



Diagrama Visual do Problema e Solução

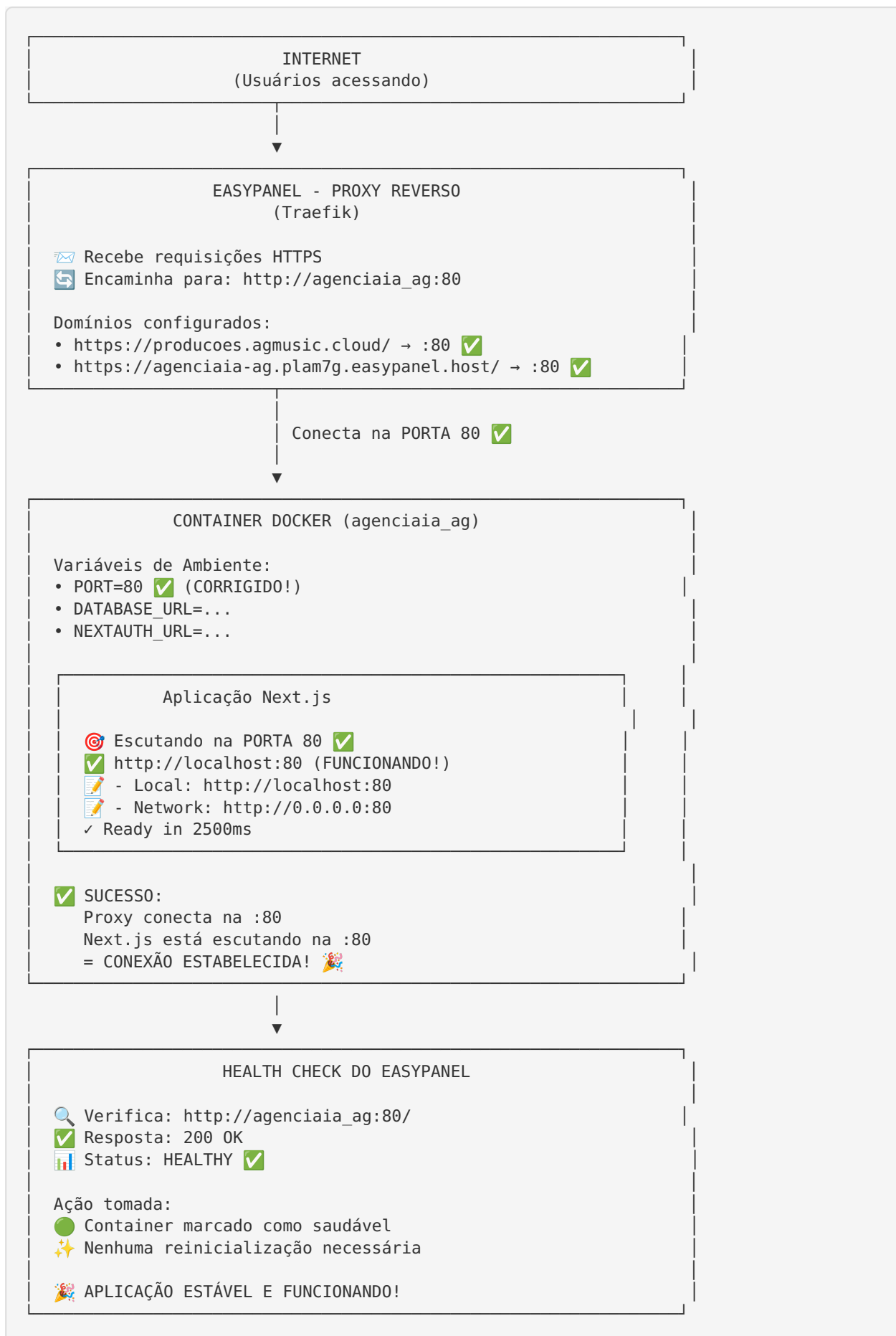


Entendendo o Problema

Cenário ANTES da Correção (X PROBLEMA)




















Cenário DEPOIS da Correção (✅ FUNCIONANDO)

















Fluxo de Requisição Comparado

ANTES (Falhando)

