

AI Platform uses images to configure the VMs that service your training and prediction requests in the cloud. These images contain the base operating system, core technology packages, pip packages (Python libraries), and operating system packages. Images are upgraded periodically to include new improvements and features. AI Platform versioning enables you to select the right configuration to work with your model.

- The default AI Platform runtime version used by the AI Platform Training and Prediction API is version **1.0**. If you do not specify a runtime version, AI Platform uses version 1.0.
- You should always test your training jobs and models thoroughly when switching to a new runtime version, regardless of whether it's a major or minor update.
- AI Platform supports each runtime version for 12 months after its release. After the 12-month period, you can no longer create training jobs, batch prediction jobs, or model versions that use the runtime version.

Twenty-four months after the release of the runtime version, AI Platform Prediction deletes all model versions that use the runtime version.

Learn more about the [timeline of availability for runtime versions](#)

(/ml-engine/docs/runtime-version-list#runtime-version-support).

The images that AI Platform uses correspond to the AI Platform runtime version. The runtime version uses the following format:

New major and minor versions are created periodically to incorporate one or more of the following:

- Releases for:
 - Operating system
 - Supported machine learning frameworks
- Changes or updates to AI Platform functionality.

A new major version may include breaking changes that require updates to code written against previous versions. A new minor version should not include breaking changes, and should be backward-compatible with all variations of the same major version.

Make sure to select the runtime version that supports the latest versions of your machine learning framework and other packages you are using. If you don't specify a runtime version for the tasks below, AI Platform uses the default version **1.0** to complete your request.

The earliest AI Platform runtime version that provides support for scikit-learn and XGBoost is version 1.4.

You can see the details of each version in the [AI Platform version list](#) (/ml-engine/docs/runtime-version-list).

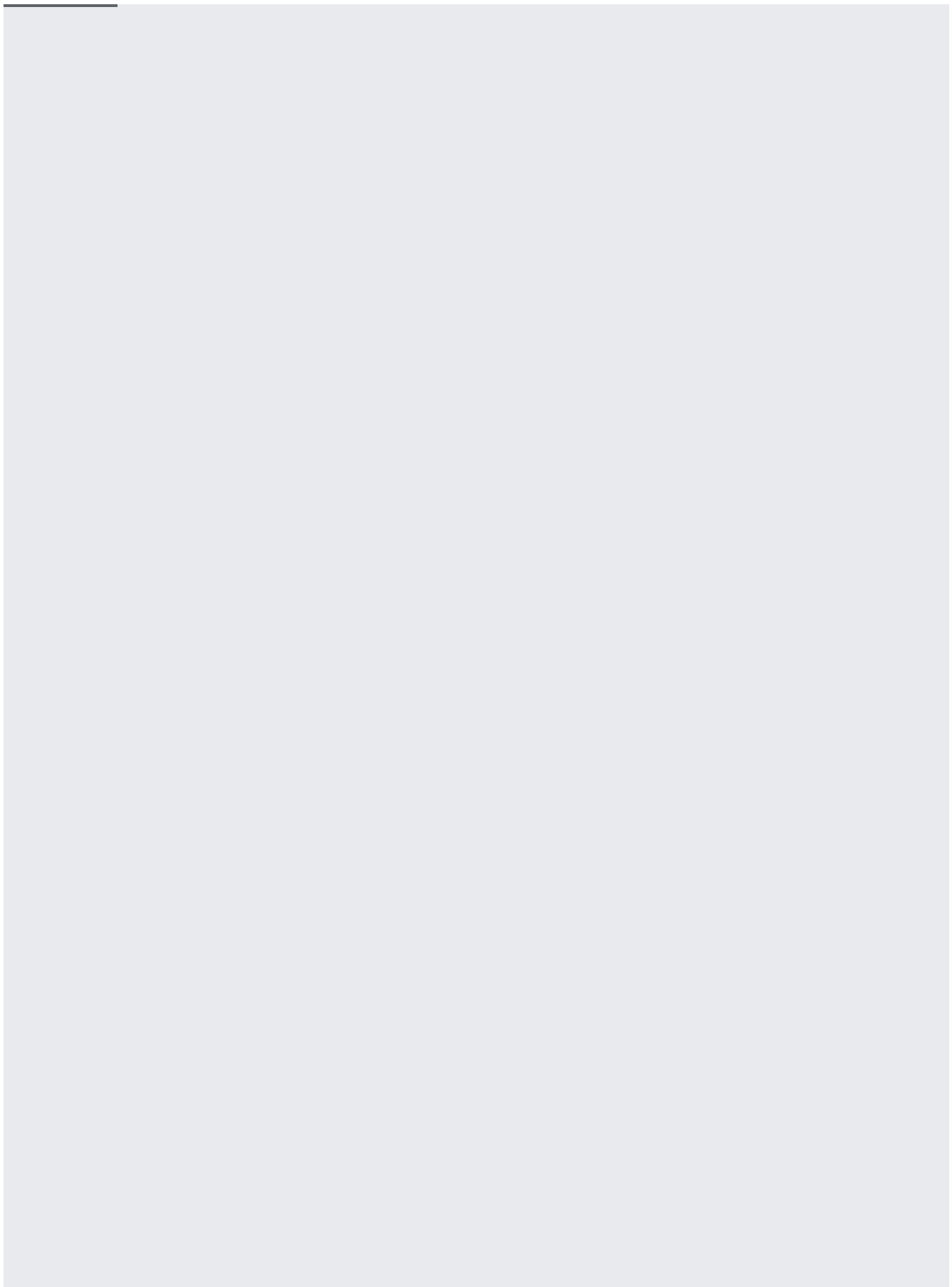
Make sure to set the runtime version when you submit a training job request. Otherwise, AI Platform uses the default version **1.0** for your training job.

