

Documentação de Manutenção de Servidor para Aplicação Next.js com PM2

1. Introdução

Este documento descreve os procedimentos e comandos necessários para a manutenção de uma aplicação Next.js gerenciada pelo PM2 em um servidor Linux. O objetivo é fornecer um guia claro para administradores de sistema ou desenvolvedores responsáveis por manter a aplicação em execução, atualizada e estável no ambiente da Justiça Federal.

Esta documentação foca exclusivamente nos aspectos de manutenção do servidor e da aplicação via PM2, não abrangendo a estrutura do código-fonte da aplicação Next.js.

2. Pré-requisitos

Antes de iniciar qualquer procedimento de manutenção, certifique-se de que possui:

- Acesso SSH ao servidor Linux onde a aplicação está hospedada.
- Permissões de superusuário (sudo) para executar comandos de gerenciamento do PM2 e do sistema, se necessário.
- Conhecimento básico de comandos Linux.
- O Node.js e o NPM instalados no servidor.
- O PM2 instalado globalmente (sudo npm install -g pm2).

3. Visão Geral do PM2

PM2 (Process Manager 2) é um gerenciador de processos de produção para aplicações Node.js. Ele permite manter as aplicações vivas "para sempre", recarregá-las sem tempo de inatividade e facilitar tarefas comuns de administração do sistema.

No contexto desta aplicação Next.js, o PM2 foi configurado para:

- Iniciar a aplicação Next.js usando o comando `npm start` (que por sua vez executa `next start -H SEU_IP_AQUI`).
- Nomear o processo da aplicação como `next-app`.
- Garantir que a aplicação reinicie automaticamente em caso de falhas.
- Garantir que a aplicação seja iniciada automaticamente após o boot do servidor.

4. Comandos Essenciais do PM2 para Manutenção

Os seguintes comandos PM2 são cruciais para a manutenção diária e a solução de

problemas da aplicação next-app.

4.1. Verificar o Status da Aplicação

Para verificar o status de todas as aplicações gerenciadas pelo PM2, incluindo a next-app:

pm2 status

Ou, para listar de forma mais concisa:

pm2 list

Este comando exibirá informações como o nome do processo (next-app), id, modo (fork ou cluster), status (online, stopping, stopped, launching, errored), uso de CPU e memória.

4.2. Reiniciar a Aplicação

Se você precisar reiniciar a aplicação (por exemplo, após uma atualização de código ou para aplicar novas configurações):

pm2 restart next-app

O PM2 tentará reiniciar a aplicação com o mínimo de tempo de inatividade possível (reload).

4.3. Parar a Aplicação

Para parar a aplicação next-app:

pm2 stop next-app

Isso alterará o status da aplicação para stopped.

4.4. Iniciar a Aplicação (se estiver parada)

Se a aplicação foi parada manualmente e você precisa iniciá-la novamente:

pm2 start next-app

4.5. Visualizar Logs da Aplicação

Os logs são essenciais para diagnosticar problemas. Para visualizar os logs em tempo real da aplicação next-app (saída padrão e erros):

pm2 logs next-app

Para visualizar logs mais antigos ou com opções específicas (como --lines <numero_de_linhas>):

pm2 logs next-app --lines 100

Os logs do PM2 geralmente são armazenados em ~/.pm2/logs/.

4.6. Deletar a Aplicação do Gerenciamento do PM2

Atenção: Use este comando com cuidado, pois ele removerá a aplicação da lista de processos do PM2. A aplicação será parada e não será mais gerenciada pelo PM2.

pm2 delete next-app

Isso pode ser útil se você precisar reconfigurar completamente o processo. Para adicioná-la novamente, você precisaria usar o comando `pm2 start npm --name "next-app" -- start -- -H SEU_IP_AQUI` novamente.

4.7. Gerenciamento do PM2 na Inicialização do Sistema

Os comandos abaixo foram utilizados na configuração inicial para garantir que o PM2 inicie com o sistema e que os processos salvos sejam restaurados.

- `pm2 startup`: Gera um comando para configurar o PM2 para iniciar no boot do sistema. Você precisará executar o comando que ele fornecer.
- `pm2 save`: Salva a lista de processos atual do PM2 para que seja restaurada no boot. **Execute este comando sempre que fizer alterações significativas na configuração dos processos (como adicionar um novo processo ou alterar configurações de um existente).**

Para verificar se o serviço de startup está ativo:

O comando exato pode variar dependendo da distribuição Linux (systemd, upstart, etc.)

