

十、Jenkins构建前端项目（分布式）

1.新建item/新建任务

2.配置item/任务

3. nginx配置文件

4.确认配置文件是否正确

十、Jenkins构建前端项目（分布式）

1.新建item/新建任务


选择“Freestyle project（自由风格的软件项目）”，起名：manstro-web


输入一个任务名称

manstro-web

» 必填项

manstro-web

**构建一个自由风格的软件项目**
这是Jenkins的主要功能.Jenkins将会结合任何SCM和任何构建系统来构建你的项目,甚至可以构建软件以外的系统.

**构建一个maven项目**
构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.

2.配置item/任务

1】 Discard old builds/丢弃旧的构建

提供了“Discard old builds”的选项可以配置保留几天/最多几个的build 和 artifacts，缺省这个选项不会勾上。不勾选时，默认值为365，显然这个值过大了。

☒ 丢弃旧的构建 ?

策略

Log Rotation

保持构建的天数

如果非空，构建记录将保存此天数

7

保持构建的最大个数

如果非空，最多此数目的构建记录将被保存

5

发布包保留天数

如果非空，比此早的发布包将被删除，但构建的日志、操作历史、报告等将被保留

7

发布包最大保留#个构建

如果非空，最多此数目大构建将保留他们的发布包

5

2】参数化构建过程

☒ 参数化构建过程 ?

添加参数 ▲

布尔值参数

选项参数

凭据参数

文件参数

Git 参数

List Subversion tags (and more)

文本参数

密码参数

运行时参数

字符参数

如果没有 “Git 参数”，需要安装Git Parameter插件

【系统管理】==>【管理插件】==>搜索Git Parameter插件==>点击 install without restart

1) 添加Git参数: branch (必配)

名称: branch

描述: git分支

参数类型: 分支或标签

默认值: */master

2) 添加字符参数: host_port (必配)

名称: host_port

默认值: 8098

描述: 服务端口 (宿主机暴露的端口)

3) 添加选项参数: profile_flag (必配)

名称: profile_flag

选项:

prod

staging

描述: 环境变量

4) 添加字符参数: images_name

名称: images_name

默认值: nginx

描述: 镜像名称

5) 添加字符参数: container_name

名称: container_name

默认值: nginx

描述: 容器名称

6) 添加字符参数: ssh_dist_file

名称: ssh_dist_file

默认值: /docker/nginx/html/manstro/web

描述: dist包文件路径

3】源码管理

<http://10.0.0.251/research/manstro-base-framework/manstro-web.git>

\$branch

源码管理

☐ 无

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

Branches to build ?

指定分支 (为空时代表any) ?

刚才定义的git参数

4】构建环境

选择: Provide Node & npm bin/ folder to PATH

Cache location (缓存位置) : Local to the workspace

构建环境

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Provide Configuration files ?

☐ Send files or execute commands over SSH before the build starts ?

☐ Send files or execute commands over SSH after the build runs ?

☐ Execute shell script on remote host using ssh ?

