

**Question #1** – Your team is designing a financial analysis model for a major Bank.

The requirements are:

1. Various banking applications will send transactions to the new system both in real-time and in batch in standard/normalized format.
2. The data will be stored in a repository
3. Structured Data will be trained and retrained
4. Labels are drawn from the data.
5. You need to prepare the model quickly and decide to use Auto ML for structured Data.


Which GCP Services could you use?

**Answer #1** – Auto ML Tables is aimed to automatically build and deploy models on your data in the fastest way possible.

It is integrated within BigQuery ML and is now available in the unified Vertex AI.

Vertex AI includes an AI Platform, too.

But AI Platform alone doesn't have any AutoML Services.



Item ID	Date and Time	Brand	Category	Price
COLD43245	2019-03-31 11:54:12	Brrr Brands	Coat	\$259.57
ACCS54326	2019-04-01 14:12:10	Top Tops	Hat	\$49.99
B00T12365	2019-04-01 14:31:34	Head Over Heels	Shoe	\$89.49
RING54903	2019-04-02 18:01:59	Stone Gold Jewelry	Jewelry	\$189.51

For any further detail:

<https://cloud.google.com/automl-tables/docs/beginners-guide>

<https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-automl>

<https://cloud.google.com/vertex-ai/docs/beginner/beginners-guide#text>

**Question #2** – What is Tensorboard?

**Answer #2** – Tensorboard is aimed at model creation and experimentation:

- Profiling
- Monitoring metrics, weights, biases
- Examine model graph

- Working with embeddings

For any further detail:

<https://www.tensorflow.org/tensorboard>

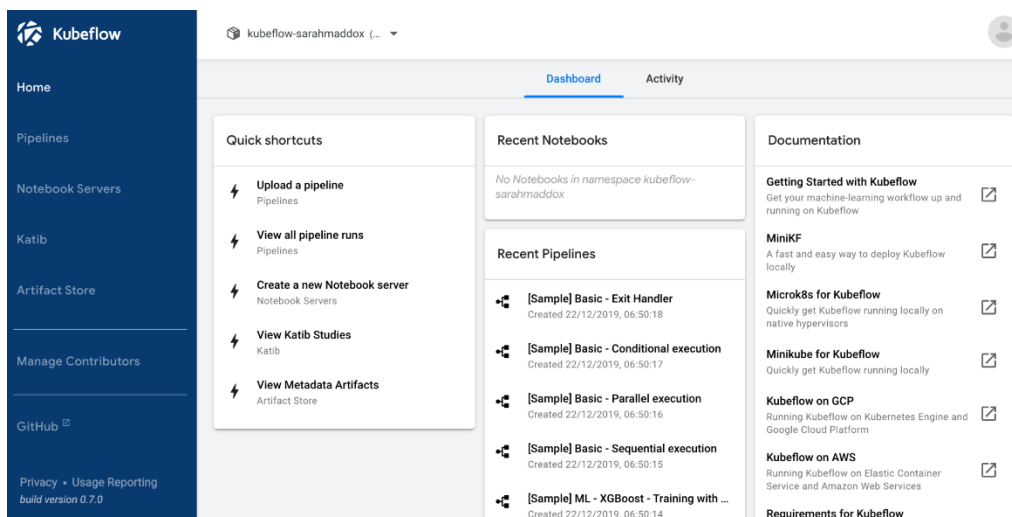
**Question #3** – What are Jupyter notebooks?

**Answer #3** – Jupyter notebooks are a wonderful tool to develop, experiment, and deploy. You may have the latest data science and machine learning frameworks with them.

**Question #4** – What is Kubeflow UIs?

**Answer #4** – The Kubeflow UIs is for ML pipelines and includes visual tools for:

- Pipelines dashboards
- Hyperparameter tuning
- Artifact Store
- Jupyter notebooks



For any further detail:

<https://cloud.google.com/vertex-ai/docs/pipelines/visualize-pipeline>

<https://www.kubeflow.org/docs/components/central-dash/overview/>

**Question #5** – What is KFServing?

**Answer #5** – KFServing is an open-source library for Kubernetes that enables serverless inferencing. It works with TensorFlow, XGBoost, scikit-learn, PyTorch, and ONNX to solve issues linked to production model serving. So, no UI.

