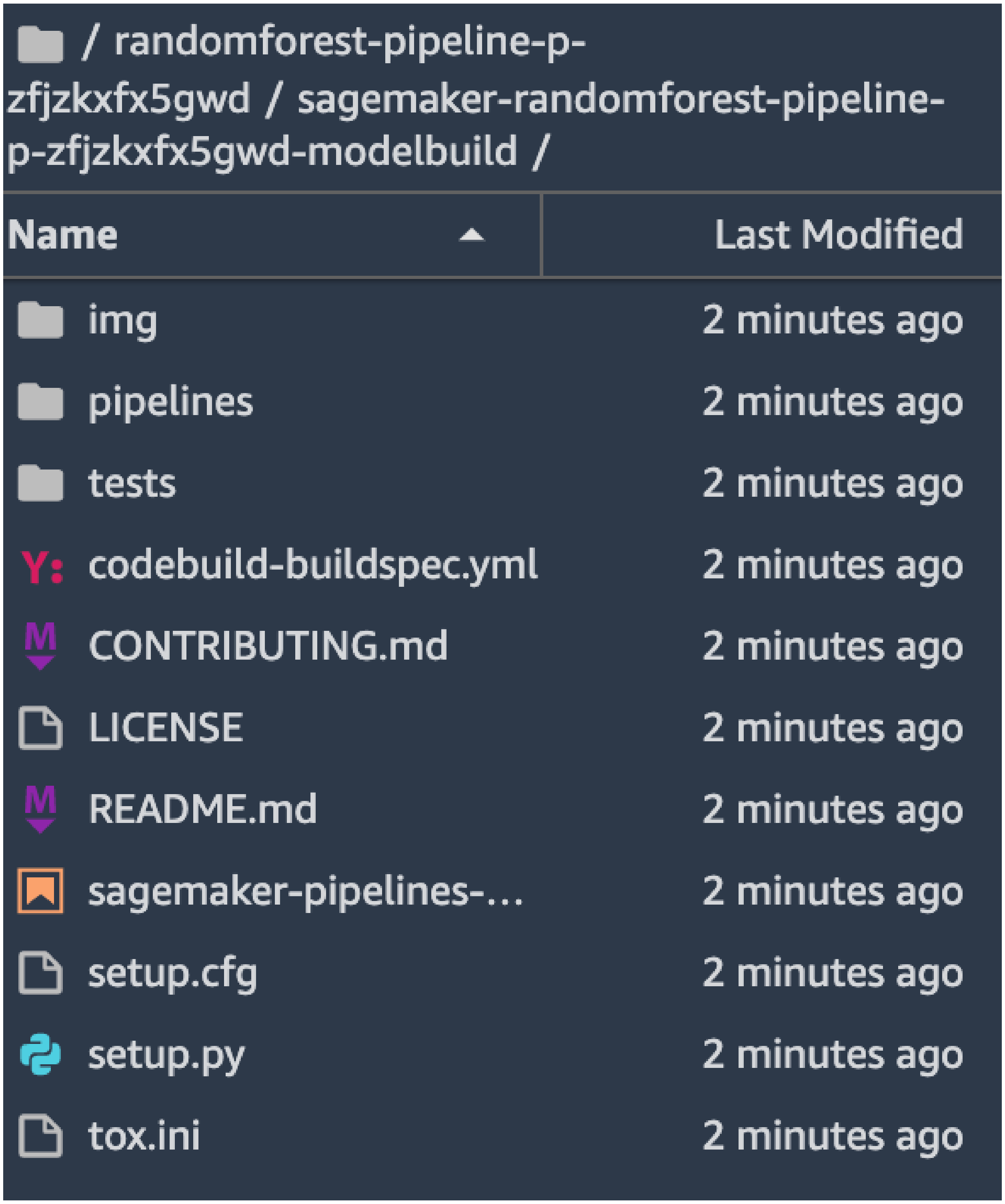
Now where can you actually edit the code that goes behind this Pipeline? If we go back to the SageMaker Projects tab to the randomforest-pipeline Project you should be able to clone this repository and it will display locally in the Studio IDE.