1. Usuário

2. DNS resolve para Easypanel

3. HTTPS  Proxy Easypanel

4. Proxy tenta: http://agenciaia_ag:80

5. **Container** porta :80   VAZIO

 Porta :3000   **Next.js** está aqui

6. Timeout / Connection Refused

7. Health Check FALHA

8. SIGTERM enviado

9. **Container** reinicia

10. VOLTA PARA O PASSO 1 (ciclo infinito) 

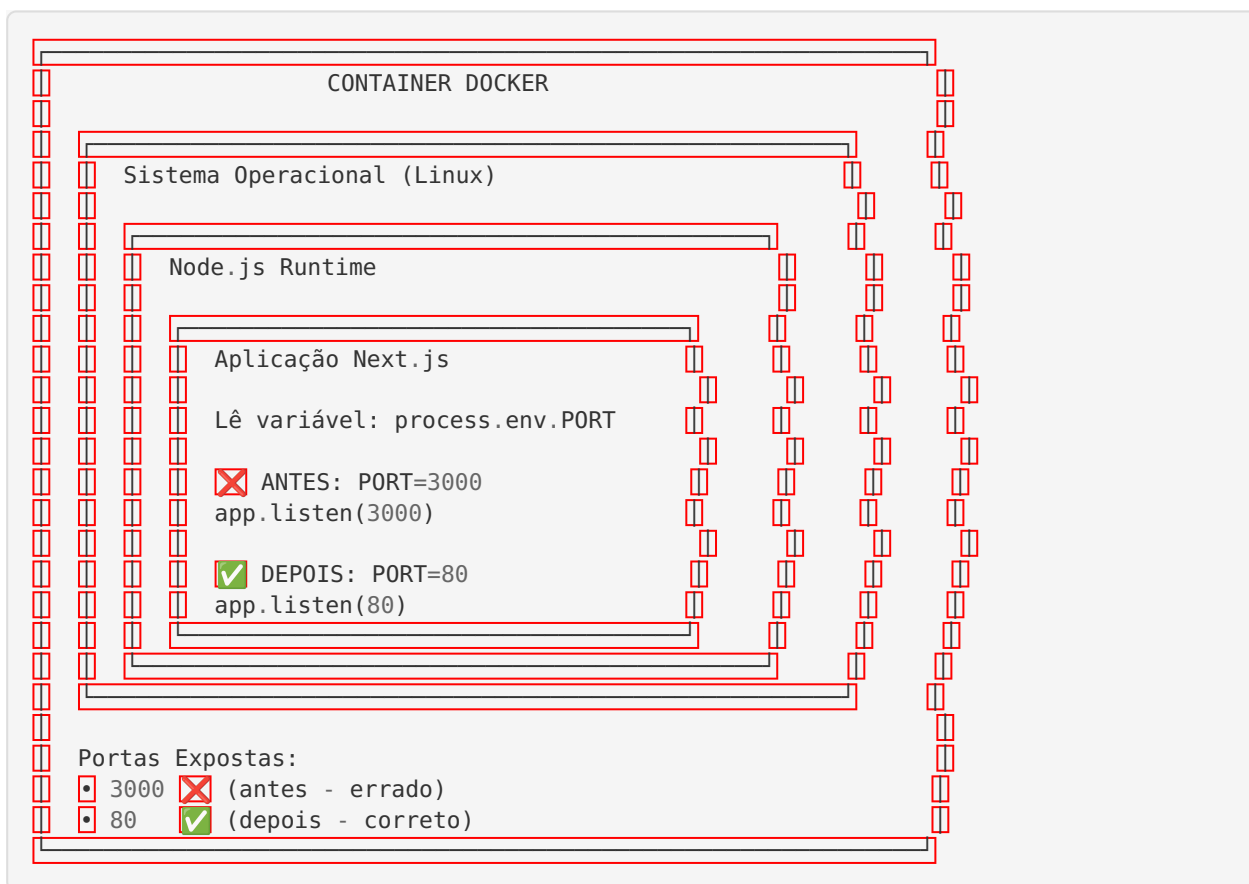
DEPOIS (Funcionando)

