

盈利宝即时通讯IM服务器部署

描述：

- IM服务器使用开源项目openfire进行二次开发，服务器命名为：**liberty**
- 服务器主机内存不低于4G

一、部署环境

1、java环境

下载DK1.8并解压，用于挂载到容器内使用

这里以解压到 /opt 为例：

```
# ls /opt/jdk1.8.0_211/  
bin  COPYRIGHT  include  javafx-src.zip  jre  lib  LICENSE  man  README.html  release  
src.zip  THIRDPARTYLICENSEREADME-JAVAFX.txt  THIRDPARTYLICENSEREADME.txt
```

2、docker环境

(1) 检查内核版本

Docker 要求 CentOS 系统的内核版本高于 3.10，通过 `uname -r` 命令查看当前的内核版本：

```
# uname -r  
3.10.0-514.el7.x86_64
```

(2) 安装和启动docker

- 安装docker

```
yum update -y  
yum -y install docker  
systemctl start docker      # 启动docker  
systemctl status docker     # 查看状态  
systemctl enable docker     # 开启启动
```

- 安装新版本docker-compose

```
curl -L https://github.com/docker/compose/releases/download/1.25.1-rc1/docker-compose-
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose
```

二、系统内核参数优化

1、修改文件句柄限制

(1) 查看系统全局变量

```
#查看允许每个程序的最大打开文件数
ulimit -n
# 输出
1024

#查看全部限制
ulimit -a

#系统全局参数file-max
cat /proc/sys/fs/file-max
```

- 查看某个进程文件数限制

```
cat /proc/$PID/limits
```

(2) 修改系统文件句柄限制

修改/etc/security/limits.conf 增加以下内容增加同时打开文件的句柄

```
*      soft    nofile    204800
*      hard    nofile    204800
```

(3) 修改/etc/profile 增加以下内容使用每次用户登录时设置为204800

```
ulimit -SHn 204800
```

2、增加可用端口范围

修改/etc/sysctl.conf 减少errot_connect_eaddrinuse错误

```
echo net.ipv4.ip_local_port_range =1024 65000 >> /etc/sysctl.conf

#立即生效
sysctl -p

#查看结果
sysctl -a | grep range
```

3、调整系统TCP缓冲区

(1) 查看缓冲区限制

```
sysctl -a | grep rmem
```

(2) 修改配置

- 打开配置文件**/etc/sysctl.conf**，加入以下内容

```
# 增加TCP最大缓冲区大小可设置使用setsockopt ()
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
# 增加Linux的自动调整TCP缓冲区限制
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
# 增加处理器的输入队列
net.core.netdev_max_backlog = 30000
# 建议巨帧的主机启用
net.ipv4.tcp_mtu_probing = 1
```

- 执行命令生效配置

```
sysctl -p
```

4、调整tcp keepalive

TCP保活的探测时间：**tcp**连接在空闲**600**秒后系统将发送探测包，探测包发送间隔为**30**秒，总共发送**5**个探测包，之后没回应则断掉该连接，减少系统资源消耗；

- 打开**/etc/sysctl.conf**，输入以下内容

```
# tcp keepalive
net.ipv4.tcp_keepalive_time = 600
net.ipv4.tcp_keepalive_intvl = 30
net.ipv4.tcp_keepalive_probes = 5
```

- 执行命令生效配置

```
sysctl -p
```

5、调整 nf_conntrack_max

(1) 查看值

```
sysctl -a |grep nf_conntrack
```

(2) 配置值

- 配置最大值

```
sysctl net.nf_conntrack_max=2097152
```

- 修改配置文件/etc/modprobe.d/nf_conntrack.conf, 输入内容:

```
# 这里配置后永久保持配置的最大值(net.nf_conntrack_max=2097152)
options nf_conntrack hashsize=262144
```

三、服务器部署

1、安装部署

(1) 解压部署包

```
tar -xf liberty_v4.5.0_deploy.tar.bz2
```

解压后liberty文件结构如下:

```
# ls
data logs docker-compose.yml liberty_v4.5.0_release.img port.sh
```