Para systemd (comum em versões recentes do Ubuntu, CentOS, etc.):

`systemctl status pm2-$(whoami)`

5. Procedimento de Atualização da Aplicação

Quando uma nova versão da aplicação Next.js estiver pronta para ser implantada (após o código ser atualizado no repositório, por exemplo), siga estes passos:

1. Acesse o Servidor:
Conecte-se ao servidor via SSH.
2. Navegue até o Diretório da Aplicação:
Use o comando `cd` para ir até o diretório raiz do projeto Next.js.
Exemplo: `cd /caminho/para/sua/aplicacao/Estrutura-JF`
3. Obtenha o Código Mais Recente:
Se você estiver usando um sistema de controle de versão como o Git (o que é altamente recomendado), puxe as últimas alterações do repositório.
`git pull origin main` # ou o nome da sua branch principal
4. Instale/Atualize as Dependências:
É importante garantir que todas as dependências do Node.js estejam atualizadas conforme o `package.json` e `package-lock.json`.

npm install

Este comando instalará novas dependências ou atualizará as existentes.

5. Construa (Build) a Aplicação Next.js:
O Next.js requer um passo de "build" para otimizar a aplicação para produção.
`npm run build`

Este comando geralmente cria a pasta `.next` com os arquivos de produção.

6. Reinicie a Aplicação com PM2:
Após o build ser concluído com sucesso, reinicie a aplicação usando o PM2 para que as novas alterações entrem em vigor.
`pm2 restart next-app`

O PM2 irá carregar a nova build da aplicação.

7. Verifique o Status e os Logs:
Após reiniciar, verifique se a aplicação está online e funcionando corretamente.

```
pm2 status  
pm2 logs next-app
```

Acesse a aplicação pelo navegador para confirmar que a atualização foi bem-sucedida.

6. Solução de Problemas (Troubleshooting)

6.1. Aplicação Não Inicia ou Entra em Estado de Erro (errored)

- **Verifique os Logs:** Este é o primeiro passo. Os logs do PM2 (`pm2 logs next-app`) geralmente contêm mensagens de erro detalhadas que indicam o problema (ex: erro de sintaxe no código, porta já em uso, falha ao conectar a um banco de dados, etc.).
- **Variáveis de Ambiente:** Certifique-se de que todas as variáveis de ambiente necessárias para a aplicação (`.env.production` ou configuradas diretamente no PM2) estão corretas.
- **Build Falhou:** Verifique se o comando `npm run build` foi executado sem erros. Se houver erros no build, a aplicação não iniciará corretamente.
- **Porta em Uso:** Se a aplicação tenta iniciar em uma porta que já está sendo usada por outro processo (o erro comum é `EADDRINUSE`), você precisará parar o outro processo ou configurar sua aplicação Next.js para usar uma porta diferente. O PM2 foi iniciado com `-H 172.22.3.31`, o que implica que o Next.js tentará usar uma porta padrão (geralmente 3000) nesse host. Verifique se essa porta está livre.
- **Dependências Ausentes ou Corrompidas:** Tente remover a pasta `node_modules` e o arquivo `package-lock.json` e reinstalar as dependências:

```
rm -rf node_modules package-lock.json  
npm install  
npm run build  
pm2 restart next-app
```

6.2. Alto Uso de CPU ou Memória

- **Logs:** Verifique os logs por atividade excessiva ou loops infinitos.
- **Otimização da Aplicação:** Pode ser necessário otimizar o código da aplicação Next.js para reduzir o consumo de recursos.
- **Vazamentos de Memória:** Use ferramentas de profiling do Node.js se suspeitar de vazamentos de memória.
- **Aumentar Recursos do Servidor:** Se a aplicação legitimamente requer mais recursos, pode ser necessário um upgrade do servidor.

6.3. PM2 Não Inicia com o Sistema

- **Verifique a Configuração de Startup:** Execute `pm2 startup` novamente e siga as instruções.
- **Salve os Processos:** Certifique-se de que você executou `pm2 save` após configurar os processos que deseja restaurar no boot.

7. Considerações Importantes

- **Monitoramento de Recursos do Servidor:** Monitore regularmente o uso de CPU, memória RAM e espaço em disco no servidor para evitar problemas de desempenho ou indisponibilidade.
- **Atualizações de Segurança:** Mantenha o sistema operacional do servidor, o Node.js, o NPM, o PM2 e todas as dependências da aplicação atualizados com as últimas correções de segurança.
- **Backups:** Embora o PM2 gerencie o processo da aplicação, ele não gerencia os dados da aplicação ou o código-fonte. Implemente uma estratégia de backup para o código-fonte e quaisquer dados persistentes que sua aplicação possa utilizar.
- **Documentação Específica da Aplicação:** Mantenha uma documentação interna sobre configurações específicas da aplicação Next.js (variáveis de ambiente, integrações, etc.) que possam ser relevantes durante a manutenção.

Este guia deve cobrir os aspectos essenciais da manutenção do servidor para sua aplicação Next.js gerenciada pelo PM2. Lembre-se de adaptar os comandos e caminhos conforme a configuração específica do seu ambiente.