CasperVPN Troubleshooting Guide

📋 Table of Contents

Quick Diagnostics

First Steps

When something goes wrong, run these commands:

```
# 1. Check service health
./scripts/health-check.sh

# 2. Check Docker containers
docker-compose ps

# 3. View recent logs
./scripts/logs.sh all --tail=50

# 4. Check system resources
docker stats --no-stream
```

Emergency Commands

```
# Restart all services
docker-compose restart

# Restart specific service
docker-compose restart <service-name>

# Stop everything
docker-compose down

# Fresh start
docker-compose down && docker-compose up -d
```

Common Issues

Issue: Services Won't Start

Symptoms:
- Container exits immediately
- "Unhealthy" status
- No response from service

Diagnosis:

```bash
# Check logs
./scripts/logs.sh <service-name>

# Check container status
docker-compose ps <service-name>

# Inspect container
docker inspect <container-name>
```

Solutions:

1. **Port already in use**
   ```bash
   # Find process using port
   lsof -i :8080 # Mac/Linux
   netstat -ano | findstr :8080 # Windows

# Kill process or change port in .env
```

1. **Insufficient resources**
   ```bash
   # Check Docker resources
   docker system df

# Clean up
docker system prune -a
```

1. **Missing dependencies**
   ```bash
   # Rebuild container
   docker-compose up -d --build <service-name>
   ```

2. **Configuration error**
   ```bash
   # Check .env file
   cat .env | grep

# Validate docker-compose.yml
docker-compose config
```

Issue: Cannot Connect to Service

Symptoms:

- Connection refused
- Timeout errors
- 502 Bad Gateway

Diagnosis:

```
# Check if service is running
docker-compose ps

# Check logs
./scripts/logs.sh <service-name>

# Test connection
curl http://localhost:<port>/health
```

Solutions:

1. **Service not running**
   ```bash
   docker-compose up -d <service-name>
   ```

2. **Wrong port**
   ```bash
   # Check port mapping
   docker-compose ps

# Or in .env
cat .env | grep PORT
```

1. **Firewall blocking**
   ```bash
   # Check firewall rules
   sudo ufw status

# Allow port
sudo ufw allow /tcp
```

Issue: Slow Performance

Symptoms:

- High response times
- Timeouts
- CPU/Memory spikes

Diagnosis:

```
# Check resource usage
docker stats --no-stream

# Check system metrics
top
df -h

# Check slow queries (PostgreSQL)
docker-compose exec postgres psql -U casperuser -d caspervpn \
  -c "SELECT query, mean_exec_time FROM pg_stat_statements ORDER BY mean_exec_time
DESC LIMIT 10;"
```

Solutions:

1. **High CPU usage**
   ```bash
   # Scale service
   docker-compose up -d –scale api=3
   ```

Or increase resources
Edit docker-compose.yml resources.limits
```

1. **High memory usage**
   ```bash
   # Check memory leaks
   ./scripts/logs.sh | grep -i "memory"
   ```

Restart service
docker-compose restart
```

1. **Database slow**
   ```bash
   # Analyze and vacuum
   docker-compose exec postgres psql -U casperuser -d caspervpn \
   -c "VACUUM ANALYZE;"
   ```

Check connections
docker-compose exec postgres psql -U casperuser -d caspervpn \
-c "SELECT count(*) FROM pg_stat_activity;"
```

## Issue: Database Connection Errors

**Symptoms:**
- "Connection refused"
- "Too many connections"
- "Authentication failed"

**Diagnosis:**

```
# Check database status
docker-compose ps postgres

# Test connection
docker-compose exec postgres pg_isready

# Check logs
./scripts/logs.sh postgres
```

Solutions:

1. **Database not running**
   bash
   ```
   docker-compose up -d postgres
   ```

2. **Wrong credentials**
   ```bash
   # Check .env
   cat .env | grep DB_
   ```

Test manually
docker-compose exec postgres psql -U casperuser -d caspervpn
```

1. **Too many connections**
   ```bash
   # Check active connections
   docker-compose exec postgres psql -U casperuser -d caspervpn \
   -c "SELECT count(*) FROM pg_stat_activity;"
   ```

