

# Node.js 14.15.2 CentOS 8.2 64 位

## 【商品介绍】

操作系统版本：CentOS 8.2 64 位

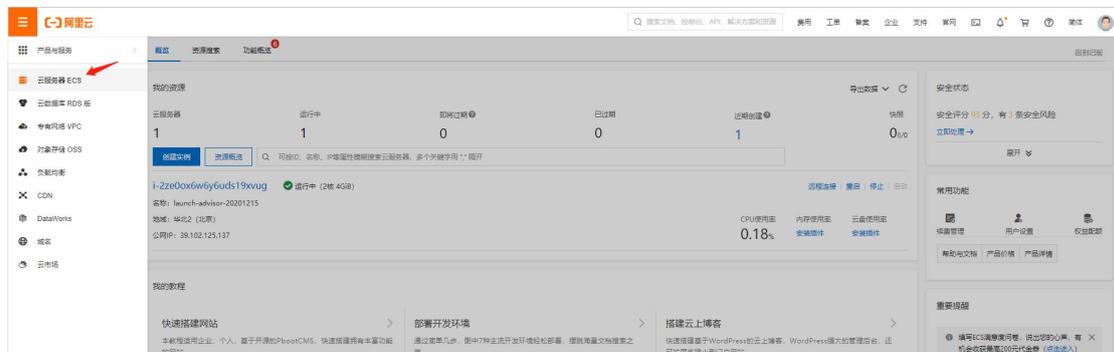
Node.js 版本：14.15.2

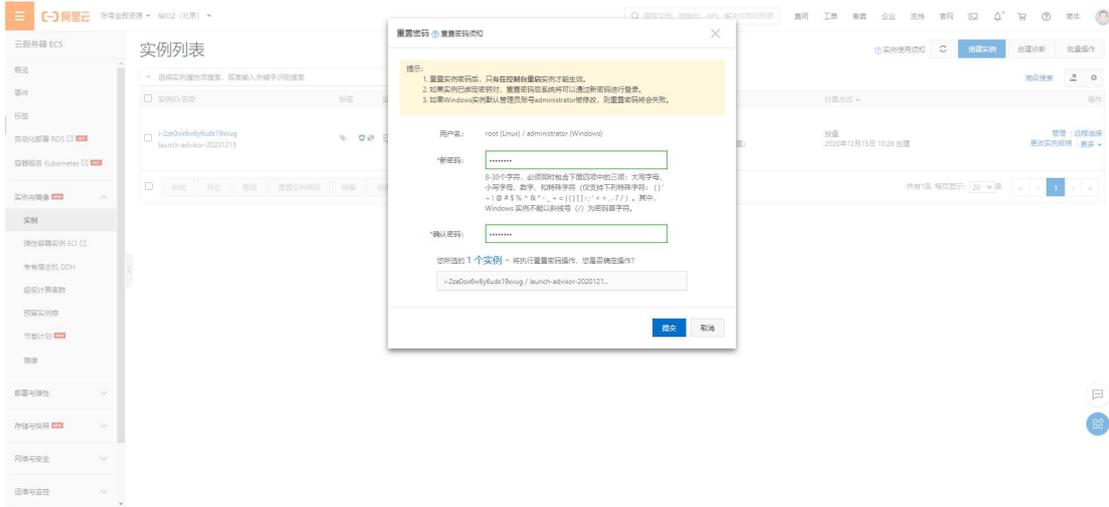
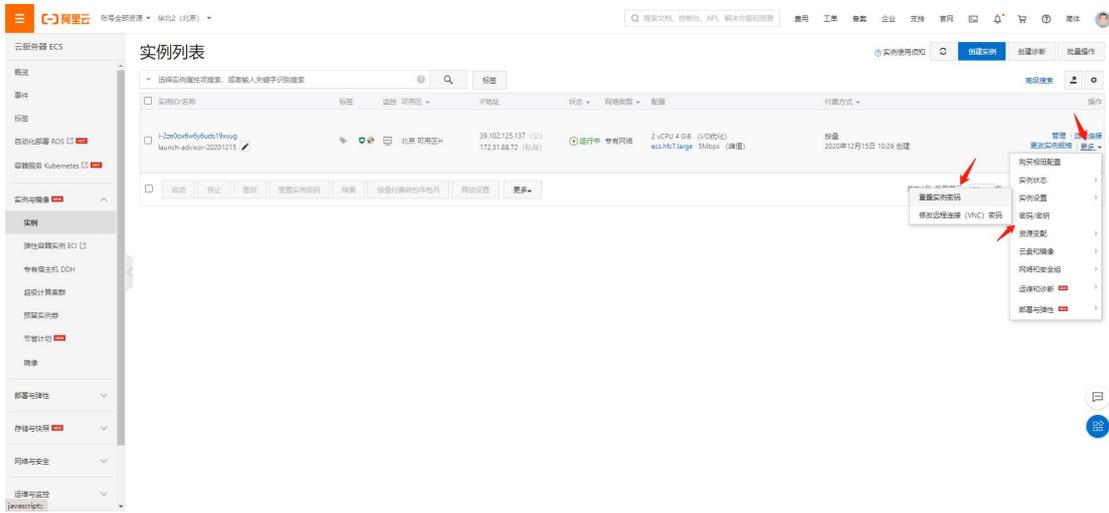