☒ Provide Node & npm bin/ folder to PATH

NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

npmrc file

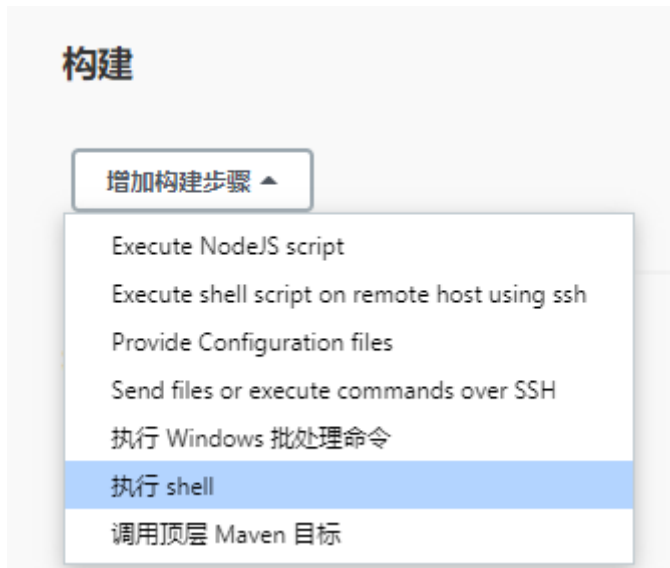
Cache location

5】构建

1) 执行shell

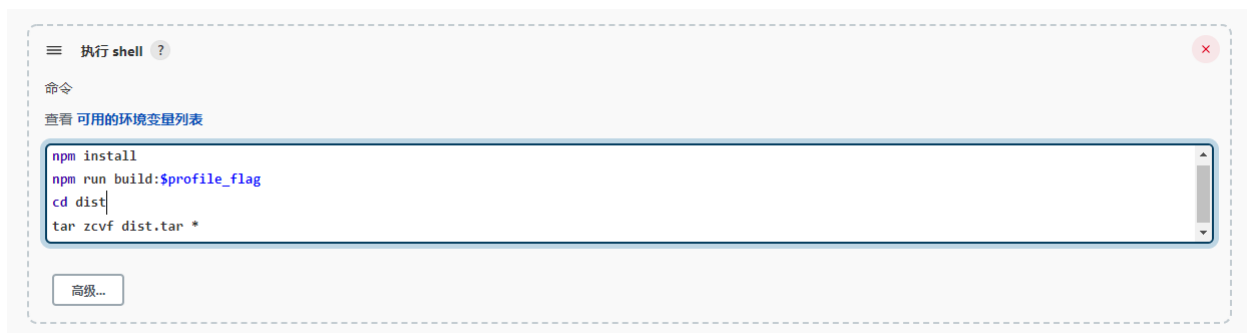
点击：Add post-build step（增加构建步骤）按钮

选择：执行 shell



对前端项目dist文件夹进行压缩打包，输入：

```
1 npm i --legacy-peer-deps
2 npm install
3 npm run build:$profile_flag
4 cd dist
5 tar zcvf dist.tar *
```



备注：开头写npm i --legacy-peer-deps

是因为遇到报错：npm ERR! ERESOLVE unable to resolve dependency tree

原因是：

升级 npm 至 v7.6.2 后, 运行 npm i 安装依赖, 报错 ERESOLVE unable to resolve dependency tree

等依赖项中存在无法解决的冲突, npm@7 现在尝试安装它们, 而npm@6没有。

解决方案：

使用 npm i --legacy-peer-deps

2) Send files or execute commands over SSH (上传文件到远程服务器)

点击：Add post-build step (增加构建步骤) 按钮

选择：Send files or execute commands over SSH

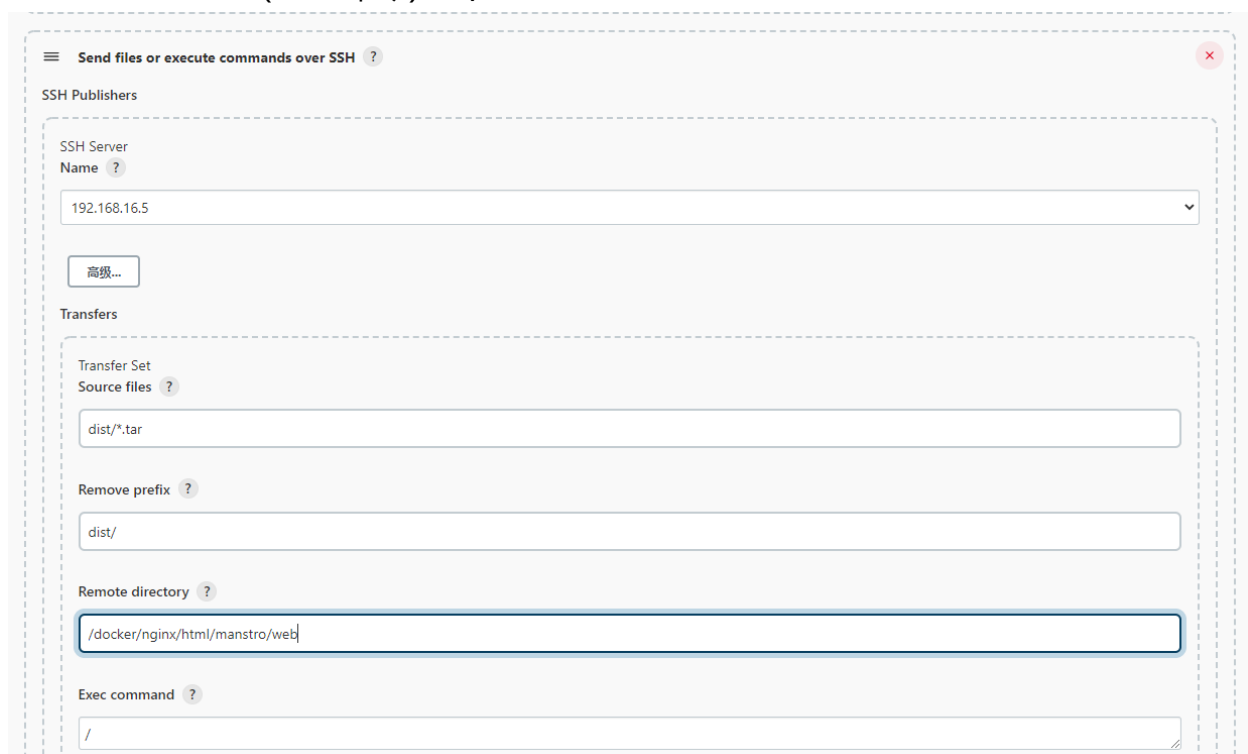
SSH Server: 选择之前配置ssh地址

Source files (准备上传的源文件) : dist/*.tar

Remove prefix (删除前缀) : dist/

Remote directory (远程目录, 和上面配置的ssh_dist_file保持一致) : /docker/nginx/html/manstro/web

