

蚂蚁链

区块链服务平台 部署指南

产品版本：V3.1.1

文档版本：20250314



蚂蚁集团
ANT GROUP

法律声明

蚂蚁集团版权所有©2022，并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

商标声明

 蚂蚁集团 ANT GROUP 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置 > 网络 > 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.产品简介	05
2.系统规划	06
3.部署流程	08
3.1. 数据库和缓存	08
3.2. 应用服务	08
3.3. 负载均衡代理	20
3.4. 拷贝mychain部署安装包	24
3.5. 配置中途岛集群节点信息	24
4.部署验证	25
5.公司图标及名字定制配置	26
6.REST服务部署	29

1. 产品简介

区块链是一个典型的分布式协同系统，多方共同维护一个不断增长的分布式数据记录，这些数据通过密码学技术保护内容和时序，使得任何一方难以篡改、抵赖、造假。区块链根据不同需求和场景可分为公有链、联盟链和私有链。

蚂蚁金服金融科技区块链以联盟链为目标，突破商业与金融应用场景，率先实现有自主权的工业级与金融级区块链系统，具有高可靠性、高可运维性、高安全性和适配全球部署等优势。

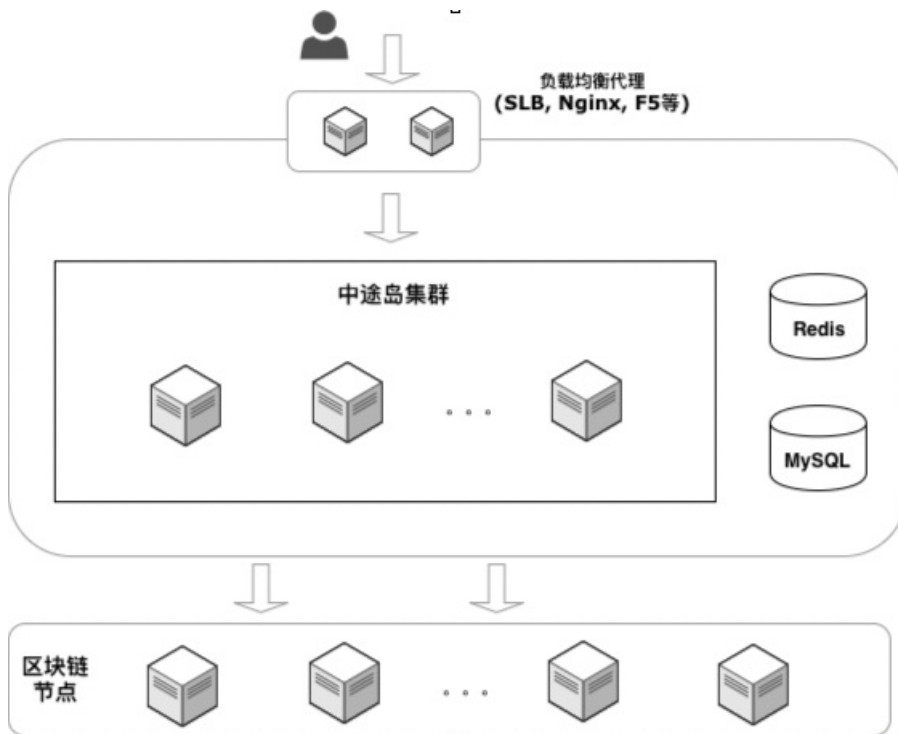
蚂蚁链私有化服务标准版提供简单易用，一键部署，快速验证，灵活可定制的区块链服务，加速区块链业务应用开发、测试、上线，助力各行业区块链商业应用场景落地。提供高性能，稳定可靠，隐私安全，多种类型数据的区块链存证能力。

蚂蚁链私有化服务标准版基于蚂蚁链提供基础技术能力，并输出定制化的区块链整体解决方案，应用于诸如数据存证与溯源、多方参与的业务协同、资产登记流转等场景。

2. 系统规划

本章节首先给出一个整体的部署架构，并根据部署架构列出部署产品所需的软硬件资源等信息。

部署结构



中途岛集群版的标准部署结构，如上图所示，包含几部分内容：

- 中途岛集群应用服务，提供区块链BaaS标准的管理功能，用户通过浏览器登录访问。为了高可用，采用集群化独立部署，建议至少两台以上。
- 数据库RDS，如MySQL，为中途岛集群应用服务提供数据持久化的存储和管理。
- 缓存Redis，为中途岛集群应用服务提供数据缓存，提升访问性能，如区块链数据统计指标，分布式登录Session等。
- 负载均衡（Nginx，SLB，F5等），提供中途岛集群应用服务的负载均衡代理，默认采用SLB或者Nginx，采用权重策略。

资源环境

本小节介绍中途岛集群版在部署实施前的资源和环境的规划建议，包括资源规格，软件环境依赖，网络资源，环境权限等。

部署物	资源规格	软件环境	网络资源	环境权限	备注
中途岛集群应用服务	集群化独立部署，建议至少2台及以上(规格4C8G 存储100G)	java>=1.8，可通过命令 <code>Java -version</code> 查看是否满足条件	需对外开放Web端口（如80），提供浏览器登录访问（或者负载均衡代理转发访问）	机器账户采用admin 以上权限	搭配 Redis（主从或集群均可，以客户环境为准）