- data: 数据映射目录
- logs: 日志映射目录
- docker-compose.yml: docker配置文件
- liberty_v4.5.0_release.img: docker镜像文件
- port.sh: 端口占用查看脚本

(2) 安装镜像

```
docker load < liberty_v4.5.0_release.img
```

安装过程显示：

```
ebb9ae013834: Loading layer [=====>]
105.6MB/105.6MB
0ca7f54856c0: Loading layer [=====>]
24.1MB/24.1MB
1f59a4b2e206: Loading layer [=====>]
8.005MB/8.005MB
4955570ef830: Loading layer [=====>]
10.53MB/10.53MB
2b5bc5f550a6: Loading layer [=====>]
3.584kB/3.584kB
b1561d2c5ac2: Loading layer [=====>]
104.8MB/104.8MB
512b8c1e7d0b: Loading layer [=====>]
48.83MB/48.83MB
37560d68afd5: Loading layer [=====>]
4.608kB/4.608kB
6beb90127a82: Loading layer [=====>]
53.23MB/53.23MB
Loaded image: liberty:v4.5.0
```

查看镜像：

```
docker images
```

显示信息：

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
liberty	v4.5.0	a715e436ab1e	19 hours ago	347MB

(3) 修改配置文件

修改docker-compose.yml配置

需要修改地方：

- environment - 启动参数，给服务器分配的内存范围
- JDK路径的映射
- 端口映射的配置

```
version: '3.7'
services:
  liberty:
    restart: always
    image: liberty:v4.5.0
    container_name: liberty
    hostname: liberty-node1
    environment:
      DAEMON_OPTS: -Xms3072m -Xmx4096m
```

```

ports:
  - "9090:9090/tcp"          # http 方式登录到管理界面，
  - "9091:9091/tcp"          # https方式登录到管理界面，
  - "5222:5222/tcp"          # 客户端连接服务器tcp通信端口，
#   - "7777:7777/tcp"          # 文件传输代理服务器端口，可关闭
  - "5701:5701/tcp"          # 集群端口
volumes:
  - /etc/localtime:/etc/localtime:ro
  - /opt/jdk1.8.0_161:/usr/jdk # 映射主机JDK路径到容器内使用
  - ./data:/var/lib/openfire
  - ./logs:/var/log/openfire
dns:
  - 8.8.8.8
networks:
  - liberty_net
#   extra_hosts:              # 内网开启集群时需要映射主机，才能找到对应主机
#   - liberty-node2:172.20.0.2
#   - liberty-node3:192.168.10.233

networks:
  liberty_net:
    name: liber_net

```

配置文件说明：

- services：运行的容器服务
 - image：运行的镜像
 - hostname：指定容器内的主机名hostname
 - environment：配置参数变量，-Xms3072m -Xmx4096m 指定服务器可用的内存限制
 - ports：主机和容器的端口映射
 - 9090：http管理界面端口，服务器配置完成后可取消
 - 9091：https管理界面端口，web服务端口
 - 5222：客户端连接服务器tcp通信端口
 - 7777：文件传输代理服务器端口，可取消
 - 5701：集群使用端口，开启集群时，需要使用
 - volumes：数据卷映射
 - /etc/localtime:/etc/localtime:ro：映射主机时间给服务器
 - /opt/jdk1.8.0_161:/usr/jdk：映射主机JDK给容器使用，修改/opt/jdk1.8.0_161为主机JDK对应路径
 - ./data:/var/lib/openfire：服务器数据映射，包含服务器集群的配置文件
 - ./logs:/var/log/openfire：日志文件映射
- extra_hosts：配置外部主机的映射，开启集群时，如果是内网部署，需要映射主机地址，如果集群的每个主机都有域名能访问到，则不需要配置映射。