For any further detail:

<https://www.kubeflow.org/docs/components/kfserving/kfserving/>

**Question #6** – Do Vertex AI combines AutoML and AI Platform?

**Answer #6** – Yes, Vertex AI is a suite of services that combines AutoML and AI Platform – you can use both AutoML training and custom training in the same environment.

**Question #7** –What is tf.TFRecordReader?

**Answer #7** – The TFRecord format is efficient for storing a sequence of binary and not-binary records using Protocol buffers for serialization of structured data.



For any further detail:

[https://www.tensorflow.org/tutorials/load\\_data/tfrecord](https://www.tensorflow.org/tutorials/load_data/tfrecord)

**Question #9** – What is tf.RaggedTensor?

**Answer #9** – RaggedTensor is a tensor with ragged dimensions, that is with different lengths like this: `[[6, 4, 7, 4], [], [8, 12, 5], [9], []]`.

**Question #10** – What is Tf.quantization?

**Answer #10** – Tf.quantization is aimed to reduce CPU and TPU GCP latency, processing, and power.

**Question #11** – What is tf.train.Feature?

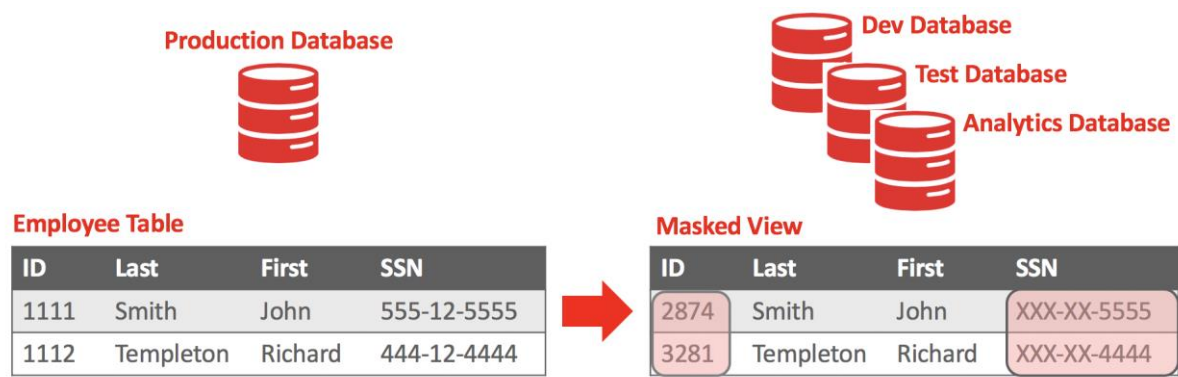
**Answer #11** – tf.train is a feature for Graph-based Neural Structured model training.

**Question #12** – What is TensorFlow Profiler or TFProfiler?

**Answer #12** – TensorFlow Profiler is a tool for checking the performance of your TensorFlow models and helping you to obtain an optimized version.

In TensorFlow 2, the default is eager execution. So, one-off operations are faster, but recurring ones may be slower. So, you need to optimize the model.





For any further detail:

[https://en.wikipedia.org/wiki/Data\\_masking](https://en.wikipedia.org/wiki/Data_masking)

<https://www.mysql.com/it/products/enterprise/masking.html>

**Question #18** – What is k-anonymity?

**Answer #18** – k-anonymity is a way to anonymize data in such a way that it is impossible to identify person-specific information. Still, you maintain all the information contained in the record.

For any further detail:

<https://en.wikipedia.org/wiki/K-anonymity>

**Question #19** – What is Format-preserving encryption?

**Answer #19** – Format-preserving encryption (FPE) encrypts in the same format as the plaintext data.

For example, a 16-digit credit card number becomes another 16-digit number.