数据库-MySQL	2台 4C8G 200G	MySQL5.7+	开放访问权限，提供中途岛应用服务连接访问	机器账户采用 admin 以上权限	建议优先采用标准的RDS云产品。 提供中途岛应用服务连接访问
缓存-Redis	2台 4C8G 200G	Redis5.0+	开放访问权限，提供中途岛应用服务连接访问	机器账户采用 admin 以上权限	建议优先采用标准的云产品。 支持中途岛应用服务的缓存使用，其中包含浏览器数据指标，缓存分布式登录Session等
负载均衡 (Nginx, F5, SLB等)	4C8G 200G 高可用推荐2台+	Nginx 1.20.1+	对外开放web端口（如80），为中途岛应用服务提供负载均衡代理转发	机器账户采用 admin 以上权限	建议优先采用标准的云产品，例如SLB。 Nginx upstream配置同时支持两个代理通路，采用权重策略： 1.Web浏览器访问中途岛应用服务集群

3. 部署流程

3.1. 数据库和缓存

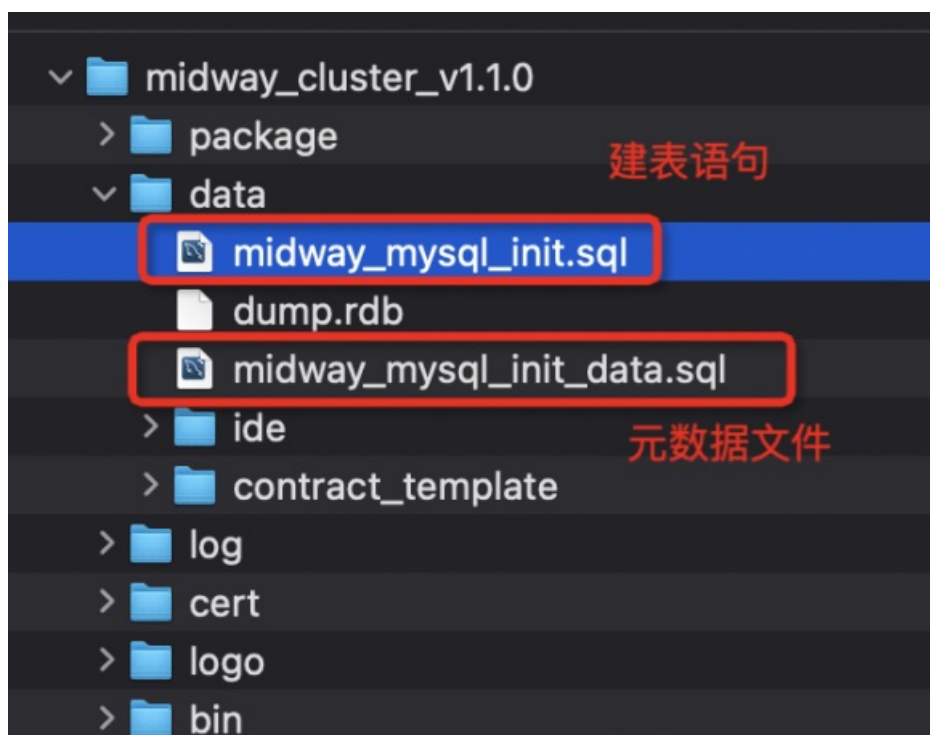
部署步骤

根据项目情况，优先采用客户提供的标准云产品，满足资源清单要求所需即可（参见[资源环境](#)），这里不单独提供数据库和缓存的部署步骤。

MySQL数据库初始化

在数据库准备完成后，需要创建项目所需的表和元数据，通过执行初始化SQL脚本完成，见版本包 `/data` 目录。

版本包结构如下图所示：



- 建表语句：midway_mysql_init.sql
- 初始化元数据：midway_mysql_init_data.sql

将以上两个文件拷贝至MySQL执行即可。

3.2. 应用服务

部署过程

1. 登录承载控制台的服务器。
2. 准备中途岛安装包并拷贝至 `/home/admin` 目录下。
3. 解压中途岛安装包。

```
cd /home/admin
tar -zxvf midway.tar.gz 或 unzip midway.zip
```

4. 进入中途岛 bin 目录。

```
cd /home/admin/midway/bin
```

部署文件树展示如下：

```
midway_v2.7.0
├── bin                                ## 运行目录
│   ├── applicaton-prod.yaml          -- 配置文件
│   ├── midway.jar                    -- 执行文件
│   ├── start.sh                      -- 启动脚本
│   ├── bass-midway-collector         -- 中途岛链日志收集器
│   ├── genCA.sh                     -- ca.key，ca.crt生成脚本
│   ├── ilogtail-1.7.0.linux-amd64.tar.gz -- ilogtail的安装包
│   ├── mychain-demo.tar.gz          -- mychain-java-sdk 示例程序
│   └── monitor                       ## 监控文件目录
│       ├── metric_name.json          -- 指标名称列表
│       ├── group_metric_meta.json    -- 指标分组数据
│       ├── condition_template.json   -- 告警模板
│       ├── log_name.json             -- 日志名称列表
│       └── default_chain_metric.json -- 默认显示指标列表
├── cert                             ## 证书目录，初始为空
│   ├── ca.cert                      -- CA公钥证书
│   ├── tls.key                      -- TLS通信私钥
│   ├── tls.cert                     -- TLS通信公钥
│   ├── admin.key                    -- 管理员私钥文件
│   ├── admin1.key                   -- 管理员私钥文件
│   ├── admin2.key                   -- 管理员私钥文件
│   ├── admin3.key                   -- 管理员私钥文件
│   └── admin4.key                   -- 管理员私钥文件
├── data                             ## 数据库目录
│   └── contract_template             -- 合约模板
├── log                              ## 日志目录，初始为空
│   ├── midway.log                   -- 中途岛日志文件
│   └── gc.log                       -- Java虚拟机日志文件
├── logo                             ## 公司图标存放目录
│   ├── index.png                   -- 控制台页面顶部导航栏图标文件
│   └── login.png                    -- 登录页上的图标文件
└── package                          ## 中途岛临时文件存放包
```

