NO→LATENCY

Vertex AI

- ✓ Managed machine learning platform for ML practioners

- ✓ Covers end to end Lifecycle of a Machine Learning Model :
  1. Data preparation/gathering
  2. Model Training/versioning
  3. Model Evaluation
  4. Model Deployment

- ✓ Supports both UI based and code-based Model development/deployments

- ✓ Jupyter Notebook based version for developmenet

- ✓ Supported Frameworks :
  - Scikit-learn
  - XGBoost
  - Tensorflow
  - PyTorch

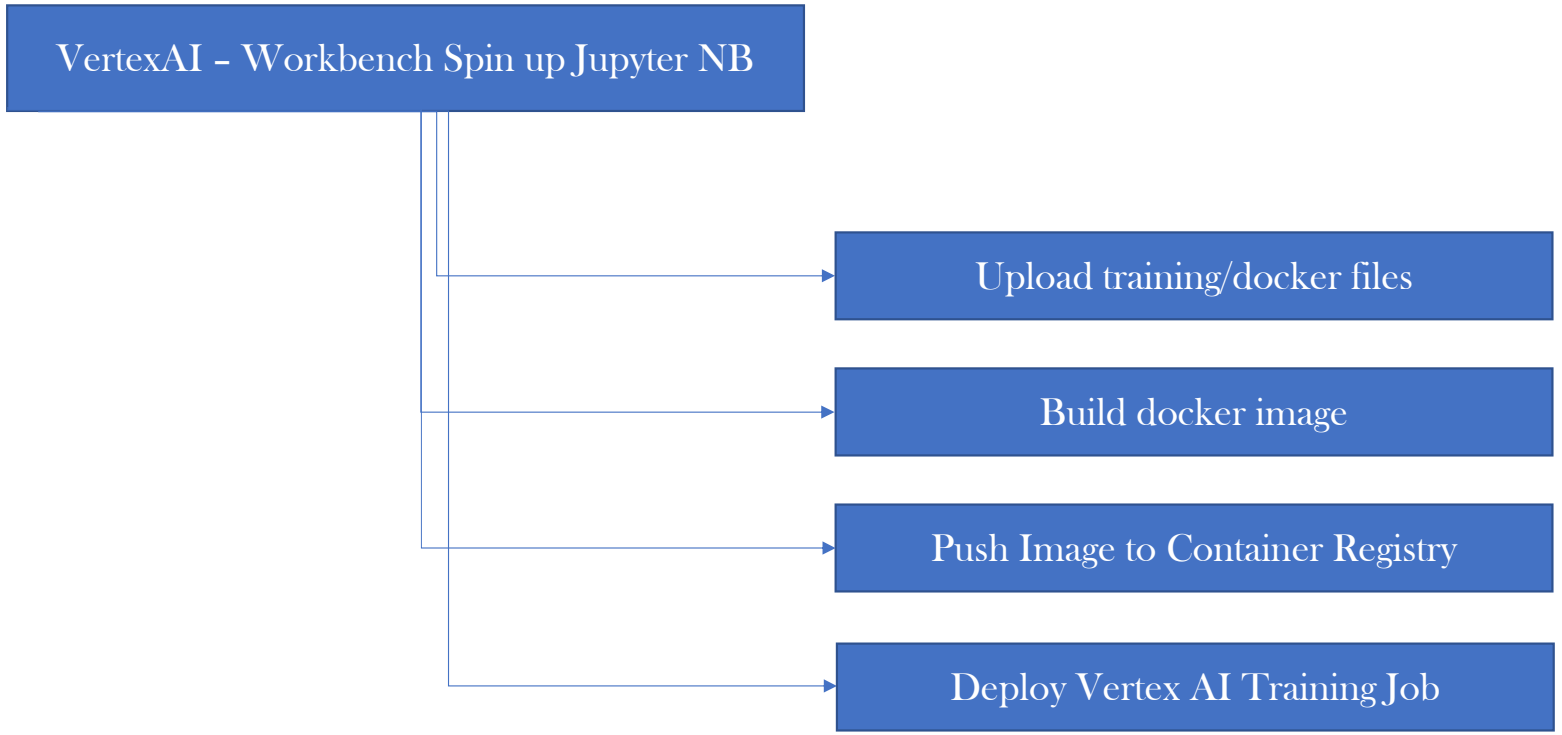
NO→LATENCY

Vertex AI

- ✓ Interactive Model development using Jupyter Notebooks
- ✓ Model containerization and model training using Vertex AI Training (UI and SDK)
- ✓ Model Registry (UI and Python SDK)
- ✓ Model production deployment to an endpoint (UI and SDK)
- ✓ Prediction Model serving (Online and Batch)
- ✓ Kubeflow Pipelines (ML Orchestration)
- ✓ Focuss on Scikit-learn framework
- ✓ Custom predictions/routines (Not automl)