**Question #20** – What is Replacement?

**Answer #20** – Replacement just substitutes a sensitive element with a specified value.

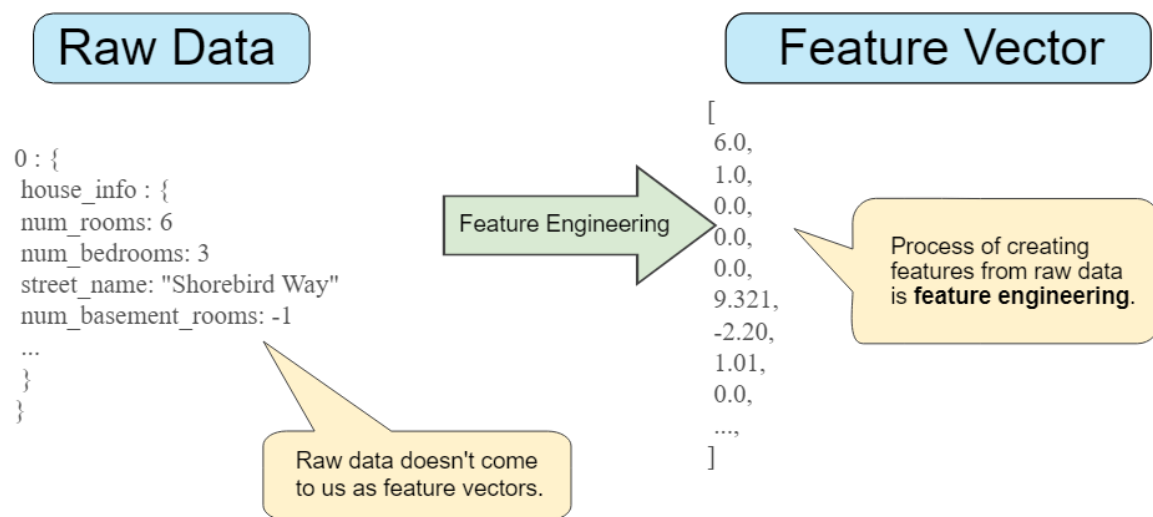
**Question #21** – What is Feature Store?

**Answer #21** – Feature engineering means transforming input data, often strings, into a feature vector. Lots of effort is spent in mapping categorical values in the best way: we have to convert strings to numeric values. We have to define a vocabulary of possible values, usually mapped to integer values. We remember that in an ML model everything must be translated into numbers. Therefore, it is easy to run into problems of this type. Vertex Feature Store is a service to organize and store ML features through a central store. This allows you to share and optimize ML features important for the specific environment and to reuse them at any time.

All these translate into the greater speed of the creation of ML services. But these also allow minimizing problems such as processing skew, which occurs when the distribution of data in production is different from that of training, often due to errors in the organization of the features.

For example, Training-serving skew may happen when your training data uses a different unit of measure than prediction requests.

So, Training-serving skew happens when you generate your training data differently than you generate the data you use to request predictions. For example, if you use an average value, and for training purposes, you average over 10 days, but you average over the last month when you request prediction.



For any further detail:

<https://developers.google.com/machine-learning/crash-course/representation/feature-engineering>

