

Aplicando Segurança com NGINX

Erick dos Santos Alves

*STIC / UFRJ
2017*

Roteiro

- Breve história
- Não apenas um mero servidor Web
- Garantindo os 3 pilares
- “Proxeando” dificultamos mais
 - Balanceamento de Carga
 - Disponibilidade melhorada com Cache
- Considerações Finais

Breve História

- O Nginx (*engine x*) é um serviço de **HTTP** e **proxy reverso**.
- Desenvolvido em 2005 por Igor Sysoev
- É bastante versátil podendo ser utilizado como uma espécie de “**middleware**” entre o cliente e os serviços.

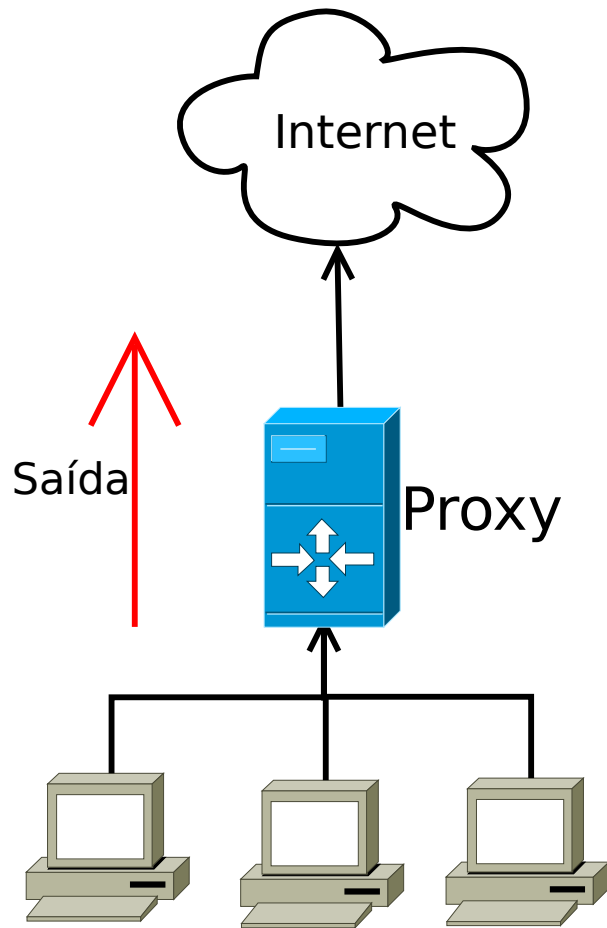
The Nginx logo is displayed in a bold, green, sans-serif font. The letters are slightly irregular, with some gaps and a modern, tech-oriented feel.

Não apenas um mero servidor Web

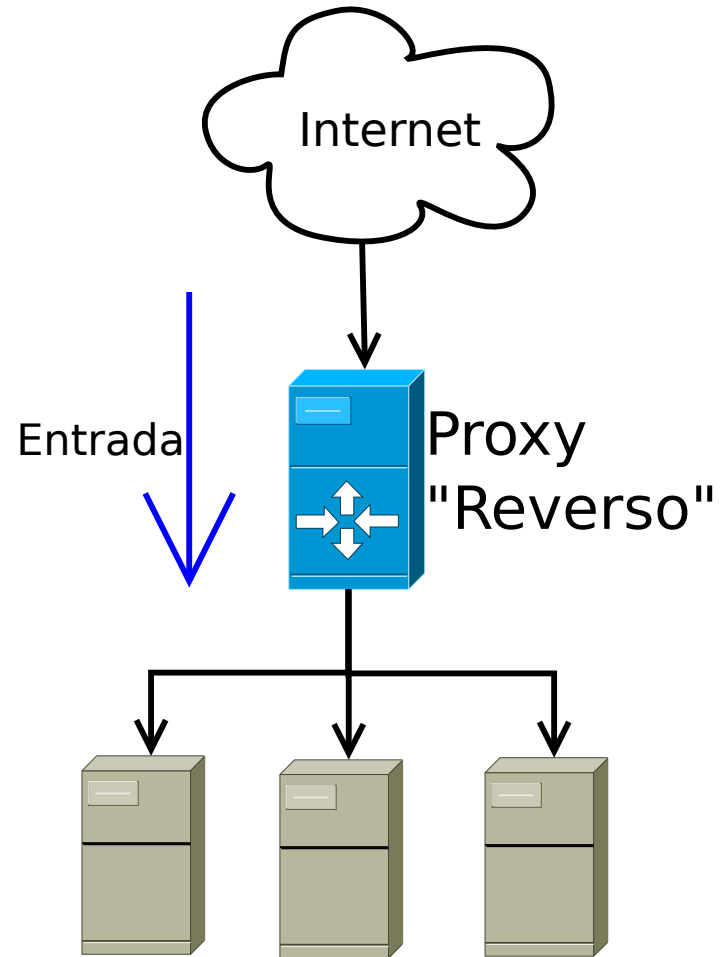
- Além de ser amplamente utilizado como servidor de HTTP, o Nginx pode estar presente em estruturas de servidores onde não necessariamente ele terá essa função.
- Ao utilizar o Nginx como **proxy reverso**, podemos controlar o fluxo de acesso a servidores e serviços de uma infraestrutura de T.I.

