CECyber | Center of Excellence in Cybersecurity

CYBERSECURITY EM MINUTOS

ROTEIRO DE CONFIGURAÇÃO

ALTA DISPONIBILIDADE COM HAPROXY EM UBUNTU 22.04

POR **ALMIR ALVES**

Leia o Qr Code
Confira os cursos da CECyber

SOBRE ESSE DOCUMENTO

Detalha o processo de configuração de um serviço de alta disponibilidade usando **HAProxy** como balanceador de carga para dois servidores de aplicação, com um terceiro servidor atuando como gerenciador do HAProxy.

Todos os servidores utilizam **Ubuntu 22.04** e estão virtualizados (por exemplo, no VMware Workstation Pro).

PREMISSAS

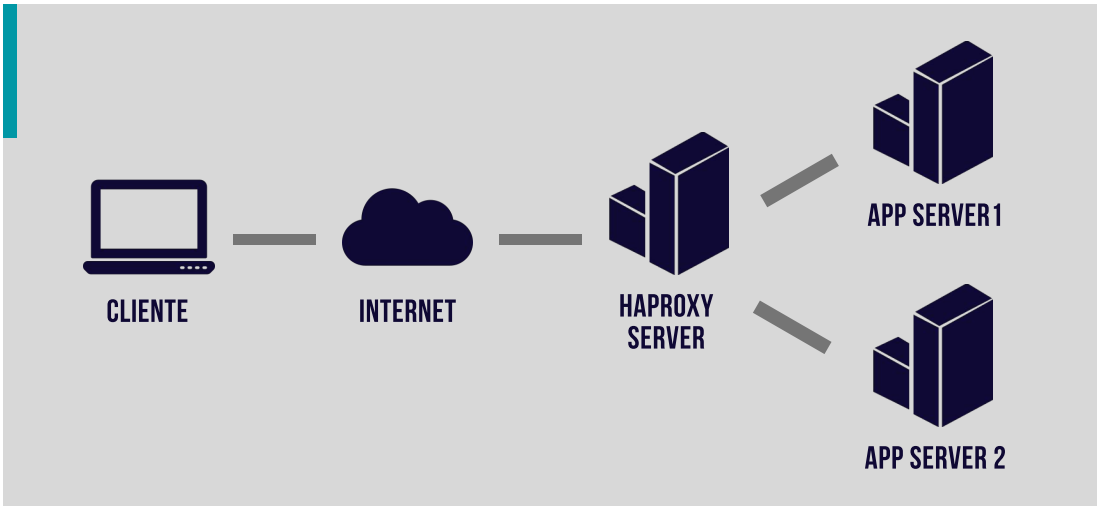
1. Você possui **3 VMs Ubuntu 22.04 Server** instaladas e com acesso à rede.

2. As VMs conseguem se **comunicar entre si** na rede configurada no ambiente de virtualização.

3. Você tem acesso ˜sudo˜ em todas as VMs.

4. Para este exemplo, um **serviço web simples (Apache)** será configurado nos servidores de aplicação para demonstrar o balanceamento.

CENÁRIO DE CONFIGURAÇÃO DO AMBIENTE



NOMENCLATURA E IPS (EXEMPLO)

Ajuste os IPs conforme a configuração da sua rede virtual. Recomenda-se o uso de IPs estáticos.

SERVIDOR HAPROXY (GERENCIADOR):

Hostname: **haproxy-manager**

IP: 192.168.X.100 (Substitua pelo seu segmento de rede)

SERVIDOR DE APLICAÇÃO 1:

Hostname: **appserver1**

IP: 192.168.X.101

SERVIDOR DE APLICAÇÃO 2:

Hostname: **appserver2**

IP: 192.168.X.102

PARTE 1

CONFIGURAÇÃO DOS SERVIDORES DE APLICAÇÃO

(APPSERVER 1 E APPSERVER2)

Execute os seguintes passos em appserver1 e appserver2:

1. ATUALIZAR O SISTEMA

```
sudo apt update
sudo apt upgrade -y
```

2. INSTALAR UM SERVIDOR WEB (EX: APACHE2):

```
sudo apt install apache2 -y
```

3. CRIAR PÁGINAS DE TESTE DISTINTAS:

EM APPSERVER1:

```
echo "<marquee bgcolor=yellow<h1>Servidor de Aplicacao 1 -
$(hostname)" | sudo tee /var/www/html/index.html
```

EM APPSERVER2:

```
echo "<marquee direction= right bgcolor=blue>Servidor de Aplicacao 2 -
$(hostname)" | sudo tee /var/www/html/index.html
```

4. INICIAR E HABILITAR O SERVIÇO APACHE:

```
sudo systemctl start apache2
sudo systemctl enable apache2
```

5. VERIFICAR O APACHE (OPCIONAL):

Acesse `http://IP_DO_APPSERVER1` e `http://IP_DO_APPSERVER2` de um navegador ou use curl:

```
curl http://IP_DO_APPSERVER1 # Ex: curl [http://192.168.] (http://192.168.)X.101
curl http://IP_DO_APPSERVER2 # Ex: curl [http://192.168.](http://192.168.)X.102
```

6. (OPCIONAL) CONFIGURAR FIREWALL (UFW):

Se o UFW estiver ativo, permita tráfego HTTP:

```
sudo ufw allow 80/tcp # ou 'Apache'
sudo ufw reload       # Se já estiver habilitado
sudo ufw enable       # Para habilitar, caso não esteja
```

PARTE 2

CONFIGURAÇÃO DOS SERVIDORES HAPROXY

(HAPROXY-MANAGER)