NO→LATENCY

Vertex AI – Model Training

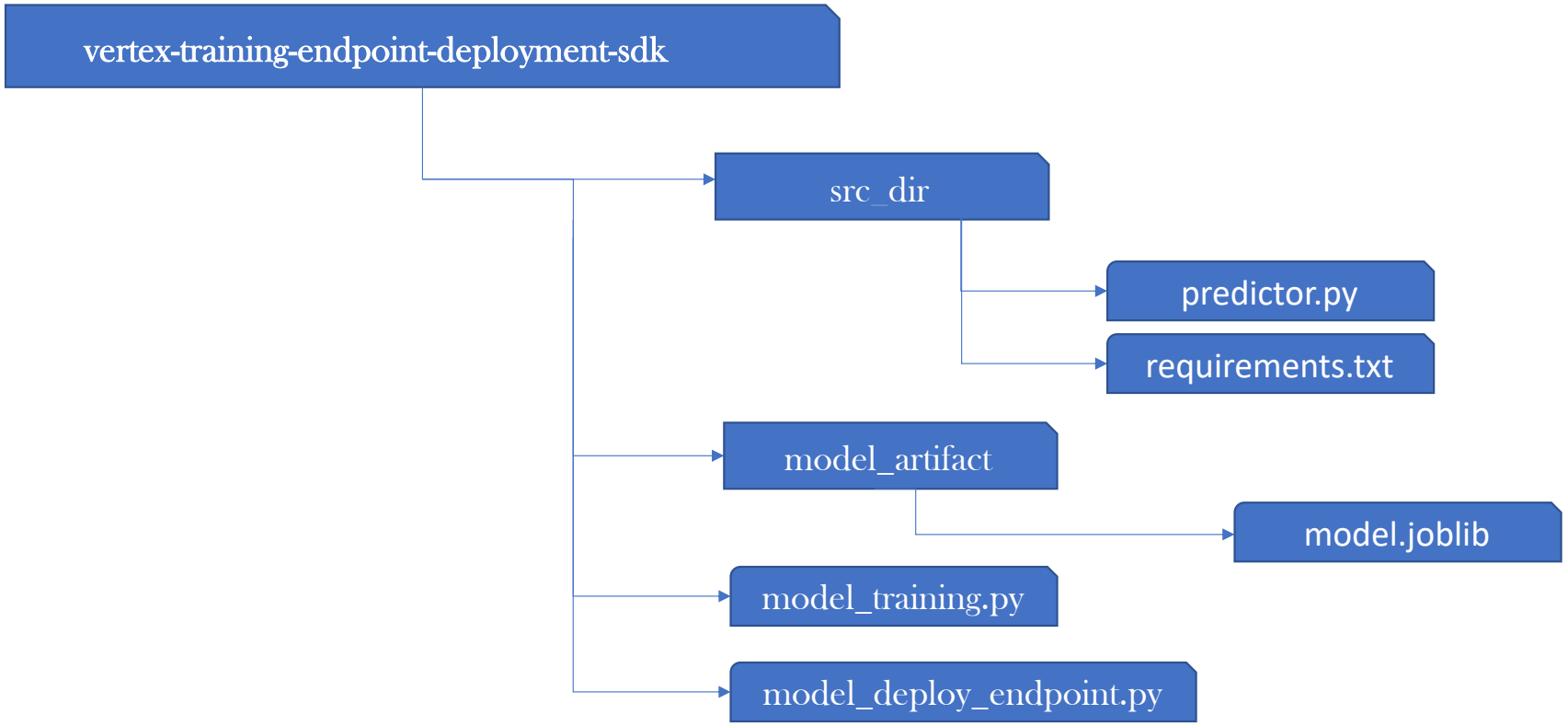


NO→LATENCY

Vertex AI – Model Training & Endpoint Deployment | Python SDK

Step-1	Model Training Using Python SDK/Web Console 
Step-2	Upload Model to Vertex AI Model Registry – Python SDK 
Step-3	Deploy Model to Vertex AI Endpoint

Vertex AI – Model Training & Endpoint Deployment | Python SDK



NO→LATENCY

Vertex AI

1. Train Model using Python SDK
    1. Model artifact in local **src_dir** folder
    2. Upload the artifact to **gcs bucket**

2. Build and Deploy model to a local endpoint as a Docker image including the predictor

3. Run predictions/health-checks using the local endpoint

4. Push and upload model from local endpoint to **Vertex AI Model Registry**

5. Deploy model from **Model Registry** to **Vertex AI Endpoint**

6. Run predictions on the endpoint