**crawlab直接部署文档**

官方部署文档：<https://tikazyq.github.io/crawlab-docs/Installation/Direct.html>

要求（直接部署）

Go 1.12+

Node 8.12+

Redis

MongoDB 3.6+

**Master节点部署**

部署服务器：10.8.26.23

源码位置：/data/server/crawlab-master

1. 安装前后端需要的库
2. 安装前端所需的库

npm install -g yarn

cd frontend

/usr/bin/yarn install

1. 安装后端所需的库

cd ../backend

go install ./...

1. 前后端配置文件修改
2. 修改前端配置文件：./frontend/.env.production

NODE\_ENV='production'

VUE\_APP\_BASE\_URL='http://10.8.26.23:8000'

1. 修改后端配置文件：./backend/conf/config.yml

api:

address: "10.8.26.23:8000"

mongo:

host: 10.8.26.26

port: 27017

db: crawl

username: ""

password: ""

authSource: "admin"

redis:

address: 10.8.26.26

password: ""

database: 1

port: 6379

log:

level: info

path: "/data/code/crawlab-master/logs/crawlab"

isDeletePeriodically: "Y"

deleteFrequency: "@hourly"

server:

host: 0.0.0.0

port: 8000

master: "Y"

secret: "crawlab"

register:

# mac地址 或者 ip地址，如果是ip，则需要手动指定IP

type: "mac"

ip: ""

spider:

path: "/data/code/crawlab-master/spiders"

task:

workers: 4

other:

tmppath: "/data/code/crawlab-master/tmp"

1. 构建前后端
2. 构建前端

cd ../frontend

npm run build:prod

构建完成后，会在./frontend目录下创建一个dist文件夹，里面是打包好后的静态文件。

安装nginx，yum install nginx

添加/etc/nginx/conf.d/crawlab.conf文件，输入以下内容。

server {

listen 8080;

server\_name dev.crawlab.com;

root /data/server/crawlab-master/frontend/dist;

index index.html;

}

运行命令：

cd /usr/sbin

./nginx -s reload

1. 构建后端

cd ../backend

go build

生成执行文件，默认路径为：/data/server/go/pkg/bin/crawlab

1. 启动服务

/data/server/go/pkg/bin/下新建conf目录，将上面更新的config.yml文件放进来

运行启动命令：nohup crawlab &

启动成功，打开浏览器：[http://10.8.26.23:8080](http://10.8.26.23:8080/)

启动成功后日志如下：

[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.

- using env: export GIN\_MODE=release

- using code: gin.SetMode(gin.ReleaseMode)

2019/10/14 16:48:28 info 初始化配置成功

2019/10/14 16:48:28 info 初始化日志设置成功

2019/10/14 16:48:28 info 初始化定期清理日志配置成功

2019/10/14 16:48:28 info 初始化Mongodb数据库成功

2019/10/14 16:48:28 info 初始化Redis数据库成功

2019/10/14 16:48:28 info 初始化定时任务成功

2019/10/14 16:48:28 info 初始化任务执行器成功

2019/10/14 16:48:28 info register type is :\*register.MacRegister

{subscribe nodes:master 1}

2019/10/14 16:48:28 info 初始化节点配置成功

2019/10/14 16:48:28 info 初始化爬虫服务成功

{subscribe nodes:public 1}

2019/10/14 16:48:28 info 初始化用户服务成功

[GIN-debug] POST /login --> crawlab/routes.Login (4 handlers)

[GIN-debug] PUT /users --> crawlab/routes.PutUser (4 handlers)

[GIN-debug] GET /nodes --> crawlab/routes.GetNodeList (5 handlers)

[GIN-debug] GET /nodes/:id --> crawlab/routes.GetNode (5 handlers)

[GIN-debug] POST /nodes/:id --> crawlab/routes.PostNode (5 handlers)

[GIN-debug] GET /nodes/:id/tasks --> crawlab/routes.GetNodeTaskList (5 handlers)

[GIN-debug] GET /nodes/:id/system --> crawlab/routes.GetSystemInfo (5 handlers)

[GIN-debug] DELETE /nodes/:id --> crawlab/routes.DeleteNode (5 handlers)

[GIN-debug] GET /spiders --> crawlab/routes.GetSpiderList (5 handlers)

[GIN-debug] GET /spiders/:id --> crawlab/routes.GetSpider (5 handlers)

[GIN-debug] POST /spiders --> crawlab/routes.PutSpider (5 handlers)

[GIN-debug] POST /spiders/:id --> crawlab/routes.PostSpider (5 handlers)

[GIN-debug] POST /spiders/:id/publish --> crawlab/routes.PublishSpider (5 handlers)

[GIN-debug] DELETE /spiders/:id --> crawlab/routes.DeleteSpider (5 handlers)

[GIN-debug] GET /spiders/:id/tasks --> crawlab/routes.GetSpiderTasks (5 handlers)