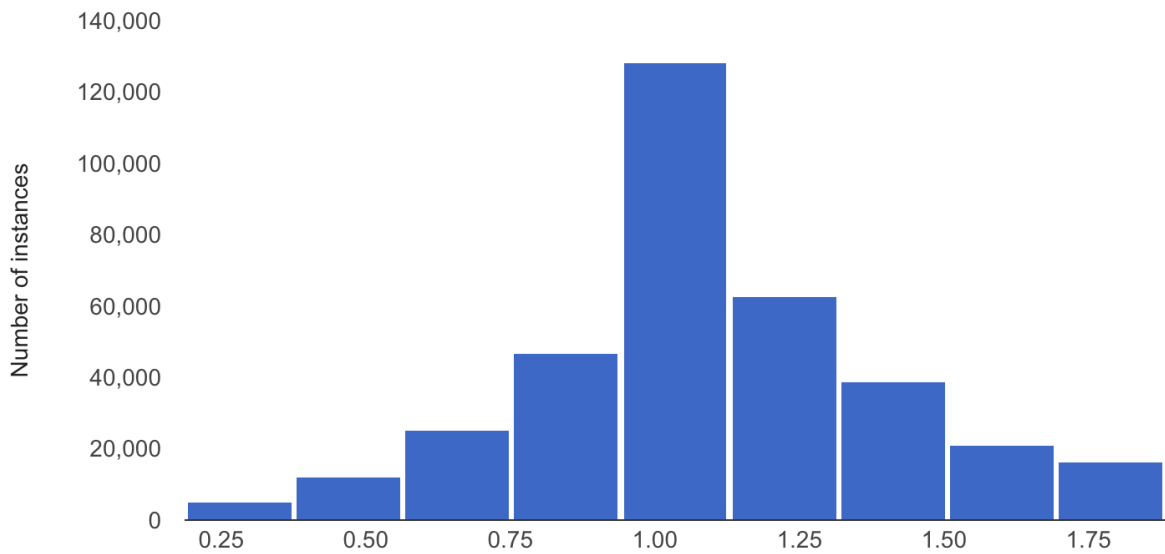
<https://cloud.google.com/architecture/ml-on-gcp-best-practices#use-vertex-feature-store-with-structured-data>

<https://cloud.google.com/blog/topics/developers-practitioners/kickstart-your-organizations-ml-application-development-flywheel-vertex-feature-store>

### Question #22 – What is Model Monitoring?

**Answer #22** – Input data to ML models may change over time. This can be a serious problem, as performance will obviously degrade. To avoid this, it is necessary to monitor the quality of the forecasts continuously. Vertex Model Monitoring has been designed just for this. The main goal is to cope with feature skew and drift detection. For skew detection, it looks at and compares the feature's values distribution in the training data. For drift detection, it looks at and compares the feature's values distribution in the production data. It uses two main methods:

- Jensen-Shannon divergence for numerical features.
- L-infinity distance for categorical features. More details can be found here.



For any further detail:

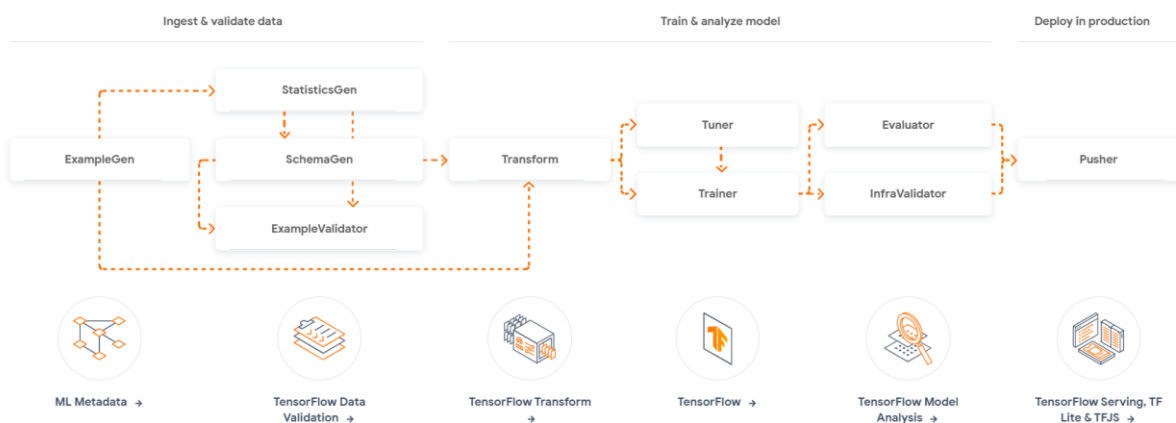
<https://cloud.google.com/vertex-ai/docs/model-monitoring/overview>

**Question #23** – What are the functions of TFX?

**Answer #23** – TensorFlow Extended (TFX) is a set of open-source libraries to build and execute ML pipelines in production. Its main functions are:

- Metadata management
- Model validation
- Deployment
- Production execution.

The libraries can also be used individually.



