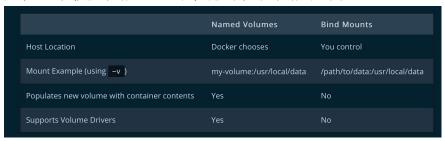
6 Use bind mounts

2021年4月18日 星期日 下午10:20

使用绑定挂载,可以控制主机上的确切挂载点,可使用它保留数据,向其他容器提供数据。在处理应用程序时,我们可以使用绑定挂载将源代码挂载到容器中,以使其查看代码更改,做出响应并立即查看更改。

快速的卷类型比较

绑定安装Bind mounts和命名卷named volumes是Docker引擎随附的两种主要卷类型。但 是,可以使用其他卷驱动程序来支持其他用例。



开启一个dev模式容器

要运行我们的容器以支持开发工作流程,做如下步骤:

- 1.挂载源代码到容器里
- 2.安装所有依赖,包括dev依赖
- 3.开启nodemon去观测文件系统的改变

接下来开始做

- 1.确保没有运行状态下的getting-started容器
- 2.运行下列命令(注意必须进入app目录)

docker run -dp 3000:3000 \

-w /app -v "\$(pwd):/app" \

node:12-alpine \

sh -c "yarn install && yarn run dev"

注释:

- -w /app 设置工作目录
- -v "\$(pwd):/app"绑定容器主机的当前目录到/app目录

node:12-alpine 使用的镜像,这是从Dockerfile的基础镜像

- sh -c "yarn install && yarn run dev" 我们使用 sh 启动一个 shell,并运行 yarn install 来安装所有依赖项,然后运行 yarn run dev。如果我们查看 package.json,我们会看到 dev 脚本正在启动 nodemon
- 3.使用docker logs -f < container-id> 查看日志

```
[enzeyu@enzeyudeMacBook-Pro app % docker logs -f ecd
varn install v1.22.5
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.9: The platform "linux" is incompatible with this module. info "fsevents@1.2.9" is an optional dependency and failed compatibility check.
Excluding it from installation.
[3/4] Linking dependencies..
[4/4] Building fresh packages...
Done in 143.72s.
varn run v1.22.5
$ nodemon src/index.is
[nodemon] 1.19.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] starting `node src/index.js`
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
```

4.接下来对app进行改动,在src/static/js/app.js里,修改第109行的Add Item 改为 Add 5.刷新网页即可以发现改动,查看docker日志可以发现改动

```
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
```

6.如果结束更改,停止容器并建立新的镜像,使用 docker build -t getting-started.

```
enzeyu@enzeyudeMacBook-Pro app % docker build -t getting-started .
[+] Building 107.6s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
                                                                              0.05
=> => transferring dockerfile: 235B
                                                                              0.0s
=> [internal] load .dockerignore
=> => transferring context: 2B
                                                                              0.0s
=> [internal] load metadata for docker.io/library/node:12-alpine
=> [1/5] FROM docker.io/library/node:12-alpine
=> [internal] load build context
 => => transferring context: 54.65MB
=> CACHED [2/5] RUN apk add --no-cache python g++ make
                                                                              0.0s
=> CACHED [3/5] WORKDIR /app
=> [4/5] COPY .
                                                                              0.7s
=> [5/5] RUN yarn install --production
                                                                            104 15
=> exporting to image
                                                                              1.3s
=> => exporting layers
                                                                              1.3s
=> => writing image sha256:610906f0c28873444f2263200a2f85b7e957c2b2abb6e
                                                                              0.0s
=> => naming to docker.io/library/getting-started
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and 1
earn how to fix them
```

-t表示标签 docker build即使用当前目录的Dockerfile建立镜像

使用绑定挂载对于本地开发设置非常常见。优点是开发机器不需要安装所有的构建工具和环境。使用单个 docker run 命令,开发环境就被拉取并准备就绪。