See more details about submitting a training job in the [TrainingInput API](#) (/ml-engine/reference/rest/v1/projects.jobs#TrainingInput).

Python 3.5 is available when you use AI Platform runtime version 1.4 through 1.14. To submit a training job with Python 3.5, set the Python version to "3.5" and use a compatible runtime version.

Python 3.7 is available in runtime version 1.15 and later. To submit a training job with Python 3.7, set the Python version to "3.7" and use a compatible runtime version.

If the Python version is not specified, it defaults to "2.7".



See more details about submitting a training job in the [TrainingInput API](#) (/ml-engine/reference/rest/v1/projects.jobs#TrainingInput).

Make sure to specify a runtime version when you create a deployed model version from a trained model. This sets the default runtime version for online and batch prediction requests. If you do not specify a runtime version, AI Platform uses [version 1.0](#) (/ml-engine/docs/runtime-version-list#1.0).

This following section only applies to prediction with TensorFlow.

You can specify a runtime version to use when you create a batch prediction job. If you don't, AI Platform uses the default runtime version set in the model version.

When you create your model version, make sure to specify the runtime version you want to use for online prediction requests. If your model version's default runtime version is incorrect, create a new model version with the correct runtime version.

Online prediction requests always use the model version's default runtime version. You cannot specify a runtime version to override this in your online prediction request.

There are three ways for you to change the packages on your training instances:

- manually uploading package files (tarballs) and including their paths as training input
- specifying PyPI (<https://pypi.python.org/pypi>) packages as dependencies to your trainer package
- building a custom container that pre-installs your dependencies on an image

You can include extra package files as part of your training job request. These will be installed on each training instance. AI Platform installs all packages with pip. Packages designed for other package managers are not supported.

You can specify PyPI packages and their versions as dependencies to your trainer package using the normal setup tools process:

1. In the top-level directory of your trainer application, include a `setup.py` file.

2. When you call `setuptools.setup` in `setup.py`, pass a list of dependencies and optionally their versions as the `install_requires` parameter. This example `setup.py` file demonstrates the procedure:

AI Platform forces reinstallation of packages, so you can override packages that are part of the runtime version's image with newer or older versions.

Instead of using a runtime version, you can build a [Docker](https://docs.docker.com/) (<https://docs.docker.com/>) container to include your dependencies. Learn more about [how to use custom containers](/ml-engine/docs/using-containers) (</ml-engine/docs/using-containers>).

Using a more recent version of TensorFlow than the latest supported runtime version on AI Platform is possible for training, but not for prediction.

To use a version of TensorFlow that is not yet supported as a full AI Platform runtime version, include it as a custom dependency for your trainer using one of the following approaches:

1. Specify the TensorFlow version in your `setup.py` file as a PyPI dependency. Include it in your list of required packages (`#to_include_pypi_package_dependencies`) as follows:

2. Build a TensorFlow binary from sources (https://www.tensorflow.org/install/install_sources), making sure to follow the instructions for TensorFlow with CPU support only. This process yields a pip package (.whl file) that you can include in your training job request (`#to_provide_package_files`) by adding it to your list of packages.

Building a TensorFlow binary to include as a custom package is a more complex approach, but the advantage is that you can use the most recent TensorFlow updates when training your model.

- Review the [list of supported runtime versions](/ml-engine/docs/runtime-version-list) (`/ml-engine/docs/runtime-version-list`).