For any further detail:

<https://www.tensorflow.org/tfx>

**Question #24** – What are AI Platform main functions?

**Answer #24** – AI Platform main functions are:

- Train an ML model
- Evaluate and tune model
- Deploy models
- Manage prediction: Batch, Online and monitoring
- Manage model versions: workflows and retraining

**Question #25** – What are the main features of Kubeflow Pipeline?

**Answer #25** – Kubeflow Pipelines don't deal with production control. Kubeflow Pipelines is an open-source platform designed specifically for creating and deploying ML workflows based on Docker containers. Their main features:

- Using packaged templates in Docker images in a K8s environment
- Manage your various tests / experiments
- Simplifying the orchestration of ML pipelines
- Reuse components and pipelines

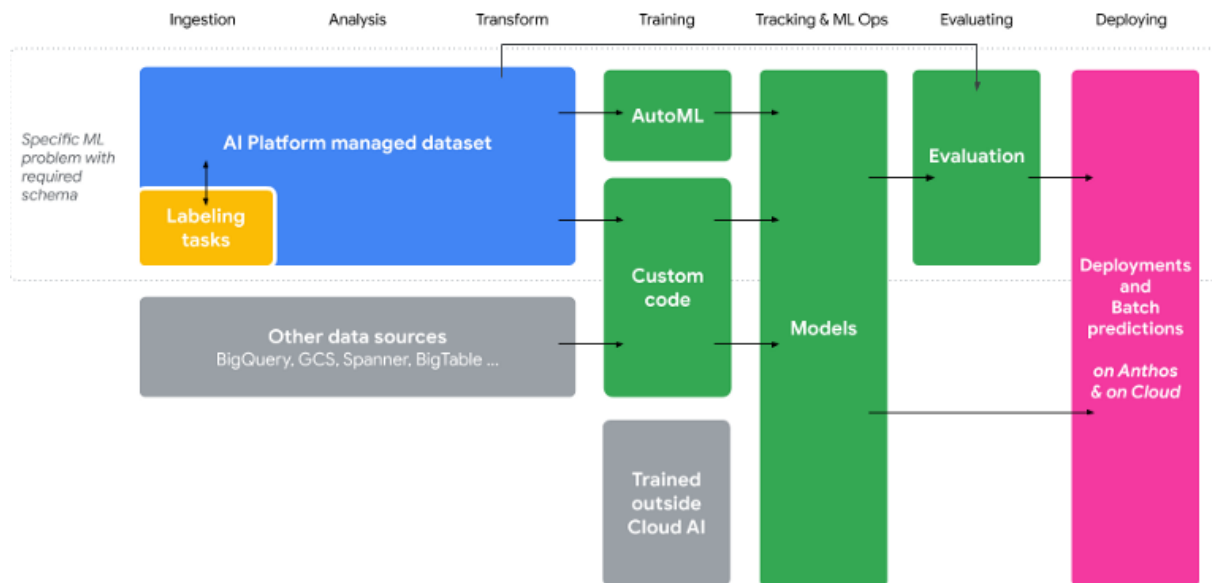
**Question #26** – Your company runs a big retail website. You develop many ML models for all the business activities. You migrated to Google Cloud when you were using Vertex AI. Your models are developed with PyTorch, TensorFlow and BigQueryML. You also use BigTable and CloudSQL, and of course Cloud Storage. In many cases, the same data is used for multiple models and projects. And your data is continuously updated, sometimes in streaming mode. Which is the best way to organize the input data?

**Answer #26** – Vertex AI integrates the following elements:

- Datasets: data, metadata and annotations, structured or unstructured. For all kinds of libraries.
- Training pipelines to build an ML model.
- ML models, imported or created in the environment.
- Endpoints for inference

Because Datasets are suitable for all kinds of libraries, it is a useful abstraction for this requirement.





