

Apollo是一个自动驾驶的平台，推荐的参考运行环境为：ThinkPAD X240、CPU：i5、四核、内存8G、硬盘容量40G以上。

01 安装过程

如图1所示，安装主要分为三步。安装基础环境，拉取Docker镜像并创建容器，进入容器编译源码。下面介绍具体的安装过程。



图1 Apollo系统的安装过程

首先是安装git，因为Apollo代码是托管在github平台的，所以需要git工具。然后使用git将Apollo源码克隆到本地，如图2所示。



图2 安装git和拉取镜像

下载源码之后，还要安装Docker环境，可以使用Apollo提供的脚本安装也可以根据Docker官网的指南进行安装。安装完Docker之后需要验证。重启系统并进入Apollo项目所在文件夹，然后运行 `docker ps` 命令，显示容器数据为0，则表示Docker安装完成。

Docker环境安装好之后，使用官方提供的脚本拉取Apollo Docker镜像文件，运行 `dev_start.sh -C` 命令，其中 `-C` 选项表示使用中国服务器进行加速。在拉取成功之后，该脚本会基于镜像创建一个容器 Container。



注意：每次都要执行这个操作，因为CONTAINER需要每次都创建。

接下来是对Apollo的操作，如果没有编译，需要先使用`apollo.sh`脚本进行编译。它有很多编译选项，默认的是Build和OP。还可以选择面向GPU编译。RS是对RB速腾聚创的激光雷达进行编译，USB Camera是对Camera的编译，这几种编译方式所涉及的类不同，所以使用的编译方式也不尽相同。

编译完成之后，需要对Apollo各个模块进行调试，我们会在每个版本发布的时候给出对应的Rosbag数据包，方便去做验证。比如说Apollo 1.0提供了一个循迹数据包，2.0时发布了一个激光点云的数据包。

我们可以启动`bootstrap.sh`脚本，对Apollo的bag进行回放，看一下效果。这是一个引导脚本，它做了以下事情，启动进程守护工具Supervisor，假如有进程出现不可预知的异常，这些进程通常会挂掉，经过配置后Supervisor可以保证在进程挂掉后，会将该进程拉进来。然后启动Roscore、voice_dectore和Dreamviewer。

对于Apollo平台，很多的模块都被启动，交由Supervisor进程进行监控，包括Can Bus、激光雷达、控制模块、GPS、Mobileeye、NG等模块。

在运行完`bootstrap.sh`脚本之后，在浏览器地址栏输入`localhost:8888`查看Demo的演示效果。Demo加载bag对应的数据，包括**车辆的数据、障碍物数据、绿色障碍物ID、速度、形态**。车在运行过程中需要查看的不仅仅是仿真出来的场景，还要看一些跟Planner、控制相关的信息。



使用仿真平台

仿真平台的使用方式大概分为以下几个步骤，如图4所示。



图4 Apollo 仿真平台

按照图5所指示的方法可以进入官方仿真平台，前提是有百度的账号。



图5 如何使用仿真平台

