

¹ vue项目

效果：

<http://36.139.159.39:9090/> 访问网站出现数据

流程：

本地修改代码-->提交到svn仓库-->jenkins拉取svn代码进行编译构建，启动nodejs后端服务，并把dist资源发送到nginx服务器，由容器内的nginx展示

环境：

本地电脑安装nodejs、npm、vue、svn或git、Visual Studio

两台服务器：

120.79.205.221 安装nodejs、npm、jenkins、svn或git

36.139.159.39 安装docker、nginx镜像

步骤：

一、安装本地环境

svn安装

使用svnbucket教程安装 <https://svnbucket.com/>

Please make sure that you choose the right installer for your PC, otherwise the setup will fail.

for 32-bit OS

TortoiseSVN 1.14.5 - 32-bit

for 64-bit OS

TortoiseSVN 1.14.5 - 64-bit

for ARM64 OS

TortoiseSVN 1.14.5 - ARM64

old LTS release is also available:

for 32-bit OS








TortoiseSVN 1.10.5 - 32-bit

for 64-bit OS

TortoiseSVN 1.10.5 - 64-bit

下拉下载汉化包

Language packs

Country	32 Bit	64 Bit	ARM64	Separate manual (PDF)	
 Albanian	Setup	Setup	Setup	Translate to Albanian	
 Arabic	Setup	Setup	Setup	Translate to Arabic	
 Belarussian (Tarask)	Setup	Setup	Setup	Translate to Belarussian (Tarask)	
 Bulgarian	Setup	Setup	Setup	Translate to Bulgarian	
 Catalan	Setup	Setup	Setup	Translate to Catalan	
 Chinese, simplified	Setup	Setup	Setup	TSVN	TMerge
 Chinese, traditional	Setup	Setup	Setup	Translate to trad. Chinese	

svnbucket 创建仓库

本地创建文件夹上传或checkout下载项目

nodejs和npm安装

<https://www.cnblogs.com/stcweb/articles/15585559.html>

Visual Studio安装

window命令行操作

下载vue的时候太慢，可以配置淘宝源，使用cnpm

```
npm config set registry https://registry.npm.taobao.org
```