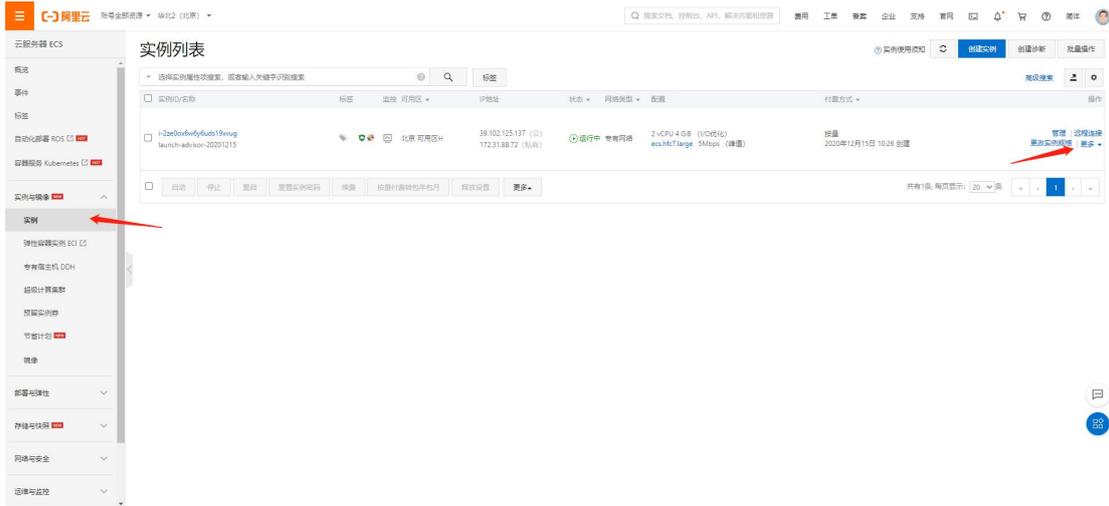
## 【功能亮点】

所有软件均使用官方下载地址，请放心使用；服务和实例均已配置重启自动启动。

## 【产品说明】

1. 服务器账户信息：CentOS 系统默认管理员账户为 root，默认密码为新购云主机设置的密码，如需重置密码请登陆阿里云管理控制台，如图所示，找到左侧菜单栏【云服务器 ECS】