Proxy X Proxy Reverso

Proxy "tradicional"



Proxy reverso



Garantindo os 3 pilares

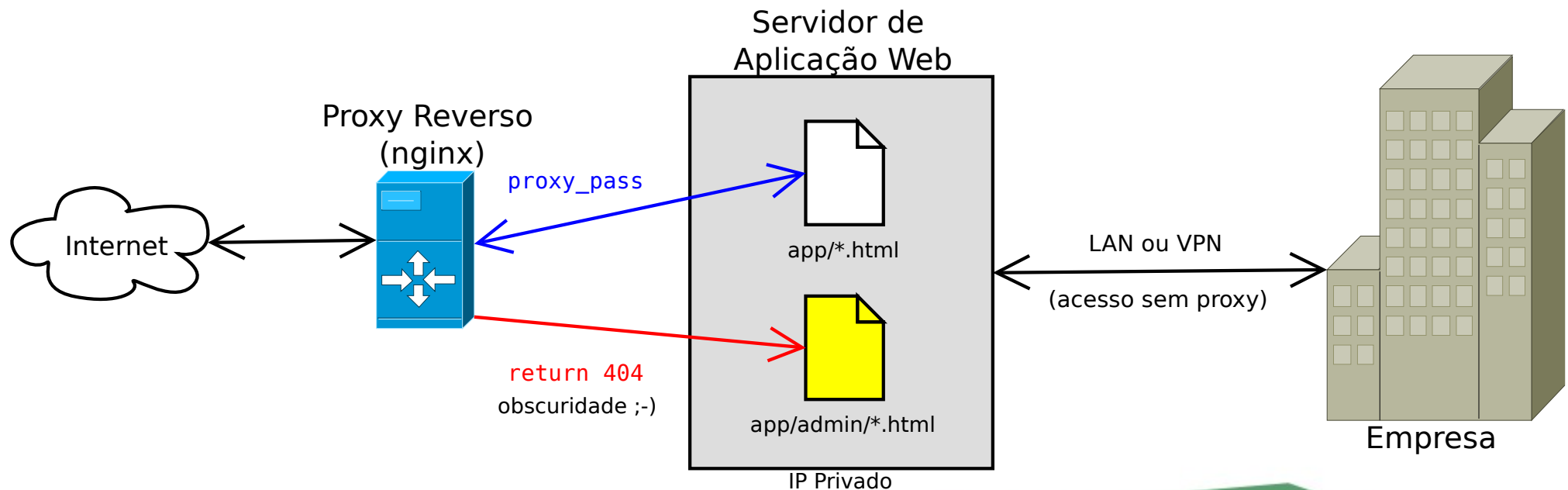
- **Disponibilidade:** Melhorada com balanceamento de carga e cache.
- **Integridade:** Com o controle de cache os dados originais ficam intocados.
- **Confidencialidade:** Com o serviço de proxy reverso é possível controlar o que será exibido através de regras.



“Proxeando” dificultamos mais (1)

- **Cenário 1: “Limitação de acesso”**

- Acesso completo a aplicação somente pela rede privada por **acesso direto**.
- Conexões vindas da Internet passam pelo proxy e não tem acesso a parte administrativa da aplicação.
- Pode ser utilizado **caching** para poupar esforço do servidor Web das conexões vindas da Internet.



“Proxeando” dificultamos mais (2)