(4) 运行服务

```
docker-compose up -d
```

查看运行状态：

```
docker ps -a
```

显示信息：STATUS状态显示**Up**则已经正常启动

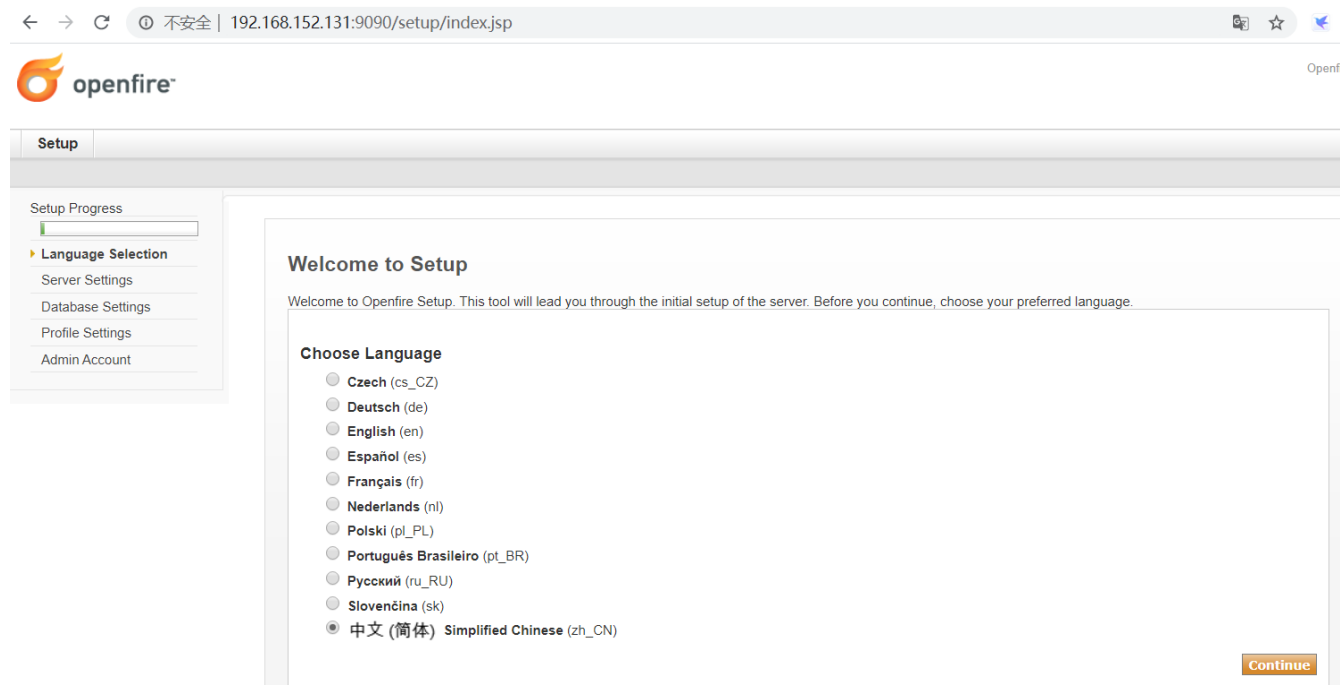
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
629c7cef1512	liberty:v4.5.0	"entrypoint.sh -Xms3..."	45 seconds ago	Up 44 seconds
3478-3479/tcp, 5223/tcp, 5229/tcp, 5262-5263/tcp, 5275-5276/tcp, 7070/tcp, 7443/tcp, 0.0.0.0:5222->5222/tcp, 7777/tcp, 0.0.0.0:9090-9091->9090-9091/tcp				
liberty				

(5) 部署配置

这里以内网主机**192.168.152.131**为例

1) 登录管理界面

浏览器输入地址：<http://192.168.152.131:9090/>，选择中文部署



2) 配置服务器

- 域：指定为 **liberty**（配置同一个集群时，所有服务器主机这个值必须一致）
- Host Name：服务器的域名（配置集群时，该域名能够在同一个集群其他主机访问到）