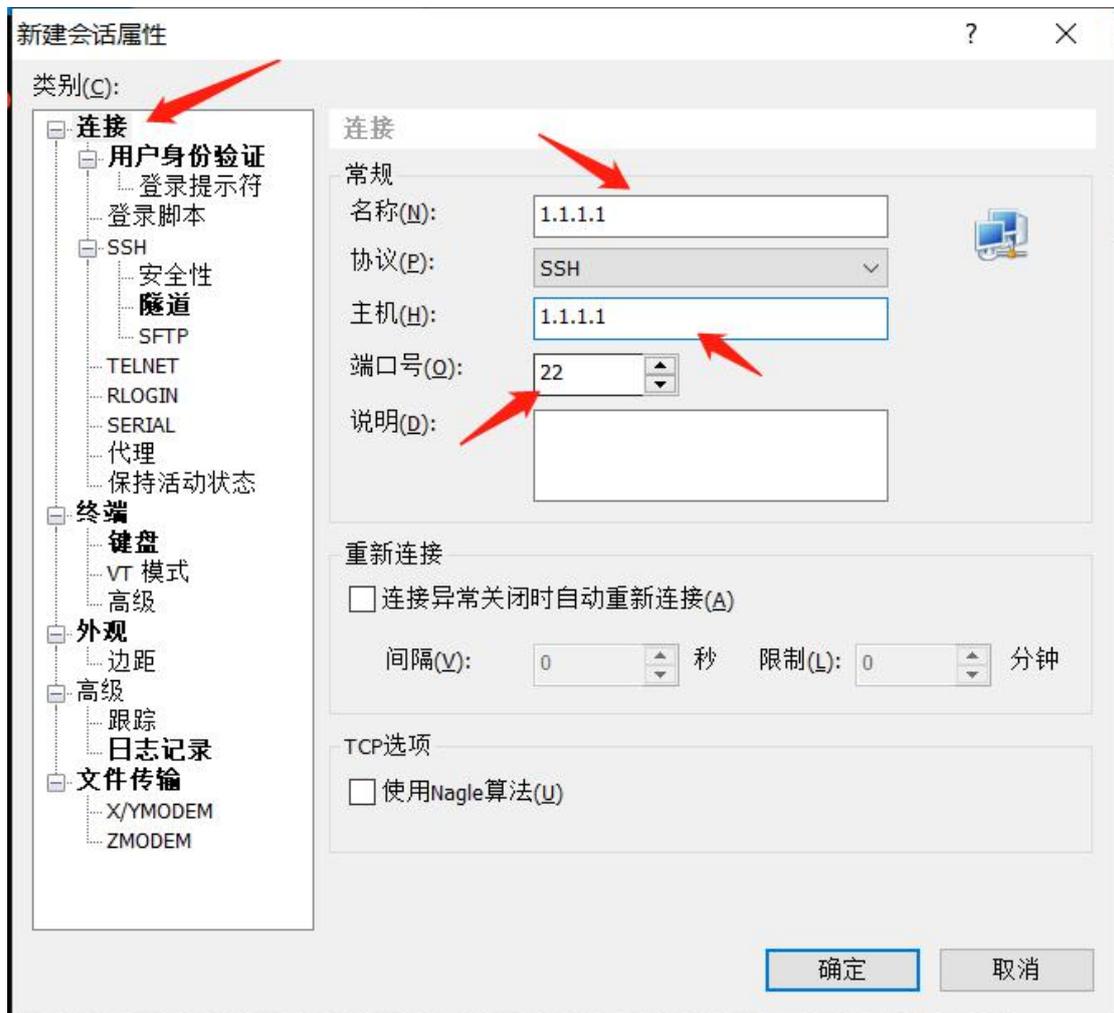


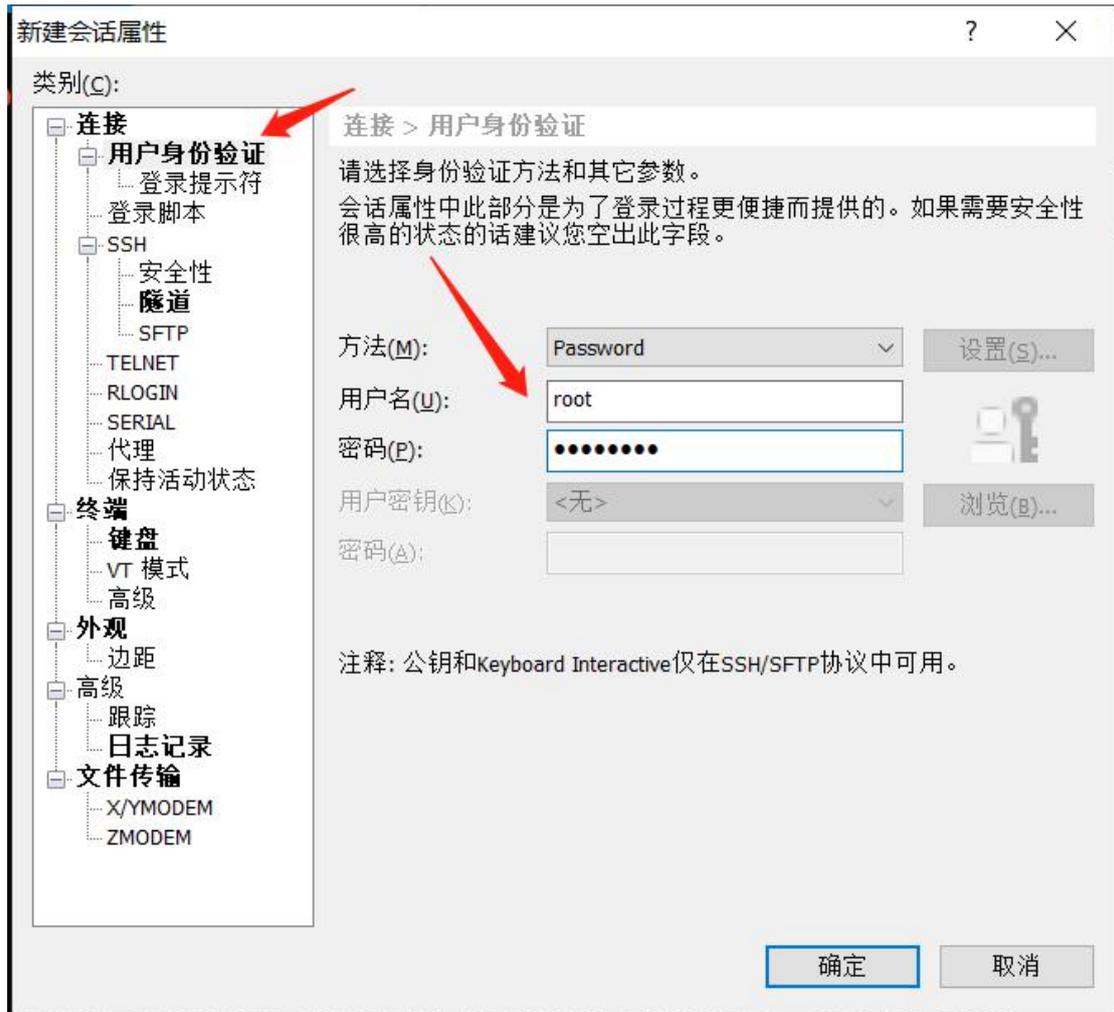
提交之后重启实例即可。

2. 默认防火墙是关闭的

## 【系统登录】

本地个人电脑安装 Xshell 或者 CRT SSH 远程连接工具，





```
Last failed login: Wed Dec 16 16:25:29 CST 2020 from 167.71.147.212 on ssh:notty
There were 4 failed login attempts since the last successful login.
Last login: Wed Dec 16 16:02:56 2020 from 222.175.151.170
[root@nodejs ~]#
```

## 一、npm 安装模块

### Npm 安装模块

npm install <Module Name> 安装模块到项目目录下

npm install <Module Name> -g 安装模块到全局

例如:

```
[root@nodejs ~]# npm install -g pm2
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated fsevents@2.1.3: Please update to v 2.2.x
/usr/local/node/bin/pm2 -> /usr/local/node/lib/node_modules/pm2/bin/pm2
/usr/local/node/bin/pm2-dev -> /usr/local/node/lib/node_modules/pm2/bin/pm2-dev
/usr/local/node/bin/pm2-docker -> /usr/local/node/lib/node_modules/pm2/bin/pm2-docker
/usr/local/node/bin/pm2-runtime -> /usr/local/node/lib/node_modules/pm2/bin/pm2-runtime
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules/pm2/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

+ pm2@4.5.0
added 198 packages from 195 contributors in 111.77s
```

