1，Ubuntu上nodejs搭建

a.准备

安装gcc

$ sudo apt-get install g++ curl libssl-dev apache2-utils

安装git

$ sudo apt-get install git-core

b.到官网下载最新nodejs，下载.tar.gz的

英文网址：https://nodejs.org/en/download/

中文网址：http://nodejs.cn/download/

解压：

$ tar -xvf node-v6.10.0-linux-x64.tar.xz

$ mv node-v6.10.0-linux-x64 nodejs

建立好软连接（注意结对路径）

$ ln -s /home/iceld/dev/nodejs/bin/npm /usr/local/bin/

$ ln -s /home/iceld/dev/nodejs/bin/npm /usr/local/bin/

c.测试是否安装成功

$ node -v

$ npm -v

2，webstorm搭建

a.准备

下载jdk：http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html

下载webstorm：http://www.jetbrains.com/webstorm/download/#section=linux-version

b.jdk安装

解压：

$ sudo mkdir /usr/lib/jvm

$ sudo tar -zxvf jdk-8u111-linux-x64.tar.gz -C /usr/lib/jvm

$ cd /usr/lib/jvm/

$ sudo mv jdk-8u111-linux-x64 java

配置环境变量：

$ sudo vi ~/.bashrc

在尾部加入如下代码：

# jdk env config

export JAVA\_HOME=/usr/lib/jvm/java

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

让配置生效：

$ sudo source ~/.bashrc

测试：

$ java -version

$ java

c.webstorm安装

$ sudo tar -zxvf WebStorm-2017.1.4.tar.gz

$ cd WebStorm-2017.1.4

$ su

$ ./webstorm.sh

下一步，下一步，你懂的~

d.启动webstorm设置node

<img class="alignnone size-medium wp-image-189" src="http://www.iceld.top/wp-content/uploads/2017/08/Selection\_005-300x74.png" alt="" width="300" height="74" />

<img src="http://www.iceld.top/wp-content/uploads/2017/08/Selection\_006-300x110.png" alt="" width="300" height="110" class="alignnone size-medium wp-image-190" />

3，express搭建MVC开发环境

$ sudo apt-get install node-express-generator #安装ubuntu支持

$ npm install express-generator -g #全局安装express

$ npm install express --save #当前项目安装express并将其保存到依赖列表中

$ npm install express #临时安装express未保存到依赖列表中

$ express -V #查看版本

$ express -h #查看帮助

a.express应用生成器 http://www.expressjs.com.cn/starter/generator.html

b.创建myward项目，神奇的输出HellWorld~

$ mkdir myward

$ cd myward

$ npm init #创建package.json文件

$ npm install express --save

$ vim app.js #输入如下内容\*\_\*

'use strict';

var express = require('express');

var app = express();

app.get('/', function (req, res) {

res.send('Hello World|^\_^')

});

var server = app.listen(3000, function () {

var host = server.address().address;

var port = server.address().port;

console.log('app is linstenning on http://%s:%s', host, port)

});

c.运行myward

$ node app.js

在浏览器输入:http://localhost:3000，看到大家熟悉的 Hello World|^\_^

4，创建nodejs守护进程

a.pm2安装及命令

安装：npm install -g pm2

pm2帮助：pm2 --help

启动程序：pm2 start <app\_name|id|all>

　　 pm2 start app,js --name=test

启动进程：pm2 start index.js --name express -i 8 #开8个名字为express的进程集群

列举进程：pm2 list

退出程序：pm2 stop <app\_name|id|all>

重起应用：pm2 restart

程序信息：pm2 describe id|all

删除程序：pm2 delete <app\_name|id|all>

推出pm2进程：pm2 kill

监控：pm2 monit

实时集中log处理: pm2 logs

API:pm2 web (端口：9615 )

b.pm2支持配置文件启动：

pm2 ecosystem： 生成配置文件ecosystem.json

pm2 startOrRestart /file/path/ecosystem.json : 通过配置文件启动服务

ecosystem.json

{

/\*\*

\* Application configuration section

\* http://pm2.keymetrics.io/docs/usage/application-declaration/

\* 多个服务，依次放到apps对应的数组里

\*/

apps : [

// First application

{

name : "nova",

max\_memory\_restart: "300M",

script : "/root/nova/app.js",

out\_file : "/logs/nova\_out.log",

error\_file : "/logs/nova\_error.log",

instances : 4,

exec\_mode : "cluster",

env: {

NODE\_ENV: "production"

}

}

]

}

c.集群及80端口映射：

http://www.jianshu.com/p/225b9284cfb8

http://www.jianshu.com/p/4c8330720899

5，将项目运行到Docker上面

a.下载官方node镜像

$ docker pull hub.c.163.com/library/node:latest

b.创建Dockerfile

$ su

$ cd /home/iceld/docker/node-app/

$ vim Dockerfile #写入下面内容

from hub.c.163.com/library/node

MAINTAINER iceld 305822715@qq.com

RUN mkdir /usr/src/app

COPY package.json /usr/src/app

RUN cd /usr/src/app \

&& npm install

COPY . /usr/src/app

EXPOSE 3000 #容器暴露端口

CMD ["node", "/usr/src/app/app.js"] #运行应用

c.构建myward镜像

$ docker build -t myward:latest . #千万别忘记这个 .

$ docker images #查看构建的镜像

d.运行myward容器

$ docker run -d -p 3001:3000 myward

$ docker ps

$ netstat -an|grep 3001

$ docker exec -it 容器id /bin/bash #进入运行的myward容器

curl localhost:3001 或 在浏览器输入：http://localhost:3001,你会看上面神奇的效果

参考：

nodejs深入可以看： http://nodejs.cn/api/

express深入可以看：http://www.expressjs.com.cn/4x/api.html