5. 根据当前环境情况，修改配置文件 /home/admin/midway/bin/application-prod.yaml 中的参数。

```
vi application-prod.yaml
```

说明

application-prod.yaml 中各配置项说明及示例，请参见本文下方的[附录：配置文件说明](#)。

6. 按客户实际环境，在 application-prod.yaml 中替换MySQL 和Redis的配置项内容，如下方 {} 所示：

```
spring:
  datasource:
    #数据库连接地址
```

```
url: jdbc:mysql://{mysql_server_host}:{port}/{数据库名}?
useUnicode=true&characterEncoding=UTF-8&serverTimezone=UTC&useSSL=false
username: {数据库RDS账户}
password: {数据库RDS密码}
driver-class-name: com.mysql.cj.jdbc.Driver
http:
  multipart:
    maxFileSize: 40000Mb
    maxRequestSize: 40000Mb
  messages:
    basename: i18n/messages
    encoding: UTF-8
    cache-seconds: 3
  session:
    store-type: redis
    timeout: 3600s
    redis:
      flush-mode: on_save
      namespace: midway:session
  redis:
    database: 15
    host: {Redis访问地址或IP}
    jedis:
      pool:
        max-active: 2000
        max-idle: 200
        max-wait: 1000
        min-idle: 200
    password: {Redis登录密码}
    port: {Redis访问端口}
    timeout: 3000

#避免浏览器下载大文件超时
server:
  connection-timeout: 600000
  disableUploadTimeout: false

# watchman 地址
watchman:
  request:
    url: {watchman请求地址:8080}
  receive:
    url: {watchman接收地址:8080}
  deploy-log:
    #服务日志存放路径
    url: /data1/mychain_deploy/log

# 多方协同存储配置
costorage:
  # 绑定协同存储网络的链ID
  chain_id: M23081014****
  # abi文件所在路径
  abi_path: /Users/keyu/IdeaProjects/baas-midway-cluster/data/costorage.abi
  # 已部署的合约地址
  contract_address: 92207f2f15f52d3dcbf1a5a2d22000755160b75bbfb3045d3b6a2222a****
```

```
contract_address: 0239715119655d3dcb11e1e2d22092906755169b71bb1ba043d2b1e2355e7888
# 协作引擎接入点IP和端口
coengine_endpoint: http://47.100.XX.XX:8585
# 存储节点信息IP和ID
ipfs_node_json: "
[{'ip': '10.0.XX.XX', 'id': '12D3KooWJpmsn33iYnPoKB53emDPDHqRwTVpwpJfQH3vyXP*****'}, {'ip': '10.0.XX.XX', 'id': '12D3KooWDLc2cf6WsJhPBHhFWqqp9Wyu9bgYnVP3U4Rte3Cy*****'}]"

package:
  mychain:
    - "MYCHAIN-2.24.8-Linux-x86_64-bin.tar.gz"
  btn:
    - "btn_pkg_v2.3.0.tar"
  etcd:
    - "etcd_nossl_pkg.tar"
```

配置项	说明描述	备注
spring.datasource.username	数据库RDS账户	username: {数据库RDS账户}
spring.datasource.password	数据库RDS密码	password: {数据库RDS密码}
spring.redis.host	Redis访问地址或IP	host: {Redis访问地址或IP}
spring.redis.password	Redis登录密码	password: {Redis登录密码}
spring.redis.port	Redis访问端口	port: {Redis访问端口}

7. 配置文件修改完成后保存，启动中途岛服务。

```
sh start.sh start
```

部署验证

查询运行日志 midway.log，是否包含启动成功关键字，并且无报错。

```
cd /home/admin/midway/log
grep 'Midway start successfully' midway.log --color
```

附录：配置文件说明

application-prod.yaml 中各配置项说明及示例如下：

参数	说明	示例
# mychain配置		

mychain.run_mode	系统运行模式	FULL # DEBUG HANDY FULL
mychain.deploy_path	部署目录	/home/admin/midway/
mychain.chain_version	区块链版本	0.10.2.12.5
mychain.sqlite_path	sqlite数据库文件路径	midway.db
mychain.enable_query_admin_info		true
mychain.enable_maintain		true
# byd配置		
byd.byd_deploy_path		/data1/mychain_byd_deploy
# rest配置		
rest.openRest		false
rest.bizid		12eba21a66ed4d2795b833ce7ebd750c
rest.cipher-suit		ec
rest.rest-url		http://mychain08.inc.alipay.net:7939/
rest.access-id		baas_admin
rest.access-secret		access.key
rest.tenantid		rest_test
rest.default-account		1253000043120470X4
rest.default-account-key		1253000043120470X4.key