检查是否下载成功

```
npm config get registry
```

安装vue脚手架程序vue-cli

// 安装

```
npm install -g @vue/cli
```

// 或者

```
cnpm install -g @vue/cli
```

// 或者

```
yarn global add @vue/cli
```

卸载 npm uninstall vue-cli -g

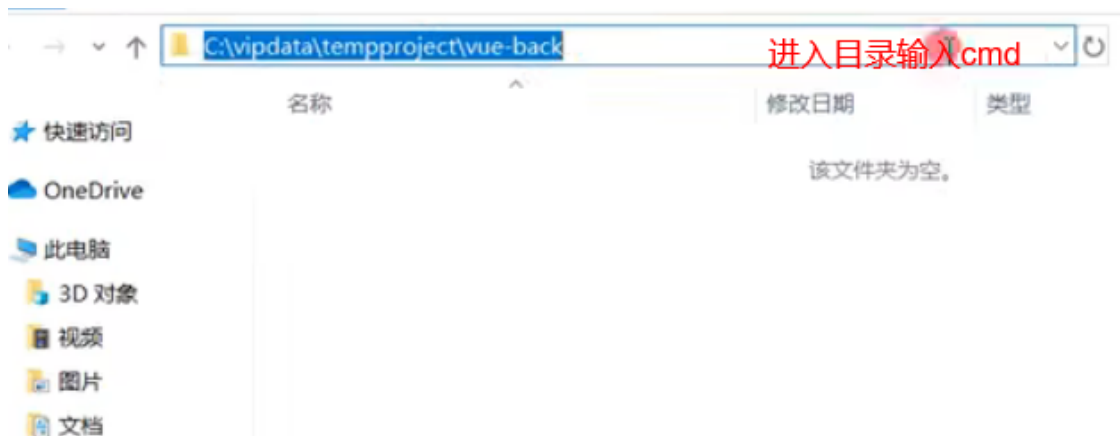
1、新建项目目录

E:\testv

testv下创建目录存放后端代码

E:\testv\back

当前目录下Shift+右键，选powershell窗口 或 路径上搜cmd，打开命令行



自动创建并初始化项目：

命令行输入 `cnpm init -y`

visual studio打开back文件夹，新建server.js文件

```
js 1 let http = require('http');
2 let users = [{id:58,name:'zhufeng1'},{id:31,name:'zhufeng2'},{id:3,name:'zhufeng3'}];
3 let server = http.createServer(function(req,res){
4   console.log(req.method,req.url);
5   if(req.url == '/api/users'){
6     res.setHeader('Access-Control-Allow-Origin','*');
7     res.end(JSON.stringify(users));
8   }else{
9     res.end('Now Found!');
10  }
11 });
12 server.listen(3000,()=>{
13   console.log('服务正在3000端口上启动!');
14 });
```

package.json

```
json 1 {
2   "name": "back",
3   "version": "0.0.0",
4   "private": true,
5   "scripts": {
6     "start": "node ./server.js "
7   }
8 }
```

安装和npm差不多的yarn包管理器

`npm install -g yarn`

回到testv文件夹下，命令行输入，（需要选择vue框架版本，vue2就行）之后会自动创建前端项目文件夹

vue create vue-front

完成后启动服务

yarn serve

vue-front下新增后端接口

yarn add axios

visual studio打开front文件夹，修改APP.vue文件

```
js 目
1 <template>
2   <ul>
3     <li v-for="(user) in users" :key="user.id">
4       {{user.id}}:{{user.name}}
5     </li>
6   </ul>
7 </template>
8 <script>
9 import axios from 'axios';
10 export default {
11   name: 'app',
12   data(){
13     return {
14       users:[]
15     }
16   },
17   mounted(){
18     axios.get('http://120.79.205.221:3000/api/users').then(response=>{
19       this.users = response.data;
20     });
21   }
22 }
23 </script>
24 <style>
25 </style>
```

svn提交更新

二、120.79.205.221 服务器环境

jenkins自由任务的配置

表示从svn仓库拉取代码到jenkins的工作空间目录下，进行编译，然后将编译结果发送到nginx 服务器

Configuration

- General
- 源码管理**
- 构建触发器
- 构建环境
- Build Steps
- 构建后操作

源码管理

- ☐ 无
- ☐ Git ?
- ☒ Subversion ?

Modules

Repository URL ?

svn://svn.svnbucket.com/shishaoxin100/vue_test

Credentials

shishaoxin100/***** (svn secret)

+ 添加

Local module directory ?

.

Repository depth ?

Infinity

☒ Ignore externals ?

☒ Cancel process on externals fail ?

Add module...

Additional Credentials ?

Add additional credentials...

Check-out Strategy

Always check out a fresh copy

Delete everything first, then perform "svn checkout". While this takes time to execute, it ensures that the workspace is in the pristine state.

☒ Quiet check-out ?

源码库浏览器 ?

(自动)

构建编译vue前端代码，会生成dist文件夹，文件夹直接扔到nginx的usr/share/nginx/html下，网站就可以运行了

Build Steps

Execute shell ?

命令

查看 [可用的环境变量列表](#)

```
cd front
rsync -a /root/svndata/module/node_modules /root/.jenkins/workspace/testv/front/
npm run build
pm2 restart ../back/server.js
```

安装pm2管理node进程工具

bash 

- 1 使用node server.js 会阻塞前台，无法进行其他操作，故用pm2
- 2 npm install pm2 -g #全局安装pm2
- 3 pm2 start back/server.js #启动服务器

```
4 pm2 list          #查看运行状态
5 pm2 logs          #查看日志
6 pm2 restart www   #重启应用
7 pm2 stop www      #停止应用
```

构建后操作

Send build artifacts over SSH ?

SSH Publishers

SSH Server

Name ?

yidongyun

高级...

