Nginx Configuration Documentation

This document provides comprehensive documentation for the nginx configuration used in the AI Research Assistant application. The configuration implements a reverse proxy setup for a 3-service architecture with load balancing, rate limiting, and proper routing.

Architecture Overview

The nginx configuration manages traffic routing between three services:

1. **Frontend (Next.js)**: Port 3000 - User interface
2. **Express DB Server**: Port 3001 - Database operations via Supabase RPC
3. **FastAPI AI Server**: Port 8000 - AI/ML operations and chat functionality

Configuration File Structure

The main configuration file is located at `/nginx.conf` in the project root.

Core Configuration Blocks

Events Block

```
events {
    worker_connections 1024;
}
```

- **Purpose**: Defines the maximum number of simultaneous connections per worker process
- **Value**: 1024 connections - suitable for moderate traffic loads
- **Scalability**: Can be increased for high-traffic scenarios

PROF

HTTP Block Configuration

```
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;
    sendfile      on;
    keepalive_timeout 65;
    # ... upstream and server blocks
}
```

Key Directives:

- `include /etc/nginx/mime.types`: Includes MIME type definitions for proper content type handling

- `default_type application/octet-stream`: Default MIME type for unknown file types
- `sendfile on`: Enables efficient file serving
- `keepalive_timeout 65`: Keeps connections alive for 65 seconds to reduce overhead

Upstream Server Definitions

Frontend Upstream

```
upstream frontend {
    server frontend:3000;
}
```

- **Service**: Next.js frontend application
- **Port**: 3000
- **Container Name**: `frontend` (Docker service name)
- **Purpose**: Serves the user interface and static assets

Express DB Upstream

```
upstream express_db {
    server express-db-server:3001;
}
```

- **Service**: Express.js database server
- **Port**: 3001
- **Container Name**: `express-db-server` (Docker service name)
- **Purpose**: Handles ALL database operations via Supabase RPC functions

FastAPI AI Upstream

PROF

```
upstream fastapi_ai {
    server fastapi-ai-server:8000;
}
```

- **Service**: FastAPI AI/ML server
- **Port**: 8000
- **Container Name**: `fastapi-ai-server` (Docker service name)
- **Purpose**: Handles AI operations and chat functionality ONLY

Rate Limiting Configuration

Rate Limit Zones

```
```

```
limit_req_zone $binary_remote_addr zone=api:10m rate=10r/s;
limit_req_zone $binary_remote_addr zone=auth:10m rate=5r/s;
```

API Zone (`zone=api`):

- **Memory**: 10MB zone storage
- **Rate**: 10 requests per second per IP
- **Purpose**: General API endpoint protection
- **Estimated Capacity**: ~160,000 IP addresses

Auth Zone (`zone=auth`):

- **Memory**: 10MB zone storage
- **Rate**: 5 requests per second per IP
- **Purpose**: Authentication endpoint protection (stricter limits)
- **Estimated Capacity**: ~160,000 IP addresses

Server Block Configuration

Basic Server Settings

```
server {
    listen 80;
    server_name bruhmain.3utilities.com;
    # ... location blocks
}
```

- **Port**: 80 (HTTP)
- **Domain**: `bruhmain.3utilities.com`
- **Protocol**: HTTP (production should add HTTPS/SSL)

PROF

Location Block Routing

Frontend Routes

```
location / {
    proxy_pass http://frontend;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_cache_bypass $http_upgrade;
}
```

Purpose: Serves the Next.js frontend application

Key Features:

- **WebSocket Support**: `Upgrade` and `Connection` headers for Next.js hot reload
- **Client IP Preservation**: `X-Real-IP` and `X-Forwarded-For` headers
- **Protocol Detection**: `X-Forwarded-Proto` for HTTPS detection
- **Cache Bypass**: Bypasses cache for WebSocket upgrades

Proxy Headers Explanation:

- `Host`: Preserves original host header
- `X-Real-IP`: Client's real IP address
- `X-Forwarded-For`: Chain of proxy IPs
- `X-Forwarded-Proto`: Original protocol (http/https)

Database API Routes

```
location /api/ {
    limit_req zone=api burst=20 nodelay;
    proxy_pass http://express_db;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

Purpose: Routes all `/api/*` requests to Express DB server

Rate Limiting:

PROF

- **Zone**: `api` (10 req/s base rate)
- **Burst**: 20 additional requests allowed in burst
- **Mode**: `nodelay` - immediate processing of burst requests

Handled Endpoints:

- `/api/users/*` - User management
- `/api/groups/*` - Group operations
- `/api/sessions/*` - Session management
- `/api/messages/*` - Message handling
- `/api/papers/*` - Paper operations
- `/api/feedback/*` - Feedback system
- `/api/group-chat/*` - Group chat functionality
- `/api/ai-metadata/*` - AI metadata management

Authentication Routes (Special Handling)

```
location /api/auth/ {
    limit_req zone=auth burst=10 nodelay;
    proxy_pass http://express_db;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}
```

Purpose: Special handling for authentication endpoints with stricter rate limits

Rate Limiting:

- **Zone**: `auth` (5 req/s base rate)
- **Burst**: 10 additional requests allowed
- **Reasoning**: Authentication attempts should be limited to prevent brute force attacks

Handled Endpoints:

- `/api/auth/status` - Authentication status
- `/api/auth/me` - User profile retrieval
- `/api/auth/sync-profile` - Profile synchronization

AI/ML API Routes