Name	Local path	URI	Last modified
sagemaker-randomforest-pipeline-p-zf...	/randomforest-pipeline-p-zfjzkxfx5gw...	https://git-codecommit.us-east-1.ama...	12 minutes ago

Clone Project Repo (Screenshot by Author)

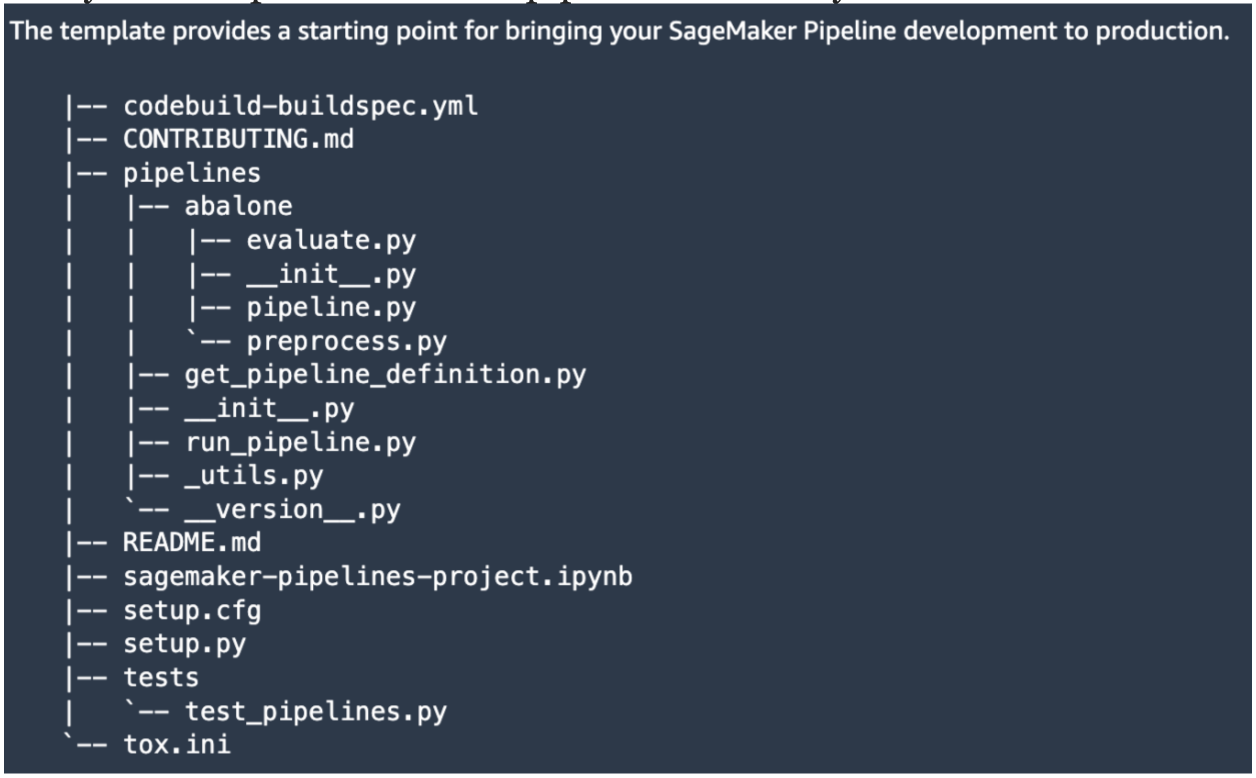
If you click the local path that displays you should then be able to see the code that was used to orchestrate and build this Pipeline.