Exec command (Exec命令) : /



3) Execute shell script on remote host using ssh (在远程服务器上执行脚本)

点击：Add post-build step (增加构建步骤) 按钮

选择：Execute shell script on remote host using ssh

Command输入:

```
1 echo '=====1.判断容器是否运行====='
```

```
2 if [[ -n $(docker ps -a | grep $container_name) ]];then
```

```
3 echo '=====1.1容器正在运行====='
```

```
4 echo '=====1.2停止容器====='
```

```
5 docker stop $container_name
```

```

6  echo '=====1.3强制删除容器===== '
7  docker rm -f $container_name
8  else
9  echo '=====1.5容器未运行===== '
10 fi
11 echo '=====2.进入dist包文件夹===== '
12 if [ ! -d "$ssh_dist_file" ]; then
13  mkdir -p $ssh_dist_file
14 fi
15 cd $ssh_dist_file
16 echo '=====3.解压并删除压缩包===== '
17 tar zxvf dist.tar
18 rm -rf dist.tar
19 echo '=====4.启动容器===== '
20 docker run --restart=always -d -p $host_port:8098 -v
   /docker/nginx/html:/usr/share/nginx/html -v
   /docker/nginx/conf/nginx.conf:/etc/nginx/nginx.conf -v
   /docker/nginx/logs:/var/log/nginx --name $container_name $images_name:lates
t

```

备注：这里建议映射端口两边保持一致

3. nginx配置文件

重点配置如下

```

location /manstro/web {
    gzip on; #开启压缩
    gzip_http_version 1.1; #协议版本配置
    gzip_min_length 50k; #低于5kb的资源不压缩
    gzip_comp_level 1; #压缩等级
    gzip_types text/plain application/xml application/javascript text/css
image/png application/json; #需要压缩的MIME类型
    gzip_disable "MSIE [1-6]\."; #配置禁用gzip条件，支持正则。此处表示ie6及以下不
启用gzip（因为ie低版本不支持）
    gzip_vary on; #是否添加 "Vary: Accept-Encoding" 响应头
    root /usr/share/nginx/html; #启动容器挂载宿主机的项目根路径
    try_files $uri $uri/ /index.html; # $uri/ 指代home文件，也就是p地址后面的路径
    index index.html;
}

```

```

location /manstro/prod-api/{
    gzip on; #开启压缩
    gzip_http_version 1.1; #协议版本配置
    gzip_min_length 50k; #低于5kb的资源不压缩
    gzip_comp_level 1; #压缩等级
    gzip_types text/plain application/xml application/javascript text/css
image/png application/json; #需要压缩的MIME类型
    gzip_disable "MSIE [1-6]\."; #配置禁用gzip条件，支持正则。此处表示ie6及以下不
启用gzip（因为ie低版本不支持）
    gzip_vary on; #是否添加 “Vary: Accept-Encoding” 响应头
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header REMOTE-HOST $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_pass http://192.168.16.5:8080/; #gateway网关地址
}

```

nginx配置如下：

```

1 user nginx;
2 worker_processes auto;
3
4 error_log /var/log/nginx/error.log notice;
5 pid /var/run/nginx.pid;
6
7
8 events {
9     worker_connections 1024;
10 }
11
12
13 http {
14     include /etc/nginx/mime.types;
15     default_type application/octet-stream;
16
17     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
18     '$status $body_bytes_sent "$http_referer" '