Transfers

Transfer Set

Source files ?

front/dist/**

Remove prefix ?

front/dist

Remote directory ?

root/data/dist

Exec command ?

docker exec -it mynginx chown -R nginx:nginx /usr/share/nginx/html

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

高级...

```
bash
```

- 1 Source files填写错误会导致SSH: Transferred 0 file(s)
- 2 填 front/dist 无法拷贝
- 3 填 front/dist/* 则dist下的子目录不会拷贝, 只会拷贝dist下层的文件
- 4 填 front/dist/** 则dist下的文件和目录均会被拷贝
- 5
- 6 文件传输完成后需要执行把容器内nginx的html目录下的文件改为nginx属主, 这样才不会报403访问拒绝, html下没文件也会报403

Source填写以工作空间目录作为/根开始

↑ 返回面板

📄 状态

</> 修改记录

📁 工作空间

🗑️ 清理工作空间

▶ Build Now

Workspace of testv on Built-In Node

testv /

📁 .svn

📁 back

📁 front

📁 hook

📁 (打包下载全部文件)

启动nodejs的后端server

```
bash
1 进入文件夹
2 .jenkins/workspace/testv/back
3 执行启动命令
4 node server.js
5
6 此实验不需要启动前端服务
7 #front里前端vue项目启动用命令 npm run serve 或 yarn serve
```

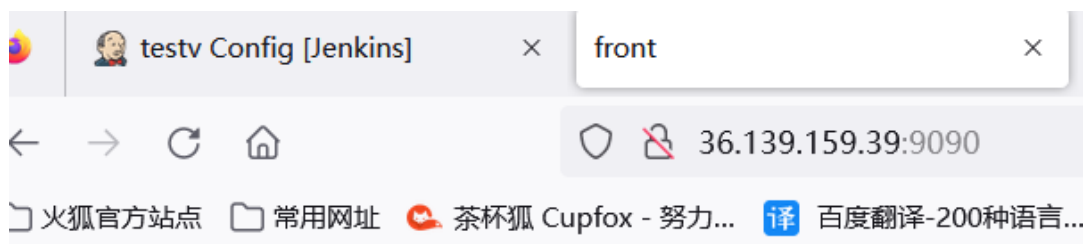
三、36.139.159.39服务器

安装docker¹

```
bash
1 启动nginx的时候把html目录挂载出来
2 docker run -di --name=mynginx -p 9090:80 -v /root/data/dist:/usr/share/nginx/html nginx
```

实验效果访问nginx，前端代码里指定了去访问后端的网站

http://120.79.205.221:3000/api/users获取数据，然后再把数据填充回前端页面



- 58:zhufeng1
- 31:zhufeng2
- 3:zhufeng3

1. 安装 docker

Docker部署²

yum -y install yum-utils

yum-config-manager --add-repo

<https://download.docker.com/linux/centos/docker-ce.repo> #添加下载仓库

yum install -y docker-ce docker-ce-cli containerd.io #安装发行版 docker、
客户端命令行、容器相关组件

systemctl enable docker --now #开机自启 + 现在马上启动

由于 docker hub 是外国网站, pull 较慢, 可以配置阿里容器镜像加速器

进入云服务器控制台-->搜容器镜像服务-->选镜像工具

```
bash
1 mkdir -p /etc/docker
2 #镜像仓库配置, 1、阿里云镜像加速器, 最好用阿里的
3 tee /etc/docker/daemon.json <<- 'EOF'
4 {
5   "registry-mirrors": ["https://82m9ar63.mirror.aliyuncs.com"],
6   #此网址必须在自己的阿里云控制台-容器镜像服务找到
7   "exec-opts": ["native.cgroupdriver=systemd"],
8   "log-driver": "json file",
9   "log-opts": { "max-size": "100m"},
10  "storage-driver": "overlay2"
11 }
12 EOF
13 #或者2、
14 tee /etc/docker/daemon.json <<- 'EOF'
15 {
16   "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
17 }
18 EOF
19 ##
20 Docker 中国官方镜像      https://registry.docker-cn.com
21 DaoCloud 镜像站  http://f1361db2.m.daocloud.io
22 Azure 中国镜像      https://dockerhub.azk8s.cn
23 科大镜像站      https://docker.mirrors.ustc.edu.cn
24 阿里云      https://<your_code>.mirror.aliyuncs.com
25 七牛云      https://reg-mirror.qiniu.com
26 网易云      https://hub-mirror.c.163.com
27 腾讯云      https://mirror.ccs.tencentyun.com
28
29 systemctl daemon-reload
30 systemctl restart docker
```

2. Docker部署

1、yum包更新到最新


```
yum update
```

2、安装需要的软件包，yum-util提供yum-config-manager功能，另外两个是devicemapper驱动依赖的

```
yum install -y yum-utils devmapper-persistent-data 1vm2
```

3、设置yum源为阿里云

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

4、安装docker

```
yum -y install docker-ce
```