设置

安装进度

语言选择

服务器设置

数据库设置

外形设置

管理员帐户

服务器设置

下面是此服务器的主机设置。注：域的建议值基于此机器的网络设置。

域: openim

Server Host Name (FQDN): 这里输入服务器域名

管理界面端口: 9090

安全管理界面端口: 9091

配置参数加密方式: Blowfish

配置参数加密密钥:

继续

3) 配置数据库连接

- 选择标准数据库连接

设置

安装进度

语言选择

服务器设置

数据库设置

外形设置

管理员帐户

数据库设置

请选择希望如何连接到 Openfire 数据库。

标准数据库连接

使用外部数据库和内置连接池。

嵌入的数据库

使用一个自带数据库，由HSQLDB驱动。此选项不需要外部数据库配置，容易设置及方便部署。但是，它不能提供与外部数据库一样的性能。

继续

- 选择数据库驱动类

数据库URL串格式：

注：这里不要指定连接数据库的字符串，同时需要保证数据库表的字符集为**utf8mb4**，否则表情消息会存储异常

```
jdbc:mysql://HOSTNAME:3306/DATABASENAME?rewriteBatchedStatements=true
```

- HOSTNAME：数据库主机地址
- 3306：数据库端口
- DATABASENAME：数据库名称
- 输入连接的用户名和密码
- Maximum Connections：45

设置

安装进度

语言选择

服务器设置

数据库设置

外形设置

管理员帐户

数据库设置 - 标准连接

指定 JDBC 驱动程序和连接属性以连接到数据库。如果需要关于该过程的更多信息，请参阅随以下项附带的数据库文档：Openfire。
注意：在 Openfire 发行版的以下目录中包含了用于流行的数据库的数据库脚本：`[Openfire_HOME]/resources/database`。

数据库驱动选项：

MySQL

JDBC 驱动程序类：

com.mysql.cj.jdbc.Driver

数据库 URL：

jdbc:mysql://HOSTNAME:3306/DATABASENAME?rewriteBatchedStatements=true

用户名：

root

密码：

••••

Minimum Connections：

5

Maximum Connections：

45

连接超时：

1.0

 Days

注：连接到数据库可能需要 30 到 60 秒时间。

继续

4) 配置文件

- 选择默认

设置

安装进度

语言选择

服务器设置

数据库设置

外形设置

管理员帐户

配置文件设置

选择用户和组系统使用 Openfire。

☒ 默认设置

在 Openfire 数据库中保存用户和组。这是简化部署最好的选择。

☐ 只使用哈希值密码

数据库中使用时不可逆的哈希值来储存密码。客户端需要支持 PLAIN 和 SCRAM-SHA-1。

☐ 目录服务器 (LDAP)

使用 LDAP 协议整合为活动目录或 OpenLDAP 的目录服务器。用户和组被存储在目录中以只读形式存在。

继续

- 设置管理员账号信息，点击继续完成安装

设置

安装进度

语言选择

服务器设置

数据库设置

外形设置

管理员帐户

管理员帐户

请在下面输入系统管理员帐户 ("admin" 的用户名) 的设置。为该帐户选择一个不易猜中的密码非常重要 - 例如，至少有 6 个字符长并且同时包含字母和数字。如果已经设置了 admin 帐户 (不是新用户)，则可以跳过本步骤。

当前密码：

•••••

管理员电子邮件地址：

admin@example.com

新密码：

••••••

确认密码：

••••••

继续

跳过这一步

2、安装插件服务

通过管理员账号登录到服务器管理界面：

(1) 安装官方插件

在插件选项下，选择有效插件，如果没有显示出插件列表，则点击**立即升级**提示



- 安装 User Status Plugin 插件， 点击+安装

	User Service	(Deprecated) Please use the REST API Plugin. Allows administration of users via HTTP requests.	2.1.0 Oct 12, 2015	Roman Soldatow, Justin Hunt	2.5 MB	
	User Status Plugin	Openfire plugin to save the user status to the database.	1.2.2 Oct 29, 2019	Stefan Reuter	27.7 K	
	XML Debugger Plugin	Prints XML traffic to the stdout (raw and interpreted XML)	1.7.2 Jan 23, 2019	Ignite Realtime	20.1 K	

