

iot 及环保构建说明

AUTHOR: 彭玲 TIME: 2022/12/28

iot 及环保构建说明

总述

iot web 构建

Dockerfile

jenkinsfile

环保构建

web 端构建

Dockerfile

jenkinsfile

api 端构建

Dockerfile

jenkinsfile

总述

iot web 端构建与环保类似，搞明白一个，另一个也就迎刃而解了。其中，关键点如下：

```
1 // server/index.js
2 app.use(express.static('./static/'));
```

指定了服务器静态资源目录为 'static'，所以，后续构建步骤围绕该点进行，详见 Dockerfile。

iot web 构建

Dockerfile

```
1 FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12-dev as builder
2 COPY . /var/app
3 WORKDIR /var/app
4 EXPOSE 9080
5 RUN npm config set registry=http://10.8.30.22:7000
6 RUN npm install
7 RUN npm run dll:console && npm run build:console
8 RUN mkdir -p static/dist && cp -r dist/assets static/dist && cp -r dist/console/*
   static
```

```

9  RUN rm -rf app build dist node_modules src webpack_cache .babelrc .eslintrc
   Dockerfile jenkinsfile package.json package-lock.json README.md
10  RUN cp server/* . && rm -rf server
11  RUN npm install --production
12
13  FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12
14  COPY --from=builder --chown=node /var/app /home/web
15  WORKDIR /home/web
16  CMD ["node","index.js"]
17

```

jenkinsfile

```

1  pipeline {
2      agent {
3          node {
4              label 'jnlp-slave'
5          }
6      }
7
8      stages {
9          stage('Testing iota console web.....') {
10             steps {
11                 buildName '#${BUILD_NUMBER} ~/iota/${JOB_NAME}:${IMAGE_VERSION}'
12                 buildDescription
13                 'harbor.anxinyun.cn/iota/${JOB_NAME}:${IMAGE_VERSION}'
14                 sh 'nerdctl build -t
15                 harbor.anxinyun.cn/iota/${JOB_NAME}:${IMAGE_VERSION} .'
16                 sh 'nerdctl push
17                 harbor.anxinyun.cn/iota/${JOB_NAME}:${IMAGE_VERSION}'
18             }
19         }
20     }
21 }

```

环保构建

web 端构建

Dockerfile

```

1  FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12-dev as builder
2  COPY . /var/app
3  WORKDIR /var/app
4  EXPOSE 9080
5  RUN npm config set registry=http://10.8.30.22:7000
6  RUN npm cache clean -f
7  RUN npm install
8  RUN npm run dll:fs-env && npm run build:fs-env

```

```

9  RUN mkdir -p static/dist \
10     && cp -r dist/fs-env/* static && mv static/assets static/dist && mv static/font
    static/dist
11  RUN rm -rf app build dist node_modules src webpack_cache .babelrc .eslintrc
    Dockerfile jenkinsfile package.json package-lock.json README.md
12  RUN cp server/* . && rm -rf server
13  RUN npm install --production
14
15  FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12
16  COPY --from=builder --chown=node /var/app /home/web
17  WORKDIR /home/web
18  CMD ["node","index.js"]
19

```

jenkinsfile

```

1  pipeline {
2      agent {
3          node {
4              label 'jnlp-slave'
5          }
6      }
7
8      stages {
9          stage('Testing env web.....') {
10             steps {
11                 buildName '#${BUILD_NUMBER} ~/env/${JOB_NAME}:${IMAGE_VERSION}'
12                 buildDescription
13                 'harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION}'
14                 sh 'nerdctl build -t
    harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION} .'
15                 sh 'nerdctl push harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION}'
16             }
17         }
18     }
19 }

```

api 端构建

Dockerfile

```
1 FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12-dev as builder
2 COPY . /var/app
3 WORKDIR /var/app
4 EXPOSE 9094
5 RUN npm install --production --registry http://10.8.30.22:7000
6
7 FROM registry.cn-hangzhou.aliyuncs.com/fs-devops/node:12
8 COPY --from=builder --chown=node /var/app /home/api
9 WORKDIR /home/api
10 CMD ["node","index.js"]
11
```

jenkinsfile

```
1 pipeline {
2     agent {
3         node {
4             label 'jnlp-slave'
5         }
6     }
7
8     stages {
9         stage('Testing env service-product.....') {
10            steps {
11                buildName '#${BUILD_NUMBER} ~/env/${JOB_NAME}:${IMAGE_VERSION}'
12                buildDescription
13                'harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION}'
14                sh 'nerdctl build -t
15                harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION} .'
16                sh 'nerdctl push harbor.anxinyun.cn/env/${JOB_NAME}:${IMAGE_VERSION}'
17            }
18        }
19    }
20 }
```