



# MACHINE LEARNING MODEL DEPLOYMENT WITH IBM CLOUD WATSON STUDIO

## PHASE 4

Development part-2

### **Title:**

**Machine learning model deployment with IBM cloud Watson studio**

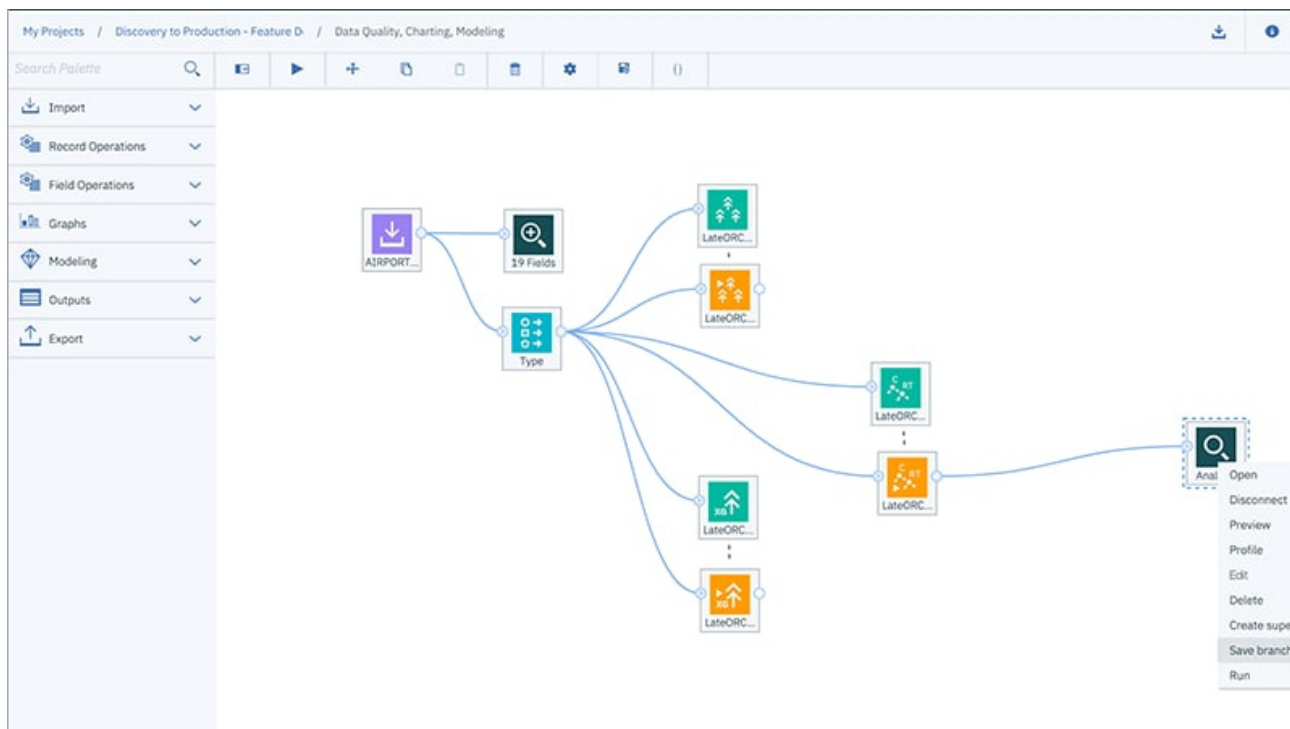
### **Introduction :**

IBM Cloud Watson Studio provides a suite of tools for data scientists, application developers, and subject matter experts to collaboratively work with data to build and train models at scale <sup>1</sup>. The platform offers various tools for designing, training, and managing machine learning models.

### **DEPLOYING MODELS WITH WATSON MACHINE LEARNING:**

Using IBM Watson Machine Learning, you can deploy models, scripts, and functions, manage your deployments, and prepare your assets to put into production to generate predictions and insights.

This graphic illustrates a typical process for a machine learning model. After you build and train a machine learning model, use Watson Machine Learning to deploy the model, manage the input data, and put your machine learning assets to use.



# IBM Watson Machine Learning architecture and services:

Watson Machine Learning is a service on IBM Cloud with features for training and deploying machine learning models and neural networks. Built on a scalable, open source platform based on Kubernetes and Docker components, Watson Machine Learning enables you to build, train, deploy, and manage machine learning and deep learning models.

## Deploying and managing models with Watson Machine Learning:

Watson Machine Learning supports popular frameworks, including: TensorFlow, Scikit-Learn, and PyTorch to build and deploy models.

## Programming Interfaces:

Use Python client library to work with all of your Watson Machine Learning assets in a notebook.

Use REST API to call methods from the base URLs for the Watson Machine Learning API endpoints.

When you call the API, use the URL and add the path for each method to form the complete API endpoint for your requests.

**Example:** data\_asset payload:

```
"input_data_references": [{  
  "type": "data_asset",  
  "location": {  
    "href": "/v2/assets/<asset_id>?space_id=<space_id>"  
  }  
}]
```

**Program:**

```
import requests
```

```
url = "https://us-south.ml.cloud.ibm.com/ml/v4/deployment_job_definitions?  
version=2019-10-25"
```

```
payload = "{ \"tags\": [ \"test\", \"frank\" ], \"space_id\": \"d814cfc4-fba7-4130-b31a-  
06151b88ea38\", \"name\": \"test\", \"description\": \"test td\", \"custom\": { \"test\": \"hi\" },  
\"deployment\": { \"id\": \"1223\" }, \"decision_optimization\": { \"input_data\": [ { \"id\": \"123\", \"fields\":  
[], \"values\": [] } ], \"output_data\": [ { \"id\": \"123\" } ] }, \"hardware_spec\": { \"id\": \"  
{hardware_spec_id}\", \"rev\": \"12\", \"name\": \"string\", \"num_nodes\": 2 },  
\"hybrid_pipeline_hardware_specs\": [ { \"node_runtime_id\": \"auto_ai.kb\", \"hardware_spec\":  
{ \"id\": \"{hardware_spec_id}\", \"rev\": \"12\", \"name\": \"default_py3.6\", \"num_nodes\": 2 } } ] }"
```

```
headers = {
```

```
'Authorization': 'Bearer <token>',  
'Content-Type': 'application/json',  
'Cookie': '_cfduid=dc6cd6997a0c08467ec30c4ecf7a8db7c1601476841'  
}  
  
response = requests.request("POST", url, headers=headers, data = payload)  
  
print(response.text.encode('utf8'))
```