可以自动升级的有效插件列表 2020-1-8 17:40:58. 自动升级打开 [立即升级](#)

(2) 安装本地插件服务

在插件选项下，选择文件，选中本地需要安装的插件包，然后点击上传插件即可。



安装以下三个插件：

- 群管理插件：mucextendsplugin.jar
- web服务插件：restAPI.jar
- 监控服务插件：monitoring.jar

最终完成的插件列表如下：

插件

插件可以为服务器增加新的功能。当前已安装的插件见下表。下载新的插件请访问[可用的插件](#)页面。

插件	描述	版本	作者	重新启动	删除
 Monitoring Service	  Monitors conversations and statistics of the server.	1.8.2 Release 1	Jimmy		
 mucextendsplugin	 房间扩展插件，实现获取用户加入房间列表和加入房间入库。	1.0.1 Release 1	Jimmy		
 REST API	  Allows administration over a RESTful API.	1.3.10 Release 1	Jimmy		
 Search	  Provides support for Jabber Search (XEP-0055)	1.7.3	Ryan Graham		
 User Status Plugin	  Openfire plugin to save the user status to the database.	1.2.2	Stefan Reuter		

上传插件

插件文件（.jar）可以直接通过下面的表单上传。

未选择任何文件

- 如果不使用web管理界面安装插件，可直接把插件拷贝到服务器安装目录下：**data/plugins/**

3、服务器配置

通过管理员账号登录到服务器管理界面

(1) 设置服务属性

- 开启消息回执机制
 - 服务器->服务器管理器->系统属性->新加系统属性->添加新属性

属性名：stream.management.active
属性值：true

- 开启客户端多点登录
 - 服务器->服务器管理器->系统属性->新加系统属性->添加新属性

属性名：route.all-resources
属性值：true

(2) 缓存优化配置

- 在服务器->服务器管理器->系统属性->新加系统属性->添加新属性，输入完后，保存属性

添加新属性

属性名:

属性值:

配置值加密: ☐ 加密此配置值 ☒ 不加密此配置值

需要配置的属性项：（注意不能有空格，特别是size后面）

- 缓存大小的限制，设置为不限制，值为：-1

```
cache.ClientSessionInfoCache.size
cache.username2roster.size
cache.RosterItems.size
cache.userCache.size
cache.offlinemessage.size
cache.offlinePresence.size
cache.lastActivity.size
cache.listsCache.size
cache.lockOutCache.size
```

- 缓存时间的限制，设置为6小时：21600000（毫秒）

```
cache.ClientSessionInfoCache.maxLifetime
cache.username2roster.maxLifetime
cache.RosterItems.maxLifetime
cache.userCache.maxLifetime
cache.offlinemessage.maxLifetime
cache.offlinePresence.maxLifetime
cache.lastActivity.maxLifetime
cache.listsCache.maxLifetime
cache.lockOutCache.maxLifetime
```

配置项说明：

- cache.ClientSessionInfoCache：客户端会话缓存
- cache.username2roster：联系人关系缓存
- cache.RosterItems：联系人信息缓存
- cache.userCache：用户账号缓存
- cache.offlinemessage：离线消息缓存
- cache.offlinePresence：离线通知缓存
- cache.lastActivity：在线状态缓存
- cache.listsCache：用户隐私列表缓存
- cache.lockOutCache：封锁账号缓存

(3) 设置服务器时区

- 在服务器->服务器管理器->语言和时间->选择时区->保存设置

(4) 服务配置

- 设置客户端连接
- 在服务器->服务器设置->客户端连接->闲置连接设置->保存设置
 - 选择断开闲置客户端，输入值为**300**秒，选择发送Ping请求给闲置客户端

