syntax = "proto3";

package tensorflow.serving;

import "google/protobuf/any.proto";

import "tensorflow\_serving/config/file\_system\_storage\_path\_source.proto";

import "tensorflow\_serving/config/logging\_config.proto";

option cc\_enable\_arenas = true;

// The type of model.

// TODO(b/31336131): DEPRECATED.

enum ModelType {

MODEL\_TYPE\_UNSPECIFIED = 0 [deprecated = true];

TENSORFLOW = 1 [deprecated = true];

OTHER = 2 [deprecated = true];

}

// Common configuration for loading a model being served.

message ModelConfig {

// Name of the model.

string name = 1;

// Base path to the model, excluding the version directory.

// E.g> for a model at /foo/bar/my\_model/123, where 123 is the version, the

// base path is /foo/bar/my\_model.

//

// (This can be changed once a model is in serving, \*if\* the underlying data

// remains the same. Otherwise there are no guarantees about whether the old

// or new data will be used for model versions currently loaded.)

string base\_path = 2;

// Type of model.

// TODO(b/31336131): DEPRECATED. Please use 'model\_platform' instead.

ModelType model\_type = 3 [deprecated = true];

// Type of model (e.g. "tensorflow").

//

// (This cannot be changed once a model is in serving.)

string model\_platform = 4;

reserved 5, 9;

// Version policy for the model indicating which version(s) of the model to

// load and make available for serving simultaneously.

// The default option is to serve only the latest version of the model.

//

// (This can be changed once a model is in serving.)

FileSystemStoragePathSourceConfig.ServableVersionPolicy model\_version\_policy =

7;

// String labels to associate with versions of the model, allowing inference

// queries to refer to versions by label instead of number. Multiple labels

// can map to the same version, but not vice-versa.

//

// An envisioned use-case for these labels is canarying tentative versions.

// For example, one can assign labels "stable" and "canary" to two specific

// versions. Perhaps initially "stable" is assigned to version 0 and "canary"

// to version 1. Once version 1 passes canary, one can shift the "stable"

// label to refer to version 1 (at that point both labels map to the same

// version -- version 1 -- which is fine). Later once version 2 is ready to

// canary one can move the "canary" label to version 2. And so on.

map<string, int64> version\_labels = 8;

// Configures logging requests and responses, to the model.

//

// (This can be changed once a model is in serving.)

LoggingConfig logging\_config = 6;

}

// Static list of models to be loaded for serving.

message ModelConfigList {

repeated ModelConfig config = 1;

}

// ModelServer config.

message ModelServerConfig {

// ModelServer takes either a static file-based model config list or an Any

// proto representing custom model config that is fetched dynamically at

// runtime (through network RPC, custom service, etc.).

oneof config {

ModelConfigList model\_config\_list = 1;

google.protobuf.Any custom\_model\_config = 2;

}

}