rest.default-account-pwd		t1e2s3t4
rest.read-file-from-ext		false
rest.account		transferTestAccount
rest.kmsId		rest_test_transferTestAccount
# http端口设置		
http.port		8080
# server服务端设置		
server.port		8443
server.connection-timeout		15000
server.compression.enabled		true
server.compression.mime-types		text/html,text/css,application/javascript,application/json
server.compression.min-response-size		2048
# ssl设置		
ssl.enabled	是否开启ssl，CFCA登录场景下需要开启	false
ssl.protocol		TLS
ssl.ciphers	服务端支持的加密套件	ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA

ssl.key-store	打包服务端证书和私钥成p12文件，放在指定目录下	/home/admin/midway/cert/server.p12
ssl.key-store-password	服务端p12密码	123456
ssl.key-store-type		PKCS12
ssl.key-alias	服务端证书别名	localhost
ssl.trust-store	服务端信任列表，需要将客户端证书的CA证书链打包到jks当中	/home/admin/midway/cert/certLogin/cfca.jks
ssl.trust-store-password	服务端信任列表密码	123456
ssl.client-auth		need
# 服务器IP配置		
allow_services		127.0.0.1
# 跨链配置		
odats.url		http://baas05.inc.alipay.net:9154
odats.userKeyPwd		123456
odats.userAccountNamePrefix		BAAS_ODATS_ACCOUNT
odats.timeout		30 # seconds
# spring配置		
spring.datasource.url	数据库连接地址	jdbc:sqlite:/home/admin/midway/data/midway.db
spring.datasource.username	数据库RDS账户	

spring.datasource.password	数据库RDS密码	
spring.datasource.driver-class-name		com.mysql.cj.jdbc.Driver
spring.http.multipart.maxFileSize		40000Mb
spring.http.multipart.maxRequestSize		40000Mb
spring.messages.basename		i18n/messages
spring.messages.encoding		UTF-8
spring.messages.cache-seconds		3
# mybatis配置		
mybatis.mapper-locations		classpath:mybatis/mapper/*Mapper.xml
mybatis.type-aliases-package		com.antfinancial.mychain.baas.midway.core.model
mybatis.type-handlers-package		com.antfinancial.mychain.baas.midway.core.dao.handler
# 日志存放路径设置		
logging.file	系统日志文件	/home/admin/midway/log/midway.log
logging.level.com.alipay.mychain		off
# 模块是否透出，多个模块用逗号隔开，kv形式		

modules

chain-insight-external 配置为 true 时，数据洞察版本依赖为 V3.0.0+，并确保 public-host 为可访问的数据洞察前端所在 host。

② 说明

各配置项的参数说明见下表。

```
modules:
  data-export: false
  odats: false
  odats-subs:
    register: true
    plugin: true
    relayer: true
    platform: true
  odats-support-chains:
    mychain: true
    fabric: true
    other: true
  chain-insight: false
  chain-insight-external: true
  chain-insight-subs:
    timeline-hyper-
link: true
  aldaba: true
  rest: false
  mychain-3: true
  contract: true
  costorage: false
  watchman: false
```

数据服务网关配置

data-gateway	- public-host 为数据洞察前端公网地址，如果数据洞察前端无法被公网访问，则 chain-insight-external 应配置为 false。 - private-host 为数据洞察内网地址。 - 此版本依赖数据洞察版本 V3.0.0 之后版本。	<pre>data-gateway: public-host: 139.224.206.12:8721 private-host: 127.0.0.1:8721 access_key_id: 123 access_key_secret: 123</pre>
# 自定义logo设置		
logo.enable		false
logo.name		蚂蚁链
# sso单点登录设置		
sso.enable		false
sso.gateway_url		https://oams.newone.com.cn/api/cms_login/oasso/tokenAuth
sso.sso_url		https://oams.newone.com.cn/api/cms_login/oasso/login
sso.return_url		http://localhost:8089/index.html#/login
sso.account_url		http://localhost:8082/getAccountList
sso.login_url		https://oams.newone.com.cn/cmsPortal2018/login
sso.cookie_name		auth.token
gateway.access_key		lelah7mies3zail
gateway.access_secret		chi9ohfuopheipha1uiW
# brain智能诊断及自愈设置		

brain.enable		false
brain.chain_black_list		
brain.treatment_black_list		
brain.event_check_period		60
brain.event_check_count		3
brain.event_check_count_expire		7200
brain.event_namespace		midway_brain_
brain.event_suffix		_event
brain.event_count_suffix		_eventCount
# interconn设置		
interconn.url		http://localhost:8888
interconn.enable		false
# 登录设置		
login.enable_ukey_login	是否开启ukey登录	false
ukey_option	ukey类型	CFCA
# 保密设置		
enable_data_integrity_check	是否开启数据完整性校验	false
enable_data_confidentiality	是否开启数据保密。保密范围：平台用户登录密码、联盟链admin账户密码、tls证书密码、ca私钥密码、trustCA密码，链节点ssh密码、ssh sudo密码	false

sm4_enc_dec_key_num	数据保密密钥	1
sm3_hmac_key_num	完整性校验密钥	1
# 数据库脱敏设置		
jasypt.encryptor.enable	数据库敏感字段加密设置，设置为 <code>true</code>，则数据库 <code>blockchain_node</code> 中的 <code>ssh_pwd</code>、<code>sudo_pwd</code> 字段会加密显示 ② 说明 如application-prod.yaml中的敏感字段也需要加密展示，除 <code>jasypt.encryptor.enable</code> 设置为 <code>true</code> 外，还要在启动文件 <code>start.sh</code> 中设置 - <code>Djasypt.encryptor.password=Bt%XJ^nlj8mz</code>	false