/ randomforest-pipeline-p-zfjzkxfx5gwd / sagemaker-randomforest-pipeline-p-zfjzkxfx5gwd-modelbuild /

Name	Last Modified
img	2 minutes ago
pipelines	2 minutes ago
tests	2 minutes ago
codebuild-buildspec.yml	2 minutes ago
CONTRIBUTING.md	2 minutes ago
LICENSE	2 minutes ago
README.md	2 minutes ago
sagemaker-pipelines-…	2 minutes ago
setup.cfg	2 minutes ago
setup.py	2 minutes ago
tox.ini	2 minutes ago

Pipelines Code (Screenshot by Author)

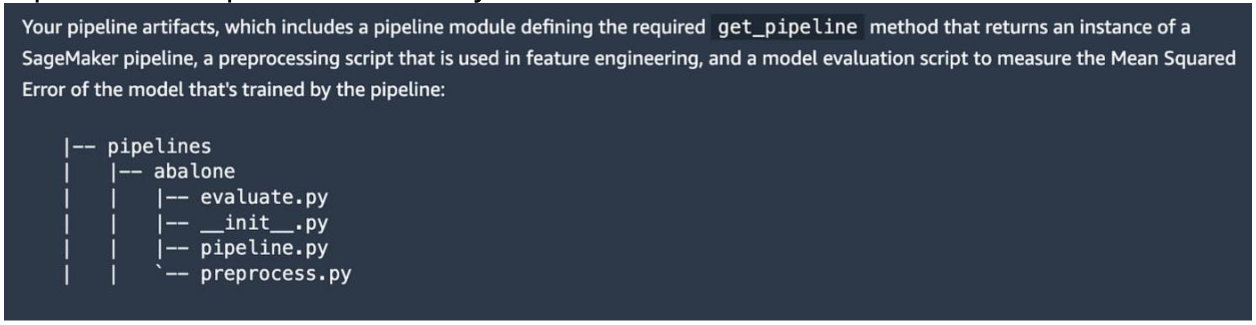
By default Pipelines builds an example with the Abalone dataset, but you can edit this code for your data and the model you will want to build/train. If we go to the Pipelines directory you should be able to see a lot of this boilerplate code that you can build off of. There's also a notebook at the root directory that helps execute the Python scripts that you have provided in the pipelines directory.



The template provides a starting point for bringing your SageMaker Pipeline development to production.

```
|-- codebuild-buildspec.yml
|-- CONTRIBUTING.md
|-- pipelines
|   |-- abalone
|   |   |-- evaluate.py
|   |   |-- __init__.py
|   |   |-- pipeline.py
|   |   `-- preprocess.py
|   |-- get_pipeline_definition.py
|   |-- __init__.py
|   |-- run_pipeline.py
|   |-- _utils.py
|   `-- __version__.py
|-- README.md
|-- sagemaker-pipelines-project.ipynb
|-- setup.cfg
|-- setup.py
|-- tests
|   `-- test_pipelines.py
`-- tox.ini
```

Pipelines Setup (Screenshot by Author)



Your pipeline artifacts, which includes a pipeline module defining the required `get_pipeline` method that returns an instance of a SageMaker pipeline, a preprocessing script that is used in feature engineering, and a model evaluation script to measure the Mean Squared Error of the model that's trained by the pipeline:

```
|-- pipelines
|   |-- abalone
|   |   |-- evaluate.py
|   |   |-- __init__.py
|   |   |-- pipeline.py
|   |   `-- preprocess.py
```

Python Scripts To Define Pipeline (Screenshot by Author)

The main Python script you should be concerned about is the **pipeline.py**. This is the script that wraps together all of your steps and captures the entirety of the workflow that you have defined. You can see this at the end of the Python file.

These scripts are then executed in the notebook cells that are already pre-made for you.

Execute Pipeline

4. Additional Resources & Conclusion

This was a gentle introduction into how you can utilize SageMaker Pipelines for your MLOps journey. Utilizing Projects you can get pre-built templates that you can easily adjust for your own use case. Pipelines offer a vast range of steps and possibilities coupled with the computing scale and power that SageMaker offers as a whole.

Additional Resources

SageMaker Pipelines End to End Example

Pipelines Youtube Demo

MLOps with SageMaker

If you enjoyed this article feel free to connect with me on LinkedIn and subscribe to my Medium Newsletter. If you're new to Medium, sign up using my Membership Referral.