客户端连接

远程服务器到服务器

外部组件

连接管理器

HTTP绑定

更新管理

注册和登录

会话重复

离线信息

消息审计

私有数据

通信压缩

文件传送

Search Service Properties

XML Debugger Properties

User Status Settings

REST API

客户端连接设置

下面选项用来设置XMPP客户端如何连接到XMPP域服务器。点击查看当前连接本服务器的[客户端会话](#)。

纯文本连接（使用STARTTLS）

Openfire可以接受纯文本连接，这取决于设置，在这里配置可设置为加密连接（使用STARTTLS协议）。
☒ 启用
端口
[高级配置...](#)

加密连接（传统模式）

这种连接类型将使用加密方式建立（相对于使用STARTTLS）。这种连接类型通常被称为“旧式”或“传统”模式。
☒ 启用
端口
[高级配置...](#)

闲置连接设置

Openfire可以断开闲置状态的客户端连接。客户端连接闲置超过设定时间会被判断为闲置状态。
☐ 保留闲置客户端。
☒ 断开闲置客户端
 秒

服务器可以在断开闲置连接前发送XMPP Ping请求给该客户端。客户端必须回复 Ping请求，这样服务器能判断客户端必须响应 Ping请求。如果客户端不支持该Ping请求，必须返回错误（这本身就是一个响应）。
☒ 发送XMPP Ping请求给闲置客户端。

- 设置离线消息
- 在服务器->服务器设置->离线消息->离线消息策略->保存设置
 - 选择总是存储

服务器管理器

服务器设置

TLS/SSL证书

媒体服务

PubSub

统计表

档案文件

配置文件

客户端连接

远程服务器到服务器

外部组件

连接管理器

HTTP绑定

更新管理

注册和登录

会话重复

离线信息

消息审计

私有数据

通信压缩

文件传送

Search Service Properties

XML Debugger Properties

User Status Settings

离线消息设置

XMPP 提供了选项当 IM 消息是发送给尚未登录的用户时，由服务器存储并转发该消息。支持“离线消息”的存储转发是 X1 子邮件一样，可能会消耗大量的服务器空间。下面选项是关于离线消息处理的策略；请选择最适合的策略。

当前全部离线消息总大小： 20.52 MB

离线消息策略

☒ 存储 - 存储离线消息以便以后检索。下次接收者登录后将收到此消息。在下面选择一个存储策略及存储容量。

☐ 超出返回 - 存储消息至最大存储容量。超出存储容量之后，把消息退回发送者。

☒ 总是存储 - 总是存储消息（即使超出容量）。

☐ 超出丢弃 - 存储消息至最大存储容量。超出存储容量之后，直接丢弃消息并不通知。

单个用户离线消息的存储容量： KB

☐ 退回 - 从不存储离线消息，将消息退回发送者。

☐ 丢弃 - 从不存储离线消息，丢弃此消息且不通知发送者。

保存设置

- 设置搜索服务
- 在服务器->服务器设置->Search Service Properties->保存设置
 - 选择 enabled
 - 选上 Username, Name, Email

服务器管理器

服务器设置

TLS/SSL证书

媒体服务

PubSub

统计表

档案文件

配置文件

客户端连接

远程服务器到服务器

外部组件

连接管理器

HTTP绑定

更新管理

注册和登录

会话重复

离线信息

消息审计

私有数据

通信压缩

文件传送

Search Service Properties

XML Debugger Properties

User Status Settings

REST API

Search Service Properties

Use the form below to edit search service settings, these settings do not affect the user search in the admin console.

Service Enabled

You can choose to enable or disable user searches from clients. Disabling this services does not prevent user search

☒ Enabled - Clients will be able to search for users.

☐ Disabled - Clients will not be able to search for users.

Service Name

Search service Name: .myopenfire

Searchable Fields

Use the form below to enable which fields users can search on.