- Cenário 1: “Limitação de acesso”

```
server {
    listen 80;
    server_name app.ufrj.br;

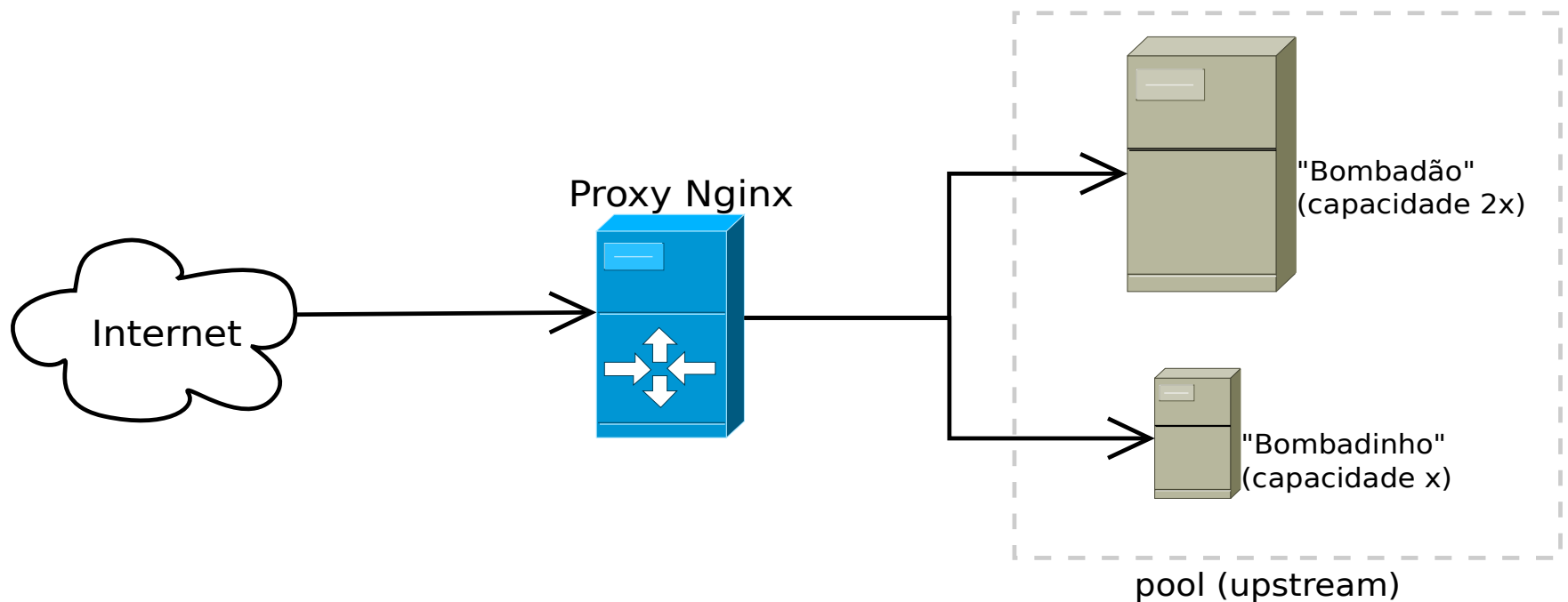
    location / {
        proxy_pass original.app.com.br;

        # Substitui a URL para não quebrar links
        sub_filter_once off;
        sub_filter "original.app.ufrj.br" $server_name;
    }

    location /admin {
        return 404; # ← Retorna página não encontrada
    }
}
```


“Proxeando” dificultamos mais (3)

- **Cenário 2: “Pool de servidores” (cluster / load balancing)**
 - Organiza um conjunto de servidores para responderem em uma única URL
 - Também é possível regular a carga de processamento de acordo com a capacidade de cada servidor do cluster



“Proxeando” dificultamos mais (4)

- Cenário 2: “Pool de servidores” (cluster / load balancing)