1. Usuário

2. DNS resolve para Easypanel

3. HTTPS  Proxy Easypanel

4. Proxy conecta: http://agenciaia_ag:80

5. **Container** porta :80   **Next.js** escutando aqui 

6. Requisição processada

7. Resposta enviada ao usuário

8. Health Check SUCESSO 

9. **Container** permanece rodando

10. Tudo funcionando normalmente  


Anatomia do Container



Estrutura do Container








Análise Detalhada do Problema

Timeline do Problema

```
T = 0s
├─ Container inicia
├─ Next.js lê PORT=3000
├─ Aplicação escuta em localhost:3000
└─  Aplicação iniciou com sucesso


T = 5s
├─ Easypanel faz primeiro health check
├─ Tenta conectar em agenciaia_ag:80
├─  Conexão recusada (nada na porta 80)
└─  Health check falha (1/3)

T = 35s
├─ Easypanel faz segundo health check
├─ Tenta conectar em agenciaia_ag:80
├─  Conexão recusada
└─  Health check falha (2/3)

T = 65s
├─ Easypanel faz terceiro health check
├─ Tenta conectar em agenciaia_ag:80
├─  Conexão recusada
├─  Health check falha (3/3)
└─  Container marcado como UNHEALTHY

T = 70s
├─ Easypanel decide reiniciar o container
├─ Envia SIGTERM para o processo
├─ Next.js recebe o sinal
├─ "SIGTERM received"
├─ "Gracefully shutting down. Please wait..."
└─ "Shutdown complete"


T = 75s
├─ Container para completamente
└─ Easypanel agenda restart


T = 80s
└─ VOLTA PARA T=0 (ciclo se repete) 
```

Comparação Visual das Soluções





Solução Principal: Mudar PORT para 80

CONFIGURAÇÃO

Variável de Ambiente:
PORT=80 


Proxy Reverso:
→ http://agenciaia_ag:80 


RESULTADO

 Porta 80  Porta 80 = MATCH!
 Configuração mais simples
 Apenas 1 mudança necessária





Alternativa 1: Manter PORT 3000

CONFIGURAÇÃO

Variável de Ambiente:
(Removida ou PORT=3000) 

Proxy Reverso:
→ http://agenciaia_ag:3000  (ALTERADO)

RESULTADO




 Porta 3000  Porta 3000 = MATCH!
 Requer alterar TODOS os domínios
 Mais passos necessários






Impacto da Correção

Métricas Esperadas

ANTES DA CORREÇÃO:

Uptime:  30%
Disponibilidade: Intermitente
Reinicializações: 10-20 por hora
SIGTERM:  Frequente
Usuários afetados: 100%
Status:  CRÍTICO

DEPOIS DA CORREÇÃO:

Uptime:  100%
Disponibilidade: Constante
Reinicializações: 0 (apenas deploys)
SIGTERM:  Nenhum
Usuários afetados: 0%
Status:  SAUDÁVEL

Teste de Validação

Como Testar se Está Funcionando

TESTE 1: Verificar Logs

Procure **por**:

✓ "- Local: http://localhost:80"

✓ "✓ Ready in XXXXms"

NÃO deve **ter**:

✗ "SIGTERM received"

✗ "Gracefully shutting down"

TESTE 2: Acessar Site

URL: <https://producoes.agmusic.cloud/>

Resultado **esperado**:

✓ Carrega em < 3 segundos

✓ Status 200 OK

✓ Conteúdo exibido corretamente

✓ Sem mensagens de erro

TESTE 3: Monitorar por 30 Minutos

Verificações:

✓ Container permanece "Running"

✓ Nenhum reinício inesperado

✓ Métricas estáveis (CPU, RAM)

✓ Site acessível durante todo o período

Lições Aprendidas

Por Que Isso Aconteceu?

CAUSA 1: Configuração Manual

- Variável PORT foi configurada manualmente
- Valor não correspondia ao proxy reverso
- Falta de documentação sobre requisitos de porta

CAUSA 2: Health Check Padrão

- Easypanel usa health check automático
- Verifica a porta configurada no proxy (80)
- Não foi ajustado para a porta real da aplicação (3000)

CAUSA 3: Comportamento do Next.js

- Next.js respeita a variável PORT do ambiente
- Se PORT=3000, escuta APENAS na porta 3000
- Não escuta em múltiplas portas automaticamente

Como Prevenir no Futuro?

