# 概述

RG-ONDP机器学习库（以下简称“AI库”）呈现方式为容器化的CentOS7系统，可以基于此系统使用python直接调用scikit-learn和fbprophet库，另外可以根据业务需求按照镜像制作步骤完成其他库的添加。本文主要用于描述以下内容：

1. 容器化环境中各个软件及算法库的版本说明；
2. 算法库的选择标准说明
3. 镜像制作流程
4. 容器化部署方式

# 软件说明

下表主要描述容器中各个软件的版本号

|  |  |  |
| --- | --- | --- |
| 软件 | 版本号 | 说明 |
| CentOS | 7.5.1804 | Linux基础系统 |
| gcc | 4.9.2 | C语言编译库，用于编译ython3.5.2。该版本为最新版，每次制作镜像时将自动升级。 |
| 4.8.5 | C语言编译库，用于编译ython2.7.5 |
| python | 2.7.5 | 系统自带版本，使用python命令唤起命令行工具 |
| 3.5.2 | 新加入版本，使用python3命令唤起命令行工具 |
| sshd | OpenSSH\_7.4p1 | 用于服务器登录，容器内部端口为22，对外端口为22346 |

## 文件目录

为了便于维护，特此将部署文件罗列出来：

# 部署和使用

## 部署方式

AI库主要以容器的形式供用户调用，启动方式主要依托于kubenetes集群(以下简称k8s)环境。

（1）验证是否已经安装：

使用ssh登录服务器：

|  |
| --- |
| ssh rgibns1:22346  账户：root/rJ1#iBns |

如果连接成功则表示k8s中已经有该容器，则直接在该服务环境中使用即可。

（2）重新部署

当无法通过k8s连接到该集群时，表示未安装AI库容器，或者AI库容器出现故障。

登录到rgibns1的宿主机上查看k8s中是否存在该容器：



发现未显示容器的详细信息，则表示该容器未部署，需要手动部署。部署方式是运行脚本start.sh

|  |
| --- |
| chmod +x start.sh &&  ./start.sh |

执行过程中会提示创建了两个服务：



再次查看容器是否存在：



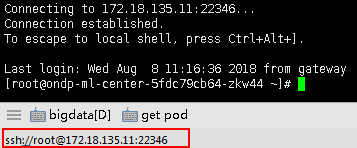
至此，出现Running状态，说明容器部署成功，可以再次使用ssh命令登录服务器。如果不是该Runing状态，则请联系管理员。

## 使用方式

使用ssh登录服务器：

|  |
| --- |
| ssh rgibns1:22346  账户：root/rJ1#iBns |

使用客户端程序登录则为：



在该容器中可以使用python、python3、pip、pip3等命令。

# 镜像的创建和修改

## 创建镜像

从软件说明来看，镜像层次较少，通常可以按需制作，下面介绍镜像制作过程，当业务上需要支持更多的算法库或者要修改软件版本，则可以快速手动修改镜像制作文件来创建合适的编程环境。