For any further detail:

<https://cloud.google.com/vertex-ai/docs/datasets/datasets>

<https://cloud.google.com/vertex-ai/docs/training/using-managed-datasets>

<https://codelabs.developers.google.com/codelabs/vertex-ai-custom-code-training>

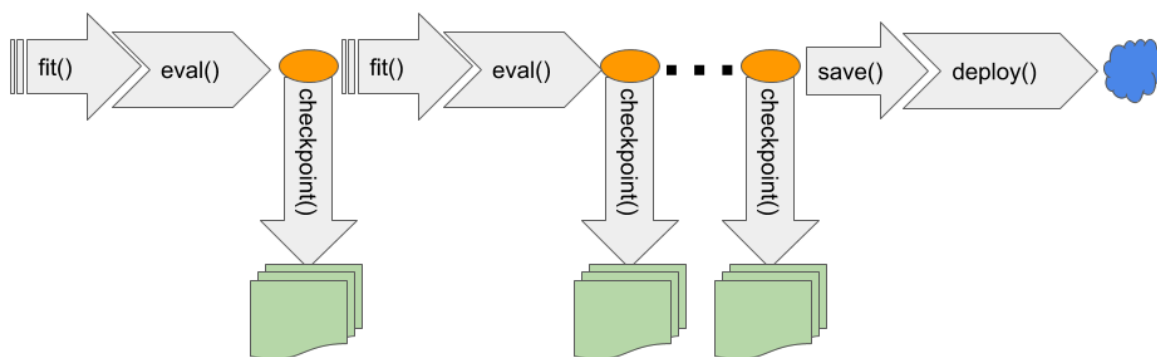
**Question #27** – How to save checkpoints in PyTorch?

**Answer #27** – PyTorch is a popular library for deep learning that you can leverage using GPUs and CPUs. When you have to save a model for resuming training, you have to record both models and updated buffers and parameters in a checkpoint. A checkpoint is an intermediate dump of a model's entire internal state (its weights, current learning rate, etc.) so that the framework can resume the training from that very point. In other words, you train for a few iterations, then evaluate the model, checkpoint it, then fit some more. When you are done, save the model and deploy it as normal.

To save checkpoints, you must use `torch.save()` to serialize the dictionary of all your state data,

In order to reload, the command is `torch.load()`.

## ML Design Pattern #2: Checkpoints



<https://medium.com/@lakshmanok>

@lak\_gcp

For any further detail:

[https://pytorch.org/tutorials/recipes/recipes/saving\\_and\\_loading\\_a\\_general\\_checkpoint.html](https://pytorch.org/tutorials/recipes/recipes/saving_and_loading_a_general_checkpoint.html)

<https://towardsdatascience.com/ml-design-pattern-2-checkpoints-e6ca25a4c5fe>

**Question #28** – You are a Data Scientist. You are going to develop an ML model with Python. Your company adopted GCP and Vertex AI, but you need to work with your developing tools.

What are you going to do?

**Answer #28** – Client libraries are used by developers for calling the Vertex AI API in their code.

The client libraries reduce effort and boilerplate code.

The correct procedure is:

- Enable the Vertex AI API or AI Platform Training & Prediction and Compute Engine APIs.
- Enable the APIs
- Create/Use a Service account and a service account key
- Set the environment variable named `GOOGLE_APPLICATION_CREDENTIALS`

For any further detail:

Installing the Vertex AI client libraries

<https://cloud.google.com/ai-platform/training/docs/python-client-library>

**Question #29** – You work as a Data Scientist for a major banking institution that recently completed the first phase of migration in GCP. You now have to work in the GCP Managed Platform for ML. You need to deploy a custom model with Vertex AI so that it will be available for online predictions.

Which is the correct procedure?

**Answer #29** – AI Platform/Vertex Prediction is a managed serving platform that supports both CPU and, optionally, GPU. Its main functions are aimed to:

- infrastructure setup
- Maintenance
- Management

Its main elements are:

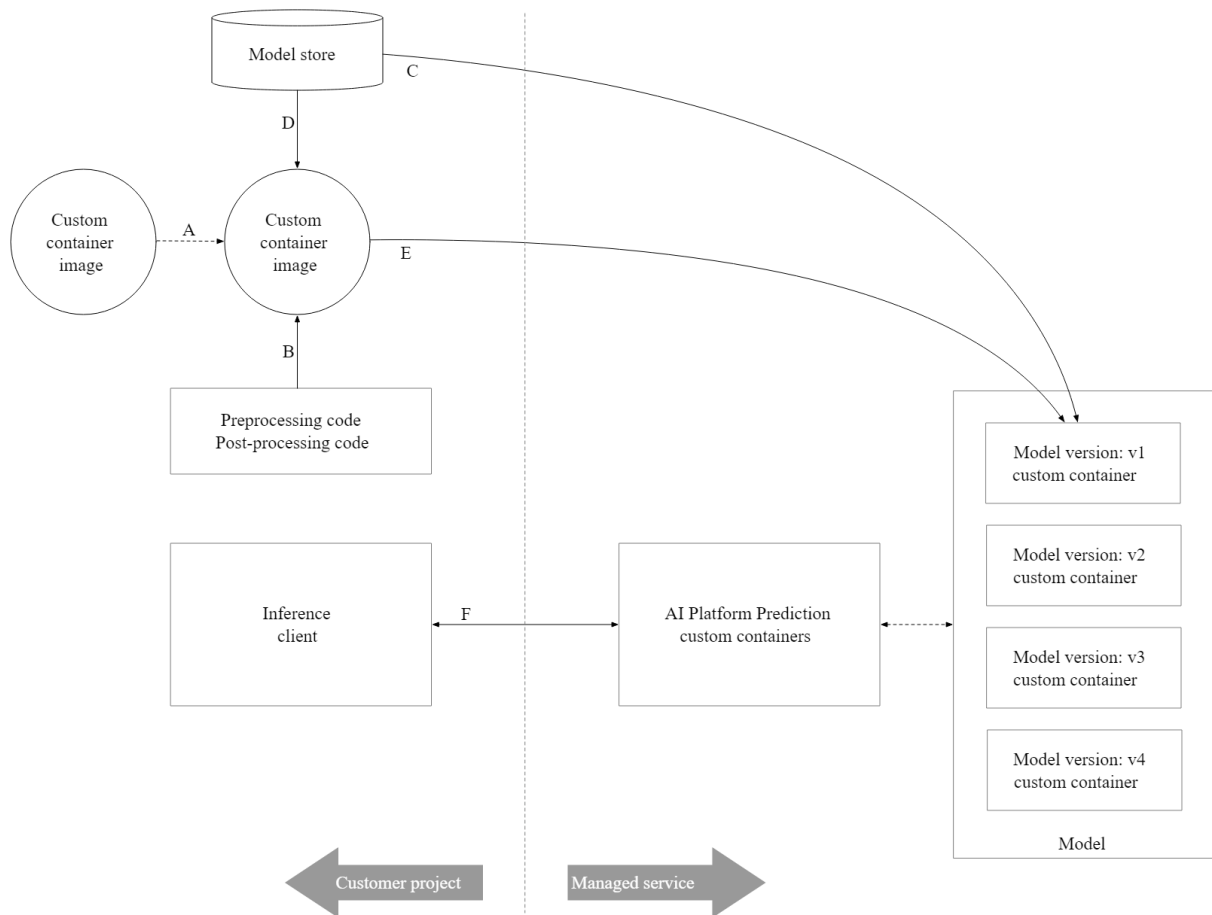
- Base image
- Custom container image (A) of the model.
- Model

AI Platform Prediction uses an architectural paradigm that is based on immutable instances of models and model versions.

- Direct model server
- Direct access to the model server
- Model server with listener
- Listener between the service and the model server

Machine types may be configured with:

- Different number of virtual CPUs (vCPUs) per node
- Desired amount of memory per node
- Support for GPUs, which you can add to some machine types



For any further detail:

Vertex Prediction AI Platform Prediction: Custom container concepts

<https://www.tensorflow.org/tfx/guide/serving>

<https://cloud.google.com/vertex-ai/docs/general/deployment>

<https://cloud.google.com/architecture/ai-platform-prediction-custom-container-concepts>

**Question # –**

**Answer # –**

**Question # –**

**Answer # –**

**Question # –**

**Answer # –**

