

1. 登录系统
 - a) <http://nap.artemisprojects.org>
 - b) 用户名: 密码: 目前还不开放用户注册
2. 项目管理
 - a) 查看项目列表, web 项目 (ip, 端口号)
 - b) 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。
 - c) 进入项目 shell
 - d) 删除项目
3. 项目描述文件 `compose.yml` 格式和示例
 - a) `compose.yml`。。。和 `docker compose` 文件类似, 但是不支持 `network`、`volumes-from` 参数
 - b) mpi 示例
https://github.com/monkey-H/mpi_app

docker-compose.yml

master:

image: #使用的镜像

command: #启动镜像的命令, 其他容器名字用{{}}括起来

container_name: #容器名字, 默认是service name

slave1:

image:

command:

slave2:

image:

command:

- c) web 示例

https://github.com/monkey-H/web_app

docker-compose.yml

web:

image: `docker.iwanna.xyz:5000/hmonkey/webapp:v2`

ports: #暴露的端口

- 5000

4. 存储服务 and 文件管理

- a) 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
- b) 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。
- c) 可以新建、删除、编辑必要的文件
- d) 删除项目后，这些文件和会被一并删除！

5. 网络环境

- a) 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。
- b) 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。

6. 镜像仓库

- a) 私有仓库 url，提供 ubuntu、centos、jdk、python 等常用基础镜像
- b) 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。
- c) 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
- d) push 命令：
- e) 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