```

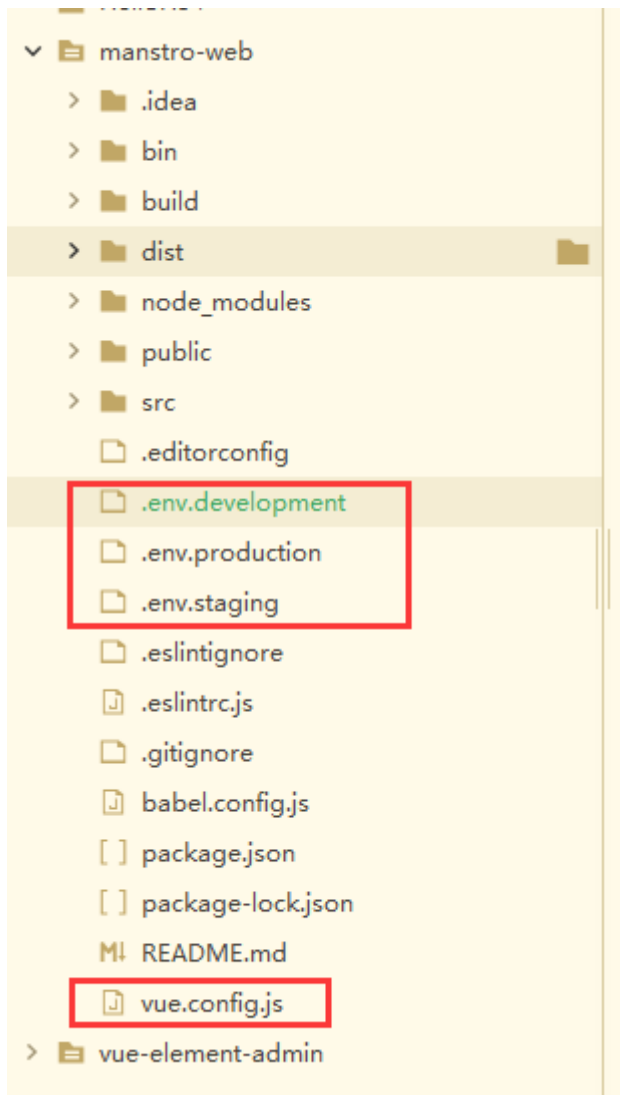


```
19  "$http_user_agent" "$http_x_forwarded_for";
20
21  access_log /var/log/nginx/access.log main;
22
23  sendfile on;
24  #tcp_nopush on;
25
26  keepalive_timeout 65;
27
28  server {
29  listen 8098;
30  server_name localhost;
31
32  #charset koi8-r;
33
34  #access_log logs/host.access.log main;
35  location / {
36  try_files $uri $uri/ /index.html;
37  }
38
39  location /manstro/web {
40  gzip on; #开启压缩
41  gzip_http_version 1.1; #协议版本配置
42  gzip_min_length 50k; #低于5kb的资源不压缩
43  gzip_comp_level 1; #压缩等级
44  gzip_types text/plain application/xml application/javascript text/css i
image/png application/json; #需要压缩的MIME类型
45  gzip_disable "MSIE [1-6]\."; #配置禁用gzip条件，支持正则。此处表示ie6及以下
不启用gzip（因为ie低版本不支持）
46  gzip_vary on; #是否添加“Vary: Accept-Encoding”响应头
47  root /usr/share/nginx/html;
48  try_files $uri $uri/ /index.html;
49  index index.html;
50  }
51
52  location /manstro/prod-api/{
53  gzip on; #开启压缩
54  gzip_http_version 1.1; #协议版本配置
55  gzip_min_length 50k; #低于5kb的资源不压缩
56  gzip_comp_level 1; #压缩等级
```

```
57  gzip_types text/plain application/xml application/javascript text/css i
mage/png application/json; #需要压缩的MIME类型
58  gzip_disable "MSIE [1-6]\."; #配置禁用gzip条件，支持正则。此处表示ie6及以下
不启用gzip（因为ie低版本不支持）
59  gzip_vary on; #是否添加“Vary: Accept-Encoding”响应头
60  proxy_set_header Host $http_host;
61  proxy_set_header X-Real-IP $remote_addr;
62  proxy_set_header REMOTE-HOST $remote_addr;
63  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
64  proxy_pass http://192.168.16.5:8080/;
65  }
66
67  error_page 500 502 503 504 /50x.html;
68  location = /50x.html {
69  root html;
70  }
71  }
72  }
```

4. 确认配置文件是否正确

1) 确认存在问题的配置文件



vue.config.js

需要修改为gateway网关的ip
和 静态资源访问路径

```

10
11 const port = process.env.port || process.env.npm_config_port || 80 // 端口
12
13 // vue.config.js 配置说明
14 //官方vue.config.js 参考文档 https://cli.vuejs.org/zh/config/#css-loaderoptions
15 // 这里只列一部分，具体配置参考文档
16 module.exports = {
17   // 部署生产环境和开发环境下的URL。
18   // 默认情况下，Vue CLI 会假设你的应用是被部署在一个域名的根路径上
19   // 例如 https://www.ruoyi.vip/。如果应用被部署在一个子路径上，你就需要用这个选项指定这个子路径。例如，如果你的应用被
20   publicPath: process.env.NODE_ENV === "production" ? "/" : "/",
21   // 在npm run build 或 yarn build 时，生成文件的目录名称（要和baseUrl的生产环境路径一致）（默认dist）
22   outputDir: 'dist',
23   // 用于放置生成的静态资源（js、css、img、fonts）的；（项目打包之后，静态资源会放在这个文件夹下）
24   assetsDir: 'static',
25   // 是否开启eslint保存检测，有效值：ture | false | 'error'
26   lintOnSave: process.env.NODE_ENV === 'development',
27   // 如果你不需要生产环境的 source map，可以将其设置为 false 以加速生产环境构建。
28   productionSourceMap: false,
29   // webpack-dev-server 相关配置
30   devServer: {
31     host: '0.0.0.0',
32     port: port,
33     open: true,
34     proxy: {
35       // detail: https://cli.vuejs.org/config/#devserver-proxy
36       [process.env.VUE_APP_BASE_API]: {
37         target: `http://localhost:8080`,
38         changeOrigin: true,
39         pathRewrite: {
40           ['^' + process.env.VUE_APP_BASE_API]: ''
41         }
42       }
43     },
44     disableHostCheck: true
45   },
46   configureWebpack: {