Kill idle connections
docker-compose exec postgres psql -U casperuser -d caspervpn \
-c "SELECT pg_terminate_backend(pid) FROM pg_stat_activity WHERE state = 'idle';"
```

1. **Database corrupted**
   bash
   ```
   # Restore from backup
   ./scripts/restore.sh backups/latest.tar.gz
   ```

# Service-Specific Issues

## .NET API Issues

### Issue: API returns 500 errors

```
# Check logs
./scripts/logs.sh api --tail=100

# Check dependencies
docker-compose exec api dotnet --info

# Restart API
docker-compose restart api
```

Issue: Hot reload not working

```
# Ensure using dev environment
docker-compose -f docker-compose.dev.yml down
docker-compose -f docker-compose.dev.yml up -d

# Check volume mounts
docker inspect caspervpn-api-dev | grep Mounts -A 20
```

React Admin Issues

Issue: Blank page / White screen

```
# Check browser console (F12)
# Check logs
./scripts/logs.sh admin-react

# Rebuild
docker-compose up -d --build admin-react
```

Issue: API calls failing

```
# Check API URL in environment
docker-compose exec admin-react env | grep API_URL

# Check CORS settings in API
./scripts/logs.sh api | grep -i cors
```

Rust Agent Issues

Issue: Agent not reporting metrics

```
# Check metrics endpoint
curl http://localhost:8081/metrics

# Check logs
./scripts/logs.sh server-agent

# Rebuild
docker-compose up -d --build server-agent
```

PHP Admin Issues

Issue: PHP errors / 500 status

```
# Check PHP logs
./scripts/logs.sh admin-php | grep -i error

# Check permissions
docker-compose exec admin-php ls -la storage/

# Clear cache
docker-compose exec admin-php php artisan cache:clear
docker-compose exec admin-php php artisan config:clear
```

Issue: Composer dependencies

```
# Reinstall dependencies
docker-compose exec admin-php composer install

# Update dependencies
docker-compose exec admin-php composer update
```

Database Issues

PostgreSQL

Issue: Database won't start

```
# Check logs
./scripts/logs.sh postgres

# Check data volume
docker volume inspect caspervpn-postgres-data

# If corrupted, restore
./scripts/restore.sh backups/latest.tar.gz
```

Issue: Slow queries

```
# Enable slow query log
docker-compose exec postgres psql -U casperuser -d caspervpn \
  -c "ALTER SYSTEM SET log_min_duration_statement = 1000;"

# View slow queries
./scripts/logs.sh postgres | grep "duration:"

# Analyze query
docker-compose exec postgres psql -U casperuser -d caspervpn \
  -c "EXPLAIN ANALYZE <your-query>;"
```

Issue: Disk space full

```
# Check database size
docker-compose exec postgres psql -U casperuser -d caspervpn \
  -c "SELECT pg_size_pretty(pg_database_size('caspervpn'));"

# Vacuum
docker-compose exec postgres psql -U casperuser -d caspervpn \
  -c "VACUUM FULL;"
```

Redis

Issue: Redis out of memory

```
# Check memory usage
docker-compose exec redis redis-cli INFO memory

# Clear cache
docker-compose exec redis redis-cli FLUSHALL

# Restart Redis
docker-compose restart redis
```

Issue: Redis connection timeout

```
# Check Redis status
docker-compose exec redis redis-cli ping

# Check max connections
docker-compose exec redis redis-cli CONFIG GET maxclients

# Increase max connections
docker-compose exec redis redis-cli CONFIG SET maxclients 10000
```

Network Issues

Issue: Services Can't Communicate

Diagnosis:

```
# Check network
docker network ls

# Inspect network
docker network inspect caspervpn-network

# Test connectivity
docker-compose exec api ping postgres
```

Solution:

```
# Recreate network
docker-compose down
docker network prune
docker-compose up -d
```

Issue: External Access Not Working

Diagnosis:

```
# Check port mappings
docker-compose ps

# Check firewall
sudo ufw status

# Test from outside
curl http://<server-ip>:8080/health
```

Solution:

```
# Open ports
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Check nginx config
docker-compose exec nginx nginx -t

# Restart nginx
docker-compose restart nginx
```

Performance Issues

High CPU Usage

Diagnosis:

```
# Check which container is using CPU
docker stats --no-stream

# Check processes in container
docker-compose exec <service> top
```

Solutions:

1. **Optimize code:** Profile and fix hot spots
2. **Scale horizontally:** Add more instances
3. **Increase resources:** Edit docker-compose.yml