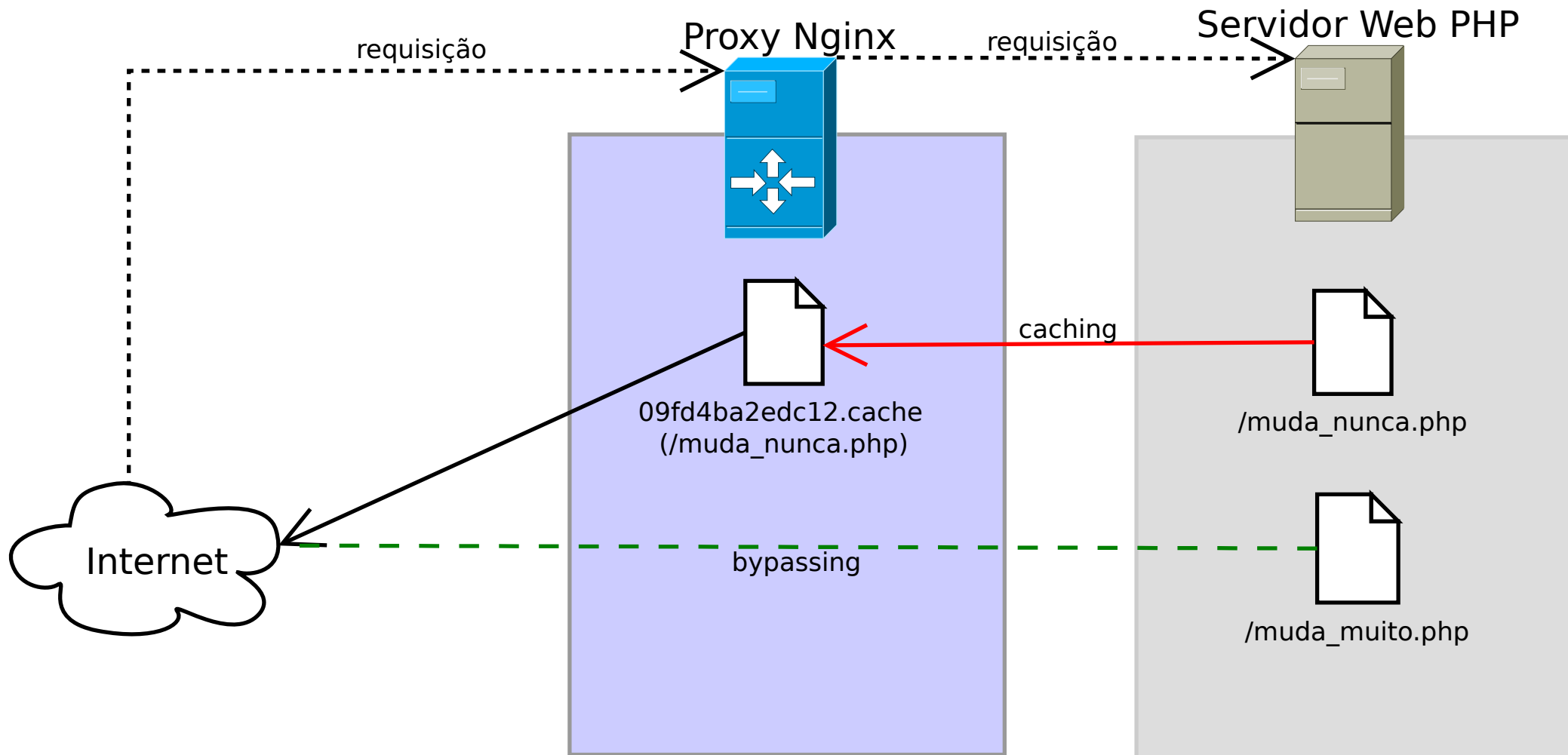
```
upstream pool.ufrj.br {  
    server bombadinho.ufrj.br weight=1;  
    server bombadao.ufrj.br weight=2;  
}  
  
server {  
    listen 80;  
    server_name app.ufrj.br;  
  
    location / {  
        proxy_pass http://pool.ufrj.br;  
    }  
}
```

Disponibilidade melhorada com Cache (1)

- Com o recurso de cache é possível guardar um recurso “pós-processado” para que não seja necessária sua reconstrução frequente, agilizando o processamento.
- Economiza recursos computacionais fazendo com que alguns recursos sejam acessados menos vezes (ex.: bancos de dados)
- Ideal para elementos estáticos ou que sofrem poucas modificações com o tempo.
- Pode ser aplicado parcialmente em uma aplicação através da diretiva **location**.

Disponibilidade melhorada com Cache (2)

- Cenário 3: Arquivos “muda_nunca” e “muda_muito”



Disponibilidade melhorada com Cache (3)

```
proxy_cache_path /var/cache/ keys_zone=cache_app:10m levels=1:2;

server {
    listen 80;
    server_name app.ufrj.br;

    location /muda_nunca.php {
        proxy_pass http://original.app.ufrj.br/muda_nunca.php;
        proxy_cache cache_app;

        proxy_cache_use_stale error timeout updating http_403
http_404 http_500 http_502 http_503 http_504;

        proxy_cache_valid 200 301 302 307 308 1h;
        proxy_cache_valid 404 2m;
        expires max;
    }
    location /muda_muito.php {
        proxy_pass http://original.app.ufrj.br/muda_muito.php;
    }
}
```

Considerações Finais

- O Nginx tem um papel muito importante para garantir a segurança de aplicações Web, não somente pelas suas utilidades como **proxy reverso** e **caching** como foi visto nesta apresentação.
- Com o devido conhecimento de sua aplicação, é possível definir regras específicas para cada **diretório**, **tipo de arquivo**, **expressões regulares** e até mesmo **headers** e **user-agents**, fortalecendo ainda mais a proteção de sua aplicação.

ERIGADO !

Erick dos Santos Alves

erick_sa@ufrj.br

Referências Bibliográficas

- <https://pt.wikipedia.org/wiki/Nginx>
- SARKAR, Dipankar; Nginx 1 **Web Server Implementation Cookbook**. Packt Publishing