```

部署项目后会发现访问静态资源的时候，没有项目路径ip后面直接就是static文件夹，没有项目名。

假如项目在nginx配置root根路径后的路径为/manstro/web 这里配置为"prodeuction" ? "/manstro/web/" : "/"

改成gateway网关

.env.production和.env.development和.env.staging

在nginx.conf文件中，配置的 location /manstro/prod-api/，所以这里也要改成/manstro/prod-api

package.json	vue.config.js	.env.production
1 # 生产环境配置		1 ENV = 'production'
2		2
3		3
4 # 万洲管理系统/生产环境		4
5		5 VUE_APP_BASE_API = '/prod-api'
6		6

src/router/index.js

需要增加路由的base路径，/manstro/web，和nginx配置项目监听地址保持一致

```
export default new Router({
  mode: 'history', // 去掉url中的#
  base: "/manstro/web", // 需要增加路由的base路径
  scrollBehavior: () => ({ y: 0 }),
  routes: constantRoutes
})
```

src/layout/components/Navbar.vue

需要修改登出路由地址，/manstro/web，和nginx配置项目监听地址保持一致

```
async logout() {
  this.$confirm('确定注销并退出系统吗?', '提示', {
    confirmButtonText: '确定',
    cancelButtonText: '取消',
    type: 'warning'
  }).then(() => {
    this.$store.dispatch('LogOut').then(() => {
      location.href = '/manstro/web'; // 需要修改登出路由地址
    })
  })
}
```

2) 在不修改git远端配置文件的前提下，通过插件在构建项目之前替换配置文件

若没有Config File Provider插件

【系统管理】==>【管理插件】==>【可选插件】==>搜索Config File Provider插件
==>点击 install without restart

这里还是需要重启jenkins的，看安装提示。

安装成功后，系统配置这里有这个图标



选择Custom file ==> next

☐ Groovy file
a reusable groovy script

☒ Custom file
a custom file (e.g. text or any other not yet available format)

☐ Extended Email Publisher Groovy Template
A Groovy template used by the Extended Email Publisher plugin to generate emails.

☐ Extended Email Publisher Jelly Template
A Jelly template used by the Extended Email Publisher plugin to generate emails.

