Highlight Note

Configuring Environment Containers

Usually, you should create environments in containers (this is the default unless the **docker.enabled** property is set to **False**, in which case the environment is created directly in the compute target)

```
env.docker.enabled = True
deps = CondaDependencies.create(conda_packages=['scikit-learn','pandas','pip'],
                                pip_packages=['azureml-defaults']
env.python.conda_dependencies = deps
```

Azure Machine Learning uses a library of base images for containers, choosing the appropriate base for the compute target you specify (for example, including Cuda support for GPU-based compute). If you have created custom container images and registered them in a container registry, you can override the default base images and use your own.

```
env.docker.base_image='my-base-image'
env.docker.base_image_registry='myregistry.azurecr.io/myimage'
```

Alternatively, you can have an image created on-demand based on the base image and additional settings in a dockerfile.

```
env.docker.base_image = None
env.docker.base_dockerfile = './Dockerfile'
```

By default, Azure machine Learning handles Python paths and package dependencies. If your image already includes an installation of Python with the dependencies you need, you can override this behavior by setting **python.user_managed_dependencies** to **True** and setting an explicit Python path for your installation.

```
env.python.user_managed_dependencies=True
env.python.interpreter_path = '/opt/miniconda/bin/python'
```