



SÃO
PAULO
TECH
SCHOOL

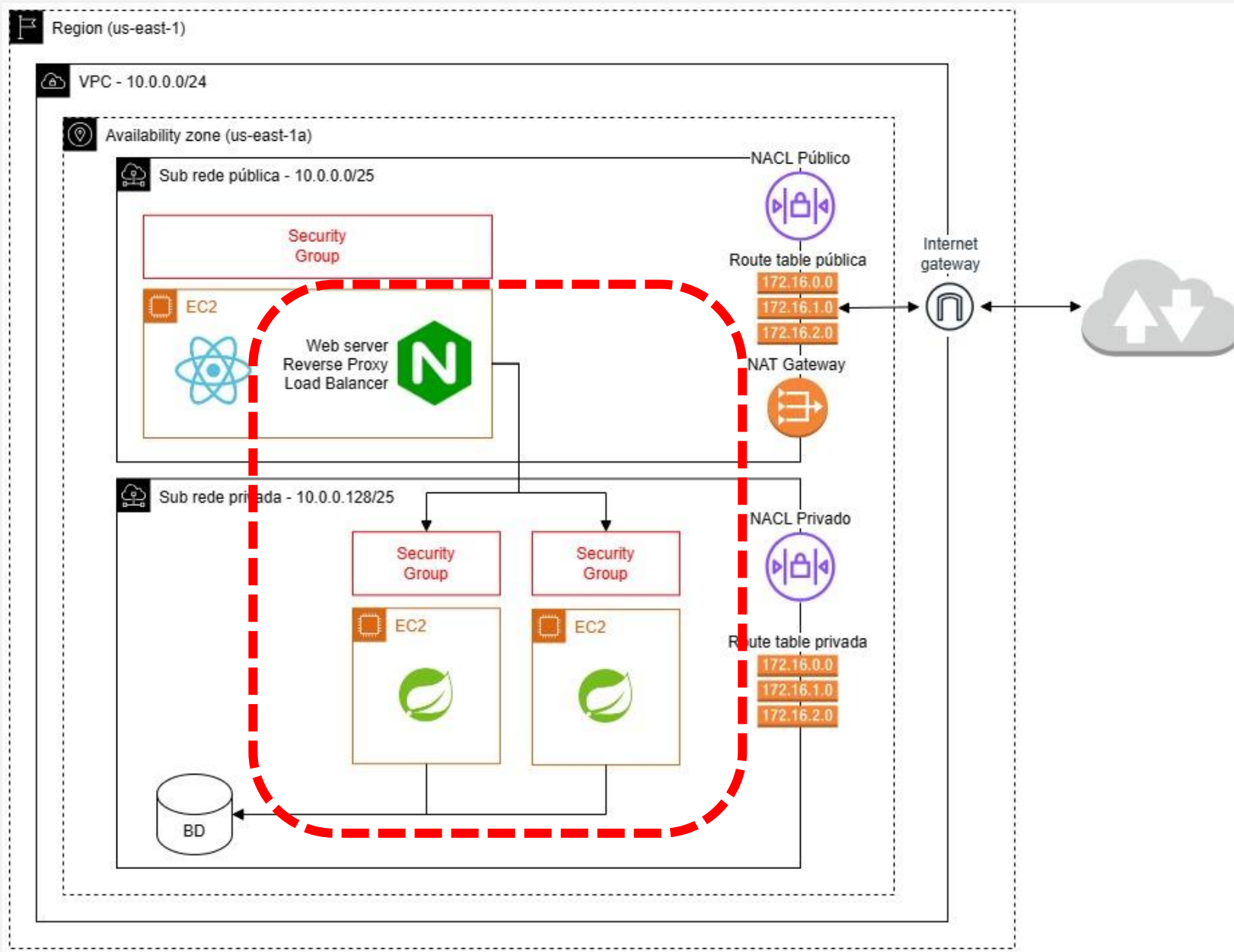
Computação e sistemas distribuídos

NGINX – Arquitetura e configuração – pt.2

Eduardo Verri

eduardo.verri@sptech.school

NGINX – Arquitetura e configuração pt.2



Principais funcionalidades – exemplos de configuração

Servidor Web

Como servidor web, o NGINX é capaz de servir conteúdo estático diretamente do disco, sendo extremamente eficiente nesse processo. Ele lida com arquivos HTML, imagens, arquivos de estilo (CSS) e scripts do lado do cliente (JavaScript) com rapidez e eficiência, tornando-o uma escolha popular para sites que necessitam de alta performance.