ID
ID of the config file

7a7f1d5d-866c-44b4-88a2-f77af82342af

Next

填写文件名和文件内容

填写配置文件的名字（这个时候的名字可以跟实际应用时的配置文件名不一致，jenkins调用配置文件覆盖源代码的时候，还会进行更名），配置文件的内容，提交保存

The configuration

ID

7a7f1d5d-866c-44b4-88a2-f77af82342af

Name

MyCustom 这里填写文件名

Comment

Tokenized Credentials

新增

Content

1 这里填写文件内容

Submit

示例1：

Name: manstro-web-vue.config.js

Content:

```
1 'use strict'
2 const path = require('path')
3 const defaultSettings = require('./src/settings.js')
4
5 function resolve(dir) {
6   return path.join(__dirname, dir)
7 }
8
```

```
9  const name = defaultSettings.title || '万洲管理系统' // 标题
10
11  const port = process.env.port || process.env.npm_config_port || 80 // 端口
12
13  // vue.config.js 配置说明
14  //官方vue.config.js 参考文档 https://cli.vuejs.org/zh/config/#css-loaderoptions
15  // 这里只列一部分，具体配置参考文档
16  module.exports = {
17    // 部署生产环境和开发环境下的URL。
18    // 默认情况下，Vue CLI 会假设你的应用是被部署在一个域名的根路径上
19    // 例如 https://www.ruoyi.vip/。如果应用被部署在一个子路径上，你就需要用这个
    // 选项指定这个子路径。例如，如果你的应用被部署在 https://www.ruoyi.vip/admin/，则
    // 设置 baseUrl 为 /admin/。
20    publicPath: process.env.NODE_ENV === "production" ? "/manstro/web" :
    "/",
21    // 在npm run build 或 yarn build 时 ，生成文件的目录名称（要和baseUrl的生产
    // 环境路径一致）（默认dist）
22    outputDir: 'dist',
23    // 用于放置生成的静态资源（js、css、img、fonts）的；（项目打包之后，静态资源
    // 会放在这个文件夹下）
24    assetsDir: 'static',
25    // 是否开启eslint保存检测，有效值：ture | false | 'error'
26    lintOnSave: process.env.NODE_ENV === 'development',
27    // 如果你不需要生产环境的 source map，可以将其设置为 false 以加速生产环境构
    // 建。
28    productionSourceMap: false,
29    // webpack-dev-server 相关配置
30    devServer: {
31      host: '0.0.0.0',
32      port: port,
33      open: true,
34      proxy: {
35        // detail: https://cli.vuejs.org/config/#devserver-proxy
36        [process.env.VUE_APP_BASE_API]: {
37          target: `http://192.168.16.5:8080`,
38          changeOrigin: true,
39          pathRewrite: {
40            ['^' + process.env.VUE_APP_BASE_API]: ''
41          }
42        }
43      }
44    }
45  }
```

```
43   },
44   disableHostCheck: true
45   },
46   configureWebpack: {
47     name: name,
48     resolve: {
49       alias: {
50         '@': resolve('src')
51       }
52     }
53   },
54   chainWebpack(config) {
55     config.plugins.delete('preload') // TODO: need test
56     config.plugins.delete('prefetch') // TODO: need test
57
58     // set svg-sprite-loader
59     config.module
60       .rule('svg')
61       .exclude.add(resolve('src/assets/icons'))
62       .end()
63     config.module
64       .rule('icons')
65       .test(/\.svg$/)
66       .include.add(resolve('src/assets/icons'))
67       .end()
68       .use('svg-sprite-loader')
69       .loader('svg-sprite-loader')
70       .options({
71         symbolId: 'icon-[name]'
72       })
73       .end()
74
75     config
76       .when(process.env.NODE_ENV !== 'development',
77         config => {
78           config
79             .plugin('ScriptExtHtmlWebpackPlugin')
80             .after('html')
81             .use('script-ext-html-webpack-plugin', [{
82               // `runtime` must same as runtimeChunk name. default is `runtime`
```



```
83 inline: /runtime\..*\.js$/
84 })
85 .end()
86 config
87 .optimization.splitChunks({
88   chunks: 'all',
89   cacheGroups: {
90     libs: {
91       name: 'chunk-libs',
92       test: /[\\/]node_modules[\\/]/,
93       priority: 10,
94       chunks: 'initial' // only package third parties that are initially dependent
95     },
96     elementUI: {
97       name: 'chunk-elementUI', // split elementUI into a single package
98       priority: 20, // the weight needs to be larger than libs and app or it will be packaged into libs or app
99       test: /[\\/]node_modules[\\/]_?element-ui(.*)/ // in order to adapt to cnpm
100     },
101     commons: {
102       name: 'chunk-commons',
103       test: resolve('src/components'), // can customize your rules
104       minChunks: 3, // minimum common number
105       priority: 5,
106       reuseExistingChunk: true
107     }
108   }
109 })
110 config.optimization.runtimeChunk('single'),
111 {
112   from: path.resolve(__dirname, './public/robots.txt'), //防爬虫文件
113   to: './', //到根目录下
114 }
115 }
116 )
117 }
118 }
```

示例2:

Name: `manstro-web-.env.development`

Content:

```
1 # 开发环境配置
2 ENV = 'development'
3
4 # 万洲管理系统/开发环境
5 VUE_APP_BASE_API = '/manstro/prod-api'
6
7 # 路由懒加载
8 VUE_CLI_BABEL_TRANSPILE_MODULES = true
```

示例3:

Name: `manstro-web-.env.production`

Content:

```
1 # 生产环境配置
2 ENV = 'production'
3
4 # 万洲管理系统/生产环境
5 VUE_APP_BASE_API = '/manstro/prod-api'
```

示例4:

Name: `manstro-web-.env.staging`

Content:

```
1 NODE_ENV = production
2
3 # 万洲环境配置
4 ENV = 'staging'
5
6 # 万洲管理系统/测试环境
7 VUE_APP_BASE_API = '/manstro/prod-api'
```

示例5:

Name: `manstro-web-src/router/index.js`

Content:

```
1 import Vue from 'vue'
2 import Router from 'vue-router'
```

```

3
4  Vue.use(Router)
5
6  /* Layout */
7  import Layout from '@layout'
8  import ParentView from '@components/ParentView';
9
10 /**
11  * Note: 路由配置项
12  *
13  * hidden: true // 当设置 true 的时候该路由不会再侧边栏出现 如401, login等页面, 或者如一些编辑页面/edit/1
14  * alwaysShow: true // 当你一个路由下面的 children 声明的路由大于1个时, 自动会变成嵌套的模式--如组件页面
15  * // 只有一个时, 会将那个子路由当做根路由显示在侧边栏--如引导页面
16  * // 若你想不管路由下面的 children 声明的个数都显示你的根路由
17  * // 你可以设置 alwaysShow: true, 这样它就会忽略之前定义的规则, 一直显示根路由
18  * redirect: noRedirect // 当设置 noRedirect 的时候该路由在面包屑导航中不可被点击
19  * name: 'router-name' // 设定路由的名字, 一定要填写不然使用<keep-alive>时会出现各种问题
20  * meta : {
21    noCache: true // 如果设置为true, 则不会被 <keep-alive> 缓存(默认 false)
22    title: 'title' // 设置该路由在侧边栏和面包屑中展示的名字
23    icon: 'svg-name' // 设置该路由的图标, 对应路径src/assets/icons/svg
24    breadcrumb: false // 如果设置为false, 则不会在breadcrumb面包屑中显示
25  }
26  */
27
28  // 公共路由
29  export const constantRoutes = [
30    {
31      path: '/redirect',
32      component: Layout,
33      hidden: true,
34      children: [
35        {
36          path: '/redirect/:path(.*)',
37          component: (resolve) => require(['@views/redirect'], resolve)
38        }
39      ]