Enabled Fields

☒ Username

☒ Name

☒ Email

- 设置用户状态

- 在服务器->服务器设置->User Status Settings->保存设置
 - 选择 Archive user status forever.

服务器	用户 / 用户群	会话	分组聊天	插件
服务器管理器	服务器设置	TLS/SSL证书	媒体服务	PubSub
配置文件		统计表	档案文件	

客户端连接

远程服务器到服务器

外部组件

连接管理器

HTTP绑定

更新管理

注册和登录

会话重复

离线信息

消息审计

私有数据

通信压缩

文件传送

Search Service Properties

XML Debugger Properties

► User Status Settings

User Status Settings

This plugin automatically saves the last status (presence, IP address, logon and logoff time) per user. Optionally you can archive user status entries (IP address, logon and logoff time) for a specified time.

Basic

- ☐ Do not archive user status.
- ☒ Archive user status forever.
- ☐ Archive user status for days.

Save Settings

- 设置web服务
- 在服务器->服务器设置->REST API->保存设置
 - 选上 Enabled
 - 选上 Secret key auth
 - Logging -> Disabled

服务器管理器服务器设置TLS/SSL证书媒体服务PubSub统计表档案文件

配置文件
客户端连接
远程服务器到服务器
外部组件
连接管理器
HTTP绑定
更新管理
注册和登录
会话重复
离线信息
消息审计
私有数据
通信压缩
文件传送
Search Service Properties
XML Debugger Properties
User Status Settings
REST API

REST API Properties

Use the form below to enable or disable the REST API and configure the authentication.

REST API

The REST API can be secured with a shared secret key defined below or a with HTTP basic authentication. Moreover, for extra security you can specify the list of IP addresses that are allowed to use this service. An empty list means that the service can be accessed from any location. Addresses are delimited by commas.

☒ **Enabled** - REST API requests will be processed.
☐ **Disabled** - REST API requests will be ignored.

☐ HTTP basic auth - REST API authentication with Openfire admin account.
☒ **Secret key auth** - REST API authentication over specified secret key.
Secret key:

☐ Custom authentication filter classname - REST API authentication delegates to a custom filter implemented in some other plugin.

Note: changing back and forth from custom authentication filter forces the REST API plugin reloading

Filter classname:

Allowed IP Addresses:

Logging
☐ **Enabled** - Logging Enabled
☒ **Disabled** - Logging disabled

You can find here detailed documentation over the Openfire REST API: [REST API Documentation](#)

Save Settings

- 设置订阅号服务
- 在服务器->PubSub->更新设置
 - Service Enabled
 - Node Creation Restricted

服务器用户 / 用户群会话分组聊天插件

服务器管理器服务器设置TLS/SSL证书媒体服务PubSub统计表档案文件

PubSub Service
Node Summary

PubSub Service Configuration

Use the form below to update service settings.

Details

Service Enabled ☒

Node Creation Restricted
A true value means that not anyone can create a node, only the JIDs listed in 'Allowed to Create' are allowed to create nodes. ☒

Allowed to Create
Users that are allowed to create nodes. An empty list means that anyone can create nodes.

User	Action
<input type="text"/>	添加

System Administrators
Users that are system administrators of the PubSub service. A sysadmin has the same permissions as a node owner.

User	Action
<input type="text"/>	添加

Update 取消

- 设置消息存储

- 在服务器->归档文件->存档设置->更新设置
 - 选上 会话状态存档
 - 消息存档 三个都选上
 - 空闲时间：1
 - 最大时间：10
 - Max Message Age: 7
 - Retrievable Message: 7

启用或禁用信息和元数据存档

会话消息加密存档: ☐
 开启会话消息加密存档功能

会话状态存档: ☒
 记录某两人之间的会话，每次会话的信息数和会话持续时间。当信息存档启用时实际的信息内容将被记录。

信息存档:
 在用户之间的传送的所有信息存档文本可以使用关键词方式查找。

Archive one-to-one chats ☒

Archive group chats ☒

Archive stanzas for group chats ☒

Only archive conversations of the following room names (separated by comma)

空闲时间:
 会话结束后空闲的时间

最大时间:
 会话结束前可持续的最大时间

Max Message Age:
 The maximum number of days to keep messages before purging them from the database.

