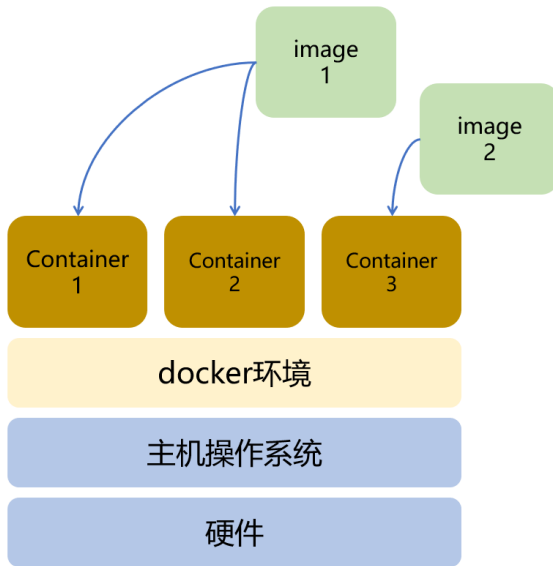


docker 基础 (1)



【操作步骤】

1. 安装 docker 环境

下载安装脚本: `curl -fsSL https://get.docker.com -o get-docker.sh`

执行安装脚本: `sudo sh get-docker.sh`

将操作用户添加加入 docker 组: `sudo usermod -aG docker ubuntu`

查看 docker 环境信息: `docker version; docker info`

2. docker 镜像

<https://hub.docker.com/>

下载 debian image 最新版: `docker pull debian`

下载 python:3.5 的 image: `docker pull python:3.5`

下载 python image 的最新版: `docker pull python`

列出本地的 image: `docker image ls`

3. docker 容器

生成容器并运行: `docker run hello-world`

生成容器并运行时, 指定容器的名字: `docker -name myapp hello-world`

启动容器 (不生成): `docker start xxxxx`

列出系统中的容器 (1): `docker container ls -a`

列出系统中的容器 (2): `docker ps -a`

4. 删除容器与镜像

删除容器: `docker rm xxxxx`

删除镜像: `docker rmi hello-world`

【参考】

- docker 命令参考

<https://docs.docker.com/engine/reference/commandline/cli/>

docker 基础 (2)

【操作步骤】

1. docker 容器主程序

生成新的容器，更改其主程序：`docker run --name p1 python echo abcd`

生成新的容器，增加容器的标准输入：`docker run --name p2 -it python`

2. attach 与 detach

启动已有容器 p1：`docker start p1`

启动已有容器 p1，进入 attach 状态：`docker start -a p1`

启动已有容器 p2：`docker start p2`

容器 p2 进入 attach 状态：`docker attach p2`

停止容器 p2：`docker stop p2`

启动容器 p2，进入 attach 状态：`docker start -ai p2`

以 detach 方式生成并启动容器 p3：`docker run --name p3 -itd python`

3. 在容器中执行命令

执行非交互式的命令：`docker exec p3 uname`

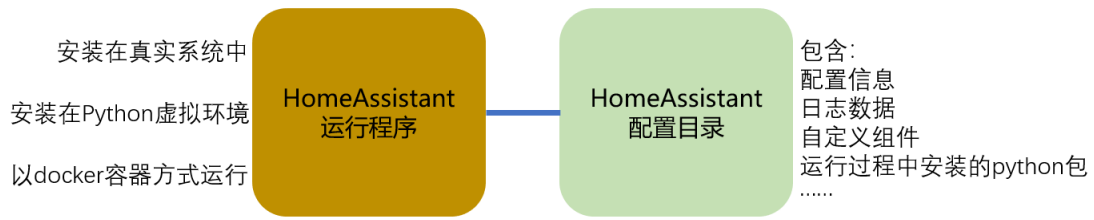
执行交互式的命令：`docker exec -it p3 bash`

【参考】

- docker 命令参考

<https://docs.docker.com/engine/reference/commandline/cli/>

HomeAssistant 的 docker 安装



【操作步骤】

1. 找到合适的 HomeAssistant docker 镜像
2. 下载 HomeAssistant 镜像（可省略）
3. 启动 HomeAssistant 容器

```
docker run --name="home-assistant" -d --init --restart always \  
-v /etc/ha_config:/config -v /etc/localtime:/etc/localtime:ro \  
--net=host homeassistant/home-assistant:latest
```