```
location /ai/ {
    limit_req zone=api burst=5 nodelay;
    proxy_pass http://fastapi_ai/;
    proxy_http_version 1.1;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # Longer timeout for AI operations
    proxy_read_timeout 300s;
    proxy_connect_timeout 300s;
    proxy_send_timeout 300s;
}
```

Purpose: Routes all `/ai/*` requests to FastAPI AI server

Rate Limiting:

- **Zone**: `api` (10 req/s base rate)

- **Burst**: 5 additional requests (lower than general API)
- **Reasoning**: AI operations are computationally expensive

Extended Timeouts:

- **Read Timeout**: 300 seconds (5 minutes)
- **Connect Timeout**: 300 seconds
- **Send Timeout**: 300 seconds
- **Reasoning**: AI/ML operations can take significant time to complete

Handled Endpoints:

- `/ai/chat/*` - Chat functionality
- `/ai/health` - AI service health check
- `/ai/docs` - FastAPI auto-generated documentation

Important Note: The trailing slash in `proxy_pass http://fastapi_ai/;` ensures proper URL rewriting.

Health Check Route

```
location /health {
    access_log off;
    return 200 "healthy\n";
}
```

Purpose: Simple nginx-level health check

Features:

- **No Logging**: `access_log off` to reduce log noise
- **Direct Response**: Returns immediately without proxying
- **Status**: HTTP 200 with "healthy" message
- **Use Case**: Load balancer health checks, monitoring systems

PROF

Security Features

Rate Limiting Protection

1. **API Protection**: 10 requests/second with burst capability
2. **Auth Protection**: 5 requests/second with smaller burst
3. **Per-IP Limiting**: Uses client IP address for rate limiting
4. **Memory Efficient**: Uses binary representation of IP addresses

Header Security

1. **Client IP Preservation**: Maintains audit trails
2. **Protocol Detection**: Enables proper HTTPS redirects

3. **Host Header Validation**: Prevents host header attacks

Request Size Limits

Default nginx limits apply:

- **Client Max Body Size**: 1MB (default)
- **Large Client Header Buffers**: 4 8k (default)

Performance Optimization

Connection Management

1. **Keep-Alive**: 65-second timeout reduces connection overhead
2. **HTTP/1.1**: Modern HTTP version for better performance
3. **Sendfile**: Efficient static file serving

Caching Strategy

1. **Cache Bypass**: WebSocket upgrades bypass cache
2. **Static Assets**: Served directly by frontend service
3. **API Responses**: No caching for dynamic content

Monitoring and Logging

Access Logs

- **Location**: Default nginx access log location
- **Format**: Standard nginx log format
- **Exception**: Health check endpoint excludes logging

Error Logs

PROF

- **Location**: Default nginx error log location
- **Level**: Standard error logging
- **Upstream Errors**: Logged with upstream server details

Production Considerations

SSL/HTTPS Configuration

Current: HTTP only (port 80)

Recommended Production Addition:

```
server {
    listen 443 ssl http2;
    server_name bruhmain.3utilities.com;

    ssl_certificate /path/to/certificate.crt;
```

```
    ssl_certificate_key /path/to/private.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # ... existing location blocks
}

# HTTP to HTTPS redirect
server {
    listen 80;
    server_name bruhmain.3utilities.com;
    return 301 https://$server_name$request_uri;
}
```

Additional Security Headers

Recommended Production Addition:

```
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;
```

Load Balancing

For high availability, consider multiple upstream servers:

```
upstream frontend {
    server frontend1:3000;
    server frontend2:3000;
    server frontend3:3000;
}
```

PROF

Troubleshooting

Common Issues

1. **502 Bad Gateway**: Upstream service unavailable
    - Check service status: `docker-compose ps`
    - Check service logs: `docker-compose logs [service]`
2. **Rate Limiting Errors**: Too many requests
    - Check client request patterns
    - Adjust rate limits if needed

- Implement client-side rate limiting

3. **Timeout Errors**: AI operations taking too long
    - Check AI service performance
    - Consider increasing timeout values
    - Implement async processing for long operations

Debugging Commands

```
# Test nginx configuration
nginx -t

# Reload nginx configuration
nginx -s reload

# Check nginx status
nginx -s status

# View nginx access logs
tail -f /var/log/nginx/access.log

# View nginx error logs
tail -f /var/log/nginx/error.log
```

Docker Context Commands

```
# Check service connectivity
docker-compose exec nginx curl http://frontend:3000/health
docker-compose exec nginx curl http://express-db-server:3001/health
docker-compose exec nginx curl http://fastapi-ai-server:8000/health

# Test rate limiting
for i in {1..15}; do curl -w "%{http_code}\n"
http://localhost/api/health; done
```

PROF

Configuration Best Practices

1. **Service Discovery**: Use Docker service names for upstream servers
2. **Rate Limiting**: Implement different limits for different endpoint types
3. **Timeout Configuration**: Set appropriate timeouts for different service types
4. **Health Checks**: Implement health checks at multiple levels
5. **Logging**: Balance between monitoring needs and log volume
6. **Security**: Always implement SSL/TLS in production
7. **Monitoring**: Set up upstream health monitoring
8. **Scalability**: Design upstream configuration for horizontal scaling

Development vs Production

Development Environment

- HTTP only (port 80)
- Relaxed security headers
- Detailed logging enabled
- Direct service names (Docker Compose)

Production Environment Additions Needed

- HTTPS/SSL termination
- Enhanced security headers
- Log rotation and management
- External load balancer integration
- CDN integration for static assets
- Web Application Firewall (WAF)
- DDoS protection

This nginx configuration provides a solid foundation for the AI Research Assistant's 3-service architecture while maintaining performance, security, and scalability considerations.