✓ PREVENÇÃO 1: Documentação

- Adicionar README com configurações necessárias
- Especificar requisito de PORT=80 para Easypanel
- Incluir screenshots das configurações corretas

✓ PREVENÇÃO 2: Validação

- Adicionar health check endpoint na aplicação
- Logar a porta em que a aplicação está escutando
- Adicionar testes de integração

✓ PREVENÇÃO 3: Automação

- Usar Docker Compose com configuração correta
- Criar script de validação pré-deployment
- Configurar alertas para reinicializações frequentes

✓ PREVENÇÃO 4: Monitoramento

- Configurar alertas para SIGTERM
- Monitorar **uptime** da aplicação
- Dashboard de health checks

Resumo Visual Final

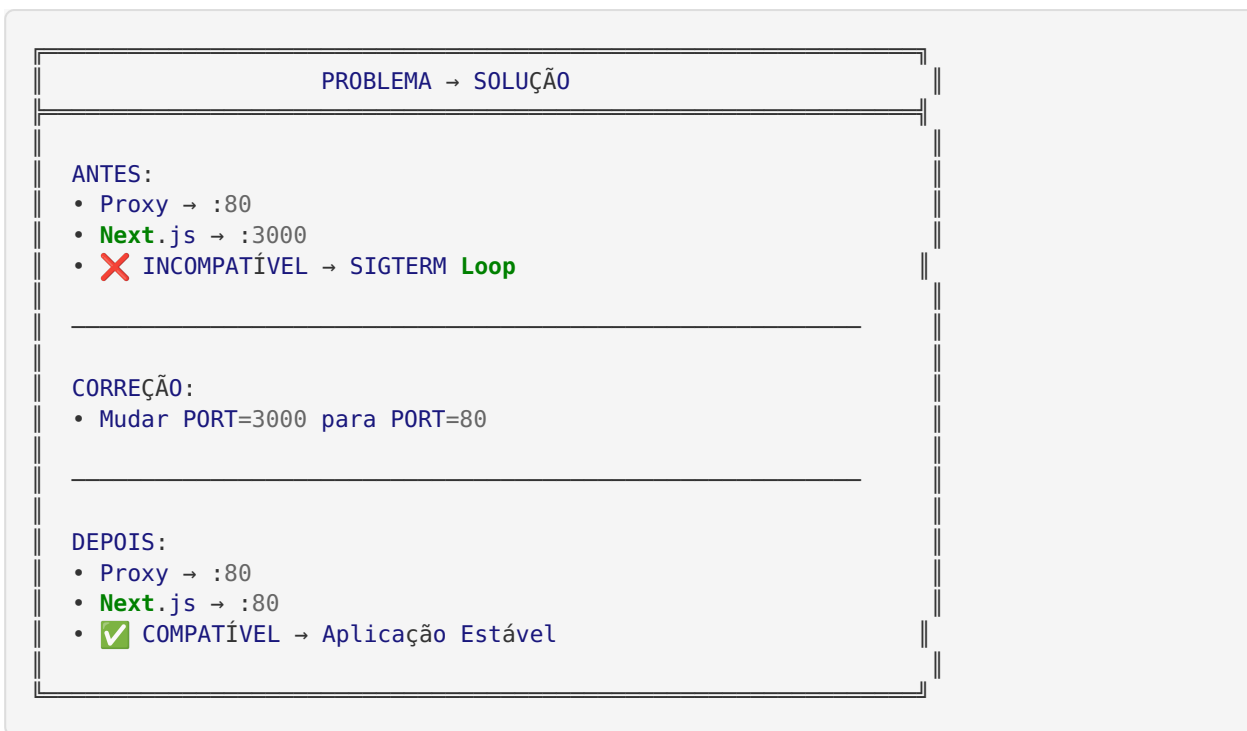


Diagrama criado em: 27 de outubro de 2025

Versão: 1.0

Compatível com: Easypanel, Next.js 15.0.3, Docker