```yaml
services:
  api:
    deploy:
      resources:
        limits:
          cpus: '2'
          memory: 2G
```

High Memory Usage

Diagnosis:

```
# Check memory usage
docker stats --no-stream

# Check for memory leaks
./scripts/logs.sh <service> | grep -i "out of memory"
```

Solutions:

1. **Restart service:** Temporary fix
2. **Increase memory limit:** Edit docker-compose.yml
3. **Fix memory leaks:** Profile application

Disk Space Issues

Diagnosis:

```
# Check disk usage
df -h

# Check Docker disk usage
docker system df

# Check log sizes
du -sh /var/lib/docker/containers/*/*-json.log
```

Solutions:

```
# Clean Docker
docker system prune -a --volumes

# Clean logs
sudo truncate -s 0 /var/lib/docker/containers/*/*-json.log

# Rotate logs
# Edit docker-compose.yml
logging:
  driver: "json-file"
  options:
    max-size: "10m"
    max-file: "3"
```

Docker Issues

Issue: Docker Daemon Not Running

```
# Start Docker (Mac)
open -a Docker

# Start Docker (Linux)
sudo systemctl start docker

# Check status
sudo systemctl status docker
```

Issue: Permission Denied

```
# Add user to docker group
sudo usermod -aG docker $USER

# Logout and login again
# Or temporarily:
newgrp docker
```

Issue: Out of Disk Space

```
# Clean everything
docker system prune -a --volumes

# Remove specific items
docker image prune -a
docker volume prune
docker network prune
```

Issue: Build Failures

```
# Clear build cache
docker builder prune

# Build with no cache
docker-compose build --no-cache

# Check Dockerfile syntax
docker-compose config
```

Monitoring Issues

Prometheus Not Scraping

```
# Check targets
curl http://localhost:9090/api/v1/targets | jq .

# Check service metrics
curl http://localhost:8080/metrics

# Reload Prometheus config
curl -X POST http://localhost:9090/-/reload
```

Grafana Not Showing Data

```
# Check datasource
curl http://localhost:3001/api/datasources

# Test Prometheus connection
curl http://localhost:9090/api/v1/query?query=up

# Check Grafana logs
./scripts/logs.sh grafana
```

Alerts Not Firing

```
# Check alert rules
curl http://localhost:9090/api/v1/rules | jq .

# Check Alertmanager
curl http://localhost:9093/api/v1/alerts

# Test alert
curl -X POST http://localhost:9093/api/v1/alerts \
  -d '[{"labels":{"alertname":"test","severity":"warning"}}]'
```

Getting Help

Logs to Collect

When asking for help, provide:

```
# Service logs
./scripts/logs.sh all --tail=500 > logs.txt

# Health check
./scripts/health-check.sh > health.txt

# Docker status
docker-compose ps > docker-status.txt

# System info
docker version > system-info.txt
docker-compose version >> system-info.txt
docker system df >> system-info.txt
```

Debug Mode

Enable debug logging:

```
# In .env
RUST_LOG=debug
APP_DEBUG=true
ASPNETCORE_ENVIRONMENT=Development

# Restart services
docker-compose restart
```

Creating an Issue

Include:
1. Clear description of the problem
2. Steps to reproduce
3. Expected vs actual behavior
4. Logs (from above)
5. Environment (dev/staging/production)
6. Docker/OS versions

Preventive Measures

Regular Maintenance

```
# Weekly
./scripts/backup.sh
docker system prune

# Monthly
# Review logs for patterns
# Update dependencies
# Check disk space
# Review monitoring alerts
```

Health Monitoring

Set up automated health checks:

```
# Add to crontab
*/5 * * * * /path/to/caspervpn/scripts/health-check.sh
0 2 * * * /path/to/caspervpn/scripts/backup.sh
```

Additional Resources

- DevOps Overview (./DEVOPS.md)
- Deployment Guide (./DEPLOYMENT.md)
- Monitoring Guide (./MONITORING.md)
- Docker Documentation (https://docs.docker.com/)
- Prometheus Documentation (https://prometheus.io/docs/)

Last Updated: December 5, 2025
Version: 1.0.0