```

```
40  },
41  {
42    path: '/login',
43    component: (resolve) => require(['@/views/login'], resolve),
44    hidden: true
45  },
46  {
47    path: '/404',
48    component: (resolve) => require(['@/views/error/404'], resolve),
49    hidden: true
50  },
51  {
52    path: '/401',
53    component: (resolve) => require(['@/views/error/401'], resolve),
54    hidden: true
55  },
56  {
57    path: '',
58    component: Layout,
59    redirect: 'index',
60    children: [
61      {
62        path: 'index',
63        component: (resolve) => require(['@/views/index'], resolve),
64        name: '首页',
65        meta: { title: '首页', icon: 'dashboard', noCache: true, affix: true }
66      }
67    ]
68  },
69  {
70    path: '/user',
71    component: Layout,
72    hidden: true,
73    redirect: 'noredirect',
74    children: [
75      {
76        path: 'profile',
77        component: (resolve) => require(['@/views/system/user/profile/index'],
78        resolve),
79        name: 'Profile',
```

```
79 meta: { title: '个人中心', icon: 'user' }
80 }
81 ]
82 },
83 {
84 path: '/dict',
85 component: Layout,
86 hidden: true,
87 children: [
88 {
89 path: 'type/data/:dictId(\\d+)',
90 component: (resolve) => require(['@/views/system/dict/data'], resolve),
91 name: 'Data',
92 meta: { title: '字典数据', icon: '' }
93 }
94 ]
95 },
96 {
97 path: '/job',
98 component: Layout,
99 hidden: true,
100 children: [
101 {
102 path: 'log',
103 component: (resolve) => require(['@/views/monitor/job/log'], resolve),
104 name: 'JobLog',
105 meta: { title: '调度日志' }
106 }
107 ]
108 },
109 {
110 path: '/gen',
111 component: Layout,
112 hidden: true,
113 children: [
114 {
115 path: 'edit/:tableId(\\d+)',
116 component: (resolve) => require(['@/views/tool/gen/editTable'],
117 resolve),
118 name: 'GenEdit',
```

```

118   meta: { title: '修改生成配置' }
119   }
120   ]
121   }
122   ]
123
124   export default new Router({
125     mode: 'history', // 去掉url中的#
126     base: "/manstro/web",
127     scrollBehavior: () => ({ y: 0 }),
128     routes: constantRoutes
129   })
130

```

示例6:

Name: [manstro-web-src/layout/components/Navbar.vue](#)

Content:

```

1  <template>
2    <div class="navbar">
3      <hamburger id="hamburger-container" :is-active="sidebar.opened" class="hamburger-container" @toggleClick="toggleSideBar" />
4
5      <breadcrumb id="breadcrumb-container" class="breadcrumb-container" />
6
7      <div class="right-menu">
8        <template v-if="device!=='mobile'">
9          <search id="header-search" class="right-menu-item" />
10
11         <!-- <el-tooltip content="源码地址" effect="dark" placement="bottom">-->
12         <!-- <ruo-yi-git id="ruoyi-git" class="right-menu-item hover-effect" />-->
13         <!-- </el-tooltip>-->
14
15         <!-- <el-tooltip content="文档地址" effect="dark" placement="bottom">-->
16         <!-- <ruo-yi-doc id="ruoyi-doc" class="right-menu-item hover-effect" />-->
17         <!-- </el-tooltip>-->
18
19         <screenfull id="screenfull" class="right-menu-item hover-effect" />
20
21         <el-tooltip content="布局大小" effect="dark" placement="bottom">