NOTE: Setting this value above 0 will PERMANENTLY DELETE any messages older than the specified number of days.

Retrievable Messages:
 The number of days worth of messages a user is allowed to retrieve.

- 群组默认配置
- 分组聊天->聊天服务->conference->默认房间
 - 在目录中列出房间
 - 永久房间
 - 允许成员邀请其他人
 - 允许成员修改昵称
 - 允许用户注册房间



(5) 禁用http访问

编辑docker配置文件：docker-compose.yml

去掉9090端口映射，重启容器

```
docker-compose restart
```

使用https访问管理界面的9091端口，可正常访问即可

四、集群配置

注：同一个集群内所有服务器主机的域必须一致，共享一个数据库，是一个无中心的集群。

1、开启集群端口

编辑docker配置文件，映射主机端口号5701

2、安装集群插件

在插件选项里安装以下两个插件

- Broadcast
- Hazelcast Clustering Plugin
- 注意集群内主机安装的插件版本要一致，避免兼容性问题

3、修改集群配置

配置文件位于服务器安装目录下：**data/conf/hazelcast-local-config.xml**

- 关闭广播方式加入集群：multicast enabled="false"
- 开启tcp-ip方式加入：tcp-ip enabled="true"
- 集群成员：{成员ip或域名地址}:5701
- 集群当前主机：interface 为当前主机的ip

```
# 原配置
    <join>
        <multicast enabled="true">
            <multicast-group>224.2.2.3</multicast-group>
            <multicast-port>54327</multicast-port>
        </multicast>
        <tcp-ip enabled="false"/>
    </join>
    <interfaces enabled="false">
        <interface>10.10.1.*</interface>
    </interfaces>

# 修改为
    <join>
        <multicast enabled="false"/>
        <tcp-ip enabled="true">
            <member>192.168.10.195:5701</member>
            <member>192.168.10.233:5701</member>
        </tcp-ip>
    </join>
    <interfaces enabled="true">
        <interface>192.168.10.233</interface>
    </interfaces>

# interface中是当前机器的IP地址
```

4、开启集群

在服务器管理界面->服务器管理器->集群->启用->保存设置



集群开启成功后，等几分钟（刚开始加入集群会比较慢）会在集群概述下显示成员列表

5、负载均衡

集群是一个无中心得集群，需要外加负载均衡支持，负载均衡代理的是TCP流

负载均衡两个方案：

- nginx
- haproxy

6、其他配置属性说明

Hazelcast有数个配置选项可作为Openfire的系统属性。

- 1、hazelcast.startup.delay.seconds (5) 运行Hazelcast插件的等待时间（秒）。这可以让Openfire在初始化集群的缓存时部署其他插件。
- 2、hazelcast.startup.retry.count (1) 集群首次启动失败后重复启动的次数。
- 3、hazelcast.startup.retry.seconds (10) 相邻两次启动的间隔时间。（秒）
- 4、hazelcast.max.execution.seconds (30) 在集群成员间执行同步任务时的最大等待时间。
- 5、hazelcast.config.xml.filename (hazelcast-cache-config.xml) Hazelcast插件配置文件的名字，便于安装自定义的缓存配置文件。
- 6、hazelcast.config.xml.directory ({OPENFIRE_HOME}/conf) 可以添加到插件类路径下的目录，便于加载自定义的Hazelcast配置文件。

7、hazelcast.config.jmx.enabled (false) 为Hazelcast集群启用JMX功能。

Hazelcast插件使用了XML配置建立器来初始化集群。默认情况下，集群的成员间会尝试通过广播下面的位置实现彼此互相发现。 IP地址： 224.2.2.3 端口： 54327