

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 10. shell脚本启动go服务

全部开发者教程

8. go中常见的错误

第22周 设计模式和单元测试

1. go最常用的设计模式 - 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动



bobby · 更新于 2022-11-16

上一节 9. 定时构建的语法1. 有哪些规范我... 下一节

1. centos7 下安装nodejs

1. 下载

1. 从官网下载最新的nodejs, <https://nodejs.org/en/download/>



通过这里找到下载的路径 然后使用wget下载, 比如我这里:

<> 代码块

```
1 wget https://nodejs.org/dist/v12.18.3/node-v12.18.3-linux-x64.tar.xz
```

2. 解压和建立软连接

<> 代码块

```
1 tar -xvf node-v12.18.3-linux-x64.tar.xz
```

3. 建立软连接

一定要先找到node可执行文件的完整路径比如我这里



<> 代码块

```
1
2 ln -s /root/node-v12.18.3-linux-x64/bin/node /usr/bin/node
3 ln -s /root/node-v12.18.3-linux-x64/bin/npm /usr/bin/npm
```

注意ln指令用于创建关联 (类似与Windows的快捷方式) 必须给全路径, 否则可能关联错误。

3. 测试

<> 代码块

```
1 node -v
```

4. 安装cnpm

<> 代码块

```
1 npm install -g cnpm --registry=https://registry.npm.taobao.org
```

5. 安装nginx

<> 代码块

6. 安装nginx - 挂载目录

想要正确运行下面的命令实现需要从一个可以运行的nginx容器中拷贝对应的文件进入宿主机
运行下面下面的命令（不能在容器中运行）
然后 我们将nginx的一些配置信息 copy到刚创建的对应目录中去

<> 代码块

```
1 docker cp nginx:/etc/nginx /docker/nginx/config/
2 docker cp nginx:/usr/share/nginx/html /docker/nginx/data/
3 docker cp nginx:/var/log/nginx /docker/nginx/logs/
```

//

<> 代码块

```
1 docker run --name nginx -p 80:80 \
2 -v /docker/nginx/config/nginx:/etc/nginx \
3 -v /docker/nginx/data/html:/usr/share/nginx/html \
4 -v /docker/nginx/logs:/var/log/nginx \
5 -d nginx:latest
```

2. centos7 安装golang

1. 准备安装目录

<> 代码块

```
1 mkdir ~/go && cd ~/go
```

2. 下载

<> 代码块

```
1 wget https://dl.google.com/go/go1.15.10.linux-amd64.tar.gz
```

3. 执行 tar 解压到 /usr/local 目录下（官方推荐），得到 go 文件夹等

<> 代码块

```
1 tar -C /usr/local -zxvf go1.15.10.linux-amd64.tar.gz
```

4. 添加 /usr/local/go/bin 目录到PATH变量中。添加到 /etc/profile 或 \$HOME/.profile 都可以

<> 代码块

```
1 # 习惯用vim, 没有的话可以用命令`sudo apt-get install vim`安装一个
2 vim /etc/profile
3 # 在最后一行添加
4 export GOROOT=/usr/local/go
5 export PATH=$PATH:$GOROOT/bin
6 # 保存退出后source一下 (vim 的使用方法可以自己搜索一下)
7 source /etc/profile
```

5. 开启go module

<> 代码块

```
1 go env -w GO111MODULE=on
2 go env -w GOPROXY=https://goproxy.io
```

3. shell脚本启动go服务

<> 代码块

```
1  srv_name="goods_web_main"
2  chmod +x ./${srv_name}
3  #重启, 如果已经存在则关闭重启
4  if pgrep -x ${srv_name} > /dev/null
5  then
6      echo "${srv_name} is running"
7      echo "shutting down ${srv_name}"
8      if ps -a | grep ${srv_name} | awk '{print $1}' | xargs kill $1
9      then
10         echo "starting ${srv_name}"
11         ./${srv_name} > /dev/null 2>&1 &
12         echo "start ${srv_name} success"
13     fi
14 else
15     echo "starting ${srv_name}"
16     ./${srv_name} > /dev/null 2>&1 &
17     echo "start ${srv_name} success"
18 fi
```

9. 定时构建的语法 ◀ 上一节 下一节 ▶ 1. 有哪些规范我们应该遵循

✎ 我要提出意见反馈