```

```
22 <size-select id="size-select" class="right-menu-item hover-effect" />
23 </el-tooltip>
24
25 </template>
26
27 <el-dropdown class="avatar-container right-menu-item hover-effect" trigger="click">
28 <div class="avatar-wrapper">
29 
30 <i class="el-icon-caret-bottom" />
31 </div>
32 <el-dropdown-menu slot="dropdown">
33 <router-link to="/user/profile">
34 <el-dropdown-item>个人中心</el-dropdown-item>
35 </router-link>
36 <el-dropdown-item @click.native="setting = true">
37 <span>布局设置</span>
38 </el-dropdown-item>
39 <el-dropdown-item divided @click.native="logout">
40 <span>退出登录</span>
41 </el-dropdown-item>
42 </el-dropdown-menu>
43 </el-dropdown>
44 </div>
45 </div>
46 </template>
47
48 <script>
49 import { mapGetters } from 'vuex'
50 import Breadcrumb from '@components/Breadcrumb'
51 import Hamburger from '@components/Hamburger'
52 import Screenfull from '@components/Screenfull'
53 import SizeSelect from '@components/SizeSelect'
54 import Search from '@components/HeaderSearch'
55 import RuoYiGit from '@components/RuoYi/Git'
56 import RuoYiDoc from '@components/RuoYi/Doc'
57
58 export default {
59   components: {
60     Breadcrumb,
61     Hamburger,
```

```
62  Screenfull,
63  SizeSelect,
64  Search,
65  RuoYiGit,
66  RuoYiDoc
67  },
68  computed: {
69    ...mapGetters([
70      'sidebar',
71      'avatar',
72      'device'
73    ]),
74    setting: {
75      get() {
76        return this.$store.state.settings.showSettings
77      },
78      set(val) {
79        this.$store.dispatch('settings/changeSetting', {
80          key: 'showSettings',
81          value: val
82        })
83      }
84    },
85  },
86  methods: {
87    toggleSideBar() {
88      this.$store.dispatch('app/toggleSideBar')
89    },
90    async logout() {
91      this.$confirm('确定注销并退出系统吗？', '提示', {
92        confirmButtonText: '确定',
93        cancelButtonText: '取消',
94        type: 'warning'
95      }).then(() => {
96        this.$store.dispatch('LogOut').then(() => {
97          location.href = '/manstro/web';
98        })
99      })
100  }
101  }
```



```
102 }
103 </script>
104
105 <style lang="scss" scoped>
106 .navbar {
107   height: 50px;
108   overflow: hidden;
109   position: relative;
110   background: #fff;
111   box-shadow: 0 1px 4px rgba(0,21,41,.08);
112
113   .hamburger-container {
114     line-height: 46px;
115     height: 100%;
116     float: left;
117     cursor: pointer;
118     transition: background .3s;
119     -webkit-tap-highlight-color: transparent;
120
121     &:hover {
122       background: rgba(0, 0, 0, .025)
123     }
124   }
125
126   .breadcrumb-container {
127     float: left;
128   }
129
130   .errLog-container {
131     display: inline-block;
132     vertical-align: top;
133   }
134
135   .right-menu {
136     float: right;
137     height: 100%;
138     line-height: 50px;
139
140     &:focus {
141       outline: none;
142     }
143   }
144 }
```

```
143
144 .right-menu-item {
145   display: inline-block;
146   padding: 0 8px;
147   height: 100%;
148   font-size: 18px;
149   color: #5a5e66;
150   vertical-align: text-bottom;
151
152   &.hover-effect {
153     cursor: pointer;
154     transition: background .3s;
155
156     &:hover {
157       background: rgba(0, 0, 0, .025)
158     }
159   }
160 }
161
162 .avatar-container {
163   margin-right: 30px;
164
165   .avatar-wrapper {
166     margin-top: 5px;
167     position: relative;
168
169     .user-avatar {
170       cursor: pointer;
171       width: 40px;
172       height: 40px;
173       border-radius: 10px;
174     }
175
176     .el-icon-caret-bottom {
177       cursor: pointer;
178       position: absolute;
179       right: -20px;
180       top: 25px;
181       font-size: 12px;
182     }
183   }
184 }
```

```
183   }
184   }
185   }
186   }
187 </style>
```

3) 构建环境

保存成功后，在项目构建配置里构建环境那一栏，勾选provide Configuration files

The screenshot shows the 'Build Environment' configuration tab. At the top, there are tabs for 'General', '源码管理', '构建触发器', '构建环境' (selected), 'Pre Steps', 'Build', 'Post Steps', '构建设置', and '构建后'. Below the tabs, there is a checkbox for '轮询 SCM' with a question mark. The main section is titled '构建环境' and contains several checkboxes with question marks: 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Provide Configuration files' (highlighted with a red box), 'Send files or execute commands over SSH before the build starts', 'Send files or execute commands over SSH after the build runs', 'Abort the build if it's stuck', 'Add timestamps to the Console Output', 'Execute shell script on remote host using ssh', 'Inspect build log for published Gradle build scans', and 'With Ant'.

替换文件：

File：选择新创建的配置文件

Target：需要替换的 文件目录+文件名+后缀

以下是我需要替换的4个文件路径：

```
1 vue.config.js
2 .env.development
3 .env.production
4 .env.staging
```

File ?

manstro-web-vue.config.js

view selected file

Target ?

vue.config.js

Variable ?

☐ Replace Tokens

5.构建任务测试

点击按钮构建试试，看看能否顺利

所有 +

添加说明

S	W	名称	上次成功	上次失败	上次持续时间	
		manstro-gateway	4 天 17 小时 #15	5 天 15 小时 #9	47 秒	
		manstro-web	没有	无	无	

图标 小 中 大

删除

Atom feed 全部

Atom feed 失败

Atom feed 最新的构建

如果构建成功，可以查看容器运行日志

```
1 docker logs -f nginx
```

可以访问验证下是否启动成功

<http://192.168.16.5:8098/manstro/web>