- **Alta Performance e Baixo Consumo de Recursos:** Otimizado para operações de I/O não bloqueantes, reduzindo significativamente o consumo de recursos do sistema.
- **Servindo Conteúdo Estático:** Capacidade de servir milhares de arquivos estáticos simultaneamente sem degradar o desempenho.

Servidor Web

Para configurar o NGINX como um servidor web para servir conteúdo estático, você pode usar uma configuração simples:

```
server {  
    listen 80;  
    server_name exemplo.com;  
  
    location / {  
        root /var/www/exemplo.com/html;  
        index index.html index.htm;  
    }  
}
```

Balanceamento de Carga

O NGINX pode funcionar como um balanceador de carga, distribuindo as solicitações de entrada entre vários servidores backend, o que aumenta a capacidade de carga e a confiabilidade do sistema.

- **Métodos de Balanceamento:** Suporta vários algoritmos, incluindo **round-robin**, **least connections** e **hash de IP**, permitindo uma distribuição equitativa das cargas.
- **Saúde dos Servidores:** Monitora a saúde dos servidores backend para garantir que as solicitações sejam redirecionadas apenas para os servidores disponíveis.

Balanceamento de Carga

Para configurar o NGINX como um balanceador de carga distribuindo solicitações entre vários servidores backend, você pode usar a seguinte configuração:

```
http {  
    upstream backend {  
        server backend1.exemplo.com;  
        server backend2.exemplo.com;  
    }  
  
    server {  
        listen 80;  
  
        location / {  
            proxy_pass http://backend;  
        }  
    }  
}
```


Balanceamento de Carga

Métodos de Balanceamento de Carga Disponíveis

O NGINX suporta diversos métodos de balanceamento de carga, incluindo:

- **Round-robin:** Distribui as requisições de maneira sequencial entre os servidores(**padrão**).
- **Least connections:** Direciona novas requisições para o servidor com o menor número de conexões ativas.
- **IP hash:** As requisições de um mesmo cliente são sempre enviadas ao mesmo servidor, baseando-se em um hash do endereço IP do cliente.

Balanceamento de Carga

```
upstream meuapp_backend {  
    least_conn;  
    server 192.168.1.1:3000;  
    server 192.168.1.2:3000;  
}
```

```
upstream meuapp_backend {  
    ip_hash;  
    server 192.168.1.1:3000;  
    server 192.168.1.2:3001;  
}
```

Proxy Reverso

Atuando como um proxy reverso, o NGINX aceita solicitações HTTP de clientes, as encaminha para um servidor interno e retorna as respostas do servidor interno para os clientes. Isso pode aumentar a segurança, a escalabilidade e a performance de aplicações web.

- **Desacoplamento de Aplicações:** Permite a separação entre clientes e servidores de aplicação, facilitando a escalabilidade e a manutenção.
- **Criptografia e SSL/TLS:** Pode gerenciar a criptografia SSL/TLS, descarregando essa tarefa dos servidores de aplicação.

Proxy Reverso

Para configurar o NGINX para cachear conteúdo, adicionando regras no bloco server ou location, você pode fazer o seguinte:

```
server {  
    listen 80;  
    server_name app.exemplo.com;  
  
    location / {  
        proxy_pass http://servidor_aplicacao_interno;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

Cache de Conteúdo

O NGINX possui capacidades de cache poderosas, que permitem armazenar conteúdo dinâmico e estático. Isso reduz a carga nos servidores de aplicação e melhora o tempo de resposta para os usuários finais.

- **Configuração Flexível:** Oferece controle granular sobre o que é armazenado em cache, por quanto tempo e como o cache é validado.
- **Redução de Latência:** Serve respostas diretamente do cache, diminuindo significativamente a latência percebida pelos usuários.

Cache de Conteúdo

Para usar o NGINX como um proxy reverso, encaminhando solicitações para um servidor de aplicativos interno, a configuração seria semelhante à seguinte:

```
http {  
    proxy_cache_path /path/to/cache levels=1:2 keys_zone=my_cache:10m max_size=10g inactive=60m use_temp_path=off;  
  
    server {  
        location / {  
            proxy_cache my_cache;  
            proxy_pass http://backend;  
            proxy_set_header Host $host;  
            proxy_cache_valid 200 302 60m;  
            proxy_cache_valid 404 1m;  
        }  
    }  
}
```

Segurança (Forçar HTTPS usando SSL/TLS)

O NGINX inclui uma série de recursos destinados a aumentar a segurança das aplicações web, protegendo contra uma variedade de ataques e vulnerabilidades.

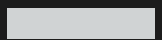
- **Limitação de Taxa:** Protege contra ataques de DDoS e força bruta limitando o número de solicitações que um usuário pode fazer em um determinado período.
- **Bloqueio de IP e Controle de Acesso:** Permite ou nega solicitações com base no endereço IP do cliente, oferecendo uma camada adicional de segurança.
- **Integração com Let's Encrypt:** Facilita a implementação de HTTPS, oferecendo uma maneira automatizada de obter e renovar certificados SSL/TLS gratuitos.

Segurança

[Forçar HTTPS usando SSL/TLS]

Para redirecionar todo o tráfego HTTP para HTTPS e configurar o SSL/TLS, você precisará de certificados SSL e pode usar a seguinte configuração:

```
server {  
    listen 80;  
    server_name exemplo.com www.exemplo.com;  
    return 301 https://$server_name$request_uri;  
}  
  
server {  
    listen 443 ssl;  
    server_name exemplo.com www.exemplo.com;  
  
    ssl_certificate /path/to/ssl/certificate.pem;  
    ssl_certificate_key /path/to/ssl/key.pem;  
  
    location / {  
        root /var/www/exemplo.com/html;  
        index index.html;  
    }  
}
```

NGINX Tips And Tricks



Gerenciando Nginx

- **Teste de Configuração:** Sempre use o comando **nginx -t** para testar suas configurações do NGINX antes de reiniciar o serviço. Este comando verifica se há erros sintáticos em seus arquivos de configuração, ajudando a evitar a interrupção do serviço devido a configurações inválidas.
- **Recarregar Configuração:** Para aplicar mudanças na configuração sem interromper as conexões ativas, você pode recarregar o NGINX com: **systemctl reload nginx**
- **Reiniciar o NGINX:** Se precisar reiniciar completamente o serviço NGINX (o que interromperá brevemente as conexões), use: **systemctl restart nginx;**
- **Verificar Status do NGINX:** Para verificar se o NGINX está rodando: **systemctl status nginx;**

Logs – como encontrar?

- **Localização Padrão dos Logs:** Os logs do NGINX geralmente estão localizados no diretório **/var/log/nginx/**, a menos que especificado de outra forma na configuração do NGINX. Dentro deste diretório, você encontrará **access.log** para os logs de acesso e **error.log** para os logs de erros.
- **Log de Acesso (access.log):** Mostra todas as requisições ao servidor NGINX. Para visualizar as últimas linhas do log de acesso, você pode usar: **tail -f /var/log/nginx/access.log;**
- **Log de Erro (error.log):** Registra mensagens de erro detalhadas, incluindo problemas no servidor, problemas de configuração, etc. Para verificar as últimas mensagens de erro, use: **tail -f /var/log/nginx/error.log;**



Combinações de comando

- **Análise de Logs:** Para uma análise mais detalhada dos logs, você pode utilizar ferramentas como **grep**, **awk**, ou **cut** para filtrar e analisar entradas específicas. Ferramentas de análise de log de terceiros ou sistemas de gerenciamento de log também podem oferecer insights mais profundos sobre o tráfego do seu site e possíveis problemas.

Agradeço
a sua atenção!



SÃO
PAULO
TECH
SCHOOL