在[ml-center\_v1.0.zip]文件中，其文件结构如图 1所示：



图 1 镜像文件目录结构

使用build.sh来创建镜像，该脚本主要是用于创建一个AI库镜像，其中代码就仅有一行：

|  |
| --- |
| docker build -t ml-center:1.0 . |

如果是要执行该脚本，请在宿主机上将build.sh加上可执行权限。如图 2所示。

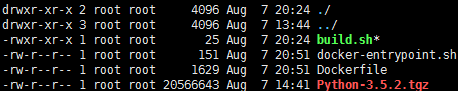


图 2 文件的权限信息

当执行build.sh后就能够发现多出了一个ml-center的镜像，且镜像版本为1.0（镜像版本号会更新，请注意）。如图 3所示。

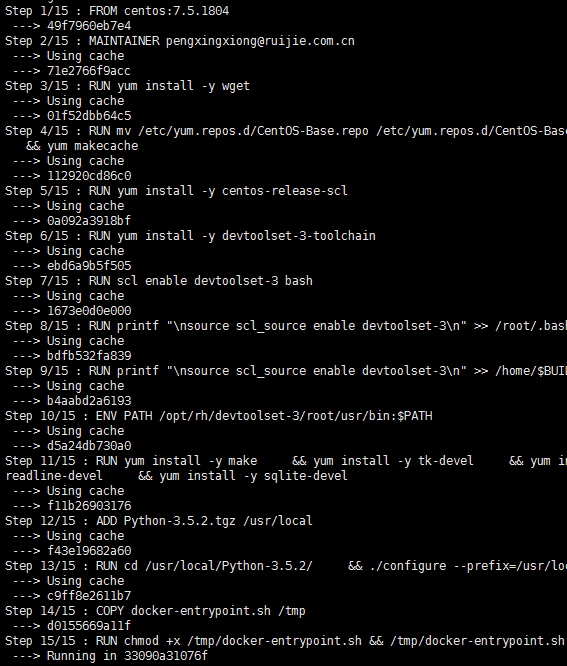


图 3 镜像制作中的分层（分步骤）

镜像共有15个层，最后第15步骤是调用docker-entrypoint.sh来下载最新的python的依赖库。该步骤所耗时间依网络环境而定。镜像创建成功后会产生如图 4所示的提示。

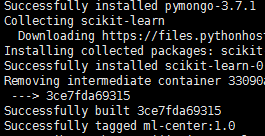


图 4 创建镜像成功的提示

最终可以使用docker命令查看创建的镜像是否存在，如图 5所示。



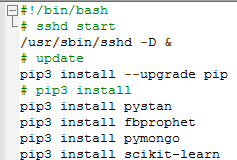
图 5 用docker命令查看创建的镜像

使用镜像

## 修改镜像

通常情况下，当缺少依赖的工具时，直接从网上下载工具或者使用pip命令进行依赖。但是对于镜像而言，每次重启后，就会初始化，将仅仅下载初始的依赖工具。因此有必要将需要的依赖工具集成到镜像中。

打开docker-entrypoint.sh脚本文件，可以看到已经存在部分初始化内容：



当需要加入新的依赖时，可以简单的使用 pip3 install 命令。

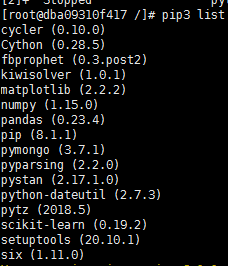
当镜像修改完成后，可以选择替换原有

# 部署方式

# 算法说明

（1）平台提供环境现集成了scikit-learn和fbprophet的算法库，运行环境为单机版，以容器镜像方式提供，配合环境使用文档了算法库列表。

（2）镜像环境内包含的库及相关依赖包版本信息如下所示。



**一、scikit-learn**

**1、分类**

支持向量机（SVM），最近邻，逻辑回归，随机森林，决策树以及多层感知器（MLP）神经网络

**2、回归**

支持向量回归（SVR），脊回归，Lasso回归，弹性网络（Elastic Net），最小角回归（LARS ），贝叶斯回归，以及各种不同的鲁棒回归算法

**3、[聚类](http://sklearn.apachecn.org/cn/0.19.0/modules/clustering.html" \l "clustering)**

K-均值聚类，谱聚类，均值偏移，分层聚类，DBSCAN聚类。

**4、[降维](http://sklearn.apachecn.org/cn/0.19.0/modules/decomposition.html" \l "decompositions)**

主成分分析（PCA）、非负矩阵分解（NMF）、特征选择

**5、[模型选择](http://sklearn.apachecn.org/cn/0.19.0/model_selection.html" \l "model-selection)**

格点搜索，交叉验证和各种针对预测误差评估的度量函数

**6、[预处理](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html" \l "preprocessing)**

标准化、[非线性转换](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#preprocessing-transformer)、[归一化](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#preprocessing-normalization)、[二值化](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#preprocessing-binarization)、[分类特征编码](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#preprocessing-categorical-features)、[缺失值插补](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#imputation)、[生成多项式特征](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#polynomial-features)、[自定义转换器](http://sklearn.apachecn.org/cn/0.19.0/modules/preprocessing.html#function-transformer)。

关于scikit-learn的算法库内容，更详细文档可参考以下链接：

<http://sklearn.apachecn.org/cn/0.19.0/documentation.html>

**二、fbprophet**

时间序列的预测算法

# FAQ

## 镜像制作失败

镜像制作失败，通常原因是网络环境导致的，因为需要拉取gcc等软件的最新版本，从国外网站下载可能由于网速太慢而导致下载超时，建议换其他网络环境或者在其他时间段重新制作。