4. 访问
5. 在容器中访问主机硬件

在 docker run 命令中，增加参数 `--device /dev/ttyUSB0:/dev/ttyUSB0`

【参考】

- hub.docker 中的 HomeAssistant 镜像
<https://hub.docker.com/search?q=homeassistant&type=image>
- 在 docker 中安装 HomeAssistant
<https://www.home-assistant.io/docs/installation/docker/>

HomeAssistant docker 容器的典型使用

docker容器
HomeAssistant

主程序:

```
python -m homeassistant --config /config
```

HA配置目录:

容器中的/config

也就是主机上的/etc/ha_config

【操作步骤】

1. 配置

编辑配置目录下的 configuration.yaml 文件

sensor:

```
- platform: bitcoin
  display_options:
    - exchangerate
    - trade_volume_btc
```

2. 重启

```
docker restart home-assistant
```

3. 查错

前端菜单: 开发者工具-info

日志文件: 配置目录下的 home-assistant.log

attach 后看标准输出: `docker attach home-assistant` (需要 kill -9 进程号退出)

4. 升级

```
docker stop home-assistant
```

```
docker rm home-assistant
```

```
docker pull homeassistant/home-assistant:latest
```

```
docker run --name="home-assistant" -d --init --restart always \
  -v /etc/ha_config:/config -v /etc/localtime:/etc/localtime:ro \
  --net=host homeassistant/home-assistant:latest
```

5. 容器的典型使用方式与备份

备份配置目录: `sudo tar -czvf ha_config.tar.gz /etc/ha_config`

恢复配置目录: `sudo tar -xzvf ha_config.tar.gz -C /`

HomeAssistant docker 容器的非典型使用

【操作步骤】

1. 运行容器中的命令

查看容器内进程: `docker exec home-assistant ps -ef`

生成 abc.txt 文件: `docker exec home-assistant touch abc.txt`

2. 进入容器的 bash

`docker exec -it home-assistant bash`

3. 软件包安装 apk

`apk update`

`apk add vim`

4. pip 安装

配置国内镜像

```
vi /etc/pip.conf
```

内容:

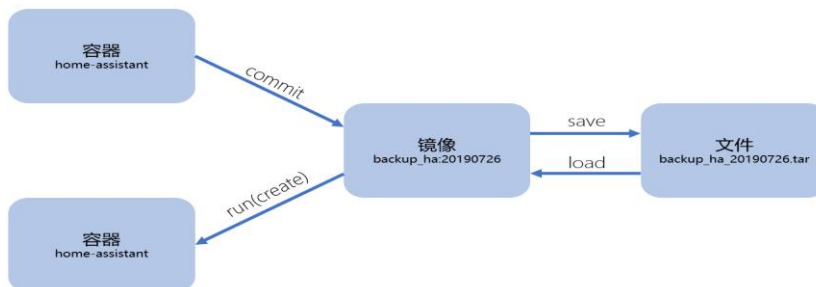
```
[global]
```

```
index-url=https://mirrors.aliyun.com/pypi/simple/
```

安装 python 包

```
pip install esptool
```

5. 容器的备份与恢复



从容器生成镜像: `docker commit home-assistant backup_ha:20190726`

从镜像生成文件: `docker save -o backup_ha_20190726.tar backup_ha:20190726`

从文件导入镜像: `docker load --input backup_ha_20190726.tar`

从镜像生成容器:

```
docker run --name="recovery-ha" -d --init --restart always \
-v /etc/ha_config:/config -v /etc/localtime:/etc/localtime:ro \
--net=host backup_ha:20190726
```

6. 使用 Dockerfile 生成新的镜像

Dockerfile 文件

```
FROM homeassistant/home-assistant:latest
```

```
COPY pip.conf /etc/pip.conf
```

```
RUN apt-get update -y
```

```
RUN apt-get install vim nano -y
```

```
RUN pip install --no-cache-dir esptool
```

生成镜像命令

```
docker build --tag=my_customize_ha:20190726 .
```

【参考】

- docker 官网
<https://www.docker.com/>
- Dockerfile 参考
<https://docs.docker.com/engine/reference/builder/>