[GIN-debug] GET /spiders/:id/file --> crawlab/routes.GetSpiderFile (5 handlers)

[GIN-debug] POST /spiders/:id/file --> crawlab/routes.PostSpiderFile (5 handlers)

[GIN-debug] GET /spiders/:id/dir --> crawlab/routes.GetSpiderDir (5 handlers)

[GIN-debug] GET /spiders/:id/stats --> crawlab/routes.GetSpiderStats (5 handlers)

[GIN-debug] GET /spider/types --> crawlab/routes.GetSpiderTypes (5 handlers)

[GIN-debug] GET /tasks --> crawlab/routes.GetTaskList (5 handlers)

[GIN-debug] GET /tasks/:id --> crawlab/routes.GetTask (5 handlers)

[GIN-debug] PUT /tasks --> crawlab/routes.PutTask (5 handlers)

[GIN-debug] DELETE /tasks/:id --> crawlab/routes.DeleteTask (5 handlers)

[GIN-debug] POST /tasks/:id/cancel --> crawlab/routes.CancelTask (5 handlers)

[GIN-debug] GET /tasks/:id/log --> crawlab/routes.GetTaskLog (5 handlers)

[GIN-debug] GET /tasks/:id/results --> crawlab/routes.GetTaskResults (5 handlers)

[GIN-debug] GET /tasks/:id/results/download --> crawlab/routes.DownloadTaskResultsCsv (5 handlers)

[GIN-debug] GET /schedules --> crawlab/routes.GetScheduleList (5 handlers)

[GIN-debug] GET /schedules/:id --> crawlab/routes.GetSchedule (5 handlers)

[GIN-debug] PUT /schedules --> crawlab/routes.PutSchedule (5 handlers)

[GIN-debug] POST /schedules/:id --> crawlab/routes.PostSchedule (5 handlers)

[GIN-debug] DELETE /schedules/:id --> crawlab/routes.DeleteSchedule (5 handlers)

[GIN-debug] GET /stats/home --> crawlab/routes.GetHomeStats (5 handlers)

[GIN-debug] GET /users --> crawlab/routes.GetUserList (5 handlers)

[GIN-debug] GET /users/:id --> crawlab/routes.GetUser (5 handlers)

[GIN-debug] POST /users/:id --> crawlab/routes.PostUser (5 handlers)

[GIN-debug] DELETE /users/:id --> crawlab/routes.DeleteUser (5 handlers)

[GIN-debug] GET /me --> crawlab/routes.GetMe (5 handlers)

[GIN-debug] GET /ping --> crawlab/routes.Ping (4 handlers)

[GIN-debug] Listening and serving HTTP on 0.0.0.0:8000

**Worker节点部署**

部署服务器：10.8.26.24

1. 将master节点服务器上/data/server/go/pkg/bin目录下的conf和crawlab拷贝到2624上相同目录下，配置conf中的config.yml文件：

api:

address: "10.8.26.23:8000"

mongo:

host: 10.8.26.26

port: 27017

db: crawl

username: ""

password: ""

authSource: "admin"

redis:

address: 10.8.26.26

password: ""

database: 1

port: 6379

log:

level: info

path: "/data/code/crawlab-master/logs/crawlab"

isDeletePeriodically: "Y"

deleteFrequency: "@hourly"

server:

host: 0.0.0.0

port: 18000

master: "N"

secret: "crawlab"

register:

# mac地址 或者 ip地址，如果是ip，则需要手动指定IP

type: "mac"

ip: ""

spider:

path: "/data/code/crawlab-master/spiders"

task:

workers: 4

other:

tmppath: "/data/code/crawlab-master/tmp"

改动之处已用红色标出，当然，配置中的log、spider、tmp目录要建好

1. 回到/data/server/go/pkg/bin目录，运行nohup crawlab &，部署完成，日志内容：

[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.

- using env: export GIN\_MODE=release

- using code: gin.SetMode(gin.ReleaseMode)

2019/10/15 10:07:27 info 初始化配置成功

2019/10/15 10:07:27 info 初始化日志设置成功

2019/10/15 10:07:27 info 初始化定期清理日志配置成功

2019/10/15 10:07:27 info 初始化Mongodb数据库成功

2019/10/15 10:07:27 info 初始化Redis数据库成功

2019/10/15 10:07:27 info 初始化任务执行器成功

2019/10/15 10:07:27 info register type is :\*register.MacRegister

{subscribe nodes:5da529e2816e24395b97e4c5 1}

2019/10/15 10:07:32 info 初始化节点配置成功

2019/10/15 10:07:32 info 初始化爬虫服务成功

{subscribe nodes:public 1}

2019/10/15 10:07:32 info 初始化用户服务成功

[GIN-debug] GET /ping --> crawlab/routes.Ping (3 handlers)

[GIN-debug] Listening and serving HTTP on 0.0.0.0:18000