modules中各个配置项参数说明如下：

配置项	说明
data-export	数据导出
odats	跨链
odats-subs	跨链子功能
register	跨链注册与授权
plugin	跨链插件服务管理
relayer	跨链中继管理
platform	跨链平台互联管理

odats-support-chains	跨链支持的链
mychain	MYCHAIN
fabric	Fabric
other	其他
chain-insight	ChainInsight
chain-insight-external	是否外部独立化部署ChainInsight
chain-insight-subs	ChainInsight子功能
timeline-hyper-link	ChainInsight时间线超链接
aldaba	ALDABA
rest	REST
mychain-3	MYCHAIN V3
contract	合约开发
costorage	多方协同存储
watchman	监控中心

3.3. 负载均衡代理

安装依赖

```
yum -y install gcc zlib zlib-devel  
yum install -y pcre pcre-devel
```

下载安装包

```
cd /home/admin
#若可访问外网
wget http://nginx.org/download/nginx-1.20.1.tar.gz

#其他情况
手动拷贝安装 nginx-1.20.1.tar.gz

#解压
tar -zxvf nginx-1.20.1.tar.gz
```

编译安装Nginx

```
#设定配置：
./configure --prefix=/home/admin/nginx --conf-path=/home/admin/nginx/nginx.conf

#编译安装
cd /home/admin/nginx/
make
make install
```

修改配置文件

文件路径：`vi/home/admin/nginx/nginx.conf`

```
user root;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

#配置tcp转发
stream {
    upstream midway_8099 {
        server 10.0.7.18:8099 weight=50;
        server 10.0.7.19:8099 weight=50;
    }

    server {
        listen 8099;
        proxy_pass midway_8099;
        proxy_protocol on;
    }
}
```

```
http {
    include      mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #              '$status $body_bytes_sent "$http_referer" '
    #              '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile      on;
    #tcp_nopush    on;

    #keepalive_timeout    0;
    keepalive_timeout  65;

    client_max_body_size 8192m;

    #gzip  on;
    upstream midway{
        server 中途岛ipl:port weight=50;
        server 中途岛ip2:port weight=50;
        ...
    }

    server {
        #Nginx默认监听80端口，请根据项目实际情况修改
        listen      80;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location ~ (.*) {
            proxy_pass http://midway;
            proxy_connect_timeout 10;
            proxy_set_header    Host            $host;
            proxy_set_header    X-Real-IP        $remote_addr;
            proxy_set_header    X-Forwarded-For  $proxy_add_x_forwarded_for;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
```

```
# proxy_pass http://127.0.0.1;
#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
#location ~ /\.php$ {
#    root            html;
#    fastcgi_pass    127.0.0.1:9000;
#    fastcgi_index   index.php;
#    fastcgi_param   SCRIPT_FILENAME /scripts$fastcgi_script_name;
#    include         fastcgi_params;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen        8000;
#    listen        somename:8080;
#    server_name   somename  alias  another.alias;

#    location / {
#        root     html;
#        index    index.html index.htm;
#    }
#}

# HTTPS server
#
#server {
#    listen        443 ssl;
#    server_name   localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

#    ssl_session_cache    shared:SSL:1m;
#    ssl_session_timeout  5m;

#    ssl_ciphers  HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers  on;

#    location / {
#        root     html;
#        index    index.html index.htm;
#    }
#}
```

```
# }  
#}  
  
}
```

启动Nginx

```
#启动，默认采用80端口  
cd /home/admin/nginx/sbin  
./nginx
```

3.4. 拷贝mychain部署安装包

拷贝mychain部署安装包到各个集群节点的package目录下：

```
cp /home/admin/midway/package/MYCHAIN-2.24.8-Linux-x86_64-bin.tar.gz 到各个集群节点相同目录下
```

3.5. 配置中途岛集群节点信息

请前往 `cluster_node_info` 表配置中途岛集群节点信息。

```
INSERT INTO `cluster_node_info`  
(`id`,`ssh_ip`,`ssh_port`,`ssh_account`,`ssh_password`,`ssh_sudo_password`) VALUES (1,'  
127.0.0.1','22','root','123456','123456');  
INSERT INTO `cluster_node_info`  
(`id`,`ssh_ip`,`ssh_port`,`ssh_account`,`ssh_password`,`ssh_sudo_password`) VALUES (2,'  
127.0.0.2','22','root','123456','123456');  
//...插入更多集群节点信息...
```

⚠ 重要

配置完成后，请运维同学务必配置中途岛集群中各个节点时钟同步。

4. 部署验证

服务日志

查询运行日志 `midway.log`，是否包含启动成功关键字，并且无报错。

```
cd /home/admin/midway/log
grep 'Midway start successfully' midway.log --color
```

产品界面

使用浏览器访问负载均衡代理的服务地址，使用缺省的管理员账户登录中途岛，查看功能页面是否正常显示。

```
缺省地址： http://127.0.0.1:80/index.html#/login
缺省管理员用户名/密码： admin/MyChain@123.com!
```