Execute os seguintes passos em haproxy-manager:

1. ATUALIZAR O SISTEMA

```
sudo apt update
sudo apt upgrade -y
```

2. INSTALAR O HAPROXY:

```
sudo apt install haproxy -y
```

3. CONFIGURAR O HAPROXY:

Faça um backup do arquivo de configuração original:

```
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bkp_$(date +%F)
```

Edite o arquivo sudo nano /etc/haproxy/haproxy.cfg e substitua o conteúdo pelo seguinte, ajustando os IPs dos appserver conforme necessário:

```
global
    log /dev/log    local0
    log /dev/log    local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

defaults
    log     global
   mode    http
   option  httplog
   option  dontlognull
   timeout connect 5000
   timeout client  50000
   timeout server  50000

frontend http_frontend
   bind *:80                               # HAProxy escuta na porta 80
   default_backend web_servers                 # Encaminha para o backend 'web_servers'

backend web_servers
   balance roundrobin                        # Algoritmo de balanceamento
   option httpchk GET /                      # Checagem de saúde HTTP
   # option forwardfor                       # Adiciona X-Forwarded-For
   # http-request set-header X-Forwarded-Proto https if { ssl_fc }

   # Substitua pelos IPs REAIS dos seus servidores de aplicação!
   server appserver1_vm 192.168.34.112:80 check    # 'check' habilita health check
   server appserver2_vm 192.168.34.122:80 check

# Opcional: Painel de Estatísticas do HAProxy
listen stats
   bind *:8404                             # Porta para o painel
   stats enable
   stats uri /haproxy_stats                  # URL do painel
   stats realm 'HAProxy Statistics
 '  stats auth admin:Acm123!@#      # MUDE ESTA SENHA!
   # stats admin if TRUE                     # Permite gerenciar backends via painel
```

ATENÇÃO:

Altere 192.168.X.101 e 192.168.X.102 para os IPs reais dos seus appserver1 e appserver2.

MUDE** a senha admin:SuaSenhaSuperSeguraAqui! para algo forte.

4. VERIFICAR A SINTAXE DA CONFIGURAÇÃO:

```
sudo haproxy -c -f /etc/haproxy/haproxy.cfg
```

A saída deve ser: **"Configuration file is valid"**.

5. REINICIAR E HABILITAR O HAPROXY:

```
sudo systemctl restart haproxy
sudo systemctl enable haproxy
```

6. VERIFICAR O STATUS DO HAPROXY:

```
sudo systemctl status haproxy
```

Deverá mostrar **"active (running)"**.

7. (OPCIONAL) CONFIGURAR FIREWALL (UFW) NO SERVIDOR HAPROXY:

Permita tráfego nas portas 80 e 8404 (se o painel estiver habilitado):

```
sudo ufw allow 80/tcp
sudo ufw allow 8404/tcp # Se habilitou o painel
sudo ufw reload
sudo ufw enable
```

PARTE 3

TESTANDO A ALTA DISPONIBILIDADE

1. ACESSAR O SERVIÇO VIA HAPROXY:

Abra um navegador e acesse o IP do seu servidor `haproxy-manager`:

http://IP_DO_HAPROXY_MANAGER (Ex: `http://192.168.X.100`)

Atualize a página várias vezes. O conteúdo deve alternar entre **"Servidor de Aplicacao 1"** e **"Servidor de Aplicacao 2"**.

2. TESTAR O PAINEL DE ESTATÍSTICAS (SE HABILITADO):

Acesse: http://IP_DO_HAPROXY_MANAGER:8404/haproxy_stats

(Ex: `http://192.168.X.100:8404/haproxy_stats`)

Use o usuário e senha configurados. Ambos os backends (`appserver1_vm`, `appserver2_vm`) devem estar "UP" (verdes).

2. SIMULAR UMA FALHA:

Pare o Apache em appserver1:

 # No appserver1

```
sudo systemctl stop apache2
```

3. RESTAURAR O SERVIDOR:

Inicie o Apache em appserver1:

No appserver1

```
sudo systemctl start apache2
```

CONSIDERAÇÕES ADICIONAIS IMPORTANTES

ALGORITMOS DE BALANCEAMENTO:**

* **`roundrobin`: Distribuição sequencial.**
* **`leastconn`: Envia para o servidor com menos conexões ativas.**
* **`source`: Persistência baseada no IP de origem.**

HEALTH CHECKS AVANÇADOS:**

Configure verificações mais específicas (arquivo, status HTTP esperado).

PERSISTÊNCIA DE SESSÃO (STICKY SESSIONS):**

Essencial para aplicações que guardam estado da sessão no servidor. Pode ser implementada com cookies.

Exemplo no `backend`:

```
# backend web_servers
# ... (outras configs)
# cookie SERVERID insert indirect nocache
# server appserver1_vm 192.168.X.101:80 check cookie s1
# server appserver2_vm 192.168.X.102:80 check cookie s2
```

SSL/TLS TERMINATION:

HAProxy pode gerenciar certificados SSL/TLS, descriptografando HTTPS e enviando HTTP para os backends.

REDUNDÂNCIA DO HAPROXY:

O servidor `haproxy-manager` é um Ponto Único de Falha (SPOF). Para produção, considere um cluster de HAProxy com **Keepalived** para um IP virtual flutuante (VIP).

LOGGING DETALHADO:

Configure o `rsyslog` para separar os logs do HAProxy em arquivos dedicados.

LOGGING DETALHADO:

Configure o `rsyslog` para separar os logs do HAProxy em arquivos dedicados.

RECURSOS:

Monitore CPU, memória e rede nos servidores.

www.cecyber.com