Nginx - Formas de Balanceamento

Fonte: https://nginx.org/en/docs/http/load_balancing.html

Usando nginx como balanceador de carga HTTP

Introdução

O balanceamento de carga entre várias instâncias de aplicativos é uma técnica comumente usada para otimizar a utilização de recursos, maximizar a taxa de transferência, reduzir a latência e garantir configurações tolerantes a falhas.

É possível usar o nginx como um balanceador de carga HTTP muito eficiente para distribuir tráfego para vários servidores de aplicativos e melhorar o desempenho, a escalabilidade e a confiabilidade de aplicativos da web com o nginx.

Métodos de balanceamento de carga

Os seguintes mecanismos (ou métodos) de balanceamento de carga são suportados no nginx:

- round-robin — as solicitações aos servidores de aplicativos são distribuídas em formato round-robin,
- menos conectado — a próxima solicitação é atribuída ao servidor com o menor número de conexões ativas,
- ip-hash — uma função hash é usada para determinar qual servidor deve ser selecionado para a próxima solicitação (com base no endereço IP do cliente).

Configuração de balanceamento de carga padrão

A configuração mais simples para balanceamento de carga com nginx pode ser parecida com a seguinte:

> ```
> http {
>     upstream myapp1 {
>         servidor srv1.example.com;
>         servidor srv2.example.com;
>         servidor srv3.example.com;
>     }
>
>     servidor {
>         ouvir 80;
>
>         localização / {
>             proxy_pass http://myapp1;
>         }
>     }
> }
> ```

No exemplo acima, há 3 instâncias do mesmo aplicativo em execução em srv1-srv3. Quando o método de balanceamento de carga não é configurado especificamente, o padrão é round-robin.

Todas as solicitações são encaminhadas por proxy para o grupo de servidores myapp1, e o nginx aplica o balanceamento de carga HTTP para distribuir as solicitações.

A implementação de proxy reverso no nginx inclui balanceamento de carga para HTTP, HTTPS, FastCGI, uwsgi, SCGI, memcached e gRPC.

Para configurar o balanceamento de carga para HTTPS em vez de HTTP, basta usar “https” como protocolo.

Ao configurar o balanceamento de carga para FastCGI, uwsgi, SCGI, memcached ou gRPC, use as diretivas fastcgi_pass , uwsgi_pass , scgi_pass , memcached_pass e grpc_pass respectivamente.

Balanceamento de carga menos conectado

Outra disciplina de balanceamento de carga é a menos conectada. Essa abordagem permite controlar a carga em instâncias de aplicativos de forma mais justa em situações em que algumas solicitações demoram mais para serem concluídas.

Com o balanceamento de carga menos conectado, o nginx tentará não sobrecarregar um servidor de aplicativos ocupado com solicitações excessivas, distribuindo as novas solicitações para um servidor menos ocupado.

O balanceamento de carga menos conectado no nginx é ativado quando a diretiva least_conn é usada como parte da configuração do grupo de servidores:

> ```
> upstream myapp1 {
>     conexão_menor;
>     servidor srv1.example.com;
>     servidor srv2.example.com;
>     servidor srv3.example.com;
> }
> ```

Persistência de sessão

Observe que, com o balanceamento de carga round-robin ou de menor conexão, cada solicitação de cliente subsequente pode ser potencialmente distribuída para um servidor diferente. Não há garantia de que o mesmo cliente será sempre direcionado para o mesmo servidor.

Se houver necessidade de vincular um cliente a um servidor de aplicativo específico — em outras palavras, tornar a sessão do cliente "fixa" ou "persistente" em termos de sempre tentar selecionar um servidor específico — o mecanismo de balanceamento de carga ip-hash pode ser usado.

Com o ip-hash, o endereço IP do cliente é usado como uma chave de hash para determinar qual servidor em um grupo de servidores deve ser selecionado para as solicitações do cliente. Este método garante que as solicitações do mesmo cliente sejam sempre direcionadas ao mesmo servidor, exceto quando este estiver indisponível.

Para configurar o balanceamento de carga ip-hash, basta adicionar a diretiva ip_hash à configuração do grupo do servidor (upstream):

> ```
> upstream myapp1 {
>     ip_hash;
>     servidor srv1.example.com;
>     servidor srv2.example.com;
>     servidor srv3.example.com;
> }
> ```

Balanceamento de carga ponderado

Também é possível influenciar ainda mais os algoritmos de balanceamento de carga do nginx usando pesos de servidor.

Nos exemplos acima, os pesos do servidor não são configurados, o que significa que todos os servidores especificados são tratados como igualmente qualificados para um método específico de balanceamento de carga.

Com o round-robin em particular, isso também significa uma distribuição mais ou menos igual de solicitações entre os servidores — desde que haja solicitações suficientes e quando as solicitações sejam processadas de maneira uniforme e concluídas com rapidez suficiente.

Quando o parâmetro de peso é especificado para um servidor, o peso é contabilizado como parte da decisão de balanceamento de carga.

> ```
>     upstream myapp1 {
>         servidor srv1.example.com peso=3;
>         servidor srv2.example.com;
>         servidor srv3.example.com;
>     }
> ```

Com essa configuração, a cada 5 novas solicitações serão distribuídas entre as instâncias do aplicativo da seguinte maneira: 3 solicitações serão direcionadas para srv1, uma solicitação irá para srv2 e outra para srv3.

Da mesma forma, é possível usar pesos com balanceamento de carga menos conectado e ip-hash nas versões recentes do nginx.

Verificações de saúde

A implementação de proxy reverso no nginx inclui verificações de integridade do servidor em banda (ou passivas). Se a resposta de um servidor específico falhar com um erro, o nginx marcará esse servidor como falho e tentará evitar a seleção deste servidor para solicitações de entrada subsequentes por um tempo.

A diretiva max_fails define o número de tentativas consecutivas malsucedidas de comunicação com o servidor que devem ocorrer durante o fail_timeout . Por padrão, max_fails é definido como 1. Quando definido como 0, as verificações de integridade são desativadas para este servidor. O parâmetro fail_timeout também define por quanto tempo o servidor será marcado como com falha. Após o intervalo fail_timeout após a falha do servidor, o nginx começará a sondar o servidor normalmente com as solicitações do cliente ativo. Se as sondagens forem bem-sucedidas, o servidor será marcado como ativo.

Leitura adicional

Além disso, existem mais diretivas e parâmetros que controlam o balanceamento de carga do servidor no nginx, como proxy_next_upstream , backup , down e keepalive . Para mais informações, consulte nossa documentação de referência .

Por último, mas não menos importante, o balanceamento de carga de aplicativos, as verificações de integridade dos aplicativos, o monitoramento de atividades e a reconfiguração dinâmica de grupos de servidores estão disponíveis como parte de nossas assinaturas pagas do NGINX Plus.

Revisão #2
Criado 25 setembro 2025 20:09:34 por EduStore
Atualizado 25 setembro 2025 20:09:44 por EduStore