5. 公司图标及名字定制配置

本章节介绍如何对中途岛平台登录页上公司logo图标、产品名字，以及产品页面顶部导航栏的logo图标进行定制化配置。

⚠ 重要

目前支持在2.0.1及以上的中途岛版本上进行logo定制。

第一步：修改配置文件

在未修改配置文件前，登录页上的默认文案展示如下：



通过修改配置文件 `application-prod.yaml` 中logo部分的内容，可以实现将登录页上欢迎使用蚂蚁链的文案修改成您想要的定制化文案，修改位置如下。

```
logo:
  enable: true # 启用:true, 禁用:false
  name: 自定义名称区块链 #设置自定义名称, 例如xx区块链, 如果没有自定义要求, 默认需要设置为“蚂蚁链”
```

🔍 说明

`enable` 字段必须设置为 `true`，示例中 `name` 所对应的文案内容最终在登录页上会展示成欢迎使用自定义名称区块链。

第二步：替换/新增logo图片

替换/新增如下两个图片（logo图片名称固定），位置如下：

```
[root@iZuf63pk6j9ja1xmevdyzyZ midway]# tree -L 1
.
├── bin
├── cert
├── data
├── logo
└── package

5 directories, 0 files
[root@iZuf63pk6j9ja1xmevdyzyZ midway]# tree logo/
logo/
├── index.png
└── login.png

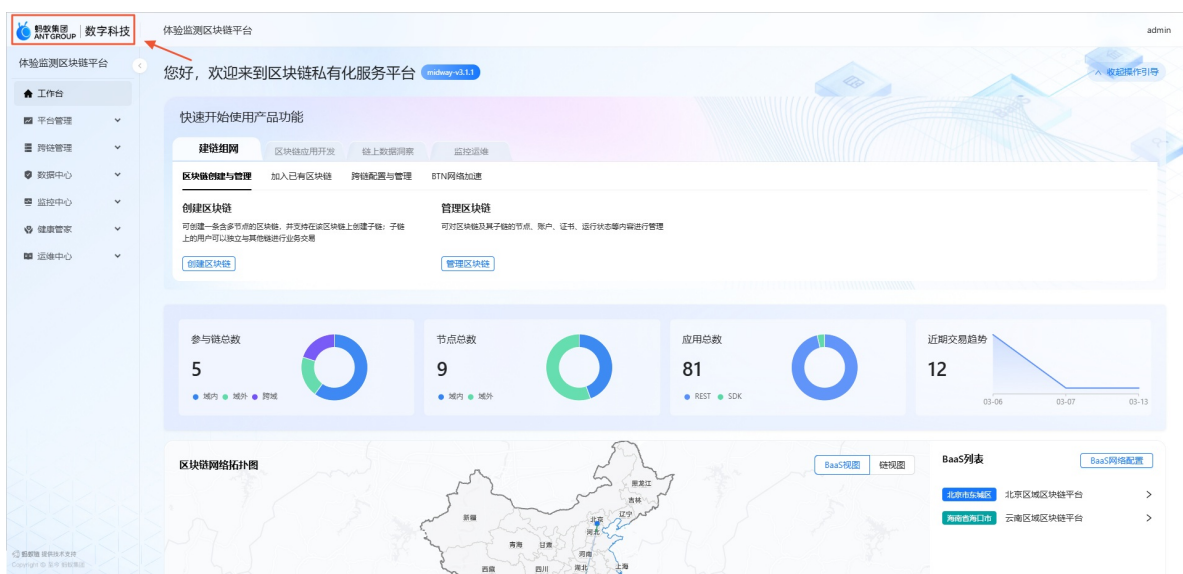
0 directories, 2 files
```

替换/新增的两个图片说明如下：

- login.png 为登录页logo，推荐图片大小设置：280 × 40 pixels。



- index.png 为页面顶部导航栏logo，推荐图片大小设置：231 × 39 pixels。



第三步：重启控制台服务（中途岛）

- 如果是第一次开启自定义logo
需要重启中途岛，执行如下命令：

```
cd /home/admin/midway/bin
./start.sh stop && ./start.sh start
```

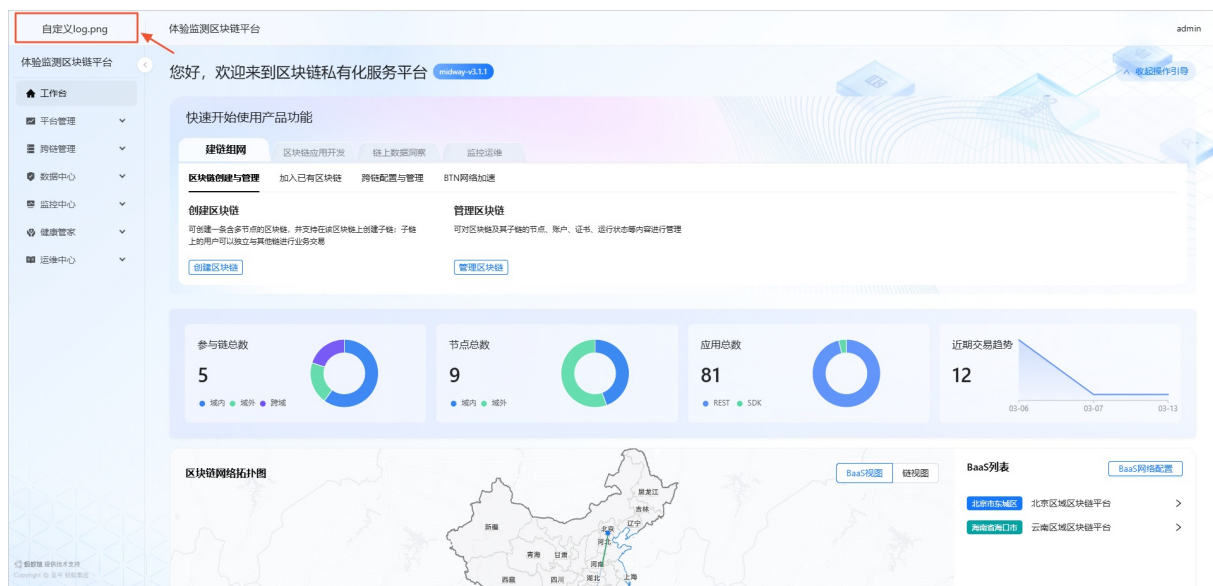
- 如果已经启用过该功能，希望再次更新logo
则直接覆盖logo目录下的对应logo图片文件，无需重启中途岛。

