TI no Dia-a-Dia << http://tinodiaadia.wordpress.com !

Failover e balanceamento de carga de aplicações HTTP em servidores Linux.

Failover e balanceamento com Keepalived + Apache.

Heitor Felipe Lessa
29/11/2010

ÍNDICE DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

Este documento visa demonstrar a configuração do *keepalived* para o *failover* entre os SO´s (Linux) e configuração do Proxy reverso e balanceamento de carga das requisições HTTP (Apache), de forma a manter o ambiente operacional em caso de queda de um dos servidores, mantendo um serviço HTTP ou qualquer que seja em funcionamento utilizando *failover*. Este trabalho foi pode ser feito em qualquer distribuição Linux, desde que haja ferramentas de compilação e o kernel seja compatível ou posterior á versão 2.6. Os requisitos para este processo são os softwares *keepalived* e Apache, sendo que, as versões utilizadas estão explicitas em suas respectivas seções deste documento.

2. KEEPALIVED

O software *Keepalived* pode ser obtido em sua página oficial em [1], a última versão estável utilizada neste documento foi a 1.1.20, portanto não é garantido o funcionamento correto em versões inferiores ou posteriores onde o core pode ser drasticamente modificado.

A instalação é simples, descompacte o tarball em um diretório temporário, recomenda-se que se descompacte no diretório "/tmp" ou em "/opt", após a instalação segue seu padrão compilando o pacote com os comandos:

 # ./configure
 # make
 # make install

A compilação pode ser feito como usuário comum, porém a instalação (make install) deve ser feito como "root".

O script de inicialização do *keepalived* nesta versão não funcionou 100% em testes, portanto crie um arquivo chamado "*keepalived*" no diretório "/etc/init.d" e cole o conteúdo do arquivo "script_*keepalived*" (contido no diretório desta documentação).

Crie um link simbólico no diretório de seu init corrente (como por exemplo /etc/rc5.d em distribuições RedHat ou /etc/rc2.d em distribuições Debian, mais informações consulte o arquivo /etc/inittab).

 # ln -s /etc/init.d/*keepalived* /etc/rc5.d/S99*keepalived*

Neste momento, vamos configurar o arquivo de configuração do *keepalived*, localizado em "/etc/*keepalived*/*keepalived*.conf" (caso o diretório não exista, crie-o), copie exatamente como na figura 1, trocando somente as definições de e-mail e endereço IP de acordo com seu ambiente:

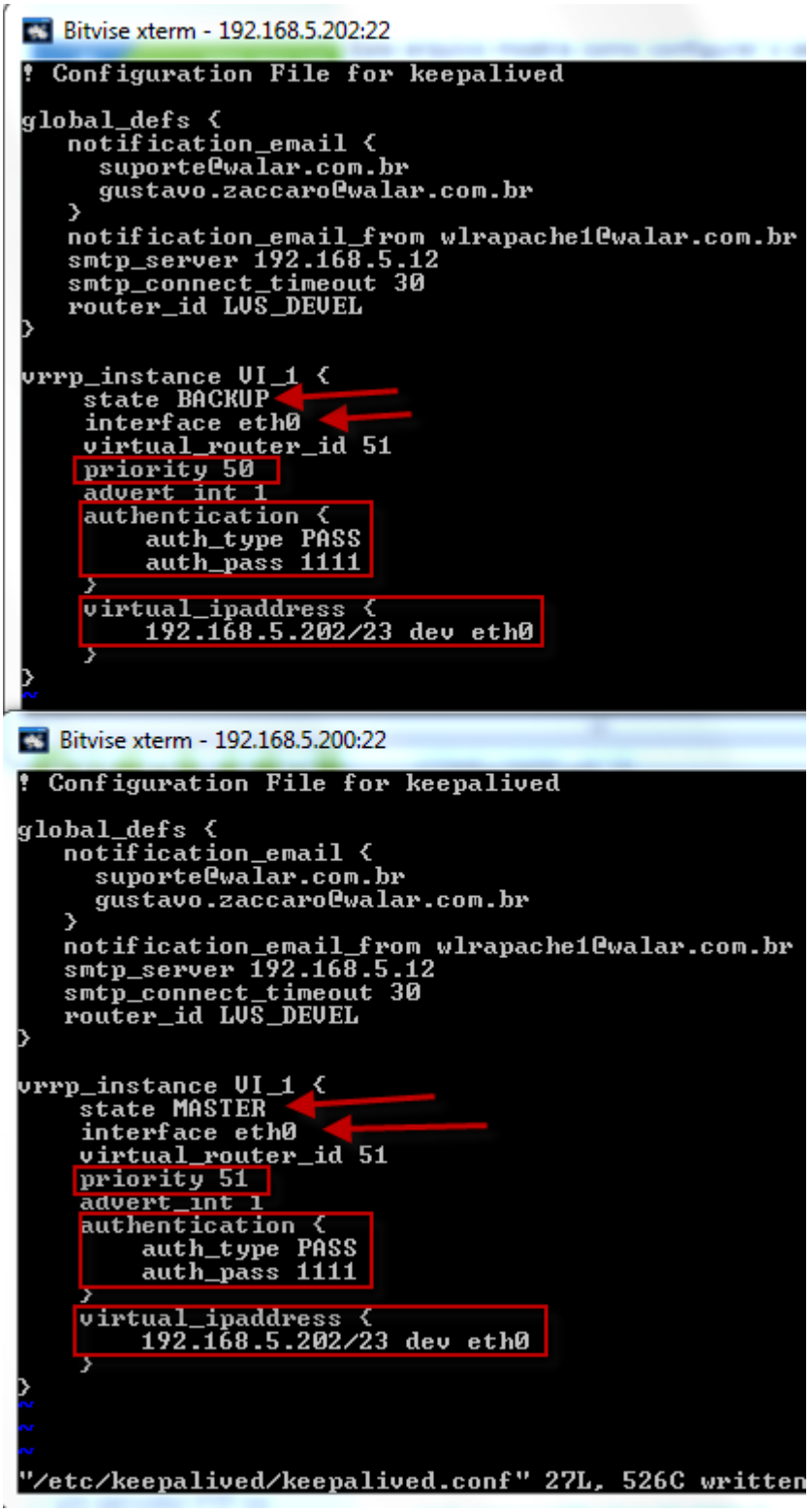

Figura 1 - Arquivo de configuração *Keepalived*

Atente-se aos campos marcados na figura, segue os detalhes destes parâmetros:

- **State** – Esta opção define quem será o servidor *MASTER* responsável por receber o tráfego e o servidor *BACKUP* que assumirá em caso de falhas no *MASTER*;
- **Interface** – Esta opção define qual será a interface responsável pelo uso do VRRP [2];
- **Priority** – Esta opção deve ter o valor maior para o servidor *MASTER* e menor para o *BACKUP*;
- **Authentication** – Esta opção define qual será a tipo de autenticação (PASS definindo do tipo senha) e a senha utilizada na troca das informações, a senha deve ser a mesma nos servidores *MASTER* e *BACKUP*;
- **Virtual_ipaddress** – Esta opção define qual será o IP virtual que será compartilhado entre os servidores *MASTER* e *BACKUP*, assim sendo deve ser também especificado em qual interface o ip virtual irá residir.

Agora que o serviço está configurado vamos iniciá-lo através do script que criamos, portanto execute-o com o comando:

 # /etc/init.d/*keepalived* start

Os avisos e alertas são gerados no log padrão do SO disponível em "/var/log/messages", após iniciar o *keepalived* consulte este arquivo e verifique por erros, mas caso ocorra tudo ok, irá encontrar a seguinte mensagem como na figura 2:

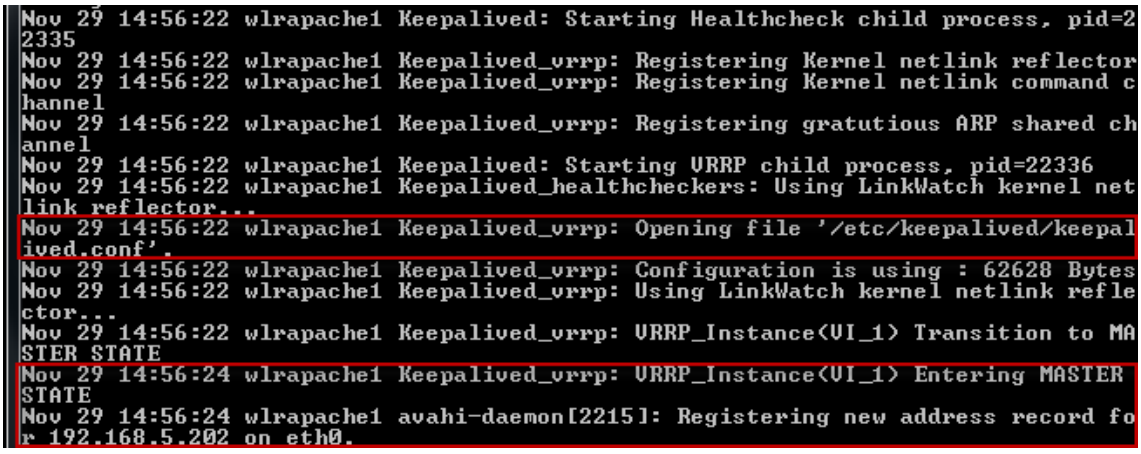

Figura 2 - Log de inicialização *Keepalived*

Repare na linha “Opening file”, caso seu arquivo não esteja neste local, a configuração não irá funcionar da forma correta. Nas linhas finais exibe-se que o servidor está entrando no estado *MASTER*, definido anteriormente em nosso *keepalived*.conf, e, como última linha o endereço de ip virtual “192.168.5.202” é registrado na interface “eth0”.

Faça o mesmo processo no segundo servidor e terá a mesma mensagem, porém o STATE será trocado para *BACKUP*. Para confirmar se o endereço foi realmente registrado e está em funcionamento, digite o comando:

 # ip addr show

O comando acima vai mostrar todos os endereços IP´s configurados nas interfaces do servidor, neste caso, o endereço com final 202 deve estar disponível na listagem conforme a figura 3:

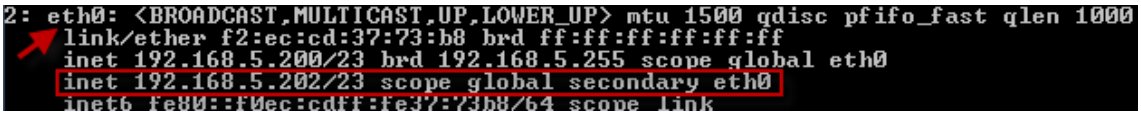

Figura 3 - Endereço IP virtual.

Repare que ele é definido como secundário e também está dentro de nossa rede com o CIDR /23, atente-se a isto para que o endereço seja válido e não somente disponível ao próprio servidor como um endereço com CIDR /32

Para testar a funcionalidade do *failover* do *keepalived*, inicie o serviço tanto no servidor *MASTER* quanto no *BACKUP*, após execute um ping apontando para o endereço de IP virtual dos servidores, feito isso baixe o serviço *keepalived* do servidor *MASTER* e verás que o log do serviço mostrará algumas informações a respeito da queda, para tal segue figura 4:

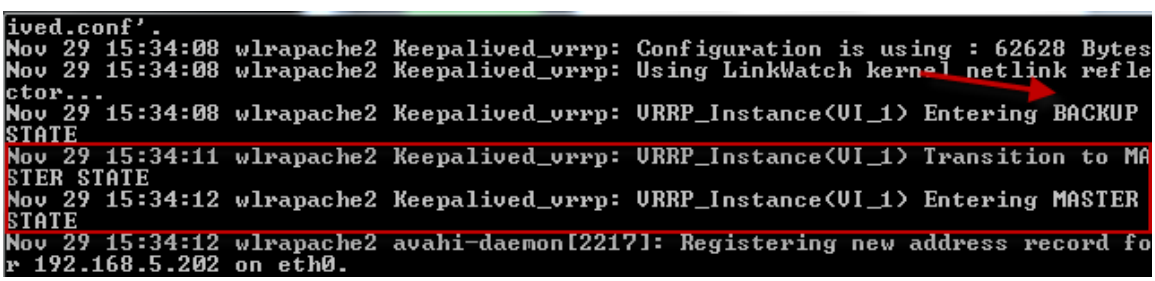

Figura 4 - Transição do estado *BACKUP* para *MASTER*.

O serviço foi iniciado no estado de *BACKUP*, mas assim que o serviço caiu no servidor *MASTER* aos “15:34:11”, a transição para o estado *MASTER* foi feita automaticamente e todas as requisições são redirecionadas á este servidor.

Pronto, desta forma terminamos a configuração do *keepalived* e os testes, caso o script não pare o serviço corretamente devido ao uso do kill manual do processo, apague o lock contido em “/var/lock/subsys/*keepalived*”, assim o script volta a funcionar corretamente.

Para maiores informações sobre o software *keepalived*, consulte a documentação oficial em [4].