## 二、构建项目

进入项目所在目录，执行 `cnpm install&&npm run build`,生成的包会在 `dist` 文件夹下。

例如：

```
[root@nodejs ~]# cd example
[root@nodejs example]# cnpm install&&npm run build
: [15/28] Installing postcss-value-parser@3.0.0Platform unsupported @vue/cli-plugin-typescript@3.12.1 > fork-ts-checker-webpack-plugin@0.5.2 > chokidar@2.1.8
> fsevents@1.2.7 Package require os(darwin) not compatible with your platform(linux)
Platform unsupported sass@1.30.0 > chokidar@3.4.3 > fsevents@2.1.2 Package require os(darwin) not compatible with your platform(linux)
[fsevents@1.2.7] optional install error: Package require os(darwin) not compatible with your platform(linux)
[fsevents@2.1.2] optional install error: Package require os(darwin) not compatible with your platform(linux)
✓ Installed 28 packages
✓ Linked 973 latest versions
[1/4] scripts.install husky@4.0.9 run "node husky install", root: "/root/example/node_modules/_husky@4.3.6@husky"
husky > Setting up git hooks
fatal: 不是一个 git 仓库 (或者任何父目录) : .git
husky > Failed to install
[1/4] scripts.install husky@4.0.9 finished in 89ms
[1/4] scripts.postinstall husky@4.0.9 run "opencollective-postinstall || exit 0", root: "/root/example/node_modules/_husky@4.3.6@husky"
Thank you for using husky!
If you rely on this package, please consider supporting our open collective:
> https://opencollective.com/husky/donate

[1/4] scripts.postinstall husky@4.0.9 finished in 52ms
[2/4] scripts.install @vue/cli-plugin-typescript@3.12.1 > yorkie@2.0.0 run "node bin/install.js", root: "/root/example/node_modules/_yorkie@2.0.0@yorkie"
setting up git hooks
can't find .git directory, skipping Git hooks installation
[2/4] scripts.install @vue/cli-plugin-typescript@3.12.1 > yorkie@2.0.0 finished in 63ms
[3/4] scripts.postinstall @vue/cli-service@3.12.1 > webpack-bundle-analyzer@3.9.0 > ejs@2.6.1 run "node ./postinstall.js", root: "/root/example/node_modules/_ejs@2.7.4@ejs"
Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)

[3/4] scripts.postinstall @vue/cli-service@3.12.1 > webpack-bundle-analyzer@3.9.0 > ejs@2.6.1 finished in 50ms
[4/4] scripts.postinstall @vue/cli-plugin-babel@3.12.1 > @vue/babel-preset-app@3.12.1 > core-js@2.6.5 run "node -e `try(require('./postinstall'))catch(e){}`", root: "/root/example/node_modules/_core-js@2.6.12@core-js"
[4/4] scripts.postinstall @vue/cli-plugin-babel@3.12.1 > @vue/babel-preset-app@3.12.1 > core-js@2.6.5 finished in 48ms
✓ Run 4 scripts
peerDependencies WARNING sass-loader@10.0.4 requires a peer of node-sass@4.0.0 || ^5.0.0 but none was installed
peerDependencies WARNING sass-loader@10.0.4 requires a peer of fibers@>= 3.1.0 but none was installed
deprecate popper.js@1.14.7 You can find the new Popper v2 at @popperjs/core, this package is dedicated to the legacy v1
deprecate @vue/cli-plugin-babel@3.12.1 > @vue/cli-shared-utils@3.12.1 > request-promise-native@1.0.7 request-promise-native has been deprecated because it extends the now deprecated request package, see https://github.com/request/request/issues/3142
deprecate @vue/cli-plugin-babel@3.12.1 > @vue/cli-shared-utils@3.12.1 > request@2.87.0 request has been deprecated, see https://github.com/request/request/issues/3142
deprecate @vue/cli-plugin-babel@3.12.1 > @vue/cli-shared-utils@3.12.1 > @hapi/joi@15.0.1 joi is leaving the @hapi organization and moving back to 'joi' (https://github.com/sidewav/joi/issues/2411)

warning

webpack performance recommendations:
You can limit the size of your bundles by using import() or require.ensure to lazy load some parts of your application.
For more info visit https://webpack.js.org/guides/code-splitting/

File                                Size                                Gzipped
dist/js/chunk-vendors.28c532b3.js    1161.20 KiB                        393.42 KiB
dist/js/app.10f1c69a.js              670.98 KiB                          203.61 KiB
dist/css/app.05de3a48.css            64.98 KiB                           12.07 KiB

Images and other types of assets omitted.

DONE Build complete. The dist directory is ready to be deployed.
INFO Check out deployment instructions at https://cli.vuejs.org/guide/deployment.html

[root@nodejs example]# ls
babel.config.js  LICENSE          package.json     postcss.config.js  README.md  src          tslint.json
dist             node_modules    package-lock.json  public             run.sh     tsconfig.json  vue.config.js
[root@nodejs example]#
```