第四步：登录验证

在浏览器地址栏里输入中途岛服务器的IP地址，进入登录页后，可以看到logo和文案已被替换，替换后展示如下。



输入账户名及密码后，进入产品页，可以看到页面顶部导航栏上的logo也被成功替换，替换后展示如下。



6.REST服务部署

REST服务为上层业务应用提供易用、高效、可靠的统一区块链接入服务，本章节介绍如何部署REST服务。如果用户选择基于区块链SDK开发业务应用，则可以略过本章节。

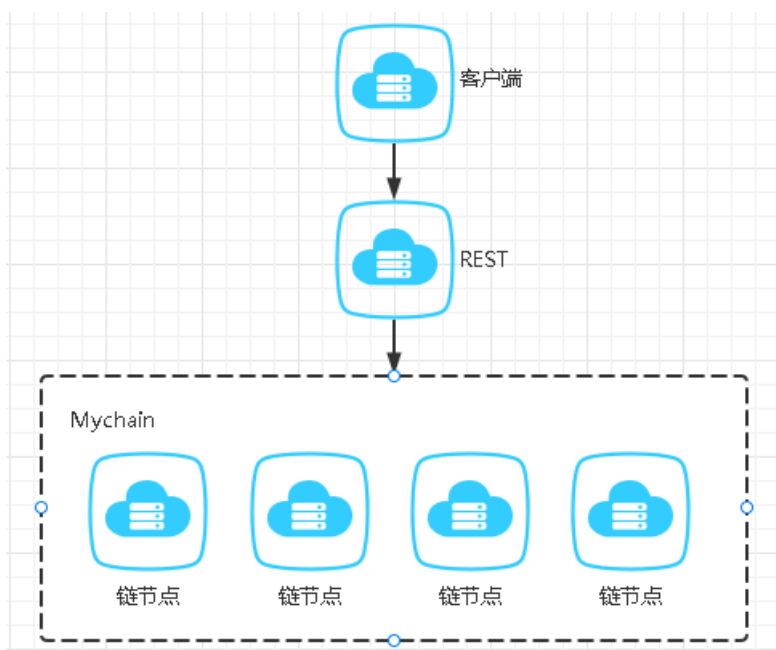
依赖及调用关系

- 数据库

用户需要准备一个MySQL数据库，用于存储REST服务的数据，其初始化SQL语句在sql目录中，目录展示如下。

```
baasrest
├── bin
│   ├── baasrest.sh
│   ├── start.sh
│   └── stop.sh
├── conf
│   ├── baasrest.config
│   ├── baas_admin.key <参考的rest管理账号私钥信息>
│   └── keys
│       └── private
│           ├── infosec-alipubkey.key
│           └── infosec-privatekey.key
├── lib
│   └── rest-gateway-web-2.1.0-executable.jar
└── sql
    ├── baas-rest.sql
    └── baas-rest-2.2.0.sql
```

