Pycharm是支持Docker的，我们可以使用Dockers容器来调试我们的程序，当然，如果我们采用Docker调试的话就需要忍受一下Docker的一些缺点。从我的使用情况来看，Docker有以下缺点：

1. 需要重复构建Image：当我们每增加一个Python依赖包时，我们就需要重构一次我们的Image，这需要我们对Docker有一定的了解。
2. 启动/停止速度慢：主要是Container的启动占用了时间，我们的程序真正启动的时间只比平常多了大概20%。
3. Container的重构：该操作是由Pycharm在自动完成的，我们知道Docker是会在每次未指定name创建Container时是会给我们一个名字的。显然Pycharm是记录了这个名字的，但Pycharm会在“程序/Image/系统状态”发生了更改后，比如我们重启了，我们给我们的程序添加了一个.py文件，这种情况下Pycharm就会重构我们的Containers，而且不会删除原来的。（虽然我们不必在意这些，多了就手动全部删除就行，但看着不爽----处女座慎看）
4. Container系统组件的缺失：如果你是做运维开发的，或者你涉及到了系统操作，请不要用Docker，除非你确定你的Image包含了你想要的。

以上就是我总结的Docker的缺点。

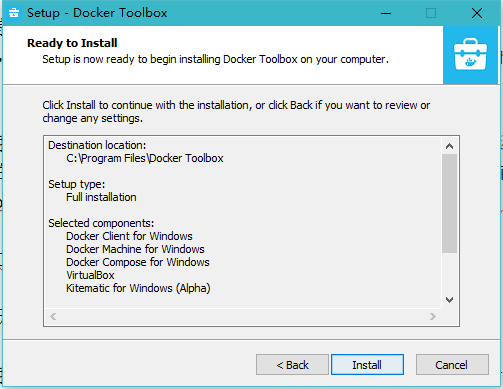
当然，我想Docker在生产中更多的是用在不影响现有生产环境的背景下。

由于我是在Windows下，我采用的是Docker Toolbox环境，因为我还要用Vmware和Virtualbox，所谓的“原生”Docker会打开Hyper-V，相信玩虚拟机的朋友对它深恶痛绝。而且所谓的原生Docker其实也是在Hyper-V中建立一个基于Debian的Docker虚拟机。

DaoIO上说不要用Docker Toolbox，我是反对的。

正式开始：

1. 我们去Docker官网下载Docker Toolbox，然后下一步下一步的安装就可以了



1. 使用QuickStart或直接在Virtualbox里启动我们的Dockers

。。。此处省略一万行

。。。此处省略两万行

（不会就去看文档去）

如果是初次使用，我们要先Pull一个镜像下来，还需要到修改下我们的加速器，我们可不想忍受那50K的小水管。具体方法参见Daocloud（免费的）

<https://www.daocloud.io/mirror#accelerator-doc>

然后我们先pull一个镜像下来

docker pull python

我一般都采用ubuntu原始image来生成我想到的镜像

1. 初始化我们的Image

。。。此处省略一万行

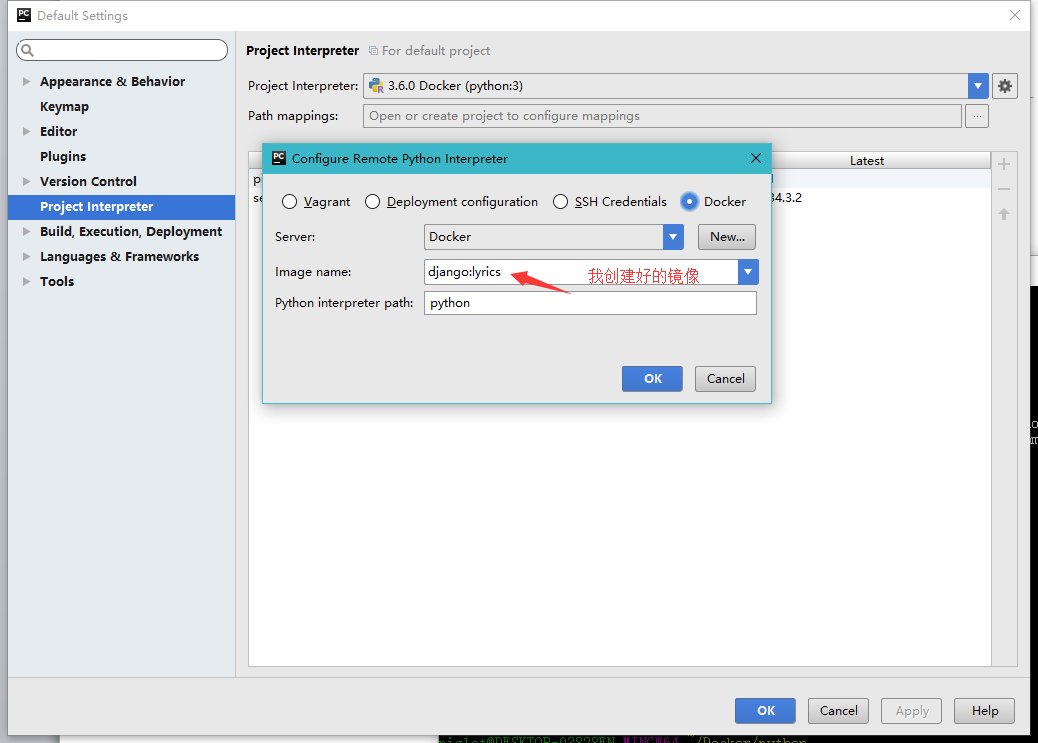
。。。此处省略两万行

（不会就去看文档去）

我已经做好一个用于Django的Image，基于Ubuntu的

1. 创建Pycharm到Docker的连接

打开Pycharm的Interpreter，添加我们的Docker服务



Pycharm会自动配置的，等待我们的只是OK

如果你需要多个，那就按照这种方法去创建多个基于不同镜像的Interpreter就行

1. 开始我们的程序