## 三、启动项目

1) `package.json` 文件 `scripts` 的 `serve` 字段中配置了服务启动的命令。

例如：

```
"scripts": {
  "serve": "vue-cli-service serve",
  "build": "vue-cli-service build",
  "lint": "vue-cli-service lint"
},
```

`npm run serve`

```
[root@nodejs example]# npm run serve

> rocketbot@1.1.0 serve /root/example
> vue-cli-service serve

INFO Starting development server...
Starting type checking and linting service...
Using 1 worker with 2048MB memory limit
98% after emitting CopyPlugin

DONE Compiled successfully in 12724ms

No type errors found
No lint errors found
Version: typescript 3.9.7, tslint 5.20.1
Time: 12466ms

App running at:
- Local:   http://localhost:8080/
- Network: http://172.21.0.9:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

2)将 dist 生成的静态文件，放置应用中启动

3) 通过 pm2 管理启动

#### 四、npm 命令

```
npm init 初始化 package.json 文件
npm uninstall <Module Name> 卸载模块
npm update <Module Name> 更新模块
npm outdated <Module Name> 检查模块是否过时
npm start 启动模块（该命令写在 package.json 文件 scripts 的 start 字段中，可以自定义命令来配置一个服务器环境和安装一系列的必要程序，如果没有配置，默认直接启动 node server.js）
npm stop 停止模块
npm restart 重新启动模块
npm --version 查看 npm 版本
npm ls 查看当前目录下安装的模块
pm2 start app.js 启动 app 项目
pm2 list 列出由 pm2 管理的所有进程信息
pm2 logs 显示所有进程日志
pm2 stop all 停止所有进程
```

```
pm2 restart all 重启所有进程
```

## 五、package.json 语法

### **name**

在 package.json 中最重要的就是 name 和 version 字段。他们都是必须的，如果没有就无法 install。name 和 version 一起组成的标识在假设中是唯一的。改变包应该同时改变 version。

name 是这个东西的名字。注意：

不要把 node 或者 js 放在名字中。

### **version**

version 必须能被 node-semver 解析, 它被包在 npm 的依赖中。(要自己用可以执行 npm install semver)

### **description**

放简介，字符串

### **keywords**

关键字，数组、字符串

### **license**

许可证。

### **repository**

指定代码存放的地方。

### **Scripts**

“scripts”是一个由脚本命令组成的 hash 对象，他们在包不同的生命周期中被执行。key 是生命周期事件，value 是要运行的命令。

### **config**

"config" hash 可以用来配置用于包脚本中的跨版本参数。

#### **dependencies**

依赖是给一组包名指定版本范围的一个 hash。

#### **devDependencies**

用于测试开发的 dependencies

#### **【声明】**

- 1、镜像中如有收费软件，请根据软件官方说明购买使用版权，因版权问题产生的纠纷本公司概不负责。
- 2、镜像操作系统为公司定制，并经过反复测试验证，请参照商品详情中信息使用，免费镜像为客户体验使用，收费镜像为镜像制作费用，除镜像本身默认环境问题，均不含任何人工技术支持。
- 3、部分付费镜像有安全加固，但不保证服务器绝对安全，互联网中不存在绝对安全的服务器，请做好代码安全，并培养良好的使用习惯。

#### **【售后问题】**

- 1、如有软件不能正常使用的情况请联系在线技术支持，技术支持请优先选择旺旺客服；
- 2、如需在线技术支持，配置、调试、故障排查、安全加固、安全测试、高并发测试、负载均衡、集群等等服务请参照本公司服务类产品说明联系技术咨询