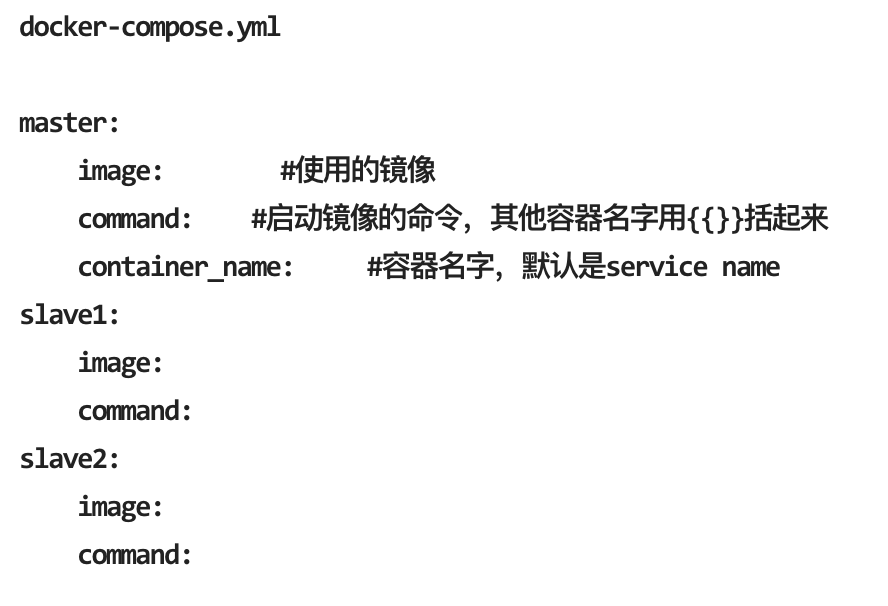
1. 登录系统
   1. <http://nap.artemisprojects.org>
   2. 用户名：密码：目前还不开放用户注册
2. 项目管理
   1. 查看项目列表，web项目（ip，端口号）
   2. 查看项目日志：由于运行在服务器上，输出到标准输出stdout和标准错误stderr的信息不会直接显示出来，而是收集到日志文件中。可以在项目日志中查看这些信息。
   3. 进入项目shell
   4. 删除项目
3. 项目描述文件compose.yml格式和示例
   1. compose.yml。。。

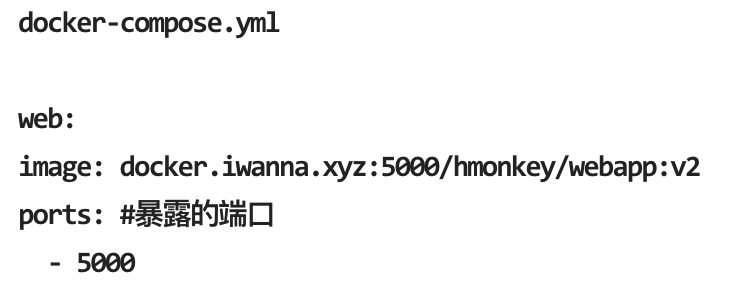
和docker compose文件类似，但是不支持network、volumes-from参数

* 1. mpi示例

<https://github.com/monkey-H/mpi_app>



* 1. web示例

<https://github.com/monkey-H/web_app>****

1. 存储服务和文件管理
   1. 对应每个项目自动生成一个文件夹，已经将项目的git文件clone到了这个文件夹中
   2. 这个文件夹是一个分布式文件系统上的共享目录，比直接在docker内存储文件的可靠性高。每个容器启动后会自动挂载到的/data目录下。
   3. 可以新建、删除、编辑必要的文件
   4. 删除项目后，这些文件和会被一并删除！
2. 网络环境
   1. 多个容器在同一个私有的网络中，彼此可以直接通过ip地址或机器名来通信。机器名可以在compose.yml中预先指定。
   2. 对web类型的应用项目，会自动绑定一个公网ip和随机端口，由于ip资源有限，目前还不支持浮动ip或域名绑定。
3. 镜像仓库
   1. 私有仓库url，提供ubuntu、centos、jdk、python等常用基础镜像
   2. 由于学校网络限制，服务器不能直接连接外网，故不能提供镜像构建功能。
   3. 可以基于docker公共镜像仓库或私有仓库的基础镜像（私有仓库速度快一些）在客户端build好镜像，然后push到私有镜像
   4. push命令：
   5. 需将镜像push到私有仓库后才能在compose.yml中使用该镜像，镜像名格式为hub.iwanna.xyz:5000/镜像名