- 调用关系



- 通信策略

- 确保客户端与REST服务之间能连通
- 确保REST服务与接入的链之间的18130通信端口能链通

部署REST服务

准备好MySQL数据库后，在承载REST服务的机器上部署REST服务的资源包，配置并启动REST服务，具体部署如下。

1. 配置服务参数。

REST服务配置内容在资源包中 `conf/baasrest.config` 文件中，主要包含数据库连接和区块链配置目录等配置项，根据实际环境进行客户化配置。

目前版本gateway和core使用同一个库，需要填写一致的db信息，配置文件示例及说明如下。

```
cat baasrest.config

# Gateway DB驱动类
db.gateway.driverClass=com.mysql.cj.jdbc.Driver
# Gateway DB连接URL，如果数据库查询展示时间存在问题，需要在数据库连接串中增加时区的设置，如：serverTimezone=Asia/Shanghai
db.gateway.jdbcUrl=jdbc:mysql://xxxxxxx:3306/dbname?
useUnicode=yes&characterEncoding=UTF-8
# Gateway DB用户名
db.gateway.username =root
# Gateway DB密码
db.gateway.password=${Gateway DB密码}
# Gateway DB连接池最大连接数
db.gateway.maximum-pool-size=10
# Gateway DB连接池最小空闲连接
db.gateway.minimum-idle=3
# Core DB驱动类
db.core.driverClass=com.mysql.cj.jdbc.Driver
# Core DB连接URL，如果数据库查询展示时间存在问题，需要在数据库连接串中增加时区的设置，如：serverTimezone=Asia/Shanghai
db.core.jdbcUrl=jdbc:mysql://xxxxxxx:3306/restcore?
useUnicode=yes&characterEncoding=UTF-8
# Core DB用户名
db.core.username =root
# Core DB密码
db.core.password=${Core DB密码}
# Core DB连接池最大连接数
db.core.maximum-pool-size=10
# Core DB连接池最小空闲连接
db.core.minimum-idle=3
# 日志级别
log.level=INFO
# 服务端口
server.port=8088
# 区块链配置目录
chain.rootpath=/home/admin/baasrest/rest-local
# Redis
spring.redis.host=10.xx.xx.xx
spring.redis.password=${redis密码}
spring.redis.database=4
spring.redis.port=6379
#模式：standalone、Cluster、single、sentinel
spring.redis.mode=standalone
# 是否使用数据库进行链配置管理
deploy.privateDeployUseDB=true
```

```
# 内容安全环境配置
financial.tech.gateway.endpoint=https://openapi.alipay.com/gateway.do
financial.tech.gateway.appScene=baasrest_openchain
financial.tech.gateway.appId=2021003162699047
financial.tech.gateway.readTimeout=200
# 配置的是访问蚂蚁开放平台加载的应用私钥和蚂蚁开放平台公钥的基础目录，对应的keys/private下的文件内容
# 需要根据实际情况进行调整。（和上面的financial.tech.gateway.appId保持一致，具体获取方式需要找账号提供者）
financial.tech.gateway.baseDir=/home/admin/baasrest/conf/
#配置的是需要经过内容安全审核的链列表，多个之间使用“,”分割，如果配置ALL代表所有的链都进行内容安全审核
#，配置NONE等无意义的，则所有链都不进行内容安全审核。
security.content.check.chainids=ALL
#内容安全检测场景类型，如有多个用英文半角逗号分隔（涉政：politics，涉黄：porn，违禁：illegal，暴恐
#：terrorism，谩骂：abuses，广告：advert）
financial.tech.gateway.checkLabels=politics,porn,illegal,terrorism,abuses
security.content.check.methods=DEPOSIT
```

2. 数据库初始化。

数据库初始化，需要执行目录 `baasrest/sql` 下的sql文件，具体如下。

- i. 先执行 `baas-rest.sql`，然后执行 `baas-rest-2.2.0.sql`。

说明

如果是从2.1.1版本升级，只需要执行 `baas-rest-2.2.0.sql`。

- ii. 最后补充执行以下两个语句。

```
alter table keys_and_other_data modify column create_time timestamp null;
alter table keys_and_other_data modify column update_time timestamp null;
```

3. 配置接入链。

当前版本的中途岛和BaaS REST进行了联动，相关链配置数据无需进行配置文件配置，通过中途岛的页面进行链的BaaS REST开通即可。

重要

请不要手动向数据库表 `blockchain_config` 中插数据。

4. 启动REST服务。

- i. 配置完REST服务参数和接入链后，执行资源包中的 `bin/start.sh` 脚本启动REST服务，执行命令如下所示。

```
./start.sh
```

- ii. 查看部署目录下的nohup.log，有无报错信息，待正常启动后执行如下查看版本是否正常返回。

```
curl 127.0.0.1:8088/version
```

检查修改midway的配置文件

1. 将rest部分的注释打开，进行相应修改，修改后示例如下。

```
mychain:
  run_mode: FULL # DEBUG | HANDY | FULL
  deploy_path: /home/admin/midway/
  chain_version: 0.10.2.12.5
  sqlite_path: midway.db
  enable_query_admin_info: true
  enable_maintain: true

rest:
  openRest: true
  bizid: 12eba21a66ed4d2795b833ce7ebd750c
  cipher-suit: ec
  rest-url: http://[fe80::216:3eff:fe1e:4b15]:8088
  access-id: baas_admin
# # access-secret: access.key
  access-secret: /home/admin/midway/cert/access.key
  access-secret-type: File
  tenantid: rest_test
# default-account: 1253000043120470X4
# default-account-key: 1253000043120470X4.key
# default-account-pwd: t1e2s3t4
```

- access-id是固定的内置id。
- access-secret是生成的，根据sql语句中插入公钥的对应私钥。
- 增加配置内容：`access-secret-type: File`。

❓ 说明

access-secret请使用绝对路径的方式，且确保已增加配置内容：`access-secret-type: File`。

2. 修改完成后重新启动中途岛。
3. （可选）如果需要更换accessId对应的公钥/私钥，可以重新生成一对公私钥，私钥放到中途岛中，公钥更新到 `user_private_key` 中，相应的更新sql参考如下。

```
update user_private_key
set public_key = '<匹配的公钥>'
where access_id = '<对应的AK>'
```

通过中途岛开通链的BaaS REST

登录midway的web页面，在左侧边栏中选择REST连接服务，根据实际情况进行链的开通和链与应用授权。



开通后，可以进入BaaS REST数据库，确认表：`blockchain_config`、`keys_and_other_data`的相关数据。

等待几分钟后，可以进入BaaS REST的服务器，使用命令查看链的连接情况：`netstat -an | grep 节点ip`

如：`netstat -an | grep 10.21.23.10`。

也可以通过 `netstat -an | grep 18130` 查看所有的链的连接情况。