实验三：网络聊天室

1. 实验内容

建立云服务器，在云服务器上使用Nginx进行适当配置，然后使用Javascript、Nodejs、Websocket编程，实现网络聊天室。

1. 实验环境

腾讯云轻量级服务器

1. 实验步骤
2. 配置Nginx，nodejs，npm环境

| **组件** | **版本/配置** |
| --- | --- |
| 服务器 | 腾讯云 OpenCloudOS 8.6 |
| 公网 IP | 139.155.27.201 |
| Node.js | v16.9.0 |
| Nginx | 1.22.1 (自定义安装路径 /www/server/nginx) |
| PM2 | 5.3.0 |
| WebSocket 端口 | 8016 |
| 静态页面目录 | /var/www/chat/public |

1. 服务端部署​​
   1. 安装依赖​
2. mkdir -p /var/www/chat
3. cd /var/www/chat
4. npm init -y
5. npm install ws silly-datetime

2.2 编写 WebSocket 服务代码 server.js​

1. const WebSocket = require('ws');
2. const sd = require('silly-datetime');
3. const wss = new WebSocket.Server({ port: 8016 });
4. let clients = [];
5. wss.on('connection', (ws) => {
6. ws.name = '';
7. clients.push(ws);
8. console.log(`新用户连接，当前在线: ${clients.length}人`);
9. *// 处理消息*
10. ws.on('message', (message) => {
11. const msg = message.toString();
12. if (msg.startsWith('mynameis:') && msg.length > 8) {
13. ws.name = msg.replace('mynameis:', '');
14. broadcast(`${ws.name} 进入聊天室`, ws);
15. } else {
16. broadcast(`${ws.name}: ${msg}`, ws);
17. }
18. });
19. *// 处理断开连接*
20. ws.on('close', () => {
21. clients = clients.filter(client => client !== ws);
22. broadcast(`${ws.name} 离开聊天室`, ws);
23. console.log(`${ws.name} 离开，当前在线: ${clients.length}人`);
24. });
25. });
26. *// 广播消息*
27. function broadcast(msg, sender) {
28. clients.forEach(client => {
29. if (client !== sender && client.readyState === WebSocket.OPEN) {
30. client.send(msg);
31. }
32. });
33. }
34. *// 每秒推送时间*
35. setInterval(() => {
36. const time = sd.format(new Date(), 'YYYY-MM-DD HH:mm:ss');
37. clients.forEach(client => {
38. if (client.readyState === WebSocket.OPEN) {
39. client.send(`系统时间: ${time}`);
40. }
41. });
42. }, 1000);

2 **. 客户端部署​​**

**​​2.1 创建静态页面 public/index.html​**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>WebSocket 聊天室</title>
5. <style>
6. body { font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto; padding: 20px; }
7. #chat { border: 1px solid #ccc; height: 400px; overflow-y: auto; padding: 10px; margin: 10px 0; }
8. .message { margin: 5px 0; padding: 5px; background: #f0f0f0; border-radius: 4px; }
9. input { width: 200px; padding: 5px; }
10. button { padding: 5px 10px; }
11. </style>
12. </head>
13. <body>
14. <input type="text" id="username" placeholder="输入用户名">
15. <button onclick="setUsername()">设置用户名</button>
16. <div id="chat"></div>
17. <input type="text" id="message" placeholder="输入消息">
18. <button onclick="sendMessage()">发送</button>
19. <script>
20. const ws = new WebSocket('ws://139.155.27.201/chat');
21. function setUsername() {
22. const username = document.getElementById('username').value;
23. if (username) {
24. ws.send(`mynameis:${username}`);
25. }
26. }
27. function sendMessage() {
28. const message = document.getElementById('message').value;
29. if (message) {
30. ws.send(message);
31. document.getElementById('message').value = '';
32. }
33. }
34. ws.onmessage = (event) => {
35. const chatBox = document.getElementById('chat');
36. const div = document.createElement('div');
37. div.className = 'message';
38. div.textContent = event.data;
39. chatBox.appendChild(div);
40. chatBox.scrollTop = chatBox.scrollHeight;
41. };
42. </script>
43. </body>
44. </html>

**​**3. Nginx 反向代理配置​​

​​3.1 编辑配置文件 /www/server/nginx/conf/conf.d/chat.conf​

1. server {
2. listen 80;
3. server\_name 139.155.27.201;
4. *# WebSocket 代理*
5. location /chat {
6. proxy\_pass http://localhost:8016;
7. proxy\_http\_version 1.1;
8. proxy\_set\_header Upgrade $http\_upgrade;
9. proxy\_set\_header Connection "Upgrade";
10. proxy\_set\_header Host $host;
11. }
12. *# 静态文件服务*
13. location / {
14. root /var/www/chat/public;
15. index index.html;
16. }
17. }

3.2 重载 Nginx​

systemctl restart nginx

4. PM2 进程管理​​

​​4.1 启动服务​

1. cd /var/www/chat
2. pm2 start server.js --name chat-server
3. pm2 save

4.2 设置开机自启​

pm2 startup

四、功能测试​​

​​1 用户加入聊天室​​

​​操作​​：访问 http://139.155.27.201，输入用户名并点击“设置用户名”。

其他用户页面显示 [用户名] 进入聊天室。

服务器日志输出 新用户连接，当前在线: 1人。

​​2 消息发送​​

​​操作​​：在消息输入框输入内容并点击“发送”。

所有用户页面显示 [消息内容]。

服务器日志无错误输出。

​​3 用户退出​​

​​操作​​：关闭浏览器标签页。

其他用户页面显示 [用户名] 离开聊天室。

服务器日志输出 [用户名] 离开，当前在线: 0人。