3. APACHE

A instalação do Apache não será coberta por este documenta, pois envolve diversas peculiaridades particulares de cada ambiente, mas o que precisamos de antemão é que os módulos “mod_proxy”, “mod_proxy_balancer” e “mod_rewrite” estejam compilados e instalados em seu ambiente, além disso, a versão do Apache utilizada é a 2.2, portanto pede-se por maior compatibilidade que se utilize a versão 2.x e que os módulos estejam disponíveis.

Este parte de configuração parte do princípio que o leitor já conheça os conceitos de Proxy reverso, balanceamento de carga e a utilização dos módulos mod_proxy e balancer, portanto caso não tenha conhecimento sobre estes, podes iniciar a pesquisa sobre VirtualHosts e Proxy reverso com mod_proxy[5] e o *load balancer* com mod_proxy_balancer[6].

A configuração deste apache é rápida, portanto somente copie e cole em httpd.conf (ou seu arquivo principal de configuração, varia da forma como a compilou) e mude os parâmetros de acordo com seus nomes DNS, nos demais itens, deixe-os da mesma forma:

```
<VirtualHost *:80>
RewriteEngine On
ProxyRequests Off
RewriteCond %{REQUEST_METHOD} !^(GET|HEAD|OPTIONS|POST)$
RewriteRule .* - [F]
<Proxy balancer://nfe>
BalancerMember http://host0:38080 route=r0 loadfactor=1
BalancerMember http://host1:38080 route=r1 loadfactor=1
</Proxy>
ProxyPass / balancer://nfe/ stickysession=JSESSIONID nofailover=Off
ProxyPassReverse / balancer://nfe/
ProxyPreserveHost On
</VirtualHost>
```

Nas linhas em vermelho, deve-se mudar os hosts0 e 1 e suas devidas portas configuradas em seu *Application Server (Glassfish* no caso), os valores route=r0 e router=r1 devem ser os mesmos nas aplicações, adicionadas como “*Properties”* no glassfish com o nome “jvmRoute” e o valor de “r1”, caso não seja feito, a sessão não será mantida com o usuário e pode haver queda de sessões e um novo login deverá ser feito á cada nova navegação.

É importante lembrar que o nome da aplicação que será instalada no *Glassfish* chama-se “nfe”, portanto o nome do balancer “nfe”, além disso, deve prestar atenção na última “/” da linha “ProxyPassReverse”, pois a aplicação encontra-se na raiz do servidor de aplicação.

Uma dica importante é que se sua aplicação PHP não funcionar corretamente utilizando este método, recorra a este HOW-TO[7].

Para fins de confirmação, segue na figura 5 o teste de acesso da aplicação utilizando toda a estrutura criada até aqui:

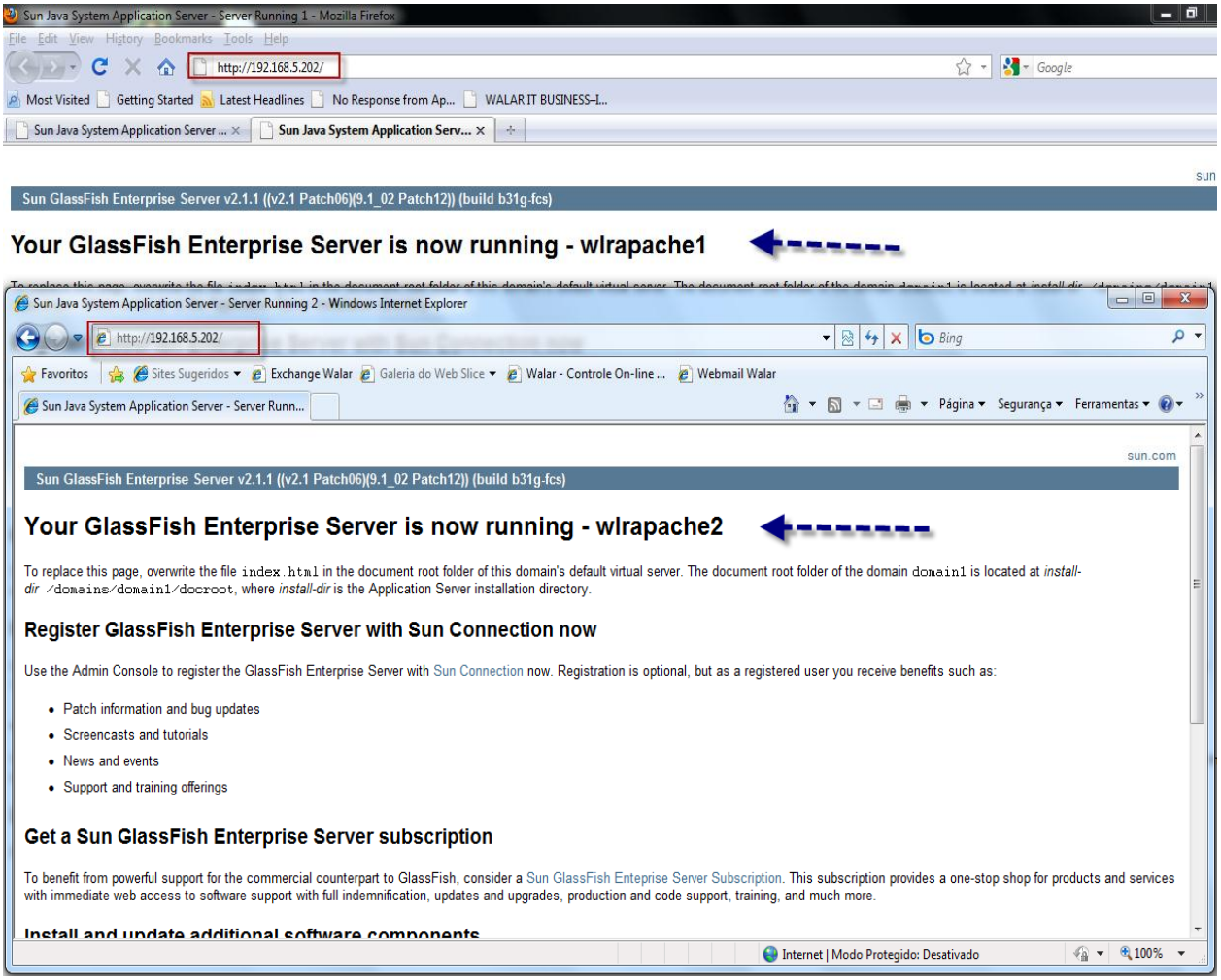

Figura 5 - Balanceamento de carga

Repare que foram abertas duas sessões em diferentes navegadores, assim, foi possível verificar que a cada sessão aberta ele faz o balanceamento para servidores descritos na linha “BalancerMember”, além de que utilizamos o endereço virtual criado pelo *Keepalived*,

garantindo também a alta disponibilidade em caso de falha de um dos servidores, pois neste momento todo o tráfego é direcionado para um só apache e ele faz o balanceamento de carga para os servidores *glassfish* (neste exemplo, utilizamos o mesmo servidor Apache para o *glassfish*), mas neste caso teríamos um ponto de falha caso tivéssemos somente um Apache para o balanceamento, portanto, a necessidade do *Keepalived* efetuando o *failover* em caso de queda, repassando todo o tráfego para outro Apache.

Desta forma, temos uma solução transparente ao usuário, altamente disponível e com alto desempenho, de forma que a disponibilidade é gerida pelo *Keepalived*, o balanceamento feito pelo Apache e o *cluster* feito pelo *glassfish*, além disso, temos um único endereço IP virtual para os dois servidores, de forma que o usuário somente precisa acessar um nome, como por exemplo: sistema.empresa.com.br, este nome DNS apontará para o endereço virtual e será transparente ao usuário sobre qual servidor ele está acessando.

REFERÊNCIAS

[1] *Keepalived* Download - http://www.*keepalived*.org/download.html

[2] Virtual Router Redundancy Protocol (VRRP) - http://www.faqs.org/rfcs/rfc2338.html

[3] Classless Inter-Domain Routing (CIDR) - http://www.faqs.org/rfcs/rfc1817.html

[4] User Guide *Keepalived* - http://www.*keepalived*.org/pdf/UserGuide.pdf

[5] Como montar proxy reverse no servidor Apache - http://www.vivaolinux.com.br/artigo/Como-montar-um-proxy-reverse-no-servidor-Apache?pagina=3

[6] Session-Aware Loadbalancer using mod_proxy_balancer – http://www.howtoforge.com/load_balancing_apache_mod_proxy_balancer

[7] Mod_proxy balancing with PHP sticky session - http://www.markround.com/archives/33-Apache-mod_proxy-balancing-with-PHP-sticky-sessions.html